Web Profiling and Navigation Path Analysis

(Student Name: Lam Ming Yan
Student No.
Programme Code: BSCCS
Supervisor: Dr. Wong, Hau San
1st Reader: Dr. Ngo, Chong Wah
2nd Reader: Dr. Kwong, Sam)
Student Final Year Project Declaration

I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

Web Profiling and Navigation Path Analysis

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name: Lam Ming Yan
Signature: ________________________
Student ID: ________________________ Date: 18th April, 2006
Abstract

The rich information in the World Wide Web has created an information overloading problem for web users. The technology of Web personalization targets to solve such a problem and thus recommendation systems were studied. Nowadays, Web Usage Mining technique is overtaking the position of traditional collaborative filtering technology, moving traditional ranking style into web log and navigation path analysis approach. The Markov model is one of the well performed prediction model that is suitable for a recommendation system. However, it suffers from state complexity problem and needs to be improved. Association Rule mining is another technique for data mining and pattern discovery. There is also room to improve for the performance of Association Rule. This project focuses on the studies of different optimization schemes for Markov model and Association Rule. It takes experiments on different kinds of improvement studies. It also shows the result of different kinds of model optimization scheme. A profiling modeler and an online recommendation engine are also implemented for testing and simulation.
Acknowledgement

I would like to thank my supervisor, Dr Wong, Hau San, for his guidance, encouragement and invaluable advice during the year. His opinions have given me great inspirations on my studies. Without his suggestions, it would be hard to imagine finishing this project on the limited given time. My gratitude also goes to my first reader Ngo, C. W., whose teaching and lecture notes on Data Mining help me understand many of the concepts related to my project. I would also like to acknowledge the developers of ACM Digital Library, who creates a helpful knowledge base search service that help me find out most of the papers I wanted. Finally, I would like to thank my families for giving me enormous support and unselfish love during my studies.
# Table of Content

Abstract ..................................................................................................................................... i  
Acknowledgement................................................................................................................... ii  

1. Introduction.......................................................................................................................... 1  
   1.1 The Problem ................................................................................................................ 1  
   1.2 Web Personalization .................................................................................................... 2  
   1.3 WUM and Navigation path analysis ............................................................................ 3  
   1.4 Project Objectives ........................................................................................................ 3  
   1.5 Background and Literature .......................................................................................... 5  
      1.5.1 Recommender System ......................................................................................... 5  
      1.5.2 Collaborative Filtering Web Profiling .................................................................... 6  
      1.5.3 The Well Known Collaborative Filtering Problems ............................................... 7  
      1.5.4 Web Mining and Web Usage Mining .................................................................... 8  
         1.5.4.1 Web Mining .................................................................................................... 8  
         1.5.4.2 Web Usage Mining ...................................................................................... 10  
         1.5.4.3 The Process of Web Usage Mining ..............................................................11  
         1.5.4.4 Pattern Discovery and Analysis ................................................................... 12  
      1.5.5 Other Related Works .......................................................................................... 14  

2. SMMAR ............................................................................................................................... 15  
   2.1 Selective Markov Model ............................................................................................ 15  
      2.1.1 The Markov Model .............................................................................................. 15  
      2.1.2 Order of States ................................................................................................... 16  
      2.1.3 All $K^{th}$ Order Markov Model ........................................................................... 18  
      2.1.4 Exact $K^{th}$ Order Markov Model ........................................................................ 19  
      2.1.5 Trust on Markov States ....................................................................................... 21  
         2.1.5.1 Error Rate of State ....................................................................................... 22  
         2.1.5.2 Support Frequency of State ......................................................................... 23  
   2.2 Association Rule Mining ............................................................................................. 23  
      2.2.1 Frequent Itemset ................................................................................................ 24  
      2.2.2 Rule Item Size .................................................................................................... 25  
      2.2.3 Error Rate of Rule ............................................................................................... 26  
      2.2.4 Support Frequency of Rule ................................................................................ 26  
   2.3 Combination of both Models ...................................................................................... 27  

3. Model Optimization Scheme............................................................................................. 28  
   3.1 Selective Markov Model Pruning Schemes ............................................................... 29  
      3.1.1 Frequency Pruning Scheme ............................................................................... 29  
      3.1.2 Confidence Pruning Scheme .............................................................................. 30  
      3.1.3 Error Pruning Scheme ........................................................................................ 30  
   3.2 Association Rule Mining Pruning Schemes ............................................................... 31  
      3.2.1 Fine Tuning Scheme ......................................................................................... 32  
         3.2.1.1 Maximum Item Size Pruning Scheme ......................................................... 32
1. Introduction

1.1 The Problem

The World Wide Web has become the richest source of information. Web sites are providing large amount of useful information to users. However, most of the content is not targeted at any specific group or layer. The huge variety of content has caused an information overloading problem. Through navigating pages from pages, it is not uncommon for a user to get lost or have numerous of ambiguous results as there is only a small portion that is truly relevant or useful to one specific user group. To the users who have an inquiry goal, this is exhausting and time consuming. Much time and effort has to be spent to filter irrelevant information. For those who have eventually missed the goal through the confusing navigations, they may regard the site as not useful and leave. Such result would be a big failure for one resourceful web site.

To tackle this information overloading problem, site owners must find a way to alleviate the difficulty in order to retain users. The requirement for anticipating the needs of users and improving the web site usability were therefore addressed. Many E-Commerce systems such as Amazon.com and Ebay.com have applied methodology to construct server-directed recommender system of Web personalization technology for this purpose (Schafer et al. 1999). These commercial systems recommend products to potential consumers based on previous transactions and feedback. They are becoming part of the standard e-business technology that can enhance e-commerce sales by converting browsers to buyers, increasing cross-selling and building customer loyalty (Schafer et al. 1999).


1.2 Web Personalization

The term Web personalization is defined by Eirinaki et al. 2003 as any action that adapt the site information or services with knowledge retrieved from users’ navigational behavior and individual interests, to combine with the content and structure of the Web site. A Web personalization system aims to “provide users with the information they want or need, without expecting from them to ask for it explicitly” (Mulvenna et al. 2000).

General Web personalization applications are including layout customization and recommendation systems. There are various methodologies and research studies developed for Web personalization. Many of them are related to collaborative filtering algorithms (Herlocker et al. 2000) and Web usage mining technique (Spiliopoulou, 2000). For instance, virtually all personalization approaches are based on analyzing historical customer session data (Datta et al. 2001). The historical session data is used for formulating real-time responses based on current online navigation patterns. However, as the amount of history data grows in the database, the efficiency of providing real-time responses decreases. The problem is even worsening with the increases of user loading for the web site, thus defined a scalability problem related to personalization technologies (Datta et al. 2001).

Later studies have proposed combinations of different models and approaches with Web usage mining (WUM) technique to solve the scalability and sparsity problems of collaborative filtering (e.g. Xue et al. 2005). Though approaches using clustering or eliminating historical data can alleviates the scalability problem, it inevitably decreases the accuracy in recommendation. The reliability of a Web
recommendation system is affected. There is still room for improvement regarding the performance of Web personalization.

### 1.3 WUM and Navigation path analysis

*Web usage mining* is the process that uses statistical and data mining methods to analyze the Web log data (usage or clickstream data), creating a set of useful patterns that indicate user’s navigational behavior (Erinaki et al. 2003). Web usage mining plays a key role in personalization and there are many research studies such as Web Usage Miner from Berendt and Spiliopoulou 2000 and WebSIFT from Cooley et al. 1999.

*Pattern discovery and analysis* are important steps in Web usage mining procedure. *Pattern discovery* is the step that uses methodologies or algorithms of data mining, statistics, machine learning, and pattern recognition to find out statistics, rules and patterns from the pre-processed data. *Pattern analysis* refers to the process to filter out uninteresting rules or patterns from the result constructed in the pattern discovery step. Content and structure information can be used during filtering.

### 1.4 Project Objectives

This project aims at experimenting and exploring different models or methodologies in pattern discovery and analysis. It will experiment on different ways to make improvement in the performance of pattern analysis, and to contribute to a better accuracy in Web recommendation.
In the Web domain, user profiling is the “process of gathering information specific to each visitor, either explicitly or implicitly” (Eirinaki et al. 2003). For explicit user profiling, there is a problem of sparsity, which refers to insufficient user data to identify customer interests (Huang et al. 2004). Therefore implicit user profiling is emphasized in this project.

In this project we also focus on recommender system of Web personalization using Web usage mining – the process that use statistical and data mining methods to analyze the Web log data, creating a set of useful patterns and user profiles that indicate users’ navigational behavior (Eirinaki et al. 2003). This knowledge can be used to the systems to personalize the Web site or create recommendations according to each user’s behavior and profile.

Objective Checklist:

- Study knowledge regarding Web Usage Mining and navigation pattern analysis
- Explore pattern discovery and analysis models and algorithms for Web usage mining.
- Explore different recommender technology
- Experiments on different pattern discovery models and algorithms.
- Experiments on different optimization schemes
- Implement a recommendation engine
- Evaluate and make improvement to the recommendation quality.
1.5 Background and Literature

This part introduces the background history and literature of related technology used in recommendation systems. The part covers collaborative filtering, Web usage mining and pattern discovery knowledge.

1.5.1 Recommender System

One important form of Web personalization is recommendation of products, documents or services. A recommender system is an application that trying to advice users about product or services they may be interested in. Research on recommender systems usually overlaps two main computer science topics: Information Retrieval (IR) and Artificial Intelligence (AI) (Caesarea Rothschild Institute, 2005).

For IR, recommendation system derives the vision of users who are engaged in an information search process. They query a content repository in order to obtain a collection of matching items, typically in the form of a ranked list.

For the AI perspective, user’s interest will be examined from the user profile or some piece of user information. The recommendation system will try to exploit past search behaviors of a community of users, in order to predict the user’s interest on unseen items.

There has been an increasingly important type of recommender system which comprises those generated recommendations for groups rather than for individuals.
For it is believed that the experiences of a group of members would have supportive similarities to infer another group member.

### 1.5.2 Collaborative Filtering Web Profiling

Collaborative filtering methodology is the most common and traditional recommendation system technology so far. Over the years, there has been many research studies related to modeling and developing a recommendation system. Recommender Web personalization has been widely used among many Web sites and Web applications, for example, Amazon.com and Launch.com. There is much work published regarding the step of analyzing the Web data which aims to discover users’ interests and preferences in order to determine the actions that should be performed. Approaches can be generally categorized into three categories: content-based filtering, collaborative filtering and rule-based filtering (Mobasher 2000; Erinaki et al. 2003).

*Content-based filtering* systems, for example WebWatcher (Joachims et al. 1997), are those based on individual users’ preferences and recommends item to them that are similar to the user liked in the past.

*Collaborative filtering* approach, which is the most popular approach among all, works by building a database of preferences and interest of customers, and match new customer based on the similarity of the other customers’ information in the database, then returns recommendations that is predicted to be interesting to them. Such approach is based on the assumption that users with similar behavior have similar interests.
Rule-based filtering systems (also called Manual decision rule systems by Mobasher 2000), for example, Broadvision (www.broadvision.com), asked users to answer a set of questions (usually during registration), then response the user with a tailored result (e.g., a list of products) to his needs. The questions and results can be based on a set of rules that are specified by administrators. Rule-based filtering systems can also combine with content-based and collaborative filtering system for more accurate conclusions.

Among these three approaches, the collaborative filtering approach (Hill et al. 1995; Resnick et al. 1994; Shardanand and Maes 1995) has been the most commonly used and successful one for both research and practice (Huang et al. 2004). Research projects like GroupLens (Konstan et al. 1997), Ringo (Shardanand and Maes 1995), Video Recommender (Hill et al. 1995), and MovieLens (Dahlen et al. 1998) gained large attentions and was followed widely on the Internet (Herlocker et al. 2000). For instance, some of the highest profile commercial web sites like Amazon.com, CDNow.com, MovieFinder.com and Lauch.com are making use of successful collaborative filtering technology (Herlocker et al. 2000).

1.5.3 The Well Known Collaborative Filtering Problems

Though the collaborative filtering technology has been widely adopted, it suffered from three major problems: data sparsity, system scalability, and synonymy (Sarwar et al. 2000). Data sparsity problem refers to the difficulty to correctly identify customer interests due to insufficient transactional and feedback data.
Accuracy rate also decreased as static user profiles aged. *System scalability* problem refers to the computational complexity due to large scale of history data which decreases the response time of a personalization system in real-time recommendation. *Synonymy* refers the inability for correlation based recommender systems to differentiate objects with similar names and yet different categories.

Problems of the collaborative filtering technology lead to research studies in reliability of recommendation systems. In Herlocker et al. 2000, the authors explained the reason why collaborative filtering maybe successful in low risk entertainment domains, but yet to be trusted for high-risk content domains. Later, studies were made to overview the factors that should be considered in evaluating recommender systems (Herlocker et al. 2004). In O’Donovan and Smyth 2005, two trust models were developed for evaluating the accuracy of predictions with recommender system profiles over an extended period of time.

### 1.5.4 Web Mining and Web Usage Mining

#### 1.5.4.1 Web Mining

The problem of collaborative filtering technology has also lead to a new revolution of Web personalization, which is to incorporate with the benefit of Web Mining techniques. Usually, there are three types of data to be managed in a web site: content, structure and log data. *Content data* consists of web pages, images or documents, whatever that construct a web page; *structure data* refers to the organization of the content, how the pages link and how the contents organized; *log data* are sever logs and usage logs that store the usage history of the web site which
contains interesting usage patterns. There are three research directions of the Web mining techniques to these different data sets: web content mining, web structure mining and web usage mining.

Web content mining refers to the discovery of useful information from the web content data. It is the process of automatically retrieving, filtering, and categorizing Web documents. Typically, Web content mining will only make use of texts on Web pages, thus valuable information implicitly contained in hyperlinks is overlooked.

Web structure mining (Chakrabarti et al. 1999) reveals more information than just the web content. It infers useful pattern from the Web’s link topology to help retrieve high quality Web pages. The most common Web structure mining algorithms are HITS (Gibson 1998) and PageRank (Brin and Page 1998). An example of Web structure mining project is LinkSelector (Fang and Sheng 2004).

Web usage mining is the process that use statistical and data mining methods to analyze the Web log data (usage or clickstream data), creating a set of useful patterns that indicate user’s navigational behavior (Erinaki et al. 2003). Web usage mining plays a key role in personalization and there are many research studies such as Web Usage Miner from Berendt and Spiliopoulou 2000 and WebSIFT from Cooley et al. 1999.
1.5.4.2 Web Usage Mining

Among the three Web mining techniques, Web usage mining is the most adaptive technique to Web personalization and there are increasing focuses on Web usage mining for Web personalization research studies. There are several reasons (Mobasher et al. 2000) for this. First of all, the usage data reflect real user behaviours during their activities in the web site, which means the input is not a subjective description of the users by themselves. Therefore it is not subject to biases. Secondly, the user profiles are dynamically obtained from navigation patterns, thus the system performance will not suffer from the degradation as the profiles age. Thirdly, Web usage mining can reduce the need for obtaining subjective user ratings or registration-
based personal preferences and thus can improve the problem of missing semantic relationships among Web objects by just using content similarity to obtain aggregate profiles.

1.5.4.3 The Process of Web Usage Mining

As shown in the Figure 1.2, there are mainly three steps in the process of Web usage mining (Srivastava et al. 2000). During the process, raw data log will be analyzed and turn into interesting patterns and statistics. The three main tasks are including data preprocessing, pattern discovery, and pattern analysis.

*Data preprocessing* refers to the step of converting the usage data, content and structure information from various data sources into clean and necessary format for pattern discovery purpose. Session and page view data will be identified in this stage.
Pattern discovery is the step that use methodologies or algorithms of data mining, statistics, machine learning, and pattern recognition to find out statistics, rules and patterns from the preprocessed data.

Pattern analysis refers to the process to filter out uninteresting rules or patterns from the result constructed in the pattern discovery step. Content and structure information can be used during filtering.

1.5.4.4 Pattern Discovery and Analysis

Nowadays more and more attention has been paid to the study of user behavioural pattern. Pattern information like clickstream data and purchase history data reflect meaningful rules that are valuable essence to an intelligence system.

Pattern discovery and analysis are the most important steps of Web usage mining. The purpose of pattern discovery and analysis is to extract meaningful patterns (also referred as rules) and statistics from users’ usage data that reflects the user interests and behaviours. These meaningful usage patterns and statistics can provide useful information for site modification, site improvement and business intelligent development. For a recommender system, it can be used to predict a user’s interest in web content and web browsing behaviour.

To cope with the enormous amount of data in web logs, many statistical models of web surfing are proposed (Borges and Levene 1999; Chen and Zhang 2003; Spiliopoulou and Faulstich 1998).
One of the most popular pattern discovery techniques is association rule mining (Agrawal and Srikant 1994). Association rule mining discovers correlations between items in transactional databases. An association rule carries an implication of one frequent itemset implies another (R: X ⇒ Y, X∩Y=∅), given the minimum support and confidence. Association rule mining in Web personalization and recommendation system is usually used in analysing purchasing history data and browsing activity of web visitors.

Another typical example is Hypertext Probabilistic Grammar (HPG) model (Borges and Levene 1999). “HPG is a probabilistic regular grammar which has a one-to-one mapping between the set of non-terminal symbols and the set of terminal symbols.” (Borges and Levene 1999) This model rely on the Makov assumption with history depth k (Markov Models). This is because the probabilities of a HPG corresponds to probabilities of the transition matrix in a Markov Chain (Ross 1998).

More about association rule mining and Markov model will be discussed in the next section.
1.5.5 Other Related Works

There are other related Web usage mining research studies related to recommendation system and navigation path analysis. Jin et al. 2004 have invented a PACT framework based on PLSA model for discovery and analysis of Web navigational patterns. Kim et al. 2004 have investigated a clickstream-based collaborative filtering personalization model to solve the scalability problem of collaborative filtering. Jin et al. 2005 proposed a Web recommendation system based on Latent Dirichlet Allocation (LDA) to discover the hidden semantic relationships among items. Eirinaki et al. 2005 have proposed a PageRank-style algorithm based on Markov models.
2. SMMAR

2.1 Selective Markov Model

The Selective Markov Models (SMM) are introduced by Deshpande and Karypis 2004 to solve the state complexity problem of All-Kth-Order Markov models, it eliminates most of the unreliable states and retains the confidence states. By applying pruning schemes, state searching efficiency and state reliability can be improved while performance of the overall model can still be maintain. Three schemes in SMM are designed for pruning unnecessary states: Frequency-Pruned Markov Model; Confidence-Pruned Markov Model; and Error-Pruned Markov Model. Frequency-Pruned Markov Model prunes the states with low occurrences while Confidence-Pruned Markov Model prunes the states with confidence coefficient lower than the threshold. Finally, Error-Pruned Markov Model will prune out the states with higher error rate. Details of state pruning will be described in the next sections.

2.1.1 The Markov Model

Markov Models (Papoulis 1991) is used in studying and understanding stochastic processes. It has been shown to be well suited for modeling and predicting a user’s browsing behavior on a Web site (Jespersen et al. 2003). It is based on the transition probabilities between web pages.
The fundamental property of Markov model is the dependency on the previous state. A Markov chain consists of a set $S = \{s_1, s_2, \ldots, s_n\}$, called the state space. In the context of Web usage mining, the state space is equal to a set of pages. This set of pages is referred to as a Web session, and it is the sequence of pages being accessed. A matrix $M$ stores the probabilities from one state to another. Each element of the matrix $M[s_i, s_j]$ can be estimated as:

$$M[s_i, s_j] = \frac{\text{Count}(s_i, s_j)}{\sum_{s_j} \text{Count}(s_i, s_j)}$$  \hspace{1cm} (1)$$

In Markov chain we move from state $s_i$ to $s_j$ with probability $M[s_i, s_j]$, where $s_i$ is at time $t = k$ and $s_j$ is at $t = k + 1$.

**2.1.2 Order of States**

The number of preceding pages $k$ that the next page depends on is called the order of Markov Model, and the resulting model is called the $k^{th}$ – order Markov Model (Papoulis 1991). Figure 2.2 shows a state transition of model with history depth 2.
The 1st - order Markov models provide a simple way to capture sequential dependency, but do not take into consideration the “long-term memory” aspect of web surfing behaviour. In general, higher – order Markov model have greater accuracy for predicting navigational paths. However, by using one higher – order Markov model, the coverage will be reduced and accuracy could also be reduced. Moreover, as the number of order increases, the number of states increases, thus there will be a problem of state-space complexity. Figure 2.3 shows the correlation of number of order to the accuracy, coverage and number of states.

As we can see from the graph, the model accuracy increases as the number of order increases. In the web page navigation point of view, one can predict more accurate recommendations while more browsing history is given. The more clickstreams are provided, the more precise results are returned. However, it also
reveals from the graph that the coverage decreases, and the number of states increased dramatically in the same time. The use of one $K^{th}$ order Markov model requires $K^{th}$ input pattern items, which does not cover the patterns with insufficient items. Moreover, in a Web site with $P$ pages, there are a total of $\Theta\left(\left[P^k\right]\right)$ states in a $K^{th}$-order Markov model, and a total of $\Theta\left(\left[P^{k+1}\right]\right)$ conditional probabilities that need to be estimated from the training set. It is unrealistic to expect all or most of the $\Theta\left(\left[P^k\right]\right)$ high order states to be collected in the training set. The occurrence of the high order states is usually very low. Therefore the coverage of the model drops as the number of states increases.

### 2.1.3 All $K^{th}$ Order Markov Model

The combination of All – $K$th – Order Markov model (Pitkow and Pirolli 1999) is to solve the coverage problem of a $K^{th}$-order Markov model. The idea is to train different order of Markov models, i.e. order from 1 to $K$, and then combines all the models together for prediction.

During the prediction, high order model will be used first for the prediction. Given a long sequence of pattern that forms an order of $L$ state ($0 \leq L \leq K$), the $L$ order Markov model will be used first for the prediction. Once the high order state is not covered in the model, a lower order state $L-1$ will be considered and the $L-1$ order Markov model will be used, and so on, until the state is finally discovered. Since the $1^{st}$ order Markov model usually covers all of the $1^{st}$ order states, the combination of all $K^{th}$ Markov models will ultimately covers all of the given patterns and returns a
result. Taking advantages of both high order Markov model and low order Markov model. Thus the coverage can be retained as high as the 1\textsuperscript{st} order Markov model.

While this approach may solve the problem of reducing coverage of high order Markov model, it has worsen the problem of state-space complexity, which refers to the problem of handling large scale of states. For there are $\Theta(P^k)$ states in a $K$\textsuperscript{th}-order Markov model, the combination of all $K$\textsuperscript{th}-order Markov models will make a total of $\sum_i \Theta(P^k)$ states and a total of $\sum \Theta(P^{k+1})$ conditional probabilities that need to be estimated from the training set. The high number of states will be a burden for a real-time response recommendation system.

### 2.1.4 Exact $K$\textsuperscript{th} Order Markov Model

The Exact $K$\textsuperscript{th} order Markov Model is different from All $K$\textsuperscript{th} Order Markov model above. But it is also make use of the combination of all $K$\textsuperscript{th} Order Markov model. It can solve the coverage problem of insufficient pattern items but the accuracy rate will be higher. The difference is that it does not automatically switch the high order states into lower order until the 1\textsuperscript{st} order is met or the state is found. Although the coverage rate of Exact $K$\textsuperscript{th} Order Markov Model is lower than the All $K$\textsuperscript{th} Order Markov Model, our experiments show that the accuracy of Exact $K$\textsuperscript{th} Order Markov Model is generally higher than the All $K$\textsuperscript{th} Order Markov Model. Figure 2.4 shows the comparison of accuracy of two prediction models after applying all the three Selective Markov Models pruning schemes.
The pruning scheme of Selective Markov Model favour lower order Markov model than the higher order Markov model. Most of the unreliable high order states were pruned after applying pruning schemes, meaning that the remaining higher order states are generally more reliable. Moreover, the pruning scheme of Selective Markov Models never prunes the 1st order states in order to retain the coverage of the model. Therefore the error rates of the 1st order Markov model are brought into the overall accuracy, taking into account the lower accuracy rate.

As the Exact Kth Order Markov Model favours the reliable higher order Markov model, it is taking advantage of all the Selective Markov Model pruning schemes. However, the coverage rate of the Exact Kth Order Markov Model is still lower than the All Kth Order Markov Model and has to be improved. Figure 2.5 shows the comparison of accuracy of two prediction models after applying all the three Selective Markov Models pruning schemes.
Initially both models return a coverage rate of 100%. After applying several pruning schemes, the coverage rate of Exact Kth Order Markov Model dropped. But the coverage rate of All Kth Order Markov Model remains the same as the 1st order states were used to cover states that are not found in higher order models. To solve this problem, we can combine the exact Kth order Markov model with other methodologies such as Association Rule mining and will be discussed later.

### 2.1.5 Trust on Markov States

The number of states in one Markov model increases dramatically as the number of order increases, bringing the burden on recommendation performance and accuracy. However, not all of the discovered Markov states are reliable to a prediction model.

The Markov states are collected using training dataset which is the navigation usage log of a large amount of users who visit the site for various reasons. Regular
users, member users, users with visiting reasons or targets create valid and reasonable navigation paths in the usage log. Other users who do not have any inquiry goals or targets may browse the site with arbitrary decisions, affecting the reliability of generated usage log. There are also some exceptional users whose interests are very different to the majority.

On the other hand, if the training data is collected during the promotion period of some of the pages, or in some case one page is visited frequently in a special period (e.g. release of examination result for a department web site), the record of the outgoing pages of one state will have burst of visits to these particular pages. Such record affects the prediction on the future. Therefore, there exist invalid states in the model which should be eliminated from the model.

2.1.5.1 Error Rate of State

In the prediction testing, some states give high accuracy and some always give a wrong prediction. The error rate of one Markov state reflects the reliability of the state. Moreover, the recommendation items are based on the most frequent outgoing pages of one state. As mentioned before, the outgoing pages can be affected by promotional period of some pages that affects the future predictions. Therefore, the factor of unrepresentative high frequency outgoing pages also exists to affect the state error rate.
Since the error rate of states reflects the precision of prediction model, if the error rate of each state can be collected, it could be able to contribute an optimization on model accuracy.

2.1.5.2 Support Frequency of State

It is observed that states with low occurrence in the training data generally tend to have lower prediction accuracies (Deshpande and Karypis 2004). This is due to the reason that for such states, the maximum likelihood estimation of the conditional probabilities will not be reliable. Markov prediction is highly dependent on the training data provided. Low frequency state may imply the invalidity of the training data which construct this state. As mentioned before, such sparse data could be some usage of users who loafed around with no special target or browsed with arbitrary decisions. Sometimes it could be some users who wrongly clicked one link.

The support frequency of a state reveals the state reliability. It should be taken in to consideration of elimination in order to shape down the model complexity.

2.2 Association Rule Mining

Association rule is a data mining algorithm used to perform summarization of the data, classification of data with respect to a target attribute (Nayak and Cook 2001). It identifies correlations between items in a transactional database. An association rule is an implication of the form (R: X ⇒ Y, X∩Y=∅), given the minimum support and confidence.
Given a set of transactions, each contains a set of items. An association rule $X \Rightarrow Y$ may be discovered in the transaction, where $X$ and $Y$ are frequent itemsets in a transactional database. The percentage of records that contain both $X$ and $Y$ in the database is called the support of the rule, and the percentage of records containing $X$ and also contain $Y$ is call the confidence of the rule. Association rule mining discovers the rules from training transaction data which has the support and confidence above or equal to the minimum rates.

Most association rule algorithm generate association rules in two steps: (1) Generate all frequent itemsets, and (2) Construct all rules using these itemsets. Well known association rule algorithms are including Apriori, Charm, FP-growth, Closet and MagnumOpus.

### 2.2.1 Frequent Itemset

An itemset is a collection of one or more items in a transactional database. E.g. \{coffee, sugar, milk\} is an itemset. The number of times of all items in an itemset occurs in a transactional dataset is called the count (frequency) of that itemset. The percentages of records that contain both $X$ and $Y$ in the database is called the support. Frequent Itemset is an itemset whose support is greater than or equal to a minimum support threshold and minimum confidence threshold.
<table>
<thead>
<tr>
<th>TID</th>
<th>ITEMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Coffee, Sugar</td>
</tr>
<tr>
<td>2</td>
<td>Coffee, Milk, Bread, Egg</td>
</tr>
<tr>
<td>3</td>
<td>Sugar, Milk, Bread, Bacon</td>
</tr>
<tr>
<td>4</td>
<td>Coffee, Sugar, Milk, Bread</td>
</tr>
<tr>
<td>5</td>
<td>Coffee, Sugar, Milk, Bacon</td>
</tr>
</tbody>
</table>

<Table 2.1 Example of Market-Basket transactions>

Given the above transaction dataset, the support and confidence of the rule \{Coffee, Sugar\} \Rightarrow Milk are:
\[
s = \frac{\sigma(\text{Coffee, Sugar, Milk})}{|T|} = \frac{2}{5} = 0.4, \quad c = \frac{\sigma(\text{Coffee, Sugar, Milk})}{\sigma(\text{Coffee, Sugar})} = \frac{2}{3} = 0.67
\]

2.2.2 Rule Item Size

A rule contains itemsets of LHS and RHS such that LHS \Rightarrow RHS and LHS \cap RHS=\emptyset. The total item size is equals to the size of frequent item set generated in the first step. Frequent itemset with item size one does not contain a RHS itemset, so is not need after the construction of association rules. Therefore it can be eliminated after the construction. In a web recommendation system, the number of user clickstreams will be controlled to a limit. The number of clickstreams used in the recommendation calls the window size. If one rule contains more LHS items than the input window size, the rule will not be applicable for prediction and can be neglected.

Item size of frequent itemsets of a rule is correlated to the design of a recommendation system. Therefore pruning scheme should be considered for controlling maximum and minimum frequent itemset size.
2.2.3 Error Rate of Rule

Sometimes the high support or high confidence of one frequent itemset does not actually give a high accuracy rate. For number rule recommendations with similar support or confidence rate, the hidden high error rule will worsen the recommendation performance. Those rules with high error rate should be considered less reliable and should be removed to eliminate the affect on prediction.

2.2.4 Support Frequency of Rule

Support frequency is usually used during Frequent Itemset discovery step. It reflects the occurrence of one itemsets in the transactional dataset. Sometimes higher support rate does not necessary give higher accuracy. However, experiments show after applying some pruning schemes to the rules, for example pruning high error rate rules from rule database, the prediction by support rate can give a better performance.

Support rate is also correlated to the coverage of rule model in prediction system. If the minimum support is too high, the coverage of navigation path prediction will be affected. If the minimum support rate is too low, it will generate too many rules. Different dataset have different characteristic and applies to different minimum support. Therefore, during the testing and validation phase of association rule, it is necessary to make adjustment to the minimum support rate.
2.3 Combination of both Models

The Exact Kth Order Markov Model gives higher accuracy rate (precision) to recommendation result after applying pruning schemes. Although it also improved the coverage problem of high order Markov model, its coverage rate is still not satisfactory and can be improved. The prediction model built by association rule mining generally has high coverage. There can also have a sequence of rule pruning and optimization scheme to improve the accuracy of the association rule prediction. This model can be a compliment to the states that is not covered by the Exact Kth Order Markov Model. Therefore, in this project we propose a combination of the Selective Markov Models and Association Rules (SMMAR) as a recommendation prediction model scheme.

As Exact Kth Order Markov Model favor higher order Markov states, it never discard the navigation information collected from the user within the given window size. By using the association rule model to compliment the Exact Kth Order Markov Model, the user clickstreams collected can be fully utilized for the recommendation process. Also, the coverage can be increased to 100%.
3. Model Optimization Scheme

After the model constructions, there will be a group of different order Markov models that contains many states, and there will also be a list of association rules that has different item size.

Before the model is used in a recommendation generation process, there is a need to eliminate useless, high error rate, unreliable states and rules from the models. Therefore we apply different pruning schemes to the constructed Markov models and rule model to optimize the model efficiency, accuracy and space usage.

In this project we adapt the pruning schemes of Selective Markov Models proposed by Deshpande and Karypis 2004 for constructed Markov models. They are Frequency Pruned Markov Model, Confidence Pruned Markov Model and Error Pruned Markov Model. We further elaborate the Error Pruned Markov Model to add the validation for outgoing pages in order to optimize the model space usage.

For the association rule pruning, we have investigated and taken experiments on our pruning schemes on Association Rule model to try to increase the rule accuracy. There are also three rule pruning scheme experimented, they are Fine Tuning Scheme, and Error Pruning Scheme.
3.1 Selective Markov Model Pruning Schemes

The Deshpande and Karypis 2004 have proposed three schemas for pruning the states of the All-Kth-Order Markov model in order to solve the state complexity problem and meanwhile maintain the coverage and accuracy of the prediction model. The three pruning schemes are Frequency-Pruned Markov Model, Confidence-Pruned Markov Model and Error-Pruned Markov Model.

3.1.1 Frequency Pruning Scheme

*Frequency-Pruned Markov Model* (FPMM) is the pruning scheme that prunes the states with low frequency. By observation, states have lower prediction accuracies if they occur with low frequency in the training set. Consequently, they can be eliminated without affecting the accuracy of the result. The amount of pruning in this scheme will be controlled by a parameter $\Phi$ call the frequency threshold. States that has the frequency lower than $\Phi$ will be pruned.

However, the frequency parameter $\Phi$ must be carefully chosen as FPMM tends to prune higher order states. For a web site that have thousands of pages, the higher order states usually have lower occurrence in the training data. As higher order Markov model have higher accuracy, there is possibility that the pruning will affect the accuracy if the frequency parameter $\Phi$ is not in a proper value. The target of FPMM pruning states are those rare pattern and invalid browsing patterns. Therefore the frequency parameter $\Phi$ must be small enough and not to affect the real valuable high order states.
3.1.2 Confidence Pruning Scheme

*Confidence-Pruned Markov Model* (CPMM) prunes a state if the probability difference of the most probable page and the second most probable page is below the confidence threshold ($\Phi$). This confidence threshold is calculated by using the probability ($\hat{p}$) of the most probable page, the frequency of the Markov state ($n$), and the confidence coefficient ($z_{\alpha/2}$). The confidence threshold ($\Phi$) is computed by:

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

The confidence coefficient ($z_{\alpha/2}$) are varied corresponding to $z$ value associated with the different confidence levels. For a 55% confidence, the $z_{\alpha/2}$ will be 0.75, for a 99% confidence, the $z_{\alpha/2}$ value will be 2.57.

3.1.3 Error Pruning Scheme

*Error-Pruned Markov Model* (EPMM) prunes the states with higher error rate. This approach is accomplished by performing a validation step after training. The validation step is referred to the step of estimating the state errors using part of the training set (validation set) which is not used during the model-building phase. There are two strategies for EPMM, first one is referred as overall error pruning, and the second one is individual error pruning.
In the Overall Error Pruning scheme, each K-Markov model is individually validated using the validation set. For each Web session in the validation set, the state of the Markov model is identified and the result of the prediction is recorded for that state. After performing the validation evaluation for all Web sessions in the validation set, each state will carry one error rate. If the higher-order Markov state is higher than any of its corresponding lower-order states, the state is pruned.

For example, the 3rd order state \( \langle p_i, p_j, p_k \rangle \) has a correspondent 2nd order state \( \langle p_j, p_k \rangle \) and 1st order state \( \langle p_k \rangle \). If the error rate of \( \langle p_i, p_j, p_k \rangle \) is higher than \( \langle p_j, p_k \rangle \) or \( \langle p_k \rangle \), the state \( \langle p_i, p_j, p_k \rangle \) is pruned. The same to second order state, if \( \langle p_j, p_k \rangle \) has higher error rate than \( \langle p_k \rangle \), \( \langle p_j, p_k \rangle \) is also pruned. However, in order to retain the coverage of the model, the first order state is never pruned.

### 3.2 Association Rule Mining Pruning Schemes

After the discovery of correlation of items, a number of rules will be generated for the association rule prediction models. The number of support rate chosen for generating rule should be depend on the available dataset size. It should also depend on the size of itemset. Figure 4.1 shows the size of rule generated with the MSWEB dataset.
The size of rules generated decreases as the support increase. Although high support rate may imply a more reliable set of rules, there could still be useful rules with lower support %. For example, a 0.1% support rule already has a 44 itemset count in the dataset. It is possible that the smaller group of user also gives a precise correlation on navigation path. Therefore, we can consider a smaller initial support for rule discovery, and then apply other pruning scheme for discovered unreliable rules discovered.

### 3.2.1 Fine Tuning Scheme

The Fine Tuning Scheme contains Maximum Item Size Pruning Scheme, Minimum Support Pruning Scheme, and Minimum Confidence Pruning Scheme.

#### 3.2.1.1 Maximum Item Size Pruning Scheme

The item size refers to the total items in itemsets of a generated rule. As the number of input navigation patterns can be controlled by our program, the maximum item size of the rule can also be adjusted. The sequence of items in a rule does not
necessary link together, we can set a maximum item size for the rule to eliminate unnecessary rules. The maximum item size $K$ can be specified during the modeling phase, but we have taken it out as a control parameter during the model optimization phase. Figure 4.1 shows an example of optimization result after pruning with different item size.

![MSWEB Dataset](image)

Results of the experiment show that rules with bigger item size can be pruned without affecting the accuracy of the model. It can also contribute to an increase of prediction accuracy.

3.2.1.2 Minimum Support Pruning Scheme

Different datasets have different optimal support parameters for prediction. The support parameter can be specified upon construction of the rule model, it should also be one control parameter during the optimization of the model. Upon discovery of frequent itemset, we can set a smaller initial support parameter in order to find out more potential rules. During the testing and optimization phase, the parameter can be further adjusted to prune out the unnecessary rules.
### 3.2.1.3 Minimum Confidence Pruning Scheme

<table>
<thead>
<tr>
<th>Frequent Itemset</th>
<th>Frequency</th>
<th>Support</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1038 1041 1026 1034</td>
<td>90</td>
<td>0.20%</td>
<td>0.33</td>
</tr>
<tr>
<td>1058 1018 1034</td>
<td>90</td>
<td>0.20%</td>
<td>0.35</td>
</tr>
<tr>
<td>1037 1003 1009 1004</td>
<td>90</td>
<td>0.20%</td>
<td>0.49</td>
</tr>
<tr>
<td>1001 1017 1018 1009 1004</td>
<td>90</td>
<td>0.20%</td>
<td>0.50</td>
</tr>
<tr>
<td>1039 1034 1008</td>
<td>90</td>
<td>0.20%</td>
<td>0.59</td>
</tr>
<tr>
<td>1037 1001 1017 1018 1009</td>
<td>90</td>
<td>0.20%</td>
<td>0.90</td>
</tr>
</tbody>
</table>

<Table 3.1 Rule support and Confidence>

As we can observe from the above table, rule items that have the same support do not necessarily have the similar confidence. The above example show the confidence of the same support itemset varies from 0.33 to 0.90.

Higher confidence rule generally gives a higher accuracy, for those low confidence rule, we can prune it away to improve the model accuracy.

### 3.2.2 Error Pruning Scheme

The prediction performance of the generated rules shall be known after the testing. The rules with frequent error predictions shall be eliminated from the model. Thus we propose a validation phase for validating the performance of generated association rules. During the validation phase, validation data will be used to validate the individual rule accuracy. The validation data is a part of the training data that is not used in the training phase. After the validation phase, error rate will be computed for each rule. A maximum error tolerance parameter $\omega$ is then be specified and used to prune out the rule items that have error rates higher than the given tolerance parameter.
4. Implementation

This section gives an overview of implementation detail. It describes the dataset, methodology and technology used in implementation. It also gives a picture on the application and website we have implemented.

4.1 Overview

In this project, we adopted the process of Web Usage Mining and Web Personalization for recommendation generation. There are mainly five phases for mining Web usage data. The first phase is data collection, where datasets will be collected from different sources. The second phase is data preprocessing, the collected dataset will be cleaned and presented in desirable format. The preprocessed data will be put into modelling phase for pattern discovery, where we applied Markov model and association rule. There is a model optimization phase after that to filter out rules and states for space optimization and accuracy optimization. Finally the modelled pattern and profiles will be stored into database system for retrieval of the recommendation engine.
4.1.1 Data Collection

The first step of a Web Usage Mining is Data Collection. The data collected are including usage web logs that comprised of a collection of user requests. Web server logs records the remote user’s host name or IP address, the time when the request arrived, the HTTP method (GET, POST, etc.) user used, the URL of the visited document, the status code of HTTP response (404, 200, etc.) and the number of bytes returned to the user. The following is an example of Web server logs.

<table>
<thead>
<tr>
<th>#</th>
<th>IP Address</th>
<th>Time</th>
<th>Method/URL/Protocol</th>
<th>Status</th>
<th>Size</th>
<th>Referrer</th>
<th>Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>210.118.78.9</td>
<td>[05/03/2006:03:04:41]</td>
<td>&quot;GET index.html HTTP/1.0&quot;</td>
<td>200</td>
<td>3290</td>
<td>-</td>
<td>Firefox 1.5.0 (Win XP)</td>
</tr>
<tr>
<td>2</td>
<td>210.118.78.9</td>
<td>[05/03/2006:03:05:34]</td>
<td>&quot;GET aboutus.html HTTP/1.0&quot;</td>
<td>200</td>
<td>2050</td>
<td>index.html</td>
<td>Firefox 1.5.0 (Win XP)</td>
</tr>
<tr>
<td>3</td>
<td>210.118.78.9</td>
<td>[05/03/2006:03:06:02]</td>
<td>&quot;GET contactus.html HTTP/1.0&quot;</td>
<td>200</td>
<td>4130</td>
<td>aboutus.html</td>
<td>Firefox 1.5.0 (Win XP)</td>
</tr>
</tbody>
</table>

The web log data contains all the necessary information to form navigation sequences. It could be the record of traffic of a web site in a period of time, recording large number of user sessions. Two important characteristics of a good dataset are: session collected must be large (i.e. over 10,000) and long (i.e. most session have over 3 navigation path) enough in order to provide adequate support on pattern discovery; there should not be too many web pages inside the web log (i.e. <1000, or <500). Recall the total number Markov states of Kth order, for P web pages is $\Theta(P^K)$, having a large $P$ will cause high model complexity and low prediction accuracy. For those e-commerce web sites that contain over millions of web pages, we suggest breaking the web pages into sub systems and have separate web log collections. It is also suggested to periodically add the new collected usage log into the Web Usage Mining modelling process in order to update the model items.
In the experiments of this project, we conduct the experiments using two of the collected real web usage log datasets from real web site. The first one is the msnbc.com anonymous web data. It contains 989,807 sessions of 17 categories of web pages visiting history, recorded in 28th, September 1999. The second one is the www.microsoft.com usage data of Vroots areas. It contains 42,711 sessions of the browsing history of 275 web pages, recorded in one-week timeframe in Faburary 1998. Part of these historical web pages can be found in www.archive.org. The following two tables describe the information regarding these two datasets.

### Datasets 1

<table>
<thead>
<tr>
<th>Name</th>
<th>msnbc.com anonymous web data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Discrete sequence</td>
</tr>
<tr>
<td>Description</td>
<td>The data describe the page visits of users who visited msnbc.com on September 28, 1999. Visits are recorded at the level of URL category and are recorded in time order.</td>
</tr>
<tr>
<td>Sources</td>
<td>David Heckerman (<a href="mailto:heckerma@microsoft.com">heckerma@microsoft.com</a>)</td>
</tr>
</tbody>
</table>

**Example**
```
% Different categories found in input file:
{crafts, page, news, tech, local, opinion, on-six, wise, weather, news, movie, health, living, business, win-sports, sports summary, bio, travel}
% Sequences:
1
1
2 2 4 2 2 3 3
5
1
4
1
4 4 4 4 4 4 4 4 10 3 10 5 10 4 4 4
1 1 1 1 1 1 11 12 12
1
```

**Past Usage**


### Datasets 2

<table>
<thead>
<tr>
<th>Name</th>
<th><a href="http://www.microsoft.com">www.microsoft.com</a> anonymous web data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Relational, multivariate</td>
</tr>
<tr>
<td>Description</td>
<td>The dataset records which areas (Vroots) of <a href="http://www.microsoft.com">www.microsoft.com</a> each user visited in a one-week timeframe in Faburary 1998.</td>
</tr>
<tr>
<td>Sources</td>
<td>Jack S. Breese, David Heckerman, Carl M. Kadie</td>
</tr>
</tbody>
</table>

37
4.1.2 Data Preprocessing

The data collected needs to be preprocessed before putting into the modeller. The dataset and web log collected are in different kinds of format. There are many information stored together with the web log or datasets that are not useful to the modeller, for example, HTTP status and Browser information. They must be cleaned and reformatted in order to fit in our modelling program.

To clean and identify users and sessions from the row web logs, Mobasher and Spiliopoulou 2002 have proposed a preprocessing process, suggesting the steps of Data Cleaning (eliminate information), Data Integration (synchronize multiple logs), Data Transformation (users, sessions, pageview identification) and Data Reduction (ignoring items). They have explained the identifications and process in detail. The
row web log can also be cleaned using WebSIFT system (Cooley et al. 1999) or Web Utilization Miner (Spiliopoulou and Faulstich 1998).

For other datasets that have already identified session and pageview information, we write programs for each individual dataset preprocessing in order to reformat the dataset.

After session identification, we reformat the clickstreams into a sequence of number. We then divide these clickstreams into 3 files, one for training purpose, one for validation, and one for testing. A category file representing the mapping of the page name, page path against these numbers shall be created in the same time. These files will be used as input of the modeller program.

Example clickstream:

```
1008 1051 1038 1031 1052 1053 1018
```

Example page map items:

```
1057  MS PowerPoint News,/powerpoint  http://www.microsoft.com/powerpoint/
1031  MS Office,/msoffice         http://office.microsoft.com
```

<table>
<thead>
<tr>
<th>Input of this step:</th>
<th>Web logs, Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output of this step:</td>
<td>Clickstream files</td>
</tr>
<tr>
<td></td>
<td>(training set, validation set, testing set)</td>
</tr>
<tr>
<td></td>
<td>Category file</td>
</tr>
<tr>
<td></td>
<td>(page ID, page name, page path)</td>
</tr>
</tbody>
</table>
4.1.3 Pattern Discovery and Modeling

The pattern discovery phase concerns about discovery and modelling the clickstream sequences identified from the preprocessing step. The purpose of this step is to extract patterns and rules from the users’ usage in order to dynamically construct user profiles for recommendation preparation. The training dataset that previously generates will be used in this phase. During this step, Selective Markov Model and Association Rule Mining will be used. Both Markov states and association rules will be generated.

The output of this step will be the Markov model that consists of different Kth order states, and association rules with support and confidence computed.

**Input of this step:** Training dataset

**Output of this step:** All Kth Order Markov States, Association Rules

4.1.4 Model Optimization

After the pattern discovery phase, we will have a number of states and rules. They are then put into different optimization schemes in order to prune out unreliable and high error rate items. The pruning schemes for Markov models are FPMM, CPMM and EPMM. The pruning schemes for association rules are Item Size pruning, Minimum Support pruning and Error pruning. The validation dataset will be used to compute the error rate of Markov states and association rules.
During the model optimization phase, different kind of tests will be conducted in order to test out the optimization effects. The testing dataset will be used for this testing. We break each clickstream in the testing set into various test cases and the testing result will be compared with actual result in the clickstream. For example, \{1008 1051 1038 1031 1052 1053 1018\} will generate 4 test cases as the following table:

<table>
<thead>
<tr>
<th>Input</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1008 1051 1038</td>
<td>1031</td>
</tr>
<tr>
<td>1051 1038 1031</td>
<td>1052</td>
</tr>
<tr>
<td>1038 1031 1052</td>
<td>1053</td>
</tr>
<tr>
<td>1031 1052 1053</td>
<td>1018</td>
</tr>
</tbody>
</table>

<Table 4.4. Test cases>

The optimization parameters and value assigned will be investigated through different testing in order to find the parameter value that best suit the specific dataset.

**Input of this step: All Kth Order Markov States, Association Rules, Validation dataset, Testing dataset**

**Output of this step: Optimized Markov States and Rules**

### 4.1.5 Recommendations Generation

After the model optimization, the set of user profiles (states and rules) are ready for the recommendation generation. The states and rules will be stored into database so that the server side recommendation engine can use the data to generate recommendations.
Each time user click a link, the server will add the clicked page id into the cookies to form a clickstream sequence. The sequence will be analyzed immediately to look for a similar profile in the database. Then based on the profiling model (i.e. SMM or AR), the recommendations are proposed. The links of the recommendations are immediately displayed upon user enters the page. We set the maximum recommendations to be 5.

Different prediction choices can be set for generating recommendations. For SMM, we can choose to predict the next pages using all Kth order Markov models, or exact Kth order Markov models; choose recommendation by favoring error rate of outgoing pages, or choose recommendation by favoring outgoing page frequency; predict the next page user may visit, or predict the page after next page user may visit. For AR, we can choose to generate the recommendations by favoring support value, validation result, half support half validation result, or confidence value. We can also set the window size (clickstream item size) for input data. We can choose to ignore rules that contain the page which is already visited, or consider the rules that contains pages already visited. Finally, just like the SMM, we can also choose to predict the page after the current page, or page after the next page.

Input of this step: Markov States and Rules

Output of this step: Page recommendations
4.2 System Architecture

Two main systems are developed for the process of Modeling and Recommending. The systems are SMMAR Profile Modeler and Recommendation Engine. The following DFD shows the level 0 data flow on the system.

The SMMAR Profile Modeler system is written in C++ using .NET framework for interface. It includes the modeling process and optimization process. It has the following structure:
testing and combined testing function, together with a result viewer. The DB Updater helps update the profiles into database. DB connection string can be specified.

<Figure 4.7 Low level structure of Testing and DB Updater modules>

<Figure 4.8 Screen capture of Combined Testing Interface>

The above graph show the combined test interface for SMMAR. We can input a testing file to run prediction testing. The result can be view by clicking the result list in the right hand side. The SMM and AR prediction choices can be selected below.
The individual testing module is designed for single session testing. We can load test file that contains user sessions. The sessions will be loaded in the left bottom. We can click each session in the list and it will simulate the user clicking step by step.

The DBUpdater helps update the profiling information into the database. As there are different datasets, we have different prefix tables to update. We can set this kind of information in the database setting.
4.2.1 Profiling Modeler

The Modeler module contains SMM modeler and AR modeler. Both modelers contain the function of Training, Pruning, Output File, Testing.

For the SMM modeler, there will be pruning schemes of FPMM, CPMM and EPMM. For the AR modeler, there will have Fine Tuning, Error Pruning.
In the SMM modeler, we have several functions. We can input training file to train up the model, then run different pruning scheme. Before we finalize our states, we can output the states into file and we can read a state file previously generated. We can run testing for SMM prediction and select different kind of prediction methods.

![Figure 4.13 Screen capture of AR modeler module](image)

The association rule modeler has training part, pruning part and testing part. We can choose a initial support % and generate the rules. The rules will be displayed in the middle. In the pruning part, we can input validation file to validate the rules and use different different pruning scheme for eliminating unnecessary rules.
4.2.2 Recommendation Engine

The server for recommendation engine testing uses the dataset of MSWEB for construction. Although there are only page ID and page names in the MSWEB dataset, the site structure can still be resolved from the usage log. We have resolved all the outgoing page lists for the web pages in MSWEB dataset and write a program to simulate the transitions from page to page.

This server is written in C# ASP.NET and using MS SQL as database tier. It simulates the transitions of 274 web pages of www.microsoft.com in 1998 February. It will record the user clickstreams and use the record to generate real-time recommendations.

Although the links in 2006’s “Products” page is quite different from the links in 1998’s “Products” page, most of the links in the 1998’s “Products” page can still be found in the 2006’s “Products” page.

Each of the links in the simulation page contains 3 items: the page name of the link, the page ID of the link, and the real microsoft.com page link (either a link to the current page, or the link to the archive.org repository). As we can see from the screenshot, there are many links in one page. In the real situation, some are revealed clearly, some are hidden in layers that without opening the layer, it cannot be found (e.g. the “Quick Links” layer in 2006’s page). A user may need to spend much time on screening and searching the link items in the page in order to find a link that may suit his need. If the system can give recommendations to the user instantly and in eye
catching area (e.g. in a popup area), it will save the effort for the user to search and screening the existing links.

**4.3 Algorithm**

**4.3.1 Algorithm for Kth Order Markov Models**

Traditional algorithms used for Hidden Markov Chain are Forward algorithm and Viterbi algorithm. We adapt the basic concept of and have the following algorithm for constructing our Kth order Markov model (described in pseudocode):

![Pseudocode describing the construction initial Markov States](image)

\[
\text{For each Session } i \\
\text{Array SessionItems = Session}[i] \\
\text{For each Item } j \text{ in SessionItems} \\
\quad \text{//Find and update State in map} \\
\quad \text{//If state doesnt exist, add state} \\
\quad \text{//If state exist, update frequency,} \\
\quad \text{//incoming page, and outgoing page frequency} \\
\quad \text{State = Empty;} \\
\quad \text{For } k=0, k<\text{window size and } j+k < \text{SessionItems size} \\
\quad \quad \text{State += SessionItems}[j+k] \quad \text{//Kth Order State} \\
\quad \quad \text{If } j==0 \\
\quad \quad \quad \text{IncommingPage = NULL} \\
\quad \quad \text{Else} \\
\quad \quad \quad \text{IncommingPage = SessionItems}[j-1] \\
\quad \quad \text{End if} \\
\quad \quad \text{OutgoingPage = SessionItem}[j+k+1] \\
\quad \quad \text{FindUpdateState(State, IncommingPage, IncomingPage)} \\
\quad \text{End} \\
\text{End} \\
\text{For each state} \\
\text{Compute Probabilities for Each Outgoing page} \\
\text{End}
\]
4.3.2 Apriori Algorithm

Popular algorithms used for association rule mining are Apriori, Charm, FP-growth, Closet and MagnumOpus. In this project we adapt Apriori algorithm proposed by Srikant and Agrawal 1997. The implementation is based on the Apriori algorithm implementation version of Boden 2003, which uses a tier (prefix-tree) to store the itemset in stead of a hash-tree. It is claim to be the faster version of Apriori. In deciding which k+1 candidate itemsets are supported, it has used a Tier in stead of a hash-tree.

### Apriori by Boden 2003

```plaintext
candidate_size=1
support_of_items = new Vector of Integer
basket_number = Find frequent items( min_supp, support_of_items )
temp_set = new Set of item type

For index = 0 to support_of_items's size
    Add index to temp_set
    Output basket and counter( temp_set, support_of_items[index] )
    Erase temp_set's begining
End

apriori_trie = new Apriori_Trie( basket_number )
insert frequent items( support_of_items ) into apriori_trie
min_supp_abs = min_supp * (basket_number - 0.5)
candidate_size++;

Generate candidate (candidate_size-1) in apriori_trie

While apriori_trie has candidate
    support( candidate_size )
    Delete infrequent(min_supp_abs,candidate_size) from apriori_trie
    If candidate_size = size_threshold
        break
    End if
    candidate_size++
    Generation candidate(candidate_size-1) in apriori_trie
End While
End Algorithm
```

<Figure 4.17 Pseudocode for Apriori algorithm>
5. Experiment Result and Discussion

Experiments are conducted to test the model performance against different pruning schemes, model performance on different aspects. From the experiments, we found some of the pruning schemes return satisfactory results and some do not. The SMM prediction is also compared with AR. Finally, we include some results on other related experiments conducted.

5.1 Recommendation Result without any pruning

The follow graphs show the SMM recommendation results of MSWEB and MSNBC dataset before any pruning scheme.

<Figure 5.1 MSWEB Dataset: Plot comparing prediction accuracy of All Kth Order Markov Prediction and Exact Kth Order Markov PredictionBefore any Pruning Scheme>

The above figure shows that the result of all Kth order Markov Prediction and Exact Kth Order Markov predictions for MSWEB dataset. It shows that the
prediction accuracies of both prediction methods are the same for the MSWEB dataset.

![MSNBC Dataset](image)

The above figure shows that the result of all Kth order Markov Prediction and Exact Kth Order Markov predictions for MSNBC dataset. The result is different from MSWEB dataset. The prediction accuracies of Exact Kth order Markov model is higher than all kth order Markov model.

### 5.2 Recommendation Result after Pruning

The following results show the effect of applying different pruning schemes to the SMM and AR models.
5.2.1 SMM EPMM

![MSWEB Dataset After EPMM](image)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MSWEB</th>
<th>Training Session</th>
<th>44,402</th>
<th>Test Cases</th>
<th>20,130</th>
</tr>
</thead>
<tbody>
<tr>
<td>No of States</td>
<td>18,655</td>
<td>Testing Session</td>
<td>6,590</td>
<td>Coverage</td>
<td>0.8167</td>
</tr>
</tbody>
</table>

<Figure 5.3 MSWEB Dataset: Plot comparing prediction accuracy of Exact Kth Order Markov Prediction before EPMM and After EPMM>

<table>
<thead>
<tr>
<th># of Recommendations</th>
<th>No EPMM</th>
<th>EPMM</th>
<th>% Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6325</td>
<td>0.7251</td>
<td>14.6388</td>
</tr>
<tr>
<td>2</td>
<td>0.7968</td>
<td>0.8760</td>
<td>9.9421</td>
</tr>
<tr>
<td>3</td>
<td>0.8725</td>
<td>0.9264</td>
<td>6.1746</td>
</tr>
<tr>
<td>4</td>
<td>0.9149</td>
<td>0.9543</td>
<td>4.3018</td>
</tr>
<tr>
<td>5</td>
<td>0.9399</td>
<td>0.9693</td>
<td>3.1219</td>
</tr>
<tr>
<td>6</td>
<td>0.9567</td>
<td>0.9787</td>
<td>2.2964</td>
</tr>
<tr>
<td>7</td>
<td>0.9670</td>
<td>0.9844</td>
<td>1.7946</td>
</tr>
<tr>
<td>8</td>
<td>0.9737</td>
<td>0.9880</td>
<td>1.4620</td>
</tr>
<tr>
<td>9</td>
<td>0.9786</td>
<td>0.9904</td>
<td>1.2008</td>
</tr>
<tr>
<td>10</td>
<td>0.9825</td>
<td>0.9922</td>
<td>0.9874</td>
</tr>
</tbody>
</table>

<Table 5.1 MSWEB Dataset: Percentage of increase after applying EPMM>

After applying EPMM, we can see a clear increase on accuracy. The accuracy for one recommendation has the highest increase rate. After that it increases slowly.
5.2.2 SMM CPMM

![Graph comparing prediction accuracy of Exact Kth Order Markov Prediction before CPMM and After CPMM]

<table>
<thead>
<tr>
<th># of States</th>
<th>Coverage</th>
<th>Dataset</th>
<th>Training Session</th>
<th>Testing Session</th>
<th>Test Cases</th>
<th>Prune After EPMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before CPMM</td>
<td>19,762</td>
<td>MSWEB</td>
<td>44,402</td>
<td>6,590</td>
<td>20,130</td>
<td>TRUE</td>
</tr>
<tr>
<td>CPMM 0.93</td>
<td>18,612</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPMM 1.03</td>
<td>18,558</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPMM 1.15</td>
<td>18,430</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>%Increase</th>
<th>15.3834</th>
<th>0.7155</th>
<th>0.7275</th>
<th>0.7298</th>
<th>0.7298</th>
<th>15.3834</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.7298</td>
<td>0.965</td>
<td>0.9699</td>
<td>0.9705</td>
<td>0.9721</td>
<td>0.7275</td>
</tr>
<tr>
<td></td>
<td>0.9721</td>
<td>0.991</td>
<td>0.9921</td>
<td></td>
<td></td>
<td>0.7155</td>
</tr>
</tbody>
</table>

The CPMM that applied after EPMM shows slight increases for each confidence parameter increase. It is shown in the table that the CPMM do contribute to a more reliable state storage.
5.2.3 SMM FPMM

The above figure shows the pruning of frequency parameter 1 on MSWEB dataset have a slight increase in accuracy. The result also shows that after frequency 2 prune, the accuracy drops. This is due to the reason that the MSWEB has more pages such that the average frequency for each state is low. Therefore it is sensitive to the frequency pruning such that a even small frequency pruning will eliminate large number of states. From this example we can see the important of using large number of usage sessions as training.

To further elaborate the finding, we have run the FPMM testing for dataset MSNBC. The MSNBC contains 17 categories of pages with 989,807 training sessions. The following graph shows the benefit of having a large training set.
As we can see from the experiment result, the MSNBC dataset is very suitable for FPMM pruning. Result shows the continue increase after pruning 84.6% of states, while coverage of the model still maintain at above 94%.
5.2.4 AR Validation Prune

![Graph showing MSWEB Dataset results of accuracy and rule size after pruning different error rates.]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MSWEB</th>
<th>Training Session</th>
<th>34,232</th>
<th>Test Cases</th>
<th>13,524</th>
<th>Ignore Visited</th>
<th>TRUE</th>
<th>Favor</th>
<th>Support</th>
<th>Testing Session</th>
<th>22,999</th>
<th>Coverage</th>
<th>1</th>
<th>Window Size</th>
<th>5</th>
</tr>
</thead>
</table>

<Figure 5.8 MSWEB Dataset: Results of accuracy and rule size after pruning different error rates>

<table>
<thead>
<tr>
<th>Error Rate</th>
<th>Accuracy</th>
<th># of Rules</th>
<th>% Increase</th>
<th>% Prune</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before Prune</td>
<td>0.55</td>
<td>2621</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Err&lt;1</td>
<td>0.61</td>
<td>1895</td>
<td>11.8779</td>
<td>27.6994</td>
</tr>
<tr>
<td>Err&lt;0.97</td>
<td>0.62</td>
<td>1786</td>
<td>11.9720</td>
<td>31.8581</td>
</tr>
<tr>
<td>Err&lt;0.95</td>
<td>0.66</td>
<td>1697</td>
<td>19.4108</td>
<td>35.2537</td>
</tr>
<tr>
<td>Err&lt;0.94</td>
<td>0.65</td>
<td>1669</td>
<td>18.1195</td>
<td>36.3220</td>
</tr>
</tbody>
</table>

<Table 5.9 MSWEB Dataset: Results of accuracy and rule size after pruning different error rates>

The graph above show the effect of pruning high error rate rules from the AR model. The result shows that the pruning brought an increase on prediction while the coverage remains 100%. For the MSWEB dataset, the maximum error prune parameter we can choose is 0.95 because after that the accuracy starts to fall.
5.2.5 AR Fine Tuning

5.2.5.1 Item Size Prune

The above experiment results show that for item 5-3 pruning, the accuracy and coverage remain unchanged, after applying rules size prune, the prediction result accuracy increased, while coverage remains 100%.
5.2.5.2 Support Prune

When we first construct the association rules from the training files, we select a support frequency or rate for generation. Such a rate may not reflect the actual level of rule that have high accuracy and most frequently used. We can apply the support pruning scheme to prune out rules that are not necessary after the construction.

From the experiment result, we can see that for the MSWEB dataset, the actual rules used for prediction are rules having support of over 3%. From the chart we can also see that over 90% of unnecessary rules were pruned that does not reduce the prediction accuracy or coverage. Therefore, this pruning scheme not only can help reducing the rule size but also help us reveal the actual rules used and the initial support we should choose.
5.2.5.3 Confidence Prune

![Graph showing MSWEB Dataset](image)

Similar to the Support Pruning, at the initial rule generation phase, we may not correctly choose the best suitable confidence parameter for the dataset. This pruning scheme can help us further adjust the parameter after rule construction.

The experiment result show that there are increases after increase the confidence parameter.
5.3 Result of Combined Testing

### MSWEB Dataset

<table>
<thead>
<tr>
<th>Recommendation Method</th>
<th>No. of Test Case</th>
<th>Covered Cases</th>
<th>Correct Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM</td>
<td>15869</td>
<td>15744</td>
<td></td>
</tr>
<tr>
<td>AR</td>
<td>10788</td>
<td>20123</td>
<td></td>
</tr>
<tr>
<td>SMMAR</td>
<td>17509</td>
<td>20130</td>
<td></td>
</tr>
</tbody>
</table>

<Figure 5.13 MSWEB Dataset: SMM, AR, SMMAR result>

### MSNBC Dataset

<table>
<thead>
<tr>
<th>Recommendation Method</th>
<th>No. of Test Case</th>
<th>Covered Cases</th>
<th>Correct Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM</td>
<td>2505</td>
<td>2331</td>
<td></td>
</tr>
<tr>
<td>AR</td>
<td>3912</td>
<td>1405</td>
<td></td>
</tr>
<tr>
<td>SMMAR</td>
<td>4190</td>
<td>4190</td>
<td>3181</td>
</tr>
</tbody>
</table>

<Figure 5.14 MSNBC Dataset: SMM, AR, SMMAR result>

Experiment results shows that coverage increased to 100% after combination testing of SMMAR. The correct recommendations are also increased for both dataset.
As we can observe from the experiments, AR recommendation results are usually lower than SMM recommendation. However, AR generally has a higher coverage than SMM. From the experiment result, we can see that AR is a good compliment for SMM.

5.4 Recommendation Engine

Recommendations for Windows NT Server Page

<Figure 5.16 MSWEB Dataset: Screen capture for Windows NT Server simulation page and the real Windows NT server page>
Results show that the top 1 recommendation is the most relevant to the Windows NT server page. Other recommendations are also relevant to the Windows NT Server.

Note that the results are a reflection of past user visits and reflect the majority’s selection of the next pages.
Recommendations for MS Office Info >> MS Word News

Figure 5.18 MSWEB Dataset: Recommendations for MS Word News Page

Figure 5.19 MSWEB Dataset: Recommendations for MS Word News Page
The experiment shows the result of clickstream sequence of *MS Office Info >> MS Word News*. Results show that the top recommendations are most relevant to the navigation sequence.

### 5.5 Other Experiment Results

#### 5.5.1 Predict By Lower Error Rate

![MSWEB Dataset](image)

<Figure 5.20 MSWEB Dataset: Compare prediction result using low error rate information with high frequency>

The purpose of this experiment is to find out whether the error rate computed for the top N most frequent outgoing pages of a state can be used to predict result.

By computing the error rate of top N most frequent outgoing pages of a state, we can have one more parameter to be considered when recommending items.
Discarding the outgoing pages that are not in top N can also save the model space. In many cases a state will have many outgoing pages stored in model.

The experiment compares the prediction of original nude model without any pruning, together with the EPMM pruned model which favor high frequency rate of outgoing pages for the prediction, and the EPMM pruned model which favor low error rate outgoing page for the prediction.

Results show that favor low error gives a similar result of favoring high frequency rate. For recommendation item size 1 and 5, the later one even has higher accuracy.

### 5.5.2 Predict the Page After Next Page

![MSWEB Dataset: Predict the page after the next page](image)

This experiment predicts the page after the next page. It seems that the study of the prediction of pages after the next page is a future trend in the Navigation Path prediction. However, in this project we do not target on this kind of prediction study.
5.6 Performance

<table>
<thead>
<tr>
<th>Activity</th>
<th>File</th>
<th>Generation</th>
<th>Sessions</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMM</td>
<td>Training</td>
<td>msnbctest.train</td>
<td>1929 states</td>
<td>4983</td>
</tr>
<tr>
<td>SMM</td>
<td>Testing</td>
<td>msnbctest.test</td>
<td>4190 testcase, result file</td>
<td>1000</td>
</tr>
<tr>
<td>AR</td>
<td>Training</td>
<td>msnbctest.train</td>
<td>7460 rules</td>
<td>4983</td>
</tr>
<tr>
<td>AR</td>
<td>Testing</td>
<td>msnbctest.test</td>
<td>2527 testcase, result file</td>
<td>1000</td>
</tr>
</tbody>
</table>

<Table 5.22 MSWEB Dataset: Performance>

**Online Recommender Engine**: Return recommendation in average of 0.1482 second.
6. Conclusion and Future Work

6.1 Conclusion

Web personalization is a kind of technology to solve the World Wide Web information overloading problem. A recommender system is one kind of Web personalization. In this project, we have studied various kinds of navigation path analysis methodologies for recommendation generation. We have chosen Selective Markov Models and Association Rule Mining as the modeler. We also studied and experimented on different model optimization schemes in order to improve the recommendation accuracy. We have proposed a combination of Exact Kth Order Markov Model with Association Rule. We have also proposed three pruning scheme for Association Rules. For the model optimization schemes, we conclude that the Error Pruning scheme for both Markov model and Association Rule is the most effective scheme. Finally, we have implemented a SMMAR Profiling Model for navigation path analysis, profile generation, and recommendation performance optimization. We have also implemented an online recommendation server to simulate the Microsoft.com web page from MSWEB dataset. Our experiments show the evidence of performance improvement on recommendation quality.
6.2 Future Work

The studies of the prediction on page after the next possible page will be the further research concern. For a recommendation system, sometimes it will be more valuable to predict one or more page ahead. The Markov model has the characteristic for forwarding the prediction, but the accuracy shall be further improved. Moreover, in future work, more site knowledge such as site structure, web content, web page category can be collected make use to a recommendation system. The approaches of combining different framework can also be considered.
7. References


the Web Personalization process. Proceedings of the 9th ACM SIGKDD International Conference on

Markov models. Proceedings of the 7th Annual ACM International Workshop on Web Information and Data


Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work. New York, N.Y.:

Press, 5-53.

Community of Use. Proceedings of ACM CHI’95 Conference on human factors in computing systems. Denver,
CO., 194-201.

problem in collaborative filtering. ACM Transactions on Information Systems (TOIS) 22 (1). New York, N.Y.:
ACM Press, 116-142.

Proceedings of the 5th ACM International Workshop on Web Information and Data Management. New York,
N.Y.: ACM Press, 82-89.

Record 35 (1). New York, N.Y.: ACM Press, 14-19


## Appendix I: Monthly Log

| OCT       | 1. Collected datasets for testing  
|           | 2. Studied and implemented parts of Selective Markov Model 
|           | 3. Prepared and submitted Project Plan  
|           | 4. Studied web personalization history  
|           | 5. Tested on MSNBC dataset  
| NOV       | 1. Cleaned msweb dataset  
|           | 2. Tested on msweb dataset  
|           | 3. Studied and experiment on FPMM and CPMM for both msweb and msnbc dataset  
|           | 4. Reconstructed classes and data structure of experiment programs  
|           | 5. Prepared Interim Report  
|           | 6. Submitted Interim Report  
| DEC       | 1. Search for new dataset  
|           | 2. Evaluating experiment program for structure, efficiency, and correctness  
|           | 3. Processing and cleaning musicmachine dataset  
|           | 4. Studying Hypertext Probabilistic Grammars (HPG) model  
|           | 5. Studying traditional HMM  
| JAN       | 1. Cleaned musicmachine dataset  
|           | 2. Tested on musicmachine dataset  
|           | 3. Studied Markov related history  
|           | 4. Studying Forward Algorithm and Viterbi Algorithm  
|           | 5. Refined SMM program  
|           | 6. Implemented Error Prune Markov Model  
| FEB       | 1. Constructed SMM interface using .NET framework  
|           | 2. Constructed MSSQL Server for states data  
|           | 3. Constructed database systems for SMM  
|           | 4. Added functions for SMM database connection  
|           | 5. Studied FP-Growth algorithm and Apriori algorithm  
|           | 6. Implementing Association Rule Mining  
|           | 7. Evaluating SMM program  
|           | 8. Testing performance  
|           | 9. Testing with different datasets  
|           | 10. Combined SMM and AR testing  
| MAR       | 1. Implemented C++.NET version SMMAR  
|           | 2. Integrated Association Rule Miner into SMMAR  
|           | 3. Implemented Top N outgoing page validation  
|           | 4. Implemented SMM Error rate recommendation scheme  
|           | 5. Implemented AR validation scheme  
|           | 6. Implemented AR support pruning scheme  
|           | 7. Implemented AR error pruning scheme  
|           | 8. Implemented AR item size pruning scheme  
|           | 9. Implemented Individual testing  
|           | 10. Refined Database Updater  
|           | 11. Implemented C# Recommendation Web Server  

75