A Simulation-based Approach to Evaluating the Effectiveness of Navigation Compression Models

Tong Zheng and **Randy Goebel**
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
{tongz, goebel}@cs.ualberta.ca

Abstract

Within the growing literature on web mining, there is a relatively coherent thread of ideas focused on improvements to web navigation. In this paper we focus on the idea of web usage mining, and present a general framework for deploying the mining results and evaluating the performance improvement. The generalized objects created by the application of learning methods are called Navigation Compression Models (NCMs), and we show a method for creating them and using them to make dynamic recommendations.

Of note is the observation that no application of any learning method to web data makes sense without first formulating a goal framework against which that method can be evaluated. This simple idea is typically the missing ingredient of many WWW mining techniques. In this paper we present a simulation-based approach to evaluating the effectiveness of Navigation Compression Models non-intrusively by measuring the potential navigation improvement.

We evaluate the improvement of user navigation using a quantitative measure called *navigation improvement* (NI), which indicates whether we are actually "improving" the user's navigation by reducing the number of hyperlinks traversed to find "relevant" pages.

Introduction

With the rapid growth of the World Wide Web (WWW or web), web users constantly suffer from unguided and inefficient web usage. Web mining was introduced for its ability to discover "interesting" knowledge (or patterns) inside the web data that can be used to improve the performance of various web applications.

Within the growing literature on web mining, there is a relatively coherent thread of ideas focused on improvements to web navigation (Perkowitz & Etzioni 1998; Zaïane, Xin, & Han 1998; Mobasher, Cooley, & Srivastava 1999; Zheng, Niu, & Goebel 2002; Zhu, Greiner, & Häubl 2003). Perkowitz and Etzioni (Perkowitz & Etzioni 1998) proposed an approach to creating adaptive web sites by generating synthetic index pages based on clusters of pages that tend to co-occur in user sessions. Mobasher et al. (Mobasher, Cooley, & Srivastava 1999) proposed a usage-based web personalization system that makes personal recommendations to individual users based on the results of a number of data mining techniques such as association rules and URL clusters. Zhu et al. (Zhu, Greiner, & Häubl 2003) proposed another interesting approach which learns *information content* words from users' browsing features and user-assessed "relevant" pages, and then uses those words to generate dynamic recommendations.

One common approach of applying web mining to navigation improvement is to make personal recommendations to each individual user based on the knowledge learned from previous navigation patterns of aggregate users. This process is also referred to as collaborative filtering (Resnick *et al.* 1994). The primary idea of collaborative filtering requires the users to manually rank each object in the information space (in our case, to label each web page as "relevant" or "irrelevant"). In the case of web mining, these labels are, more practically, assigned automatically by heuristics or models learned from pre-labeled training samples.

A typical collaborative filtering recommendation system requires at least two essential components: (1) a knowledge base, which contains the knowledge learned from aggregate user behaviors, and (2) a recommender, which makes recommendations to each individual user based on the user's current status and related knowledge retrieved from the knowledge base. In the case of web usage mining, the knowledge base is a set of learning results obtained by applying various data mining techniques to the web usage data. In this paper, these learning results are generalized as Navigation Compression Models (NCMs).

While NCMs represent an abstract form of user navigation patterns, there is a significant problem within these patterns: a traveled path is not necessarily a desired path. Therefore, we propose a recommendation mechanism which ignores those auxiliary pages and makes recommendations only on "relevant" pages.

Of note is the observation that no application of any learning method to the web data makes sense without first formulating a goal framework against which that method can be evaluated. In our case, we require some evaluation method that can measure the improvement of navigation after the learning methods have been applied.

In the case of using dynamic recommendations to help user navigation, precision and recall can be used to measure the accuracy of the recommendations. But they are not capable of evaluating the actual improvement in user navigation, because an accurate recommendation is not necessarily a "useful" recommendation. For example, if the recommended page is exactly the page the user originally planned to visit next, the recommendation is actually useless. In this paper, we evaluate the improvement of user navigation using a quantitative measure called *navigation improvement* (NI), which indicates whether we are actually "improving" the user's navigation by reducing the number of hyperlinks traversed to find "relevant" pages.

The rest of this paper is organized as follows. The second section describes the idea, the creation and use of Navigation Compression Models. The third section introduces the simulation-based evaluation method used to measure the navigation improvement and a navigation compression mechanism based on dynamic recommendations. The fourth section presents our experiments with discussions on the results. The last section concludes this paper.

Navigation Compression Models

A typical web mining architecture (Zheng, Niu, & Goebel 2002; Mobasher, Cooley, & Srivastava 1999; Zaïane, Xin, & Han 1998) generally requires a module to support data capture, a module to support the specification and deployment of a repertoire of learning methods, and, perhaps less common, an explicit module designed to support evaluation of any combination of the first two.

In the deployment of this architecture to the navigation improvement problem, the missing middle component is that object to be created by various learning algorithms, and then inspected to see whether the learning algorithm has found something "interesting" that can be used to provide navigation improvements, as measured by appropriate evaluation methods. We call the objects created by the application of various learning methods Navigation Compression Models (NCMs).

The idea and name arise from the principle behind learning. All learning creates an abstraction of the initial input data, which somehow represents the input data in some other representation. For example, a classification hierarchy over a number of input instances, is a compressed representation of that input. Similarly, our Navigation Compression Models are simply some representation of an abstract navigation space, determined as a function of selected attributes of a collection of user navigation paths on actual websites.

We can employ NCMs to generate dynamic recommendations for the user based on the pages the user has already visited, as depicted in Figure 1. For this purpose, each NCM can be formulated as a function that generates recommendations based on a given path:

$$NCM = f(path \Rightarrow recommendations)$$

where the recommendations can be as simple as the frequently visited pages, or as complex as the predictions of web documents that might satisfy the user's incoming requests.

For example, in the domain of web log mining, association rules capture the co-occurrence relationships between different web pages based on users' navigation activities.

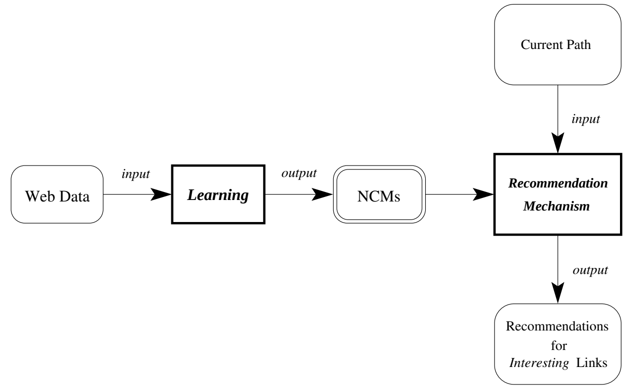

Figure 1: Using NCMs for Dynamic Recommendations

An association-rule-based NCM can be applied for dynamic recommendations as follows: whenever the pages in the antecedent of the association rule have fully appeared in the user's current path, we recommend those "interesting" but unvisited pages in its consequence. As another example, a page cluster represents a set of web pages which are similar to each other in terms of certain metrics such as usage patterns or page content. A cluster-based NCM can be applied as follows: if one or more pages in the cluster have been visited in the user's current path, we recommend those "interesting" but unvisited pages in the cluster.

Simulation-based Evaluation

While we have various learning methods to create NCMs, a proper evaluation method is crucial to assess the improvement of the navigation. However, there has been little effort directed at this problem, and so far we know of no evaluation method which can evaluate navigation improvement quantitatively.

Evaluation Methodology

The basic idea behind the evaluation for navigation improvement is that we can compare the compressed navigation paths with the corresponding paths without compression. However, we can not expect to obtain both of these navigation paths from the same user without the previous navigation experience somehow affecting the later one. Such being the case, we can envisage an ideal experiment conducted as follows:

> Suppose we have a group of user subjects with the same or similar educational background, similar interests, and the same level of web experience. Each subject is asked to fulfill a fixed number of tasks. Each task can be described as finding some specific information starting from a given entry point. Moreover, each task is randomly assigned to a fixed number of users such that half the users are helped with the dynamic recommendation mechanism, and half the users are not. In this way we can collect compressed navigation paths together with the corresponding uncompressed paths from different, but similar, users.

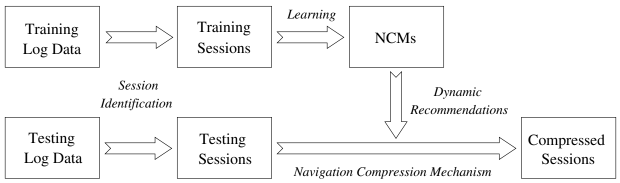

Figure 2: Evaluation Procedure

Such an experiment involves certain problems like user subject composition and task distribution, but these problems are not impossible to address. However, our objective requires a different approach: since intrusiveness has become a big concern in today's WWW communities, we want to minimize the use of intrusive data collection, even for the purpose of evaluation. Based on this consideration, we propose a non-intrusive, quantitative evaluation approach, which is designed to use the commonly collected web log data for both training and testing, as shown in Figure 2. With respect to this idea, the evaluation procedure can be described as follows:

1. Transform original log data (both training and testing) into user sessions. We acknowledge that this work involves enormous challenges in heuristic identification of individual users, sessions, and "relevant" pages (or content pages).
2. Apply a selection of learning algorithms to the training sessions to obtain NCMs. These NCMs have the ability to predict the user's possible interests and therefore can be used for making dynamic recommendations.
3. Apply the NCMs to the testing sessions through a simulation using our navigation compression mechanism, and generate a new set of compressed sessions.
4. Finally, the value of the NCMs is determined by using a family of evaluation functions based on the number of traversal links saved and the potential cost of recommendations.

Evaluation Function

Measuring the efficiency of navigation is not a straightforward task. While this measurement can be perceived by humans through various kinds of visualizations (Niu *et al.* 2003), we also require some quantitative measures which can be obtained automatically from the navigation trails with and without the help of dynamic recommendations. As noted above, these quantitative measurements are obtained by comparing the original and compressed sessions.

One possible approach to the development of such quantitative measurements is based on the number of hyperlinks traversed. Based on this idea, we propose a specific measure called *navigation improvement* (NI), which indicates a quantitative improvement in the navigation efficiency of the compressed sessions over the original ones. Intuitively, NI can be computed as:

$$NI = \frac{N_{org} - N_{com}}{N_{org}} \tag{1}$$

where N_{org} is the number of traversal links in the original sessions, and N_{com} is the number of traversal links in the compressed sessions.

For example, suppose we have obtained an association rule $\{A \rightarrow \underline{D}\}$, where $\underline{D}$ is a content page. Then an original session $S_{org} = \{A, B, C, \underline{D}, E\}$, where B and C are auxiliary pages and $\underline{D}$ is the target page, can be compressed by skipping B and C to obtain $S_{com} = \{A, \underline{D}, E\}$. The navigation improvement for this session would be:

$$NI(S) = \frac{N_{S_{org}} - N_{S_{com}}}{N_{S_{org}}} = \frac{5-3}{5} = 40\%$$

However, this simple measure does not take into account the cost of dynamic recommendations. In an extreme example, suppose the user is provided with one hundred recommendations, and only the last one is useful in guiding the user to the next content page by skipping one auxiliary page. Such being the case, most users will probably consider this recommendation set useless because it will take them less effort simply browsing to the content page with a few more clicks rather than picking it up from the huge recommendation list.

So another quantitative method could compute improvement by estimating the cost of the recommendations and subtracting it from the NI. The basic idea is that the more recommendations we provide to the user, the more cost we should subtract from the NI. Here we define a specific parameter to represent the cost of one recommendation, which is denoted as r_cost. So $\frac{1}{r\_cost}$ recommendations will cancel out the benefit of one saved traversal link.

Note that the determination of r_cost actually reflects the user's (or web designer's) consideration on the tradeoff between size of the recommendation space and proportion of "interesting" information contained in that space. A larger recommendation space generally contains more "interesting" information, but also introduces more "uninteresting" information (or noise) into that space. Therefore, a higher setting of r_cost indicates that lower noise level is preferred, and a lower setting of r_cost indicates that the total amount of "interesting" information is a bigger concern than the noise level.

Based on the above idea, a *cost-based navigation improvement* (NI_c) can be computed as:

$$NI_c = \frac{N_{org} - N_{com} - n_r \times r\_cost}{N_{org}} \tag{2}$$

where n_r is the number of distinct recommendations provided to the user.

In the previous example, given $n_r = 4$ and $r\_cost = 0.1$, which means that during the session there were four distinct recommendations provided to the user, and ten recommendations will cancel out the benefit of one saved traversal link, then the cost-based navigation improvement of the session would be:

$$
\begin{aligned}
NI_c(S) &= \frac{N_{S_{org}} - N_{S_{com}} - n_r \times r\_cost}{N_{S_{org}}} \\
&= \frac{5 - 3 - 4 \times 0.1}{5} \\
&= 32\%
\end{aligned}
$$

Navigation Compression Mechanism

Given the NCMs obtained from the learning process and the evaluation functions for measuring the navigation improvement, the only remaining problem is that of obtaining the compressed sessions from the original ones.

Our evaluation is designed to obtain the compressed sessions by simulating dynamic-recommendation-engaged user navigation behaviors on the original sessions, instead of collecting data from real web usage. This approach requires that we address two issues: (1) the dynamic recommendation mechanism, and (2) the simulation of user actions on recommendations.

The dynamic recommendation mechanism determines how the recommendations are generated and presented to the user. In our case, the recommendations are generated by applying NCMs obtained from the learning process to the user's current path. While the user navigates in the web site, the user path changes continuously, and the recommendations are updated along the way.

How recommendations are presented to the user is also an important issue. First, the number of recommended hyperlinks must be limited. As previously discussed, finding the useful page in a huge recommendation list could be more difficult and time-consuming than simply navigating without any recommendation. For this purpose, we use a specific parameter to represent the maximum number of recommendations that can be presented to the user in one screen, which is denoted as r_limit. Similar to r_cost, the choice of r_limit also reflects the user's preference and could be application-driven.

Since we want to display the most important recommendations at the top of the recommendation list, the second issue of the dynamic recommendation mechanism is to determine the importance, or priority, of each recommended hyperlink. We determine this priority based on the following three criteria, which are applied in order:

1. The exact position in the user path where the recommendation is generated. More recently generated recommendations are given higher priority. This is based on an assumption that in most cases, the user interest at a specific point of time is more highly related to the documents visited close to that point.
2. The quality of the matching between the current user path and the NCM used to generate the recommendation. For example, for association-rule-based NCMs, this matching quality can be defined as length of the antecedent of the rule. For cluster-based NCMs, this matching quality can be defined as the number of common pages between the cluster and the user's current path. Recommendations generated by NCMs with higher matching quality are given higher priorities. This determination is based on our assumption that the more information used for learning, the more accurate results we should obtain.
3. The importance of the NCM used to generate the recommendation. For example, for association-rule-based NCMs, this importance can be defined based on the support and confidence of the rule, e.g., as their Geometric Mean ($\sqrt{sup. \times conf.}$) or Harmonic Mean ($\frac{2 \times sup. \times conf.}{sup. + conf.}$). For cluster-based NCMs, this importance can be defined as the similarity between the recommended document and the center of the cluster used to generate it. NCMs with higher importance values are considered more useful. Therefore, the recommendations they generate are given higher priorities.

The user action model we use is simple, which assumes that the user will follow all the "correct" recommendations. Here "correct" means that the recommended page is a content page to be visited in the session, and there is no other content page between the current position and the recommended page. Though our assumption on user action seems "optimistic", the measured navigation improvement isn't necessarily the best possible result. For example, in real usage the user may jump to any recommended page without such restriction that the content pages must be visited in a particular order.

An example is shown in Figure 3 which illustrates how our navigation compression mechanism works. This example uses association-rule-based NCMs, and each page is represented as a page ID (content pages in parentheses).

It is worth noting that content pages play an important role in this process. First, we recommend only those pages recognized as content pages in the training set. Second, we have to make sure that the compressed sessions keep all those content pages in the original sessions so that the compression actually makes sense.

Experiments

In our experiments we used a set of publicly available server access logs from the music machines web site (currently at "http://machines.hyperreal.org/"). These web logs are used by the "adaptive web sites" project (Perkowitz & Etzioni 1998) and made available online at "http://www.cs.washington.edu/ai/adaptive-data/".

The music machines web site provides a wide variety of information related to music equipments - including images, softwares, schematics, as well as tips and comments from musicians. In our experiments, we have collected six months of server access logs from the music machines web site, from October 1, 1998 to March 31, 1999.

The data preparation process we used is similar to that as described in (Zheng, Niu, & Goebel 2002). The results show that each month the music machines web server generated approximately one million records and 40000 *useful* sessions (1<*session length*≤100) with an average session length of approximately 7.43.

The identification of content pages is itself a difficult problem. In our experiments, we used the *Reference Length*

Original Session	:	1, 2, (3), 4, (5), 6, (7), 8, (9), 10
Recommendation Content Page Set	:	(5), (7), (9)
Test Content Page Set	:	(3), (5), (7), (9)
NCM	:	(a) $1 \rightarrow 9$
		(b) $2 \rightarrow 5$
		(c) $1, 2 \rightarrow 4, 7$
		(d) $1, 3 \rightarrow 9$ ($importance = 0.5$)
		(e) $2, 3 \rightarrow 5$ ($importance = 0.8$)

Navigation Path	Recommendation List	NCMs Applied
1	(9)	(a)
1, 2	(7), (5), (9)	(c), (b)
1, 2, (3)	(5), (9), (7)	(e), (d)
1, 2, (3), (5)	(9), (7)	
1, 2, (3), (5), (7)	(9)	
1, 2, (3), (5), (7), (9)	(null)	
1, 2, (3), (5), (7), (9), 10	(null)	

Compressed Session	:	1, 2, (3), (5), (7), (9), 10

Figure 3: Navigation Compression Example

method (Cooley, Mobasher, & Srivastava 1999), and we applied a clustering algorithm to determine the cutoff viewing time (assuming that any content page has a longer viewing time than any auxiliary page) instead of assigning an arbitrary value to it. An experiment on a "travel study" data set (in which all content pages were manually labeled by the users, see (Zhu, Greiner, & Häubl 2003)) showed that our cluster-based Reference Length method can identify the "true" content pages with $precision = 40\%$ and $recall = 73\%$. Though the recorded viewing time of the "travel study" data set was not perfectly accurate, the result still shows that our method makes sense. However, the "travel study" data set is not suitable for our navigation improvement experiment because its duration was too short (45 minutes).

In each case of the experiments, the previous one-month logs were used as the training set, and the later one-month logs were used as the test set. We report experiments with two kinds of NCMs, one from association rules and another one from co-occurrence oriented page clusters generated using PageGather (Perkowitz & Etzioni 1998). In both cases, we use $r\_cost = 0.05$ and $r\_limit = 10$.

The experimental results are shown in Table 1. In this table, each navigation improvement (both NI and NI_c) is represented as three values $<v/v_c/v_r>$, where v is the navigation improvement on all sessions, v_c is the navigation improvement on those sessions with at least one content page, and v_r is the navigation improvement on those sessions with at least one correct recommendation. It can be expected that $v \leq v_c \leq v_r$.

At first glance of the results, one might conclude that cluster-based NCMs are more suitable than association-rule-based NCMs for the application of improving user navigation by dynamic recommendations: with a fairly small number of page clusters, we can obtain navigation improvements (both NI and NI_c) even better than those obtained with a much larger number of association rules.

However, further investigation showed that cluster-based NCMs tend to generate many more recommendations than association-rule-based NCMs. In other words, while cluster-based NCMs generate more "correct" recommendations, at the same time they also generate many more "incorrect" recommendations. For example, in a one-month experiment 500 clusters generate an average of 188000 recommendations (in which 15300 are "correct"), while 2000 rules generate an average of only 112000 recommendations (in which 10200 are "correct"). Such being the case, we can expect that setting a higher recommendation cost (r_cost) will reduce the NI_c of cluster-based NCMs more than the association-rule-based NCMs.

The underlying reason behind this result is that clusters tend to have much larger coverage than association rules. The coverage of a set of association rules or clusters is defined as the number of distinct items included in them. The coverage of association rules is generally small. For example, a 3-itemset can generate up to 12 association rules but these rules only have a coverage of three. In our case, the coverage of 500 clusters can be three times larger than that of 2000 association rules. For this reason, the number of pages that can be recommended by association rules is highly limited.

Note that our experimental results are indeed determined by a number of heuristics and parameter settings, including the heuristics for user identification, session identification, content page identification, and the settings of r_cost and r_limit. Different choices on these heuristics and parameter settings can undoubtedly lead to different results. The appropriate choices of these heuristics and parameters should be task-determined, and in most cases can only be obtained empirically.

Summary and Conclusions

We have presented an application of web usage mining to improving user's web navigation through dynamic recommendations, and have developed the idea of Navigation Compression Models (NCMs) to represent the results of

Table 1: Experimental Results

Association-rule-based NCMs

Training Data	Test Data	#Rules	NI	NI_c
10/1998	11/1998	2008	4.50% / 4.86% / 10.40%	2.47% / 2.88% / 8.23%
11/1998	12/1998	2015	4.33% / 4.69% / 9.86%	2.18% / 2.60% / 7.60%
12/1998	01/1999	2057	4.09% / 4.43% / 9.87%	1.99% / 2.39% / 7.67%
01/1999	02/1999	2077	3.86% / 4.19% / 9.13%	1.92% / 2.31% / 7.15%
02/1999	03/1999	2062	4.32% / 4.73% / 10.50%	2.16% / 2.63% / 8.26%
mean±standard deviation			4.22±0.25% / 4.58±0.27% / 9.95±0.55%	2.14±0.21% / 2.56±0.22% / 7.78±0.47%

Co-occurrence oriented Cluster-based NCMs

Training Data	Test Data	#Clusters	NI	NI_c
10/1998	11/1998	491	6.94% / 7.50% / 11.96%	3.48% / 4.14% / 8.81%
11/1998	12/1998	498	7.01% / 7.60% / 11.80%	3.50% / 4.21% / 8.66%
12/1998	01/1999	535	6.73% / 7.29% / 11.75%	3.20% / 3.88% / 8.64%
01/1999	02/1999	509	6.56% / 7.12% / 11.48%	3.27% / 3.95% / 8.58%
02/1999	03/1999	494	6.76% / 7.39% / 11.73%	3.07% / 3.85% / 8.49%
mean±standard deviation			6.80±0.18% / 7.38±0.19% / 11.74±0.17%	3.30±0.18% / 4.00±0.16% / 8.64±0.12%

learning "better" navigation paths from web usage data.

We proposed a simulation-based approach to evaluating the effectiveness of NCMs non-intrusively, i.e., without interaction with users or human experts. We presented a navigation compression mechanism and a family of evaluation functions which measure the navigation improvement based on the number of traversal links and the potential cost of recommendations.

There are several directions for extending this work. The identification of content pages is still a challenging problem. In addition to the NCMs used in this paper, other kinds of NCMs created with different learning methods can also be applied to examine their effects on navigation improvement. Also, it will be interesting to examine the possibility of combining NCMs obtained from different learning methods. There could be other valuable evaluation functions, and the recommendation mechanism can be further refined, e.g., giving larger penalties to more important recommendations by assigning higher r_cost to recommendations at the top of the recommendation list.

Note that the dynamic recommendation mechanism used in our experiments makes recommendations *whenever it can*. But we can imagine a more "intelligent" mechanism that makes recommendations only *when the user needs help*. Such a functional change involves the determination of the user's "lostness moment". There has been some interesting work in this problem (Gwizdka *et al.* 2004), which might help better understand user navigation and lead to a more "intelligent" recommendation mechanism.

The experiments we have conducted used heuristically identified content pages and a simplified user model. We hope to collect further "real" data with user-labeled content pages and recommendation-enabled user access logs, and use that data to validate the various methods and heuristics used in our experiments.

Acknowledgments

Our work is supported by the Canadian Natural Sciences and Engineering Research Council (NSERC), and by the Alberta Ingenuity Centre for Machine Learning (AICML).

References

Cooley, R.; Mobasher, B.; and Srivastava, J. 1999. Data Preparation for Mining World Wide Web Browsing Patterns. *Journal of Knowledge and Information Systems* 1(1).

Gwizdka, J.; Spence, I.; Takeshita, H.; and Seergobin, K. 2004. Quantifying Navigation. Exhibition Poster, CASCON 2004.

Mobasher, B.; Cooley, R.; and Srivastava, J. 1999. Automatic Personalization Based on Web Usage Mining. Technical Report TR99-010, Department of Computer Science, Depaul University.

Niu, Y.; Zheng, T.; Goebel, R.; and Chen, J. 2003. WebKIV: Visualizing Structure and Navigation for Web Mining Applications. In *2003 IEEE/WIC International Conference on Web Intelligence (WI'03)*.

Perkowitz, M., and Etzioni, O. 1998. Adaptive Web Sites: Automatically Synthesizing Web Pages. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*.

Resnick, P.; Iacovou, N.; Suchak, M.; Bergstorm, P.; and Riedl, J. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, 175–186. Chapel Hill, North Carolina: ACM.

Zaïane, O.; Xin, M.; and Han, J. 1998. Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. In *Advances in Digital Libraries (ADL'98)*, 19–29.

Zheng, T.; Niu, Y.; and Goebel, R. 2002. WebFrame: In Pursuit of Computationally and Cognitively Efficient Web Mining. In *PAKDD'02*.

Zhu, T.; Greiner, R.; and Häubl, G. 2003. An Effective Complete-Web Recommender System. In *The 12th International World Wide Web Conference (WWW2003)*.