Evaluation of Web Usage Mining Approaches for User's Next Request Prediction

Mathias Géry, Hatem Haddad
Information Technology Department
VTT Technical Research Centre of Finland,
Espoo, Finland
Mathias.Gery, Ext-Hatem.Haddad @vtt.fi

ABSTRACT

Analysis of Web server logs is one of the important challenge to provide Web intelligent services.

In this paper, we describe a framework for a recommender system that predicts the user's next requests based on their behaviour discovered from Web Logs data. We compare results from three usage mining approaches: association rules, sequential rules and generalised sequential rules. We use two selection rules criteria: highest confidence and last-subsequence. Experiments are performed on three collections of real usage data: one from an Intranet Web site and two from an Internet Web site.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining

General Terms

Algorithms, Experimentation

Keywords

Web Usage Mining, Association Rules, Frequent Sequences, Frequent Generalised Sequences, Evaluation

1. INTRODUCTION

In the WWW context, recommender systems are becoming widely used by users and information retrieval systems to perform results of both prefetching and recommendation [11]. In the literature, most researchers focus on Web usage mining that analyse Web logs with a process of discovering knowledge in databases. Indeed, Web sites are generating a big amount of Web logs data that contain useful information about the user behaviour. The term "Web Usage Mining" was introduced by Cooley et al. in 1997 [10] when a first attempt of taxonomy of Web Mining was done; in particular they define Web mining as the "discovery and analysis of useful information from the World Wide Web". It is also defined as "the application of data mining techniques to large Web data repositories" [9]. By citing the definition that Cooley et al. gave in [10], Web usage mining is the "automatic discovery of user access patterns from Web servers".

We define Web Usage Mining as the application of established data mining techniques to analyze Web site usage. With Web usage, we refer to the behaviour of one or more users on one or more Internet Web sites; the main character is the user, and in order to analyze and to study his behaviour, some data is needed.

There are a lot of works done in the field of Web Usage Mining (WUM), trying to improve Web search by analysis of user actions. The idea is to investigate the action that the user makes on the search results and/or during the navigation that follows. If, given 10 best matches for a query, a user selects the item with rank 5, this tells lots of valuable information about that item, and also something about the 4 preceding potential answers. Also, if the user has visited a given set of pages after his query, this tells lots about the interest of these pages related to the user's query. Without considering any query, it is also very interesting to investigate the user's action while navigating a Web site.

In this paper we describe a framework for a recommender system that predicts the user's next requests based on their behaviour discovered from Web Logs data. The paper is organized as follows: the section 2 introduces the problematic. The section 3 presents some basic considerations and definitions about the user and his navigation. Then, we describe the three experimented Web Usage Mining approaches (association rules, frequent sequence rules and frequent generalised sequence rules) in section 4 and the following prediction in section 5, while section 6 describes our results using three collections of real Web usage data.

2. MOTIVATION AND RELATED WORK

Today, millions of visitors interact daily with Web sites around the world and massive amounts of data about these interactions are generated.

We believe that this information could be very precious in order to understand the user's behaviour.

Web Usage Mining is achieved first by reporting visitors traffic information based on Web server log files. For example, if various users repeatedly access the same series of

WIDM'03, November 7–8, 2003, New Orleans, Louisiana, USA.

pages, a corresponding series of log entries will appear in the Web log file, and this series can be considered as a Web access pattern.

In the recent years, there has been an increasing number of research works done in Web Usage Mining (cf. [7, 8, 18, 3, 4, 13] and their developments). The main motivation of these studies is to get a better understanding of the reactions and motivations of users navigation. Some studies also apply the mining results to improve the design of Web sites [16], analyse system performances and network communications or even build adaptive Web sites [20]. We can distinguish three Web Mining approaches that exploit Web logs: association rules (AR) [11] [5], frequent sequences [17] and frequent generalised sequences [15] [12]. Algorithms for the three approaches were developed but few experiments have been done with real Web log data. In this paper, we compare results provided from the three approaches using the same Web log data.

3. CONSIDERATIONS AND DEFINITIONS

With the aim to study Web usage mining, we present in this section our definition of user and navigation.

3.1 User and navigation

A user is identified as a real person or an automated software, such as a Web Crawler (i.e. a spider), accessing files from different servers over WWW. The simplest way to identify users is to consider that one IP address corresponds to one distinct user. A click-stream is defined as a time-ordered list of page views. User's click-stream over the entire Web is called the user session. Whereas, the server session is defined to be the subset of clicks over a particular server by a user, which is also known as a visit. Catledge has studied user page view time over WWW and recommended 25.5 minutes for maximal session length [6]. An episode is defined as a set of sequentially or semantically related clicks. This relation depends on the goals of the study.

3.2 Web log files

The easiest way to find information about the users navigation is to explore the Web server logs. The server access log records all requests processed by the server. Server log L is a list of log entries each containing timestamp, host identifier, URL request (including URL stem and query), referer, agent, etc. Every log entry conforming to the Common Log Format (CLF) contains some of these fields: client IP address or hostname, access time, HTTP request method used, path of the accessed resource on the Web server (identifying the URL), protocol used (HTTP/1.0, HTTP/1.1), status code, number of bytes transmitted, referer, user-agent, etc. The referer field gives the URL from which the user has navigated to the requested page. The user agent is the software used to access pages. It can be a spider (ex.: GoogleBot, openbot, scooter, etc.) or a browser (Mozilla, Internet Explorer, Opera, etc.).

Here is an example from the MRIM Web server log:

```
64.68.82.52 - - [26/Jan/2003:15:05:37 +0100]
"GET /membres/mathias.gery/ HTTP/1.0" 200 12845 "-"
"Googlebot/2.1 (+http://www.googlebot.com/bot.html)"
```

In this example, somebody using the machine with the IP 64.68.82.52 has requested the page /membres/mathias.gery/ on January, 26th at 3.05 pm. In fact, the "user" is a spider from Google (Googlebot).

We never know when a user leaves a site: users do not warn Web servers that their visit has ended. A visit is a collection of user clicks on a single Web server during a user session. The only criterion proposed to identify a server session is to detect when the client has not accessed the site for a reasonably long time interval (e.g. 25.5 minutes [6]) by examining the next entries in the access log.

Another problem concerns the user identification. It is probably more appropriate to refer to client instead of user. If the server do not use authentication techniques, it is impossible to know exactly who really requests for resource on the Web. The user identification is important because the user is the character which creates a transaction; therefore, in the pre-processing phase, the user identification is applied before the transaction one.

4. WEB USAGE MINING

4.1 Association rules (AR)

The first studied approach we use is based on association rules. The problem of association rules is well defined in the literature [1, 2].

Association rules mining was first proposed to find all rules in a basket data (also called transaction data) to analyze how items purchased by customers in a shop are related (one data record per customer transaction). The association rule generation is achieved from a set F of frequent itemsets in an extraction context $\mathcal{D}$, for the minimal support *minsup*. An association rule r is a relation between itemsets of the form $r : X \Rightarrow (Y - X)$, in which X and Y are frequent itemsets, and $X \subset Y$. The itemsets X and $(Y - X)$ are called, respectively, *antecedent* and *consequence* of the rule r. The valid association rules are those of which the measure of support and confidence, is greater than or equal to the minimal thresholds of support and confidence, called *minsup and minconf*. Support and confidence are calculated as follows:

$$Support(X) = \frac{|\{t \in \mathcal{D} \mid X \subseteq t\}|}{|\mathcal{D}|} \quad (1)$$

$$Confidence(r) = \frac{support(Y - X)}{support(X)} \quad (2)$$

The problem of finding Web pages visited together is similar to finding associations among itemsets in transaction databases. Once transactions have been identified, each of them could represent a basket, and each resource an item.

We define a set of sessions $S = \{S_1, ..., S_n\}$ where n is the number of sessions in S and a set of URLs $R = \{url_1, ..., url_u\}$ where u is the number of URLs in S.
Each session S_i is defined by an IP and URLS:

$$S_i = (IP_i, URLs_i)$$

IP identifies the user of the session and URLs is the set of pages requested by the user. We define m as the number of pages requested by a user IP_i in a session S_i. It depends on the gap clicks Δ used to define a session (i.e. Δ is the maximum session length). In our experiments, we have used a value of $\Delta = 25.5$ minutes (as in [19] and recently in [11]).

$$URLs_i = \{url_{i,1}, ..., url_{i,k}, ..., url_{i,m}\} \text{ where } url_{i,k} \in R$$

Many algorithms can be used to mine association rules from the data available; one of the most used and famous is the *Apriori* algorithm proposed and detailed in [1].

For our recommender system we need association rules with one single item in the consequence of the rule.

A disadvantage of association mining is that it does not inherently use the notion of temporal distance which we believe is crucial for deciding which rules to apply for a given Web transaction. Association mining capture patterns relating to itemsets irrespective of the order in which they occur in a given transaction.

4.2 Frequent Sequences (FS)

Frequent sequence Mining can be thought of as association mining over temporal datasets and a sequence as an ordered (over time) list of non-empty itemsets. The attempt of this technique is to discover time ordered sequences of URLs that have been followed by past users, in order to predict future ones. We define t_i as url_i associated time,

$$URLs_i = \{(url_{i,1}, t_{i,1}), ..., (url_{i,k}, t_{i,k}), ..., (url_{i,m}, t_{i,m})\}$$

where $t_{i,k} < t_{i,k+1}$ and $t_{i,m} - t_{i,1} <= \Delta$.

Frequent sequence is based on the concept of N-Gram. As in [11], we define an N-Gram S_N of a session S_i as any subset of N consecutive URLs of S_i. The support of S_N is defined as:

$$support(S_N) = \frac{|S_i \in S / S_N \subset S_i|}{|S|}$$

Let S_{N+1} be the concatenation of S_N and url_k. Then, the confidence of a sequential rule $r : S_N \Rightarrow url_k$ is defined as:

$$Conf(r) = \frac{support(S_{N+1})}{support(S_N)}$$

4.3 Frequent Generalised Sequences (FGS)

A generalized sequence is a sequence allowing wildcards [12], in order to reflect the user's navigation in a flexible way. According to Gaul, we define a generalized sequence as a sequence in the symbols $R \cup \{*\}$ with an additional symbol $* \notin R$ called wildcard, such that no two wildcards are adjacent [12]:

$$R^{gen} := \{x \in (R \cup \{*\}) * | !\exists i \in \mathcal{N} : x_i = x_{i+1} = *\} \quad (3)$$

Similarly to Gaul, we consider that a sequence or a generalized sequence y *matches* a generalized sequence x (denoted $x \vdash y$) when it exists a mapping between x and y [12]. That means that each "*concrete*" element in x exists in y (in the same order), and each '*' element in x corresponds to one or several elements in y. For example, if $x = (A * EF * J)$ and $y = (ABCDEFGHIJ)$, then $x \vdash y$.

Let S be a finite list of ordinary sequences $x \in R^*$. According to Gaul, the support of a generalized sequence $x \in R^{gen}$ with respect to S is the relative frequency of sequence containing a subsequence which matches x:

$$Support(x) = \frac{|\{s \in S | \exists y \leq s : x \vdash y\}|}{|S|} \quad (4)$$

In order to extract frequent generalized subsequences, we have used the general algorithm proposed by Gaul. This is an Apriori algorithm adapted to the generalized sequences, based on the fact that the support of any subsequence of a closed generalized sequence is greater than or equal to the support of the sequence itself. Thus, we can build every generalized sequence of length n, for $n \geq 3$ as a junction of two smaller overlapping generalized sequences. The modified join step of the Apriori algorithm is described in [12].

5. RECOMMENDATION

5.1 Rule function selection

We define two criteria to select discovered rules matching the pages requested by a user: *Highest Confidence* (HC) and *Last Sequence* (LS).

The first rule selection criterion is highest confidence. Since different discovered rules antecedents can match pages requested by a user, rules with highest confidence are chosen to predict the next page. The second rule selection (LS) considers the distance between pages requested by a user and the consequence of a rule; this distance is the number of clicks from one page to another. LS strategy selects rules where the requested pages are the closest to the consequent. If different rules are equals considering LS, those with highest confidence are always chosen.

We notice that the LS strategy is not applicable in the case of AR, because this technique produces unordered results. Thus, we obtain the following combinations of rules types and selection strategies: AR, FS-HC, FS-LS, FGS-HC, FGS-LS.

5.2 Next page prediction

A predictive model is built based on the extracted usage patterns. Our hypothesis is that pages accessed recently have a great influence on pages that will be accessed in the near future.

The prediction is based on discovered rules matching the current user session to find candidate for prediction. If we consider a user session S_i as defined in 4.1, if S_i matches discovered rule antecedent then the discovered rule consequence is considered as S_i next page prediction.

6. EXPERIMENTATIONS

6.1 Web logs characteristics

We have used three collections of Web log datasets: *MRIM*, *VTT* and *IntraVTT*, one from an Intranet Web site and two from an Internet Web site.

- MRIM: the Web site describing the activities of the MRIM research group from french CLIPS Laboratory, accessible on `http://www-mrim.imag.fr`.
- VTT: the Web site of the VTT research institute, accessible on `http://www.vtt.fi`.
- IntraVTT: the intranet Web site of the VTT, accessible only from VTT machines.

The main characteristics of three Web logs from each of these Web sites are summarized in the next table:

	MRIM	VTT	IntraVTT
From	November 2002	January 2003	April 2003
To	June 2003	March 2003	July 2003
# hits	1 227 308	5 816 729	6 073 185
Size (Mb)	255	1 512	1 471

Figure 1: Characteristics of MRIM, VTT and IntraVTT logs.

6.2 Preparing the Data

6.2.1 Data pre-processing

Data pre-processing is a set of operations that process the available sources of information and lead to the creation of an ad-hoc formatted dataset to be used for knowledge discovery through the application of data mining techniques. Data cleaning aims to remove unwanted information in Web server logs, as for example all the requests made by spiders, or all the requests about secondary resources (images, programs, etc.). Data Transformation aims to include user identification and pageview/session identification, in order to transform rough data into structured information.

We consider Web log data as a sequence of distinct Web pages accesses, were subsequences (user sessions) can be observed by unusually long gaps between consecutive requests. For example, assume that the Web log consists of the following user visit sequence: (A (by user 1), B (by user 2), C (by user 2), D (by user 3), E (by user 1)).

This sequence can be divided into user sessions according to IP address: Session 1 (by user 1): (A E); Session 2 (by user 2): (B C); Session 3 (by user 3): (D), where each user session corresponds to a user IP address. We identify a user visit as a set of page views that are sufficiently close over time by using a maximum time gap. We identify a page view as an HTML or a dynamically generated file that is sufficiently far over time from the previously identified page, using a minimum time gap specified manually. A Web access sequence is defined as a time-ordered set of visits.

6.2.2 Training set and evaluation

Each Web log dataset is split into training dataset and evaluation dataset. The training dataset is mined in order to extract rules, while the evaluation dataset is considered in order to evaluate the predictions made based on these rules.

We consider each user's session in test dataset as a *query*, used to predict the last accessed Web page considering all the first accessed Web pages. For example, if the test dataset contains the user's session (A, B, C, D), we query our system using the beginning of the path (i.e. (A, B, C)), considering that the relevant page to predict is (D).

6.3 Characteristics of preprocessed logs

The evaluation set is composed by the first 400 000 hits from the initial dataset, while the training set is composed by the remaining hits. The final characteristics of preprocessed collections are given in the next two tables, as well for the training set as for the evaluation set.

	MRIM	VTT	IntraVTT
# hits	827 308	5 416 729	5 673 185
# hits after filtering	38 894	542 854	646 835
# pages	2 098	20 419	5 130
# users	5 913	81 072	3 901
# sessions	10 786	154 302	132 439
# atomic sessions	7 298	83 350	46 764
Average size (#nodes)	3.68	4.46	5.53
Average time (seconds)	147	125	183

Figure 2: Characteristics of MRIM, VTT and IntraVTT training sets.

	MRIM	VTT	IntraVTT
# hits	400 000	400 000	400 000
# hits after filtering	22 014	38 137	51 296
# pages	1 325	8 419	1 903
# users	5 121	8 257	2 343
# sessions	7 733	10 893	10 247
# atomic sessions	5 337	5 916	4 265
Average size (#nodes)	3.47	4.45	5.26
Average time (seconds)	107	121	192

Figure 3: Characteristics of MRIM, VTT and IntraVTT evaluation sets.

We notice that the collections size dramatically decrease during the data cleaning step, mostly due to the elimination of secondary accesses and spider accesses. It also remains a majority of atomic sessions (i.e. sessions composed by only one hit, that are not usable as queries). After another filtering step, removing these atomic sessions as well as the longest ones (those containing more than 7 hits) and those containing some hits to resources that are unknown in the training set, the final queries set obtained is presented in the next table:

	MRIM	VTT	IntraVTT
# queries	944	3 040	2 431
Average size (#nodes)	2.36	6.96	5.39

Figure 4: Characteristics of MRIM, VTT and IntraVTT queries sets.

6.4 Prediction evaluation: accuracy and coverage

For measuring the performance of prediction, we use common measures: *accuracy* and *coverage*. Accuracy measures the system ability to provide correct predictions:

$$Accuracy = \frac{\text{correct predictions}}{\text{total predictions made}}$$

In fact, we use two variants of accuracy: $accuracy_{all}$ and $accuracy_1$. The $accuracy_1$ measure considers that the system has found a correct prediction only if it has ranked first *the* correct prediction. The $accuracy_{all}$ measure considers that the system has found a correct prediction if *the* correct prediction is in the results set, whatever its rank. Thus, a system that predicts all the pages in the collection has $accuracy_{all} = 1$.

Coverage measures the system ability to provide predictions for the testing database :

$$Coverage = \frac{\text{queries with predictions}}{\text{total number of queries}}$$

6.5 Prediction results

In this section, we present the accuracy and coverage results for each kind of extracted rule and for each recommendation strategy. We present also the average number of predictions per query.

6.5.1 Average number of predictions per query

As shown in figures 5, 6 and 7, for the same support threshold the average number of predictions per query is by far the most important with AR technique, and is quite important with FGS technique associated with HC selection strategy. That is caused by the great number of possible combinations obtained using these techniques. In fact, AR technique does not consider order, and FGS integrates a flexible '*'.

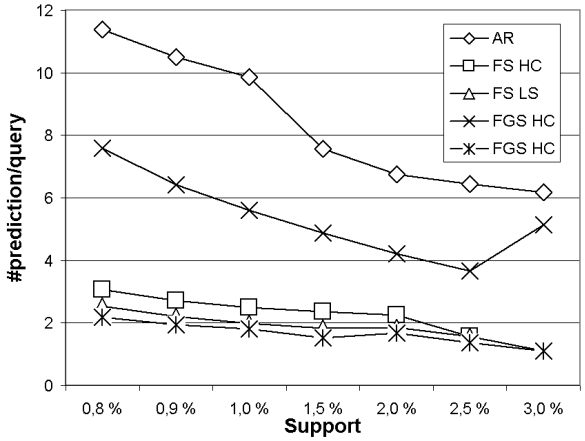

Figure 5: Average number of predictions par query for MRIM log dataset.

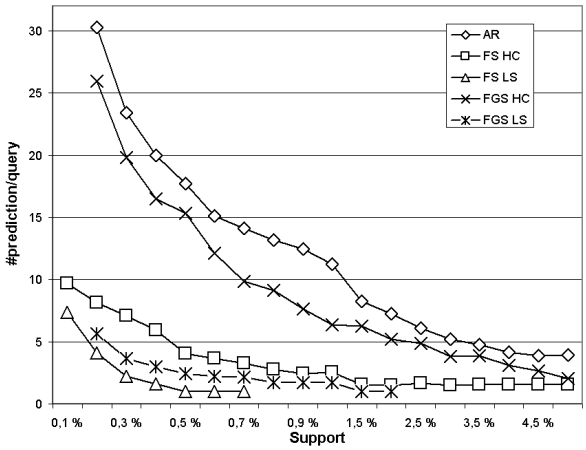

Figure 6: Average number of predictions par query for VTT log dataset.

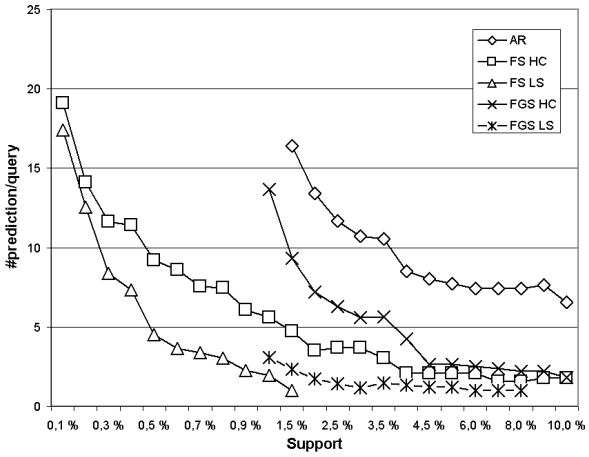

Figure 7: Average number of predictions par query for IntraVTT log dataset.

We notice that it is difficult (and sometimes impossible) to compare the different techniques considering a given support threshold. For example, the figure 7 shows that AR technique produces a lot of rules with threshold = 1.5% but needs a lot of processing time in order to be computed, while FS technique produces a very small number of rules with the same threshold.

6.5.2 Coverage

The figures 8, 9 and 10 show that the coverage is related to the average number of predictions per query. For example, AR technique gives the best coverage results for every collection. But FGS technique (especially FGS associated with HC strategy) gives very good coverage without giving too much predictions per query as AR do.

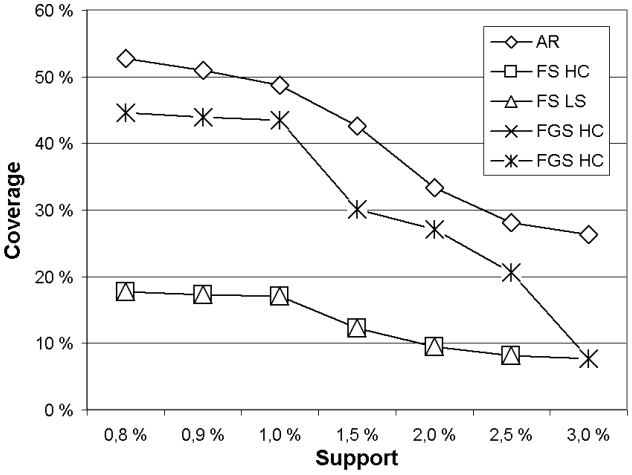

Figure 8: Coverage results varying support value for MRIM log dataset.

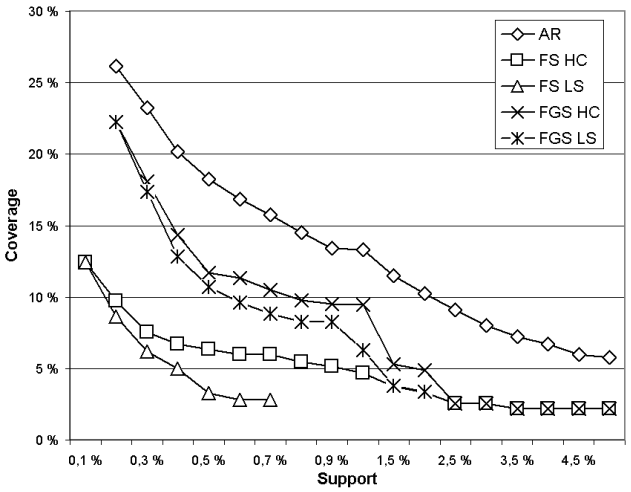

Figure 9: Coverage results varying support value for VTT log dataset.

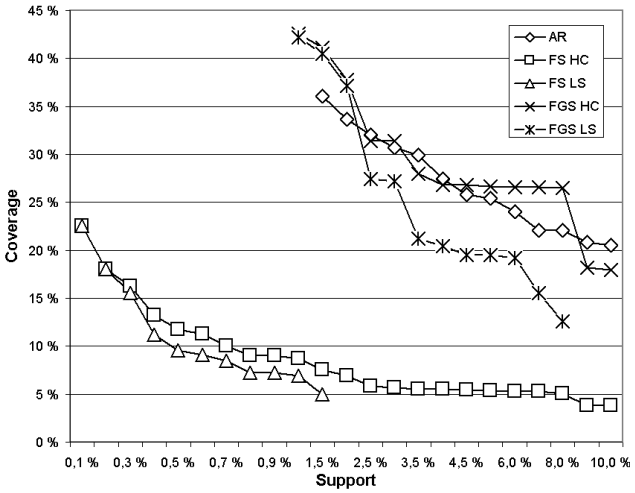

Figure 10: Coverage results varying support value for IntraVTT log dataset.

6.5.3 $Accuracy_{all}$

The accuracy values show that results are better when the support threshold is low. Varying support values has more impact on FS results than AR and FGS. In the other hand, this values depend on the Web log datasets as shown in figures 8 and 11 for MRIM, figures 9 and 12 for VTT and figures 10 and 13 for IntraVTT. Contrary to the results presented in [17], results for AR are more satisfactory than the other approaches. The cause for this difference is that the average number of predictions per query is greater, as shown in figure 5, especially for high support value where average predictions per query for FS and FSG is 1 as shown in figures 6 and 7. It's impossible to compare results in [17] and results in this paper since different Web log datasets were used with different characteristics.

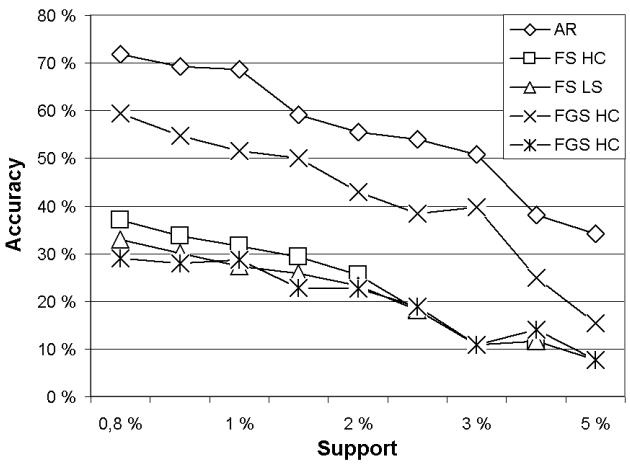

Figure 11: Accuracy results varying support value for MRIM log dataset.

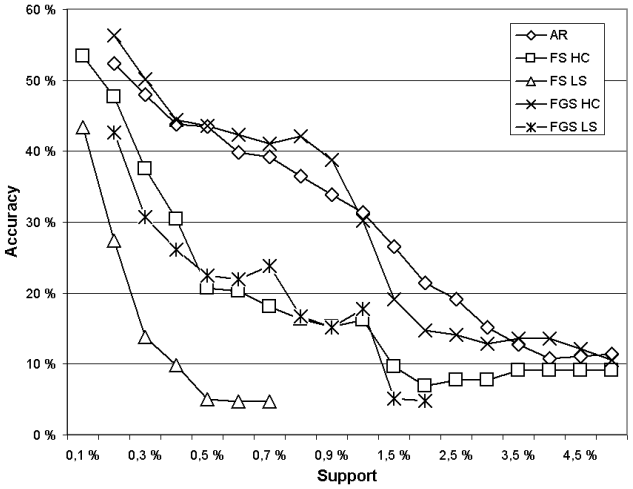

Figure 12: Accuracy results varying support value for MRIM log dataset.

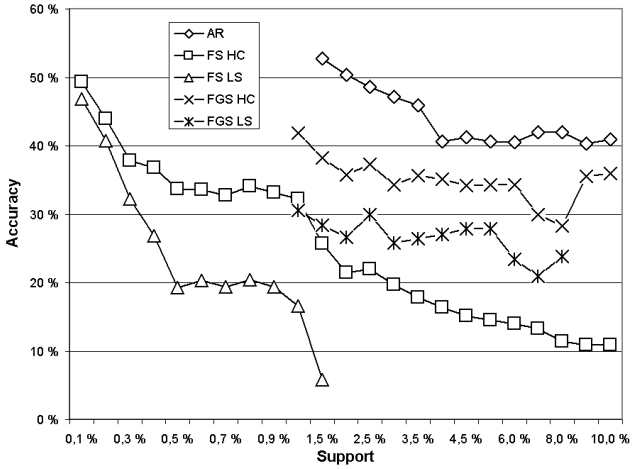

Figure 13: Accuracy results varying support value for MRIM log dataset.

6.5.4 $Accuracy_1$

Figures 14, 15 and 16 present accuracy results considering only one prediction for each query. Indeed, our assumption in a user navigation context is that system proposes one URL at time to the user, and when dealing with Web prefetching, system prediction propose few URLs. Results for FS are better but far from being acceptable regarding coverage: only 17.69% of coverage for the best accuracy (21.56%) for MRIM, 12.43% of coverage for the best accuracy (22.49%) for VTT and 22.50% of coverage for the best accuracy (27.79%) for IntraVTT.

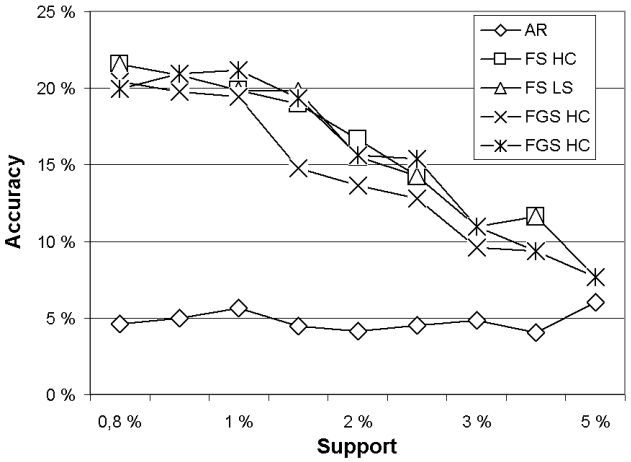

Figure 14: Accuracy results for one prediction varying support value for MRIM log dataset.

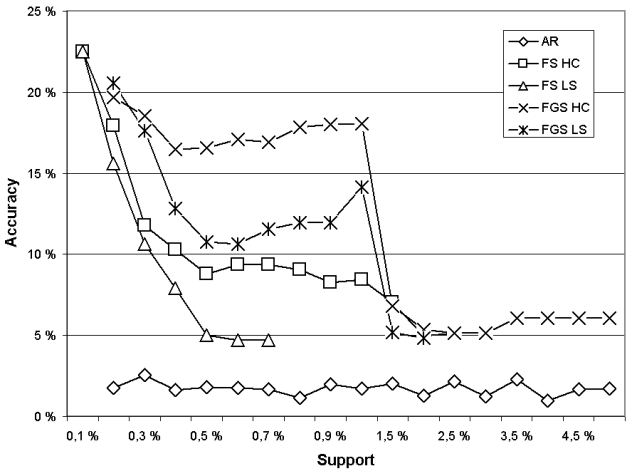

Figure 15: Accuracy results for one prediction varying support value for VTT log dataset.

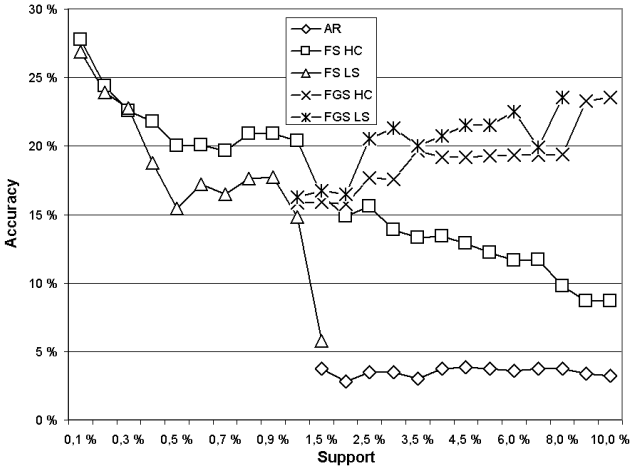

Figure 16: Accuracy results for one prediction varying support value for IntraVTT log dataset.

6.5.5 Prediction strategies: LS, HC

Results show that *Highest Confidence* criterion provides better prediction accuracy and coverage than *Last Sequence* criterion for different Web log datasets and different support values. Our hypothesis is that *Last Sequence* criterion is too restrictive to provide predictions since few FS and FSG match discovered rules. We think that a partial matching function should be more suitable for this approach.

7. CONCLUSION

In this paper, we have presented and experimented a framework for a recommender system that predicts the user's next requests based on their behaviour discovered from Web Logs data. We have combined three Web Usage Mining approaches (association rules, frequent sequence rules and frequent generalised sequence rules) with two prediction strategies, and we have evaluated these combinations using three collections of real Web usage data.

Results show that FS give better accuracy than AR and FGS when we need to find the correct prediction within the very first predictions.

We think that system should give better results if the gap clicks Δ is well studied for each Web log dataset. Indeed, the Δ value (25.5 minutes) used in our experiments was proposed by [6]. He and al. in [14] proposed a promising approach to detect session boundaries from Web user logs based on close proximity in time that we will use in the near future.

Results show that the coverage measure is quite low, for each one of the Web Usage Mining techniques used (cf. 6.5.2). We think that these poor results are caused by the huge number of possible navigations on a Web site. Even the AR and FGS techniques, that allow predicting some new navigation possibilities compared to FS technique, are not able to give an acceptable coverage result. We think that this drawback shows the limit of Web Usage Mining techniques, based on the usage data of previous users. In fact, that is very difficult to predict new navigations that have not been done before.

Thus, a very interesting and promising research development of Web Usage Mining is to integrate some information extracted from content and structure of the Web site to the Mining process. For example, we are developing techniques in order to consider the linkage of a Web site during mining, combined with similarity between linked pages. That should allow predicting navigations that have never been done before, and it should also allow predicting navigations that have not been planned by the author of the Web site.

8. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26–28 1993.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In P. S. Yu and A. S. P. Chen, editors, *Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan, 1995.

[3] M. Baumgarten, A. G. Bchner, S. S. Anand, M. D. Mulvenna, and J. G. Hughes. User-driven navigation pattern discovery from internet data. In *International ACM Workshop on Web Usage Analysis and User Profiling (WebKDD'99)*, pages 74–91, 1999.

[4] B. Berendt. Web usage mining, site semantics, and the support of navigation. In *Workshop Web Mining for E-Commerce - Challenges and Opportunities*, Boston, MA, August 2000.

[5] J. Bollen, H. V. de Sompel, and L. M. Rocha. Mining associative relations from website logs and their application to context-dependent retrieval using spreading activation. In *Workshop on Organizing Web Space (WOWS)*, Berkeley, California, August 1999.

[6] L. D. Catledge and J. E. Pitkow. Characterizing browsing strategies in the World-Wide Web. *Computer Networks and ISDN Systems*, 27(6):1065–1073, 1995.

[7] M.-S. Chen, J. S. Park, and P. S. Yu. Data mining for path traversal patterns in a web environment. In *16th International Conference on Distributed Computing Systems*, pages 385–392, May 1996.

[8] D. Cheung, B. Kao, and J. Lee. Discovering user access patterns on the worldwide web. In *1st Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'97)*, February 1997.

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

[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)*, November 1997.

[11] E. Frias-Martinez and V. Karamcheti. A prediction model for user access sequences. In *WEBKDD Workshop: Web Mining for Usage Patterns and User Profiles*, July 2002.

[12] W. Gaul and L. Schmidt-Thieme. Mining web navigation path fragments. In *Workshop on Web Mining for E-Commerce – Challenges and Opportunities*, pages 319–322, Boston, MA, August 2000.

[13] M. Hansen and E. Shriver. Using navigation data to improve IR functions in the context of web search. In *CIKM*, pages 135–142, 2001.

[14] D. He and A. Goker. Detecting session boundaries from web user logs. In *BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research*, 2000.

[15] H. Mannila and H. Toivonen. Discovering generalized episodes using minimal occurrences. In *Knowledge Discovery and Data Mining*, pages 146–151, 1996.

[16] F. Masseglia, P. Poncelet, and M. Teisseire. Using data mining techniques on web access logs to dynamically improve hypertext structure. *ACM SigWeb Letters*, 8(3):13–19, October 1999.

[17] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Using sequential and non-sequential patterns for predictive web usage mining tasks. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'2002), Maebashi City, Japan*, December 2002.

[18] M. Spiliopoulou and L. C. Faulstich. Wum: A tool for web utilization analysis. In *EDBT Workshop WebDB'98*, Valencia, Spain, March 1998.

[19] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.

[20] T. W. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From User Access Patterns to Dynamic Hypertext Linking. In *5th World Wide Web Conference (WWW'96)*, Paris, France, May 1996.