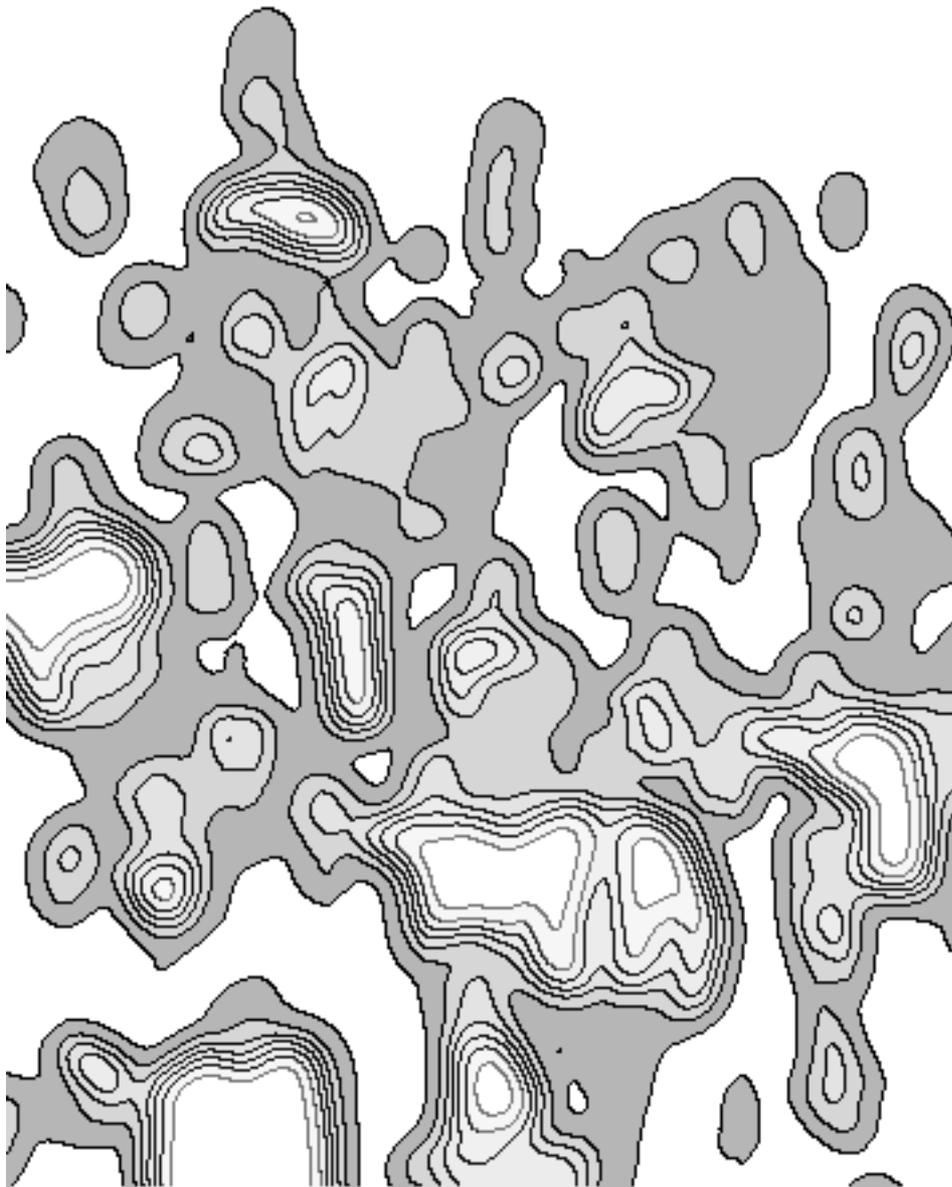


Papers on the Development of a Hybrid Approach to Web Usage Mining

Master Thesis

submitted by

Søren Enemærke Jespersen & Jesper Thorhauge
Department of Computer Science, Aalborg University 2002



TITLE:

Papers on the Development of a
Hybrid Approach to Web Usage
Mining

PROJECT PERIOD:

DAT6 - Master Thesis,
February 1th 2002 -
June 14th 2002

AUTHORS:

Søren Enemærke Jespersen
Jesper Thorhauge

SUPERVISOR:

Torben Bach Pedersen

COPIES: 8

PAPERS: 26 / 27 / 30 pages

APPENDICES: 0 / 12 / 6 pages

TOTAL: 103 pages

Summary

This master thesis is focused on the area of web usage mining, and in particular on a novel approach named the *hybrid approach*. The hybrid approach combines a compact, aggregated structure named the Hypertext Probabilistic Grammar (HPG) and a fine-grained data warehouse schema named the click fact schema to offer a flexible tool for performing constraint-based knowledge extraction on usage data from a web server. The hybrid approach uses the HPG model, consisting of a number of states and productions, to generate rules representing the most preferred user trails on the web site. A specialized HPG, representing a subset of the collection of sessions from the usage data is built through queries on the information in the click fact schema and the specialized HPG can readily be mined for the most preferred trails of the sessions represented in the HPG. The HPG model inherently assumes independency of states in the grammar and therefore knowledge extracted can be unsupported in the usage data.

The master thesis is presented as three separate papers. Each paper is focused on a separate topic relating to web usage mining and the hybrid approach and is presented as an individual article including separate abstracts, reference sections and page numbering.

Paper 1 presents a case study where the hybrid approach is applied. Zenaria A/S, an e-learning company developing interactive stories, desire to move their distribution from CD-ROMs to an Internet setting. This transition requires that user interaction with the website is tracked in order for Zenaria to evaluate users of the dedicated stories. Existing web usage mining tools is not able to meet all of the demands of Zenaria and a number of concrete requirements of a new framework is defined by Zenaria and presented. The technologies underlying the hybrid approach is briefly described and the hybrid approach is explained. A dedicated framework called MIMER is developed which is able to meet the requirements of Zenaria and the principle design of the user interface to MIMER is presented.

Paper 2 presents an expansion of the hybrid approach. As the hybrid approach utilizes the HPG model to extract knowledge, the hybrid approach inherits the weakness of extracting knowledge which can be unsupported in the usage data from the web server. The paper describes the expansion which involves considering the rules extracted from the HPG as candidate rules which are validated, or post-checked, against the usage data stored in the data warehouse. The novel approach is named the Post-check Expanded Hybrid Approach (PEHA). The expansion is implemented and experiments

using materialized views indicate that rules can be post-checked in near constant time regardless of the length and number of rules. Compared to queries directly on a rival database schema, PEHA arguably performs comparable when considering the flexibility, running time and storage requirements.

Paper 3 presents an investigation of the quality of the HPG model. The HPG model can be extended to increase the precision of the extracted knowledge by introducing additional states and productions in the grammar. The paper defines a measures of similarity and accuracy between the rules extracted from the HPG model and the rules found in the usage data. These measures are used to investigate the quality of the knowledge extracted from the HPG model. Experiments are conducted using data from two websites with different usage patterns and the experiments indicate that the HPG extracts rules that have a relative high quality, however, as the size of the rules extracted grow compared to the precision modeled in the HPG, the quality of the rules drop.

Resume

Dette speciale er fokuseret på området web usage mining og mere specifikt på en ny tilgang til web usage mining kaldet den hybride metode (the hybrid approach). Tilgangen kombinerer en kompakt, aggregeret struktur, kaldet en Hypertext Probabilistic Grammar (HPG), og et fin-granuleret data warehouse skema, kaldet et click fact skema, for at tilbyde et fleksibelt værktøj til at udføre vidensudtræk med begrænsninger på brugsdata fra en web server. Metoden benytter en HPG model, bestående af et antal tilstand og et antal transitioner, til at generere regler som repræsenterer de mest foretrukne gennemgange på det pågældende website. En specialiseret HPG, repræsenterende en delmængde af samlingen af bruger-gennemgange i brugsdataene, bliver konstrueret via forespørgsler på informationen der er gemt i click fact skemaet og den specialiserede HPG kan umiddelbart benyttes til at udtrække de mest foretrukne gennemgange for netop de bruger-gennemgange repræsenteret i den specialiserede HPG. HPG modellen har i sin natur en antagelse om uafhængighed mellem tilstande i grammatikken og derfor kan det udtrukne viden fra en HPG være ukorrekt i forhold til det brugsdata den repræsenterer.

Specialet indeholder tre separate artikler. Hver artikel er fokuseret på et specifikt emne i forhold til web usage mining og den hybride metode og hver artikel indeholder en synopsis, reference sektion og individuel side nummerering.

Artikel 1 præsenterer et case-study hvor den hybride metode benyttes. Zenaria A/S, en e-learning virksomhed der udvikler interaktive historier, ønsker at flytte fra et CD-ROM baseret til en Internet baseret distributionen. Denne transition påkræver at bruger interaktion med historien på webserveren bliver overvåget for at Zenaria kan evaluere brugeres benyttelse af de dedikerede historier. Eksisterende web usage mining værktøjer er ikke i stand til at opfylde Zenarias ønsker og et antal konkrete krav fra Zenaria til et værktøj bliver defineret og fremstillet i artiklen. Teknologierne indeholdt i den hybride metode beskrives kort og ideen bag metoden er fremstillet. Et dedikeret værktøj kaldet MIMER er udviklet som er i stand til at opfylde kravene fra Zenaria og princip skitser for et bruger interface bliver præsenteret.

Artikel 2 præsenterer en udvidelse til den hybride metode. Eftersom den hybride metode benytter HPG modellen til at udtrække viden, arver den hybride metode den svaghed at udtrukket viden kan være ukorrekt i forhold til brugsdataene fra webserveren. Artiklen beskriver en udvidelse som involverer

at betragte regler udtrukket fra en HPG som kandidater der skal valideres (post-checkes) imod det brugsdata der bliver gemt i data warehouset. Denne nye metoder bliver kaldt Post-check Expanded Hybrid Approach (PEHA). Udvidelsen er implementeret og en række eksperimenter der benytter materialiserede views indikerer, at regler kan post-checkes i omtrent konstant tid uanset længden og antallet af regler der betragtes. Betragtet i forhold til forespørgsler direkte på et rivaliserende data warehouse schema, argumenteres der i artiklen for at PEHA er en konkurrencedygtig tilgang hvis man betragter den samlede fleksibilitet, køretid og pladsforbrug.

Artikel 3 præsenterer en undersøgelse af kvaliteten af HPG modellen. Præcisionen af HPG modellen kan øges ved at introducere yderligere tilstande og transitioner i grammatikken. Artiklen definerer mål for ensartethed og nøjagtighed mellem regler udtrukket fra HPG modellen og reglerne som findes i brugsdata fra webserveren. Disse mål benyttes til at undersøge den generelle kvalitet af viden udtrukket igennem HPG modellen. Eksperimenter udføres på brugsdata fra to forskellige websites, hver med forskellige brugsmønstre, og eksperimenterne indikerer at regler udtrukket fra en HPG har relativ høj kvalitet men efterhånden som længden af de udtrukne regler vokser i forhold til den benyttede præcision i HPG modellen falder kvaliteten.

Preface

Paper 1:

MIMER: A Web Usage Mining Framework for E-learning.

Paper 2:

PEHA: The Post-check Expanded Hybrid Approach to Web Usage Mining.

Paper 3:

Investigating the Quality of the Hypertext Probabilistic Grammar Model.

Preface

This master thesis is focused on the area of web usage mining, that is, the discovery of interesting usage patterns from a website. More specifically, the thesis is centered around the application and investigation of a novel technique for performing constraint-based knowledge discovery on long sequences of clicks, called the *hybrid approach*.

The thesis is divided into three papers that are to be considered self-containing and each paper is presented as a separate article, including abstract, reference section, appendices and page numbering. As the papers are self-contained, overlapping issues and descriptions will occur. Since the papers all revolve around topics relating to the hybrid approach, we allow for papers to refer directly to other papers within this thesis.

We believe that presenting three separate papers will allow for a more focused, condensed and cohesive presentation of the topics of each paper. If a large single report was to be presented, we would risk introducing information relating to many different aspects and thus cluttering up the individual topics. We believe that by presenting three separate papers, we are able to present the same amount of information as could be done in a larger, single report and furthermore be able to be more focused on the separate topic within each paper.

Paper 1 is focused on presenting a use case in which an e-learning company are to distribute interactive stories utilizing the Internet, including discussions and specifications of the requirements of Zenaria to a framework for storing and extracting usage data on their interactive stories, an overall presentation of the hybrid approach and the presentation of a number of principle ideas on the development of a user interface to suit the specific needs of the company. Paper 2 is focused on the development of an extension to the hybrid approach which will enable the discovery of correctly supported knowledge. It presents the problems inherent in the HPG model and the hybrid approach, describes the post-checking mechanism developed for enabling extraction of correct knowledge in the constraint-based web usage mining process and presents the results of a number of performance experiments conducted on an implementation of the mechanism. Paper 3 is focused on an investigation of the quality of the knowledge extracted from the Hypertext Probabilistic Grammar model utilized in the hybrid approach, more specifically on the inherent assumption of limited browsing history in the model, and the paper will include descriptions of the assumption, define

measures of quality between rules extracted on the HPG and the knowledge in the true traversals and present a number of experiments which attempt to evaluate the quality of the extracted knowledge.

We would like to thank the people and employees at Zenaria A/S for valuable input and patience during the project period.

Søren Enemærke Jespersen

Jesper Thorhauge

Paper 1

MIMER: A Web Usage Mining
Framework for E-learning

MIMER: A Web Usage Mining Framework for E-learning

Søren E. Jespersen Jesper Thorhauge

13th June 2002

Abstract

The application areas of hypermedia learning are growing as a still larger number of companies educate employees using various software tools. The use of the Internet in learning and training has grown, as the Internet has become the de facto standard for distributing information. Details on the usage of learning material presented to a wide range of users could be a benefit for learning providers. This paper presents a case study on creating and customizing a framework for the e-learning company Zenaria A/S which includes the ability to discovering knowledge on the usage of their interactive stories distributed over the Internet. The paper presents the requirements to a new framework utilizing a novel technique called the post-check hybrid approach from the research area of web usage mining. A system architecture and principle ideas on a user interface which adopts to user requirements are also presented.

1 Introduction

The application areas of hypermedia¹ learning are growing as a still larger number of companies and institutions train and educate employees using various software tools and products designed to help users think and learn. The developments in the field of hypermedia and e-learning span from the rather primitive representation of knowledge with a limited possibility of user interaction[18, 23] to highly interactive, educational games where the development of Hyper-stories[9, 12, 20] is a significant contribution. Several tools

¹Hypermedia is a common term for the use of different media e.g. sound and graphics in a hyper-textual environment, where movement between related nodes is easy.

and websites for educating and training people via the Internet exists but extracting knowledge on the use of a given website is complicated at best, since each website uses different methods of distributing the information. Typically, discovering knowledge on the usage patterns of an educational website requires significant modifications to the existing software framework to enable a seamless integration into an Internet setting.

This paper presents a case study on creating and customizing a framework for the e-learning company Zenaria A/S[4] which includes the ability to discovering knowledge on the usage of their interactive stories distributed over the Internet. Zenaria is faced with a new challenge in moving their e-learning platform from being distributed on CD-ROMs to being accessed via the Internet. The existing platform does not include any central storage of the usage of the interactive stories and a framework for storing and extracting usage information could prove a significant selling point for Zenaria. The paper is focused on the research area of *web usage mining*, i.e., the discovery of knowledge from the use of a website, and utilizes a novel approach named the *hybrid approach*[14] that combines existing usage mining techniques to form a powerful approach to constraint-based knowledge discovery on long sequences of clicks. The paper describes the technologies underlying the hybrid approach and the strengths gained from joining the techniques and presents *MIMER*, the framework developed for Zenaria. The paper also briefly describes the extension of the hybrid approach aimed at retrieving *true traversals*, i.e., the rules found in the usage data (for more on the extension of the hybrid approach, see Paper 2). The paper also presents a number of principle ideas on the design of a user interface supporting the knowledge extraction in MIMER.

The research area of *web usage mining* is focused on knowledge discovery from the use of a website and the developed framework utilizes techniques from this field. Existing web usage mining approaches have different inherent weaknesses in knowledge discovery which invalidates their use in Zenaria, including huge storage requirement consuming an unacceptable amount of space[13], multiple scans over the data that would degrade performance[3, 16] and the inability to introduce additional information such as user demographics when investigating the raw data[10, 11], limiting the flexibility of the knowledge discovery process.

We believe this paper to be the first to presents a complete framework for web usage mining utilizing the post-check expanded hybrid approach (PEHA). The paper includes discussions of adoption of the concepts of the hybrid approach to a specific use case including the principle ideas on how discovered knowledge is to be presented to users.

The remainder of this paper is organized as follows. Section 2 describes the use case which have driven the development of PEHA. Section 3 briefly describes the technologies and ideas upon which PEHA is founded. Section 4 presents an overview of the system architecture of MIMER. Section 5 presents principle user interfaces to be build into the framework. Section 6 concludes on the development of the PEHA framework and presents future work.

company's characteristics. A number of authors focus on creating a story that can, e.g., estimate a psychological profile of a user or educate the user on how specific actions will effect the work environment in the story. The evaluation performed by the consultant will typically focus on how the users have used the story and what this means in terms of the designed purpose of the story. Several factors weigh into the profiling of a single user, but the complete story as formed by the choices, the semantic values assigned to the selected choices and the time spent from a scene has been viewed to a choice is made, referred to as the *decision-time*², are important parameters in evaluating a user.

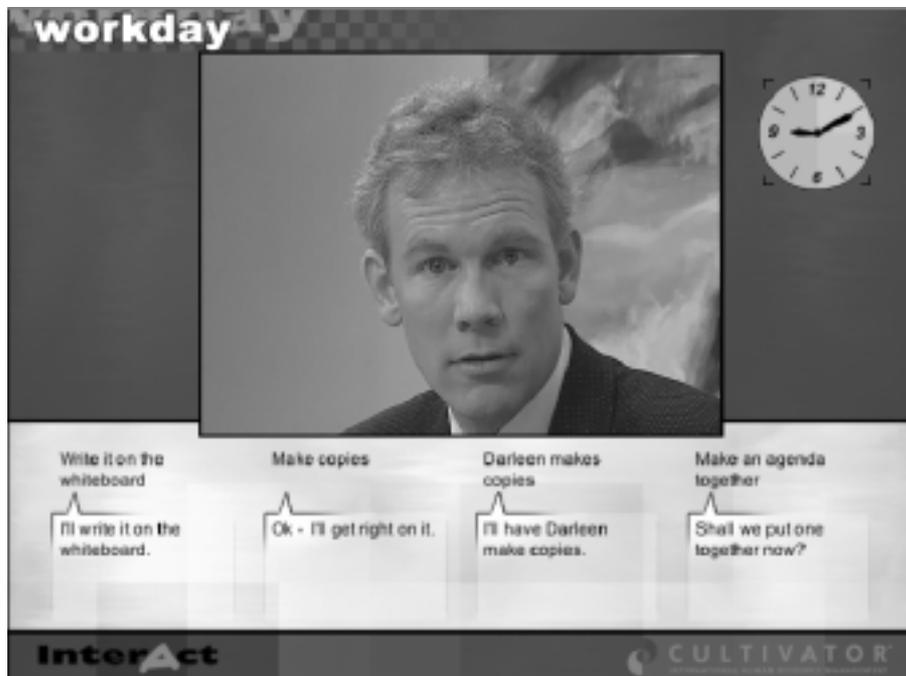


Figure 2.2: Screenshot from a Zenaria Story.

The authors of the stories use a specialized authoring-tool to create the actual stories and this tool includes several ways of assigning semantic meaning to choices and video-sequences. The stories are distributed on CD-ROMs and do not at present include any way of storing the information generated by walkthroughs in a central database. The consultant must therefore supervise the walkthrough of most users to be able to evaluate the employees of a company as a whole.

²Not to be confused with *dwelt-time*, see later.

2.2 Future Approach

Zenaria would like to change this way of distributing their interactive stories and use the Internet instead. Utilizing the Internet will allow for users of the stories to potentially access the stories at all times and from any computer connected to the Internet³. As mentioned above, the current approach does not include any central storage of information about the usage of the stories. Such a centralized storage will allow Zenaria to better evaluate the use of an interactive story.

Knowledge Discovery Having a central database containing usage information will allow Zenaria to offer customers added value in the form of an extended evaluation of a story. In the current approach, the consultant gives an evaluation of a company based on the supervision of all the user walkthroughs. With a large number of users, a consultant could have a hard time discovering all the usage patterns that might be of value to the company. With a central database storing the usage data and a tool to query upon it, the consultant could extract usage patterns and knowledge that might not have been caught during all the supervised walkthroughs. For instance, usage patterns across different user groupings could be very valuable information for a company, but could also be hard for a consultant to spot amongst a large number of users each belonging to several different user groupings. The additional evaluation made possible by such a framework would be a strong selling point for Zenaria.

In total, what Zenaria is in need of in their future approach, is a framework for retrieving, storing and querying the usage of their interactive stories distributed on the Internet.

2.3 Web Usage Mining

Web data mining is the search for knowledge in a web setting. Web data mining includes both *web structure mining* which focus on the structural design of websites, *web content mining* which focus on discovery of knowledge across the Internet and *web usage mining* which focus on knowledge discovery of usage patterns[22]. Web usage mining is a research area that presents itself as a solution to the requirements of Zenaria. All access to an interactive story distributed on the Internet will be done through the HTTP protocol[1]. Web

³In the preliminary stages of this future approach, the web server distributing an interactive story is imagined to be located on a company intranet because of the bandwidth requirements for streamed video.

servers log all requests received in a *web log* and we will use this web log to develop a framework as desired by Zenaria.

Much work has been performed on extracting usage patterns from web logs and the application of such usage patterns range from improving the design of the website to customizing the user experience. One line of work features techniques for working directly on the web log[10, 11]. Another line of work concentrates on creating aggregated representations of the information in the web log[17, 21]. One such aggregated structure is the Hypertext Probabilistic Grammar[6](HPG), which utilizes the theory of language and grammars. Yet another line of work focuses on using database technologies in the clickstream analysis[5, 13], building so-called “web datahouses”[15]. Finally, a prominent line of work focuses on mining sequential patterns in general sequence databases[3, 16]. However, all the mentioned approaches have inherent weaknesses in that they either have huge storage requirements, slow performance due to many scans of the data, or problems when additional information such as user demographics are introduced into the knowledge discovery process. The Post-check Expanded Hybrid Approach (PEHA) used in the described framework utilizes advantages from the HPG and a click fact schema in a web datahouse. PEHA is described in Section 3.

2.4 User Requirements

During development of the framework, we had several meetings with the employees of Zenaria. In these meetings we presented prototypes of the system and attempted to find some concrete requirements that a final framework should meet. This section will describe some of the main points in these talks and try to extract some precise requirements to be used during development.

Context is important The perhaps most important thing, that surfaced during the meetings, was that a walkthrough viewed entirely as a sequence of URL-pages was useless. The stories, and in particular the scenes and the choices presented, are carefully designed to have a very specific meaning that play the major part in evaluating and educating a user. Therefore, any presentation of the usage of the story should be done in the context of the story. This context could include the author description of what a given scene presents, a score or a description relating to a given choice taken or the psychological aspect of a profile that a scene is trying to assess. This becomes even more relevant as some stories are designed so several choices after a specific scene will present the same next scene, but each choice has semantically very different meanings.

Our framework must display this context along with any usage information extracted on the stories otherwise the usage information in itself is without value. Information presented along with context does however allow for a much stronger understanding of the extracted usage information.

Format of the Extracted Information We presented Zenaria with two different interpretations of the overall usage of an interactive history. We will present them here using an analogy consisting of a mass of water flowing around in system of pipes, corresponding to a collection of user-sessions from a web site. On one hand, Zenaria could view the usage of a story in terms of certain amounts of water that run from one place in the system to another, using a specific sequence of pipes. On the other hand, Zenaria could focus on the precise path of each water molecule and not on how much water overall had run from one place in the system to another.

To see why there is an important difference between the two interpretations, consider the following two statements; "The flow of users from page A, via page B to page C was three" and "The three users 1, 2 and 3 have gone through page A, via page B to page C". The first statement focuses on the overall number of users which have used the web site in a certain way, but does not require that it is the same three users that have taken both the individual links between A and B and between B and C. This is the focus of the second statement, which is concerned with the fact that it is the same three users that have navigated the specific path. The difference actually springs from a common assumption used when speaking of navigation on the Internet, where it is sometimes assumed that the choice of a user when navigating is only influenced by the last X pages seen and not on the entire browsing history. This assumption corresponds to the first interpretation presented.

Zenaria interpreted the walkthroughs of users in accordance with the second statement, where the entire path of a user is important and influences all later choices made by the user. Utilizing the first interpretation could lead Zenaria to false conclusions about the usage of their history. The false conclusions could arise because the previous actions of users beyond the range of the browsing history are ignored. Such an assumption is not valid within the context of interactive stories, since all choices made by the users add to the characteristics of the particular user.

This rises a need to develop some mean of only extracting usage information that does not assume a limited browsing history and indicate a correct support level.

Usage Patterns of Different User Groups Zenaria is interested in being able to evaluate very different user groups, to create a much stronger selling point for their stories. Enabling the consultant to extract usage information about a specific grouping of the users would allow a strong evaluation of the company, potentially targeting, e.g., a specific grouping of employees that does not perform as intended by the management.

The framework should facilitate the evaluation of a configurable user grouping, so the consultant could easily specify a number of constraints and retrieve interesting usage information for this specific grouping.

Accessing Extracted Information During the process of presenting prototypes, we also presented different ways of visually showing extracted information about the stories and their usage. The main point in the following discussions was that Zenaria was not in need of a complex tool. The consultants, which will be the primary users of the framework, are non-technical persons and will not be able to specify complex knowledge discovery requests on a developed framework or interpret too complex or cluttered visual structures. The visualization should be done in an already familiar notation and the current authoring framework for the stories already included a formal visual notation for, e.g., scenes and choices. We were recommended to utilize this notation for any visualization in our framework, so extracted usage information would be visualized “on top of” this notation using, e.g., colors or emphasized objects.

Furthermore, any visualization of the story and the usage of the story should be highly interactive, supporting quick access to the information behind the visualization (including both the story context and the usage of individual parts of the story). Having to wait long for, e.g., the story context being retrieved would hinder the interpretation process of usage patterns. Therefore, the framework should be able to present information for a given user grouping and a given setting relatively fast.

Utilizing Zenaria Evaluation Models Zenaria use a number of different visual models to evaluate users of their stories. These models include mapping the psychological profile into a specialized coordinate system and assigning a score to the user. Zenaria would like to be able to map both single users and groups of users into a visual overview of a models to enable consultants to get an overall feel of how the users rate in a specific model⁴. The consultants could utilize such an overview of a group of users plotted in

⁴Note that a specific story only supports some and not necessarily all models

some evaluation model to, e.g., single out users that did not perform well or fall outside expectations.

A framework should include mapping both single, and groups of, user sessions into a certain visual evaluation model in order to create an overview of the results of a high number of user walkthroughs.

2.5 Summary

This section has described the requirements of Zenaria for a framework for storing and accessing usage patterns of their interactive stories on the Internet. Zenaria have a number of specific requirements, including restricting the group of users to extract knowledge on, which cannot easily be solved utilizing the existing technologies, either because of storage problems, response times or the inability to extract knowledge for specific user groupings, as mentioned in Section 2.3. The underlying technologies used in developing the framework is described next.

3 The Hybrid Approach

This section will briefly describe the key technologies behind the *hybrid approach*[14] utilized in the developed framework. This includes the *click fact schema* used for accessing the usage data of the interactive story and the *Hypertext Probabilistic Grammar*(HPG)[6] used for extracting interesting usage information on the usage data. The section will conclude with a discussion of the Hybrid Approach that combines the two technologies to form a strong basis for finding information of the usage of the interactive stories. Note that this section will not in detail describe the post-checking performed in the developed framework, for more on this see Paper 2.

3.1 Click Fact Schema

The click fact schema uses the individual clicks on the web site as the essential fact when storing the requests in the data warehouse[15]. This will preserve most of the information found in a web log and store it in the data warehouse at a very fine granularity. Very little information in the web log is therefore lost when loaded into the data warehouse. The high granularity allows for extracting information about very specific clicks in the usage data. However, retrieving detailed information on sequential clicks require a number of self-join operations performed on the fact table[13]. The schema is shown in Figure 3.1.

3.2 Hypertext Probabilistic Grammar

The nature of web sites, web pages and link navigation has a nice parallel which proves rather intuitive and presents a model for extracting information about user sessions. The model uses a *Hypertext Probabilistic Grammar* that rests upon the well established theoretical area of language and grammars.

The model maps web pages to grammar states⁵ and adds two additional artificial states, the start state S and the end state F, to form all states of the grammar⁶. We will throughout the paper use the terms state and page interchangeably.

The probability of a production between two states in the grammar is assigned based on the information in the web log. The probability of a production is proportional to the number of times the link between the two pages was traversed relative to the number of times the state on the left side

⁵This is only true if the HPG is created with a history depth of 1, see later.

⁶Actually, it is the states of the equivalent Deterministic Finite Automata, but we will refer to the grammar in this paper.

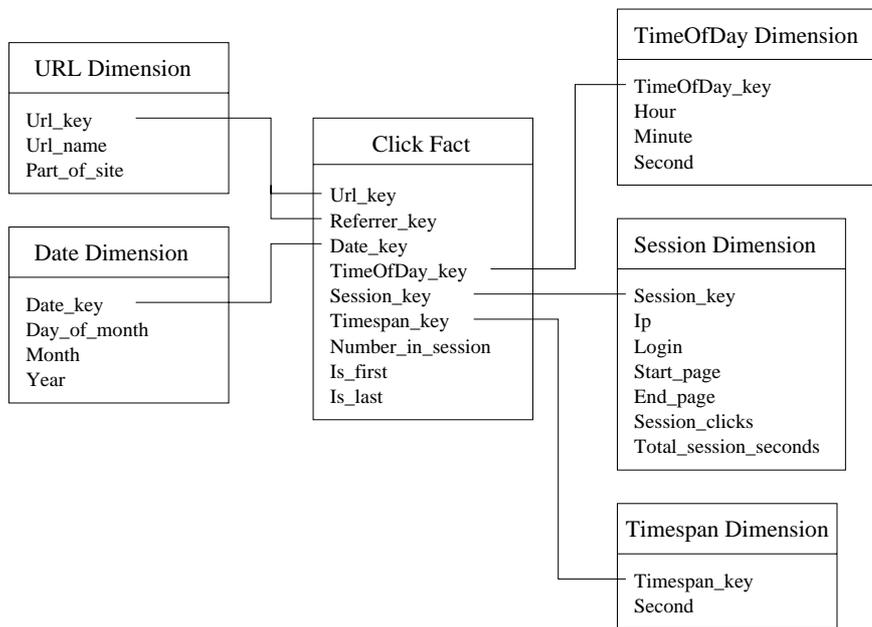


Figure 3.1: The click fact schema.

of the production was visited overall. Note that not all links within a web site may have been traversed so some of the links might not be represented as productions in an HPG. An example of an HPG is shown in Figure 3.2.

The probability of a string in the language of the HPG can be found by multiplying the probabilities of the productions needed to generate the string. Note that web pages might be linked in a circular fashion and therefore the language of the HPG could be infinite. An HPG specifies a threshold η against which all strings are evaluated. Only strings with probability above the threshold is included in the language of the HPG (with the given threshold), L^η .

Mining an HPG is essentially the process of extracting high-probability strings from the grammar. These strings are called *rules*.⁷ These rules will describe the most preferred trails on the web site since they are traversed with a high probability. Mining can be done using both a breath-first and a depth-first search algorithm[6].

The mining of rules on the HPG using a simple breath-first search algorithm has been shown to be too imprecise for extracting a manageable number of rules. Heuristics have been proposed to allow for a better control that more accurately and intuitively presents relevant rules mined from an

⁷The notion of a rule and a string will be used interchangeably

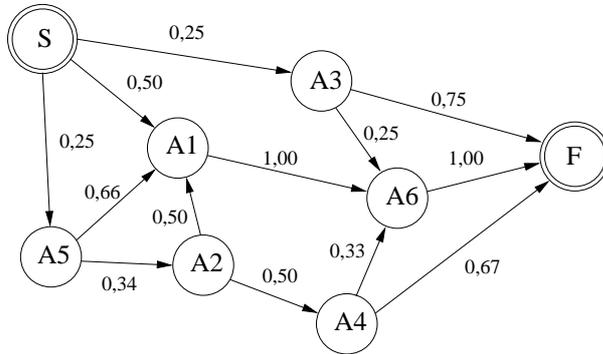


Figure 3.2: Example of a Hypertext Probabilistic Grammar.

HPG and allow for e.g. generation of longer rules and for only returning a subset of the complete rule-set[7, 8].

The HPG model inherently assumes that a choice on a given web page is only dependent upon a limited browsing history. This means that the sequence of web pages in a rule mined from an HPG is not necessarily supported as a whole in the collection of session the HPG represents, but the different parts in the sequence will be represented in the collection. This inherent assumption can lead to *false trails* being extracted from the HPG, since rules are generated using the assumption of independent states in the grammar[19]. For more on the problem of false trails and the assumption of limited browsing history, please refer to Paper 3.

History Depth In Figure 3.2, a web page maps directly to a state in the grammar. The HPG can also be generated with a *history-depth* N above 1. With a history-depth of e.g. 2, a state in the HPG could be $[A1A6]$ representing the two web pages $A1$ and $A6$ requested in sequence. The basic structure of the HPG remains the same but each state includes a “memory” of N states traversed in sequence and the HPG will therefore have a greater accuracy and consist of a greater number of states and productions. For more on history depth, please refer to Paper 3.

Representing additional information Since the HPG is a compact model it does not contain information at a high granularity. This means that the HPG cannot represent information about the single clicks on a website but only present an aggregated overview. This proves a decisive weakness since the HPG cannot easily be utilized to represent additional information on

clicks and therefore cannot be used to specify constraints on what knowledge should be mined[14].

3.3 Combining the HPG and the Click Fact

As described, the click fact schema and the HPG have almost opposite advantages. The click fact schema holds information at a high granularity but is not effective in extracting information on sequences, whereas the HPG is a compact, aggregated structure that can readily be mined for sequences, but is unable to represent data at a high granularity. Combining the two techniques could however form an approach that could both hold data at a high granularity as well as extract information on sequences. This approach is named the *hybrid approach*.

The Hybrid Approach The main idea is to create a specialized HPG on the fly, where the HPG will represent exactly the user sessions of interest. This would overcome the inherent problems of the complete HPG model in representing additional information. The specialized information is extracted from the click fact schema and used to build the specialized HPG. Extracting information from this HPG will result in rules relating only to the requested sessions and not on all sessions, as is the case with the complete HPG model. The structure of the click fact schema is readily usable for creating the HPG and allows for easy constraining of desired information. The constraints can be implemented as a database query that restricts the number of sessions being used to build the HPG (see later). In Paper 2, the hybrid approach is shown to perform comparably to a rival data warehouse schema when considering flexibility, running time and storage requirements.

Post-check Extension Once the HPG has been built, it can be used for mining rules using the HPG algorithms. The rules mined can, as described in Section 3.2, potentially be false rules, so these rules are used as candidates for achieving rules with a correct support level. The correct support for a given rule is found by querying the click fact schema for the actual support of a given sequence of requests within single sessions. We will refer to the process of finding the correct support level for extracted rules as *post-checking* (for more on post-checking, see Paper 2).

Constraining the Information As mentioned, the framework allows for constraints to be placed on what sessions should be represented in the specialized HPG. The framework allows for two different kinds of constraints

to be specified, namely *session-specific* and *click-specific* constraints. The session-specific constraints are constraints placed on properties that relate to entire sessions such as IP-address, number of requests in a session or total time spent in a session. Click-specific constraints are constraints relating to properties on individual requests in a session such as the requested URL, the dwell-time, i.e., the time between to requests within a session, or the specific time of an interaction with the web site. When constraining the information to be extracted into the HPG, it is important to ensure that an interconnected HPG is built. In other words, we do not want to build an HPG where some states are not linked with other states. This could happen if we did not differ between click- and session-specific constraints. Session-specific constraints ensure that all individual requests valid under the constraint are interconnected (since all requests within a session will be related to the same session specific property), but click-specific constraints would extract single requests that might not be interconnected when mapped to states in the HPG. Therefore, we adopt the interpretation that click-specific constraints go through a two-step process[14].

The main idea is to find all session identifiers where at least one request in the session is valid under the click-specific constraints. Having found these session identifiers, we can use them as session-specific constraints in the final constraint since they now effectively guarantees an interconnected HPG. This means that click-specific constraints are interpreted as a constraint on at least one request in a session.

Adopting to User Requirements In Section 2.4, we presented a number of user requirements which evolved during the interaction with Zenaria. The framework that is enabled using the hybrid approach allows for all the described requirements to be met. The rules mined from the HPG are checked against the click fact schema using the post-checking explained above to find the correct support level, thus avoiding any assumptions inherent in the HPG model about a limited browsing history. The need for extracting information on specific user groups is supported through the ability of placing constraints on what sessions should be represented in the specialized HPG. Adopting to the requirements concerning presentation of extracted information is described in Section 5.

4 The Hybrid Framework

This section will present the overall system architecture of the developed framework and briefly describe the responsibilities of the different modules in the system.

4.1 System Overview

We have named the developed framework *MIMER* after the god in Nordic mythology who guarded the well of wisdom and was a valuable advisory to the main god Odin.

The MIMER framework has four overall parts. The *MIMER Data Warehouse Loader* is responsible for taking a web log, cleaning it, i.e., removing all requests which are considered useless and writing it to the database using the correct schema. The *MIMER HPG Engine* is responsible for querying the database using specified constraints, building the HPG structures in main memory and performing mining on these structures. The HPG Engine will also perform the post-checking on extracted rules, as described in Section 3.3. The *MIMER Story Engine* is responsible for accessing the context of a story, including querying the original story database. The *MIMER Visualizer* will present all extracted information to users, including the story context and the extracted usage knowledge (see Section 5). The parts are illustrated in Figure 4.1 and each part is described further below.

The Data Warehouse Loader The cleaning process is divided into three overall stages, namely extracting all valid requests into an independent XML format, rearranging the XML format into user sessions and writing the user sessions to the database schema. Note that the Data Warehouse Loader could in principle, since it is developed as highly configurable, be used as a single component to load a data warehouse using various data sources and database schemas. This could be utilized by Zenaria in case they wish to utilize the remaining MIMER framework on top of an alternative data source such as, e.g., log files written by the CD-ROM based applications.

The HPG Engine The HPG Engine allows for a number of different constraints to be set before the database is queried for the information needed to build the HPG. These constraints are used to build HPGs representing specialized user groups. Once the HPG is built, several parameters can be passed to a mining process and the rules extracted represent knowledge on the user groups that the HPG represents. The rules extracted directly are generated using an assumption of limited browsing history (see Section 3.2)

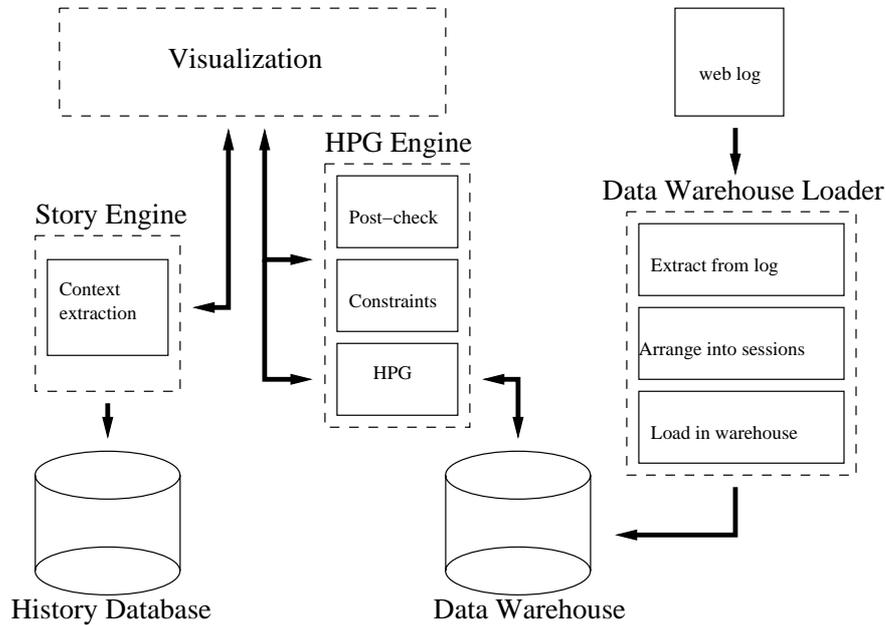


Figure 4.1: Overview of the MIMER framework

so the framework offers a mechanism for performing post-checking on the extracted rules to obtain rules with correct user support. For more on the post-checking functionality, please refer to Paper 2.

The Story Engine Zenaria holds all information concerning an interactive story in a dedicated database. The Story Engine offers an extraction-interface that can be used to retrieve the relevant history context. This history context is very important when interpreting extracted usage knowledge, as mentioned in Section 2.4. This context is used by the Visualizer to create a more intuitive user interface and user interaction with the MIMER framework.

The Visualizer The Visualizer represents usage information using the context of the actual story to allow for an easy and intuitive interaction with the MIMER framework. The visualization is developed with the aim of utilizing the formal notation used internally in Zenaria when developing and discussing the interactive stories. Principle ideas on the design of the user interface is presented in Section 5.

5 Visualization

The following screenshots and descriptions are not taken from an actual implementation of the MIMER Visualizer, the module designed to provide user interaction with the MIMER framework. They are mainly principle ideas about an interactive visual environment that could enable users at Zenaria to take full advantage of the features available in MIMER.

Building the HPG The visualization is very focused on presenting the HPG to the users in an intuitive way. However, the first thing to be done is to construct an HPG. As described in Section 3.3, a key feature of the framework is the ability to specify constraints on what information should be represented in the HPG. Since the primary users are consultants and simplicity is important, we have chosen an approach in which a *wizard* will guide the user through the process of specifying constraints. The process will mainly consist of selecting the dimensions and fields on which to specify constraints (see Figure 5.1) and specifying the concrete constraints for each dimension/field pair selected (see Figure 5.2). The main idea is that the dimensions and fields in the click fact schema is to be presented here. The framework must be initialized so that the HPG Engine is able to separate constraints into click- and session-specific (see Section 3.3).

The screenshot shows a dialog box titled "Specify HPG Constraints Dimensions". The main text inside says "Please specify which Dimensions and which fields such include constraints." There are two columns of checkboxes. The left column lists "Demographics", "Timespan", "Session", and "Date". The right column is titled "Demographics" and lists "Gender", "Age", "Title", and "Location". At the bottom, there are three buttons: "< Back", "Next >", and "Cancel".

Figure 5.1: Selecting the dimensions to constrain on.

An additional approach that could simplify input for users could be to predefine and name a number of typical constraints used by the consultant. Such predefined constraints could then be picked from a list without having

Specify HPG Constraints Dimensions

Please specify the constrain for each selected Dimension/Field pair.

- Demographics/Age
- Demographics/Title

Select Operator

- Equal
- Lower than
- Greater than

Select Value

30

< Back Next > Cancel

Figure 5.2: Specifying the actual constraints.

to go through the process of specifying each constraint individually. Such an approach would require an extensive knowledge of the problem domain of the individual story and is not presented here.

After having specified the constraints, the HPG is built and presented to the user. The created HPG is visualized to the user, using existing notation from Zenaria’s authoring tool. In this notation, ellipses represent scenes and circles represents choices after a scene. Having to generate states in the HPG not only for the individual scenes (which are the URLs requested) but also for the choices which leads to the individual scenes, the basic click fact schema (see Section 3.1) and the HPG construction algorithm is modified in order to construct a state for each scene/choice pair in the session collection. This is crucial, since the actual choice of a user is more important than the sequence of scenes viewed, i.e., URLs requested (see Section 2.4). Appending an identifier for the choice to the request and mapping these identifiers into a separate dimension in the data warehouse during the load phase will enable the filtering of the individual choices. As the HPG is constructed, these identifiers are used to divide the existing productions into *dedicated productions*, where each dedicated production is representing the request for a given scene with a specific choice. This will allow us to visualize the HPG with the notation used in Zenaria. The basic layout is shown in Figure 5.3.

Notice that Figure 5.3 includes the interactive visualization of the context of the story. Highlighting, e.g., a given choice in the visualization, will fetch the context from the story database and the overall usage information from the data warehouse. This information will be presented to the user thus

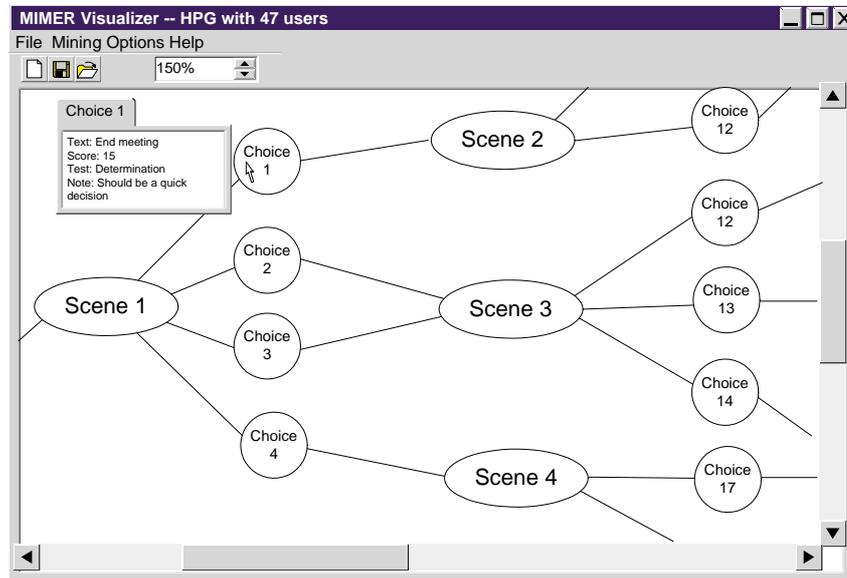


Figure 5.3: Presenting the HPG with context.

enabling a stronger understanding of the precise semantics of the otherwise rather limited visual representation.

Mining for Knowledge Once the desired HPG is built, usage knowledge must be mined on it. We have again decided to use a wizard to guide the user through the process of specifying parameters to the mining process. Since there are several parameters that could be specified on the mining process, simplifying the user input has a great significance. The wizard provides sliders instead of actual numeric input on parameters, check boxes for whether or not to use heuristics in the mining process and an option that enable a limitation of the number of rules to be considered. Figure 5.4 shows the wizard for specifying input to the mining process. Note that it would be desirable if a heuristic were developed that could warn users if the specified parameters will cause the mining process to run in an undesirable amount of time.

Visualizing Rules Once the mining process is complete, the rules found can be presented in the visual framework described above. To avoid having several rules cluttering the user interface and potentially making it hard to identify the individual rules, we adopt the convention of visualizing individual rules, which are then presented on top of the existing HPG by highlighting

Specify Parameters for Mining Process

Please provide the appropriate parameters for the mining process. The parameters are used to find what knowledge is interesting.

To which degree is knowledge from scenes inside the history relevant? 0 100

What percentage of users must have visited the first scene? 0 100

What percentage of users on a scene must have taken a specific choice for it to be interesting? 0 100

Should the mining process persue long sequences of knowledge?

Should the mining process only approximate the knowledge?

How many rules are you interested in? 25

< Back Next > Cancel

Figure 5.4: Specifying input parameters to the mining process.

the scenes and choices of the particular rule. This promotes that each rule is examined individually. Furthermore, the context can still be fetched, but the details about the visualized rule are also available from the context window. Figure 5.5 shows a visualized rule including the context and rule details for a specific choice in the HPG. The figure shows the detail of the rules as a support count and an identifier for each user, but these details could be extended.

Visualizing Evaluation Models As described in Section 2.4, Zenaria currently uses visual models in order to perform, e.g., a psychological analysis of the user. The constructed HPG effectively represents a number of users, which could be inserted in such a model in order to identify potential interesting aspects for the user grouping. Figure 5.6 illustrates the principle, where a number of users are plotted into the coordinate system of an PAEI model[2]. The PAEI model plots users into a coordinate system dependent on their individual score and the consultant can as a general guideline to what characterizes the user use their placement in the coordinate system.

Please note that the figure does not present all 47 users in the HPG to avoid cluttering up the figure. From the figure it should be possible to click on the plot of a specific user and receive any available information on this user. This would enable the consultant to further investigate any information on the specific user, potentially creating a new HPG to further drill down on interesting knowledge discovered.

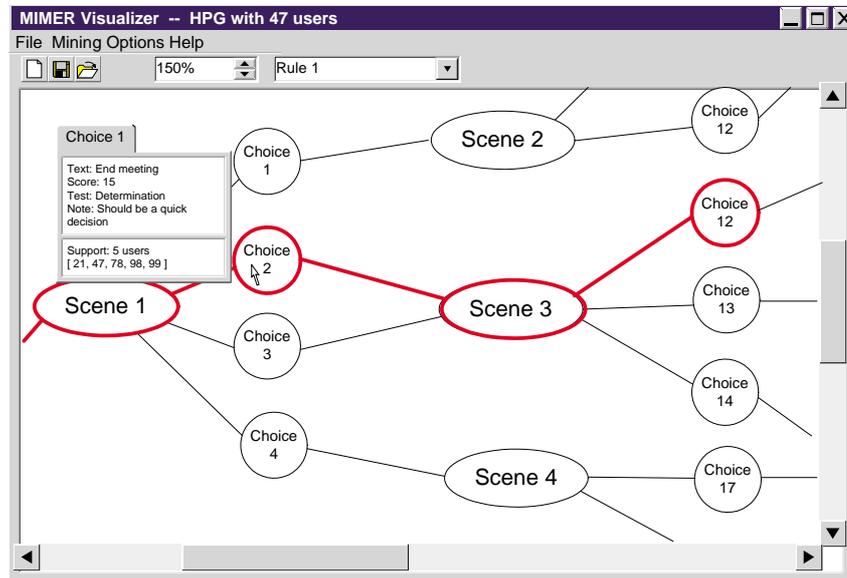


Figure 5.5: Visualizing a rule with context.

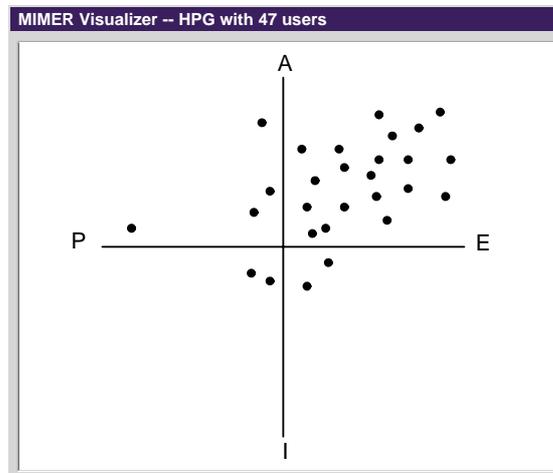


Figure 5.6: Visualizing users within an PAEI model.

General capabilities A main feature of a final visualization framework would be the possibility of saving and restoring user *studies*[15], meaning saving the constraints used to build an HPG so it can be restored at a later point. This would allow the consultant to save interesting HPGs for later review with a company. The framework should also include some way of

printing reports for an HPG, which can be presented to the company as a textual description of the extra evaluation of the use of an interactive story.

6 Conclusion and Future Work

This paper has presented the development of MIMER, a framework for storing and extracting knowledge on the usage of a website distributing interactive stories on the Internet. The development has been driven by a number of specific needs of the company Zenaria in their move from CD-ROM based to Internet based distribution. MIMER utilizes the novel hybrid approach to web usage mining, which combine the strengths of a detailed data warehouse schema and a compact, aggregated structure in order to provide a powerful, constraint-based tool. MIMER also includes the novel extension to the hybrid approach, post-checking, in order to provide the accurate usage knowledge desired by Zenaria. The paper has also presented a number of principle ideas on the design of a user interface for the employees of Zenaria.

In future work, the implementation of MIMER, including the user interface, must be completed in order for Zenaria to utilize the developed framework. Furthermore, modifying the loading of the data to handle different kinds of input besides the data in a web log could expand the areas of application of MIMER. The MIMER framework offers several areas of expansion, since it is currently designed for the specific use case of this paper, but a more generic interface could be developed, since the technologies utilized in PEHA are applicable in a large number of areas. PEHA is a novel approach to web usage mining, so further experiments are requirement for the approach to become mature, including performance measurements, expansion of the mining algorithms and investigations into the general validity and usability of the HPG model. For more on the hybrid approach and the HPG model, see Papers 2 and 3.

References

- [1] RFC 2068. <http://www.ietf.org/rfc/rfc2068.txt>.
- [2] I. Adizes. *Corporate Lifecycles: How and Why Corporations Grow and Die and What to Do About it*. Prentice Hall, 1990.
- [3] R. Agrawal and R. Srikant. Mining Sequential Patterns. In *11th International Conference on Data Engineering, ICDE*. IEEE Press, 1995.
- [4] Zenaria A/S. <http://www.zenaria.com>.
- [5] A.G. Büchner, S.S. Anand, M.D. Mulvenna, and J.G. Hughes. Discovering Internet Marketing Intelligence through Web Log Mining. In *UNICOM99 Data Mining and Datawarehousing*, 1999.
- [6] J. Borges. *A Data Mining Model to Capture User Web Navigation Patterns*. PhD thesis, Department of Computer Science, University College London, 2000.
- [7] J. Borges and M. Levene. Heuristics for Mining High Quality User Web Navigation Patterns. Research Note RN/99/68. Department of Computer Science, University College London, Gower Street, London, UK, 1999.
- [8] J. Borges and M. Levene. A Fine Grained Heuristic to Capture Web Navigation Patterns. *SIGKDD Explorations*, 2(1), 2000.
- [9] L. Cernuzzi, C. Kreitmayr, and J. Sanchez. Supporting the Design of Hypermedia Stories. In *ED-MEDIA '97*, 1997.
- [10] R. Cooley, J. Srivastava, and B. Mobasher. Web Mining: Information and Pattern Discovery on the World Wide Web. In *9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, 1997.
- [11] R. Cooley, P. Tan, and J. Srivastava. Websift: the Web Site Information Filter System. In *1999 KDD Workshop on Web Mining, San Diego, CA.*, 1999.
- [12] Sanchez J., Lumbreras M., and Bibbo. L. Hyperstories for Learning. In *Lectures on Computing*, 1995.
- [13] J.Andersen, A. Giversen, A. H. Jensen, R. S. Larsen, T. B. Pedersen, and J. Skyt. Analyzing Clickstreams Using Subsessions. In *International Workshop on Data Warehousing and OLAP*, 2000.

- [14] S. E. Jespersen, T. B. Pedersen, and J. Thorhauge. A Hybrid Approach to Web Usage Mining - Technical Report R02-5002. Technical report, Dept. of CS Aalborg University, 2002.
- [15] R. Kimball and R. Merz. *The Data Webhouse Toolkit*. Wiley, 2000.
- [16] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth. In *ICDE*, 2001.
- [17] J. Pei, J. Han, B. Mortazavi-asl, and H. Zhu. Mining Access Patterns Efficiently from Web Logs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2000.
- [18] R. Pennell and E. Deane. Web Browser Support for Problem-based Learning. In *Ascilite95*, 1995.
- [19] S. Ross. *A First Course in Probability*. Prentice Hall, 1998.
- [20] J. Sánchez and M. Lumbreras. Hyperstories: A Model to Specify and Design Interactive Educational Stories. In *XVII International Conference of the Chilean Computer Science Society*, 1997.
- [21] M. Spiliopoulou and L. C. Faulstich. WUM: a Web Utilization Miner. In *Workshop on the Web and Data Bases (WebDB98)*, 1998.
- [22] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, 1(2), 2000.
- [23] S. Wiest and A. Zell. Improving Web Based Training Using an XML Content Base. In *ED-MEDIA*, 2001.

Paper 2

PEHA: The Post-check Expanded Hybrid Approach to Web Usage Mining

PEHA: The Post-check Expanded Hybrid Approach to Web Usage Mining

Søren E. Jespersen Jesper Thorhauge

13th June 2002

Abstract

With a growing amount of business-critical information on the Internet, the focus on how this information is accessed has risen. Several approaches to mining usage patterns exist, one of which is the hybrid approach combining an aggregated structure and a data warehouse. The knowledge extracted from the hybrid approach, called rules, can be unsupported in the usage data. This paper presents a novel approach, called Post-check Expanded Hybrid Approach (PEHA), in which the rules are considered candidate rules for a validating process ensuring correct support of all extracted knowledge. The implementation of the approach is shown to run in constant time using materialized views and performs comparably to a rival data warehouse schema when considering running time, flexibility and required storage.

1 Introduction

With a growing amount of business-critical information being accessible online via the Internet, the focus on how this information is accessed has risen. Knowledge of the access patterns of users on a website can be used to, e.g., improve content presented to the users or for efficiently target advertising to individual users. The research into *web usage mining*, a variant of data mining, focuses on discovering knowledge from the usage of websites. Several approaches have been developed in this area including aggregated structures[19, 20] and database techniques[4] for representing usage data. The *hybrid approach*[16] combines the Hypertext Probabilistic Grammar(HPG)[5, 6], an aggregated structure, with a data warehouse schema to facilitate a flexible and efficient web usage mining framework. The

HPG model contains a grammar representing web pages and links, and generates a language consisting of the most preferred user trails on the website, referred to as *rules*. The HPG model is combined with the data warehouse *click fact* schema[17] in the hybrid approach which enables constraint-based extraction of knowledge. However, the knowledge extracted using the HPG could potentially be unsupported in the usage data from the web server, since the aggregated model assumes a limited browsing history due to interdependency of the states in the grammar. Therefore, the hybrid approach which uses the HPG model to extract knowledge inherits this property.

This paper presents an expansion of the hybrid approach aimed at ensuring that knowledge with a correct support level is extracted. The expansion considers the rules extracted from the HPG to be *candidate rules* and validate these rules against the usage data stored in the click fact schema in order to determine the correct support level of a given rule. We call the validation of a rule against the click fact schema a *post-check* and we name the expansion the Post-check Expanded Hybrid Approach (PEHA). The expansion will overcome the disadvantages of the hybrid approach of potentially discovering knowledge that indicate an incorrect support level. The approach is implemented and experiments with a prototype shows that the post-check can be performed in constant time using a DBMS supporting materialized views. The implementation performs comparable to queries directly on a rival data warehouse schema when running time, flexibility and storage is considered.

In related work, one line of research uses techniques which work directly on the raw web logs written by the web servers[10, 11] but provides little flexibility towards constraining the extracted information. Another line of research maps the usage data from the web server to database schemas and utilize database technologies to perform *clickstream analysis*[4, 15]. These techniques demands either high storage requirements or delivers too coarse-grained information to suit a variety of scenarios. A prominent line of research focus on mining *sequential patterns* in general sequence databases[2, 14] which is not limited to web usage mining but can also be applied to, e.g., DNA databases. The approaches in this field requires either an expensive candidate generation phase or demands for several scans of the usage data which could present serious scalability issues.

We believe this paper to be the first to present an expansion of the hybrid approach to extract knowledge with a correct support level. By enabling the extraction of supported knowledge, the expanded hybrid approach proves both an efficient and very flexible approach for constraint-based mining of

web usage data.

The remainder of the paper is organized as follows. Section 2 describes the hybrid approach. Section 3 presents the post-check expansion of the hybrid approach. Section 4 presents results of the experimental evaluation of PEHA. Section 5 concludes on the paper and presents future work.

2 The Hybrid Approach

This section will present the underlying technologies used in the hybrid approach, including the *click fact schema* and the *Hypertext Probabilistic Grammar*. The section will end with a overall description of the principle ideas of the hybrid approach.

2.1 Click Fact Schema

The click fact schema uses the individual clicks on the web site as the essential fact when storing the requests in the data warehouse[17]. The schema will preserve most of the information found in a web log and store it in the data warehouse at a very fine granularity. Very little information in the web log is therefore lost when loaded into the data warehouse. The high granularity allows for extracting information about very specific clicks in the usage data. However, retrieving detailed information on sequences of clicks require a number of self-join operations performed on the fact table[15]. The schema is shown in Figure 2.1.

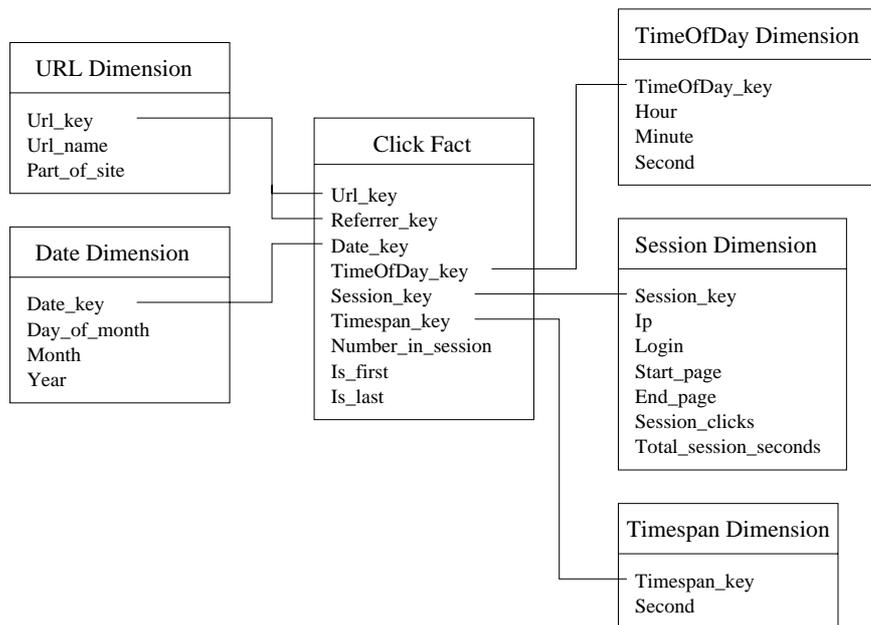


Figure 2.1: The click fact schema.

2.2 Hypertext Probabilistic Grammar

The nature of web sites, web pages and link navigation has a nice parallel which proves rather intuitive and presents a model for extracting information about user sessions. The model uses a *Hypertext Probabilistic Grammar*(HPG)[18] that rests upon the well established theoretical area of language and grammars. We will present this parallel using the example in Figure 2.2. In the following we will use the terms state and production even though these concepts are related to finite automatons and not to grammars. When using the terms we will implicitly be speaking of the equivalent Deterministic Finite Automata of the grammar.

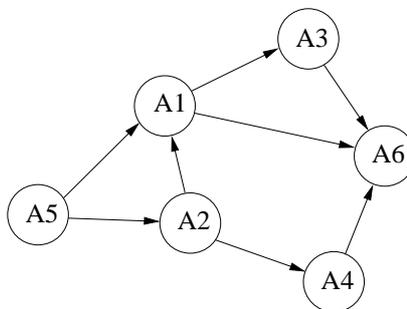


Figure 2.2: Example of a Web site structure.

The figure shows a number of web pages and the links that connect them. As can be seen, the structure is very similar to a finite automata with a number of states and a number of productions leading from one state to another. It is this parallel that the HPG model explores. The model maps web pages to grammar states¹ and adds two additional artificial states, the start state S and the end state F, to form all states of the grammar. We will throughout the paper use the terms state and page interchangeably.

From the processing of the sessions in the web log each state will be marked with the number of times it has been requested. The probability of a production between two states is assigned based on the information in the web log. The probability of a production between two pages is proportional to the number of times the link between the two pages was traversed relative to the number of times the state on the left side of the production was visited overall. Note that not all links within a web site may have been traversed so some links might not be represented as productions in an HPG. An example of an HPG is shown in Figure 2.3.

¹This is only true if the HPG is created with a history depth of 1, see later.

The probability of a string in the language of the HPG can be found by multiplying the probabilities of the productions needed to generate the string. Note that web pages might be linked in a circular fashion and therefore the language of the HPG could be infinite. An HPG specifies a threshold η against which all strings are evaluated. Only strings with probability above the threshold is included in the language of the HPG (with the given threshold), L^η .

Mining an HPG is essentially the process of extracting high-probability strings from the grammar. These strings are called *rules*.² These rules will describe the most preferred trails on the web site since they are traversed with a high probability. Mining can be done using both a breath-first and a depth-first search algorithm[5]. The parameter α is used in the mining process to specify what weight should be given to the first states that are requested first in a user session and α span from 0 (rules must begin with a state that was first in a user session) to 1 (all requests are weighted equally).

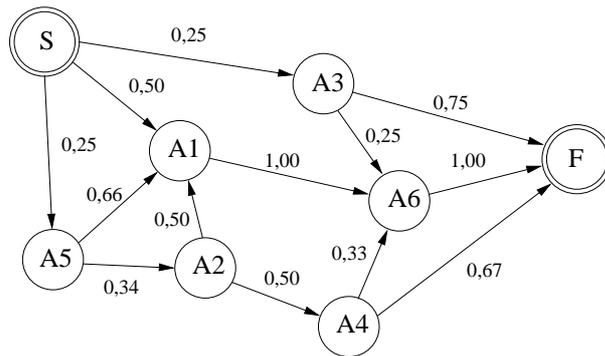


Figure 2.3: Example of a Hypertext Probabilistic Grammar.

The mining of rules on the HPG using a simple breath-first search algorithm has been shown to be too imprecise for extracting a manageable number of rules. Heuristics have been proposed to allow for a better control of the mining of rules from an HPG[7, 8]. The heuristics are aimed at specifying controls that more accurately and intuitively presents relevant rules mined from an HPG and allow for, e.g., generation of longer rules and for only returning a subset of the complete rule-set.

The HPG model has an inherent assumption that the choice of the next page to browse only depends on the current state in the HPG, i.e., the independency between states. This assumption means that the language of an HPG can include strings corresponding to trails not included in any of

²The notion of a rule and a string will be used interchangeably.

the true traversals, namely *false trails*. The HPG can be extended with the Ngram[9] concept to improve the precision of the model, by introducing an added *history depth* in the model which effectively determines the length of the assumed user memory when browsing the website. These aspects of the HPG model is further investigated in Paper 3.

Representing additional information An HPG has no memory of detailed click information in the states, so if rules for *additional information*, e.g., rules relating only to sessions for users with specific demographic parameters were to be mined, each production could be split into a number of *middle-states*, where each middle-state would represent some specific combination of the different parameters. This is illustrated in Figure 2.4 where *A* and *B* are original states and 1 to 7 represent new middle-states. Note that the probability of going from a middle-state to *B* is 1 for each middle-state.

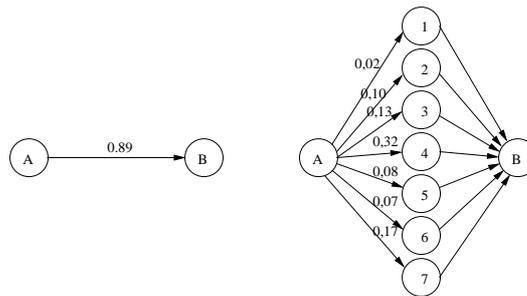


Figure 2.4: Including demographic information.

Figure 2.4 only illustrates specialized information grouped into seven different categories, which, e.g., could be some demographic information about the user. If several different kinds of information were to be represented, the number of middle-states would increase exponentially since all combinations of the different kinds of parameters would potentially have to be represented for all states in the HPG, thus creating a problem with state-explosion in the HPG. For instance, if the HPG should represent the gender and marriage status of users, each production could be split into four middle-states. However, if the HPG should also represent whether a user had children or not, each production needs to be split into eight middle-states. This factor increases with the cardinality of each additional parameter added. This solution scales very poorly. For instance, representing gender, age in years (0-110), salary (grouped into ten categories), number of children (five categories), job status (ten categories) and years of working experience (0-40) could easily require

each production to be split into over 4 million middle-states. This can be seen by multiplying the cardinality of all parameters, $2 * 110 * 10 * 5 * 10 * 40$. Doing this for an HPG that includes only ten interconnected states would require over 400 million states (including middle-states) in the full HPG. This is clearly not a scalable solution, since the number of clicks represented might not even be 400 million. Furthermore, the existing algorithms[5] should be expanded to be able to work on this new type of states in the HPG.

Alternatively a single middle-state could be inserted in each production, containing a structure indicating the distribution of clicks over the parameters. This would effectively reduce the state-explosion for middle-states but the same amount of information needs to be represented inside the single middle-state so this does not solve the problem and furthermore it would require significant changes to the existing mining algorithms to include parameters in the mining process and to support, e.g., mining of rules for a range of parameter-values.

2.3 Combining the HPG and the Click Fact

As described, the click fact schema and the HPG have almost opposite advantages. The click fact schema holds information at a high granularity but is not effective in extracting information on sequences, whereas the HPG is a compact, aggregated structure that can readily be mined for sequences, but is unable to represent additional data at a high granularity. Combining the two techniques could however form an approach which could both hold data at a high granularity as well as extract information on sequences. This approach is called the *hybrid approach*.

The Hybrid Approach The main idea is to create a *specialized HPG* on the fly, where the HPG will represent exactly the user sessions of interest. This would overcome the inherent problems of the complete HPG model in representing specialized information. The sessions of interest are extracted from the click fact schema and used to construct the specialized HPG. Extracting information from this HPG will result in rules relating only to the sessions of interest and not on all sessions as is the case with the complete HPG model. The structure of the click fact schema is readily usable for creating the HPG and allows for easy constraints on desired information. The constraints can be implemented as a database query that restricts the number of sessions being used to build the HPG.

However, since the hybrid approach utilizes the HPG model to extract information, the hybrid approach also inherits the possibility of extracting false trails. This feature could be very undesirable in certain settings, so the

focus of this paper is to examine whether or not it is possible to only extract *true traversals* from the framework. This extension of the hybrid approach is further investigated in Section 3.

The complexity considerations surrounding the extraction of rules from the HPG can be split into three separate parts, which are presented below. In the following, C represents the number of clicks, P represents the number of productions and S represents the number of states.

Database access: the aggregate results obtained from the database can, using hash-based aggregation, be computed by two single scans of the click fact table, i.e., in time $O(C)$. If the DBMS supports pre-aggregated data, this can be computed in time $O(P) + O(S) = O(P)$.

Constructing the HPG: the HPG is a compact, aggregated representation and the size is dependent upon the number of productions it must represent, not the number of sessions since they are aggregated into each state. Assuming that productions are stored in a hash table, they can be retrieved in constant time[12] and the creation of the HPG can be done in $O(P)$ time.

Mining the HPG: the mining of an HPG can be performed using both a general Breath First Search(BFS) and a Depth First Search(DFS) algorithm[5]. The complexity of both algorithms is $O(S + P)$ [12]. Note that the prototype implementation used in the experiments (see Section 4) uses BFS because of a more efficient memory usage[5].

3 Achieving Correct Rules

This section describes the concept of false trails in detail and derive a notion of a *true rule* (for more on the *probability* of extracting false trails, see Paper 3). Furthermore, a solution for achieving true rules using the hybrid approach is presented.

3.1 True Rule Semantics

To understand why a rule extracted from an HPG mining process could be indicating a false trail, it is important to note that the HPG model assumes that the productions expanded from a single state X is entirely dependent upon state X and not on any previous states. This might be a reasonable assumption in mining web access patterns, since the browsing behavior is not usually dependent upon the entire browsing history.

This assumption can be relaxed somewhat by incorporating the Ngram model[5] into the HPG model, modeling memory or *history depth* into the single states in the HPG. The Ngram model can be utilized if the general assumption was made that a browsing choice is only dependent upon e.g. the last 3 pages seen. With this assumption, the Ngram model can be used to model each state in the HPG as a sequence of three sequential clicks, and the rules mined will thereby automatically hold the assumption of being dependent upon the last 3 pages (for more, see Paper 3). Note that the HPG model presented so far in the previous section implicitly used a history depth of 1.

Note, that a rule extracted from an HPG is given with a certain probability. This probability can be translated into a number of sessions supporting (under the HPG assumption) the trail of the rule, called the *expected number of sessions* or $E_{sessions}$, by multiplying this probability with the number of sessions that have visited the first state of the rule. Other definitions of the support of a rule extracted from the HPG could be utilized, e.g., taking into account the minimal number of sessions in any of the states in the rule, but this will not be pursued in this paper.

We are interesting in examining whether it is still possible to use the HPG model to perform data mining. In other words, we wish to examine whether the HPG model, and in particular the hybrid approach, can be extended to output entirely true rules instead of rules which might or might not be false. First we adopt the following definition of an existing true rule:

Given a rule R , consisting of a sequence of states $R = \{s_1, \dots, s_n\}$, and a support level $S_{sessions}$, indicating the number of sessions

which include the sequence of states, R is an **existing rule** if and only if $S_{sessions} \leq 1$.

With this definition, we can also define a **false rule** to be a rule that is not an existing rule, i.e., a rule indicating a sequences of states not traversed by any sessions. We define a true rule as:

Given an existing rule R , consisting of a sequence of states $R = \{s_1, \dots, s_n\}$ with an expected number of sessions $E_{sessions}$, and a support level $S_{sessions}$, indicating the number of sessions which include the sequence of states, R is a **true rule** if and only if $E_{sessions} = S_{sessions}$.

With the assumption of limited browsing history in the HPG model, the expected number of sessions indicated by i given rule are therefore not necessarily the correct support level of the rule.

3.2 Obtaining True Rules

In modifying the hybrid approach, we note that the set of existing rules generated by the HPG model is a superset of the true rules, since a true rule will always be contained in an existing rule. This fact leads us towards an approach where the HPG model is used as a generator of *candidate rules*, which can then be validated (modified) to become true rules. In other words, the rules extracted from the HPG are checked against the session collection to achieve the correct support level, if any. In the following, the set of usage sessions are referred to as the *true traversals*.

From Candidate Rules to True Rules To see how we can go from a number of candidate rules to a number of true rules, we must develop a way of *post-checking* or *validating* the candidate rules in order to convert them to true rules. This conversion is not an altering of the page sequences, since the HPG ensures that the sequence is valid, i.e., it is possible for a session to have traversed the sequence of pages. The conversion is an adjustment of the number of sessions having traversed the page sequence, so the rule indicates the actual support of the page sequence amongst the true traversals. In essence, we want to adjust $E_{sessions}$ to be equal to $S_{sessions}$. The techniques of adjusting the support level is examined in the following.

Using the hybrid approach, we already have the framework in place to perform this conversion of support level. As described in Section 2, we have stored all clickstream information in a click fact schema, so the idea is to

use this schema to validate a given rule. Since the schema retains the ordering of clicks within a session, we wish to develop an automatic approach of validating the extracted rule against the information in the click fact schema.

3.3 Post-checking

Validating a rule in our click fact schema requires the execution of queries in which we join multiple instances of the click fact table. This is necessary since the click fact schema stores the individual clicks and sequences of related clicks can therefore only be formed through such self-join operations. Please note that the following descriptions makes use of the naming used in the schema illustrated in Figure 2.1

We want the check of a candidate rule against the click fact schema to results in the number of sessions containing the given rule. By containing, we mean that a session includes the given sequence of web pages in the rule at least once. If a session contains a sequence more than once, the support is only incremented with one. This definition of support is adopted elsewhere[2].

The conditions to be verified for *every* production $p[s_1 \rightarrow s_2]$ in a rule are the following:

1. s_1 is followed by s_2 (Ex. *products.html* \rightarrow *order.html*)
2. s_2 's `number_in_session` is 1 greater than s_1 's `number_in_session`.
3. s_1 and s_2 are found within the same session.

Furthermore, there are some cases in which additional conditions apply which must be checked as well. The α parameter, used when mining a HPG for (candidate) rules, must also be considered when it is set to 0. Mining with α equal to 0 means that all rules mined from the HPG, are rules in which the first web page is also the first page visited in a session. Furthermore, a rule might include the artificial **end**-state, which is not represented in the click fact schema, in its last production, indicating that the previous state was the last in a session.

Adding these two special case conditions to the list of checks performed in the query, we get:

4. if α is 0, s_1 in $p_1[s_1 \rightarrow s_2]$ *must* have `is_first` equal to 1.
5. if s_2 in $p_n[s_1 \rightarrow s_2]$ is the artificial **end**-state, s_1 *must* have `is_last` equal to 1.

Note that `is_first` and `is_last` is assumed to be implemented as 1 and 0 in the click fact schema, representing true and false respectively. To clarify how a query would look when checking a given rule, a short, but covering example is now presented. Consider the rule $1 \rightarrow 2 \rightarrow 3 \rightarrow E$, where the numbers are the schema identifiers for the specific URLs traversed in the rule, i.e., the `url_key` and `referrer_key` fields of the click fact table. The parameter α is set to 0, and the state E is the artificial end-state. Formally, we must check the productions $p_1[s_1 \rightarrow s_2]$ and $p_2[s_2 \rightarrow s_3]$. The corresponding query used to validate the rule against the click fact schema in the data warehouse can be seen in Figure 3.1.

```

SELECT COUNT(DISTINCT(cf1.session_key))
FROM Click_fact cf1 INNER JOIN Click_fact cf2 ON
(cf1.url_key = 1 AND cf1.url_key = cf2.referrer_key AND cf2.url_key = 2
AND cf1.is_first = 1 AND cf2.number_in_session = cf1.number_in_session+1
AND cf1.session_key = cf2.session_key)
INNER JOIN Click_fact cf3 ON
(cf2.url_key = cf3.referrer_key AND cf3.url_key = 3
AND cf3.is_last = 1 AND cf3.number_in_session = cf2.number_in_session+1
AND cf2.session_key = cf3.session_key)

```

Figure 3.1: Example query for validating an HPG candidate rule.

If α is larger than 0, the check for `is_first` is omitted and for rules not ending with the artificial end-state, the check for `is_last` is omitted. It is *not* possible to remove a join from the query when the check for `is_first` is not needed. Even though checking for the existence of `referrer_key` and `url_key` in the *same* table instance would do the job, condition **2** would be impossible to check as can be seen below.

Condition **2** needs to be checked even though a rule $1 \rightarrow 2 \rightarrow 3 \rightarrow E$ apparently defines the sequence of URLs visited in a rule. A short example, using the rule from above, demonstrates what happens if we do not apply condition **2**. Consider the snippet from a click fact table shown in Figure 3.2.

If we check *only* for conditions **1** and **3** (and the special condition **4**), it can be seen from rows 1, 4 and 6 that the rule $1 \rightarrow 2 \rightarrow 3$ is supported. More specifically, the first production $1 \rightarrow 2$ is found in row 4 (combined with row 1 where `is_first` is 1) followed by the second production $2 \rightarrow 3$ found in row 6. Each of these productions belong to the same session and we might conclude that a user started on page 1, then went to page 2 and then to page 2. However, referring to Figure 3.2, it can be seen that this interpretation is wrong. In between visiting page 1 and 2, a visit on page 5 is registered.

click	referrer_key	url_key	session_key	no._in_session	is_first
1	0	1	10	1	1
2	1	5	10	2	0
3	5	1	10	3	0
4	1	2	10	4	0
5	2	2	10	5	0
6	2	3	10	6	0

Figure 3.2: Click fact snippet used to validate a rule.

If we *did* check condition **2**, we would have been able to exclude the current session from the set of sessions supporting the candidate rule. Note that there are two things in this example that restrain the session from supporting the rule; The check for `is_first` and the existence of circular requests, i.e., that requests for the same page re-appear in the session. Arguably, the existence of circular requests such as the refreshing of a page in row 5 could be interpreted as a general indifferent action which should be omitted when checking for a sequence of page requests. However, we believe that the circular requests of row 5 and of rows 2 and 3 cannot be ignored, since they might contain valuable information concerning the use of a website.

3.4 Constraint Checking

The hybrid framework presents a flexible way to add constraints on rules extracted from the data warehouse. These constraints should also be applied when post-checking a rule, since we should only want to count support for the sessions from which the rule was extracted.

```

SELECT COUNT(DISTINCT(cf1.session_key))
FROM Click_fact cf1 INNER JOIN Click_fact cf2 ON
(cf1.url_key = 1 AND cf1.url_key = cf2.referrer_key AND cf2.url_key = 2
AND cf1.is_first = 1 AND cf2.number_in_session = cf1.number_in_session+1
AND cf1.session_key = cf2.session_key)
INNER JOIN Click_fact cf3 ON
(cf2.url_key = cf3.referrer_key AND cf3.url_key = 3
AND cf3.is_last = 1 AND cf3.number_in_session = cf2.number_in_session+1
AND cf2.session_key = cf3.session_key)
INNER JOIN date_dimension dd ON
(cf1.date_key = dd.date_key AND dd.year = 2002)
INNER JOIN Temp_dimension td ON
(cf1.session_key = td.session_key)

```

Figure 3.3: Example query 3.1 extended with session- and click-specific constraints.

The use of constraints in post-checking of rules is quite similar to the use of constraints when creating the HPG[16]. For *session*-specific constraints a join between the click fact table and the dimension holding the constraint is added to the query. For *click*-specific constraints, the temporary table used when building the HPG, is re-used by adding a join between the temporary table (which holds session-id's fulfilling the click-specific constraints) and the click fact table (see Section 4.1). An example of the use of constraints in post-checking a rule is presented in Figure 3.3 where extra joins on the `date_dimension` and the `temp_dimension` is added to illustrate the use of session- and click-specific constraints, respectively.

4 Experimental Evaluation

The evaluation of the Post-check Expanded Hybrid Approach (PEHA) is aimed at identifying potential bottlenecks and areas in which the approach could be optimized, as well as comparing it with another existing web usage mining approach. Our implementation of PEHA, MIMER³ (see Paper 1), was used for the evaluation. In this section we will describe the experiments and settings used in the evaluation, present and discuss selected results from the experiments conducted and summarize on the experimental results. More experimental results can be found in Appendix A.

4.1 Description of Experiments

Experimental Goals The goal of the experiments is to evaluate the performance of MIMER, which is an implementation of PEHA. We want to evaluate each of the sub-tasks making up the PEHA implemented in MIMER, which will allow us to target those areas in the approach that could benefit from further performance optimizations. We are also interested in scalability issues to examine the usefulness of the approach in a variety of scenarios. We focus specifically on the post-checking sub-task of the system, as the technique presented in Section 3.3 executes multiple joins on the DBMS and therefore could prove a bottleneck as the characteristics of the data changes. This experiment focus on post-checking performance as rules with different length and constraints are fed to the system. We wish to compare PEHA to another technique for extracting usage patterns in order to examine the effectiveness as well as identifying possibly strong and weak sides of using PEHA. We decided to compare the MIMER implementation with queries directly on the *subsession schema*[3, 15]. The subsession schema explicitly stores sequences of clicks (subsessions) from the sessions in the data warehouse and frequent sequences of clicks that are correctly supported can therefore be queried directly. We will refer to the process of querying the subsession schema for frequent subsessions as the *subsession approach*. The number of subsessions stored in the subsession approach grows relative to the number of clicks stored in the click fact table and the average session length.

Two datasets are used in the experiment. Experiments using both MIMER and the subsession approach are conducted for each datasets, namely the performance using three different ways of constraining the extracted rules (for more on different types of constraints, see Paper 1). The three different constraint types are:

³Named after the god in Nordic mythology who guarded the well of wisdom and was a valuable advisory to the main god Odin.

1. **Session specific** : Constraints which apply to *entire* sessions, e.g., the total amount of clicks in a session.
2. **Click specific** : Constraints which apply to individual clicks within a session, e.g., a specific page requested in a session.
3. **Session and Click specific** : A combination of click and session specific constraints.

For each constraint type, the threshold used for mining the HPG is varied. By decreasing the support threshold, we experienced an increasing average rule length and rule-set size for both datasets. In order to examine performance using rule-sets with different characteristics we utilize this property to evaluate the performance for different average lengths of rules. To minimize the effect of distorted results due to various initialization procedures in the implementation of MIMER, each experiment is repeated 5 times and an average is calculated. For details on the queries and the materialized views used in the experiments, refer to Appendix B. The sub-tasks of MIMER that are evaluated in this paper are:

1. **Query** data warehouse for information used to construct the HPG.
2. **Construction** of the HPG structures in main memory.
3. **Mining** the constructed HPG.
4. **Post-checking** the rules extracted from the HPG.

4.2 Experimental Settings

Experimental Data The first web log is from the website of the Computer Science Department at Aalborg University[13], in the following referred to as *CS*, and represents several months of usage. The website is primarily an information site for students and staff at the institution, but also contains personal homepages with a wide range of content and homepages for individual classes.

The second web log is taken from an educational intranet placed at major Danish financial institution, in the following referred to as *Bank*, representing several months of usage. The website contains various courses which can be followed by the employees of the institution. Table 4.1 shows various statistics for the two web logs. For more metadata information on the two web logs, refer to Appendix A.

	CS	Bank
Sessions	68745	21661
Clicks	232592	395913
Unique pages	58232	2368
Average session length	3,38	18,23

Table 4.1: Statistics for the CS and Bank web logs.

Note the differences between the datasets in the average length of a session and the number of unique pages, indicating a more scattered browsing on the CS website as oppose to the more continuous browsing on the Bank website.

Experimental Framework The DBMS used in the MIMER implementation is a large enterprise system supporting materialized views⁴. The DBMS and MIMER is running on an AMD AthlonTM XP1800+ with 512 MB of memory and Windows 2000. MIMER is implemented in Sun’s Java Development Kit 1.3.1 and all tests were performed through dedicated test classes run from the command-line on Windows 2000.

Materialized Views For most SQL queries used in MIMER, matching materialized views[1] are implemented in order to optimize the query performance. Likewise, the queries on the subsession schema are also optimized using materialized views. In the experiments, only rules of length ≤ 10 are post-checked since the materialized views used in MIMER are dedicated views for each rule length and not designed to promote the post-checking of longer rules. The space used to hold the materialized views used by MIMER and the subsession approach are approximately 185 MB and 740 MB respectively (see Appendix B.3 for details).

Insertion of Datasets Each web log is loaded into the data warehouse using a batch load utility. Because of the very time consuming task of constructing and loading very long subsessions, only subsessions of length up to 5 is generated. Notice, that this limit will prevent us from finding rules with a length longer than 5 when using queries on the subsession schema, but as the goals in the experiments are to compare the *performance* and not the *correctness* of rule extraction by the two approaches, we believe that this will not effect our experimental results.

⁴For licensing issues, we cannot name the specific DBMS used.

4.3 Results

This section will describe the results from the experimental evaluation of PEHA. It is divided into a general performance evaluation of the different sub-tasks in the MIMER implementation, an evaluation focusing on the post-checking sub-task of MIMER and a comparison of MIMER against the sub-session approach. All experiments described in this section uses an α value of 1, as this value produces the largest amount of rules and the Bank dataset except in the comparison with sub-session, which includes results from the CS dataset. The Bank dataset is presented here due to better characteristics with regards to extraction of higher number of long rules. Refer to Appendix A for the experimental results using the CS dataset.

4.3.1 MIMER Performance

We expect the results of evaluating the performance of MIMER to indicate a constant performance per rule even as the number and length of rules increase. The primary reason for expecting such constant behavior is the use of materialized views in the post-checking sub-task of MIMER, which we expect will enable us to post-check a rule in near constant time. As the total post-checking time depends on both the number and length of rules, we will present the results in this paper using the average time of post-checking a given rule length. The average time T_{AVG} used to post-check a given rule is calculated as

$$T_{AVG} = \frac{PC_{TOTAL}}{RC_{TOTAL}}$$

where PC_{TOTAL} is the total time used to post-check and RC_{TOTAL} is the total amount of rules checked.

The result of extracting and post-checking rules through MIMER, using the DB dataset with $\alpha = 1$ is shown in Figure 4.1. It shows that MIMER, and all of the sub-tasks in MIMER, performs in approximately constant time even as the average rule length is increased. On average, 75% of the total time is used to extract data from the DBMS and construct the HPG structure in main memory. The subtask of querying the data warehouse for information needed to construct the HPG does not at present utilize materialized views to optimize access to the data. Constructing the HPG in main memory takes minimal time partly due to the fact that dedicated structures to hold both productions and states in the HPG have been implemented. Mining the constructed HPG for (candidate) rules takes relative little time, the figure showing an interval of approximately 50 to 250 milliseconds. The time used to post-check the candidate rules mined in the previous stage runs in constant

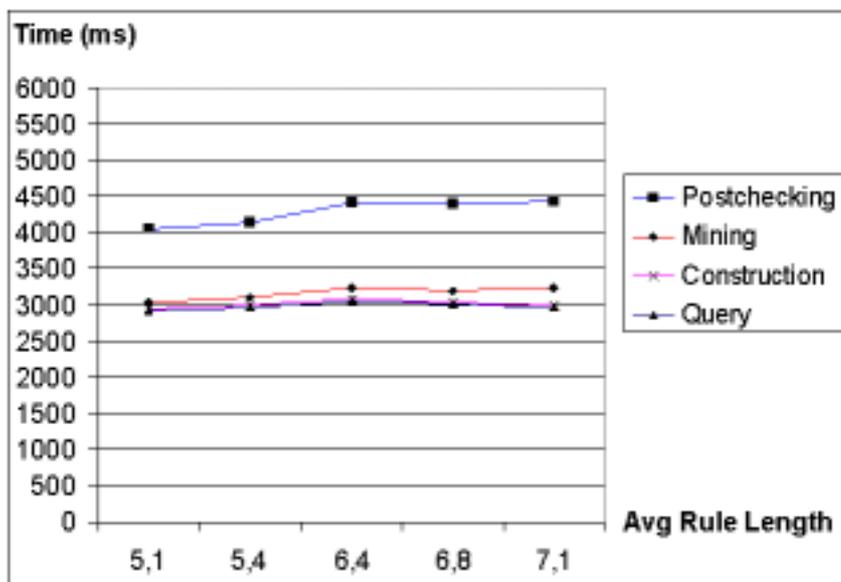


Figure 4.1: MIMER Performance with click- and session-specific constraints.

time as well. It can be seen that the time used to check a rule scales very well with the average rule length, indicating that no matter the length of a candidate rule in the experiment, post-checking only adds a near constant factor to the total time used. Notice that we would not expect such constant behavior to be achieved *without* the use of materialized views in the DBMS. Extracting rules with a lower minimum average rule length than 5,1 was not possible with the Bank dataset.

4.3.2 Post-Checking Performance

We wish to examine the performance on long rules in order to stress the post-checking subtask, i.e., post-checking a larger set of longer rules. We expect that adding more constraints will increase the time spent executing the query on the DBMS, as the materialized views must join with the dimension tables including the columns that are constrained upon. Figure 4.2 shows the results of post-checking candidate rules with the different types of constraints mentioned above.

The figure shows that both for click- and session-specific constraints, the post-check run in almost similar time, indicating that no specific type of single constraint is more time consuming. Performing post-check on a rule for a combined click- and session-specific constraint show a slight increase

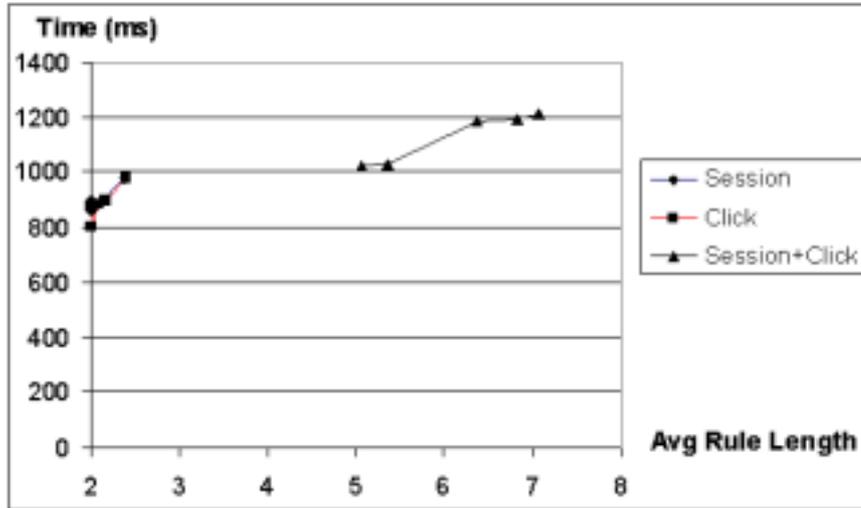


Figure 4.2: Post-Checking performance with different constraint types.

in the running time of approximately 200 ms as the average rule length is increased by 2, indicating that adding constraints on several dimension will increase the running time slightly.

4.3.3 MIMER vs. Subsession

When comparing MIMER to direct queries on the subsession schema, we expect that the more complex nature of MIMER, in particular the hybrid approach, will yield performance degradation in favor of the queries on the subsession schema. Recall that the extraction of rules using the PEHA technique in MIMER involves several subtasks, namely querying the DBMS, building an aggregated structure, mining this structure and at last re-querying the DBMS during the post-check. The results of comparing the running time of MIMER and the subsession approach are shown in Figure 4.3.

The result confirm our expectations, as the queries on the subsession schema are several times faster than MIMER, in this case, using click-specific constraints. The subsession schema thereby proves faster at locating true rules compared to MIMER, primarily because all possible rules (of length ≤ 5) are stored explicitly and finding rules by frequency of occurrence, utilizing materialized views, run relatively fast. In order to run click-specific constraints, the subsession queries use the LIKE statement of the SQL language to extract information “inside” the sequences stored in the schema. Note however, that not all click-specific constraints can be implemented on top

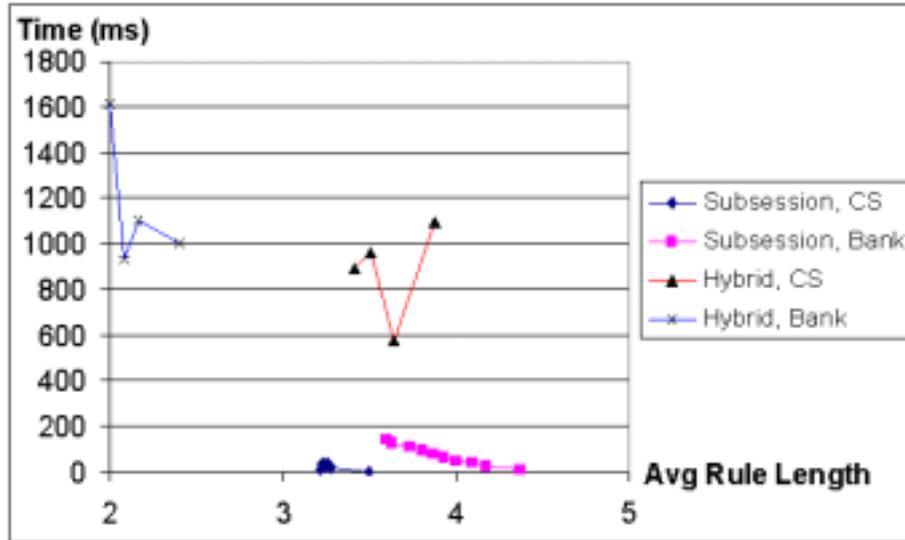


Figure 4.3: MIMER compared to the Subsession Approach for $\alpha = 1$ with click-specific constraints.

of the subsession schema, since some information is too finely grained to be stored in the schema, e.g., the time spent on a particular page, also referred to as the *dwelt-time* of a request.

The tables and materialized views of the subsession approach consume approximately 4,5 times the storage of MIMER for both CS and Bank (see Appendix B.3 for storage details). The extra space used in the subsession approach comes from the fact that each sub-sequence of clicks found in an entire session of clicks is explicitly stored.

4.4 Experimental Summary

From the experimental results, we have found that the average time of post-checking a candidate rule from an HPG run is approximately constant time, regardless of the length of the rule when utilizing materialized views. The actual running time of the post-check is also dependent on the number of constraints placed on the HPG but not on the type of constraint as both click- and session-specific constraints run in similar time. Compared to the direct queries on the subsession schema, MIMER performs several times worse, however the flexibility inherent in PEHA and in MIMER allows for more specialized information to be extracted from MIMER. The difference in space usage, is clearly in favour of the PEHA approach. Notice, that in our experiments

only subsession up to 5 was generated and storing longer subsessions would further increase the overhead in space required by the subsession approach.

5 Conclusion and Future Work

The hybrid approach, a technique for extracting knowledge on the use of a website, is a very flexible method of mining for knowledge information with various constraint settings enforced on the extracted information. The approach utilizes a fine-grained storage schema in a data warehouse and an aggregated structure called the Hypertext Probabilistic Grammar (HPG). In this paper we have presented an expansion to the hybrid approach, called the Post-check Expanded Hybrid Approach (PEHA) that allow for rules with a correct support level to be extracted from large quantities of web log data. The weakness inherent to the HPG model, namely of presenting a distorted set of rules indicating an incorrect support, have been eliminated through the use of a *post-checking* mechanism that on average adds a constant factor to the running time of the existing hybrid approach. An implementation of the PEHA has been incorporated into a prototype, MIMER, and experimental evaluations using this prototype have been performed. Using a large enterprise DBMS supporting materialized views, MIMER has a poorer running time than queries directly on the subsession schema, a rival data warehouse schema explicitly storing sequences of clicks. The difference in performance between the two techniques is approximately 1000 milliseconds bearing in mind that extra fine-grained mining parameters are available in the MIMER prototype. We have found that PEHA compared to the subsession approach scales far better as the number of sessions and their average length increases, if considering the storage requirements. We therefore conclude that PEHA is a competitive web data mining technique especially when storage requirement and flexibility in constraining the extracted information is taken into consideration.

Several areas of interest can be pointed out as possible future work. First, performance for the Query subtask could be improved by implementing the usage of materialized views. Second, as the subsession schema stores correctly supported sequences of click, work on post-checking rules using the subsession schema could prove valuable although not all kind of rules, i.e., rules with certain click-specific constraints, can be check using the subsession schema. A possible solution could be to validate rules using a combination of the two schemas. Since one of the main features of PEHA is the very flexible way of constraining extracted information, the use of materialized views might pose a problem, especially with regards to storage requirements. In the present prototype materialized views are dedicated, however construction of more general and flexible materialized views are a definite area of further research. Furthermore, work could be focused on the creation of a specialized structure

useful for the post-checking mechanism. Comparing the PEHA against other competing web usage mining approaches such as PrefixSpan[14] could also serve as a important contribution.

References

- [1] H. F. Korth A. Silberschatz and S. Sudarshan. *Database System Concepts*. McGraw-Hill, 2002.
- [2] R. Agrawal and R. Srikant. Mining Sequential Patterns. In *Proceeding of the 11th International Conference on Data Engineering, ICDE*. IEEE Press, 1995.
- [3] J. Andersen, A. Giversen, A.H. Jensen, R. S. Larsen, T. B. Pedersen, and J. Skyt. Analyzing Clickstreams Using Subsessions, Technical Report 00-5001. Technical report, Department of Computer Science, Aalborg University, 2000.
- [4] A.G. Büchner, S.S. Anand, M.D. Mulvenna, and J.G. Hughes. Discovering Internet Marketing Intelligence through Web Log Mining. In *UNICOM99 Data Mining and Datawarehousing*, 1999.
- [5] J. Borges. *A Data Mining Model to Capture User Web Navigation Patterns*. PhD thesis, Department of Computer Science, University College London, 2000.
- [6] J. Borges and M. Levene. Data Mining of User Navigation Patterns. In *WEBKDD*, 1999.
- [7] J. Borges and M. Levene. Heuristics for Mining High Quality User Web Navigation Patterns. Research Note RN/99/68. Department of Computer Science, University College London, Gower Street, London, UK, 1999.
- [8] J. Borges and M. Levene. A Fine Grained Heuristic to Capture Web Navigation Patterns. *SIGKDD Explorations*, 2(1), 2000.
- [9] E. Charniak. *Statistical Language Learning*. MIT Press, 1996.
- [10] R. Cooley, J. Srivastava, and B. Mobasher. Web Mining: Information and Pattern Discovery on the World Wide Web. In *9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, 1997.
- [11] R. Cooley, P. Tan, and J. Srivastava. Websift: the Web Site Information Filter System. In *1999 KDD Workshop on Web Mining, San Diego, CA.*, 1999.
- [12] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.

- [13] Aalborg University Department of Computer Science. <http://www.cs.auc.dk>.
- [14] Pei J., Han J., Mortazavi-Asl B., Pinto H., Chen Q., Dayal U., and Hsu M.-C. Prefixspan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In *2001 International Conference of Data Engineering (ICDE), Heidelberg, Germany, 2001*.
- [15] J. Andersen, A. Giversen, A. H. Jensen, R. S. Larsen, T. B. Pedersen, and J. Skyt. Analyzing Clickstreams Using Subsessions. In *International Workshop on Data Warehousing and OLAP, 2000*.
- [16] S. E. Jespersen, T. B. Pedersen, and J. Thorhauge. A Hybrid Approach to Web Usage Mining - Technical Report R02-5002. Technical report, Department of Computer Science, Aalborg University, 2002.
- [17] R. Kimball and R. Merz. *The Data Webhouse Toolkit*. Wiley, 2000.
- [18] M. Levene and G. Loizou. A Probabilistic Approach to Navigation in Hypertext. *Information Sciences*, 114(1-4), 1999.
- [19] J. Pei, J. Han, B. Mortazavi-asl, and H. Zhu. Mining Access Patterns Efficiently from Web Logs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2000*.
- [20] M. Spiliopoulou and L. C. Faulstich. WUM: a Web Utilization Miner. In *Workshop on the Web and Data Bases (WebDB98), 1998*.

A Additional Experimental Results

This appendix contains experimental results that have not been presented in the main paper.

A.1 MIMER Performance

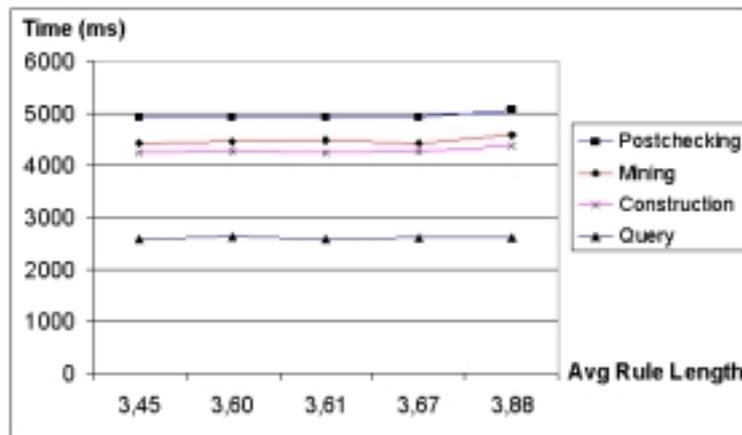


Figure A.1: MIMER performance for the CS dataset, $\alpha = 1$ using session-specific constraints.

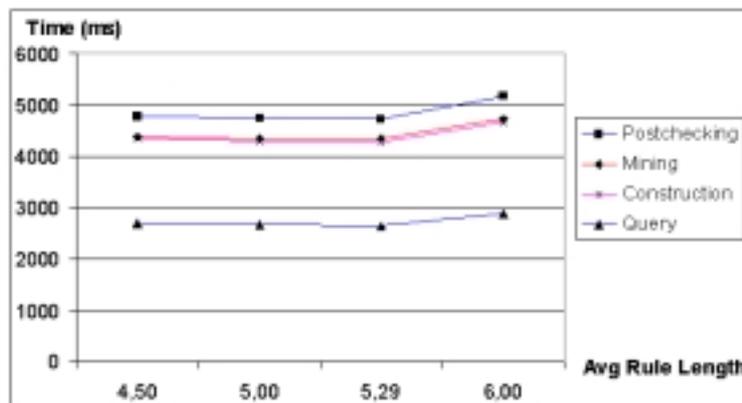


Figure A.2: MIMER performance for the CS dataset, $\alpha = 0$ using session-specific constraints.

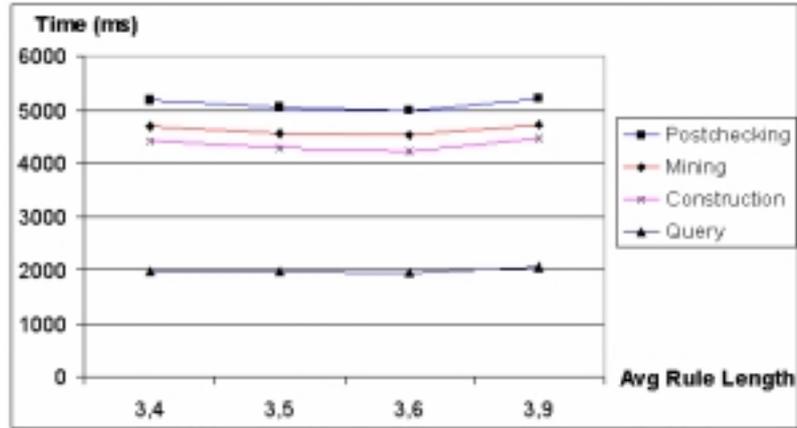


Figure A.3: MIMER performance for the CS dataset, $\alpha = 1$ using click-specific constraints.

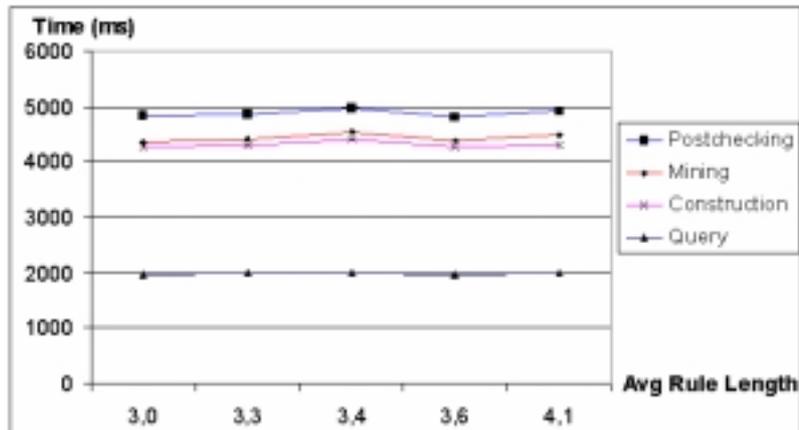


Figure A.4: MIMER performance for the CS dataset, $\alpha = 0$ using click-specific constraints.

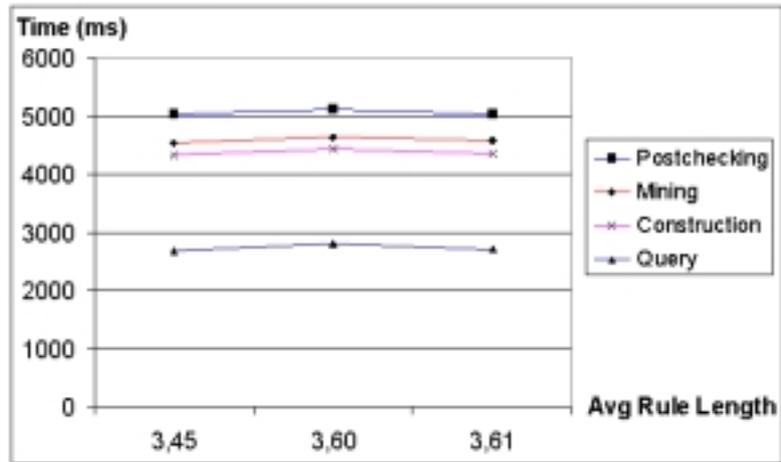


Figure A.5: MIMER performance for the CS dataset, $\alpha = 1$ using session- and click-specific constraints.

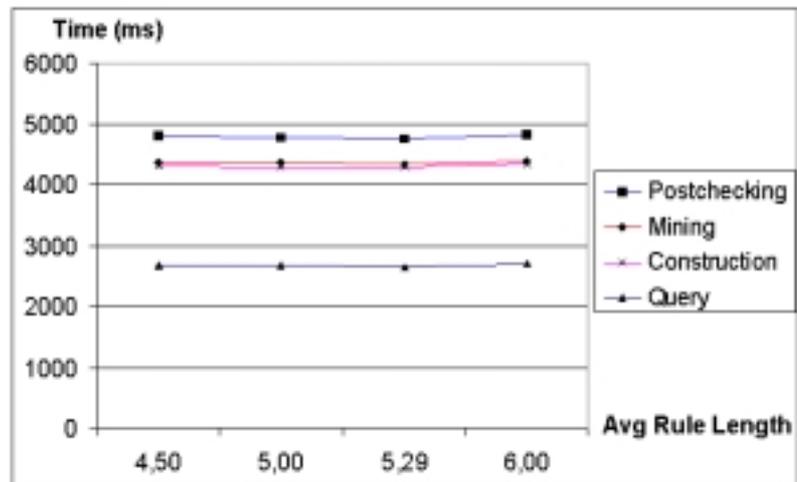


Figure A.6: MIMER performance for the CS dataset, $\alpha = 0$ using session- and click-specific constraints.

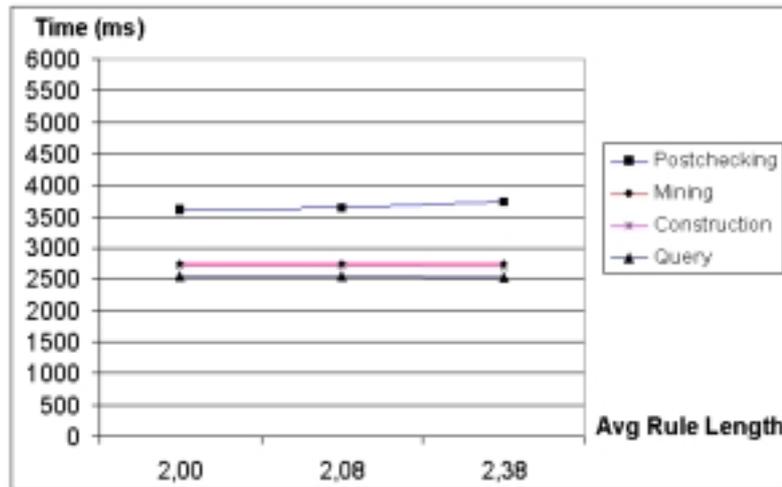


Figure A.7: MIMER performance for the Bank dataset, $\alpha = 1$ using session-specific constraints.

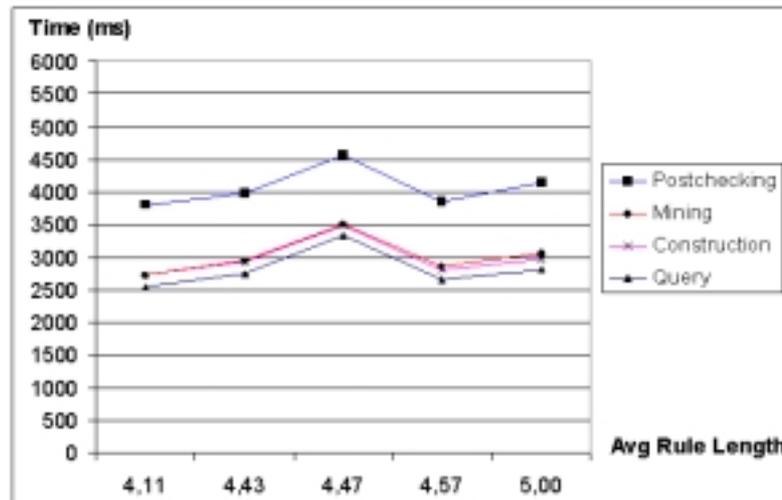


Figure A.8: MIMER performance for the Bank dataset, $\alpha = 0$ using session-specific constraints.

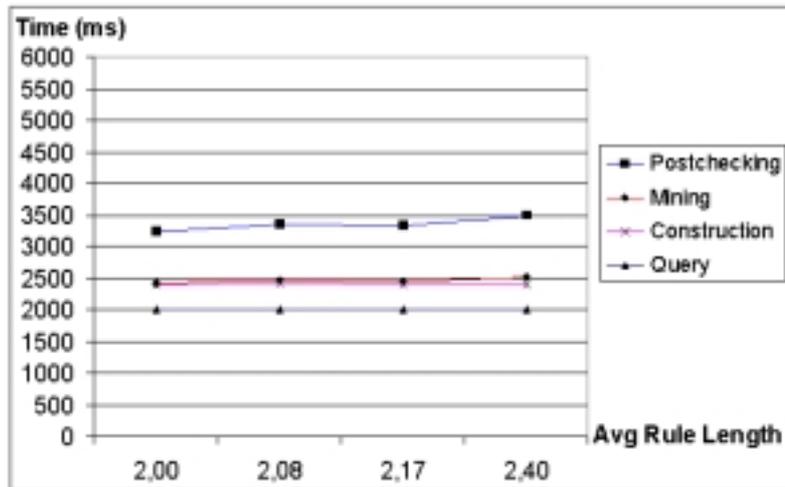


Figure A.9: MIMER performance for the Bank dataset, $\alpha = 1$ using click-specific constraints.

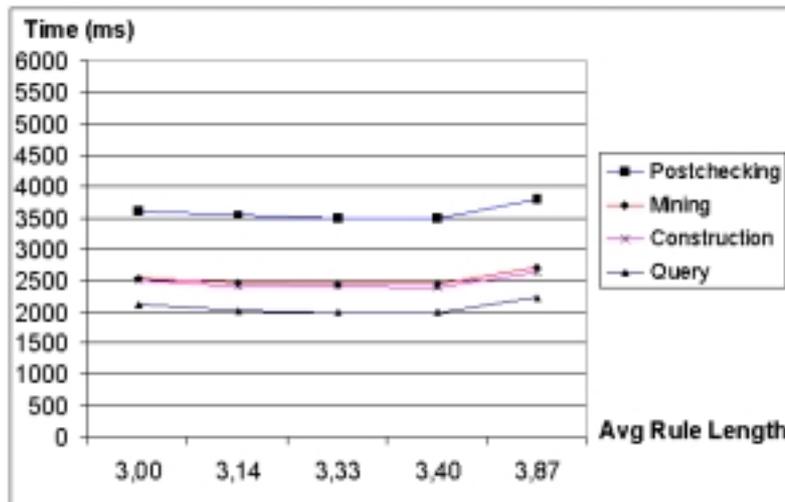


Figure A.10: MIMER performance for the Bank dataset, $\alpha = 0$ using click-specific constraints.

A.2 Post-checking Performance

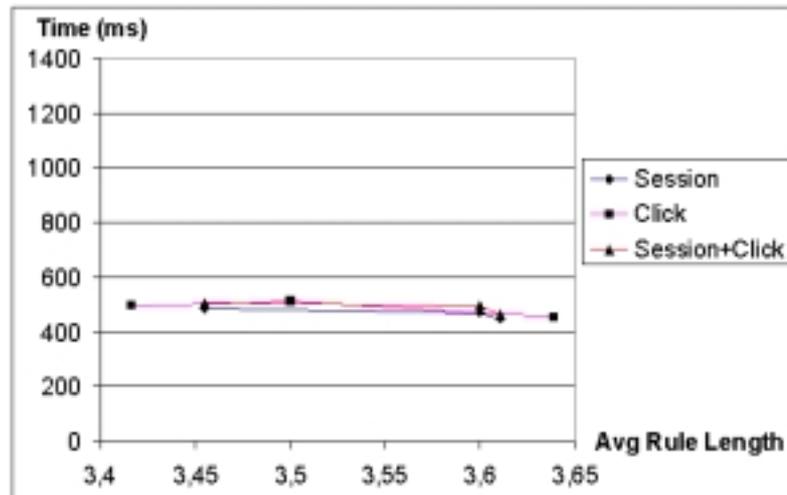


Figure A.11: Post-check performance for the CS dataset, $\alpha = 1$.

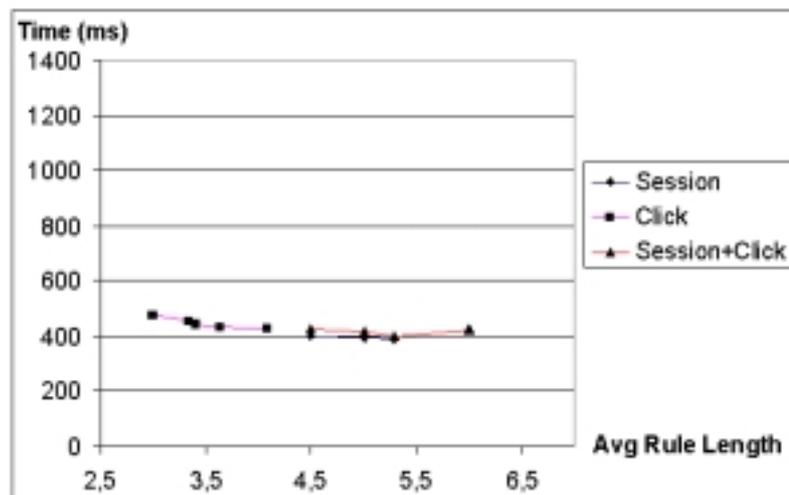


Figure A.12: Post-check performance for the CS dataset, $\alpha = 0$.

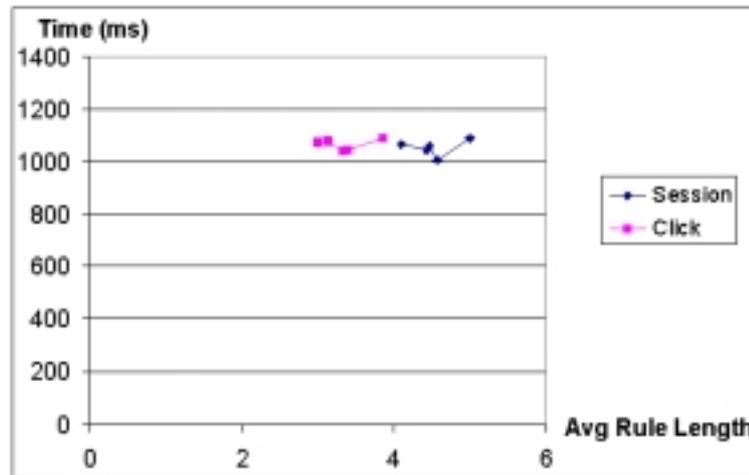


Figure A.13: Post-check performance for the Bank dataset, $\alpha = 0$.

A.3 Hybrid vs. Subsession

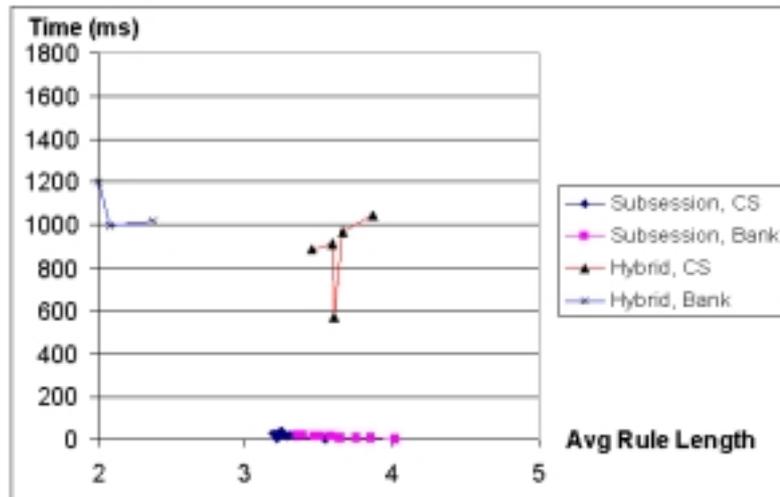


Figure A.14: MIMER compared to the Subsession approach for $\alpha = 1$ with session-specific constraints.

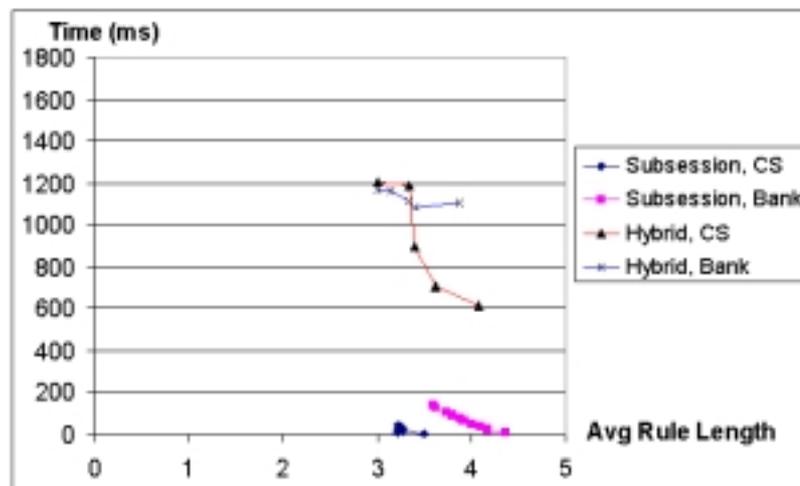


Figure A.15: MIMER compared to the Subsession approach for $\alpha = 0$ with click-specific constraints.

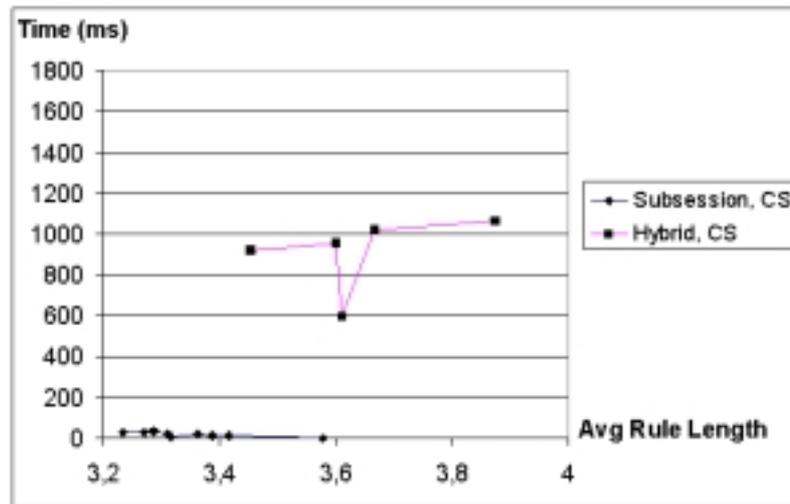


Figure A.16: MIMER compared to the Subsession approach for $\alpha = 1$ with click and session specific constraints on the CS dataset.

B Database-related Information

B.1 Subsession Example Query

Figure B.1 gives an example query to find rules in which the page `/courses/kreditdk/01intro.asp` has been requested and where the number of clicks in the session is more than 10. Furthermore, only rules with a support of minimum 4 sessions are extracted.

```
SELECT DISTINCT(url_sequence_dimension.url_sequence),
url_sequence_dimension.length, count(*) as numbers
FROM subsession_fact sf1
INNER JOIN url_sequence_dimension
ON (sf1.url_sequence_key = url_sequence_dimension.url_sequence_key)
INNER JOIN session_dimension
ON (sf1.session_key = session_dimension.session_key
AND session_dimension.start_page = '/courses/kreditdk/01intro.asp'
AND session_dimension.session_clicks > 10)
GROUP BY url_sequence, length
HAVING count(*) > 3
ORDER BY numbers DESC, length DESC
```

Figure B.1: Example query for the subsession approach.

B.2 Materialized View Example

The following examples serves to clarify the exact details in which materialized views are used to optimize query performance in MIMER and the subsession approach. Not all materialized views used in the implementation are shown as similarity between certain queries is high.

B.3 DBMS Statistics

Table B.1 shows the number of bytes occupied by the tables and materialized views used in the DBMS used by both MIMER and the subsession approach. The table names `1_1`, `2_1`, ... denote the materialized views used by the post-checking sub-task in MIMER for checking rules of length 1, 2, respectively. The table names `Sub_1`, `sub_2` and `sub_3` denotes the materialized views used in the subsession approach for the three different queries executed for each constraint setting. The total size used by MIMER is calculated as the sum of the materialized views `1_1`, `2_1`, ..., the `click_fact` table and all dimension tables referred to from the `click_fact` table. The subsession size is calculated as the sum of the materialized views `Sub_1`, `sub_2` and `sub_3`,

```

SELECT cf1.session_key, cf1.url_key u1, cf1.is_first, cf2.url_key u2,
cf3.url_key u3, cf3.is_last, session_dimension.start_page,
session_dimension.session_clicks
FROM Click_fact cf1
INNER JOIN Click_fact cf2
ON (cf1.url_key = cf2.referrer_key
AND cf2.number_in_session = cf1.number_in_session+1
AND cf1.session_key = cf2.session_key)
INNER JOIN Click_fact cf3
ON (cf2.url_key = cf3.referrer_key
AND cf3.number_in_session = cf2.number_in_session+1
AND cf2.session_key = cf3.session_key)
INNER JOIN session_dimension
ON (session_dimension.session_key = cf1.session_key)

```

Figure B.2: Example of Materialized View definition for post-checking a rule of length 3 (3_1 in Table B.1).

```

SELECT url_sequence_dimension.url_sequence, url_sequence_dimension.length,
session_dimension.start_page, session_dimension.session_clicks
FROM subsession_fact sf1
INNER JOIN session_dimension
ON (sf1.session_key = session_dimension.session_key)
INNER JOIN url_sequence_dimension
ON (sf1.url_sequence_key = url_sequence_dimension.url_sequence_key)

```

Figure B.3: Example of Materialized View definition for supporting the sub-session approach with click- and session-specific constraints (sub_3 in Table B.1).

the subsession_fact table and finally all dimension tables referred to from the subsession fact table.

Table / Materialized View	Space used in bytes (Bank)	Space used in bytes (CS)
1_1	17024259	10699232
2_1	17570151	8192350
3_1	18111630	7477800
4_1	18245898	7084380
5_1	18603964	6780475
6_1	18967164	6433767
7_1	19001905	6234796
8_1	19251828	6057088
9_1	19226448	5888820
10_1	19454404	5725637
sub_1	261649143	93986100
sub_2	217348759	76755315
sub_3	261649143	93986100
click_fact	13393042	8373312
subsession_fact	42915997	16186495
date_dimension	5369	2600
session_dimension	1971151	6324540
timeofday_dimension	618527	957164
timespan_dimension	27424	34173
url_dimension	134976	3493920
url_sequence_dimension	116933230	94592191
MIMER total size	201608140	89760054
Subsession total size	903118743	382824678

Table B.1: DBMS Storage information for the datasets used in the experimental evaluation.

Paper 3

Investigating the Quality of the Hypertext Probabilistic Grammar Model

Investigating the Quality of the Hypertext Probabilistic Grammar Model

Søren E. Jespersen Jesper Thorhauge

13th June 2002

Abstract

As the mass of information on the Internet grows, companies and institutions are faced with a major challenge in discovering knowledge on the use of their website. The Hypertext Probabilistic Grammar is a compact, aggregated structure used in web usage mining, however the model allows the extracted knowledge, rules, to be distorted compared to the patterns in the actual usage data, referred to as the true traversal patterns. This paper examines the similarity and accuracy of the knowledge extracted from the model compared to the knowledge found in the true traversal patterns of two websites. The results indicate that a large number of rules needs to be considered to achieve a high quality, that long rules are generally more distorted than shorter rules and that the model yield knowledge of a higher quality when applied to more random usage patterns.

1 Introduction

The Internet is constantly growing as more information becomes available on-line. As the amount of information on-line grows and becomes more increasingly business critical, companies and institutions are faced with a major challenge in discovering knowledge on the use of their website. Such usage knowledge could be utilized to enhance the design and usability of a website or to better target advertising to individual users. The research into *web data mining* is focused on the discovery of knowledge in a web setting and *web usage mining* is focused on the discovery of knowledge on the usage of websites. Several approaches to web usage mining have been proposed, including aggregated structures and database techniques for efficient knowledge representation. The Hypertext Probabilistic Grammar(HPG)[2, 3] is

one such compact, aggregated structure that extracts *rules*, i.e., highly probable traversals of the website. However, the HPG model aggregates the usage data into inherently independent states and therefore rules extracted from the HPG could indicate an incorrect support compared to the rules found in the usage data, referred to as the *true traversals patterns*. An extension to the HPG model, referred to as *history depth*, introduces additional states in the grammar in order to ensure rules with a better support in the true traversal patterns.

This paper is focused on examining the characteristics of the knowledge extracted from the HPG when used for mining web usage knowledge. The paper examines the quality of the rules of the HPG model compared to the true traversal patterns, using various degrees of history depth. The quality is examined using a measure of similarity and a measure of accuracy and the paper features experiment with the HPG model applied to two websites featuring different browsing characteristics. The experiments indicate that the HPG model extract a certain number of rules that are not similar or accurate compared to the true traversal patterns. Experiments indicate an increase in the similarity of the rules as additional history depth is used, but not affects the accuracy of the rules. Experiments with the different websites indicate that the HPG model extracts more similar and accurate knowledge when applied to a website where browsing patterns are of a more random and scattered nature.

In related work, other aggregated structures[13, 15] have been suggested in the field of web usage mining as well as other techniques. One line of research is focused on extracting usage patterns directly from the web log written by the web server[7, 8] which effectively limits the flexibility of the knowledge discovery process. Another line focuses on using database technologies to perform *clickstream analysis*[1, 10, 12], but existing data warehouse schemas cannot efficiently be used for mining longer sequences of knowledge. A novel approach called the *hybrid approach*[11] (see Paper 1) combines a click-oriented data warehouse schema with the HPG model to enable efficient, constraint-based knowledge discovery. The hybrid approach inherits the possibility of presenting unsupported knowledge but an extension has been developed to verify the knowledge in extracted rules against the usage data (see Paper 2).

We believe this to be the first paper that investigates the quality of the knowledge extracted using the HPG model, and in particular the assumption of a limited browsing history and the addition of history depth. We estimate quality using measures of similarity and accuracy between knowledge discovered on the HPG and knowledge present in the usage data in the web log.

The remainder of this paper is structured as follows. Section 2 describes the HPG model. Section 3 elaborates on the assumption of a limited browsing history inherent in the HPG model. Section 4 describes expanding the HPG model with history depth. Section 5 defines the measures between HPG rules and the true traversals used in the experiments. Section 6 presents the experiments and discuss the results. Section 7 concludes on the experimental results and presents future work.

2 Hypertext Probabilistic Grammars

The nature of web sites, web pages and link navigation has a nice parallel which proves rather intuitive and presents a model for extracting information about user sessions. The model uses a Hypertext Probabilistic Grammar(HPG)[2] that rests upon the well established theoretical area of language and grammars. In the following, when mentioning the terms state and production in relation to a grammar, we will implicitly be speaking of the equivalent Deterministic Finite Automata of the grammar. We will present this parallel using the example in Figure 2.1.

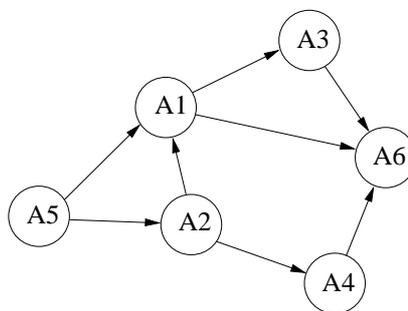


Figure 2.1: Example of a website structure.

The figure shows a number of web pages and the links that connect them. As can be seen, the structure is very similar to a grammar with a number of states and a number of productions leading from one state to another. It is this parallel that the HPG model explores. The model maps web pages to grammar states¹ and adds two additional artificial states, the start state S and the end state F, to form all states of the grammar. We will throughout the paper use the terms state and page interchangeably.

The probability of a production between two states is assigned based on the information in the web log. The probability of a production is proportional to the number of times the link between the two pages was traversed relative to the number of times the state on the left side of the production was visited overall. Note that not all links within a web site may have been traversed so some of the links might not be represented as productions in an HPG. An example of an HPG built from the use of the website is shown in Figure 2.2.

The probability of a string in the language of the HPG can be found by multiplying the probabilities of the productions needed to generate the

¹This is only true if the HPG is created with a history depth of 1, see later.

string, thereby implicitly assuming independency between states in the grammar. Note that web pages might be linked in a circular fashion and therefore the language of the HPG could be infinite. An HPG specifies a threshold η against which all strings are evaluated. Only strings with probability above the threshold is included in the language of the HPG (with the given threshold), L^η .

Mining an HPG is essentially the process of extracting high-probability strings from the grammar. These strings are called *rules*.² These rules will describe the most preferred trails on the web site since they are traversed with a high probability. Mining can be done using both a breath-first and a depth-first search algorithm[2]. The parameter α is used in the mining process to specify what weight should be given to the first states that are requested first in a user session and α span from 0 (rules must begin with a state that was first in a user session) to 1 (all requests are weighted equally).

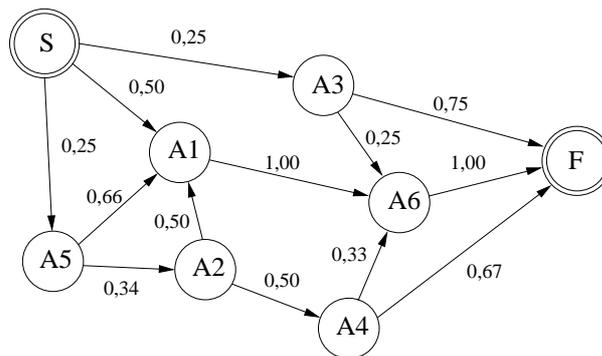


Figure 2.2: Example of a Hypertext Probabilistic Grammar.

The mining of rules on the HPG using a simple breath-first search algorithm has been shown to be too imprecise for extracting a manageable number of rules. Heuristics have been proposed to provide more control of the rules mined from an HPG[4, 5]. The heuristics are aimed at specifying parameters that more accurately and intuitively presents relevant rules mined from an HPG and allow for, e.g., generation of longer rules or only returning a high probability subset of the complete rule-set.

An important characteristic of the HPG model is the inherent assumption that the choice of the next page to browse only depends on the current state in the HPG, i.e., the independency between states. This assumption means that the language of an HPG can include strings corresponding to trails not included in any of the true traversals. This characteristic is further

²The notion of a rule and a string will be used interchangeably

investigated in Section 3. The HPG can be extended with the Ngram[6] concept to improve the precision of the model, by introducing an added *history depth* in the model which effectively determines the length of the assumed user memory when browsing the website. This extension is further investigated in Section 4.

Complexity Analysis The complexity considerations surrounding the creation and extraction of rules from the HPG can be split into two separate parts, which are presented below. In the following, P represents the number of productions and S represents the number of states.

Constructing the HPG: The HPG is a compact, aggregated representation and the size is dependent upon the number of productions it must represent, not the number of sessions since they are aggregated into each state. To construct the HPG, each production needs to be used in order to represent the full HPG. Assuming the use of a hash table holding the productions and a constant time lookup in a hash table[9], the HPG can be constructed with the complexity $O(P)$.

Mining the HPG: The mining of an HPG can be performed using both a general Breath First Search(BFS) and a Depth First Search(DFS) algorithm[2]. The complexity of mining an HPG using the BFS algorithm is $O(S + P)$ [9]. Note that the prototype implementation used in the experiments (see Section 6) uses BFS because of a more efficient memory usage[2].

3 The Markov Assumption

In this section we describe the assumption of independency between states which is inherent in the HPG model, present examples of a certain problem when used to extract knowledge from the usage of a website and summarize the pros and cons of the assumption in the context of web usage mining.

3.1 Markov Assumption Definition

The *Markov assumption* in the HPG is inherent since the probabilities of an HPG corresponds to the probabilities of the transition matrix in a *Markov chain*[14]. A Markov chain consists of a set $S = \{s_1, s_2 \dots s_n\}$ called the *state space*. We have a set of *transition probabilities* p_{ij} , between two adjacent states i and j , where $\sum_j^n p_{ij} = 1$ for every i . In the context of web usage mining and the HPG model, the state space is equal to the set of pages making up a web site, and the transition probabilities corresponds to the probability that a link between two inter-connected pages is traversed.

In a Markov chain we move from state s_i to state s_j with probability p_{ij} after one unit of time k , meaning that we move from s_i at time $t = k$ to s_j at $t = k + 1$. Figure 3.1 illustrates the Markov assumption. The state at time $t = k + 1$ *only* depends on the state at $t = k$ and *not* the state at $t = k - 1$. The probability p_{ij} of going from s_i to s_j follows this rule; it only depends on the current state at time $t = k$. In the following, the Markov assumption will refer to the described assumption of a Markov chain.

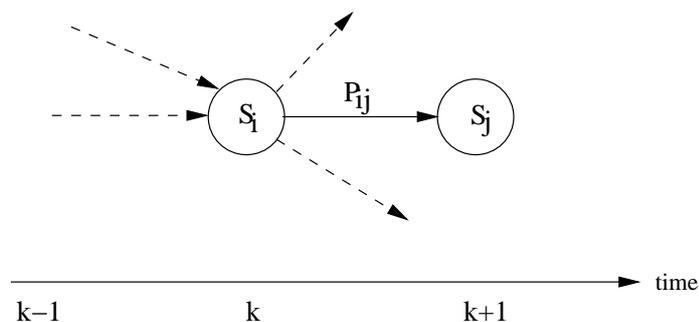


Figure 3.1: Illustrating the Markov assumption.

The formal definition of the Markov assumption using random variables and conditional probabilities is as follows: Let X_0, X_1, X_2 be random variables taking values from the state space $S\{s_1, s_2, \dots s_n\}$, using the following property.

$$P(X_{n+1} = s_j | X_n = s_i, X_{n-1} = s_{n-1}, \dots, X_0 = s_{j_0}) = P(X_{n+1} = s_j | X_n = s_i).$$

The Markov assumption, namely that “the future state only depends on the present state” might lead to a certain problem when applied in a real world case. We will illustrate this problem using two examples; one where the assumption gives correct behavior and one where it does not.

3.2 Usage Examples

Example 1 Consider the following four traversals of a very simple web site:

1. \rightarrow products.jsp \rightarrow addtobasket.jsp \rightarrow order.jsp \rightarrow
2. \rightarrow news.jsp \rightarrow addtobasket.jsp \rightarrow order.jsp \rightarrow
3. \rightarrow products.jsp \rightarrow addtobasket.jsp \rightarrow order.jsp \rightarrow
4. \rightarrow products.jsp \rightarrow order.jsp \rightarrow

The corresponding HPG constructed from these traversals is shown in Figure 3.2. Note that the state space $S = \{\text{products.jsp}, \text{addtobasket.jsp}, \text{order.jsp}, \text{news.jsp}\}$ is named p , a , o and n in the figure.

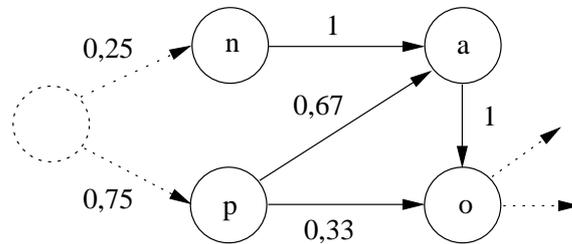


Figure 3.2: HPG for example 1.

As Table 3.1 shows, three rules can be mined from the HPG. Compared to the list of true traversals above, we see that the rules exactly match these traversals and thereby, the presence of the Markov assumption does not interfere with the extraction of supported rules in this example, i.e., the extracted rules match the true traversals.

Rank	Rule	Probability %
1	products.jsp→addtobasket.jsp→order.jsp	50
2	news.jsp→addtobasket.jsp→order.jsp	25
2	products.jsp→order.jsp	25

Table 3.1: Rules mined from example 1.

Example 2 Now, consider two other traversals of a web site. The structure of the website is the same as in example 1 apart from two new pages *gifts.jsp* (*g*) and *specials.jsp* (*s*) which have been added to the set of pages.

1. → news.jsp → addtobasket.jsp → specials.jsp → order.jsp →
2. → products.jsp → addtobasket.jsp → specials.jsp → gifts.jsp →

Figure 3.3 shows the HPG constructed from the traversals along with the calculated transition probabilities.

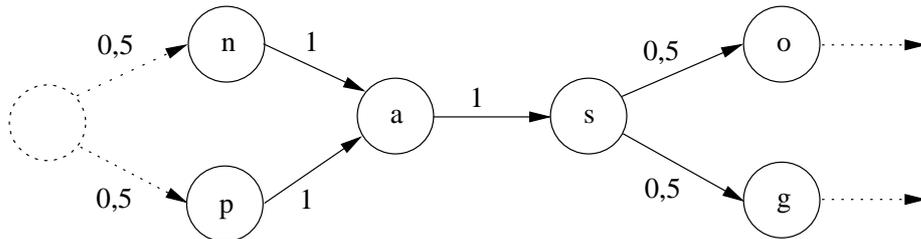


Figure 3.3: HPG for example 2.

Table 3.2 shows the result of mining the HPG for rules. We find 4 rules, each with 25% probability - although there are actually only two traversals represented in the HPG. The true traversals indicate 2 rules, each with a 50% probability. This example shows that the Markov assumption in the HPG model will distort the resulting rules compared to the true traversal. Notice however, that the correct set of rules, i.e., the true traversal patterns, is a subset of the complete set of rules found by mining the HPG, not considering the probability.

Pros Under certain circumstances the Markov assumption can be applied without causing distorted rules. Our first example shows one such situation where rules are extracted that exactly match the true traversals even though the HPG inherently includes the Markov assumption. In the context of

Rank	Rule	Probability %
1	products.jsp → addtobasket.jsp → specials.jsp → gifts.jsp	25
1	products.jsp → addtobasket.jsp → specials.jsp → order.jsp	25
1	news.jsp → addtobasket.jsp → specials.jsp → gifts.jsp	25
1	news.jsp → addtobasket.jsp → specials.jsp → order.jsp	25

Table 3.2: Rules mined from example 2.

web usage mining, one could claim that for certain browsing behaviors, the distortion of rules (especially for long traversals) actually produces a usable result. For instance, when browsing a website without a specific goal, the user's next choices would probably not depend on the choice taken previously. In such scenario, knowledge relying on the assumption of limited browsing history would be useful to indicate the overall browsing patterns even though the knowledge is not entirely supported by the true traversals.

Cons According to the result of the second example, even when we use sessions of a relatively short length, the result is distorted compared to the true traversal patterns. The degree of distortion in our example - if we look at the number of true versus false rules - is 50%, that is, only half the rules mined from the HPG are actually supported in the true traversals. Note also, that the support level indicated by a rule extracted from the HPG can also be distorted compared to the true traversals. One solution could be to verify each mined rule against the true traversals. An automated check of the rules has been developed in which the rules extracted from an HPG is *post-checked* against the usage data stored in a click fact schema in a data warehouse. As the number of rules extracted from the HPG grows, the task of checking every rule mined will present a relatively time and space consuming job. For more on post-checking, see Paper 2.

3.3 Summary

As described, the rules extracted from the HPG model will potentially be distorted compared to the true traversals, even though in some cases this property is desired. In an effort to overcome the distorted rules of the HPG, the model might be extended to include more information in the single state

and thereby effectively increasing the precision of the model. This aspect is covered in Section 4.

4 History Depth Extension

The HPG model described so far has mapped a web page to a state in the constructed HPG and thereby assumed a user memory (history depth) of a single request in the model. Extending the HPG with a history depth above 1 has been proposed by Borges[2] and utilizes the Ngram[6] concept. By extending the HPG using this technique, more than the “current web page” is included in a state in the grammar. This section will briefly describe the concept of added history depth and then re-evaluate the examples given in Section 3. Finally, we will summarize the effects of extending the HPG model with this technique and describe the added complexity.

4.1 Adding History Depth

From the Markov assumption illustration in Figure 3.1, we see that in order to calculate the probability p_{ij} of going from state s_i to state s_j , the information at time k , that is, state s_i is used. So far we have assumed that a single web page corresponds to a state in the grammar. This corresponds to a history depth of 1, since only a single web page is used to the calculate probability of a state transition. Extending the information in state s_i to hold more than a single request for a web page, i.e., holding a sequence of web page requests instead, will mean that the support for the sequence of web pages within a state will correspond to the number of traversals of this sequence in the true traversals. However, the independency between states still remains. The history depth refers to the number of web pages requested in sequence that is mapped into each state in the HPG.

Figure 4.1 shows the principle of using a history depth of 2, where $S_h S_i$ describes holding information for the two web pages S_h and S_i requested in sequence within a single state in an HPG with history depth 1. Note that the two states are overlapping on the request of web page S_i .

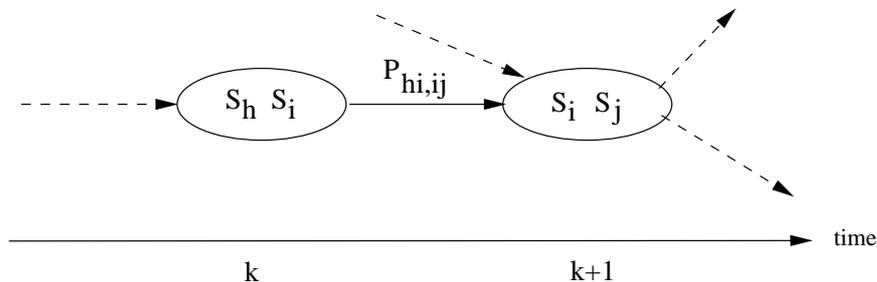


Figure 4.1: Markov assumption with history depth 2.

Extending the history depth, we would expect to increase the precision of the model, i.e., the probability that extracted rules will be closer to the true traversals, since the individual states hold a larger number of web pages that have been requested in sequence in the true traversals. We now re-evaluate the examples presented in Section 3.2.

4.2 Usage Example

For each example re-stated in this section, we extend the HPG model to use history depths 2 and 3 to show the differences in results obtained when mining the structure.

Example 1 Extending the HPG structure shown in Figure 3.2 with history depth 2 and 3 result in the HPG models shown in Figure 4.2.

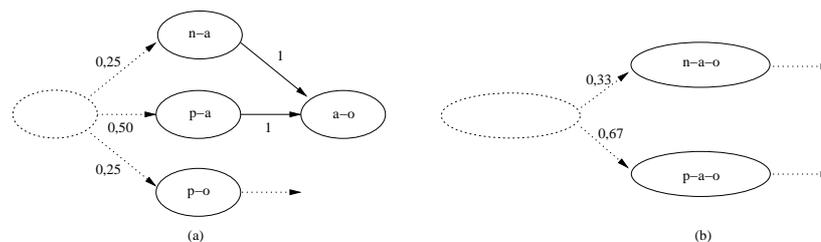


Figure 4.2: HPG for example 1 with history depth 2 (a) and 3 (b).

Table 4.1 shows the result of mining the HPGs in Figure 4.2. In Section 3.2 we found that all correct rules were mined simply by using an HPG history depth equal to 1. Therefore, we would expect to obtain the same result from using an HPG extended with history depth ≤ 1 . It should also be noted, that no rules can be extracted from an HPG with history depth N that could not be extracted from an HPG with history depth $N - 1$, for $N > 1$, since the states in the HPG modeled with history depth N consists of the information present in the adjacent states in the HPG with history depth $N - 1$. Therefore, no new rules will be generated by adding history depth to the model.

With the added history depth, Figure 4.2(b) shows that true traversals with a length below the history depth cannot be represented in the HPG. In the figure, the traversal $[\rightarrow \text{products.jsp} \rightarrow \text{order.jsp}]$ is not represented and therefore the HPG using history depth 3 cannot be mined for all the true traversals.

History Depth	Rank	Rule	Probability %
2	1	products.jsp→addtobasket.jsp→order.jsp	50
2	2	news.jsp→addtobasket.jsp→order.jsp	25
2	2	products.jsp→order.jsp	25
3	1	products.jsp→addtobasket.jsp→order.jsp	67
3	2	news.jsp→addtobasket.jsp→order.jsp	33

Table 4.1: Rules mined from example 2.

Example 2 In the second example using a history depth of 1, as shown in Figure 3.3, only a subset of the resulting rules were correct compared to the true traversals, see Table 3.1. As we add history depth above 1 to the HPG model, we expect to avoid (at least part of) the results that are incorrect, since the states used in the mining process now include additional user memory. Figure 4.3 shows the HPG structure extended with a history depth of 2.

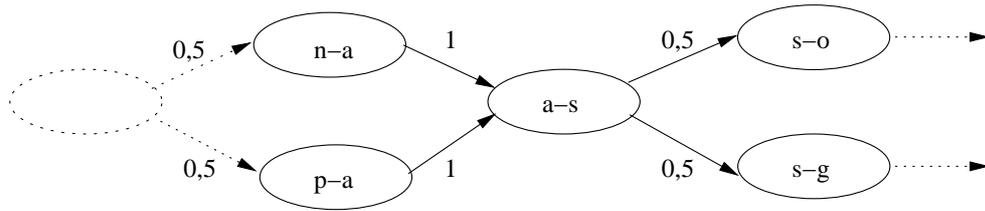


Figure 4.3: HPG for example 2 with history depth 2.

When we mine Figure 4.3 for rules, we get the exact same result as in Table 3.2. Adding a history depth of 2 to the HPG does not solve the problem, since the history depth is too small to avoid the independency between states that produce the false rules.

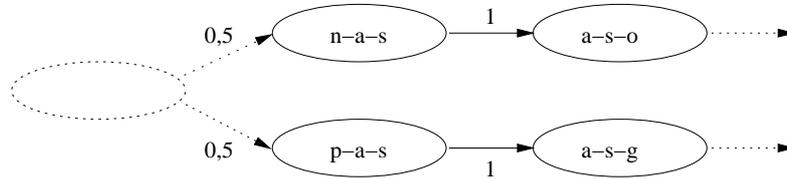


Figure 4.4: HPG from example 2 with history depth 3.

Utilizing a history depth of 3, the resulting HPG is shown in Figure 4.4 and the mining results are shown in Table 4.2. As can be seen, mining the

Rank	Rule	Probability %
1	news.jsp → addtobasket.jsp → specials.jsp → order.jsp	50
1	products.jsp → addtobasket.jsp → specials.jsp → gifts.jsp	50

Table 4.2: Rules mined from example 2 using history depth 3.

HPG with history depth 3 extracts rules exactly matching the true traversals. Furthermore, note that the number of states in the HPG with the additional history depth is lower compared to an HPG using a lower history depth. This is not a general property of adding history depth but since the structure of this particular website seems to branch out in a nice tree-like structure, most users will request similar sequences and a small number of states needs to be generated in the HPG. Although none of the above examples indicate it, we would expect the use of additional history depth to add to the size of the HPG measured by the number of states and productions (see later).

Pros Adding history depth to the HPG model offers the possibility of a higher degree of precision in the browsing information represented. For instance, with a history depth of 2, we explicitly construct the states of the HPG with information of, e.g., a traversal $A \rightarrow B$ for which the true traversals guarantee a correct support level. Modeling with a greater precision should then accordingly minimize the possibility that a distorted set of rules is mined from the HPG.

Adjusting the history depth for a specific mining task also present us with the ability to control the assumption of user memory. More specifically, if we assume that the choice of a user on a specific site only depends on the two previous choices made, we are able to adjust the history depth accordingly³. Depending on the structure of the particular website, the additional number of states and productions introduced by an additional history depth could be limited, provided that not all combinations of web pages have been requested in sequence.

Cons The advantages gained from constructing and mining a HPG under the Markov property gradually disappears as we increase the history depth. First, even with the increased history depth we still risk extracting distorted knowledge from the HPG. Second, the time used to construct and space

³For the example given, we would use a history depth of 2.

consumed to represent an HPG dramatically increases with the history depth as the number of states and productions to be constructed increase. If the traversals include a high number of different web pages requested in sequence, the number of states and productions in the constructed HPG could grow exponentially (see later). However, the actual increase in the number of states and productions depends on the nature of the browsing patterns on the website.

Third, the higher the history depth, the higher the boundary of the minimum length for a rule extracted from the HPG. Using a history depth of N will result in all rules being *at least* N long. Provided that one uses history depth with the aim of decreasing the distortion of rules, a consequence will be that possibly interesting rules having a length below N cannot be found. Alternatively you could hold several HPGs for different history depths to represent all traversals. Note that holding all HPGs lower than a history depth N cannot be more space consuming than holding the HPG with history depth N but at most double the space consumed. But as mentioned above, the space consumed by the HPG with history depth N might be considerable and a doubling might prove enormous.

Complexity Adding history depth to the HPG model will require a larger number of states and productions in the HPG as opposed to an HPG with history depth 1, since the states with history depth will contain combinations of existing web pages and not just single web pages. The complexity of adding a history depth N to an HPG which previously contained S states and P productions will create an HPG with a maximum of S^N states and P^N productions. The added complexity of creating the HPG with history depth N will therefore be $O(P^N)$ and the mining of the HPG with history depth will be $O(P^N + S^N)$.

However, the above complexity is assuming that all web pages are linked with all other web pages. The average complexity of adding 1 to the history depth will depend on the *fan-in*, denoted by f , i.e., the number of incoming productions of a state. This is because a state in the HPG will transform into one separate state for each incoming production in the HPG with the added history depth and thereby the number of states in an HPG with history depth N will be $O(f^{N-1}S)$. The average fan-in of a website can be calculated as $\frac{P}{S}$ and thereby the average case blow-up is considerable smaller than the worst case, as $f \ll S$. For instance, the fan-in for a website with a very tree-like structure where the user is guided toward a certain goal and minimal cross-linking amongst unrelated web pages is used, we would expect a fan-in close

to 1 and thus a minimal increase in the size of the HPG.

Furthermore, the number of productions in an HPG with history depth N will be exactly equal to the number of states in an HPG with history depth $N + 1$ since each production is combined with an adjacent state in order to form the correct states with added history depth. Thereby we find the average case complexity of creating an HPG with history depth N will be $O((\frac{P}{S})^N S) = O(f^N S)$ and the average case complexity of mining this HPG will be $O((\frac{P}{S})^N + (\frac{P}{S})^{N-1} S) = O(f^N S)$.

4.3 Summary

The addition of history depth allows for higher precision in the HPG model but also consumes more space due to the additional states and productions and limits the lower boundary of the length of the rules to be extracted. However, the trade off between the actual gain in precision and the added time and storage consumption from using an added history depth is unclear and will be investigated in this paper.

5 Measuring Quality

This section will include discussions on how we propose to measure the quality of rules mined from an HPG compared to the true traversals. This will include considerations on how we develop a measure which evaluate the Markov assumption and formal definition of the measures utilized in this paper.

5.1 Considerations

When attempting to evaluate the effects of the Markov assumption on the rules extracted from an HPG, we are interested in examining what the implications of such an assumption are. Specifically, we want to examine to what degree the knowledge discovered using such an assumption is similar to the true traversals. In other words, whether the knowledge found using the HPG is actually knowledge of a high quality.

The term quality is not an objective term, so in this paper, when talking of the quality of extracted knowledge, we will refer to two measures which is defined in this section. The first is a measure of *similarity*, i.e., to what degree the HPG is able to extract rules in a similar ordering compared to the rules in the true traversals. The second is a measure of *accuracy*, i.e., to what degree the HPG is able to extract rules indicating a support (probability) equivalent to the support of the rules in the true traversals. With this definition, knowledge of a low quality will corresponds to the distorted knowledge as presented in Section 3 and 4. Appendix A includes discussions on the semantics of the definitions.

Note that we do not argue that an assumption of a limited browsing history has no validity in an Internet setting. Many sites are not designed in order to guide the user in one specific direction but encourage users to choose new and unrelated links. In such and many other settings, using an assumption of a limited browsing history is quite valid, since the choices taken earlier by a user arguably has little or no impact on the following browsing behavior.

We aim to examine to what degree the use of the Markov assumption distorts the knowledge extracted. The results from such an investigation are relevant in the choice of underlying technology if the website to be examined cannot, for some reason, rely on the assumption of a limited browsing history and in the examining the general characteristics of the HPG model.

5.2 Definitions

In order to examine the characteristics of using the Markov assumption and the quality of the rules extracted from the HPG, we need to define the measures of similarity and accuracy. In the remainder of the paper, we will refer to the set of rules found in the true traversals as the *reference set* and the set which are to be compared to the reference set is referred to as the *experimental set*.

Similarity We need to define the a *super-rule* which can be used to compare a rule from a reference set to a rule from an experimental set.

Definition 1 Given a rule R , consisting of the sequence of web pages $R = \{r_1 \dots r_m\}$ and rule S , consisting of the web pages $S = \{s_1 \dots s_n\}$, S is a **super-rule** of R , denoted $R \subseteq S$, if $r_j = s_j$ for all $j = \{1 \dots m\}$ and $m \leq n$

Using the definition of a super-rule, we present a measure of similarity between a reference set and an experimental set.

Definition 2 Given a reference set $R = \{r_1, \dots, r_D\}$ and an experimental set $E = \{e_1, \dots, e_D\}$, the similarity between the two sets, referred to as $sim(E, R)$, is defined as $\frac{|SIM|}{D}$ where SIM is defined as:

$$SIM = \{e_i \in E | \exists r_j \in R \wedge r_j \subseteq e_i\}$$

In other words, the similarity is measured in terms of the percentage of rules in the experimental set that are super-rules of at least one rule in the reference set and we believe that this will indicate the usability of an HPG model in a concrete setting. If the rules extracted measure to a high percentage, the HPG model could be of use in an above mentioned setting, since the rules will be relatively similar to the correct rules. Note however, that similarity below 100 % is not a precise indication of the amount of false rules as we only compare a limited number of rules from each set allowing a reference rule matching the experimental rule to exist but outside the setsize considered.

Accuracy We need to define the difference between the probability of an experimental rule and a reference rule to be able to measure the accuracy of the rules in the experimental set.

Definition 3 Given a reference set $R = \{r_1, \dots, r_D\}$ and the probability of the elements $P_R = \{p_{r_1}, \dots, p_{r_D}\}$, and given an experimental set $E = \{e_1, \dots, e_D\}$ and the probability of the elements $P_E = \{p_{e_1}, \dots, p_{e_D}\}$, we

define the difference in probability of a given reference rule r_i in R and a potentially the matching rule e_j in E , denoted as $pDiff(r_i)$, as:

$$pDiff(r_i) = p_{r_i} - p_{e_j}$$

If a given r_i does not have a super-rule e_j in the experimental set, we define $pDiff(r_i)$ to be equal to p_{r_i} .

Using the definition of difference between rules in a set, we present a measure of accuracy between a reference set and an experimental set.

Definition 4 Given a reference set $R = \{r_1, \dots, r_D\}$ and the probability of the elements $P_R = \{p_{r_1}, \dots, p_{r_D}\}$, and given an experimental set $E = \{e_1, \dots, e_D\}$ and the probability of the elements $P_E = \{p_{e_1}, \dots, p_{e_D}\}$, we define the accuracy of E compared to R , referred to as $acc(E, R)$, as:

$$acc(E, R) = \sqrt{\frac{\sum_{i=1}^D (pDiff(r_i))^2}{D}}$$

Furthermore, the relative accuracy, referred to as $Racc(E, R)$, is defined as:

$$Racc(E, R) = \frac{acc(E, R)}{\sum_{i=1}^D p_{r_i}}$$

The definition of accuracy $acc(E, R)$ presents a measure of how accurately the HPG model can predict the probability rules, comparing the probability found in the reference set with the matching rule in the experimental set, if any. The measure will be an average deviation of the probability found in the HPG rules and the probability of a reference rule and the measure is absolute, i.e., the average amount of probability separating each experimental rule from a reference rule. This average does not take into account the actual amount of probability represented in the reference set and therefore the relative accuracy, $Racc(E, R)$, adjusts the average according to the sum of probability represented in the reference set. Since the sum of probability in the reference set is between 0 and 1, the absolute deviation is enlarged if only a small amount of probability is represented in the reference set and thereby indicate how significant the deviation becomes in respect to the reference set.

6 Experimental Evaluation

This section will include a description of the goals of the experiments, the experimental settings used, present the results of the experiments and summarize on the experiments.

6.1 Description of Experiments

In this paper we want to investigate to what degree the Markov assumption distorts the knowledge extracted from an HPG using various history depths. We do this by extracting a rule-set from an HPG representing a given website. The extracted rules are compared, using the measures described in Section 5, to the set of reference rules for the given website in order to evaluate the quality of the extracted knowledge.

We will only compare sets where the reference rules and the rules extracted from the HPG have at least the length of the modeled history depth and in comparing, e.g., the reference rules of length 4 or above with rules from the HPG, we will limit the comparison to HPG rules that are also of length 4 or above. By doing this, we measure to what degree the HPG is able to reproduce the rules found in the corresponding reference set. If we did not limit the comparison but compared the rules from the HPG as is, we would, e.g., for an HPG with history depth 1 compare the extracted rules of length 1 and above with reference rules of length 4 and above. We believe this to be the correct way of comparing a reference set and an experimental set, since only rules of equal length are considered. Furthermore, we will conduct the experiments using four different sizes of the sets, namely 10, 25, 50 and 100 elements, in order to verify whether the size of the sets considered in the comparison are significant and the experiments are performed using two different datasets. We have limited the comparison to the four sizes since we believe that considering more rules would not be supported in the datasets used in the experiments. We will refer to the first X elements in the ordered rule-set as *top X* , e.g., the first 25 elements are referred to as *top 25*.

The results of the experiments should indicate to which degree the rules extracted from an HPG is supported in the true traversals. If the measures indicate high degrees of similarity and accuracy, it will indicate that the presence of the Markov assumption in the given HPG model only distorts the extracted information to a small degree. High degrees of similarity and accuracy corresponds to our notion of knowledge of high quality.

6.2 Experimental Settings

We perform our experiments on web logs from two different websites. Our first log, called *CS*, is taken from the Department of Computer Science at Aalborg University, which is a website with a wide variety of information ranging from informational pages to personal home pages, including pages for individual university courses. We believe the typical usage patterns on the website to include both short traversals, focused on finding a specific piece of information, and long, more random traversals. The web log includes 22000 sessions containing 90000 clicks divided amongst 18000 unique pages. The most frequent requests are presented in Table 6.1.

Web page	No. of requests	Percentage
/	1788	2,0
/ strategy/Caesar/	1540	1,7
/mail/status.php3?language=da&message=Mappe...	811	0,9
/EDBT02/php3/menu.php3	671	0,8
/mail/status.php3?language=da	517	0,6

Table 6.1: Most frequent requests from *CS*.

Our second log, called *Bank*, is taken from an educational website placed at a large Danish bank the site include a large number of *training courses* that employees can complete. We believe the typical usage patterns on the web site to be quite long and quite focused, since coursesites span several web pages. The web log includes 21500 sessions containing 396000 clicks divided amongst 2400 unique pages. The most frequent requests are presented in Table 6.2. Note that the most requested web page, a dynamic page containing templates for multiple-choice tests used in all training courses, are almost five times as frequently requested as any other page.

Web page	No. of requests	Percentage
/stdelements/iis_multch/mult_ch.asp	76184	19,2
/Default.asp	13774	3,5
/discussions/default.asp	8884	2,2
/iktsystem/httperrors/http404.asp	6447	1,6
/courses/valuta/teori.asp	3163	0,8

Table 6.2: Most frequent requests from *Bank*.

The relatively small amounts of data in the web log does not interfere with the validity of the experiments since we are not interested in the perfor-

mance on large datasets but only in examining the quality of the extracted information. For more on the performance of PEHA, see Paper 2.

We performed the extraction of rules from an HPG using the MIMER framework, which is described in further detail in Paper 1. The framework is programmed in the Java programming language and utilizes MySQL 4.0.1 as the underlying DBMS running on an Intel 1 GHz machine. The true traversals are stored in a subsession schema[10] as well as a click fact schema[12]. Since the subsession fact schema explicitly stores sequences of clicks (subsessions) from the true traversals, we use the frequency of these subsessions in order to retrieve the reference rules from the set of true traversals. Furthermore, we use these subsessions in order to extract states and productions for building the HPGs with history depth above 1, as the sequences in the states are stored explicitly.

6.3 Experimental Results

The examination is conducted through four experiments. We examine whether a certain size of the rule-set present rules of a higher quality, the effects of using additional history depth in the HPG, the quality of information extracted on two different datasets and the accuracy of the probability in the rules extracted from the HPG. Finally, additional results are summarized.

Number of Rules to Consider We are interested in examining whether a specific size of the HPG rule-set in general indicate a higher similarity than others. In other words, whether a specific size of the rule-set is more similar to the equally sized set of reference set than another size of rule-set. If this is the general case, it would indicate an optimal rule-set size from the HPG to be considered. In the experiment, we compare the similarity of the four different rule-set sizes for an HPG with history depth 1 and an increasing length. If the results indicate that a specific size of the rule-set is generally more similar to the reference set, this size would be more optimal to consider. Figure 6.1 shows the results for the *CS* dataset, since we expect this dataset to contain the more random traversals and for $\alpha = 1$, since this provides the largest number of rules (no restrictions on the first state in a rule).

The HPG rule-set of size 100 is generally more similar to the reference set than the other sizes even as the history depth becomes lower than the length of the reference rules, whereas the top 10 rule-set is generally the least similar size, except for reference of length 1. This tendency becomes more evident for results obtained using $\alpha = 0$ and the *Bank* dataset (see Appendix B). The results indicate that a high number of rules from the HPG should be considered to ensure a high similarity to the true traversals. The accuracy

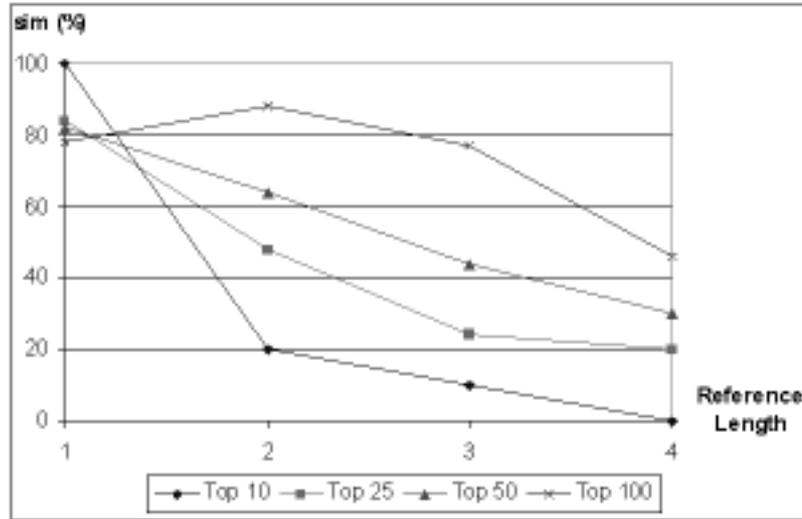


Figure 6.1: The similarity of differently sized sets.

also increase as the rule-set size increase, as can be seen from Appendix B.

Note that you would expect the results to indicate a similarity of 100% when comparing the rules of an HPG with history depth N and the reference rules of length N . This is not the case, as can be seen from Figure 6.1, since we compare rules of length *at least* N which allows for high probability rules longer than N to appear in the rule-sets and there by overtaking rules of lower length and lower probability. From Figure B.2 it can be seen that the characteristics of the *Bank* dataset as describe above, especially the frequent request, introduces a general low similarity.

Adding History Depth We are interested in examining to what degree using an HPG with additional history depth will provide rules that are more similar to the rules in the reference set. In the experiment, we extract rules using an increasing history depth in the HPG, and compare the extracted rules to the set of true rules of length 4 or above. We expect the experiment to show, that adding history depth will increase the similarity between the two sets. Figure 6.2 shows the results for both the *CS* and *Bank* datasets comparing the top 100 rules, since they indicate the most similar set sizes.

The figure shows that as a general tendency, added history depth increase the similarity of the rules extracted from the HPG compared to the rules in the reference set. Using history depth 1 will only extract around 20 to 50 % of

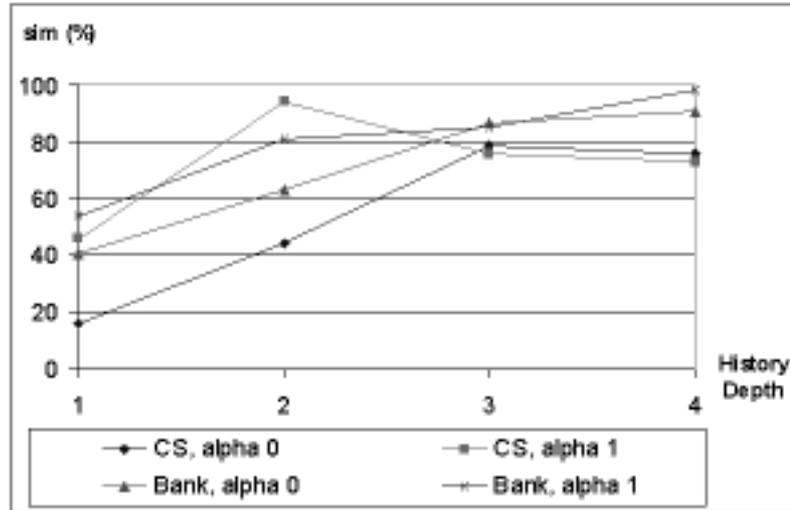


Figure 6.2: The implications of additional history depth.

the correct rules of length 4 whereas using history depth 3 will extract around 80 % of the correct rules. We believe that this indicates that the HPG model extracts a higher degree of potentially false rules, or more precisely, a rule-set that is ordered differently compared to the reference set, when attempting to extract rules that are much longer than the history depth used in the HPG. This indication can also be found for other sizes of the rule-set in Figure 6.1, where the similarity of the HPG using history depth 1 drops significantly as the length of the reference rules increase. Figure 6.2 seems to show a general tendency of a higher similarity when the rules are extracted using $\alpha = 1$ than when extracted using $\alpha = 0$.

Different Websites The HPG might extract more similar rules when applied on a particular type of website, representing certain browsing patterns. We examine the similarity of the two websites using the top 100 rules from both the *Bank* dataset, that includes the more directed usage patterns, and the *CS* dataset, that includes the more random, scattered usage patterns, by varying the length of the rules in the reference set for a constant history depth of 1. Figure 6.3 illustrates the similarity using mining settings of both $\alpha = 0$ and $\alpha = 1$.

The figure shows, that the rules extracted on the HPG of the *CS* website are generally more similar for short rule lengths compared to the *Bank* website. However, as the rule length increase the rules for the HPG of the *CS* website becomes less similar, similarity dropping around 20 % per increased

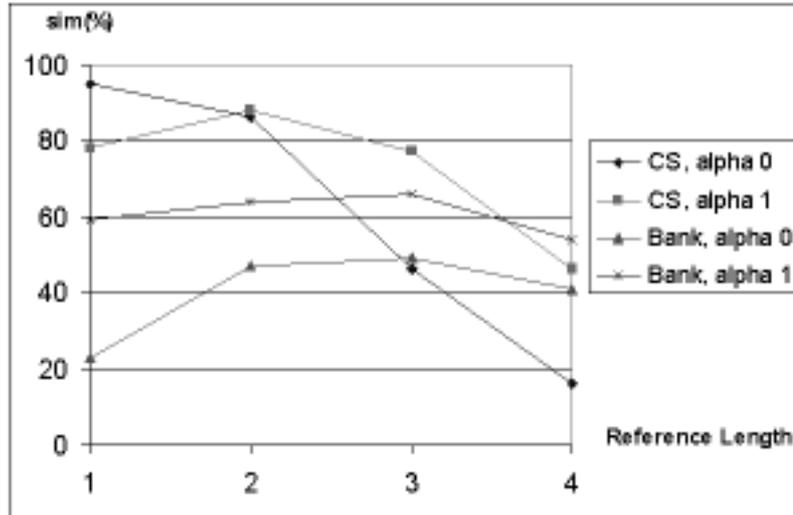


Figure 6.3: Similarity of different types of dataset.

in rule length. The *Bank* dataset does not present a clear tendency, as the similarity in the figure of a steady level of similarity is not confirmed in other results on the dataset (see Appendix B). The tendency for the rules extracted using $\alpha = 1$ to be more similar than the rules extracted using $\alpha = 0$ on the same dataset id confirmed in this figure as well.

Accuracy of Probability In this experiment, we are interested in evaluating the accuracy of the probability in the rules extracted from the HPG. The accuracy measure *Racc* presented in Definition 4 is utilized to compare how effectively the HPG is able to estimate the probability. Note that the lower the measure, the more accurately the HPG estimates the probability. Figure 6.4 shows the results using the top 100 rules for each of the datasets, utilizing various history depths to extract rules of length 4.

Figure 6.4 shows that mining the *Bank* dataset using $\alpha = 1$ proves to be the most inaccurate setting regardless of the history depth in the HPG. The three other settings generally demonstrate similar accuracy, as the low accuracy of the *CS* dataset, using $\alpha = 0$ for history depth 4 is caused by an insufficient dataset. Overall, note that the differences in accuracy are quite small so only limited conclusions can be drawn from the measure about the accuracy of the HPG in the different settings.

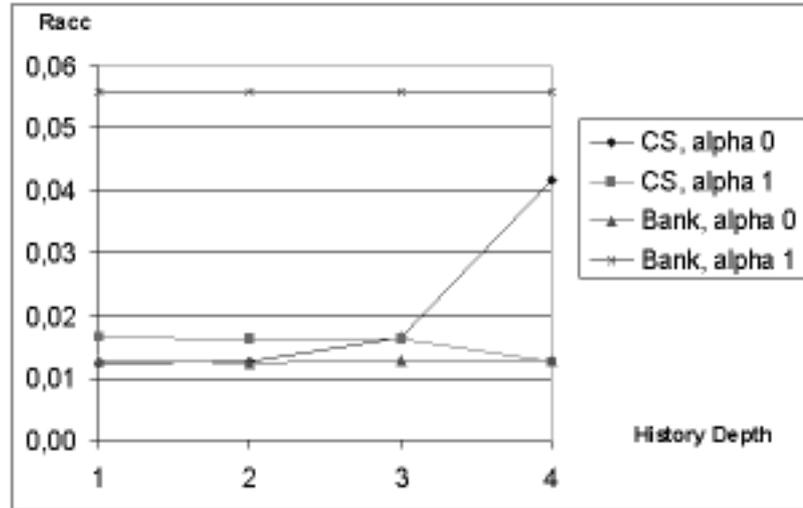


Figure 6.4: Accuracy of datasets.

6.4 Experimental Summary

The results presented above and in Appendix B indicate that in order to achieve high degrees of similarity and accuracy, a relatively large number of rules extracted from the HPG needs to be considered. The general tendency show that smaller amount of rules in the rule-sets considered generate less similar and accurate rule-sets. Adding history depth to the HPG increases the similarity of the extracted rule-set but the accuracy of these rule-sets tend to be uninfluenced by the addition of history depth. There is a tendency of rule-sets that are extracted using the mining setting of $\alpha = 1$ to be more similar but less accurate than rule-sets mined using $\alpha = 0$.

Overall, we believe the results to indicate that the HPG model is able to extract knowledge of a relatively high quality, but the quality decreases as the length of the rules considered grows compared to the history depth in the HPG model.

7 Conclusion and Future Work

The HPG model, an aggregated structure used for web usage mining, assumes independency of the states in the grammar of the model and therefore inherently risk extracting rules that represent distorted knowledge compared to the knowledge in the collection of sessions represented in the HPG, referred to as the true traversals. The precision of the HPG model can be increased by adding additional states, history depth, to the grammar to better represent sequences in the sessions. In this paper, we have examined the quality of the rules extracted from the HPG with respect to similarity, i.e., to what degree the HPG model is able to reproduce rules found in the true traversals, and accuracy, i.e., to what degree the HPG model is able to estimate the support level in the true traversals. Experiments does not present a precise measure of the quality, but indicate that the HPG model extracts increasingly distorted knowledge, especially as the length of the extracted rules grows compared to the applied history depth and as the number of rules considered decrease. Furthermore, the HPG model shows a tendency to extract knowledge that is more similar and accurate when applied to a website where browsing patterns generally are of a random, scattered nature. We conclude that the results of the experiments indicate that the HPG model is able to extract knowledge of a relative high quality in a web usage mining setting, however as the length of rules extracted grows compared to the history depth applied in the HPG, the quality degrades.

In future work, the similarity and accuracy experiments in this paper could be expanded to include datasets of other types and amount in the comparisons. This will allow for a more precise indication of the precision of how the HPG model extracts knowledge when applied in other fields than web usage mining. Research into an optimal size (or sizes) of the HPG in order to achieve knowledge of a higher quality could be conducted. Studies on the effect of using multiple HPGs modeled with different history depth could be useful in developing an approach in which performance and quality of extracted rules can be controlled. Combining statistical information and a heuristic for extracting rules based on the similarity and accuracy measures defined in this paper could be a valuable contribution to the HPG model.

References

- [1] A.G. Büchner, S.S. Anand, M.D. Mulvenna, and J.G. Hughes. Discovering Internet Marketing Intelligence through Web Log Mining. In *UNICOM99 Data Mining and Datawarehousing*, 1999.
- [2] J. Borges. *A Data Mining Model to Capture User Web Navigation Patterns*. PhD thesis, Department of Computer Science, University College London, 2000.
- [3] J. Borges and M. Levene. Data Mining of User Navigation Patterns. In *WEBKDD*, 1999.
- [4] J. Borges and M. Levene. Heuristics for Mining High Quality User Web Navigation Patterns. Research Note RN/99/68. Department of Computer Science, University College London, Gower Street, London, UK, 1999.
- [5] J. Borges and M. Levene. A Fine Grained Heuristic to Capture Web Navigation Patterns. *SIGKDD Explorations*, 2(1), 2000.
- [6] E. Charniak. *Statistical Language Learning*. MIT Press, 1996.
- [7] R. Cooley, J. Srivastava, and B. Mobasher. Web Mining: Information and Pattern Discovery on the World Wide Web. In *9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, 1997.
- [8] R. Cooley, P. Tan, and J. Srivastava. Websift: the Web Site Information Filter System. In *1999 KDD Workshop on Web Mining, San Diego, CA.*, 1999.
- [9] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.
- [10] J. Andersen, A. Giversen, A. H. Jensen, R. S. Larsen, T. B. Pedersen, and J. Skyt. Analyzing Clickstreams Using Subsessions. In *International Workshop on Data Warehousing and OLAP*, 2000.
- [11] S. E. Jespersen, T. B. Pedersen, and J. Thorhauge. A Hybrid Approach to Web Usage Mining - Technical Report R02-5002. Technical report, Dept. of CS Aalborg University, 2002.
- [12] R. Kimball and R. Merz. *The Data Webhouse Toolkit*. Wiley, 2000.

- [13] J. Pei, J. Han, B. Mortazavi-asl, and H. Zhu. Mining Access Patterns Efficiently from Web Logs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2000.
- [14] S. Ross. *A First Course in Probability*. Prentice Hall, 1998.
- [15] M. Spiliopoulou and L. C. Faulstich. WUM: a Web Utilization Miner. In *Workshop on the Web and Data Bases (WebDB98)*, 1998.

A Discussions on Quality

This appendix includes discussions on the semantic of the measures developed in Section 5. The discussions are not included in the main paper in order to maintain the focus of the paper. There are several considerations to be taken when designing a measurement of similarity between to sets, which will be discussed below. In the following we refer to the set of rules found in the true traversals as the *reference set* and the set which are to be compared to the reference set is referred to as the *experimental set*.

Range of Comparison In comparing a reference set and an experimental set, a significant consideration is how many elements of each set should be used in the comparison. Table A.1 is used to illustrate the considerations.

Rank	Reference Set	Exp. Set 1	Exp. Set 2
1	A	A	A
2	B	B	G
3	C	G	B
4	D	F	C
5	E	D	D
6	F	H	F
7		C	E
8		E	

Table A.1: Example of a differently sized reference and experimental sets.

The main consideration exemplified by Table A.1 is whether a measure can be created that incorporates sets of different sizes. One could argue that the complete sets should be compared as is, regardless of the differences in set sizes. In an Internet setting, the potential size of knowledge extracted using the Markov assumption is quite large (it is actually infinite, see Section 2) due to the circular linking on websites and therefore we argue that we cannot compare complete sets in a web setting. We therefore need to limit the size of the sets and ordering sets according to the support of each element and restricting the number of elements to compare is a natural choice. Whether or not to compare unequally sized sets rests upon the ability to restrict the sets in a well-defined manner. In other words, there needs to be a semantically sound way to decide how many elements from each set should be considered in a direct comparison. We believe that such a semantic is unclear in an Internet setting since, e.g., interesting knowledge cannot be expressed objectively, and therefore adopt the convention of comparing equally sized sets.

Ordering in Sets The potential different orderings of the experimental sets when compared to a reference set could also be included in a measure. The brief example in Table A.2 is used to illustrate some of the potential problems related to comparing ordering in sets.

Rank	Reference Set	Exp. Set 1	Exp. Set 2
1	A	A	A
2	B	B	G
3	C	G	B
4	D	E	C
5	E	D	D

Table A.2: Example of a reference set and two experimental sets.

Considering the sets in Table A.2, we believe that there is no clear way to measure which of the orderings in the experimental sets 1 and 2 are more similar to the reference set. For instance, a general heuristic choosing between the fact that set 1 has both A and B ranked similar to the reference set and the fact that set 2, with exception of element G in rank two, have a similar ordering to the reference set is unclear. Furthermore, note that the difference in probability of rules mined from the HPG will be relatively small and therefore the ordering in a sorted rule-set will rely on quite small difference in probability. We therefore argue that comparing sets according to an ordering resting upon such arguably small differences in rule probability could create an incorrect measure of the actual differences between sets, since the ordering will be performed on the basis of small differences.

Support Level The support level found in the reference set will not match the support indicated by an extracted rule from an HPG. The HPG extracts knowledge based on probabilistic considerations from the aggregated information whereas the reference rules are the actual true traversals and is therefore a precise count of user support. It is therefore important to note that the support levels indicated by the two sets are not similar, and this difference must be captured in a measure of accuracy of the HPG model. If the support level indicated by the HPG is close to the support in the true traversals, the HPG will be able to accurately determine the support of an extracted rule.

Comparing Individual Rules From Table A.3, note that the reference rule is *included* in the extracted rule. The HPG has found the trail $A \rightarrow B \rightarrow C$, but as the production leading to D had a high enough probability to be expanded, the extracted rule becomes $A \rightarrow B \rightarrow C \rightarrow D$. The probability

(and thus support) in the HPG of the rule before the last expansion will be at least as great as the probability after, but since the HPG only extracts rules which cannot be expanded any further[2], the two rules in the example will not match in a direct comparison. This aspect must be handled when matching reference rules and experimental rules..

Rule Origin	Rule
Reference Set	$A \rightarrow B \rightarrow C$
HPG	$A \rightarrow B \rightarrow C \rightarrow D$

Table A.3: A reference and an HPG rule.

B Additional Experimental Results

This appendix will present a number of additional experimental results that are not directly presented in the paper.

B.1 Similarity

The following graphs show additional results on interest in the evaluation of the similarity of the rules extracted from the HPG model compared to the reference rules. The results are presented without description as Section 6 include the evaluation of the experiments.

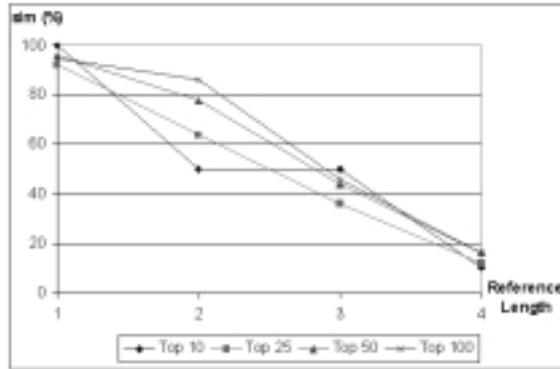


Figure B.1: Similarity measured on the *CS* dataset, using $\alpha = 0$ for a varying reference length.

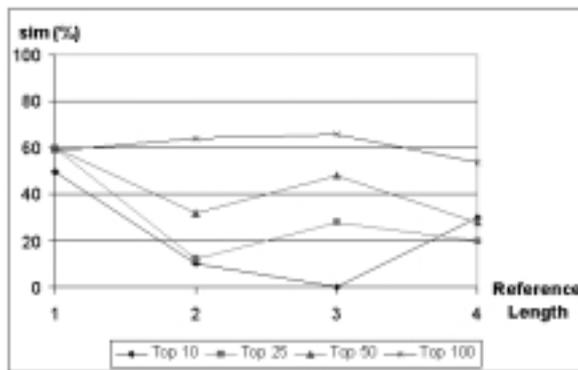


Figure B.2: Similarity measured on the *DB* dataset, using $\alpha = 1$ for a varying reference length.

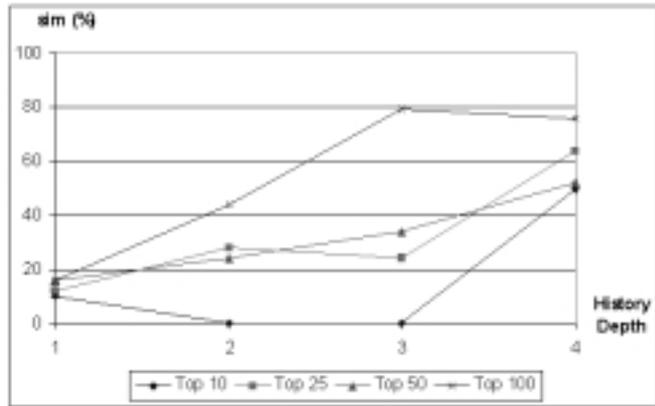


Figure B.3: Similarity measured on the *CS* dataset, using $\alpha = 0$ for a varying history depth.

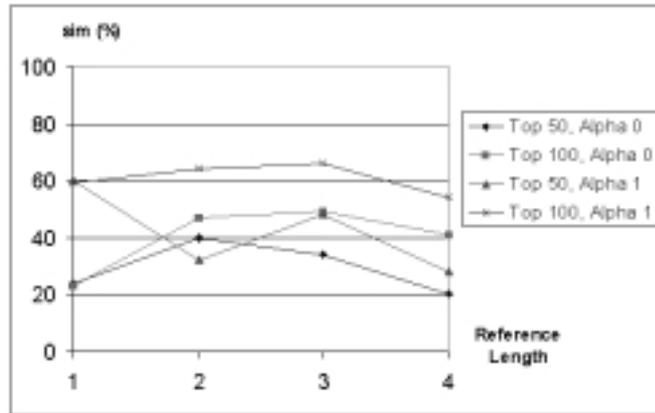


Figure B.4: Similarity measured on the *DB* dataset for a varying history depth.

B.2 Accuracy

The following graphs show additional results on interest in the evaluation of the accuracy of the rules extracted from the HPG model compared to the reference rules. The results are presented without description as Section 6 include the evaluation of the experiments.

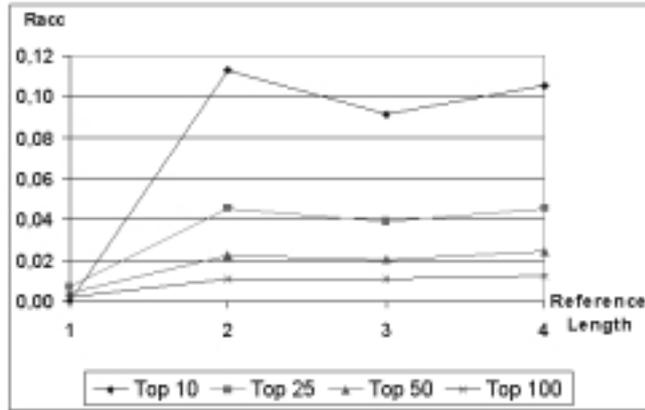


Figure B.5: Accuracy measured on the *CS* dataset, using $\alpha = 0$.

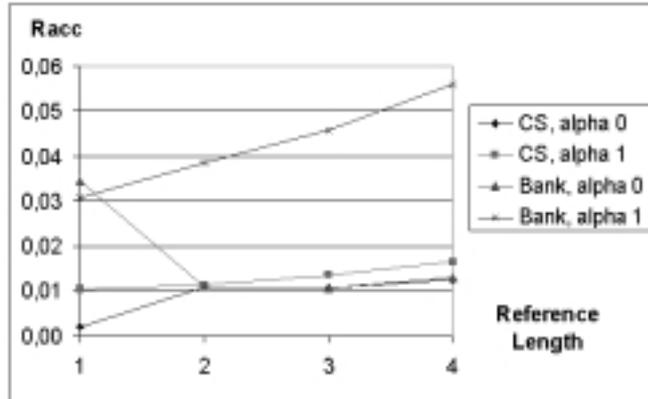


Figure B.6: Accuracy measured on the *CS* and *DB* datasets using a varying history depth.