University of Southern Queensland

Combining Web Data Mining Techniques for Web Page Access Prediction

A Dissertation submitted by

Faten Khalil M.IT.

For the award of

Doctor of Philosophy

2008

Abstract

Web page access prediction gained its importance from the ever increasing number of e-commerce Web information systems and e-businesses. Web page prediction, that involves personalising the Web users' browsing experiences, assists Web masters in the improvement of the Web site structure and helps Web users in navigating the site and accessing the information they need. The most widely used approach for this purpose is the pattern discovery process of Web usage mining that entails many techniques like Markov model, association rules and clustering. Implementing pattern discovery techniques as such helps predict the next page to be accessed by the Web user based on the user's previous browsing patterns. However, each of the aforementioned techniques has its own limitations, especially when it comes to accuracy and space complexity. This dissertation achieves better accuracy as well as less state space complexity and rules generated by performing the following combinations. First, we combine low-order Markov model and association rules. Markov model analysis are performed on the data sets. If the Markov model prediction results in a tie or no state, association rules are used for prediction. The outcome of this integration is better accuracy, less Markov model state space complexity and less number of generated rules than using each of the

methods individually. Second, we integrate low-order Markov model and clustering. The data sets are clustered and Markov model analysis are performed on each cluster instead of the whole data sets. The outcome of the integration is better accuracy than the first combination with less state space complexity than higher order Markov model. The last integration model involves combining all three techniques together: clustering, association rules and low-order Markov model. The data sets are clustered and Markov model analysis are performed on each cluster. If the Markov model prediction results in close accuracies for the same item, association rules are used for prediction. This integration model achieves better Web page access prediction accuracy, less Markov model state space complexity and less number of rules generated than the previous two models.

Certification of Dissertation

I certify that the work reported in this thesis is entirely my own effort, except where otherwise acknowledged. This report is original and contains no material offered for the award of any other academic qualification at this or any other institution, or material previously published, except where due reference is made in the text.

_____ _____

Signature of Candidate Date

Student Number: 50042419

Acknowledgements

I heartily thank my former principal supervisor Dr Jiuyong Li for his research direction of this project, his support, patience and guidance. Dr Li, provided me with clear direction and encouragement, thoughtful and constructive feedback on my work in a timely manner, as well as assistance and advice when needed even after he left USQ. I sincerely thank my current supervisor, Dr Hua Wang, for his continuous guidance, encouragement, valuable insights and support. Dr Wang was always there for me when I needed assistance and advice both as my associate supervisor as well as my principal supervisor later on. My gratitude also goes to the Department of Mathematics and Computing, University of Southern Queensland (USQ) for the financial support provided for attending conferences. I am extremely grateful to the Head of Department Professor Ron Addie and his secretary Ms Helen Nkansah. Lastly, but not least, I greatly appreciate the support from my parents and my family. Their care and love are what keeps me going during hard times.

Publications

The publications arising from this dissertation are as follows:

1. Khalil, F., Li, J. and Wang, H. (2006). A Framework of Combining Markov Model with Association Rules for Predicting Web Page Accesses. In Proc. Australasian Data Mining Conference (AusDM'06), Sydney, Australia. CRPIT, 61. pp 177-184.

2. Khalil, F., Li, J. and Wang, H. (2007). Integrating Markov Model with Clustering for Predicting Web Page Accesses. In Proc. Australian World Wide Web (AusWeb'07), Coffs Harbour, Australia. pp 63-74.

3. Khalil, F., Li, J. and Wang, H. (2008). Integrating Recommendation Models for Improved Web page prediction accuracy. In Proc. Thirty-First Australasian Computer Science Conference (ACSC'08), Wollongong, Australia. CRPIT, 74. pp 91-100.

4. Khalil, F., Li, J. and Wang, H. (In progress). Improving Web Page Prediction Accuracy by Integrating Markov model, Association rules and Clustering. Journal submission.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Data mining research interest is the result of the vast amount of data that forms part of our daily activities. Web data mining gains its importance with the increasing amount of Web information that is becoming much larger than any traditional data sources. Web data mining involves applying data mining techniques to Web data. It focuses on the Web pages link structure, their content and their usage. Web usage mining concentrates on tools and techniques used to predict users' navigational paths by discovering their Web access patterns. It includes three stages: preprocessing, pattern discovery and pattern analysis. The field of pattern discovery has been a major study for improving the efficiency of numerous Web based applications including e-commerce. Web applications today 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 users we are dealing with and the users preferences as well. Pattern discovery achieves this by extracting useful knowledge and patterns applying different tools and techniques. Some of these tools are association rules, clustering and Markov models. Each of these pattern discovery techniques has its own strengths and weaknesses. Discovered patterns of accessed Web pages helps predict the next page to be accessed by the user.

1.1 Research Objectives

The immense volume of online information covering almost all types of applications turns the Web into a huge mine that is susceptible to a wide range of information discovery and retrieval tools. There has been a lot of research covering advanced data mining techniques to extract useful knowledge from large amount of data. Association rules, clustering and Markov models have been widely used for this purpose.

Association rule mining is a major pattern discovery technique Mobasher et al. (2000), Agrawal & 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 sum-

marise relationships illustrated by the following example. Customers who buy bread and milk will most likely buy eggs, or, bread and milk → eggs. Association rules are mainly defined by two metrics: support and confidence. The main limitation of association rules is that they tend to generate many rules, which result in contradictory predictions for a user session. Markov models are also becoming very commonly used in the identification of patterns based on the sequence of previously accessed items Bouras & Konidaris (2004), Chen et al. (2002), Deshpande & Karypis (2004), Eirinaki et al. (2005), Jespersen et al. (2003), Sarwar et al. (2001), Zhu et al. (2002*a*,*b*). However, 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 is defined as the classification of patterns into groups (clusters) based on similarity between common activities Adami et al. (2003), Cadez et al. (2003), Strehl et al. (2000). The main clustering limitation is that clustering methods are unsupervised methods, and normally are not used for classification directly.

Each of the mentioned pattern discovery techniques has been widely used for Web page access prediction, as discussed in later chapters. 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 state space complexity to a minimum. So far, the in-

dividual implementation of these tools fails to achieve higher prediction accuracy together with lower state space complexity. This dissertation aims at improving the Web page access prediction accuracy while keeping the state space complexity to a minimum by using different integration models based on Markov models, association rules and clustering and according to certain constraints.

1.2 Dissertation Structure

The dissertation consists of seven chapters. Chapter 2 gives an insight into the background of data mining, Web data mining and Web usage mining in particular and their importance in Web page access prediction and applications. Chapter 3 introduces the three Web usage mining tools: Markov model, association rules and clustering emphasizing on their importance in Web page prediction and listing their limitations. Chapter 4 presents a new model that integrates Markov model with association rules. Association rule mining is only applied when Markov model implementation results in no states or in two or more similar conditional probabilities forming a tie. Chapter 5 integrates both Markov model and clustering techniques. During the prediction process, the new state is assigned to an appropriate cluster and Markov model is implemented on that particular cluster only. Chapter 6 further integrates the three models together: Markov model, clustering and association rules. During the prediction process, after assigning the new state to its cluster, if the Markov model implementation results in no state or states that

do not belong to the majority class, association rules are implemented. Chapter 7 concludes our work emphasizing its importance and listing some limitations and future directions.

Figure 1.1 summarises the structure of the dissertation chapters where MM stands for Markov model, AR stands for association rules and Clust stands for clustering techniques.

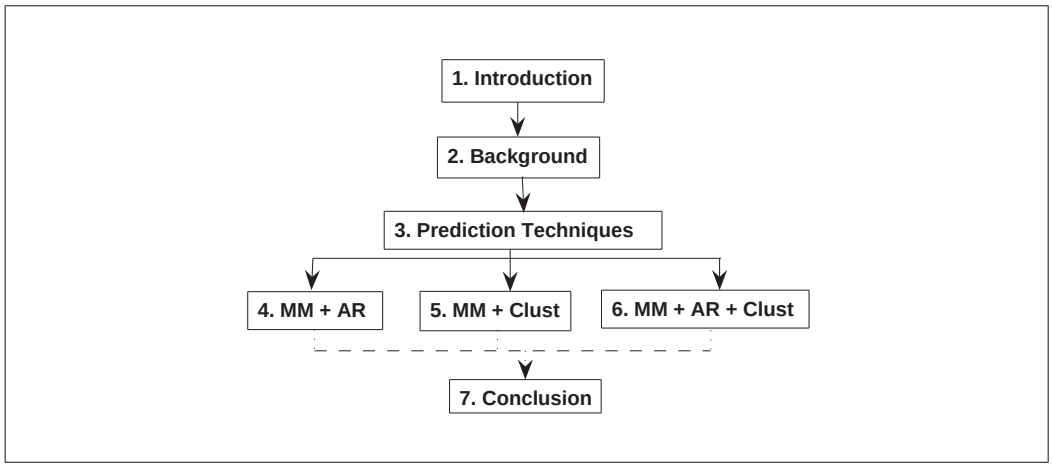

Figure 1.1: Dissertation structure.

1.3 Applications

The main motivation behind this dissertation is the correlation between Web usage mining and Web personalisation. The work on Web usage mining can be a source of ideas and solutions towards realizing Web personalisation. The ultimate goal of Web personalisation is to provide Web users with the next page they will access in a browsing session. This achieved by analysing their browsing patterns and

comparing the discovered patterns to similar patterns in history. Traditionally, this has been used to support the decision making process by Web site operators in order to gain better understanding of their visitors, to create a more efficient structure of the Web sites and to perform a more effective marketing. A number of other functions can be provided in the following areas as a result of Web page access prediction including, but not limited to:

- Guiding the Web site users by providing them with recommendations of a set of hyperlinks that are related to the users' interests and preferences and improve the users navigational experience and providing users with personalised and customised page layout, hyperlinks and content depending on their interests and preferences.

- Performance of the system of some actions on behalf of users such as sending e-mail, downloading items, completing or enhancing the users' queries, or even participating in Web auctions on behalf of Web users.

- Learning and predicting user clicks in Web based search facilities Zhou et al. (2007). This offers an automated explanation of Web user activity. Also, the measurement of the likelihood of clicks can infer a user's judgement of search results and improve Web page ranking.

- Minimizing latency of viewing pages especially image files, by pre-fetching Web pages or by pre-sending documents that a user will visit next Yang

et al. (2003), Pons (2006). Despite the apparent similarity between Web pre-fetching and Web caching Chen et al. (2002), Bouras & Konidaris (2004), Web pre-fetching goes one step further by anticipating the Web users' future requests and pre-loading the predicted pages into a cache. This is a major method to reduce Web latency which can be measured as the difference between the time when a user makes a request and when the user receives the response. Web latency is particularly important to Web surfers e-commerce Web sites Su et al. (2000), Zuckerman et al. (1999).

- Customizing Web site interfaces by predicting the next relevant pages or products and overcoming the information overload by providing multiple short-cut links relevant to the items of interest in a page Chen et al. (2003), Deshpande & Karypis (2004, 2001), Yan et al. (1996).

- Improving site topology as well as market segmentation.

- Improving the Web advertisement area where a substantial amount of money is paid for placing the correct advertisements on Web sites. Using Web page access prediction, the right ad will be predicted according to the users' browsing patterns.

Chapter 2

Background

2.1 Introduction

The ongoing increase in the amount of Web data has led to the explosive growth of Web data repositories. Web pages and their contents are accessed and provided by a wide variety of applications and they are added and deleted every day. Moreover, the Web does not provide its users with a standard coherent page structure across Web sites. These facts make it very difficult to analyze the content of Web pages by automated tools. Therefore, there arises a need for Web data mining techniques.

Data mining involves the study of data-driven techniques to discover and model hidden patterns in large volumes of raw data. The application of data mining techniques to Web data is referred to as Web data mining. Web data mining can be

divided into three distinct areas: Web content mining, Web structure mining and Web usage mining. Web content mining involves efficiently extracting useful and relevant information from millions of Web sites and databases. Web structure mining involves the techniques used to study the Web pages schema of a collection of hyper-links. Web usage mining on the other hand, involves the analysis and discovery of user access patterns from Web servers in order to better serve the users' needs. Figure 2.1 below summarises the processes involved in each of the Web data mining phases.

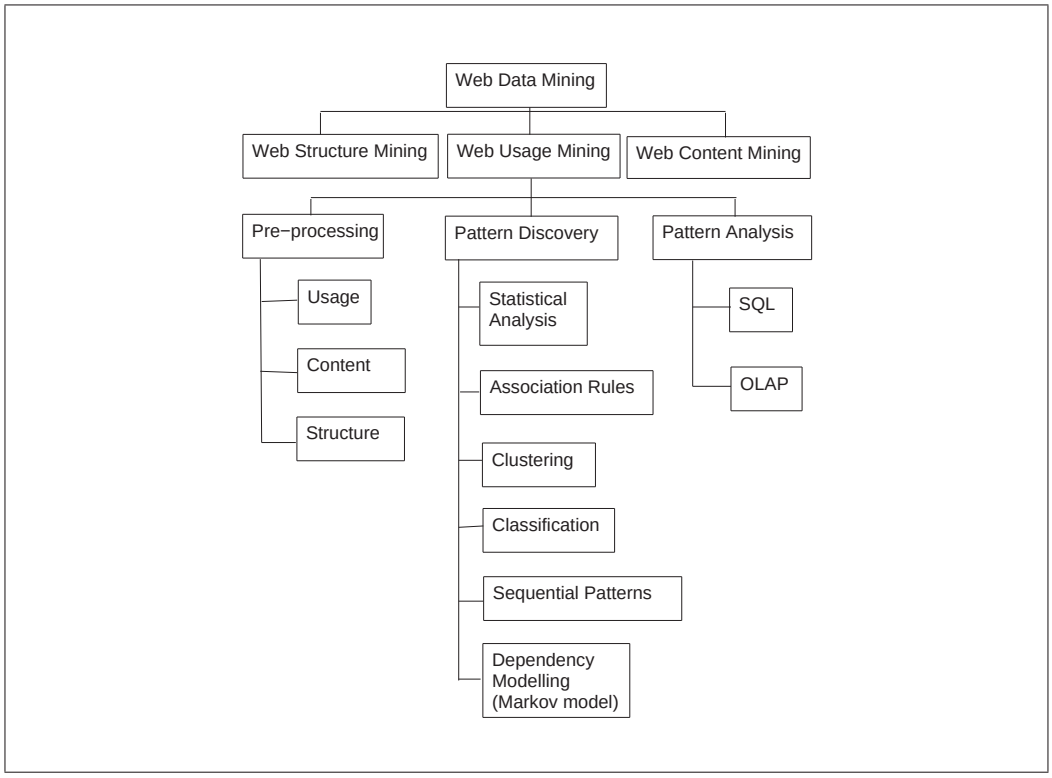

Figure 2.1: Web data mining architecture.

2.2 Web Data Mining

With the advancement in computer technologies, there has been a major need to apply data mining techniques to electronic data as the amount of information stored is increasing at a very high pace. This makes traditional data analysis tools very limited in making use of stored data for businesses to gain an edge over their competitors. Data mining is the analytical process to explore large amounts of data with the purpose of extracting patterns Cooley et al. (1997). The data mining process is composed essentially of three different stages: data preparation which is the initial exploration, search for patterns or pattern identification, and the knowledge interpretation and evaluation that involves deployment of patterns to new data.

Web data mining is the process of applying data mining techniques to Web data. Research in this area has the objectives of helping e-commerce businesses in their decision making, assisting in the design of good Web sites and assisting the user when navigating the Web. The World Wide Web data mining focuses on three issues: Web structure mining, Web content mining and Web usage mining.

2.2.1 Web Structure Mining

Web structure mining aims at generating structured summary about Web sites and Web pages in order to identify relevant documents. The focus here is on link

information, which is an important aspect of Web data. Web structure mining can be used to reveal the structure or schema of Web pages which would facilitate Web document classification and clustering on the basis of its structure Spertus (1997).

Web structure mining is very useful in generating information such as visible Web documents, luminous Web documents and luminous path which is the path common to most of the results returned Bhowmick et al. (1998). Another related work in the area of schema discovery of structured and semi-structured documents is apparent in Wang & Liu (1998, 1997).

2.2.2 Web Content Mining

Web content mining involves mining Web data contents. It focuses on various techniques that assist in searching the Web for documents whose content meets a certain goal. Those documents, once found, are used to build a knowledge base. The emphasis here is on analysing the Internet hypertext material. The Internet data that is available in digital form has to be prepared for analysis. A large number of researches have been conducted in this area in the past few years. For instance, Zaiane & Han (2000), focused on resource recovery on the Web. The authors made use of a multi-layered database model to transform the unstructured data on the Web into a form acceptable by database technology. Moreover, several intelligent search agents, information filtering/categorisation and personalised

Web agents have been developed for information retrieval and for organisation of structured and semi-structured information on the Web Cooley et al. (1997).

2.2.3 Web Usage Mining

Web usage mining involves the automatic discovery and analysis of patterns in data as a result of the user's interactions with one or more Web sites. It focuses on tools and techniques used to study and understand the users' navigation preferences and behavior by discovering their Web access patterns. These techniques are effective means that help e-commerce businesses improve their Web sites in an efficient manner Heer & Chi (2002). The goal of Web usage mining is to capture, model and analyse the users' behavioural patterns. It, therefore, involves three phases: Preprocessing of Web data, pattern discovery and pattern analysis Srivastava et al. (2000). Of these, only the latter phase is performed in real-time. The discovered patterns are represented as collections of pages that are frequently accessed by groups of users with similar interests within the same Web site.

2.2.3.1 Preprocessing

Before starting any mining technique, Web data has to be cleaned and preprocessed. Preprocessing prepares data for the pattern discovery stage. It transforms Web log files into Web transaction data that can be processed by data mining tasks. Web data could take many forms. The primary data sources are the server log files

that include Web server access logs and application server logs. Also, additional data sources may include operational databases, domain knowledge, site files and meta-data. This additional data can be available from client-side or proxy level data collection as well as from external clickstream or demographic data sources. The most important and the most easily accessed data is the Web server log report that keeps track of every single user access to the server. In general, the log entries include information like date, time, client IP, URL of the source, name of the script or file requested and the server status Zaiane et al. (1998). There are three types of preprocessing: usage preprocessing, content preprocessing and structure preprocessing Srivastava et al. (2000). Usage preprocessing is the most difficult task as it deals with the incomplete log entries and the wide usage of local caches and proxy servers. Often there is a need for using more accurate data from other sources like cookies or a client side collection method. Client side collection methods can get very complex like using a remote agent, such as Javascripts or Java applets, or by modifying the source code of the browser. Both of these methods require user cooperation. With usage preprocessing, the data usually needs to be transformed and aggregated at different levels of abstraction. The most basic level of data abstraction is pageview which represents a collection of Web objects displayed as a result of a single user action. A collection of pageviews for a single user during a single visit forms a session. Sessions may be used to analyse the user's behavioural browsing patterns.The second type of preprocessing is content preprocessing. It involves preparing text and multimedia files using classifica-

tion and clustering techniques. Static Web pages can be easily preprocessed by parsing the HTML and reformatting the information or by running additional algorithms. However, dynamic Web pages that are the result of database accesses or personalisation algorithms are usually more difficult to preprocess. Also, limiting preprocessing to certain pages that are generated by a combination of database accesses will not give definitive results. The third type of preprocessing is structure preprocessing. It consists of preprocessing the inter-page structure information or the Hyperlinks that connect one page to another. Again, pages that have a predefined structure are easily preprocessed. However, dynamically structured pages can be more difficult. Dynamic structure creates problems since a different site structure may have to be constructed for each server session.

2.2.3.2 Pattern Discovery

During this stage, algorithms are run on the data and patterns are extracted from it. Pattern discovery involves the employment of sophisticated techniques from artificial intelligence, data mining techniques, psychology and information theory in order to extract knowledge from collected and preprocessed data. Some of the most widely used pattern discovery approaches are statistical analysis, association rule mining, clustering, classification, sequential patterns and dependency modeling Srivastava et al. (2000). Statistical analysis techniques are the most common tools used to extract knowledge about Web site users. These tools could provide user information like the most frequently accessed pages, average time of viewing

a certain page, average time the user spends browsing a certain site etc... This type of knowledge can never have 100% accuracy as in most cases it is based on incomplete log reports. However, knowledge extracted using statistical analysis could be very useful for improving the system performance and for providing support for marketing purposes especially for e-commerce applications Cooley et al. (1999). Association rule mining refers to the sets of pages that are accessed together in a single server session. They are used to identify items that are likely to be purchased or viewed in a similar session. These rules are very helpful for marketing purposes. They also help Web designers improve their hyperlinks and reduce user latency when downloading a page Srivastava et al. (2000). Clustering aims at identifying a finite set of categories to describe a data set. It groups together data items with same characteristics. One type of clustering is usage clusters which involves finding users with same browsing habits. It is useful in providing personalized Web content to users. Another type of clustering is page clusters which discovers pages that have related content. This kind of clustering is very useful for Internet search engines. Classification aims at finding common properties among a set of objects and mapping those objects into a set of predefined classes Cooley et al. (1997). An example is the classification of the clients of an insurance company according to the probability of submitting a claim. Clients classified in the higher risk classes have to pay higher premiums. Sequential patterns is another type of pattern discovery. A sequence is defined as an ordered list of item and sets. Sequential pattern discovery attempts to find patterns such that

a set of items is followed by another item within a certain period of time and in a certain server session. This can be useful in predicting users future browsing patterns . Dependency modeling consists of techniques that are aimed at finding a model describing dependencies between variables in the Web domain. This is potentially useful for predicting future Web resources consumption. For example, it could help develop strategies to increase the sales of products offered by a Web siteSrivastava et al. (2000).

Pattern Analysis: Not all discovered patterns are useful and this step aims at identifying the patterns which represent new and potentially useful knowledge. Pattern analysis involves filtering out the unneeded patterns or rules discovered through the pattern discovery phase. The most common pattern analysis technique is the use of query language like SQL. Another technique could be the usage of online analytical processing (OLAP) tools Srivastava et al. (2000). Figure 2.2 summarises the Web usage mining architecture.

This dissertation examines some of the pattern discovery techniques in the Web usage mining stage. However, data preprocessing is beyond this dissertation scope and is implemented simply for the purpose of our experiments that rely on Web server log files.

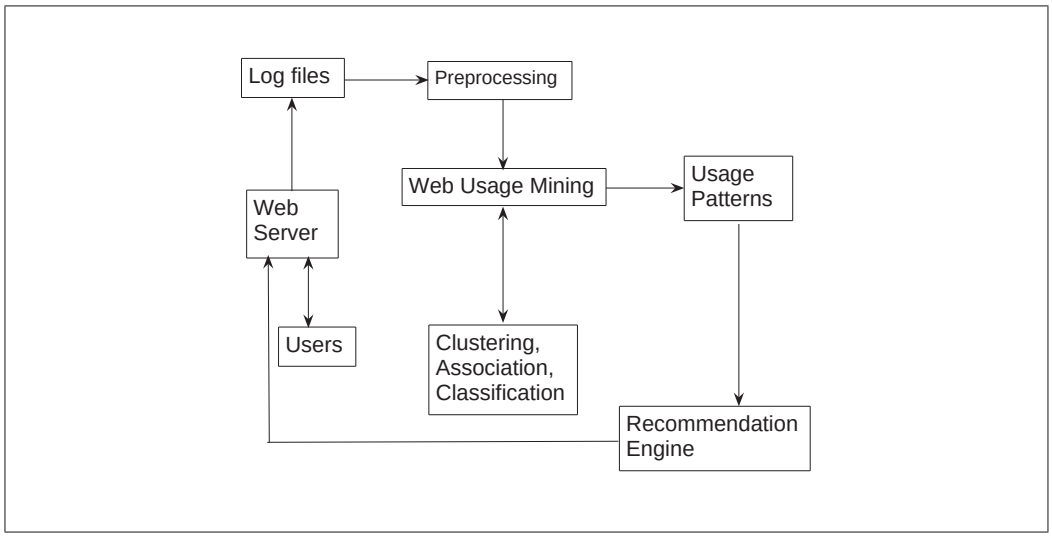

Figure 2.2: Web usage mining architecture.

2.3 Web Usage Mining Techniques and Web Page Prediction

Personalising the Web users' content and recommending appropriate Web pages imply that we are able to supply users with what they require based on their previous interactions within the same Web site. This task is viewed as a prediction task for we are trying to predict the users' level of interest in specific pages and rank these pages according to their predicted values. Different types of Web usage mining algorithms have been used to make predictions. This dissertation concentrates on model-based algorithmic approaches which are briefly discussed in chapter 3. The prediction process is achieved using the following algorithms: Markov model, association rules and clustering individually. The three models are compared according to their prediction accuracy which is calculated based on the ratio

of the specific Web page probability to the overall probability measure of the Web existence. The individual models prediction accuracy and efficiency is subject to some limitations. Therefore, combinations of these models are implemented. First, Markov model and association rules are integrated together to yield better accuracy results with fewer number of states and number of rules. Then, Markov model and clustering are integrated together resulting in improvement in prediction accuracy and prediction time complexity. Last, all three models are integrated together based on some constraints. This kind of integration results in improved efficiency with higher prediction accuracy and lower state space complexity.

Chapter 3

Prediction Techniques

3.1 Introduction

The prediction process forms part of the automatic personalisation process that consists of a data collection phase and a learning phase. Data collection phase can take many forms and it is beyond the scope of this dissertation. The learning phase can be classified into memory based learning or model based learning depending on whether the learning is performed online while prediction takes place or offline using training data. Standard user-based and content based personalisation systems rely on the memory based approach while item or page based personalisation systems rely on the model based approach. Memory based systems simply memorise all the data and generalise from it online real time. Their computational complexity is $O(MN)$ in the worst case where M refers to the number of

users and N refers to the number of items. Using memory based systems involves scanning all users to find similar users and then scanning all the items that the similar users have selected. This online computation complexity becomes a problem with typical electronic commerce Web sites. They are, however, extensively used in research and practice. Model based systems, on the other hand, can have heavier computation than user based systems, but their heavy computation is performed offline and their online computations are light. Model based techniques, including those used in the pattern discovery phase of Web usage mining, use a two stage process for prediction Suryavanshi et al. (2005). During the first phase, the data collected is mined offline and a model is generated. During the second phase, prediction takes place online as a new site visitor begins interacting with the Web site. The new visitor session is scored based on the model constructed in the first phase. Model based systems computational complexity could be $O(N^2M)$ in the worst case because they first scan the items, then for each item, they scan all the users, and finally, they find similar items by scanning the items again. Their online computational complexity is $O(N)$ in the worst case, but on average, the complexity is $O(constant)$ because the online computation depends on the number of items to look up, not on the total number of users or items. This reduction in computational complexity makes model based systems more suited for the online prediction stage than memory based systems. On the other hand, memory based systems are better at adopting to changes in the data sources. In the case of new data, model based systems have to be either incremented or rebuilt Mobasher et al.

(2000). More reasons why model based systems outperform user based systems for predicting Web pages is that model based are item based and computations are based on the items that are usually easily accessed from a Web server log file. Also, item data is more static than user data that changes with users' circumstances and environment. Since this dissertation focuses on Web personalisation where the data source is a repository of Web pages linked together according to some structure in a particular Web site, the prediction process is based on model based systems. In this chapter, we briefly describe a number of data mining algorithms used for offline model building techniques including Markov models, association rules and clustering.

3.2 Markov Model

Markov models (MMs) are commonly used in the identification of the next page to be accessed by the Web site user based on the sequence of previously accessed pages Bouras & Konidaris (2004), Chen et al. (2002), Deshpande & Karypis (2004), Eirinaki et al. (2005), Jespersen et al. (2003), Sarwar et al. (2001), Zhu et al. (2002*a*,*b*). They are the natural candidates for sequential pattern discovery for link prediction due to their suitability to modelling sequential processes. Markov models have been proposed as the underlying modelling techniques for Web prefetching applications Pons (2006), to minimize system latency or to improve Web server efficiency Mathur & Apte (2007). They can also be used to dis-

cover high-probability user navigational paths in a Web site Deshpande & Karypis (2004), Pitkow & Pirolli (1999), Sarukkai (2000), Srivastava et al. (2000).

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

$$\begin{aligned} P_{l+1} &= \text{argmax}_{p \in P}\{Prob(P_i = p|W)\} \\ &= \text{argmax}_{p \in P}\{Prob(P_i = p|p_l, p_{l-1}, \ldots, p_1)\} \end{aligned} \tag{3.1}$$

This probability, $prob(p_i|W)$, is estimated by using all W sequences of all users in history (or training data), denoted by W. Naturally, the longer l and the larger W, the more accurate $prob(p_i|W)$. However, it is infeasible to have very long l and large W and this leads to unnecessary complexity. Therefore, to overcome this problem, 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 k. In other words, the probability of visiting a page p_i does not depend on all the pages in the Web session, but only on a small set of k preceding pages, where $k << l$.

The equation becomes:

$$P_{l+1} = \text{argmax}_{p \in P}\{Porb(P_{l+1} = p|p_l, p_{l-1}, \ldots, p_{l-(k-1)})\} \tag{3.2}$$

where k denotes the number of the preceding pages and it identifies the order

of the Markov model. The resulting model of this equation is called the k^{th}-order Markov model. Of course, Markov model starts calculating the highest probability of the last page visited because during a Web session, the user can only link the page he/she is currently visiting to the next one.

Let S_j^k be a state with k as the number of preceding pages denoting the Markov model order and j as the number of unique pages in a Web site.

$S_j^k = \langle p_{l-(k-1)}, p_{l-(k-2)}, \ldots, p_l \rangle$. Using the maximum likelihood principle Duda et al. (2000), the conditional probability of $P\left(p_i|S_j^k\right)$ is estimated as follows from a history (training) data set.

$$P\left(p_i|S_j^k\right) = \frac{\text{frequency}\left(\left\langle S_j^k, p_i \right\rangle\right)}{\text{frequency}\left(S_j^k\right)} \quad . \tag{3.3}$$

This formula calculates the conditional probability as the ratio of the frequency of the sequence occurring in the training set to the frequency of the page occurring directly after the sequence.

The fundamental assumption of predictions based on Markov models is that the next state is dependent on the previous k states. The longer the k is, the more accurate the predictions are. However, longer k 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 & Karypis (2004):

1. Allk^{th} Markov model: This model is to tackle the problem of low coverage of a high order Markov model. For each test instance, the highest order Markov model that covers the instance is used to predict the instance. For example, if we build an all 4-Markov model including 1-, 2-, 3-, and 4-, for a test instance, we try to use 4-Markov model to make prediction. If the 4-Markov model does not contain the corresponding states, we then use the 3-Markov model, and so forth Pitkow & Pirolli (1999).

2. Frequency pruned Markov model: Though all-k^{th} order Markov models result in low coverage, they exacerbate the problem of complexity since the states of all Markov models are added up. Note that many states have low statistically predictive reliability since their occurrence frequencies are very low. The removal of these low frequency 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: Frequency pruned Markov model does not capture factors that affect the accuracy of states. A high frequent state may not present accurate prediction. When we use a means to estimate the predictive accuracy of states, 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 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. This proposed solution to the state space complexity of the all-k^{th} order model may not be feasible in some instances, especially when it comes to very large data sets. It requires a lot of time and effort to build the all-k^{th} order models and prune the pages according to the above three criteria. It also involves a great deal of calculations (different types of thresholds for different pruning methods).

3.3 Association Rules

Association rule mining is a major pattern discovery technique as proved by Mobasher et al. (2000). Association rule discovery on usage data results in finding groups of items or pages that are commonly accessed or purchased together. The original goal of association rule mining is to solve market basket problem. The applications of association rules are far beyond market basket applications and they have been used in various domains including Web mining. In Web mining context, association rules help optimize the organisation and structure of Web sites. Association rules are mainly defined by two metrics: support and confidence. Support is defined as the discovery of frequent itemsets (i.e. itemsets which satisfy a min-

imum support threshold) and confidence is defined as the discovery of association rules from these frequent itemsets Agrawal & Srikant (1994).

Let $P = \{p_1, p_2, , p_m\}$ be a set of pages in a Web site. 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:

$$\sigma = \text{sup}p(A) = \frac{\{W \in D : A \subseteq W\}}{|D|} \tag{3.4}$$

The confidence of the implication is:

$$\alpha = \text{conf}(A) = \frac{\text{sup}p(\langle A, P \rangle)}{\text{sup}p(A)} \tag{3.5}$$

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

The minimum support requirement dictates the efficiency of association rule mining. One major motivation for using the support factor comes from the fact that we are usually interested only in rules with certain popularity. Support corresponds to statistical significance while confidence is a measure of the rules strength. Confidence represents the conditional probability that item p_a occurs

in a transaction given that item p_b occurred in the same transaction. Support and confidence are the most commonly used metrics when using association based approaches to personalisation.

Since a full session in Web usage mining context includes many items (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 immense number of generated rules gives rise to the need of some predictive models that reduce the rule numbers and increase their quality by weeding out

the rules that were never applied. Yang et al. (2004), introduced the following predictive models:

1. Longest match: This method assumes that longer browsing paths produce higher quality information about the user access pattern. Therefore, in the case where we have more than one rule, all with support above a certain threshold and they match an observed sequence, the rule with the longest length will be chosen for predication purposes and the rest of the rules will be disregarded.

2. Most-confidence matching: This is a very common method where the rule with the highest confidence is chosen amongst the rest of all the applicable rules whose support values are above a certain threshold.

3. Least error matching: This is a method to combine support and confidence, based on the observed error rate and the support of each rule, to form a unified selection measure and to avoid the need to set a minimum support value artificially. The observed error rate is calculated by dividing the number of incorrect predictions by the number of training instances that support it. The rule with the least error rate is chosen amongst all the other applicable rules.

As a result of the experiments performed by the authors concerning the precision of association rule representations using different selection methods, the latest substring rules were proven to have the highest precision with fewer number of rules. However, the main problem with the latest substring rules is that they lead

to the same results regardless of the window size. Therefore, with a window of size one, they can be considered like first order Markov Model and they will be less accurate. Also, an increase in window size will lead to reduced efficiency and coverage.

Although not as widely used as clustering for Web personalisation Kim et al. (2004), Mobasher et al. (2001), Yong et al. (2005), the results of association rule mining on Web sessions can result in models that can be used for Web page prediction.

3.4 Clustering

The primary motivation behind the use of clustering as a model-based pattern discovery algorithm in Web usage mining stage of Web mining is to improve the efficiency and scalability of the real-time personalisation tasks Adami et al. (2003), Cadez et al. (2003), Papadakis & 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. Web page clustering is performed by grouping pages having similar content. Page clustering can be simple if the Web site is structured hierarchically. In this case, clustering is obtained by choosing a higher level of the tree structure of the Web site. 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). This process can get complicated due to the number of features that exist in each session. Wang et al. (2004) addressed this issue by using three different data features for grouping sessions. These features are service request, navigation pattern and resource usage. The authors proved using an E-rental application that all three criteria yield similar results and it is sufficient to group customers according to any one of these features. The authors suggest grouping customers by services requested because it yields better results and is simple to implement. Collaborative filtering is achieved when personalisation is performed based on both page and usage clustering Rigou et al. (2006).

Clustering can be model-based or distance-based. With model-based clustering Zhong & 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 Maximization (EM). Distance-based clustering involves determining a distance measure between pairs of data objects, and then grouping similar objects together into clusters. The most popular distance-based clustering techniques include partitional clustering and hierarchical clustering. A partitional method partitions the data objects 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 data objects and the clustering at the bot-

tom level containing one cluster for each data object. Model-based clustering have been shown to be effective for high dimensional text clustering Zhong & Ghosh (2003). Whereas, hierarchical distance-based clustering proved to be unsuitable for the vast amount of Web data. 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. The most commonly used distance measures are Euclidean distance and Manhattan or Cosine distance for data that can be represented in a vector space. Although distance-based clustering methods are computationally more complex than model-based clustering approaches, they have displayed their ability to produce more efficient Web documents clustering results Strehl et al. (2000), Gunduz & OZsu (2003).

Clustering can also be supervised Eick et al. (2004), Finley & 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 labeled. New patterns will be labeled and classified into existing labeled groups. Eick *et al.* Eick et al. (2004) examined supervised clustering and presented four representative-based supervised clustering algorithms: TDS, SCEC, SRIDHCR and SPAM. The authors show through experiments that supervised clustering improves the traditional clustering class purity. The greedy algorithms SPAM, SRIDHCR and TDS did not perform well for supervised clustering. Whereas, SCEC algorithm that centers on a more ran-

domized exploration provides better solutions. Finley *et al.* Finley & Joachims (2005) used an SVM algorithm Tsochantaridis et al. (2004) for supervised clustering in order to train a clustering algorithm by adapting the item-pair similarity measure. The algorithm optimizes the performance of correlation clustering on a variety of performance measures. The authors applied the algorithm to noun-phrase and news articles clustering. For the supervised clustering task, the users provided complete clusterings of a few of the documents to express their preferences. From these training examples, the authors learn to cluster future sets of items.

This kind of labeling is not possible with Web page classification because of the enormous volume of data. Therefore, a number of novel approaches have been proposed that involve a combination of labeled with unlabeled data. This is known as semi-supervised clustering. The problem associated with this co-training method is the type of Web data and the difficulty to draw generalizations of such algorithm. In Basu et al. (2004), the authors identify a semi-supervised clustering framework based on Hidden Markov Random Fields. The supervision is provided by the must-link and cannot-link constraints. The authors' proposed model involves a combination of constraints and Euclidean distance learning. The authors generalize on this combination through the handling of non-Euclidean measures by using Bregman divergence. The main problem associated with this model is the difficulty of applying such a supervision to large Web data sets. New approaches have been proposed to rectify the problem of large volume of Web

data. For instance, The work presented by Rigou *et al.* Rigou et al. (2006) focuses on providing an effective personalised clustering state for Web documents. The authors proposed an algorithm, based on a range tree structure, that reduces the number of Web documents the users receive in result to their queries. They focused on improving the online retrieval of Web documents and therefore, the clustering algorithm was used as a final stage after filtering the documents according to users' preferences. Relying on the document clustering algorithm, the authors helped solve the *k*-means algorithm time complexity problem by using *k*-windows algorithm and relying on the pre-existing range tree structure. The main limitation associated with this type of clustering however, is the changing nature of Web documents. It becomes very difficult to implement such algorithm on Web documents that require frequent updates.

Unsupervised clustering is where no labeling of either of the data sets is available. Unsupervised clustering of Web documents was presented by Albanese *et al.* Albanese et al. (2004). The authors addressed the issue of Web page personalization based on short user navigation history. They performed a two phase clustering technique. In the first phase, they used unsupervised clustering algorithm for pattern analysis and classification using the static user registration information. This static information helped determine the number of classes the users can belong to. For this purpose, the authors used two clustering techniques: AutoClass C, a fuzzy clustering algorithm based on the Bayesian theory, and the Rival Penalized Competitive Learning (RPCL) algorithm, that is used for training a compet-

itive neural network. In the second phase, they iteratively used re-classification to overcome the inaccuracy of the registration information based on users' navigational behavior. Re-classification was repeated until a suitable convergence in attributing each user to a class was achieved. It was accomplished by content management and log analysis based on the dynamic user navigational behavior. Content management involved identifying significant keywords by a domain expert and associating the keywords with content categories. Whereas log analysis was used to provide users' activities in order to attribute each content category to a specific user class.

Unsupervised clustering can be classified as hierarchical or non-hierarchical Jain et al. (1999). Using hierarchical clustering, data is clustered in the form of a tree where each datum in the lowest level is defined as a cluster. Larger clusters will be formed moving up to higher levels in the tree. The problem related to such technique is that it can become computationally complex with large data sets and can be difficult to analyze 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. The non-hierarchical clustering main concern is the partitioning of data into a specified number of clusters. Not all data types are suitable for such partitioning. Numerous recent papers addressed the partitional non-hierarchical clustering algorithm, *k*-

means, and attempted at improving the algorithm. Xiong *et al.* 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 favorite on *k*-means and can be an unsuitable *k*-means clustering validation measure.

Other work that attempts at improving the traditional *k*-means algorithm is presented by Geraci et al. (2006) where the authors propose a clustering method that is more accurate and faster than classical *k*-means algorithm when it comes to clustering large amount of data with real-time nature like Web snippets. Web snippets are the search engine results that are composed of the name of the document, the URL address and few statements describing the content of the document. The authors presented the furthest-point-first algorithm for *k*-center clustering in metric spaces coupled with a filtering scheme based on the triangular inequality. In another work Meneses & Rodriguez-Rojas (2006), Meneses *et al.* extended the traditional *k*-means algorithm that uses the vector model for representing documents. They used symbolic objects that perform better at representing concepts

than individuals. Symbolic objects are vectors that could be of any type. They add semantic power to Web clustering by providing the user with more information about the content of the objects. The authors proved through experiments that symbolic models can be more accurate and efficient than vectorial representations.

Hierarchical clustering algorithms can be obtained using either agglomerative or partitional algorithms. With agglomerative algorithm, each object is assigned to its own cluster and then pairs of clusters are merged to form a tree. With partitional algorithms, on the other hand, the tree is formed by a series of repeated bisections. Partitional algorithms are better suited for clustering large data sets because of their low computational requirements but they are less effective than agglomerative algorithms. The work presented by Zhao & Karypis (2002) compares both algorithms using six partitional methods and nine agglomerative methods. The authors introduced constrained agglomerative algorithms that generate hierarchical trees using both partitional and agglomerative methods. Using agglomerative algorithms, they build a hierarchical subtree for each of the intermediate partitional clusters. Then, an upper tree is built using the subtrees as leaves. The experiments performed using twelve data sets from various sources showed that partitional algorithms produced better hierarchical solutions than agglomerative methods, and that the constrained agglomerative methods improved the clustering solutions obtained by either agglomerative or partitional algorithms. These results suggest that the poor performance of agglomerative algorithms is associated with the merging errors that occur during early stages. Another research that deals

with both agglomerative and partitional clustering was presented by Cheng et al. (2005). It introduced a divide-and-merge process for clustering that combines a top-down phase with a bottom-up phase.

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. Therefore, it is essential to study most clustering algorithms well before deciding on the one algorithm that applies most to the data at hand. Adami et al. (2003) introduced a special kind of hierarchical supervised clustering in order to solve the problem of high amount of labels associated with bootstrapping or page labeling. As a solution, the authors used a baseline approach where documents are classified according to their class labels and a constraint *k*-means clustering approach at topological and terminological levels. Most recent research that addressed supervised hierarchical clustering did not take into consideration the complexity involved with bootstrapping, or page labeling, because with supervised clustering the target classes are known in advance. They are usually conducted by a human expert. Bootstrapping can get very complex due to the large number of labels with the increase of the number of categories or classes. The authors address the issue of bootstrapping and provide a preliminary categorisation hypothesis on the classification of the documents resulting in reduced human effort where a human expert is only needed to weed out wrong classifications. They introduced the TaxSOM and they proved the model to be more effi-

cient in cases where the documents descriptions are made of labels of wrong class together with labels of the correct class. The main limitation associated with TaxSOM is its undirected graph topology. Poor results were obtained when taking into consideration the parent-child relationships. Restrictive clustering methods were introduced by Siersdorfer & Sizov (2004). The authors presented an approach for automatically structuring heterogenous document collections by using restrictive clustering methods. Usual clustering methods may result in cluster impurity because the entire data set is partitioned into clusters. The paper solves this problem by clustering only a subset of the data leaving out data not assigned to any clusters. The authors refer to this process as restrictive clustering and they show through experiments that this technique results in higher cluster purity and better accuracy. They introduced 3 meta mapping algorithms (correlation-based mapping, purity-based mapping and mapping using association rules) for restrictive clustering and they used *k*-means approach as a partitioning method. Chakrabarti et al. (2006) examined an evolutionary clustering algorithm where a different sequence of clusterings is produced for each timestamp. This creates a problem if the clustering changes from one timestamp to the next. The authors solve this problem using evolutionary versions of *k*-means and agglomerative hierarchical clustering. The experiments using the collection of timestamped photo-tag pairs from flickr.com show that both *k*-means and agglomerative clustering can achieve high accuracy and high reliability in reflecting clustering history.

Various attempts were made at clustering information extracted from Web

search engines Sun et al. (2006), Ferragina & Gulli (2005), Papadakis & Skoutas (2005). Traditional search engines perform very straight forward search by responding to customers queries. Sun et al. (2006) introduced a Comparative Web System (CWS) that enhances the users' search results. It allows the user to submit more than one query. CWS automatically retrieves and ranks the information, compares the different queries results, clusters the results into different themes and extracts representative key phrases. The user is provided with two types of view modes: a pair view that displays the result and a cluster view that displays the comparative pages results with their key phrases. CWS functions are still preliminary and very basic. Also, Ferragina & Gulli (2005) introduced a hierarchical Web-snippet clustering system, SNAKET, that produces a hierarchy of labeled folders from search results. SNAKET is an open-source system that is efficient in achieving personalisation, adaptive to user needs, privacy preserving and scalable to the number of users. STAVIES is another system that improves information extraction from Web search results Papadakis & Skoutas (2005). STAVIES is an automated wrapper that identifies the pieces of the Web pages that contain the information and extracts the information using hierarchical clustering techniques.

3.5 Conclusion

Several attempts were made at using Markov models, association rule and clustering frameworks to help predict the next Web page to be accessed by the user.

Markov models are the most commonly used techniques for such a purpose but they suffer from the limitation of high state space complexity. Therefor, different modified Markkov models existed to provide for better Web page access prediction. Association rules are simple to implement but the rules they generate can get too complex to provide useful patterns. For this reason, different methods were used in order to weed out unnecessary rules. Clustering tools are used to improve the personalisation task but they are not appropriate to be used as predictive models on their own and they suffer from the wide diversity of existing clustering algorithms.

Chapter 4

Integrating Markov Model with Association Rules

4.1 Introduction

Predicting the next page to be accessed by a Web user is achieved using various pattern discovery techniques. Two of the most common approaches are Markov models and association rules. Each of the approaches used for this purpose has its own weaknesses when it comes to accuracy, coverage and performance. Lower order Markov models lack accuracy because of the limitation in covering enough browsing history; whereas higher order Markov models usually result in higher state space complexity. On the other hand, association rules have the problem of identifying the one correct prediction out of the many rules that lead to a large

number of predictions Mobasher et al. (2001), Yang et al. (2004).

This chapter introduces an improved approach, based on a combination of Markov models and association rules that results in better prediction accuracy accompanied by lower state space complexity. The approach uses lower order Markov model that is accompanied by lower state space complexity and reduced prediction accuracy. Association rules are used to provide better prediction accuracy while keeping the number of generated rules to a minimum. This is due to the fact that association rules are only used where Markov model prediction is ambiguous. Section 2 of the chapter introduces Markov model and the problems associated with it. Section 3 introduces association rules and their limitations. Section 4 examines the integration process of the new model and explains the integration algorithm. Section 5 provides proficient concept experiments. Finally, Section 6 concludes this chapter.

4.2 Markov Model

Markov models have been introduced in Chapter 3, Section 3.2. In this section we identify some main limitations of Markov models.

4.2.1 Limitations of Markov Models

One limitation of applying Markov model techniques to the Web personalisation and prediction process is the difficulty of data interpretation and visualisation. However, Cadez et al. (2000) propose a method for the visualisation of the models that provides an insight about the usage of the system.

Another main obstacle that faces Markov model users is the identification of an optimal number of Markov model orders. Until now, the optimal number does not exist and each Markov model work has its own insight about choosing the best order that fits the data on hand. The number of Markov model orders affects the system accuracy, coverage and performance. Numerous researches dealt with the topic of Markov Model as a method to solve this prediction problem keeping in mind higher coverage, better accuracy and performance. For instance, lower Markov model orders lead to reduced coverage and, therefore, accuracy due to the lack of data in previous history. Deshpande & Karypis (2004) addressed the reduced accuracy problem of the low-order Markov Models. They proposed an all-k^{th} order model instead. Although the all-k^{th} order models solve the reduced accuracy problem, they give rise to another major problem, the state space complexity. They proposed solving the problem of the all-k^{th} order model by pruning some of the states according to frequency, confidence and error representations. This proposed solution to the state space complexity of the all-k^{th} order model may not be feasible in some instances, especially when it comes to very large and

high dimensional data sets. It requires a lot of time and effort to build the all-k^{th} order models and prune the pages according to the three criteria. It can also get very difficult to set proper parameters for various pruning models. Although the authors proved to increase the coverage and reduce state space complexity, the increased accuracy problem remains unsolved.

Dongshan & Junyi (2002) proposed the use of a hybrid-order tree like Markov Model (HTMM) in order to solve the problems associated with traditional Markov Models especially the state space complexity and low coverage. They identified the suitability of HTMM with predicting the next pages to be accessed by the user and caching such pages in order to improve Web pre-fetching. HTMM combines two methods: a tree-like Markov model method and a hybrid order method. The k-order Tree-like Markov model is a tree constructed using a sequence of visited Web pages accessed by the user. Each node of the tree conforms to a visited page URL and a count that records the number of times the page was visited. The height of the tree is $k+2$ where k is the order of the Markov model and the width of the tree is no more than the number of sequences of the visited pages. The tree-like Markov model results in low coverage that results in low accuracy. As a solution, the authors proposed training varying order Markov models and combining those models together for prediction. They used two methods for combining the models: accuracy voting and blending. To evaluate the results of these methods, the authors used Web server log files of an educational site and after cleaning and pre-processing the log data, they came up with the following results: When

it comes to precision and accuracy, both HTMM methods showed better results than traditional Markov models. Also, when it comes to time associated with building the models and giving prediction, the HTMM methods showed better results than traditional Markov models. However, with prediction time, HTMM methods and traditional methods showed similar results. These results are apparent with HTMM in general. However, when it comes to building the tree, it is based on all-k^{th} order model and it has the same complexity as the all-k^{th} order model. This places a great limitation on the approach as a whole.

4.2.2 Markov Model State Space Complexity

Analysing the state space complexity of Markov model is a major issue that needs attention either when implementing Markov model alone or when combining Markov model with other models. The state space complexity increases with the increase of Markov model order. Higher orders lead to more states but they usually result in better prediction accuracy since they look at previous browsing history.

Considering the Markov model states S_j^k introduced in Chapter 3, Section 3.2, 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.

In order to prepare the data for Markov model mining, duplicate pages visited in sequence are eliminated during the preprocessing stage. A page visited more than once could be due to the fact that the user is refreshing the page or double clicking at a link. This reduces the number of states by the number of unique pages. For instance, The number of states for 2^{nd} order Markov model is 12 for 4 unique pages, and is 20 for 5 unique pages. Also, all states that lead to a prediction value of zero are disregarded. This reduces the number of states to a certain extent but the fact remains that longer transactions and higher Markov model orders result in a very large number of states. The number of states can become enormous leading to difficult, and sometimes impossible, frequent states and conditional probability computations.

Using the data sets explained in Table 4.10 in Section 4.5 of this chapter, Table 4.1 and Table 4.2 demonstrate the increase of the state space complexity as the order of all-k^{th} Markov model increases.

Table 4.1: 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 4.2: 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

4.2.3 Using Markov Model Order for Prediction

The main 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. When choosing the Markov model order, our aim is to determine a Markov model order that leads to high accuracy with low state space complexity. Although using higher order Markov models increase coverage and accuracy, using lower order Markov model, the new user sessions can be easily fit into the model and dynamic predictions can be generated based on the probability of the occurrence of the new item in the existing model.

Figure 4.1 below reveals the increase of precision as the all-k^{th} order Markov model increases. Based on the accuracy increase represented in Figure 4.1, and based on the increase in the number of states represented in Table 4.1 and Table 4.2, we use the all-2^{nd} order Markov model because it has better accuracy

than that of the all-1^{st} order Markov model without the drawback of the state space complexity of the all-$3^{r}d$ and all-$4^{t}h$ order Markov model.

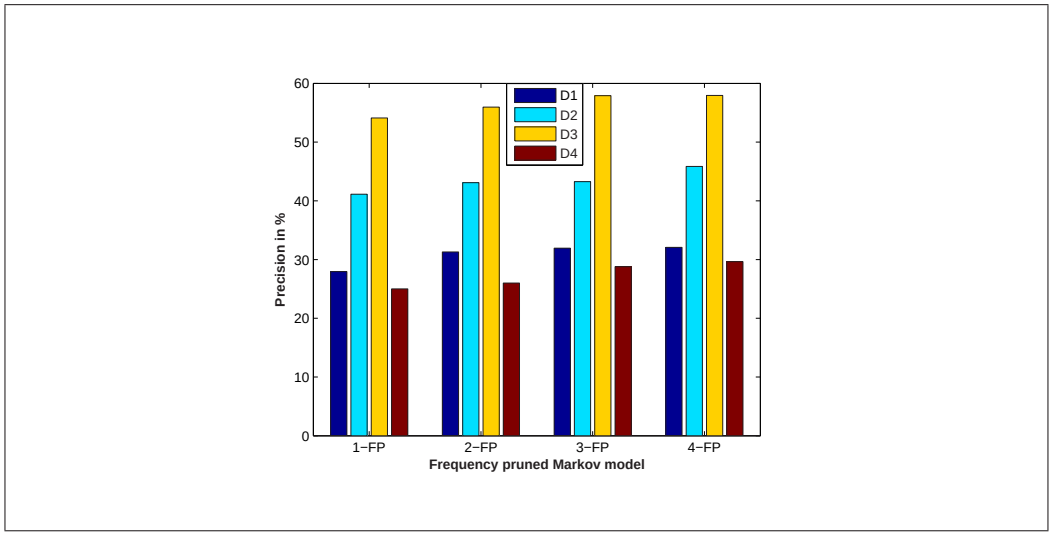

Figure 4.1: Accuracy of all 1-, 2-, 3- and 4- frequency pruned Markov model orders.

For the purpose of this dissertation, we employ the frequency pruned Markov model as explained in Section 2 of Chapter 3.

4.3 Association Rules

The application of association rule mining methods to Web usage mining is limited, focusing primarily on the prediction of the most interesting next Web page for the user. Association rules were discussed in Chapter 3, Section 3.3. In this section, we introduce the main problems related to association rule mining used

for Web page access prediction purposes.

4.3.1 Limitations of Association Rules

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. This is often the case of Web usage mining applications. For instance, consider the four transactions shown in Table 4.3:

Table 4.3: Example: Four Web transactions

T1	A C D
T2	C E
T3	B C E
T4	A C D E

According to Apriori algorithm, in each iteration, the items are scanned and candidate itemsets are identified. Then, large itemsets are determined based on the predetermined minimum support factor Park et al. (1997). The above transactions lead to the following set of candidate 1-itemsets:

$$C_1 = \{\{A\},\{B\},\{C\},\{D\},\{E\}\}$$

Taking a minimum support factor of 40% into consideration where the minimum transaction support is 2, the set of large 1-itemsets is as follwos:

$$L_1 = \{\{A\},\{C\},\{D\},\{E\}\}$$

In the next iteration, Apriori algorithm identifies the candidate 2-itemsets by concatenating two L_1 together:

$$C_2 = \{\{AC\},\{AD\},\{AE\},\{CD\},\{CE\},\{DE\}\}$$

Again, applying the minimum support factor leads to the following set of large 2-itemsets:

$$L_2 = \{\{AC\},\{AD\},\{CD\},\{CE\}\}$$

The next iteration results in the following candidate 3-itemsets:

$$C_3 = \{\{ACD\},\{CDE\}\}$$

The large 3-itemsets becomes as follows:

$$L_3 = \{\{ACD\}\}$$

This example of only four transactions with five unique pages generates a large number of rules highlighting frequent items. A Web data set with a large number of transactions would lead to redundant and complex rules.

In order to alleviate the problem of large number of rules, Mobasher et al. (2001) recommended an approach that uses association rule techniques that are based on storing the most frequent items used in a data structure and using an

algorithm to identify the most suitable items to be used with online recommendations. In order to decrease the large number of itemsets associated with association rules, the authors proposed a method that involved increasing the window size. However, their method caused scalability problems as well as lower coverage. On the other hand, using multiple support thresholds resulted in better coverage but it did not improve on accuracy. Also, faced with the same association rule problem, Agrawal & Srikant (1994)advocated variations of the original Apriori algorithm presented by Agrawal et al. (1993).

The work presented by Park et al. (1997) proposed another method to alleviate the problem of association rules large itemsets by introducing two methods. The first method is based on direct sampling, whereas, the second method is based on sampling with effective hash construction. A technique of relaxing the support factor based on the sampling size is devised to achieve the desired level of prediction accuracy.

A different approach to overcome the problems associated with association rule mining for Web personalisation is proposed by Schwarzkopf (2001) that employ Bayesian networks for defining taxonomic relations between topics covered by a particular Web site. The nodes in the network correspond to a stochastic variable associated with a certain topic. The association networks provide a graphical representation of the users' topics of interest. This approach also leads to a scalability problem due to the initial construction of the networks that is performed

manually.

Another significant issue associated with association rules in large data item sets is the uncovering of large number of rules that apply to one instance and the difficulty in identifying one rule that leads to the correct prediction for that instance. Moreover, with Web page prediction, it helps if the user can specify the number of rules that satisfy a given level of support and confidence because, typically, the user is interested in only a small number of rules. This aggravates the problem because the user may need to run the query multiple times in order to find the appropriate levels of minimum support and minimum confidence needed to mine the rules. Another issue is using a global minimum support threshold for the whole data set as it is highlighted in Liu et al. (1999). The authors argued that using one minimum support implies that all items in the data set have the same frequencies and/or are of similar nature. Unfortunately, this is untrue in real life data and some candidate, or rare, items with less frequencies may be excluded from the generated rules if the minimum support threshold is too high. Setting a low minimum support will generate a huge number of meaningless rules. For this reason, a model that allows the user to specify multiple minimum support threshold is proposed. Although this model is proven to be effective, it suffers from implementation complexity.

4.3.2 Using Association Rules for Prediction

The ultimate objective of prediction is to use itemsets to dynamically recommend Web pages to the users. In the context of Web usage mining, the discovery of association rules usually aims at the discovery of associations between Web pages based on their co-occurrence in user sessions Mobasher et al. (1999). For the purpose of this dissertation, sequential association rule mining is used on user transaction data to discover Web page usage patterns.

Assumption 4.1: *sequential Web pages means that the predecessor Web page should be browsed or accessed before the successor Web page.*

This is very important due to the fact that Web pages included in a Web session are sequential in nature and the order of the accessed Web pages is crucial in the prediction process. It has been shown that contiguous sequential association rules are restrictive and hence are more valuable in page prefetching applications where the intent is to predict the next page to be accessed by the user rather than in the more general context of recommendation generation Mobasher et al. (2002), Yong et al. (2005). In this dissertation, simple association rule mining is followed and constructing the association rules is performed based on Arpiori algorithm. Association rules are generated according to a predefined support and confidence thresholds. Prediction of the next page to be accessed by the user is performed by matching the discovered patterns against the user sessions. This is usually done online.

4.3.3 Error Estimation of Association Rules Based Prediction

The two notions for establishing the strength of a rule are the minimum support and minimum confidence introduced in Section 3.3 of Chapter 3 where W is a user session and A is a subsequence of W. Association rule prediction precision is defined as the number of correct predictions C divided by the number of test cases N.

$$Precision = \frac{C}{N} \tag{4.1}$$

The observed error rate of association rule based prediction, o, is the ratio of the number of incorrect predictions I to the number of occurrences of A or $supp(A)$ denoted by M Berti (2007).

$$o = \frac{I}{M} \tag{4.2}$$

Based on equation 4.2, a rule with a small support measure will have a higher observed error rate with the same number of wrong predictions. However, a higher support measure will face the complications of missing some useful rules.

To find the true error rate e, consider a random variable X with a mean that lies within a range of $2z$, where z is a variable with an unknown value, and with the confidence of $Cf = Prob\langle -z \leq X \leq z \rangle$. Hence, the value of z can be obtained from any value of Cf using normal distribution. From the above notation, where o is the observed error rate and e is the mean, we can set the value of the random

variable X to become:

$$X = \frac{o - e}{\sqrt{e(1-e)/M}} \tag{4.3}$$

Based on the above equation, the range of the true error rate e can be obtained knowing the observed error rate o and the number of supporting instances M.

4.4 Integration Process

Our integration model involves using low order Markov models to predict the next page to be visited by a user and then applying association rule techniques to predict the next page to be accessed by the user based on long history data.

4.4.1 Motivation for Integration

In this chapter, we integrate association rule mining with Markov models in order to improve prediction accuracy. Both association rules and Markov models have been used individually for prediction purposes, but each of them has its own limitations when it comes to Web page prediction accuracy and state space complexity. The main advantage of Markov models is that they can generate navigation paths that could be used automatically for prediction, without any extra processing and thus they are very useful for Web personalisation. In addition, they are supported by a good mathematical background. However, prediction based on Markov models is not free from disadvantages. Although higher order Markov

models provide better accuracy than lower order Markov models, they suffer from state space complexity. On the other hand, lower order Markov models provide less complex states but lower prediction accuracy. The all-k^{th} order Markov models provide better accuracy but are subject to a more complex state space. The all-1^{st} order Markov model predicts a user's next request based only on the page that was requested last. The all-2^{nd} order Markov model makes prediction based on the last two requested pages. If prediction cannot be made (i.e. the predicted page does not exist in the training data set), the all-1^{st} order Markov model is used for prediction. The more pages are examined in history, the more states are encountered and the more complex prediction will get.

Association rule mining is a major pattern discovery technique Mobasher et al. (2001). The original goal of association rule mining is to solve market basket problem but the applications of association rules are far beyond that. Association rules are also used for predicting the next page to be accessed by the Web user. They make prediction based on the users' browsing history. The more frequently the pages are accessed, the higher the probability of the user accessing the next page. Using association rules for Web page access prediction involves dealing with too many rules and it is not easy to find a suitable subset of rules to make accurate and reliable predictions Kim et al. (2004), Mobasher et al. (2001), Yong et al. (2005). Similar to Markov models, association rules endure the problem of the large number of rules generated. The number of the generated rules can get very large in large data sets that it can become impossible to make prediction.

There is apparent a direct relationship between Markov models and association rule techniques. According to the Markov model pruning methods presented by Deshpande & Karypis (2004) and association rules selection methods presented by Yang et al. (2004), there exists a great resemblance between the two. The substring association rules with most confidence prediction model form a frequency pruned all-k^{th} order Markov model, where k is the number of maximum items in the association rules. They also share similar problems. For instance, the number of states (rules) becomes unmanageable when k is large. In contrast, short history is not enough for making accurate predictions.

Keeping the disadvantages of both Markov models and association rules to a minimum, our main goal is to provide a new model that increases the prediction accuracy of both models combined with fewer number of rules. We implement a low order all-k^{th} Markov model keeping state complexity to a minimum. Referring to Chapter 3, Section 3.2, using Markov model prediction, the probability of visiting a page p_i depends on a small set of k preceding pages as follows:

$$P_{l+1} = \text{argmax}_{p \in \mathbb{P}}\{Porb(P_{l+1} = p | p_l, p_{l-1}, \ldots, p_{l-(k-1)})\} \tag{4.4}$$

where k denotes the number of the preceding pages and it identifies the order of the Markov model. In our integration model, we applied equation 4.4 for the all-2^{nd} order Markov model where $k = 2$.

Using the all-2^{nd} order Markov model, prediction is made using 2^{nd} order Markov model where $k = 2$. If prediction results in no states, prediction is made

using 1^{st} order Markov model where $k = 1$. The resulting accuracy of such low order Markov model is normally not satisfactory. Therefore, for those Markov states that provide ambiguous predictions, we make use of association rules to sample long history.

The main purpose of using association rules is to provide more accurate predictions. Association rules are complicated as well, but we only use rules to complement Markov states that provide ambiguous predictions so that we do not add too much complexity to the system. In this integration model, we consider the binary vector representation only. A 0 or 1 is used to indicate whether the page was visited or not. Let $I = i_1, i_2, ..., i_m$ be a set of items. The data set D consists of a set of transactions T. Each transaction $T_i \in T$ is a set of items, such that $T \subseteq I$. A transaction T_i is said to contain the set of items X if and only if $X \subseteq T_i$. An association rule is a condition of the form $X \Rightarrow Y$ where $X \subseteq I$ and $Y \subseteq I$ are two sets of attributes. The intuitive implication of the association rule is that a presence of the set of items X in a transaction set also indicates a possibility of the presence of the itemset Y. The larger the set of items X, the more rules are generated. For Web data sets with long Web sessions and large number of sessions, the generated rules could become very complex and sometimes misleading. In this dissertation, we only rely on association rule mining in special cases in order to limit the complexity and time needed for both processing and prediction. We generate association rules only if the Markov model prediction results in ambiguity. During processing or training, the number of rules generated is small and rules

complexity is reduced. During prediction, the rules examined are simple and they are referred to only in the case of ambiguity.

The architecture of the Integrated Markov and Association Model (IMAM) is depicted in Figure 4.2 below.

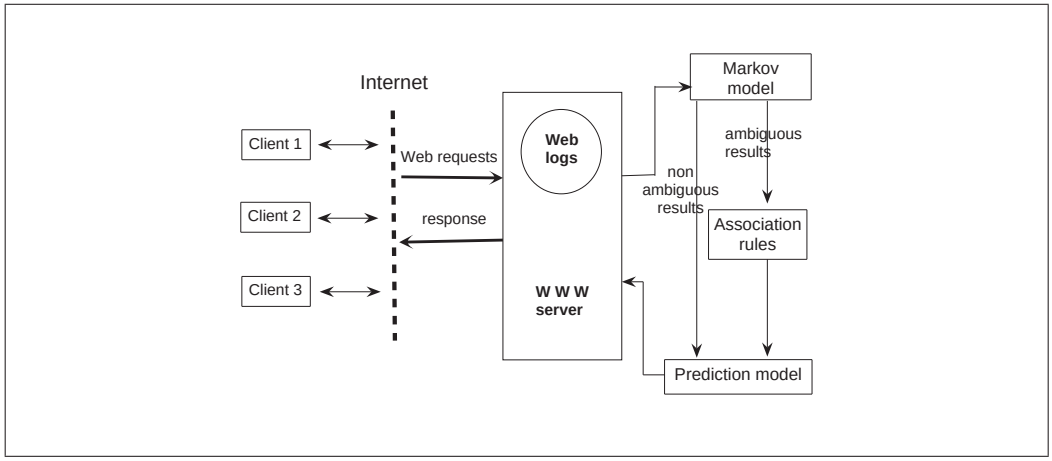

Figure 4.2: The Integrated Markov and Association Model (IMAM) architecture.

This integration model, benefits from both Markov model and association rules while keeping the models disadvantages to a minimum. The integration model profits from the decreased state space complexity of the lower order Markov model and it compensates for the decreased accuracy of the lower order Markov model by using association rule mining in case of ambiguity. The integration model also avoids the complexity of the association rules since the rules are generated only in special cases. In brief, the new integration model results in an increase in prediction accuracy and a decrease in state and rule complexity.

4.4.2 Integration Algorithm

In this chapter, we introduce the Integrated Markov and Association Model (IMAM) that inputs a database (D) and a session (W) and outputs the next page(p_n) that will be accessed by the user with high prediction.

The IMAM algorithm is summarised as follows:

Training:

```
(1)Build a low order Markov model
(2)FOR each state of the Markov model
(3)  IF the prediction is ambiguous
(4)    THEN
(5)    Collect all sessions satisfying the state
(6)    Construct association rules to resolve ambiguity
(7)    Store the association rules with the state
(8)  ENDIF
(9)ENDFOR
```

Test:

```
(1)Find a matching state of the Markov model
   for a test session
(2)IF the matching state provides an non-ambiguous prediction
(3)  THEN
(4)  Prediction is made by the state
```

```
(5)ELSE
(6)  Use its corresponding association rules to make prediction
(7)ENDIF
```

An ambiguous prediction is defined as two or more predictive pages that have the same conditional probability by a Markov model.

4.4.2.1 Markov Model Implementation

The next page prediction is computed using the Markov model probabilistic framework as follows. We define W as the user's Web session containing l pages. The probability that the user visits page p_n next is estimated, using conditional probability, by the number of times page p_n was visited immediately after the previous page p_c to the number of times page p_c was visited. This probability can be used to create a 1^{st} order Markov model. In our case, we want to create a 2^{nd} order Markov model. Therefore, we compute the next page prediction by dividing the number of times page p_n occurs immediately after the sequence of two pages $\langle p_c \rightarrow p_p \rangle$ to the number of times the sequence of the two pages $\langle p_c \rightarrow p_p \rangle$ was visited. If the probability value is zero or if two or more pages have the same highest probability value resulting in a tie, association rules are being examined.

For instance, for every current page the user clicks at, p_c, the prediction model will estimate, using conditional probability, the probability of accessing the next page p_n by examining previously accessed pages. During the training or model

building phase, the probability of accessing the next page p_n is first calculated using 2^{nd} order Markov model based on Equation (4.4).

Definition 4.1: *A non-ambiguous prediction is when there is one and only one p_n with the largest probability.*

Definition 4.2: *An ambiguous prediction is when there are two or more pages with the same probability that is equivalent to the largest one.*

Constructing the 2^{nd} order Markov model, results in one of two cases:

$$Probability\,for\,p_n = \begin{cases} 0 & if\,P(p_n|p_c,p_p) = 0 \ \ and \\ & P(p_n|p_c) = 0 \\ \neq 0 & if\,P(p_n|p_c,p_p) \neq 0 \ \ or \\ & if\,P(p_n|p_c,p_p) = 0 \ \ and \\ & P(p_n|p_c) \neq 0 \end{cases}$$

where p_p is the page accessed immediately before p_c by the same user in the same Web session W. Let Probability for $p_n = A(x)$ where $A(x) = \{P(p_1), P(p_2), ..., P(p_x)\}$. The items of $A(x)$ satisfy the following conditions:

$\sum\{(P(p_1), P(p_2), ..., P(p_x))\} = 100\%$ having the sum of all probabilities equal to 100 and

$(P(p_1) \geq P(p_2) \geq P(p_3)...P(p_x))$ satisfying the condition that the probabilities are in descending order and, $(P(p_1), P(p_2), P(p_3), ..., P(p_x))are \neq 0$ where non of the probabilities can have a zero value.

Let $F(i) \subseteq A(x)$. $F(i) = \{P(p_1), P(p_2), ..., P(p_i)\}$ where $i \leq x$ and $P(p_1) =$

$P(p_i)$ to fulfil the tie condition.

A non-ambiguous prediction takes place where:

$$\begin{cases} P(p_n) \neq 0 \quad and \\ F(i) = 0 \end{cases}$$

On the other hand, an ambiguous prediction takes place where:

$$\begin{cases} P(p_n) = 0 \quad or \\ F(i) \neq 0 \end{cases}$$

For example, if $P(p_n) = \{0.8, 0.2\}$ and $F(i) = 0$, it means that all probability values are different with no two or more probabilities having the same values. In this case, the probability of accessing page p_n would be 0.8. On the other hand, if $P(p_n) = \{0.4, 0.4, 0.2\}$, $F(i) = \{0.4, 0.4\}$. This means that we have an ambiguous prediction and the probability of accessing page p_n will not be determined based on Markov model analysis. Association rules for this state will be examined instead.

4.4.2.2 Implementation of Association Rule Mining

The use of association rule mining for Web page prediction is restricted due to the fact that most approaches used for this purpose are variants of the same algorithm, Apriori. This results in the limited scope for comparative evaluation of different methods.

The original Apriori algorithm was described by Agrawal & Srikant (1994) as

follows: (1)L_1 =Large 1-itemsets

(2)FOR ($k = 2, L_{k-1} \neq 0, k^{++}$) DO BEGIN

(3) C_K = apriori-gen(L_{k-1})

(4) FOR all transactions $T \in D$ DO BEGIN

(5) $C_t = subset(C_k, t)$

(6) FOR all candidates $c \in C_t$ do

(7) $c.count++$;

(8) END For

Please refer to Agrawal & Srikant (1994) for the apriori-gen function that performs a restricted join of L_{k-1} with L_{k-1} and generates all 1-extensions of L_{k-1} which potentially can be large itemsets.

In this dissertation, a variant to the Apriori algorithm (AprioriAll) Agrawal & 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 AprioriAll algorithm is as follows Agrawal & Srikant (1996): (1)L_1 =Large l-sequences; //litemsets

(2)FOR ($k = 2, L_{k-1} \neq 0, k^{++}$) DO BEGIN

(3) C_K = New candidates generated from L_{k-1}

(4) FOR each transaction $T \in D$ DO BEGIN

(5) Increment the count of all candidates in C_k that are contained in c.

(6) L_k = Candidates in C_k with minimum support.

(8) END For

The probability of accessing page p_n is now calculated using association rules according to AprioriAll algorithm with a predetermined window size and minimum confidence and support factors as explained later in this chapter.

Being the most common association mining algorithm Agrawal & Srikant (1994), Apriori algorithm and its variant AprioriAll algorithm have a main problem that is composed of two steps:

1. Discovery of large itemsets or litemsets in the case of AprioriAll algorighm.

2. Using the large itemsets to generate the association rules.

The second step is simple and the overall performance of mining association rules is determined by the first step. Apriori algorithm Agrawal & Srikant (1994) addresses the issue of discovering large itemsets. In each iteration, Apriori constructs a candidate set of large itemsets, counts the number of occurrences of each candidate and determines the large itemsets based on a predetermined mini-

mum support and confidence thresholds. In the first iteration, Apriori scans all the transactions to count the number of occurrences for each item and based on the minimum support threshold (σ), the first large itemset is determined. Therefore, the cost of the first iteration is $O(|T|)$, where T denotes the number of transactions in the dataset D and $|T|$ denotes the size of T. Next, the second large itemset is determined by concatenating items in the first large itemset and applying the minimum support test to the results. More iterations will take place until there are no more candidate itemsets. In simple terms, the cost of the algorithm is $O(I * D)$ where I denotes the number of iterations used Han & Plank (1996), Zhao et al. (2007). Larger transactions means larger itemsets and consequently larger I and more complex running time. Association rules are generated based on all large itemsets extracted using all Apriori algorithm iterations. The generated rules are so large and complex that they can lead to conflicting results. The use of association rules in this dissertation is restricted to special cases of Markov model prediction leading to ambiguity. This limited use of association rules cuts down on the number of rules and complexity of rules generated.

During the test or prediction phase, for every new page p_c, the probability of the user accessing the next page p_n is calculated using all-2^{nd} order Markov model and the state is defined as ambiguous or non-ambiguous accordingly and as explained above. If prediction results in a non-ambiguous state, prediction will take place according to the all-2^{nd} order Markov model results. However, if prediction results in an ambiguous state, prediction for p_n will be computed

according to the association rules that correspond to that particular state.

4.4.3 Integration Example

The following example is used in order to clarify the integration method implemented. In general, and using the traditional individual Markov model techniques, the example results in two similar prediction probabilities. As a consequence, any one of them could be used for prediction. However, using our integration approach and implementing association rules, we are able to look back at history and identify the particular accessed page that leads to a more correct prediction probability. Using association rules in this case reduces the number of rules accessed during the prediction process.

The example considers a set of Web page structure for an online computer shop in Figure 4.3. Note that letters are assigned to nodes names in Figure 4.2 for simplicity purposes. Table 4.4 examines the following 6 user sessions: Calculating

Table 4.4: User sessions

T1	A,C,G,A,D,H,M,C,F,C,G,R,I,P,H,O,J
T2	A,G,T,A,C,S,G,J,R,A,D,H,M,D,J
T3	A,F,I,B,A,E,D,H,N,P,I,Q,F,J,D,H,N,G,C
T4	A,I,J,B,A,E,C,T,D,H,M,I,Q,G
T5	F,D,H,N,J,A,D,A,E,D,J,R,H,N,G,C,F,G
T6	F,L,S,D,H,N,J,Q,E,I,P,C,I,O,A,D,H,M

the frequencies of accessed pages, Table 4.5 lists the pageviews with their frequencies. A 0% support results in a very large number of rules and is rather cumbersome. Therefore, assuming that the minimum support is 4; B, K, L, O, P,

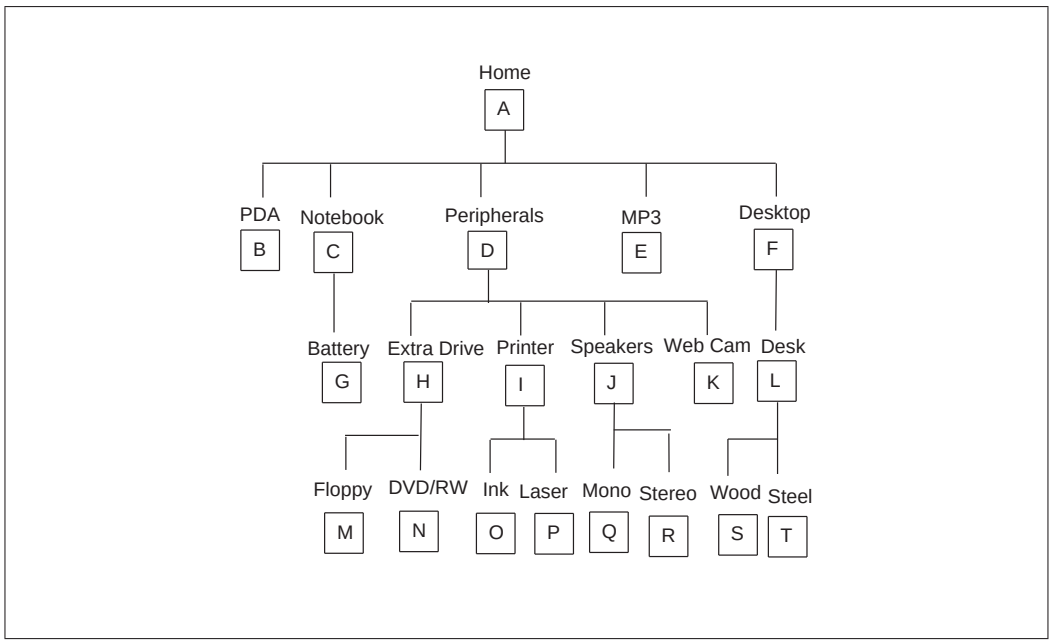

Figure 4.3: Online computer store Web page structure.

Table 4.5: Pageviews frequencies

Page	A	B	C	D	E	F	G	H	I	J
Freq	12	2	8	11	4	6	8	10	7	8

Page	K	L	M	N	O	P	Q	R	S	T
Freq	0	1	4	4	3	3	3	3	2	2

Q, R, S and T are removed from the itemsets. Table 4.6 lists the user sessions that pass the frequency and support tests. Applying the 2^{nd} order Markov Model to the

Table 4.6: User sessions after frequency and support pruning

T1	A,C,G,A,D,H,M,C,F,C,G,R,I,H,J
T2	A,G,A,C,G,J,A,D,H,M,D,J
T3	A,F,I,A,E,D,H,N,I,F,J,D,H,N,G,C
T4	A,I,J,A,E,C,D,H,M,I,G
T5	F,D,H,N,J,A,D,A,E,D,J,H,N,G,C,F,G
T6	F,D,H,N,J,E,I,C,I,A,D,H,M

above training user sessions we notice that the most frequent state is $\langle D, H \rangle$ and it

appeared 8 times as follows:

$$P_{l+1} = \text{argmax}\{P(M|H,D)\} \text{ OR}$$

$$P_{l+1} = \text{argmax}\{P(N|H,D)\}$$

Obviously, this information alone does not provide us with correct prediction of the next page to be accessed by the user as we have high frequencies for both pages, M and N. To break the tie and 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 shows in Table4.7 below.

Table 4.7: User sessions history

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

Table 4.8 and Table 4.9 summarise the results of applying subsequence association rules to the training data. Table 4.8 shows that C $\rightarrow$ M has the highest confidence of 100%, while Table 4.9 shows that F $\rightarrow$ N has the highest confidence of 100%. The confidence is calculated according to the following equation that was explained in Section 3.3 of Chapter 3:

$$\alpha = conf(A) = \frac{supp(\langle A,P \rangle)}{supp(A)} \tag{4.5}$$

Table 4.8: Confidence of accessing page M using subsequence association rules

A → M	4/10	40%
C → M	4/4	100%
E → M	2/3	67%
F → M	0/4	0%
G → M	2/3	67%
I → M	2/5	40%
J → M	3/4	67%

Table 4.9: Confidence of accessing page N using subsequence association rules

A → N	1/10	10%
C → N	0/4	0%
E → N	1/3	33%
F → N	4/4	100%
G → N	0/3	0%
I → N	2/5	40%
J → N	1/4	25%

Using Markov models, we can determine that there is a 50/50 chance that the next page to be accessed by the user after accessing the pages D and H could be either M or N. Whereas subsequence association rules take this result a step further by determining that if the user accesses page C before pages D and H, then there is a 100% confidence that the user will access page M next. Whereas, if the user visits page F before visiting pages D and H, then there is a 100% confidence that the user will access page N next.

Applying this result back to our example, we find that if the user buys a notebook, there is more chance that he/she will buy an external floppy drive. However, if the user buys a desktop, there is more chance that he/she will buy an extra DVD/RW drive. This extra bit of information is very important as knowing user

browsing history gives us an added advantage of knowing the browsing habits of our users.

4.5 Experimental Evaluation

In this section, we present experimental results to evaluate the performance of our algorithm. All experiments were conducted on a P4 2 GHz 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). Consider Figure 4.4, 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. Typically, the Web server logs contain millions of records, where each record refers to a visit by a user to a certain Web page served by a Web server. 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, August 30 1995. There were 47,748 total requests, 46,014 GET requests, 1,622 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, August 22 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 Web site 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 June 1, 1995 through June 7, 1995, a total of seven days. In this one week period there were 44,298 requests.

```
refofc1.lib.montana.edu [30:12:32:49] GET
/docs/cie/summer95/issue01j.wpd HTTP/1.0 200
549
ip61.b2.wsnet.com [30:12:32:50] GET
/docs/PressReleases/1995/August/Day−25/pr−428.html
HTTP/1.0 304 0
arctic.nad.northrop.com [30:12:32:51] GET
/logos/small_gopher.gif HTTP/1.0
200 935
ip61.b2.wsnet.com [30:12:32:53] GET
/docs/PressReleases/1995/August/Day−25/pr−427.html
HTTP/1.0 200 1944
ees−13−mso−pc7.lanl.gov [30:12:32:55] GET /
HTTP/1.0 200 4888
```

Figure 4.4: Example Web log.

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 minute 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. We consider a Web log as a data set D that is defined by a set of values $A = A_1, A_2, ..., A_m$. The values are usually the host making the request, date and time of request, requested page, HTTP reply code and bytes in the reply. A transaction T is identified by a subset of attributes A where $T \subset A$. Let U be a set of user ids and F a function that maps each unique combination of values of T to a user id of U where $F : T \rightarrow U$. F is used to derive a new user ID A_U in D. The page access time is designated by A_t. Let $A_k(t_I)$ be the value of A_k in the I^{th} transaction of data set D. Let W be a user session including a sequence of pages visited by the user in a visit where $D = \{W_1, ..., W_N\}$. The Web sessions W_N are defined as follows:

Definition 4.3: *A session W is an ordered set of transactions T in data set D which satisfy $A_U(t_{I+1}) = A_U(t_I)$ and $A_t(t_{I+1}) - A_t(t_I) < \tau$ where t_{I+1}, $t_I \in T$ and τ is a given time threshold of 30 minutes.*

Table 4.10 represents the different data sets after preprocessing. 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. Also, the frequency of each page visited by the user was calculated. The page access frequency of the EPA log file is shown in Figure 4.5 which reveals that page number 3 is the most frequent page and it was accessed 73 times.

Table 4.10: Sessions

	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

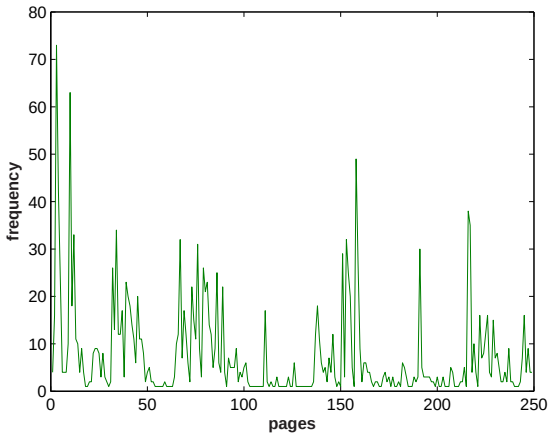

Figure 4.5: Frequency chart for the most frequent visited pages.

4.5.1 Experiments Results

Having all data sets processed, filtered and analysed, 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order Markov models were created. Then, all 1^{st}, 2^{nd}, 3^{rd}, and 4^{th} order frequency pruned Deshpande & Karypis (2004) Markov model analysis took place consid-

ering 4 as the frequency threshold. Prediction results were achieved using the maximum likelihood based on conditional probabilities as stated in equation 3.3 in Chapter 3. All implementations were carried out using MATLAB.

Figure 4.6, Figure 4.7, Figure 4.8 and Figure 4.9 below illustrate the difference between Markov model orders and Frequency pruned all-k^{th} Markov model results for data sets D1, D2, D3 and D4 respectively. The Figures demonstrate that as the order of Markov model increases, accuracy decreases due to the reduced coverage of the data. Coverage is defined as the ratio of the Web sessions in the test set that have a corresponding state in the training set to the number of Web sessions in the test set Deshpande & Karypis (2004). Also, the increase of the frequency pruned Markov model accuracy is limited due to the elimination of states that could be of importance to the precision process. The frequency threshold parameter used was a fixed parameter of size 4.

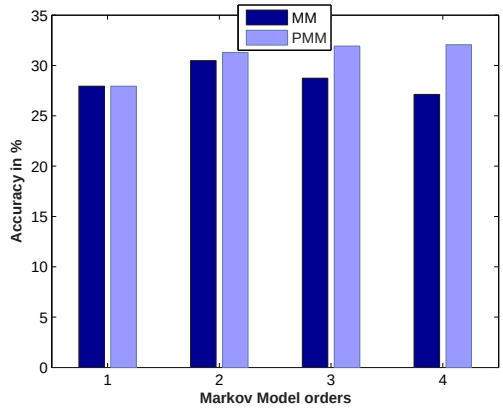

Figure 4.6: Accuracy of 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order Markov models and all 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order frequency pruned Markov models for data set D1.

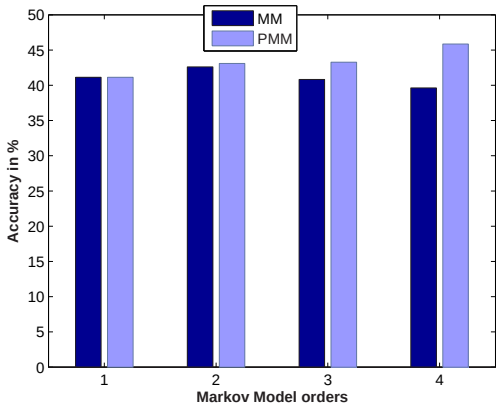

Figure 4.7: Accuracy of 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order Markov models and all 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order frequency pruned Markov models for data set D2.

Table 4.1 and Table 4.2 reveal that the all- 1^{st}, 2^{nd}, 3^{rd}, and 4^{th} order frequency pruned Markov models have considerably less states than the 1^{st}, 2^{nd}, 3^{rd}, and 4^{th} order Markov models.

The 1^{st} order and 2^{nd} order Markov model results cannot be 100% reliable simply because we did not look back into the history of pages accessed by the user. We assumed that the pages visited long before the current page in a Web session do tend to influence the users actions. These previously accessed pages affect the prediction process as they interfere with the user browsing behaviour and are not mere information providers. Performing 3^{rd} and 4^{th} order Markov models techniques solves the problem of examining the users previous browsing behaviour, but it results in an increase in the number of states as it is obvious in Table 4.1 and Table 4.2 that illustrate the number of states generated based upon non empty states. To overcome this shortcoming, we applied subsequence association rule techniques in order to generate the most appropriate rule. Before

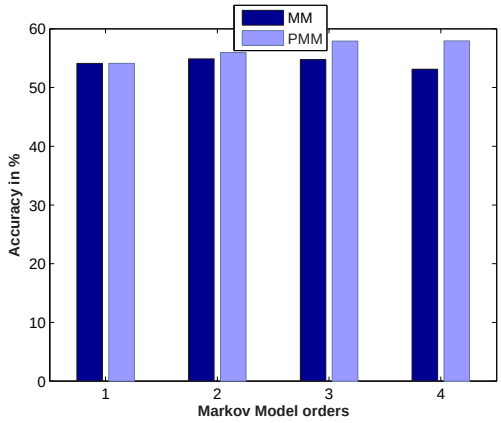

Figure 4.8: Accuracy of 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order Markov models and all 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order frequency pruned Markov models for data set D3.

applying association rule techniques, the most frequent occurrences or the Markov model frequent states are removed.

Since association rule 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.

Figure 4.10 below displays 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 4.11 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 Figure 4.10 and Figure 4.11 below that the influence of the minimum support factor is much

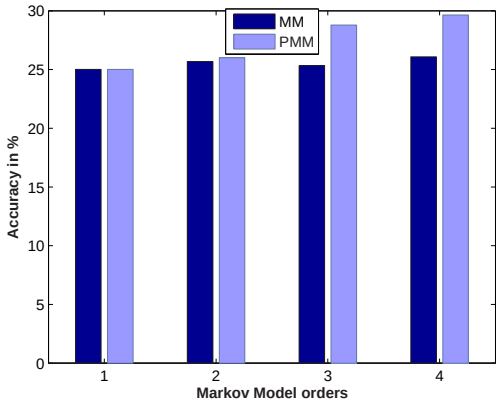

Figure 4.9: Accuracy of 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order Markov models and all 1^{st}, 2^{nd}, 3^{rd} and 4^{th} order frequency pruned Markov models for data set D4.

greater on the number of rules than the influence of the confidence factor.

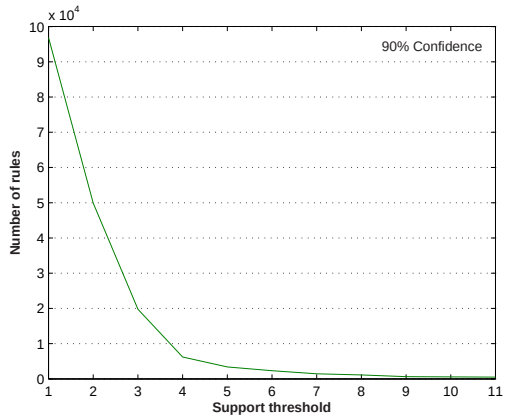

Figure 4.10: Number of rules generated according to different support threshold values and a fixed confidence factor: 90%.

Larger minimum support means less number of rules but it could also mean that genuine rules might be omitted. Figure 4.12 depicts the time complexity of generating association rules using different values of σ for D1 data set.

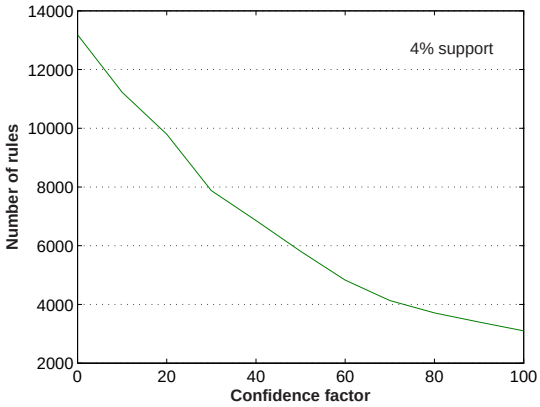

Figure 4.11: No. of rules generated according to a fixed support threshold: 4%.

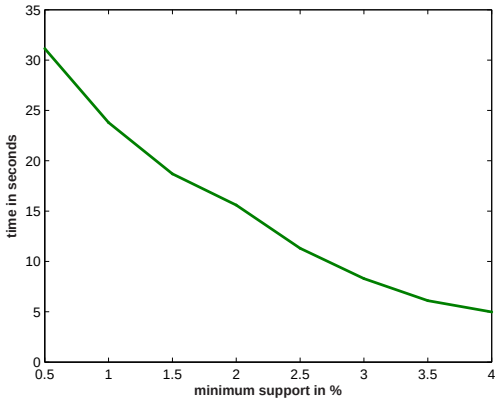

Figure 4.12: Time complexity in seconds for different support value.

4.5.2 Integration Model (IMAM) Accuracy Results

The integration model, IMAM, involves calculating association rule techniques prediction accuracy using the longest match precision method introduced in Section 3.3 of Chapter 3. In IMAM, association rules is applied in two cases:

The first case is when we are unable to make a correct prediction in the case of a 2^{nd} order Markov model because of a tie. In such a case, using association rule

techniques to look further back at previously visited pages, we were able to break the tie by looking at the page in history that leads to the most appropriate page for prediction. Looking at Figure 4.13, using 1^{st} order Markov model, the most frequently accessed page after EPA-PEST1995Aug23 is EPA-PEST1995Aug17 with 100% probability. Using 2^{nd} order Markov model, the most frequently accessed pages after EPA-PEST1995Aug17 are EPA-PEST1995July and OOPTPubs with 50% probability each. To decide which of the two pages would result in higher prediction accuracy, we look further back. Using association rules we find out that there is 100% chance that if EPA-PEST1995Aug16pr-373 is accessed before EPA-PEST1995Aug23, EPA-PEST1995July will be accessed next. And, there is 100% chance if PressReleases1995Aug is accessed before EPA-PEST1995Aug23, OOPTPubs will be accessed next. As a result, precision is calculated according to the results of association rules.

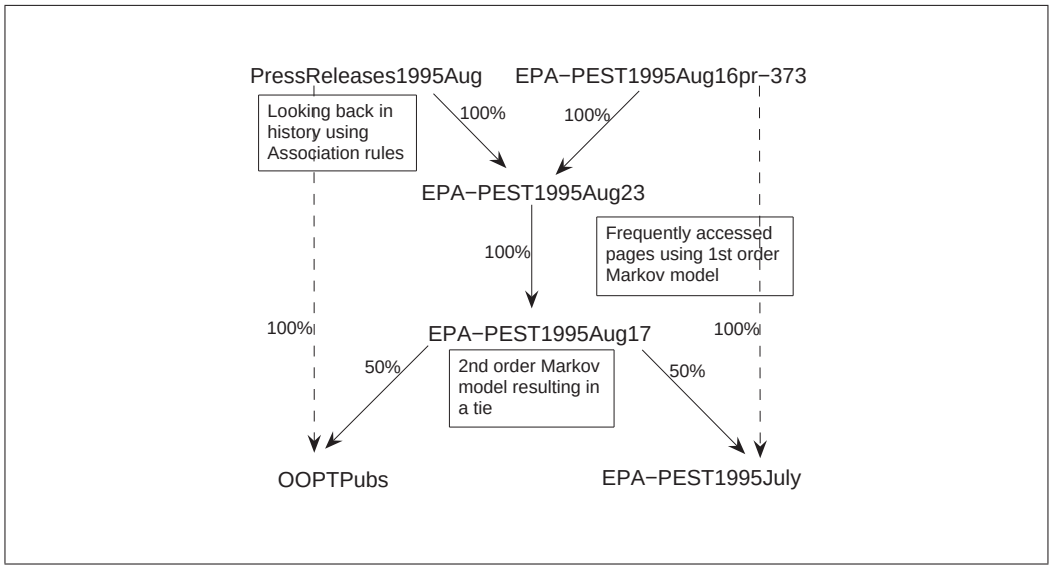

Figure 4.13: Portion of association rules results.

The second case when we use association rules is if the test data does not match any of the 2^{nd} order Markov model outcomes, we use the globally generated association rules to look back at previous user browsing history. Users have different browsing experiences, some of them get to the page they request using a shorter path than others depending upon the Web site structure and internal links. For example, the same page could be accessed by a user after visiting 5 pages and by another user after visiting 2 pages.

The Markov model prediction accuracy or prediction probability was computed by dividing the number of times the test page was visited immediately after the previous page to the number of times the previous page was visited. This was based on conditional probability. The accuracy of the proposed IMAM model was calculated by adding all successes and dividing the result by the number of states in the test data. When computing accuracy, we considered a minimum support threshold of 4%, minimum confidence threshold of 90% and a window of size 4. The reported accuracies in this section are based on 10-fold cross validation. The data was split into ten equal sets. First, we considered the first nine sets as training data and the last set for test data. Then, the second last set was used for testing and the rest for training. We continued moving the test set upward until the first set was used for testing and the rest for training. The reported accuracy is the average of ten tests. The accuracy of the IMAM model was compared to that of association rules and frequency pruned 2^{nd} order Markov model for four data sets in Figure 4.14, Figure 4.15, Figure 4.16 and Figure 4.17 be-

low. According to the accuracy figures, there is a consistency in the results and the proposed IMAM model evidences better accuracy than association rules (AR) and frequency pruned all 2^{nd} order Markov model (PMM). However, data set D1 and data set D4 have benefited most from the integration model showing more significant prediction accuracy improvement. Also, the figures reveal the increased accuracy of using PMM over association rules due to the known limitations of using association rules for Web page prediction as discussed in Section 4.3.1.

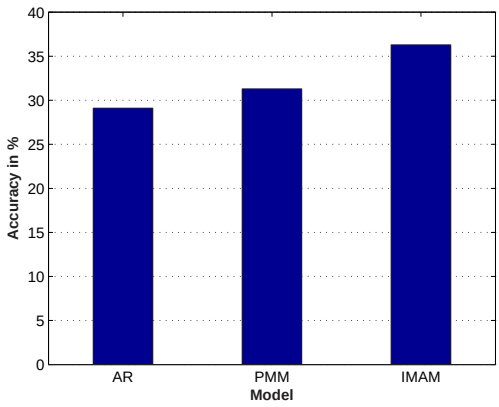

Figure 4.14: Accuracy of Association rules (AR), Frequency Pruned all 2^{nd} order Markov model (PMM) and IMAM model for data set D1.

The main problem associated with this approach is that it is dependent on the length of user sessions of data available. This is usually not a problem when modelling a particular site with long user sessions and therefore, more history. But it becomes more difficult when performing multi-site analysis with shorter user sessions. In our work, we considered a session with five pages as the minimum session length. The five page session length is reached after data filtering and preprocessing.

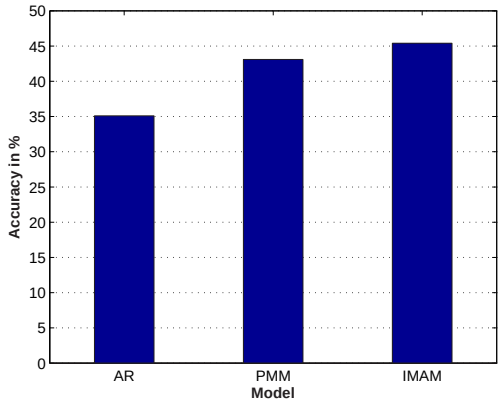

Figure 4.15: Accuracy of Association rules (AR), Frequency Pruned all 2^{nd} order Markov model (PMM) and IMAM model for data set D2.

4.5.3 Comparing IMAM to a Higher Order Markov Model

4.5.3.1 State Space Complexity

In this dissertation, the word ”state” refers to Markov model, association rules and clustering rules or states interchangeably. In this chapter, we refer to IMAM number of states as the summation of Markov model number of states and association rules number of rules. The IMAM state space complexity includes the 2-PMM complexity as well as the number of association rules involved in the case of ambiguity. In this section, we compare the state space complexity of the IMAM model to that of a higher order Markov model, 3^{rd}-order frequency pruned Markov model (3-PMM). Table 4.11 compares the 3-PMM states with those of IMAM states for all four data sets.

Table 4.11 reveals that total number of states (including Markov model and association rules) is less than the number of states generated using 3-PMM and

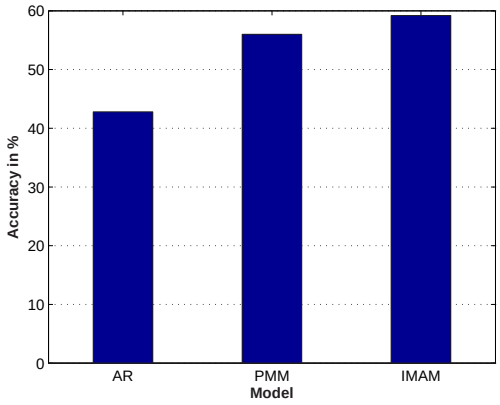

Figure 4.16: Accuracy of Association rules (AR), Frequency Pruned all 2^{nd} order Markov model (PMM) and IMAM model for data set D3.

Table 4.11: IMAM number of states

	D1	D2	D3	D4
3-PMM	14,977	18,121	11,218	19,032
IMAM	10,071	7,054	6,123	9,247
3-MM	72,524	89,815	50,971	90,123

much less than the number of states generated using 3-MM for all four data sets. This concludes that our model, IMAM, not only improves the Web page access prediction accuracy, but also reduces the state space complexity.

4.5.3.2 Accuracy

Higher order Markov models improves prediction accuracy but result in higher state space complexity. In this section we compare IMAM accuracy to that of the frequency pruned 3^{rd}-order Markov model accuracy. Figure 4.18 displays the results.

Although the prediction accuracy improvement is not significant, the IMAM

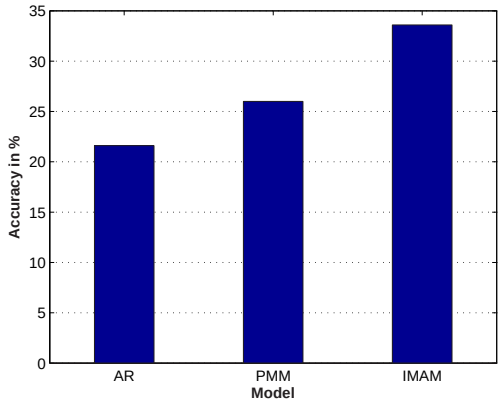

Figure 4.17: Accuracy of Association rules (AR), Frequency Pruned all 2^{nd} order Markov model (PMM) and IMAM model for data set D4.

accuracy is higher than that of 3-PMM for all four data sets.

4.6 Conclusion

In this Chapter, we introduced a method to integrate Markov model and association rules for predicting Web page accesses. The integration is based on a low order Markov model. Sets of subsequence association rules are used to complement the Markov model for resolving ambiguous predictions by using long history data. The integration avoids the complexity of high order Markov model and the limitation of Markov model using short history. This model also reduces the complexity associated with large number of association rules since association rules are only used when ambiguous predictions occur. The experimental results show that the combined model increases the accuracy of the Web page access prediction of the individual Markov model and association rule techniques.

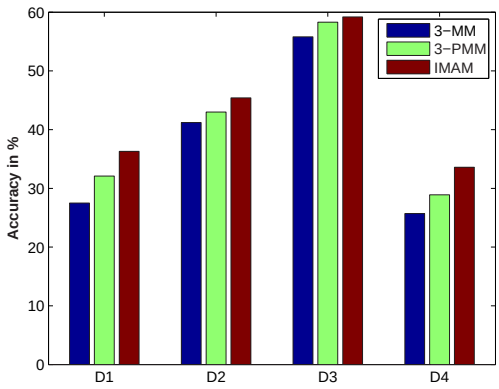

Figure 4.18: Accuracy of 3^{rd} order Markov model (3-MM), frequency pruned all 3^{rd} order Markov model (3-PMM) and IMAM model for all four data sets.

Chapter 5

Integrating Markov Model with Clustering

5.1 Introduction

Based on the improved prediction accuracy attained through the integration of Markov model and association rule techniques in Chapter 4, it is worth canvassing the results of integrating Markov model with another prediction algorithm, clustering. As explained in Chapter 4, Markov model is the most commonly used prediction model because of its high accuracy. Low order Markov models have higher accuracy and lower coverage than sequential association rules and clustering techniques Kim et al. (2004). In order to overcome low coverage, all-k^{th} order Markov models have been used Pitkow & Pirolli (1999) where the highest order is first applied to predict a next page. If it cannot predict the page, it decreases the order by one until prediction is successful. This can increase the coverage, but it is associated with higher state space complexity. Clustering methods are unsupervised methods, and normally are not used for classification

directly. However, proper clustering groups user sessions with similar browsing history together. Clusters are employed to guide the prediction system. They help predict the Web pages that are close to a user-requested page in a cluster model. Similar to the other prediction models, the cluster model tries to discover the statistical correlation between Web pages using Web access patterns mined from a Web log. However, prediction is performed on the cluster sets rather than the actual sessions. The main issue that affects the clustering accuracy is producing the selected features for partitioning. For instance, partitioning based on semantic relationships or contents Banerjee & Ghosh (2001) or link structure Zhu et al. (2002*b*) usually provides higher accuracy than partitioning based on bit vector, spent time, or frequency. However, even the semantic, contents and link structure accuracy is limited due to the unidirectional nature of the clusters and the multidirectional structure of Web pages.

This chapter involves implementing a clustering algorithm to partition Web sessions into clusters and then applying Markov model techniques to each cluster in order to achieve better accuracy and performance of next page access prediction keeping the number of states to a minimum. Section 2 introduces clustering and examines the problems associated with it. Section 3 explains the integration process including the new integration algorithm. In Section 4, we prove our new model experimentally and Section 5 concludes the chapter.

5.2 Clustering

This chapter discusses the previously introduced model, Markov model and it introduces clustering techniques for Web page prediction. Each of these algorithms can be solely used for Web page prediction with some limitations. Markov model problems and limitations were covered in Chapter 4 earlier. This section concentrates on having an insight into using clustering methods for Web page prediction and the problems encountered in the process.

5.2.1 Limitations of Clustering Techniques

Despite the variety of clustering approaches that have been used for Web usage mining, clustering alone is not an appropriate approach for Web page prediction Kim et al. (2004). Clustering involves partitioning pages or sessions into similar groups. Prediction takes place based on these groups. This process leads to decreased precision because it does not use all the pages directly. Clusters constructed based on features like content, semantics or link structure Banerjee & Ghosh (2001), Yan et al. (1996), Zhu et al. (2002*b*) have proved to outperform clusters constructed based on bit vector (visit or non-visit), spent time or frequency. However, even such improved feature selection does not always accommodate well partitioned clusters. Another problem associated with clustering is its online prediction process that could be more time expensive than Markov models or association rules. For every new instance, prediction takes place by

calculating the closest distance between the new instance and the mean of every cluster. This is performed online and requires real-time calculations. However, using association rules or Markov models, prediction is performed by matching the new instance to an existing look up table that is built offline. Also, clustering is not designed for classification using supervised learning. It is merely used to segment data into some homogenous groups so that a quality model can be built on each group.

Another clustering limitation is the ability to evaluate and compare their performance. The reason for this is the lack of an objective evaluation criteria that is independent of the specific application. Until now, there is lack of information about the correct clusters to be identified and any solution is valid until it gets rejected by an expert in the field. This makes clustering results inherently difficult to evaluate.

5.2.2 Using Markov Model and Clustering for Prediction

Web page prediction has gained its importance due to the accelerated number of Web applications and search engines. Markov model and clustering are two frameworks used for predicting the next page to be accessed by the Web user. Many research papers addressed Web page prediction by using clustering, Markov model or a combination of both techniques.

For instance, Kim et al. (2004) combine most prediction models 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 Web site link structure and the support and confidence thresholds. These factors affect the order of the applied models and the performance of the new model.

Other papers combined clustering with Markov model Cadez et al. (2003), Zhu et al. (2002*b*), Lu et al. (2005). Cadez et al. (2003) partitioned site users using a model-based clustering approach where they implemented first order Markov model using the Expectation-Maximization algorithm. After partitioning the users into clusters, they displayed the paths for users within each cluster. They also developed a visualization tool called WebCANVAS based on their model. Zhu et al. (2002*b*) 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 proposed to cluster conceptually related pages. A hierarchy of the Web site is constructed from the clustering results. The authors then combine Markov model based link prediction to the conceptual hierarchy into a prototype called ONE to assist users' navigation. 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 ab-

straction 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.

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. Kim *et. al* used a combination of models but their work improved recall but did not improve the Web page prediction accuracy Kim et al. (2004). Our work proves to outperform previous works in terms of Web page prediction accuracy and state space complexity using a combination of clustering and Markov model techniques. We implement a simple clustering algorithm, *k*-means algorithm where using different distance measures can lead to different results. A frequency pruned 2^{nd} order Markov model was used for the prediction purposes.

Figure 5.1 below describes the stages of the clustering process before Markov model implementation is carried out.

5.3 Integration Process

This Chapter provides an alternative solution to Chapter 4. The focus of this Chapter is on improving the Web page access prediction accuracy and state space complexity by combining Markov model and clustering techniques. This section explains the Markov model and clustering integration process.

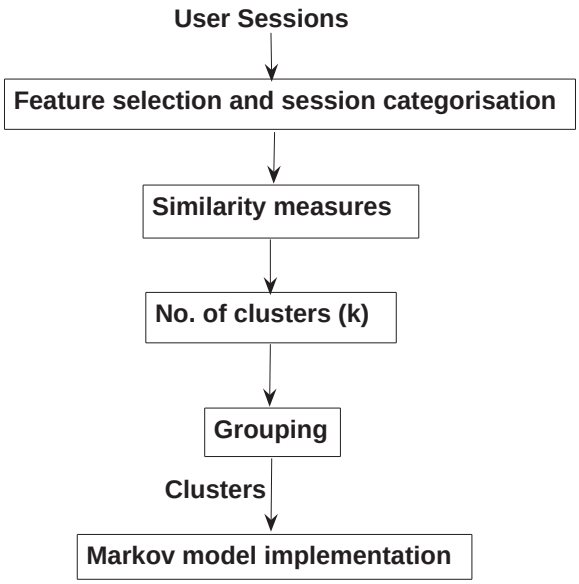

Figure 5.1: The stages of clustering before Markov model implementaion.

5.3.1 Motivation for Integration

Web page prediction involves anticipating the next page to be accessed by the user or the link the Web user will click at next when browsing a Web site. For example, what is the chance that a Web user visiting a site that sells computers will buy an extra battery when buying a laptop. Or, may be there is a greater chance the user will buy an external floppy drive instead. Users past browsing experience is very fundamental in extracting such information. This is when modeling techniques come at hand. For instance, using clustering algorithms, we are able to personalize users according to their browsing experience. Different users with different browsing behavior are grouped together and then prediction is performed based on the user's link path in the appropriate cluster. Similar kind of prediction can be

in effect using Markov models conditional probability. For instance, if 50% of the users access page D after accessing pages A, B, C, then there is a 50/50 chance that a new user that accesses pages A, B, C will access page D next. Our work improves the Web page access prediction accuracy and state space complexity by combining both Markov model and clustering techniques. It is based on dividing Web sessions into groups according to Web services and performing Markov model analysis on each cluster of sessions instead of the whole data set. This is very significant since a Markov model for a sub group, that is assumed to be more homogeneous than the whole data set, has a higher quality than the Markov model of the whole data set. As a consequence, performing Markov model analysis on a functionally related sessions leads to more accurate prediction than performing such analysis on the whole data set.

Markov models are the most effective techniques for Web page access prediction and many researchers stress the importance in the field Bouras & Konidaris (2004), Chen et al. (2002), Deshpande & Karypis (2004), Eirinaki et al. (2005), Zhu et al. (2002*b*). Other researchers use Markov models to improve the Web server access efficiency either by using object prefetching Pons (2006) or by helping reduce the Web server overhead Mathur & Apte (2007). Lower order Markov models are known for their low accuracy due to the limited availability of users' browsing history. Higher order Markov models achieve higher accuracy but are associated with higher state space complexity. Although clustering techniques have been used for personalization purposes by discovering Web site structure

and extracting useful patterns Adami et al. (2003), Cadez et al. (2003), Papadakis & Skoutas (2005), Rigou et al. (2006), Strehl et al. (2000), usually, they are not very successful in attaining good results. Proper clustering groups users sessions with similar browsing history together, and this facilitates classification. However, prediction is performed on the cluster sets rather than the actual sessions.

The integration of Markov model and clustering (IMC) is based on low-order Markov model for the same reasons explained in Chapter 4, Section 4.2.3. Using low-order Markov model, we avoid the state space complexity associated with higher order Markov models at the expense of accuracy loss. We compensate for the low order Markov model lower accuracy by using clustering techniques. Web sessions are first identified and grouped according to functionality and using meaningful features. Then, the Web sessions are grouped into a number of categories. *K*-means clustering algorithm is based on Web session categories identified and is carried out according to some distance metrics. The purity of the clusters is evaluated using the entropy technique. A major process of Web sessions clustering using *k*-means algorithm is the determination of the number of clusters (*k*). This is accomplished through an enhanced version of k-means clustering algorithm, ISODATA.

Proper grouping and clustering of Web sessions helps increase the Web page access prediction accuracy. On the other hand, using frequency pruned k^{th}-order Markov model helps keep the state space complexity to a minimum. The integra-

tion of Markov model and clustering (IMC) architecture is depicted in Figure 5.2 below:

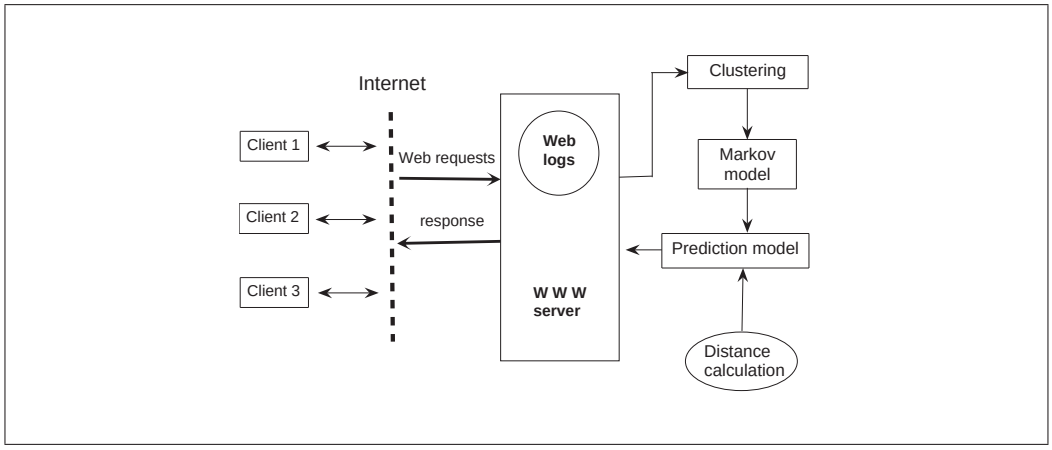

Figure 5.2: The integration model (IMC) architecture.

5.3.2 Integration Algorithm

The training process takes place as follows:

```
(1)Use feature selection, allocate similar Web sessions to
appropriate categories.

(2)Decide on a suitable $k$-means algorithm distance measure.

(2)Decide on the number of clusters k and partition the Web sessions
into clusters.

(3)FOR each cluster

(4)  return the data to its uncategorized and expanded state.

(5)  Perform Markov model analysis on each of the clusters.

(6)ENDFOR
```

The prediction process or test phase involves the following:

```
(1)FOR each coming session
(2)  Find its closest cluster
(3)  Use the corresponding Markov model to make prediction
(4)ENDFOR
```

The clustering task begins with user sessions identification. It then divides the multi-dimensional space into a number of groups of Web transactions. Each group contains transactions that are close to each other according to a distance measure or similarity among the vectors. Prediction is then performed on the discovered groups of transactions rather than the individual sessions. To make a prediction, the new item i (or session W) is assigned to the proper group of transactions or cluster. To achieve this, the centroid vector corresponding to each cluster is computed and used as the aggregate representation of the cluster. The new item i is assigned to the cluster with a vector centroid closest to i. Once the closest cluster to the item is identified, prediction accuracy is calculated on that particular cluster using one of the pattern discovery algorithms, in our case, the frequency pruned 2^{nd}-order Markov model.

5.3.2.1 Feature Selection

The first step of the training process is feature selection and categorisation. 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 Web sites 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.

5.3.2.2 Session Categorisation

Combining similar pages or assigning similar Web pages to categories is an important step in the training process of the IMC model. Categorisation or labeling 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.

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. $D = \{W_1, ..., W_N\}$. For session identification definition refer to Definition 4.3 in Section 4.5 of Chapter 4. Let $P = \{p_1, p_2, \ldots, p_m\}$ be a set of pages in a Web site. Since Markov model techniques will be implemented on the data, the pages have to remain in

the order by which they were visited. $W_i = (p_1^i, ..., p_L^i)$ is a session of length L composed of multivariate feature vectors p. 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 $n < m$. For each session, a binary representation is used assuming each page is either visited or not visited. If the page is visited, a weight factor w is added to the pages representing the number of times the page was visited in the new session S_i. $S_i = \{(c_1^i, w_1^i), ..., (c_L^i, w_j^i)\}$. D_s is the data set containing N number of sessions S_N. The categories are formed as follows:

Input: D containing N number of sessions W_N.

(1)FOR each page p_i in session W_i

(2) IF $p_i \subset C_i$

(3) w_i.count++

(4) ELSE,

(5) $w_i = 0$

(6) ENDIF

(7)ENDFOR

Output: D_s 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 5.1 below.

Before categorisation, preprocessing of Web sessions takes place and each page

Table 5.1: Example: initial Web sessions

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

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 5.2.

When performing categorisation, let us say, we find out that we have two cate-

Table 5.2: Example: Preprocessed Web sessions

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

gories and pages 1 and 2 belong to category1 and page 3 belongs to category2. The Web sessions become as it appears in Table 5.3 below.

Thus, using categorisation, the three initial Web sessions ended up being of equal

Table 5.3: Web sessions after categorisation

Category	1	2
S1	3	2
S2	3	0
S3	1	2

length. also, the length of the categorised sessions is shorter because the number of categories is usually smaller than the number of pages.

5.3.2.3 *k*-means Distance Measures

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. Define a set of items (n-by-p data matrix) to be clustered.
2. Define a chosen number of clusters (k).
3. Randomly assign a number of items to each cluster.

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

1. Calculate the mean vector for all items in each cluster.
2. Reassign the items to the cluster whose center is closest to the item.

Because the first 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 considered. 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 items in each cluster.

Distances or similarities between items are a set of rules that serve as a method for grouping or separating items. The distance measured between items 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 assumed to be. 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.

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:

$$\text{Euclidean}(x,y) = \sqrt{\sum(x_i - y_i)^2} \tag{5.1}$$

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:

$$\text{Squared Euclidean}(x,y) = \sum(x_i - y_i)^2 \tag{5.2}$$

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:

$$\text{City Block}(x,y) = \sum |x_i - y_i| \tag{5.3}$$

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:

$$\text{Hamming}(x,y) = \sum |x_i - y_i| \tag{5.4}$$

The hamming distance is the percentage of bits that differ. This makes it unsuitable distance measure for our data sets because of the following:

1. 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.

2. The hamming distance measure takes into consideration only bits that differ and not the ones that are similar. This has a larger effect on larger data sets with different session lengths. For instance, consider the following three sessions extracted from D4 data set:

```
0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1
0 0 0 1 0 1 1 0 0 1 0 0 1 1 0 0 1
0 1 1 1 0 1 1 1 0 1 0 1 1 1 0 0 1
```

The first session has 5 pages, while the second session has 7 pages and the third has 11 pages.

The hamming distance between the first and the second session is 4. Also, the hamming distance between the second and the third session is 4. It is interesting to note that there are 4 pages that are common between the first and the second session, while there are 7 pages that are similar between the second and the third session. The hamming distance fails to show the closeness between the second and the third session.

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:

$$\text{Cosine}(x,y) = \frac{\sum(x_i y_i)}{\sqrt{\sum(x_i)^2 \sum(y_i)^2}} \tag{5.5}$$

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:

$$\text{Correlation}(x,y) = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2}\sqrt{\sum(y_i - \overline{y})^2}} \tag{5.6}$$

K-means computes centroid clusters differently for different *k*-means supported distance measures. Therefore, a normalization step was necessary for Co-

sine and Correlation distance measures for comparison purposes. The points in each cluster, whose mean forms the centroid of the cluster, are normalized 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:

$$\Lambda^{(E)}(C_l) = \sum \frac{n_l^{(h)}}{n_l} \log \left(\frac{n_l^{(h)}}{n_l} \right). \tag{5.7}$$

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 l belongs to category n_l. Entropy tends to favor 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 mea-

sures evaluations, in Section 5.4.3 below, 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 was 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:

$$\text{distCosine}(Sa, Sb) = \frac{\sum(Sa_i Sb_i)}{\sqrt{\sum(Sa_i)^2}\sqrt{\sum(Sb_i)^2}} \tag{5.8}$$

Table 5.4 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 5.4: Sessions

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

Table 5.5 reveals the distances calculated using equation 5.8:

Table 5.5: Sessions distances

$\text{distCosine}(S1, S2)$	0.019
$\text{distCosine}(S1, S3)$	0.89
$\text{distCosine}(S2, S3)$	1.0
$\text{distCosine}(S1, S4)$	0.88
$\text{distCosine}(S3, S4)$	0.06

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.

5.3.2.4 Number of Clusters (k)

The third step in the training process of the IMC 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 and 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 & 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 5.3 is based on Figure 14 in Jain et al. (1999). Figure 5.3 shows 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 5.3.

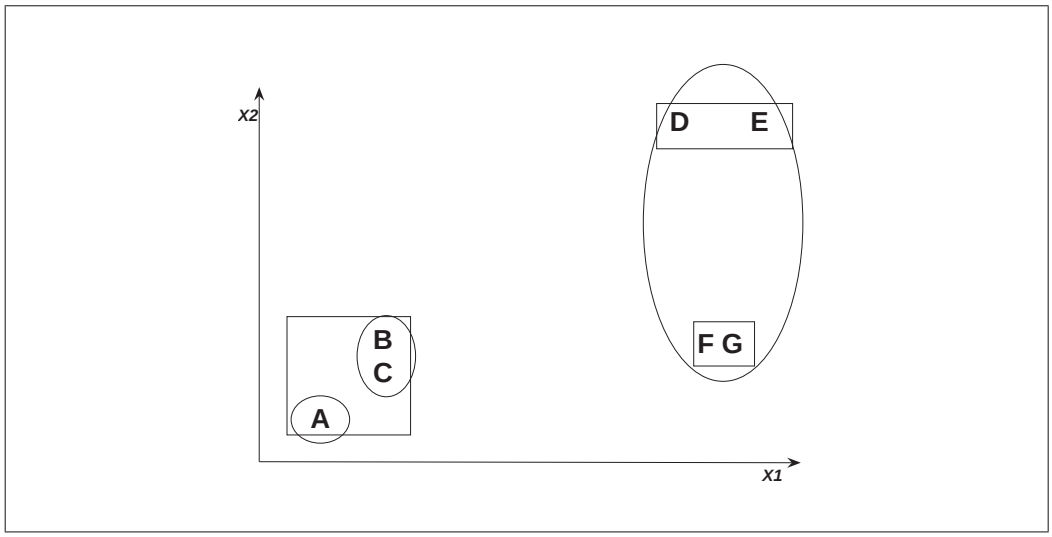

Figure 5.3: ISODATA improves the *k*-means clusters.

The running time of the ISODATA algorithm is the same as the running time of the *k*-means algorithm, $O(wkl)$ where w is the number of sessions, k is the number of clusters, and l is the number of iterations. Since k and l are fixed in advance, 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 $O(k+w)$.

5.3.2.5 Markov Model Implementation

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. Makov model implementation is carried out relying on the results accomplished in Chapter 4. Frequency pruned 2^{nd}-order Markov model is used for mining each of the clusters. Markov model implementation is performed according to the equation:

$$P_{l+1} = \text{argmax}_{p \in \mathbb{P}}\{Porb(P_{l+1} = p | p_l, p_{l-1}, \ldots, p_{l-(k-1)})\} \tag{5.9}$$

For instance, for every current page the user clicks at, p_c, the prediction model will estimate, using conditional probability, the probability of accessing the next page p_n by examining previously accessed pages. Constructing the 2^{nd} order Markov model, results in one of two cases:

$$prediction for p_n = \begin{cases} 0 & if P(p_n|p_c, p_p) = 0 \ and \\ & P(p_n|p_c) = 0 \\ \neq 0 & if P(p_n|p_c, p_p) \neq 0 \ or \\ & if P(p_n|p_c, p_p) = 0 \ and \\ & P(p_n|p_c) \neq 0 \end{cases}$$

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 as stated in Equation 3.3 in Chapter 3. All predictions in the test data that did not exist in the training data sets were assumed incorrect and were given a zero value.

5.3.2.6 Item-Cluster Proximity

During the prediction process, each new Web session the user accesses is examined and the appropriate cluster the new test item belongs to is identified. Let i_t be a new test item where $i_t \subset I$. Web sessions W are divided into K groups or clusters. The new item i_t has probability $prob(x_i = k)$ of belonging to cluster k where $\sum_k prob(x_i = k) = 1$ and x_i indicates the cluster membership of the new item i_t. The actual cluster k that the item 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 calculated in Equation (5.10), where k refers to the subscript of the components of the vectors i and μ.

$$\text{distCosine}(i_t, \mu) = \frac{\sum_{k=1}^{K}(i_t \mu)}{\sqrt{\sum_{k=1}^{K}(i_t)^2}\sqrt{\sum_{k=1}^{K}(\mu)^2}} \tag{5.10}$$

This process is carried out during the prediction stage. Although prediction using Markov model, or any other pattern discovery algorithm, is executed online and it does not require much time, allocating a new item to the closest cluster is also performed online but is more time complex. Markov model prediction accuracy is determined using the cluster the new item i_t belongs to.

5.3.3 Integration Example

The web data is heterogenous in nature. Each session is a collection of visited Web pages by the user. Every user has a different level of browsing expertise and sessions are formed mainly haphazardly because users usually follow different paths when trying to access the same page. Clustering combines similar Web page paths or user sessions together and subsets of data are therefore more homogeneous resulting in simpler Markov model computations.

By applying clustering to abstracted user sessions, it is more likely to find groups of sessions with similar pages that help increase the Markov model accuracy. For example, consider the four Web sessions in table 5.6 below: Using

Table 5.6: Example of user sessions.

W1	A , B , F , G , I
W2	A , C , D , G , I
W3	B , C , D , E , H
W4	B , C , D , E , F

ISODATA, we derive two clusters. Table 5.7 reveals cluster 1 and Table 5.8 reveals cluster 2.

Table 5.7: The first cluster.

W1	B , C , D , E , H
W2	B , C , D , E , F

Table 5.8: The second cluster.

W1	A , B , F , G , I
W2	A , C , D , G , I

Assume that there is a new Web session: A , B, C, D what is the probability that the new page to be accessed by the user is page E? According to *k*-means clustering algorithm, and according to the distance measure between the new data points and the data points in the existing clusters, the new session belongs to cluster 1. The Markov model analysis performed on the subset cluster 1 yields a 1.0 probability for accessing page E next. However, performing Markov model analysis on the whole data set, yields a 0.67 probability.

5.3.4 IMC Algorithm Efficiency Analysis

5.3.4.1 Clustering Complexity

All clustering runs were performed on a desktop PC with a Pentium IV Intel processor running at 2 GHz with 1 GB of RAM and 100 GB hard disk. The runtime of the *k*-means algorithm, regardless of the distance measure used, is equivalent to $O(nkl)$ Jain et al. (1999), where n is the number of items, *k* is the number of clusters and l is the number of iterations taken by the algorithm to converge. For our experiments, where n and *k* are fixed, the algorithm has a linear time complexity in terms of the size of the data set. The *k*-means algorithm has a $O(k+n)$ space complexity. This is because it requires space to store the data matrix. It is feasible to store the data matrix in a secondary memory and then the

space complexity will become $O(k)$. k-means algorithm is more time and space efficient than hierarchical clustering algorithms with O($n^2 logn$) time complexity and O(n^2) space complexity.

5.3.4.2 Prediction Complexity

As for the prediction process (online) complexity, IMC prediction model is more complex than prediction based on any of the individual pattern discovery models like association rules, clustering and Markov model. This is due to the necessity of the assignment of every new session to the appropriate cluster. This is more time consuming than accessing a mere look up table as is usually the case with the individual models. However, prediction is based on Markov model accuracy of one cluster as opposed to the whole data set. This reduces the prediction complexity of the IMC model.

5.4 Experimental Evaluation

5.4.1 Data Collection and Preprocessing

All experiments in this chapter were undertaken using the four data sets introduced in Section 4.5 of Chapter 4.

After Web session identification, session categorisation took place and the details of the number of categories for each data set are represented in Table 5.9.

After identifying all categories for each data set, it was necessary to run the

Table 5.9: Number of categories

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

session categorisation algorithm presented in Section 5.4.2. Table 5.10 below reveals part of session categorisation implemented on data set (D2). The first row represents the category number and each row thereafter represents a session. For instance, the first session has 7 pages where three pages belong to category 5, one page belongs to category 7, two pages belong to category 10 and one page belongs to category 11.

Table 5.10: Session categorisation

1	2	3	4	6	8	9	10	15	19	23	26	30	34	50
0	0	0	0	3	0	1	0	0	2	1	0	0	0	0
0	0	0	1	0	0	0	0	5	0	1	0	0	1	0
1	0	0	0	0	4	0	0	0	0	0	0	2	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	2	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	1	2	0

This session categorisation resulted in Web sessions of equal lengths. The extract in table 5.10 represents only around 10% of the actual categories. All categorised sessions were represented by vectors with the number of occurrence of pages as weights. This will draw sessions with similar pages closer together when performing clustering techniques. 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 4 data sets.

5.4.2 Number of Clusters (k)

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 & 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 $k = 10$. This proves that a larger number of Web sessions is best clustered using a larger k. All clusters were attained using Cosine distance measure. Figure 5.4 depicts the 7 clusters of data set D1, Figure 5.5 depicts the 9 clusters of data set D2, Figure 5.6 depicts the 14 clusters of data set D3 and Figure 5.7 depicts the 10 clusters of data set D4.

5.4.3 Distance Measures Evaluation

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

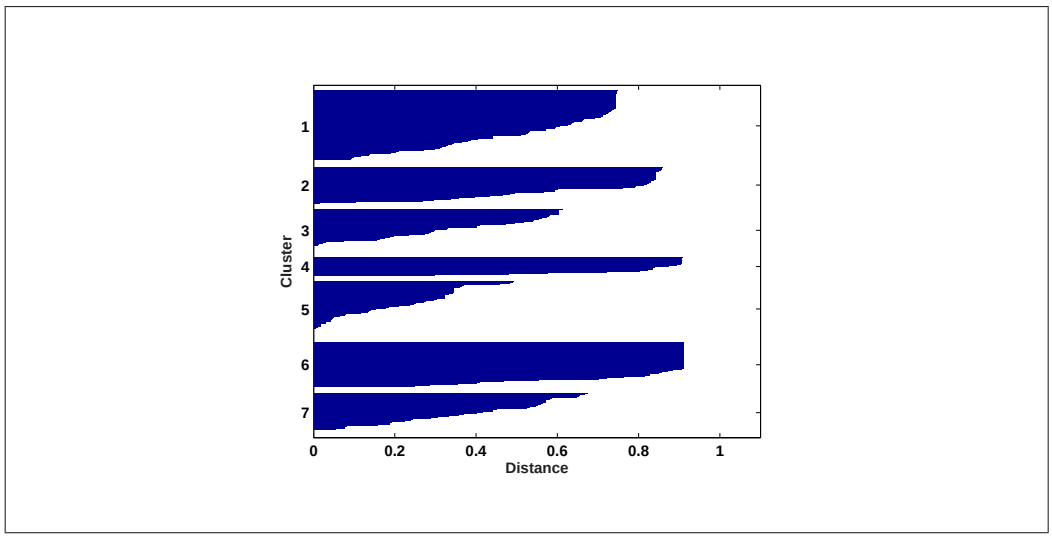

Figure 5.4: Silhouette value of D1 with 7 clusters.

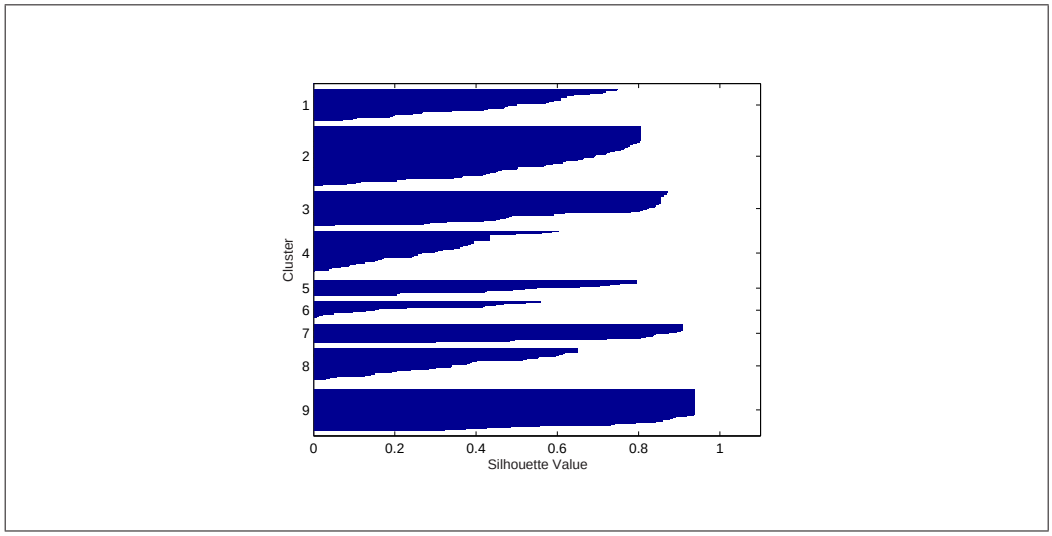

Figure 5.5: Silhouette value of D2 with 9 clusters.

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), Halkidi et al. (2003). In order to back up their findings, we calculate the entropy measures, we perform means analysis and we plot different

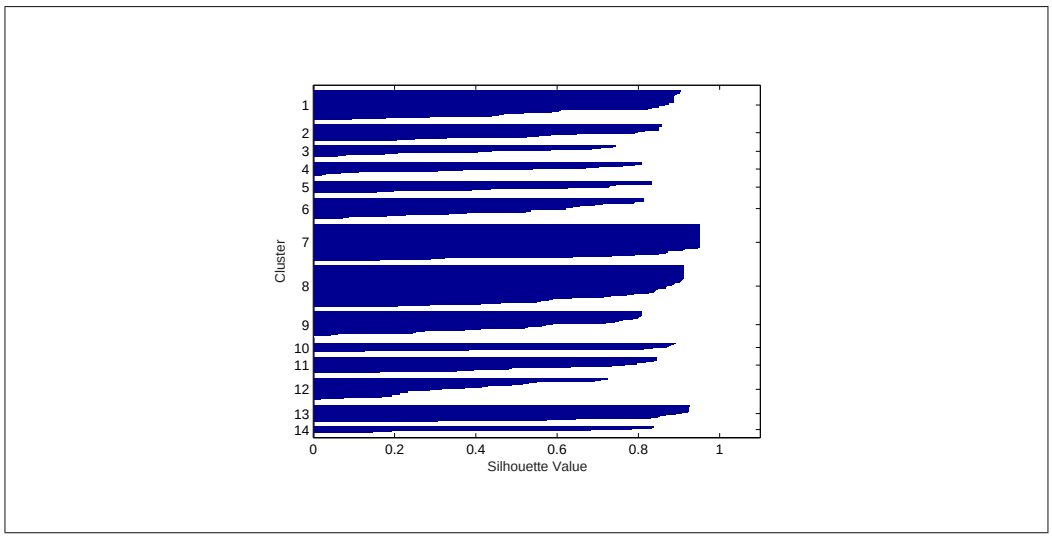

Figure 5.6: Silhouette value D3 with 14 clusters.

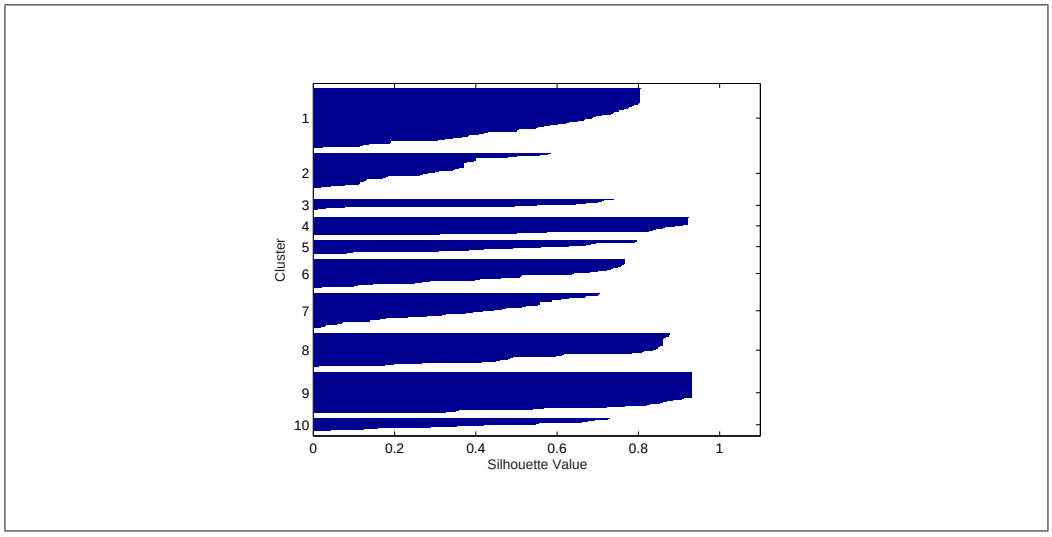

Figure 5.7: Silhouette value of D4 with 10 clusters.

clusters using different distance measures for data set D1. Table 5.11 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.

Table 5.11: Entropy measures for different clusters.

Clusters	2	3	4	5	6	7	8	9	10	20	30	40	50
Euclidean	0.42	0.38	0.32	0.58	0.31	0.28	0.25	0.30	0.26	0.21	0.19	0.23	0.22
City	0.52	0.48	0.50	0.49	0.46	0.42	0.39	0.31	0.29	0.27	0.25	0.24	0.23
Hamming	0.56	0.49	0.53	0.50	0.47	0.39	0.41	0.38	0.36	0.29	0.25	0.31	0.34
Cosine	0.36	0.32	0.37	0.43	0.25	0.21	0.22	0.21	0.17	0.16	0.19	0.22	0.23
Correlation	0.30	0.28	0.30	0.37	0.20	0.21	0.23	0.19	0.20	0.19	0.18	0.19	0.21

Figure 5.8, Figure 5.9, Figure 5.10, Figure 5.11 and Figure 5.12 represent clusters using Euclidean, Hamming, City Block, Pearson Correlation and Cosine distance measures respectively for data set D1. They 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 neighboring clusters. The silhouette measure ranges from +1, indicating points that are very distant from neighboring clusters, to 0, indicating points that do not belong to a cluster. The figures reveal 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 Figure 5.8 for Hamming distance is around 0.5, whereas, the silhouette value of Figure 5.11 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 neighboring clusters.

Figure 5.13 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 neighboring clusters. The higher the mean value, the better clusters we

get. It is worth noting that the information Figure 5.13 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 Figures 5.6-5.12, the distance mean values in Figure 5.13 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.

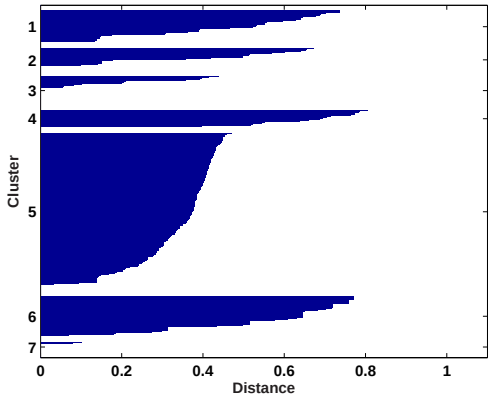

Figure 5.8: Silhouette value of Euclidean distance measure with 7 clusters.

5.4.4 Experiments Results

Web sessions in all four data sets were divided into clusters using the *k*-means algorithm and according to the Cosine distance measure. This grouping of Web sessions into meaningful clusters helps increase the Markov model accuracy. Table 5.12 below is an extract from the data set D1 clusters. It unveils how the clus-

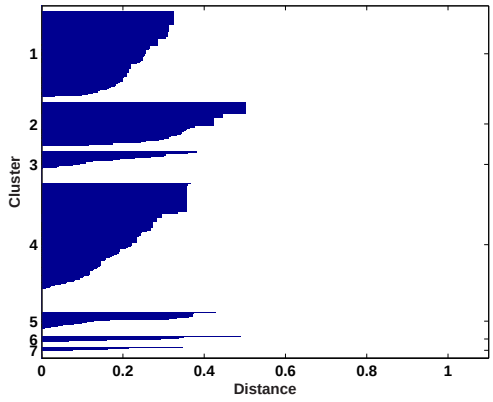

Figure 5.9: Silhouette value of Hamming distance measure with 7 clusters.

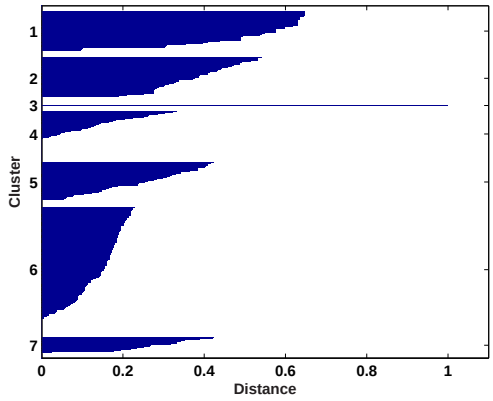

Figure 5.10: Silhouette value of City Block distance measure with 7 clusters.

ters group pages within a session according to their categories. The table columns represent the existence or non-existence of a page in a category. Numbers represent the weights or the number of pages, in that particular session, that belong to the category. It is worth noting that each of the most common categories is allocated in a cluster with the rest of the categories spread across the 7 clusters. We derived from this result that the number of clusters k is fully dependent on the nature of the data and the features selected. Therefore, it is highly unrecommended

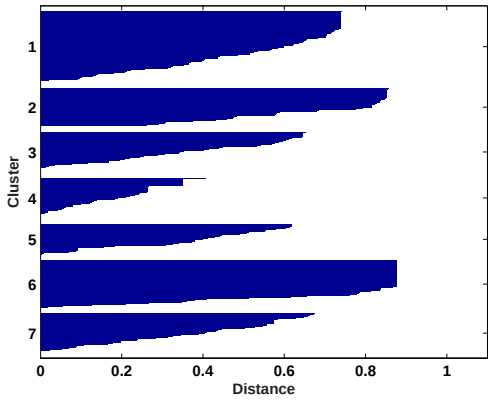

Figure 5.11: Silhouette value of Correlation distance measure with 7 clusters.

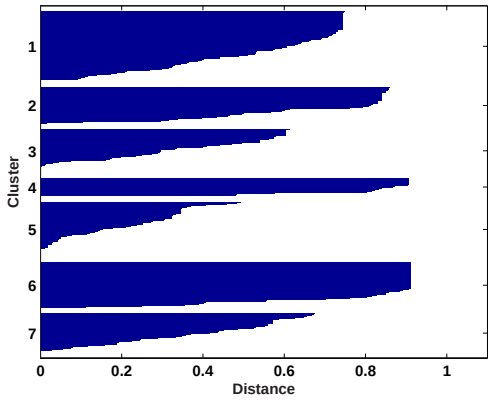

Figure 5.12: Silhouette value of Cosine distance measure with 7 clusters.

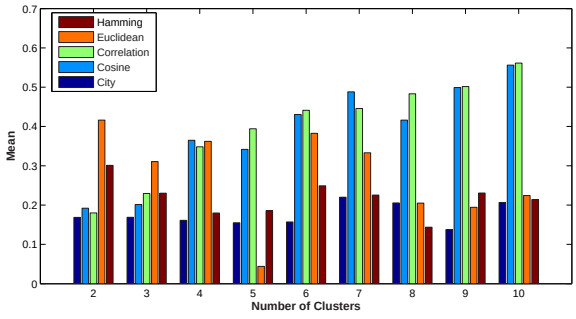

Figure 5.13: The mean value for 2...10 clusters using different distance measures.

to identify *k* before analyzing the data and identifying the purpose of grouping data into clusters.

Next step was to expand the categories back to their original form before applying Markov model techniques. This process is performed using a simple program that seeks and displays the data related to each category. If we consider the categorization example in Section 5.4.1, cie category will be expanded back to cie/metadata.txt.html cie/index.html cie/summer95 and cie/summer95/articles. If a user accesses cie/index.html, there is a chance he/she will access cie/summer95 then cie/summer95/articles next.

Markov model implementation was carried out for all data sets. Each data set was divided into training set and test set and 2-Markov model accuracy was calculated accordingly. Then, using the test set, each transaction was considered as a new point and distance measures were calculated in order to define the cluster that the point belongs to. Next, 2-Markov model prediction accuracy was retrieved as computed in the training phase. Figure 5.14 depicts a flowchart that illustrates the process of calculating prediction accuracy. In the flowchart, Tr and Te represent training data set and test data set respectively; while D stands for the minimum distance measure and i represents an item in the test data set. C stands for cluster and A means the prediction accuracy for the particular item while TA represents the sought after total prediction accuracy for the whole data set. Markov model prediction accuracy results were achieved using the maximum likelihood based

Table 5.12: Web sessions grouped into 7 clusters

	Access	enviro	EPA	hrmd	OSW	Press	Waisicons
Cluster 1	-	7	-	-	-	-	-
Cluster 1	-	5	-	-	-	-	-
Cluster 1	-	21	-	-	-	-	-
Cluster 1	-	3	-	-	-	-	-
Cluster 1	-	13	-	-	-	-	-
Cluster 1	-	1	-	-	-	-	-
Cluster 2	-	-	5	-	-	-	-
Cluster 2	-	-	27	-	-	-	-
Cluster 2	-	-	4	-	-	-	-
Cluster 2	-	-	2	-	-	-	-
Cluster 2	-	-	1	-	-	-	-
Cluster 2	-	-	16	-	-	-	-
Cluster 3	-	-	-	-	-	3	-
Cluster 3	-	-	-	-	-	3	-
Cluster 3	-	-	-	-	-	9	-
Cluster 3	-	-	-	-	-	11	-
Cluster 3	-	-	-	-	-	20	-
Cluster 3	-	-	-	-	-	6	-
Cluster 4	-	-	-	-	4	-	-
Cluster 4	-	-	-	-	4	-	-
Cluster 4	-	-	-	-	4	-	-
Cluster 4	-	-	-	-	9	-	-
Cluster 4	-	-	-	-	2	-	-
Cluster 4	-	-	-	-	4	-	-
Cluster 5	-	-	-	4	-	-	-
Cluster 5	-	-	-	5	-	-	-
Cluster 5	-	-	-	11	-	-	-
Cluster 5	-	-	-	6	-	-	-
Cluster 5	-	-	-	9	-	-	-
Cluster 5	-	-	-	4	-	-	-
Cluster 6	-	-	-	-	-	-	3
Cluster 6	-	-	-	-	-	-	12
Cluster 6	-	-	-	-	-	-	3
Cluster 6	-	-	-	-	-	-	1
Cluster 6	-	-	-	-	-	-	4
Cluster 6	-	-	-	-	-	-	2
Cluster 7	8	-	-	-	-	-	-
Cluster 7	11	-	-	-	-	-	-
Cluster 7	12	-	-	-	-	-	-
Cluster 7	8	-	-	-	-	-	-
Cluster 7	3	-	-	-	-	-	-
Cluster 7	4	-	-	-	-	-	-

on conditional probabilities as stated in Chapter 4. All predictions in the test data that did not exist in the training data sets were assumed incorrect and were given a zero value.

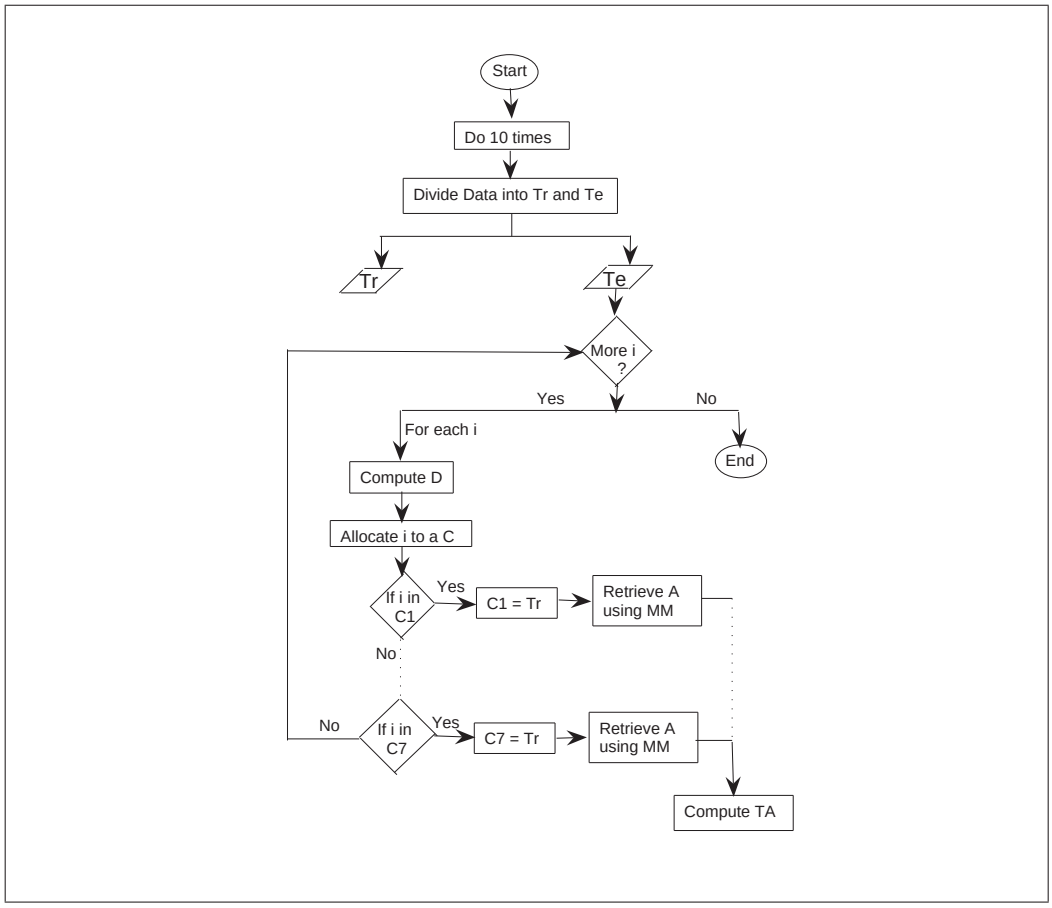

Figure 5.14: Flowchart illustrating prediction accuracy calculation process.

The Markov model accuracy was calculated using a 10-fold cross validation. The data was split into ten equal sets. First, we considered the first nine sets as training data and the last set for test data. Then, the second last set was used for testing and the rest for training. We continued moving the test set upward until the first set was used for testing and the rest for training. The reported accuracy is

the average of ten tests.

5.4.5 Comparing IMC, Clustering and MM Accuracy

Figure 5.15 compares the Markov model accuracy of the whole data set to Markov model accuracy using clusters based on Euclidean, Correlation and Cosine distance measures with $k = 7$ for data set D1.

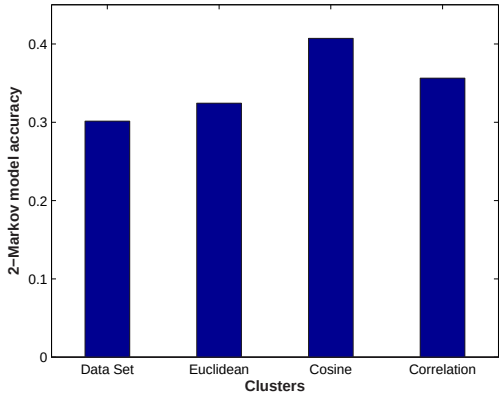

Figure 5.15: Accuracy of clustering, Markov model of whole data set and Markov model accuracy using clusters based on Euclidean, Correlation and Cosine distance measures with $k = 7$ for data set D1.

For comparison reasons, clustering techniques were implemented on each of the data sets and prediction accuracy was calculated based on clustering alone. For this purpose, *k*-means clustering algorithm was implemented on the actual Web sessions, without categorisation, using the squared Euclidean distance measure. Web sessions were represented as a vector with binary figures where the presence of a page is denoted by 1 and the non-presence by zero. The evaluation metric used for clustering the data sets was the standard Mean Absolute Error

(MAE) where for each instance in the test set, we made a prediction for the next page. We calculated the absolute deviation between the actual result and the predicted result. MAE is the sum of all the deviations divided by the number of predictions. Lower MAE values represent higher prediction accuracy. Figure 5.16, Figure 5.17, Figure 5.18 and Figure 5.19 compare the accuracy of clustering with that of PMM and the integration of Markov model and clustering (IMC) for the four data sets using Cosine distance measures for the clusters and based on the 2^{nd} order Markov model. The figures demonstrate a decrease in prediction accuracy using clustering alone. This is due in part to the distance measure used and also to non-categorisation of Web sessions. The figures also reveal the improvement in IMC precision results over PMM and clustering. Data sets D3 and D4 show more significant accuracy increase between clustering and Markov model based prediction than data sets D1 and D4. Data sets D1 and D4 reveal more conformity in accuracy increase from clustering to PMM, then IMC.

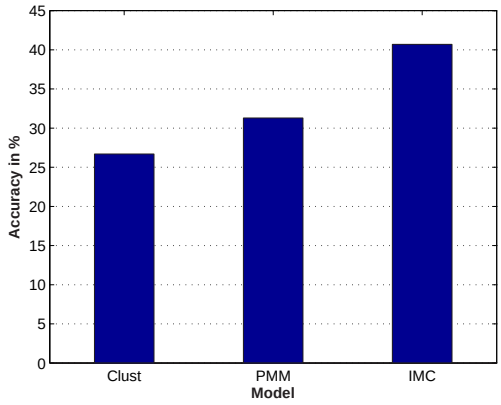

Figure 5.16: Accuracy of clustering, PMM and IMC for data set D1.

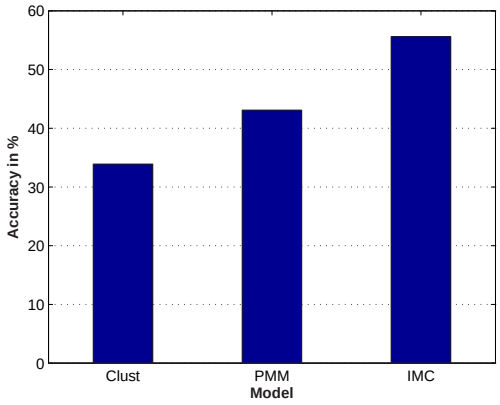

Figure 5.17: Accuracy of clustering, PMM and IMC for data set D2.

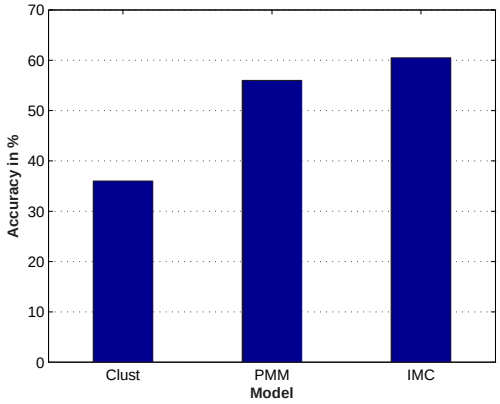

Figure 5.18: Accuracy of clustering, PMM and IMC for data set D3.

5.4.6 Comparing IMC To a Higher Order Markov Model

5.4.6.1 Comparing State Space Complexity

Section 5.4.5 experiments prove that the IMC integration model improves the accuracy of the lower order Markov model. In this section, we experiment further to prove that the IMC integration model improves the state space complexity of a higher order Markov model. Table 5.13 compares IMC state space complexity to

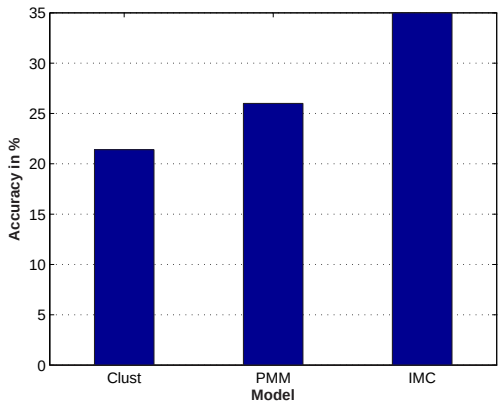

Figure 5.19: Accuracy of clustering, PMM and IMC for data set D4.

that of the frequency pruned 3^{rd}-order Markov model.

Table 5.13: IMC number of states

	D1	D2	D3	D4
3-PMM	14,977	18,121	11,218	19,032
IMC	11,682	10,388	19,035	13,634
3-MM	72,524	89,815	50,971	90,123

Table 5.13 reveals that, for some data sets, IMC involves more states than a frequency pruned higher order Markov model. However, IMC improves the state space complexity of a higher order Markov model, 3^{rd}-order Markov model.

5.4.6.2 Comparing Accuracy

acknowledging the fact that IMC improves the prediction accuracy of a lower order Markov model draws our attention to whether or not IMC provides better accuracy than a higher order Markov model. Figure 5.20 reveals the prediction accuracy of IMC as opposed to frequency pruned 3^{rd}-order Markov model.

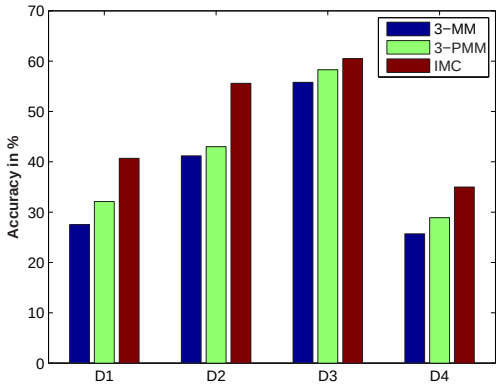

Figure 5.20: Accuracy of 3^{rd} order Markov model (3-MM), frequency pruned all 3^{rd} order Markov model (3-PMM) and IMC model for all four data sets.

Figure 5.20 shows more improvement in prediction accuracy using data set D2 and less improvement using data set D3. However, all data sets IMC prediction accuracies were above those of 3-PMM.

5.4.7 IMC Complexity

The running time of k-means clustering algorithm increases with the increase of the number of clusters regardless of the distance measure used. Figure 5.20 reveals increased time complexity with increased number of clusters for all four data sets.

Figure 5.21 depicts the prediction time of IMC model for all four data sets. For data set D1, the number of clusters used $k = 7$, for data set D2 $k = 9$, for data set D3 $k = 14$ and for data set D4 $k = 10$.

Figure 5.21 displays the increase of prediction time as the number of clusters increase. This is explained by the time required to calculate the distance, using

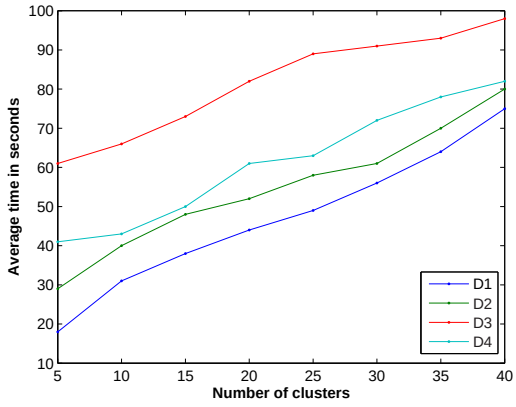

Figure 5.21: Running time of clusters for all four data sets.

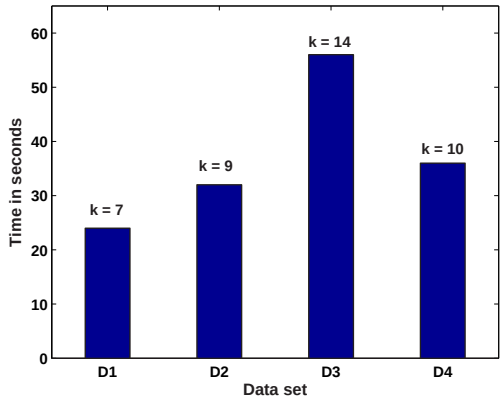

Figure 5.22: Prediction time of IMC model for all four data sets.

Cosine distance measure, of the new item to the mean of every cluster and finding the closest cluster. More clusters require more calculations regardless of the size of the cluster. According to these results, the IMC prediction time is still affordable when dealing with large data sets.

5.5 Conclusion

This chapter has presented our improvement of Markov model accuracy by grouping Web sessions into clusters. The Web pages in the user sessions are first allocated into categories according to Web services that are functionally meaningful. Then, *k*-means clustering algorithm is implemented using the most appropriate number of clusters and distance measure. Markov model techniques are applied to each cluster as well as to the whole data sets. The experimental results reveal that implementing the *k*-means clustering algorithm on the data sets improves the accuracy of a lower order Markov model while reducing the state space complexity of a higher order Markov model. The prediction accuracy achieved is an improvement to previous research papers that addressed mainly recall and coverage.

Chapter 6

Integrating Markov Model with Association Rules and Clustering

6.1 Introduction

Combining Markov model with Association rules in Chapter 4 has been shown to improve the next page prediction accuracy, and combining Markov model with clustering techniques has been proved to improve the prediction accuracy to a greater extent. Therefore, this Chapter integrates all three techniques together applying certain constraints in order to achieve even better Web page access prediction accuracy.

Since Markov model, association rules and clustering techniques have been introduced and examined in the previous chapters, this chapter focuses mainly on their integration. Section 2 of this chapter explains the new integration model and the integration algorithm giving details of each stage of the algorithm. Section 3 proves the new integration model increase in accuracy using different experiments. Section 4 concludes this chapter.

6.2 Integration Process

This chapter discusses combining clustering algorithm, association rule mining and Markov model during the prediction process.

6.2.1 Motivation For Integration

Several researchers, including our work in Chapter 5, attempted to improve the Web page access prediction precision or coverage by combining clustering with association rules Lai & Yang (2000), Liu et al. (2001). Lai & Yang (2000) have introduced a customized marketing on the Web approach using a combination of clustering and association rules. The authors collected information about customers using forms, Web server log files and cookies. They categorized customers according to the information collected. Since *k*-means clustering algorithm works only with numerical data, the authors used PAM (Partitioning Around Medoids) algorithm to cluster data using categorical scales. They then performed association rule techniques on each cluster. They proved through experimentations that implementing association rules on clusters achieves better results than on non-clustered data for customizing the customers' marketing preferences. Liu et al. (2001) have introduced MARC (Mining Association Rules using Clustering) that helps reduce the I/O overhead associated with large databases by making only one pass over the database when learning association rules. The authors group similar transactions together and they mine association rules on the summaries of clusters

instead of the whole data set. Although the authors prove through experimentation that MARC can learn association rules more efficiently, their algorithm does not improve on the accuracy of the association rules learned.

Combining association rules with Markov model is novel to our knowledge and only few of past researches combined all three models together Kim et al. (2004). Kim et al. (2004) improve the performance of Markov model, sequential association rules, association rules and clustering by combining all these models together. For instance, Markov model is used first. If MM cannot cover an active session or a state, sequential association rules are used. If sequential association rules cannot cover the state, association rules are used. If association rules cannot cover the state, clustering algorithm is applied. Kim et al. (2004) work improved recall and it did not improve the Web page prediction accuracy.

The Integrated Prediction Model (IPM) integration is novel and proves to outperform each individual prediction model as well as the different combination models addressed above. The IPM integration model improves the prediction accuracy as opposed to other combinations that prove to improve the prediction coverage and complexity. The improvement in accuracy is based on different constraints like dividing the data set into a number of clusters based on services requested by users. This page categorization method proves to yield better clustering results Wang et al. (2004). Therefore, better clusters means better Markov model prediction accuracy because the Markov model prediction will be based

on more meaningfully grouped data. It also improves the state space complexity because Markov model prediction will be carried out on one particular cluster as opposed to the whole data set. The other constraint is using association rule mining in the case of a state absence in the training data or where the state prediction probability is not marginal. This helps improve the prediction accuracy because association rules look at more history and examine more states than Markov models. Also, IPM will not be subject to the complexity associated with the number of rules generated because the rules will be examined in special cases only. Another constraint is the distance measure used in the identification of the appropriate cluster that each new page should belong to. The cosine distance measure has proved to outperform other distance measures like Euclidean, hamming, correlation and city block Strehl et al. (2000), Halkidi et al. (2003). The prediction accuracy based on the integration of the three frameworks together according to these constraints proves to outperform the prediction accuracy based on each of the frameworks individually. Figure 6.1 below depicts the architecture of the integration model (IPM).

6.2.2 IPM Algorithm

IPM involves combining the three Web usage mining prediction models clustering, Markov model and association rules together. It first clusters Web sessions according to meaningful features selection techniques using *k*-means clustering algorithm and Cosine distance measure. Each data set is grouped into different

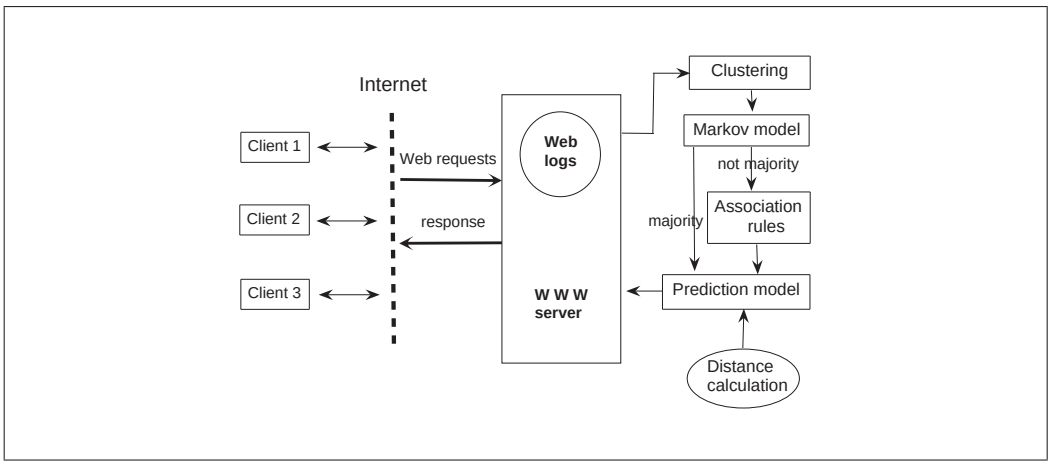

Figure 6.1: IPM model architecture.

number of clusters according to the number of clusters discussion in Chapter 5. The integration model then computes Markov model prediction on the resulting clusters. Association rules are only examined in the case where the prediction results are based on states that do not belong to the majority class.

6.2.2.1 Algorithm Training process

The training process occurs offline. It is usually more complex than the prediction process, as discussed in Chapter 3. It involves preparing the data and creating the models used for prediction. The IPM training process is as follows:

```
Training:

(1)Combine functionally related pages according to services
   requested

(2)Cluster user sessions into l-clusters

(3)Build a k-Markov model for each cluster

(4)FOR Markov model states where the majority is not clear
```

```
(5)  Collect all sessions satisfying the state
(6)  Construct association rules to resolve ambiguity
(7)  Store the association rules with the state
(8)ENDFOR
```

Combining similar pages or allocating related pages to categories is the first step in the training process of the IPM model. The categorisation of user sessions is implemented according to the feature selection process presented in Chapter 5, Section 5.4. Clustering of Web sessions is performed next according to the *k*-means algorithm using Cosine distance measure and certain number of clusters. For more details, refer to Chapter 5 .

Markov model analysis were carried out on each cluster using frequency pruned *k*-order Markov model as explained in Chapter 4.

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. 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 *k*-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 & 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:

$$\phi_c = \hat{p} - z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{6.1}$$

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 6.1 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 for the item is calculated and stored with the state. Table 6.1 displays the results of the IPM 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 6.1 also reveals the number of states that are retained for association rule implementation.

Table 6.1: Accuracy according to $z_{\alpha/2}$ value

$z_{\alpha/2}$	Accuracy	# 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.

6.2.2.2 Algorithm Prediction Process

The prediction or test phase takes place online. The IPM prediction process is as follows:

```
Prediction:
(1)For each coming session
(2)  Find its closest cluster
(3)  Use corresponding Markov model to make prediction
(4)  If the predictions are made by states that do not belong to a
     majority class
(5)    Use association rules to make a revised prediction
(6)  EndIf
(7)EndFor
```

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. Finding the closest cluster to the new session is carried out as explained in Chapter 5. 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, global association rules are used for prediction according to the following:

The probability of accessing the next page p_n is first calculated using 2^{nd} order Markov model as follows:

$$P_{l+1} = \text{argmax}_{p \in \mathbb{P}}\{Porb(P_{l+1} = p | p_l, p_{l-1}, \ldots, p_{l-(k-1)})\} \qquad (6.2)$$

Constructing the 2^{nd} order Markov model, results in one of two cases:

$$prediction for p_n = \begin{cases} 0 & if P(p_n|p_c,p_p) = 0 \ and \\ & P(p_n|p_c) = 0 \\ \neq 0 & if P(p_n|p_c,p_p) \neq 0 \ or \\ & if P(p_n|p_c,p_p) = 0 \ and \\ & P(p_n|p_c) \neq 0 \end{cases}$$

where p_p is the page accessed immediately before p_c by the same user in the same Web session W. Markov model prediction accuracy is retrieved in the following cases:

$$\begin{cases} P(p_n) \neq 0 & and \\ |P(p_n) - P(p_c)| > \phi_c \end{cases}$$

On the other hand, association rule prediction accuracy is retrieved in the following cases:

$$\begin{cases} P(p_n) = 0 & or \\ |P(p_n) - P(p_c)| < \phi_c \end{cases}$$

Again, the prediction process is more time complex because of the procedure that finds the closest cluster and it occurs online. However, predictions using Markov model or association rules are a mere finding, using a look up table, of the accuracy result that was determined and stored during the training phase.

6.2.3 Example

Consider table 6.2 that depicts data transactions performed by a user browsing a Web site.

Table 6.2: User sessions.

T1	A,F,I,J,E,C,D,H,N,I,J,G,D,H,N,C,I,J,G
T2	F,D,H,N,I,J,E,A,C,D,H,N,I,J,G
T3	E,C,A,C,F,I,A,C,G,A,D,H,M,G,J
T4	F,D,H,I,J,E,H,F,I,J,E,D,H,M
T5	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 Table 6.3 and Table 6.4 below.

Table 6.3: First cluster.

T1	A,F,I,J,E,C,D,H,N,I,J,G,D,H,N,C,I,J,G
T2	F,D,H,N,I,J,E,A,C,D,H,N,I,J,G
T4	F,D,H,I,J,E,H,F,I,J,E,D,H,M

Table 6.4: Second cluster.

T3	E,C,A,C,F,I,A,C,G,A,D,H,M,G,J
T5	G,E,A,C,F,D,H,M,I,C,A,C,G

Consider the following test data state I $\rightarrow$ J $\rightarrow$?. Applying the 2^{nd} 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:

$$P_{l+1} = \text{argmax}\{P(E|J,I)\} = \text{argmax}\{E \rightarrow 0.57\}$$

$$P_{l+1} = \text{argmax}\{P(G|J,I)\} = \text{argmax}\{G \rightarrow 0.43\}$$

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 6.5 below.

Table 6.5: 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

Table 6.6 and Table 6.7 summarise the results of applying subsequence association rules to the training data. Table 6.6 shows that F $\rightarrow$ E has the highest confidence of 100%, while Table 6.7 shows that C$\rightarrow$ G has the highest confidence of 100%. The confidence is calculated according to the following equation that was explained in Section 3.3 of Chapter 3:

$$\alpha = conf(A) = \frac{supp(\langle A,P \rangle)}{supp(A)} \tag{6.3}$$

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

Table 6.6: 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 6.7: 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%

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.

6.2.4 IPM Algorithm Efficiency Analysis

The running time of association rule mining is dependent on the complexity of the Apriori algorithm complexity which results in O(I.D) as explained in Chapter 4, Section 4.3.2. During the training stage, clustering and Markov model are implemented for the whole data sets. However, association rules are only implemented in special cases. This reduces the complexity of association rule mining.

Although, during prediction, allocating the new item to the appropriate cluster adds complexity to the IPM model, the overall IPM model prediction complexity is reduced due to the fact that the prediction process involves retrieving Markov models of one cluster as opposed to the whole data set. Also, association rules are only retrieved in the case where the state does not belong to the majority class. This gives the conclusion that the complexity of IPM depends on the size of the majority class. Larger majority class yields less complex prediction as it involves less association rule accesses. However, larger majority class does not leave a larger room for accuracy improvement.

6.3 Experimental Evaluation

In this section, we present experimental results to evaluate the performance of our algorithm. For our experiments, the four data sets mentioned in Chapter 4 were relied upon and all preprocessing tasks of Chapter 5 including session identification and categorisation were also used.

6.3.1 Clustering, Markov Model and Association Rules

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.

Markov model implementation was carried out for the original data in each cluster. 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, *k*-Markov model prediction accuracy was determined by using the Markov model accuracy of that cluster.

Since association rule 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. The association rule analysis performed in Chapter 4 was considered to determine 4% as the support threshold and 90% as the confidence factor.

6.3.2 Experiments Results

Figure 6.2, Figure 6.3, Figure 6.4 and Figure 6.5 depict better Web page access prediction accuracy for all four data sets by integrating Markov model, association

rules and clustering (IPM) than by employing the clustering, Markov model and association rules individually. Prediction accuracy was computed as follows:

1. The data set is clustered according to *k*-means clustering algorithm and Cosine distance measure.

2. For each new instance, the prediction accuracy is calculated based on Markov model prediction performed on the closest cluster.

3. The frequency of the item is also determined in that particular cluster and ϕ_c is calculated for the new instance using $z_{\alpha/2}$ value to determine if it belongs to the majority class.

4. If the prediction results in a state that does not belong to the majority class, association rules are used for prediction, otherwise, Markov model accuracy is employed.

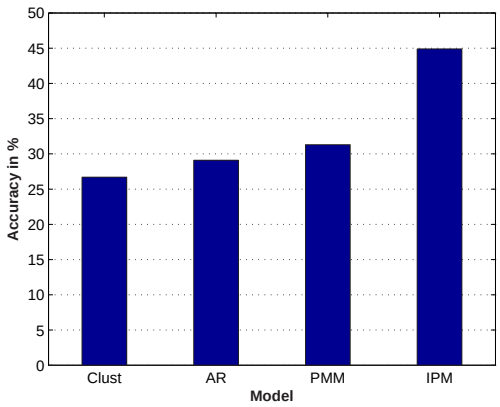

Figure 6.2: Accuracy of Clustering, AR, PMM, and IPM for data set D1.

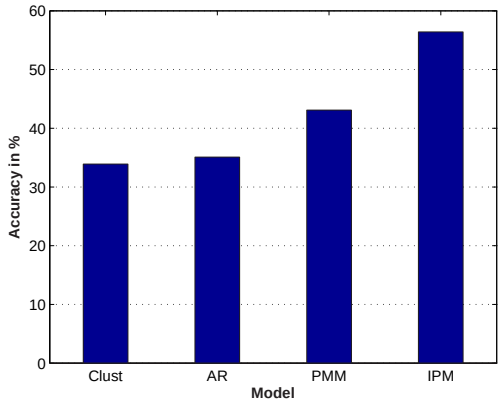

Figure 6.3: Accuracy of Clustering, AR, PMM, and IPM for data set D2.

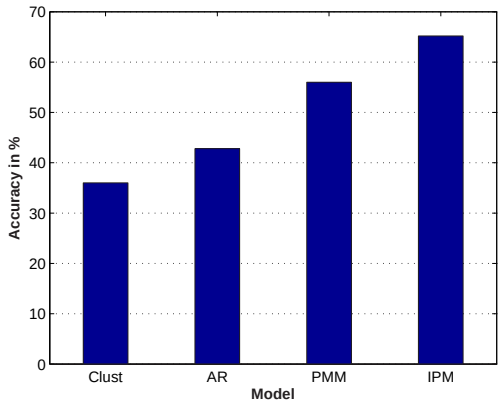

Figure 6.4: Accuracy of Clustering, AR, PMM, and IPM for data set D3.

The above Figures display that IPM 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 IPM over the individual models. On the other hand, D2 and D3 display a more consistent improvement in prediction accuracy.

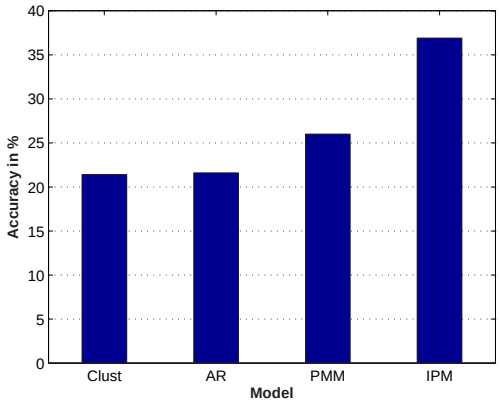

Figure 6.5: Accuracy of Clustering, AR, PMM, and IPM for data set D4.

Prediction accuracy results were achieved using the maximum likelihood based on conditional probabilities. 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.

6.3.3 Comparing All Models Accuracy Results

In comparing the IPM model results to the other combination results of Markov model and association rules (IMAM) in Chapter 4 and Markov model and clustering (IMC) in Chapter 5 as well as the individual models, we find that clustering techniques render the lowest Web page prediction accuracy. This is evident in Figure 6.6, Figure 6.7, Figure 6.8 and Figure 6.9 below. Although, association rule mining prove to achieve better prediction accuracy results than clustering, the

pruned all-2^{nd} order Markov model gives better results than association rules and this is apparent in Figure 6.6-6.9. 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 IPM than using IMC. It is important to note though that Figure 6.7 representing data set D2, displayed a more significant improvement of prediction accuracy using IMC. also, Figure 6.8 representing data set D3 depicted a more significant improvement of prediction accuracy using IPM. Data set D1 demonstrated an overall consistent improvement of prediction accuracy using IMAM, then IMC, then IPM respectively. On the other hand, the more significant improvement of prediction accuracy using IMAM over IMC and IPM was apparent in Figure 6.9, using data set D4. This is further manifested in Figure 6.10 below.

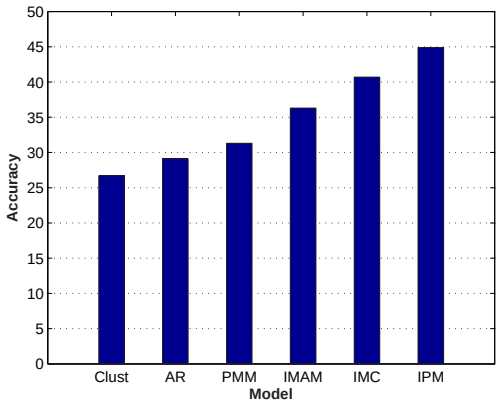

Figure 6.6: Accuracy of Clustering, AR, PMM, IMAM, IMC and IPM for data set D1.

The actual figures of the accuracy results of all models for all four data sets are represented by Table 6.8 below:

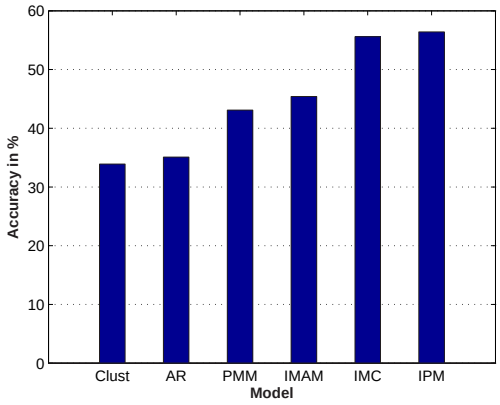

Figure 6.7: Accuracy of Clustering, AR, PMM, IMAM, IMC and IPM for data set D2.

Table 6.8: Prediction accuracy using all models for all four data sets.

	D1	D2	D3	D4
Clust	26.7	33.9	36.0	21.4
AR	29.1	35.1	42.8	21.6
MM	30.5	42.6	54.9	25.7
PMM	31.3	43.1	56.0	26.0
IMAM	36.3	45.4	59.2	33.6
IMC	40.7	55.6	60.5	35.0
IPM	44.9	56.4	65.2	36.9

Figure 6.10 combines all accuracies for all models and all four data sets together.

6.3.4 Comparing Results to a Higher Order Markov Model

6.3.4.1 State Space Complexity Comparison

Despite the efficient prediction accuracy results that were achieved using the three different integration models, it was necessary to perform state space complexity analysis for the three models. The state space complexity analysis performed for

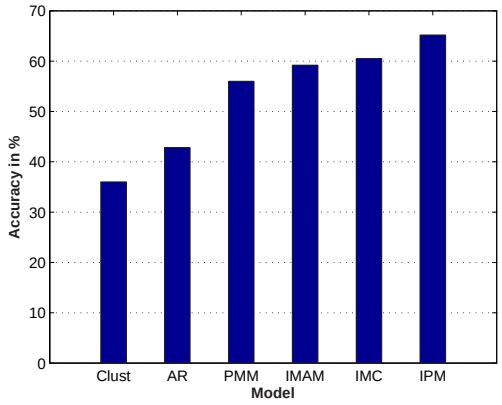

Figure 6.8: Accuracy of Clustering, AR, PMM, IMAM, IMC and IPM for data set D3.

IMAM model states included the summation of both Markov model and association rule states where applicable. Also, the IMC model states included both Markov model and clustering states. Whereas, the IPM 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 (3^{rd}) 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 6.9 below.

Table 6.9: Number of states for 3-PMM, IMAM, IMC and IPM and 3-MM using D1, D2, D3 and D4.

	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
IPM	13,388	11,511	20,020	15,116
3-MM	72,524	89,815	50,971	90,123

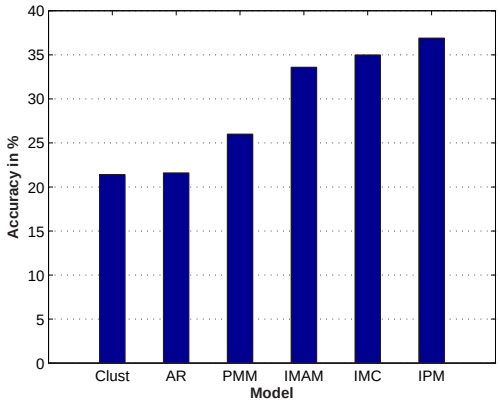

Figure 6.9: Accuracy of Clustering, AR, PMM, IMAM, IMC and IPM for data set D4.

Looking at Table 6.9, 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 3^{rd}-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 as it is shown in Table 4.10, Chapter 4. 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 IPM models that implement clustering techniques asserts our findings. It is vindicated though that the number of states of the three integration models are significantly decreased when compared to 3^{rd}-order Markov model.

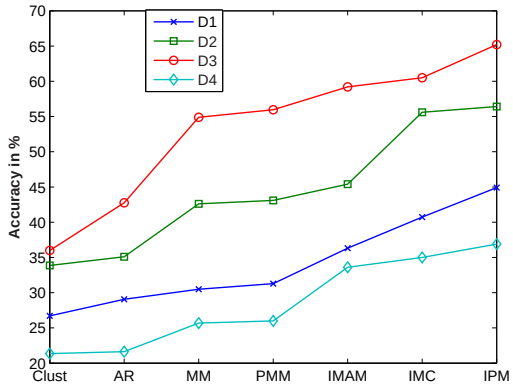

Figure 6.10: Accuracy of Clustering, AR, PMM, IMAM, IMC and IPM for all four data sets.

6.3.4.2 Accuracy Comparison

After verifying the increase of prediction accuracy using the IMAM, IMC and IPM 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 3^{rd}-order Markov model (3-MM) and frequency pruned 3^{rd}-order Markov model (3-PMM). Figure 6.11 depicts that our integration models deliver better prediction accuracy than a higher order Markov model.

6.4 Conclusion

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, IPM, integrates the three models using lower

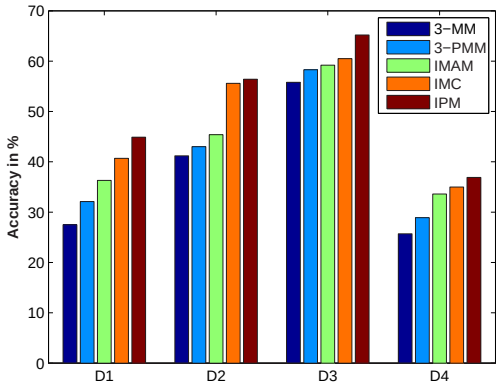

Figure 6.11: Accuracy of 3-MM and 3-PMM compared to that of IMAM, IMC and IPM for all four data sets.

order Markov model computed on clusters achieved using *k*-means clustering algorithm and Cosine distance measures for states that belong to the majority class and performing association rule mining on the rest. User sessions are first clustered using some meaningful measures. Then Markov models are implemented using the outcome of the clustering effectuation. Association rules are used for prediction only in the case of certain stipulations. IPM proves to outperform all three models implemented individually, as well as, the IMAM and IMC integrated models when it comes to accuracy. Also, IPM improves the state space complexity of a higher order Markov model.

Chapter 7

Conclusions

7.1 General Discussions

The main objective of the dissertation is to help achieve better prediction accuracy for Web page access. Recommending a next page the Web user will access is very important for various Web applications. Web page access prediction is addressed by many literature publications. The main technology implemented for this purpose is through using Web usage mining pattern discovery techniques. In this dissertation, we examined the three most important and vital Web usage mining pattern discovery techniques used for this purpose. We first discussed major issues related to each of the pattern discovery techniques in general. We then identified their limitations and integrated them differently in a way where those limitations are addressed properly and kept to a minimum. Through the pattern discovery models integration, we exhausted their varied positive impact on Web page prediction accuracy. By keeping the models limitations to a minimum and relying on their advantages, and, by integrating the different models according to

different constraints, we were able to achieve more accurate prediction results.

7.2 Conclusion of Results

In this dissertation we have improved the Web page access prediction accuracy by combining different Web usage mining techniques. First we have examined the individual Web usage mining techniques individually and demonstrated experimentally the fact that Web access prediction accuracy increases in this order of using Web usage mining techniques: clustering, association rules, Markov model, the combination of association rules and Markov model, the combination of clustering and Markov model and the combination of association rules, clustering and Markov model. As an end result, through our integration of Markov model and association rule mining and of Markov model and clustering and of all three models together, association rules, Markov model and clustering, we have proved to increase the Web access prediction accuracy significantly.

The extra advantage of our models is the low state space complexity. Combining a lower order Markov model with association rules, and with clustering, and finally, with both association rules and clustering has benefited from the low order Markov model low state space complexity. All integration models implemented proved to generated less states than a higher order Markov model.

7.3 Strengths of Findings

The best measure to employ in order to evaluate our improvement of Web access prediction accuracy is standard deviation. Through standard deviation, we are able to detect the difference between the mean accuracy value and the average of individual accuracy values. In other words, we are able to express how far off the accuracy value is from the mean. To prove that our combination of Web usage mining models increases the Web access prediction accuracy, we calculated the standard deviation of accuracy values for 2-PMM, IMAM, IMC and IPM using all four data sets. To better compare the standard deviation to the mean accuracy values, Table 7.3 lists all standard deviations alongside the mean accuracy values. The low standard deviation figures give more weight and significance to the improved prediction accuracy displayed in Figure 7.1, Figure 7.2, Figure 7.3 and Figure 7.4 above.

The standard deviation results disclose that all the standard deviation results are considerably low compared to the mean values. This means that 2-PMM, IMAM, IMC and IPM accuracy results are quite different from each other lying on an improved baseline.

Table 7.1: Accuracy values standard deviation

	D1	D2	D3	D4
PMM	31.3 ± 4.69	43.1 ± 3.90	56.0 ± 2.71	26.0 ± 1.36
IMAM	36.3 ± 3.07	45.4 ± 1.98	59.2 ± 5.32	33.6 ± 2.17
IMC	40.7 ± 2.55	55.6 ± 2.94	60.5 ± 1.45	35.0 ± 3.83
IPM	44.9 ± 1.32	56.4 ± 3.07	65.2 ± 6.19	36.9 ± 2.69

7.4 Limitations and Future Directions

The IPM model could be extended to a completely ”hands-off” or automated system. Currently, some human intervention is required especially during the features selection process.

In this dissertation, only prediction accuracy and state space complexity were accounted for. Generally speaking, there exists a number of other issues that can affect the usefulness of a prediction system such as trust in the system, transparency of the algorithm used, and the diversity of predictions and recommendations. Therefore, the evaluation of a prediction system needs to be carried out along a number of dimensions in addition to accuracy and efficiency. Some of the dimensions that are worth examining in the future include utility (or usefulness of the system), explainability, robustness, scalability and user satisfaction.

Although the results achieved in this study were satisfactory, there is still room for more extensive experimentations using, for example, model-based clustering algorithms, an association rules algorithm other than the Apriori algorithm and

different order Markov models. Also, since the main objective of this dissertation is to improve the accuracy of predicting the next page to be accessed by the Web user, it would be interesting to learn if such increase in accuracy is coupled with reduced latency. This would form a great extension to this work.

References

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

Agrawal, R., Imielinski, T. & Swami, A. (1993), 'Mining association rules between sets of items in large databases', *ACM SIGMOD Conference on Management of data* pp. 207–216.

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

Agrawal, R. & Srikant, R. (1996), 'Mining sequential patterns', *International Conference on Data Engineering (ICDE), Taiwan* .

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

Ball, G. H. & Hall, D. J. (1965), 'Isodata, a novel method of data analysis and classification', *Tech. Rep., Stanford University, Stanford, CA.* .

Banerjee, A. & Ghosh, J. (2001), 'Clickstream clustering using weighted longest common subsequences', *SIAM Conference on Data Mining, Chicago* pp. 33–40.

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

Berti, L. (2007), 'Data quality awareness: a case study for cost optimal association rule mining', *Knowledge and Information Systems* **11**(2), 191–215.

Bhowmick, S. S., S. K. Madria, W. K. N. & Lim, E. P. (1998), 'Web bags: Are they useful n web warehouse?', *FDO'98* pp. 59–68.

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

Cadez, I., Heckerman, D., Meek, C., Smyth, P. & White, S. (2000), 'Visualization of navigation patterns on a web site using model based clustering', *ACM SIGMOD Int'l Conf on Knowledge Discover and Data Mining* pp. 280–284.

Cadez, I., Heckerman, D., Meek, C., Smyth, P. & White, S. (2003), 'Model-based clustering and visualization of navigation patterns on a web site', *Data Mining and Knowledge Discovery* **7**(4), 399–424.

Casale, G. (2005), 'Combining queueing networks and web usage mining tech-

niques for web performance analysis', *ACM Symposium on Applied Computing* pp. 1699–1703.

Chakrabarti, D., Kumar, R. & Tomkins, A. (2006), 'Evolutionary clustering', *KDD'06, USA* pp. 554–560.

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

Chen, X., Kim, D., Nnadi, N., Shah, H., Shrivastava, P., Bieber, M., Im, I. & Wu, Y. (2003), 'Digital library service integration', *JCDL'03, Texas* pp. 384–396.

Cheng, D., Kannan, R., Vempala, S. & Wang, G. (2005), 'A divide-and-merge methodology for clustering', *ACM SIGMOD/PODS* pp. 196–212.

Cooley, R., Mobasher, B. & Srivastava, J. (1997), 'Web mining: Information and pattern discovery on the world wide web', *9th IEEE International Conference on Tools with Artificial Intelligence* pp. 558–567.

Cooley, R., Mobasher, B. & Srivastava, J. (1999), 'Data preparation for mining world wide web browsing patterns', *Knowledge and Information Systems* **1**(1), 5–32.

Deshpande, M. & Karypis, G. (2001), 'Item-based top-*N* recommendation algorithms', *ACM Transactions on Information Systems* **22**(1), 1–34.

Deshpande, M. & Karypis, G. (2004), 'Selective markov models for predicting web page accesses', *Transactions on Internet Technology* **4**(2), 163–184.

Dongshan, X. & Junyi, S. (2002), 'A new markov model for web access prediction', *Computing and Science Engineering* **4**(6), 34–39.

Duda, R., Hart, P. & Stork, D. (2000), 'Pattern classification', *John Wiley and Sons* **2**.

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

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

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

Ferragina, P. & Gulli, A. (2005), 'A personalized search engine based on web-snippet hierarchical clustering', *WWW'05, Japan* pp. 801–810.

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

Geraci, F., Pellegrini, M., Pisati, P. & Sebastiani, F. (2006), 'A scalable algorithm for high-quality clustering of web snippets', *SAC'06, France* pp. 1058–1062.

Gunduz, S. & 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. & Vazirgiannis, M. (2003), 'Thesus: Organizing web document collections based on link semantics', *The VLDB Journal* **2003**(12), 320–332.

Han, J. L. & Plank, A. W. (1996), 'Background for association rules and cost estimate of selected mining algorithms', *CIKM'96* pp. 73–80.

Heer, J. & Chi, H. (2002), 'Separating the swarm: Categorization methods for user sessions on the web', *Minneapolis, Minnesota* **4**(1), 243–250.

Jain, A. K., Murty, M. N. & Flynn, P. J. (1999), 'Data clustering: A review', *ACM Computing Surveys* **31**(3), 264–323.

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

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

Lai, H. & 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* pp. 37–46.

Liu, B., Hsu, W. & Ma, Y. (1999), 'Mining association rules with multiple minimum support', *KDD, San Diego* pp. 337–341.

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

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

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

Meneses, E. & Rodriguez-Rojas, O. (2006), 'Using symbolic objects to cluster web documents', *WWW'06, Scotland* pp. 967–968.

Mobasher, B., Cooley, R. & Srivastave, J. (1999), 'Creating adaptive web sites through usage-based clustering of urls', *KDEX'99* pp. 143–153.

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

Mobasher, B., Dai, H., Luo, T. & Nakagawa, M. (2001), 'Effective personalization based on association rule discovery from web usage data', *WIDM'01, USA* pp. 9–15.

Mobasher, B., Dai, H., Luo, T. & Nakagawa, M. (2002), 'Using sequential and non-sequential patterns for predictive web usage mining tasks', *IEEE on Data Mining, Japan* pp. 669–672.

Papadakis, N. K. & 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* **17**(12), 1638–1652.

Park, J. S., Philip, S. Y. & Chen, M. S. (1997), 'Mining association rules with adjustable accuracy', *CIKM'97* pp. 151–160.

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

Pons, A. P. (2006), 'Object prefetching using semantic links', *The DATA BASE for Advances in Information Systems* **37**(1), 97–109.

Rigou, M., Sirmakesses, S. & 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. & Riedl, J. (2001), 'Itembased collaborative filtering recommendation algorithms', *10th International WWW Conference, Hong kong* pp. 285–295.

Schwarzkopf, E. (2001), 'An adaptive web site for the um2001 conference', *UM2001, Workshop on Machine Learning for User Modeling* pp. 77–86.

Siersdorfer, S. & Sizov, S. (2004), 'Restrictive clustering and metaclustering for self-organizing document collections', *SIGIR'04, UK* pp. 226–233.

Spertus, E. (1997), ‘Parasite: Mining structural information on the web’, *Sixth International World Wide Web* pp. 176–189.

Spiliopoulou, M., Faulstich, L. C. & 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. & Tan, P. (2000), ‘Web usage mining: Discovery and applications of usage patterns from web data.’, *SIGDD Explorations* **1**(2), 12–23.

Strehl, A., Ghosh, J. & Mooney, R. J. (2000), ‘Impact of similarity measures on web-page clustering’, *AI for Web Search* pp. 58–64.

Su, Z., Yang, Q. & Zhang, H. (2000), ‘A prediction system for multimedia prefetching in internet’, *ACM Multimedia Conference* pp. 96–103.

Sun, J., Wang, X., Shen, D., Zeng, H. & Chen, Z. (2006), ‘CWS: A comparative web search system’, *WWW’06, Scotland* pp. 467–476.

Suryavanshi, B. S., Shiri, N. & Mudur, S. P. (2005), ‘Improving the effectiveness of model based recommender systems for highly sparse and noisy web usage data’, *IEEE/WIC/ACM International Conference on Web Intelligence (WI’05), France* pp. 618–621.

Tsochantaridis, I., Hofmann, T., Joachims, T. & Altun, Y. (2004), ‘Support vector

machine learning for interdependent and structured output spaces', *ICML'04* pp. 823–830.

Wang, K. & Liu, H. (1997), 'Schema discovery for semi-structured data', *KDD'97* pp. 271–274.

Wang, K. & Liu, H. (1998), 'Discovering typical structures of documents: A road map approach', *SIGR'98* pp. 146–154.

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

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

Yan, T. W., Jacobsen, M., Garcia-Molina, H. & Dayal, U. (1996), 'From user access patterns to dynamic hypertext linking', *Fifth International WWW Conference, France* pp. 1007–1014.

Yang, Q., Huang, J. Z. & NG, M. (2003), 'A data cube model for prediction-based web prefetching', *Journal of Intelligent information Systems* **20**(1), 11–30.

Yang, Q., Li, T. & Wang, K. (2004), 'Building association-rule based sequential classifiers for web-document prediction', *Journal of Data Mining and Knowledge Discovery* **8**(3), 253–273.

Yong, W., Zhanhuai, L. & Yang, Z. (2005), 'Mining sequential association-rule for improving web document prediction', *ICCIMA'05* pp. 146–151.

Zaiane, O. R. & Han, J. (2000), 'Webml: Querying the world wide web for resources and knowledge', *WIDM'00* pp. 9–12.

Zaiane, O. R., Xi, M. & Han, J. (1998), 'Discovering web access patterns and trends by applying olap and data mining technology on web logs', *Advances in Digital Libraries* pp. 19–29.

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

Zhao, Y. & Karypis, G. (2002), 'Evaluation of hierarchical clustering algorithms for document datasets', *CIKM'02, USA* pp. 515–524.

Zhao, Y., Zhang, H., Figueiredo, F., Cao, L. & Zhang, C. (2007), 'Mining for combined association rules on multiple datasets', *ACM SIGKDD'02, USA* pp. 18–23.

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

Zhou, D., Bolelli, L., Li, J., Giles, L. & Zha, H. (2007), 'Learning user clicks in web search', *IJCAI 2007* pp. 176–181.

Zhu, J., Hong, J. & Hughes, J. G. (2002*a*), 'Using markov chains for link prediction in adaptive web sites', *Software 2002: Computing in an Imperfect World* pp. 60–73.

Zhu, J., Hong, J. & Hughes, J. G. (2002*b*), 'Using markov models for web site link prediction', *HT'02, USA* pp. 169–170.

Zuckerman, I., Albrecht, D. & Nicholson, A. (1999), 'Predicting users' request on the www', *International Conference on User Modeling (UM99)* pp. 275–284.