Combining article content and Web usage for literature recommendation in digital libraries

San-Yih Hwang and Shi-Min Chuang

The authors

San-Yih Hwang and **Shi-Min Chuang** are based in the Department of Information Management, National Sun Yat-Sen University, Kaohsiung, Taiwan.

Keywords

Digital libraries, Literature

Abstract

In a large-scale digital library, it is essential to recommend a small number of useful and related articles to users. In this paper, a literature recommendation framework for digital libraries is proposed that dynamically provides recommendations to an active user when browsing a new article. This framework extends our previous work that considers only Web usage data by utilizing content information of articles when making recommendations. Methods that make use of pure content data, pure Web usage data, and both content and usage data are developed and compared using the data collected from our university's electronic thesis and dissertation (ETD) system. The experimental results demonstrate that content data and usage data are complements of each other and hybrid methods that take into account of both types of information tend to achieve more accurate recommendations.

Electronic access

The Emerald Research Register for this journal is available at
www.emeraldinsight.com/researchregister

The current issue and full text archive of this journal is available at
www.emeraldinsight.com/1468-4527.htm

1. Introduction

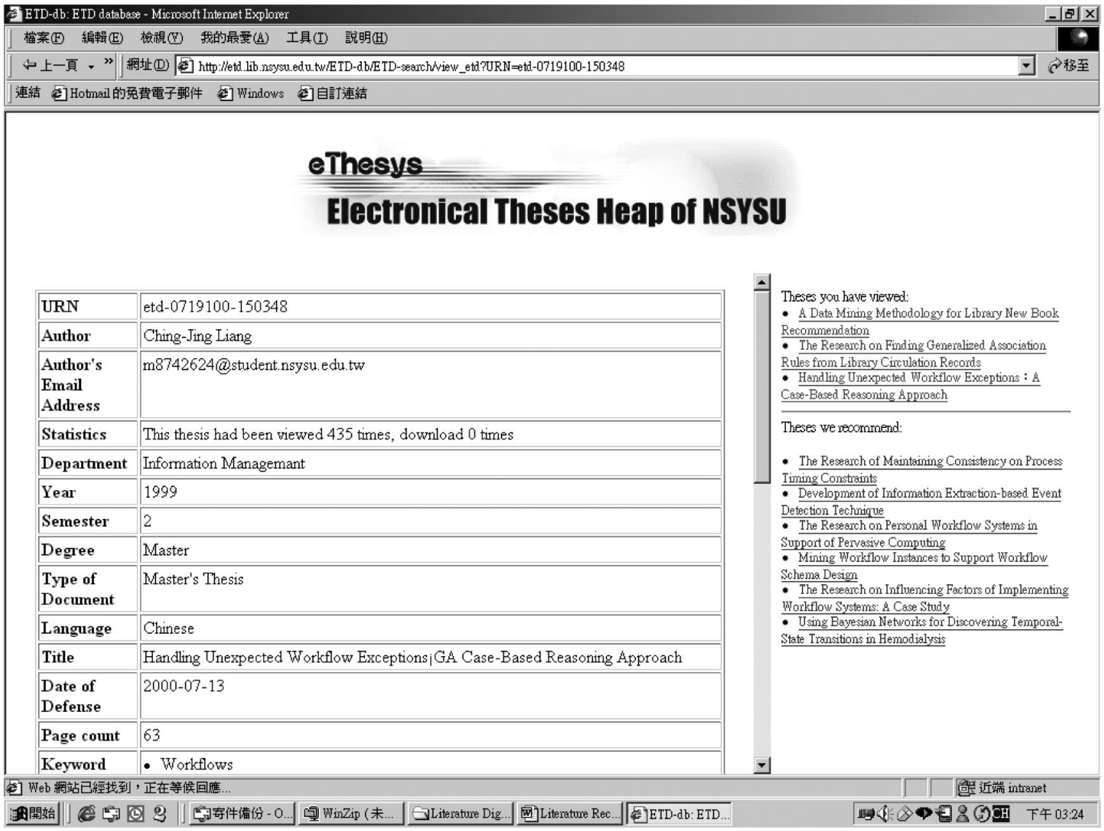
The past few years have seen the emergence of many recommendation systems intended to provide personal recommendations for various types of products and services. However, many of these recommendation techniques are not suitable for recommending literatures in digital libraries because they rely on either the explicit specification of users' interests or the implicit derivation from users' rating scores (or called relevance feedback) on sample items. This is due to the fact that identifying an individual user of a literature digital library is generally not possible, since many literature digital libraries are freely available on the Internet and users can search or browse articles without having to identify themselves. Even for proprietary literature digital libraries, many users gain access via site subscriptions, making it difficult to track an individual's (long-term) browsing behavior. Our previous work proposed using a task-focused approach for online recommending articles in a literature digital library, in which a task profile (i.e. a set of recently accessed articles) rather than the long-term interest profile is used to facilitate recommendations (Hwang *et al.*, 2003). A screenshot of our prototyping Web literature recommendation system, incorporated into our university ETD system (NSYSU-ETD, n.d.), is shown in Figure 1, in which the active user had reviewed three articles, and the literature recommendation system recommended another six articles. Once the active user browses another article, the content of the recommendation frame will update accordingly.

The recommendation framework proposed in our previous work (Hwang *et al.*, 2003) is a usage-based approach, which utilizes the Web usage log that contains users' access records for making recommendations. Two algorithms, namely hypergraph partitioning and association rule mining were developed and then evaluated by applying the Web usage log collected from NSYSU-ETD. It has been shown the hypergraph partitioning approach outperforms the association rule mining counterpart in various settings. However, the empirical results also showed that the usage-based approach suffers from a low-coverage problem – only one-tenth of the articles collected from NSYSU-ETD were recommendable. This is because an article is recommendable by a usage-based approach only

Revised article received 22 April 2004
Accepted for publication 30 April 2004
This work is supported in part by the National Science Council of Taiwan under grant no. NSC92-2622-E-110-016-003.

Online Information Review
Volume 28 · Number 4 · 2004 · pp. 260-272
© Emerald Group Publishing Limited · ISSN 1468-4527
DOI 10.1108/14684520410553750

Figure 1 Page view of an article in NSYSU-ETD


ETD-db: ETD database - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
網址(D) http://etd.lib.nsysu.edu.tw/ETD-db/ETD-search/view_etd?URN=etd-0719100-150348
移至
連結 Hotmail 的免費電子郵件 Windows 自訂連結
eThesys
Electronical Theses Heap of NSYSU
Web 網站已經找到，正在等候回應...
近端 intranet
開始
寄件備份 - O...
WinZip (未...
Literature Dig...
Literature Rec...
ETD-db: ETD...
CH
下午 03:24

URN	etd-0719100-150348
Author	Ching-Jing Liang
Author's Email Address	m8742624@student.nsysu.edu.tw
Statistics	This thesis had been viewed 435 times, download 0 times
Department	Information Managemant
Year	1999
Semester	2
Degree	Master
Type of Document	Master's Thesis
Language	Chinese
Title	Handling Unexpected Workflow Exceptions¡GA Case-Based Reasoning Approach
Date of Defense	2000-07-13
Page count	63
Keyword	• Workflows

Theses you have viewed:

- A Data Mining Methodology for Library New Book Recommendation
- The Research on Finding Generalized Association Rules from Library Circulation Records
- Handling Unexpected Workflow Exceptions：A Case-Based Reasoning Approach

Theses we recommend:

- The Research of Maintaining Consistency on Process Timing Constraints
- Development of Information Extraction-based Event Detection Technique
- The Research on Personal Workflow Systems in Support of Pervasive Computing
- Mining Workflow Instances to Support Workflow Schema Design
- The Research on Influencing Factors of Implementing Workflow Systems: A Case Study
- Using Bayesian Networks for Discovering Temporal-State Transitions in Hemodialysis

when it appears sufficient times with some other articles in the Web usage log, whereas only one-tenth of the tested articles satisfied this condition.

1.1 Contributions

This paper reports our work to remedy the low coverage problem. It introduces several content-based methods that look for articles similar to those currently browsed by the active user in their metadata/full-text and hybrid methods that combine content-based and usage-based approaches for literature recommendation. It shows that articles' contents are precious with respect to literature recommendation, and a well-developed content-based method may outperform a usage-based one. In fact, content information and usage information are complementary to each other. The performance evaluation through real-world data reveals that hybrid approaches that make use of both types of information achieve the best performance.

This paper is structured as follows. Related work is described in section 2. Section 3 discusses issues related to content-based recommendation and presents several content-based approaches. Hybrid approaches are described in section 4. Evaluation results are presented in section 5. Finally, section 6 summarizes this paper and discusses our future research directions.

2. Related work

It has long been recognized that most users are incapable of specifying precisely their needs in queries, and interest profiles, when acquired, effectively facilitate the selection of documents that meet users' information needs (Myaeng and Korfhage, 1990). The most straightforward way of acquiring interest profiles is to have the users explicitly specify their interests. An interest profile can be represented as a vector with each element indicating the preference on a term (Myaeng and Korfhage, 1990), or as a set of rules, where each rule prescribes an appropriate action taken when a condition specified on content-based factors holds (Malone *et al.*, 1987; Fischer and Stevens, 1991).

However, acquiring correct interest profiles are difficult because users may not be conscious about their interests and may not be willing to devote much effort in creating a decent interest profile. In addition, interest profiles may change over time. It has been shown that interest profiles explicitly contributed by users often result in less accurate document selection than stereotyped interest profiles (Shepherd *et al.*, 2001; Kuflik *et al.*, 2003). A body of research exist that address the problem of automatic generation of interest profiles from users' (explicit or implicit) relevance feedback on some selected documents. Explicit relevance feedback is the ratings explicitly provided by users to indicate their preferences on some documents. Implicit relevance feedback is derived by observing users actions. For example, Pennock *et al.* (2000) examined users' actions in Citeseer, a digital library for scientific literatures, and assigned 1 point for adding an article to a profile, 0.5 points for downloading an article, and −1 point for ignoring a recommended article. A typical approach to establish interest profiles from relevance feedback is to apply machine learning techniques to learn the content features of users' interests. This approach is commonly known as information filtering or content-based filtering. Content-based filtering establishes a user's interest profile by analyzing the content features of his preferred items. A user's interest profile can be represented as a vector with each element indicating the user's preference on a selected term (Lang, 1995) or a probability model that describes the probability that the user likes a content item (Mooney and Roy, 2000). The former approach measures the relevance of a given content item and a user's interest as the similarity of this recommendable item to the user's interest profile, while the later simply applies the content item to the pre-computed probability model for computing the relevance. Finally, items that have a high degree of relevance are recommended to the user. Content-based filtering is typically applied to recommend products that have parsable content or description, including Web pages, for example Syskill and Webert (Pazzani *et al.*, 1996) and Siteseer (Rucker and Polanco, 1992), textual documents (Krulwich and Burkey, 1997), news (Lang, 1995), and books (Mooney and Roy, 2000). The main challenges of content-based filtering approaches are the identification of item features from content or description and the development of user interest profiles that distinguish preferred items from disliked ones.

Content-based filtering approaches consider only a given user's preference in making recommendations. Another approach for filtering is to consider the social features of users' interests, commonly known as collaborative filtering. The collaborative filtering recommends items to a user by taking into account other users' preferences. Preferences of un-rated items are predicted for a user based on a combination of known ratings from other users. Due to the simplicity and the effectiveness found in some empirical studies, collaborative filtering is by far the most popular approach used in today's recommendation systems. There are two broad categories of approaches for estimating the preference of an unseen item to a user: memory-based and model-based (Breese *et al.*, 1998). The memory-based approach uses a rating matrix, with rows being users and columns being items, to represent users' ratings on items. It computes a weighted sum on rows or columns of the rating matrix for predicting the preference of a user to an item. Possible weighting schemes include correlation, cosine and regression. The model-based approach first structures users' preferences as a probabilistic model, and then applies this model to predict the probability that a user likes an item. Commonly used probability models include Bayesian classifiers, support vector machines, decision trees, and neural networks.

Content-based and collaborative filtering approaches are not mutually exclusive to each other, and there have been many efforts to integrate them in order to obtain more accurate recommendations. One example is the Fab system, in which content information of items associated with users in the transactional data is part of the user representation and content-based approaches are used to formulate user similarities in a collaborative filtering (Balabanovic and Shoham, 1997). Further information such as users' demographics can also be considered in making recommendations, thereby forming an integrated recommendation framework (Pazzani, 1999; Ansari *et al.*, 2000; Huang *et al.*, 2004).

The many research efforts described above were devoted to the acquisition of users' long-term interests. In contrast, users also have short-term interests, which are referred to as the immediate information need for the task at hand. Short-term interests may or may not relate to long-term interests, and thus it is inadequate to derive a user's task profile from his previous ratings or historical data. Instead, a task profile should be dynamically specified by a list of example documents that are related to the task. In its simplest form, the task profile of a user can be regarded as a single document that the user is currently looking at. When a user chooses to browse a document A, those documents that are either similar to A in their content or often accessed together with A by other users are recommended. Such a function has already been provided by many digital libraries (e.g. Citeseer and the ACM Digital Library). The task profile of a user can be extended to include a set S of documents that the user recently accessed, and the goal becomes to recommend a set of documents whose contents are similar to and/or that are often accessed together with the documents in *S*. This approach has been widely applied to the recommendation of Web pages (Srivastava *et al.*, 2000).

Recommendation techniques based on Web usage logs have been intensively explored in the literature. Several approaches have been proposed for recommending Web pages based on the Web page associations discovered by Web-usage mining algorithms (Yan *et al.*, 1996; Mobasher *et al.*, 1999; Pitkow and Pirolli, 1999; Sarwar *et al.*, 2000; Deshpande and Karypis, 2001; Yang *et al.*, 2001). While these approaches vary in their details, they follow the same recommendation framework, which starts with the preparation of Web usage log, followed by the identification of aggregate usage patterns, and ended by the recommendation which looks into the similarity between the set of recently accessed Web pages of an active user and the collected aggregate usage profiles. Cooley *et al.* (1999) proposed and compared various techniques for cleaning a Web usage log and dividing it into a number of transactions, each of which represents a semantically meaningful task for surfing the Web site. Several types of aggregate usage patterns, including association rules (Sarwar *et al.*, 2000), sequential patterns (Pitkow and Pirolli, 1999; Deshpande and Karypis, 2001), and clusters (Yan *et al.*, 1996; Mobasher *et al.*, 1999), have been explored for providing personal Web pages to an active user. For example, Sarwar *et al.* (2000) utilized association rules in making recommendations. For a given user session *S*, their approach ranks the set of rules in descending order of their confidences whose antecedents match *S* and sequentially select the *N* Web pages appeared in the consequents of these rules. Deshpande and Karypis (2001) proposed a generative Markov model from a set of sequential patterns. Given a user session *S*, the *N* Web pages that have the highest probability (inferred from the Markov model) for directly following *S* are selected. Mobasher *et al.* (1999) proposed to cluster Web pages based on how often they occur together across user transactions in a Web usage log. A clustering algorithm, Association Rule Hypergraph Partitioning (ARHP), was proposed to efficiently cluster Web pages without requiring dimensionality reduction as a preprocessing step. ARHP starts with the identification of large itemsets, each of which contains Web pages often accessed in the same session. Each such large itemset is viewed as a hyperedge with weight being the interest of the itemset. Then a hypergraph partitioning algorithm is applied to partition the set of Web pages into disjoint clusters of Web pages. Web pages in the same cluster are more similar in the sense that they are more likely to be accessed together in the same session. To reflect the fact that an article may indeed interest more than one group of users, some articles are added back to clusters, resulting in overlapping clusters. A recommendation score of each Web page *w* is computed by considering the similarity between the current user session and the cluster *C* to which *w* belongs and the coherence weight of *w* with respect to *C*. The top *N* Web pages for recommendation are those with the *N* highest recommendation scores.

Mobasher *et al.* (2000) extended the usage-based Web personalization framework to incorporate content profiles into the recommendation process. To obtain content profiles, they clustered content features of Web pages by treating each feature as an *n*-dimensional vector over the space of *n* Web pages and applying multivariate K-means clustering technique. The group profile of a feature cluster is a set of Web pages that carry higher (TF/IDF) weights on the features of the cluster. The same recommend process as proposed in Mobasher *et al.* (1999) is then followed.

Hwang *et al.* (2003) extended the personalization techniques based on Web usage mining to literature recommendation of digital libraries. They considered that a literature digital library is better visualized as a set of articles rather than a directed graph of Web pages and subsequently revised the way in which Web usage log is cleaned. Furthermore, since literature articles are incrementally inserted into a digital library, a non-uniform support threshold scheme, originally proposed in Liu *et al.* (1999), was adopted to generate frequent large items. Finally, the association rule-based and the clustering-based approaches as described in Sarwar *et al.* (2000) and Mobasher *et al.* (1999) respectively are proposed for recommending articles in a literature digital library.

3. The content-based approach

3.1 Article similarities

The information about an article can be divided into a number of content categories. Typical content categories include title, keywords, abstract, and full-text (if any). In each content category, an article is represented as a vector. The typical Information Retrieval procedure is adopted for deciding the vector, which begins by filtering out irrelevant, poor terms through consulting a stop-list that contains 300 words (Fox, 1992), followed by computing the TF/IDF value of each selected term.

After the conversion, each article a_i is represented by a number of vectors $v_{i,1}$, $v_{i,2}$, …, $v_{i,c}$, one for each content category. The weight of each content category that indicates its importance toward the similarity of two articles is automatically computed from past browsing history, specifically the Web usage log. The rationale behind this approach is that a content category should carry more weight if articles that frequently appeared together across user sessions in the Web usage log are more similar in that category. Specifically, a collection of frequent itemsets of articles are identified from the Web usage log. For a given frequent itemset S_i, the average similarity $set_sim_j(i)$ among all pairs of articles with respect to each content category f_j is computed. Each $set_sim_j(i)$ is then such normalized that $\sum_1 \leq_{\mathrm{j}} \leq_{\mathrm{c}}\ set_simj(i) = 1$, where c is the number of content categories. The weight w_j of a content category f_j is the average of its normalized itemset similarities across all frequent itemsets. Formally, assuming that there are m frequent itemsets and c content categories, the weight of content category j, denoted w_j, is computed as follows:

$$w_j = \underset{1\leq i\leq m}{Average}\,(set\_sim_j(i)), 1 \leq j \leq c$$

The similarity $sim(a_1, a_2)$ of two articles a_1 and a_2 is then defined as the weighted sum of vector similarities in all content categories:

$$sim(a_1, a_2) = \sum_{1\leq j\leq c} w_j \cdot vec\_sim(v_{1,j}, v_{2,j}),$$

where $v_{1,j}$ ($v_{2,j}$) is the vector of article a_1 (a_2) in content category j, and $vec_sim(v_{1,\mathrm{j}}, v_{2,\mathrm{j}})$ is the cosine of the angle between vectors $v_{1,j}$ and $v_{2,j}$.

3.2 Multiple reference points (MRP) approach

Korfhage (1997) proposed a document matching technique based on multiple reference points (MRP), which refer to any defined points or concepts against which a document can be judged. Queries and user profiles are obvious examples of reference points. The proposed approach selects documents that are more similar to the set of reference points. The MRP approach can be directly extended to recommend literatures by viewing each article in the current user session as a reference point and considering multiple vectors associated with each article. Let the current user session be $S = (S_1, S_2\ldots, S_n)$. Following the ellipsoidal model (Korfhage, 1997), the distance between an article A and the current user session S, denoted $Dist(S, A)$, is defined as follows:

$$Dist(S, A) = \frac{\sum_{i=1}^{n} |A, S_i|}{n},$$

The distance between an article A and a reference point S_i, denoted $|A, S_i|$, can be derived from their similarity ($sim(A, S_i)$) as described in section 3.1 by applying any function that maps from $[0, 1]$ to $[0, \infty]$. One such a mapping function is $|A, S_i| = -\ln sim(A, S_i)$ (Korfhage, 1997). Articles with shorter distances are recommended to the user. This method is called distance-based MRP.

The distance function in the above scheme can be replaced by the similarity function, resulting in a new method called similarity-based MRP. Analogously, the similarity between an article A and the current user session S is defined as follows:

$$Sim(S, A) = \frac{\sum_{i=1}^{n} sim(A, S_i)}{n}.$$

The above two schemes are referred to as equal-priority MRPs, as they do not distinguish referenced articles by their orders in a user session when computing similarities. Another scheme: prioritized MRP, in which articles that are more recently browsed carry higher weights than those browsed earlier, is also considered. In this scheme, a browsed article carries weight α times of the remaining priority, $0\leq \alpha \leq 1$, while all the previous articles together are given $(1 - \alpha)$ times of the remaining priority. Let t_i, $1 \leq t \leq n$, be the priority assigned to the i'th article S_i in S (articles with larger i are more recently browsed). Then, for a given α, $0 \leq \alpha \leq 1$, t_i is determined as follows:

$$\left\{\begin{array}{l} t_1 = (1-\alpha)^{n-1} \\ t_i = \alpha(1-\alpha)^{n-i}, 1 < i \leq n \end{array}\right.$$

Thus, for prioritized, similarity-based MRP, the similarity between an article A and a user session S is defined as:

$$Sim(S, A) = \sum_{i=1}^{n} t_i \cdot sim(A, S_i).$$

For prioritized, distance-based MRP, the distance function $Dist(S, A)$ is similarly defined:

$$Dist(S, A) = \sum_{i=1}^{n} t_i \cdot |A, S_i|$$

The four variants of the MRP approach are summarized in Table I.

Let $S' = (S_1, S_2\ldots, S_n-1)$ be the current user session. When a new article S_n is browsed, the current user session becomes $S = (S_1, S_2\ldots, S_n)$, and the ranks of articles may change. Fortunately, the similarity $Sim(S, A)$ between an article A and S can be incrementally computed from $Sim(S', A)$. The following shows how one can compute $Sim(S, A)$ from $Sim(S', A)$ in constant time for equal-priority, similarity-based MRP:

Table I The four schemes of the MRP approach

Variant	Function	Article rank
Equal-priority, distance-based MRP	$\mathrm{Dist}(S,A) = \frac{\sum_{i=1}^{n} \lvert A,S_i \rvert}{n}$	Smaller distance, higher rank
Equal-priority, similarity-based MRP	$\mathrm{Sim}(S,A) = \frac{\sum_{i=1}^{n} \mathrm{sim}(A,S_i)}{n}$	Larger similarity, higher rank
Prioritized, distance-based MRP	$\mathrm{Dist}(S,A) = \sum_{i=1}^{n} t_i \cdot \lvert A,S_i \rvert$	Smaller distance, higher rank
Prioritized, similarity-based MRP	$\mathrm{Sim}(S,A) = \sum_{i=1}^{n} t_i \cdot \mathrm{sim}(A,S_i)$	Larger similarity, higher rank

Notes: $S = (S_1, S_2 \ldots, S_n)$ is the current user session, and A is an article ($t_1 = (1-\alpha)^{n-1}$, and $t_i = \alpha(1-\alpha)^{n-i}$, $1 < i \leq n$)

$$\begin{aligned} Sim(S,A) &= \frac{\sum_{i=1}^{n} sim(A,S_i)}{n} = \frac{n-1}{n} \cdot \frac{\sum_{i=1}^{n-1} sim(A,S_i) + sim(A,S_n)}{n-1} \\ &= \frac{n-1}{n} \cdot \left(Sim(S\text{'},A) + \frac{sim(A,S_n)}{n-1}\right) \\ &= \frac{n-1}{n} \cdot Sim(S\text{'},A) + \frac{sim(A,S_n)}{n} \end{aligned}$$

Similar formulae for the other MRP schemes can also be derived. Therefore, given a newly browsed article, the time complexity for computing the similarity between every article and the new user session is O(m), where m is the number of articles. Finding the top N articles takes time O(mN). Such a computation overhead could be substantial, especially for a literature digital library that stores a large number of articles. In the next subsection, we propose clustering-based approaches that aim to perform online recommendation more efficiently.

3.3 Clustering-based approaches

The entire collection of articles in a literature digital library can be viewed as a $m \times n$ matrix C, where m is the number of articles and n is the total number of features across all content categories. Each cell in the matrix, $C(i,j)$, records the weight of feature j of article i (specifically, the TF/IDF value). The goal of clustering-based approaches is to get a set of non-disjoint group profiles, each containing a set of articles and their respective weights representing their closeness to the group. Two clustering-based approaches, namely feature partitioning and article-clustered hypergraph partitioning, are proposed.

3.3.1 Feature partitioning (FP)

Mobasher *et al.* (2000) used K-means to partition features of Web pages, and the collection of features clusters result in a set of non-disjoint Web page clusters. The extension of their approach by considering multiple content categories associated with articles, called feature partitioning (FP), is described as follows:

(1) K-means is applied to partition features of each content category. Let the feature partition of content category i, $1 \leq j \leq c$, be $\{F_{i,1}, F_{i,2}, \ldots, F_{i,\mathrm{p(j)}}\}$, where $p(i)$ is the number of disjoint feature groups in content category i.

The group profile of a feature cluster $F_{i,j}$, denoted $G_{i,j}$, is the set of articles that carry higher weights in $F_{i,j}$, formally defined as follows:

$$G_{i,j} = \{a | 1 \leq a \leq m, weight(a, F_{i,j}) \geq \tau\},$$

where τ is a user-specified significance threshold, and:

$$weight(a, F_{i,j}) = \frac{\sum\limits_{f \in F_{i,j}} C(a,f)}{\sum_{k=1}^{m} \sum\limits_{f \in F_{i,j}} C(k,f)}.$$

The coherence weight of an article a with respect to $G_{i,j}$, also denoted as $weight(a, G_{i,j})$, is defined as the weight of an article a with respect to $G_{i,j}$'s feature set. That is, $weight(a, G_{i,j}) = weight(a, F_{i,j})$.

(2) The match between the current user session S and a group profile $G_{i,j}$ can be defined as the cosine similarity between S and $G_{i,j}$ weighted by w_i, the weight of the content category i. Note that the group profile $G_{i,j}$ is treated as a m-vector $(v_1, v_2, \ldots, v_m)$, where:

$$v_i = \{ \begin{matrix} weight(a), a \in G_{i,j} \\ 0, otherwise \end{matrix},$$

and a session S is represented as a Boolean vector $(s_1, s_2, \ldots, s_m)$, where $s_i = 1$ if the ith article appears in S and 0 otherwise. The similarity between S and $G_{i,j}$ is defined as follows:

$$match(S, G_{i,j}) = w_i \cdot \frac{\sum_{k=1}^{m} v_k \cdot s_k}{\sqrt{\sum_{k=1}^{m} (s_k)^2 \times \sum_{k=1}^{m} (v_k)^2}}.$$

(3) The recommendation score $Rec(S, a)$ of an article a with respect to the current user session $S$is then defined as:

$$Rec(S, a) = \max_{a \in G} \sqrt{weight(a, G) \cdot match(S, G)}.$$

Articles with higher recommendation scores are recommended to the user. Note that the clustering of features and computation of group profiles can be conducted off-line. Thus, the coherent weight $weight(a, G)$ of each article a with respect to each group profile G can also be pre-computed. Articles in each group profile are sorted in descending order of their coherent weights. When a new article is browsed, the similarity $match(S, G)$ between the current session S and each group profile G is computed, and the top N articles are consecutively selected from top articles in all groups. Let k be the number of group profiles. The time complexity for online recommendation given a newly browsed article is $O(N + k)$. This online recommendation procedure is much more efficient than that of MRP.

3.3.2 Article clustered hypergraph partitioning (ACHP)

Another clustering method is to partition the content matrix C horizontally. It is called article clustered hypergraph partitioning (ACHP), which applies the hypergraph partitioning method (Karypis *et al.*, 1997) by treating articles as vertices and sets of close articles as hyperedges. Mobarsha *et al.* (1999, 2000) adopted the same technique for Web page recommendation using Web usage log. Their approach first identifies a collection of large itemsets of Web pages from Web usage log, which are subsequently treated as hyperedges. Instead of finding large itemsets of Web pages from Web usage log, ACHP identifies a set of cliques on close articles from the content matrix. Articles in each clique are similar in their content and form a hyperedge. A clique of articles and its associated weight are defined in the following way:

- A clique is an undirected complete graph $G = (V, E)$ with V representing the set of articles, and $(u, v) \in E$ iff $sim(u, v) \geq \tau$, where $u,v \in V$ and τ is a predefined threshold.
- The weight of a clique Q is defined as the average similarity among all pairs of articles in Q.

A challenge of this approach is on the assignment of τ such that the number of articles in any clique is not too large. It has been shown that locating a clique of maximum size is NP-complete (Garey and Johnson, 1979). However, if the maximum size of cliques is known to be no more than a constant K, enumerating all maximal cliques takes polynomial time (Mobasher *et al.*, 1999). The constant K can be assigned according to the following property.

Property: let V be a set of articles. For a given article $v \in V$, let $Ksim(v, k)$ denotes the k'th largest similarity value between an article v and any other article in V. By setting $\tau = Max_v \in_V (Ksim(v,K-1))$, the maximum size of cliques of the graph $G = (V, E)$ formed by the threshold τ is no larger than K.

Hypergraph partitioning (Hwang *et al.*, 2003) is then followed, resulting in a set of overlapping article clusters. Each article cluster is viewed as a group profile. The coherence weight of an article a with respect to a group profile G to which it belongs is defined as:

$$weight(a, G) = \frac{\sum_{a \in e, e \subseteq G} weight(e)}{\sum_{e \subseteq G} weight(e)},$$

where $weight(e)$ is the weight of a hyperedge e.

The same procedure as described in steps 2 and 3 of the feature partitioning approach is finally used for locating the top N articles. The time complexity for online recommendation given a newly browsed article is again $O(N + k)$, where k be the number of group profiles.

4. Integrating content and usage data for recommendation

The simplest hybrid approach is a loosely coupled one that simply merges the two article lists generated by two distinct recommendation systems, one adopting any content-based method described in section 3 and the other using any usage-based approach described in (Hwang *et al.*, 2003). To answer a top-N query, the hybrid approach retrieves the top βN articles from the usage-based recommendation system and the top $(1 - \beta)N$ articles from the content-based recommendation system, where β is user-specified.

Another hybrid approach is to tightly couple both approaches by considering article similarities based on both content and usage in a coherent framework. Specifically, both content- and usage-based approaches that adopt clustering techniques for grouping articles, namely FP, ACHP, and ARHP, are considered.

4.1 Integration of FP and ARHP

Feature partitioning and association rule hypergraph partitioning are first applied to obtain two sets of group profiles. Let the sets of group profiles generated by FP and ARHP be $\{F_1, F_2, \ldots, F_r\}$ and $\{U_1, U_2, \ldots, U_s\}$ respectively. The union of the two sets consists of $r + s$ group profiles. However, the coherence weights of an article a with respect to a group profile F_i and another group profile U_j, denoted $weight(a, F_i)$ and $weight(a, U_j)$ respectively, could vary substantially, resulting in the domination of one type of group profiles over the other. To prevent such an anomaly, the vector of each group profile is normalized before performing online recommendation. Specifically, the vectors of F_i's are proportionally adjusted such that $Max_{a,F_i}(weight(a,F_i)) = Max_{a,U_j}(weight(a,U_j)$. Finally, the same recommendation procedure as described in section 3.3 is followed for computing recommendation scores.

4.2 Integration of ACHP and ARHP

ACHP and ARHP differ only in the way hyperedges are determined and share the same procedures for hypergraph partitioning and article recommendation. Thus a more coherent integration approach can be developed. Specifically, this approach uses both the cliques derived from article content similarities and large itemsets derived from the Web usage log to form the set of hyperedges in the hypergraph. The same procedure for hypergraph partitioning and article recommendation, as used by both ACHP and ARHP, is followed to recommend top N articles. To avoid bias, the weights of hyperedges are such normalized that the maximum weight of hyperedges formed by content cliques is equal to the maximum weight of hyperedges formed by large itemsets. This tightly coupled approach is called ACARHP.

5. Empirical evaluations

This section reports the result of applying the article contents and Web usage logs of the ETD system at National Sun Yat-sen University (NSYSU-ETD, n.d.) to evaluate the proposed recommendation approaches. NSYSU-ETD runs on PC Solaris 2.7 and uses Apache 1.3.9 as the Web server. Up to May 2003, there were 2,951 theses in the system, among which 2,271 theses have English abstracts. Each thesis includes the information of various fields, including title, author, graduate program, abstract, bibliography, advisors, program committee members, and full-text. We chose four content categories, namely the title, the abstract, the keyword, and the advisor, for our experiments. To evaluate the usage-based approach, we made use of the Web usage log of NSYSU-ETD system collected between 1 January 2003 and 31 May 2003.

5.1 Data preprocessing

5.1.1 Processing the contents of articles

We first parsed the content of each article in the four selected content categories, using PERL module Lingua-Stem-0.50 for stemming the words. Then the top 200 terms (with the highest TF/IDF value) from title, abstract, and keyword are extracted. Regarding the advisor category, we simply included all advisor names as the set of terms. Each article is then represented as four vectors. Detailed information about the three content features is shown in Table II.

We then computed the weight of each feature category by using the method described in the section 3.1. The minimum support threshold was set to 0.12 percent (see below), and the resultant weights are shown in Table III.

5.1.2 Processing the Web usage log

The Web usage log between 1 January 2003 and 30 April 2003 was designated as the training data set, and that collected in May 2003 served as the test data. The training data set contains 52,325 article accesses, while the test data has 23,402 article accesses. After applying the session identification approach and pruning the session of robot access as described in (Hwang *et al.*, 2003), 46,518 user sessions were identified in the training data set. Since user sessions with one article accesses provide no value for article recommendation, we further eliminated them, resulting in 6,175 remaining user sessions.

The first step of the usage-based approach is to identify a collection of large itemsets. Clearly, articles not appeared in any large k-itemsets, $k \geq 2$, will not become candidates for recommendation. However, setting a very low value of minimum support (MIN_{sup}), though resulting in a larger number of candidates for recommendation, may make recommendation inaccurate. We expected each large itemset to be supported by no less than several dozens of user sessions. Thus, we specified MIN_{sup} as 0.12 percent, resulting in a total of 277 articles involved in large two-itemsets. Table IV shows these numbers.

Table II The detailed information about content categories

	Title	Abstract	Keyword
Total distinct words	4,935	18,712	5,303
Significant words	200	200	200

Table III The percentage weights of the four content categories

Title	Abstract	Keyword	Adviser
29.4	26.3	34.1	10.2

Table IV Some numbers for processing Web usage log

MIN_{sup} (%)	0.12
No. of large itemsets	1,725
No. of large-1 itemsets	1,288
No. of large-2 itemsets	360
No. of items in large-2 itemsets	277

5.1.3 Settings of feature partitioning

In each content category, we used K-means to obtain ten feature clusters, from which the corresponding (overlapping) group profiles were derived. Such a setting allows every article to be included in at least one group profile.

5.1.4 Settings of article partitioning

Recall that we used $Ksim(v, k)$ to denote the k'th largest similarity between v and any other articles and set $\tau = Max_{v \in V}(Ksim(v, k - 1))$ for determining edges between articles. In our experiments, we set $k = 25$, which results in $\tau = 0.5504$. Thus, there exists an edge between two articles if and only if their similarity is greater than 0.55. Under such a setting, the number of edges is 1,554, and the number of cliques is 714. The total number of articles that are involved in at least one edge is 1,083. Each clique is then seen as a hyperedge. We partitioned articles into 230 partitions, which result in the best performance in our preliminary experiments. Table V summarizes the total number of articles and the numbers of recommendable articles for ARHP, MRP, FP, and ACHP.

5.2 Performance metrics

To illustrate how we conducted experiments, the following notations are defined. Let T_{eval} be the set of user sessions in the test set, t_{eval} be a user sessions in T_{eval}, and $a_t(i)$ be the i'st article in t_{eval}. Given a window size W_{size}, each session t_{eval} in the test data set is divided into two list: $t_{eval}[W]$ and $t_{eval}[R]$, where $t_{eval}[W]$ is the first W_{size} article accesses of t_{eval} and t_{eval}[R] is the remaining articles in t_{eval}. By treating $t_{eval}[W]$ as the current session, the recommendation system will choose the set t_{pr} of top n articles for recommendation. In our experiments, we set n=15.

The performance metrics we adopted for measuring the quality of recommendation are precision and recall. Precision measures the ratio of the number of recommended articles accessed by a user to the total number of recommended articles, defined as $t_{pr} \cap t_{eval}[R]/t_{pr}$, and recall measures the ratio of the number of recommended articles accessed by a user to the total number of articles liked by the user, defined as $t_{pr} \cap t_{eval}[R]/t_{eval}[R]$. The precision (recall) of a recommendation approach is the average precision (recall) of all user sessions in the test data set.

Table V Number of recommendable articles for each method

	Recommendable articles			
Total articles	**ARHP**	**MRP**	**FP**	**ACHP**
2,271	277	2,271	2,271	1,038

5.3 Experimental results

5.3.1 Comparing MRP methods

This set of experiments aims to evaluate the various schemes of the MRP approach. We first compared the equal-priority, similarity-based MRP and the prioritized, similarity-based MRP. The result is shown in Figure 2 and Figure 3. As can be seen, the equal-priority MRP constantly outperforms the prioritized counterparts under various α values in both precision and recall. This meets our expectation that the order of article accesses in an active user session is not significant. Besides, precision of MRP increases monotonically with the increase of window sizes, while the recall of MRP reaches the highest at window size four and gradually decreases as window size increases. Since larger window size implies more reference points, one might expect MRP to have higher precision and recall when window sizes become larger. However, in our test data set, most user sessions are short, and thus t_{eval}[R] becomes smaller as the increase of window size. Figure 4 depicts the number of user sessions in the test data set of various lengths. Among the user sessions whose lengths are larger than eight, more than half of them are of length nine or ten. In fact, we have observed that when window size was set to eight, recalls of many transactions in the test

Figure 2 Precision values of equal-priority, similarity-based MRP and prioritized, similarity-based MRP under different α values

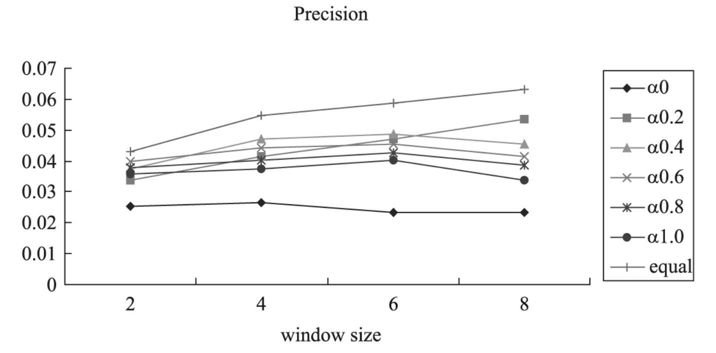


Figure 3 Recall values of equal-priority, similarity-based MRP and prioritized, similarity-based MRP under different α values

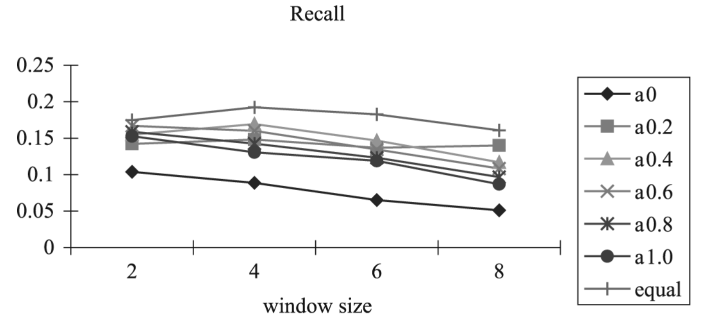
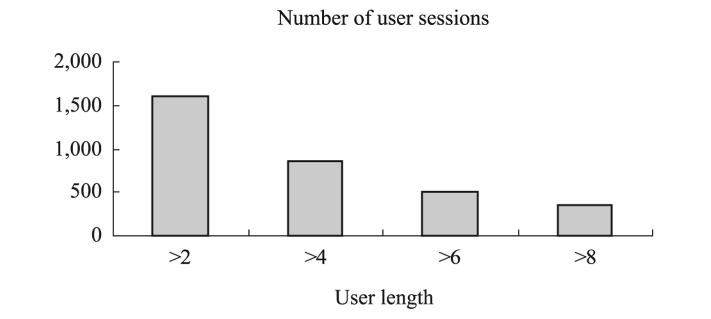


Figure 4 Number of transactions of various lengths in the test data set



data set are 0, due to smaller t_{eval}[R], which explains its deteriorated average recall. In practice, it is conjectured that larger window size will yield more accurate recommendation.

We next compared the equal-weighted, similarity-based MRP and the equal-weighted, distance-based MRP. The result is shown in Figures 5 and 6. It can be seen that the similarity-based scheme constantly outperforms the distance-based counterpart, and this is particularly true with large window sizes. This is because the distance from an article *a* to a set *S* of articles is dominated by the longest distance from *a* to any article in *S*, and for a large window size, the possibility that there exists an article in *S* that is far from *a* is increased. In contrast, the similarity measure, which is basically the logarithm of the corresponding distance measure, gracefully reduces the differences.

Figure 5 Precision values of the equal-priority, distance-based MRP and its similarity-based counterpart under various window sizes

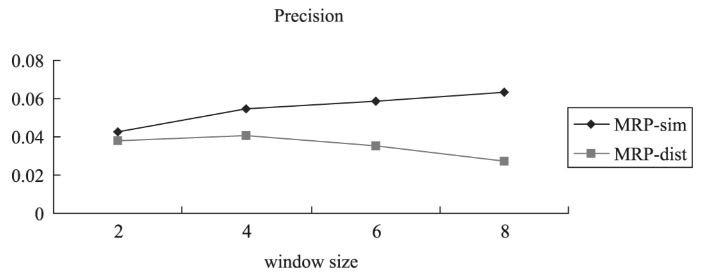
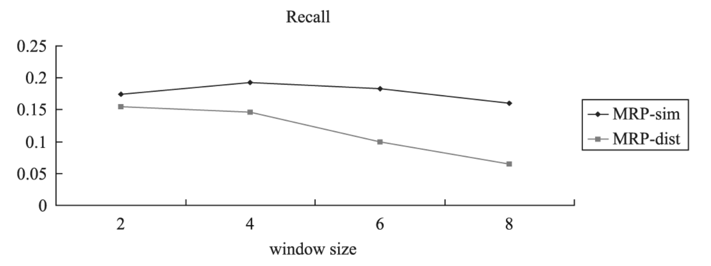


Figure 6 Recall values of the equal-priority, distance-based MRP and its similarity-based counterpart under various window sizes



As the equal-priority, similarity-based MRP resulted in the best performance, it was used in our subsequent experiments. In the following, the term MRP actually refers to the equal-priority, similarity-based MRP.

5.3.2 Comparing content-based methods

This set of experiments aims to compare performance of the three content-based methods, namely MRP, ACHP, and FP, using the association rule hypergraph partitioning method (abbreviated as ARHP) as the benchmark. Their relative performance under various window sizes is shown in Figures 7 and 8.

In both precision and recall, MRP yields the best result, while FP performs the worst. ACHP and ARHP have similar precision and recall values under various operating regions. However, considering the fact that ARHP can recommend only 277 articles, while the article partitioning method is able to recommend 1,038 articles (see Table IV), we conclude that ACHP is superior to ARHP.

We next measured the average running times for online recommendation of various methods. They are shown in Figure 9. As expected, the MRP method has much longer running time than the other clustering methods. While 0.2 seconds for the MRP do not seem to matter in our environment, in a large-scale literature digital library that contains millions of articles with thousands of concurrent users, the running time overhead associated with MRP may become intolerable.

Figure 7 Precision values of various content-based methods under different window sizes, using ARHP as the benchmark

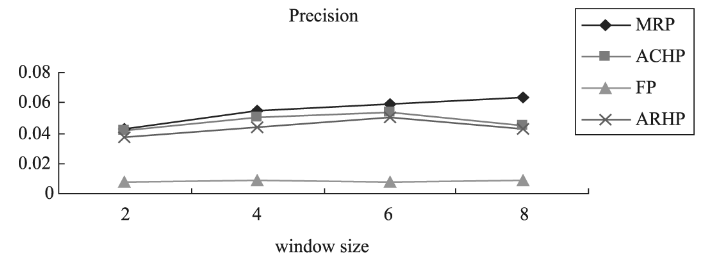


Figure 8 Recall values of various content-based methods under different window sizes, using ARHP as the benchmark

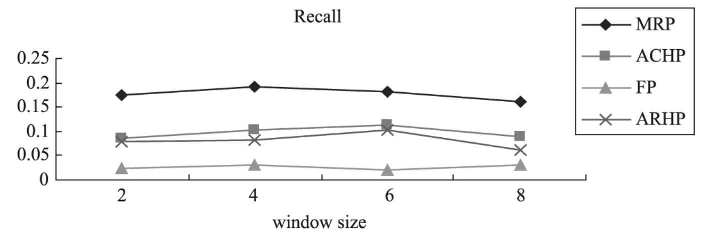
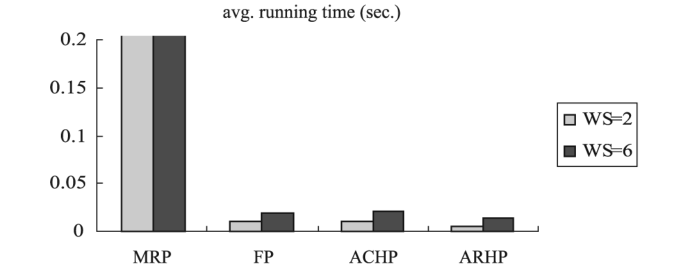


Figure 9 Running times of various content-based methods under two different window sizes



Our experimental results so far lead to the following conclusions:

- For a large-scale literature digital library, ACHP is a good candidate as it is efficient for online recommendation with only a small sacrifice in recommendation accuracy.
- For a small literature digital library with casual usage, MRP is an ideal approach as it achieves the best recommendation accuracy and is easy to implement.

5.3.3 Comparing hybrid approaches

This set of experiments aims to compare the performance of various hybrid methods described in section 4. The loosely coupled hybrid approach combines the top βN articles and the top $(1-\beta)$N articles generated by a usage-based approach (specifically the ARHP) and a content-based approach respectively. Various β values were exercised, and it was found that $\beta = 0.8$ yields the best performance in most cases. Thus, β is set to 0.8 in the subsequent experiments.

As FP performs much worse than the other content-based methods from our previous experimental results, only the other two content-based methods: MRP and ACHP, are integrated with ARHP. Figures 10 and 11 show the performance of MRP, ARHP and the hybrid method that combines MRP and ARHP. It turns out that the hybrid method has the best precisions than MRP and ARHP while still incurs a slightly lower recall than MRP. Figures 12 and 13 show the performance of ACHP, ARHP and the hybrid method that combines ACHP and ARHP. It can be seen that the hybrid method has the best precision and recall. This experiment demonstrates that in most cases, loosely coupled hybrid methods yields better performance than their individual content-based and usage-based methods.

Figure 10 Precision values of the loosely coupled hybrid approach (MRP+ARHP) under different window sizes, using ARHP and MRP as the benchmark

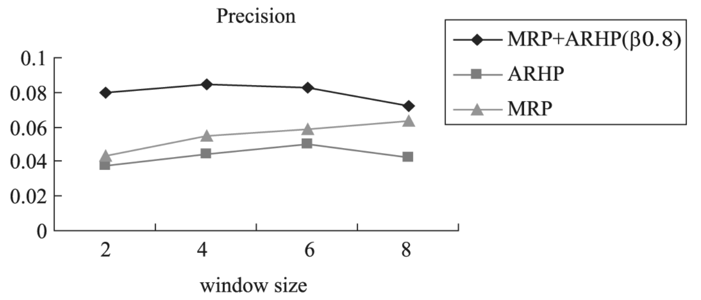


Figure 11 Recall values of the loosely coupled hybrid approach (MRP+ARHP) under different window sizes, using ARHP and MRP as the benchmark

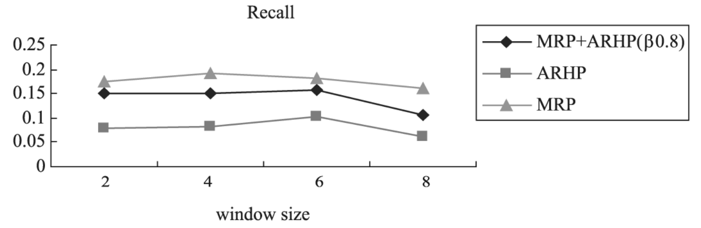


Figure 12 Precision values of the loosely coupled hybrid approach (ACHP+ARHP) under different window sizes, using ARHP and ACHP as the benchmark

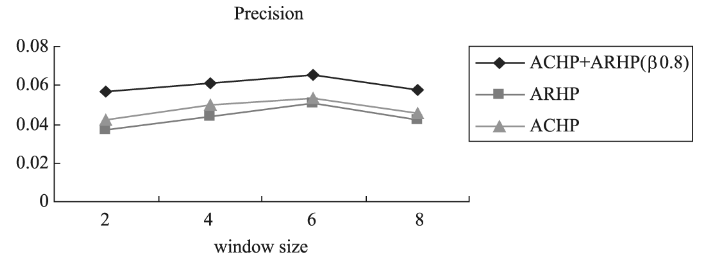
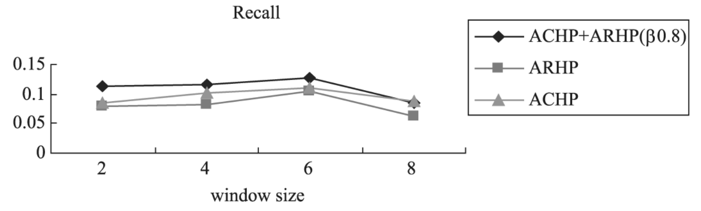


Figure 13 Recall values of the loosely coupled hybrid approach (ACHP+ARHP) under different window sizes, using ARHP and ACHP as the benchmark



We next compared the performance of the tightly coupled hybrid method (ACARHP) with the two loosely coupled hybrid methods described above. Their relative performance is shown in Figures 14 and 15. As shown, ACARHP has better recall than and comparable precision with ACHP+ARHP. When compared to MRP+ARHP, the ACARHP is inferior, especially in precision.

We draw the following conclusions for this set of experiments:

- The loosely coupled method that combines MRP and ARHP achieves the highest precision and recall.

Figure 14 Precision values of various hybrid methods under different window sizes

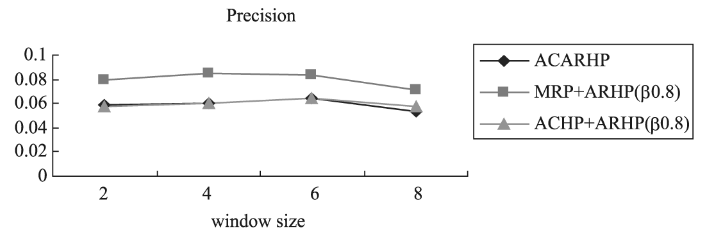
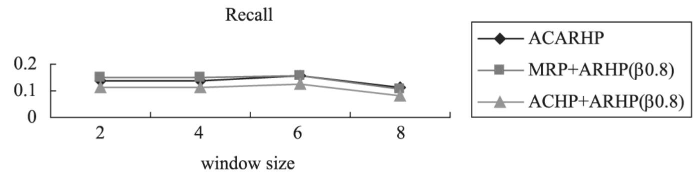


Figure 15 Recall values of various hybrid methods under different window sizes



- For a large scale literature digital library, ACARHP is a promising approach as it is efficient and achieves high precision/recall only second to the MRP-based methods, which takes a much longer time for online recommendation.

6. Conclusions

In this paper, we have developed a suite of literature recommendation approaches for digital libraries that dynamically provide recommendations to an active user when browsing a new article. The focus of the work reported in this paper is on the utilization of content information for designing recommendation algorithms. We have defined content similarity between articles in the context of literature digital libraries and proposed three content-based approaches, namely multiple reference points, feature partitioning, and article clustered hypergraph partitioning. We also proposed various hybrid approaches that integrate content and usage data in making recommendations. These approaches were evaluated using the article collections and Web usage log of an operational electronic thesis system at National Sun Yat-sen University. It has been found that multiple reference points approach achieves a good recommendation accuracy, measured by precision and recall, while suffers from low efficiency. Also the hybrid approaches that utilize both content and usage data of articles were found to generally yield better quality article recommendation than those that make use of only one type of information. In a large-scale digital library with intensive usage, the tightly coupled approach – article clustered and association rule hypergraph partitioning – has been shown to be a promising approach as it provides efficient and effective recommendation.

Our experiments demonstrated the usefulness of clustering articles based on both content and usage data, with respect to literature recommendation. We plan to extend such clustering techniques to automatic ontology learning. The preliminary approach starts by grouping articles pertaining to the same concept in a digital library, followed by the derivation of topic signature for each group. The resultant ontology then serves as the knowledge map of the digital library. We are currently investigating issues and approaches toward the construction of the digital library specific ontology.

References

Ansari, A., Essegaier, S. and Kohli, R. (2000), "Internet recommendation systems", *Journal of Marketing Research*, Vol. 37 No. 3, pp. 67-85.

Balabanovic, M. and Shoham, Y. (1997), "Fab: content-based, collaborative recommendation", *Communications of the ACM*, Vol. 40 No. 3, pp. 66-72.

Breese, J., Heckerman, D. and Kadie, C. (1998), "Empirical analysis of predictive algorithms for collaborative filtering", Technical Report MSR-TR-98-12, Microsoft Research, available at: http://research.microsoft.com/users/breese/algsweb.PS

Cooley, R., Mobasher, B. and Srivastava, J. (1999), "Data preparation for mining World Wide Web browsing patterns", *Journal of Knowledge and Information Systems*, Vol. 1 No. 1, pp. 5-32.

Deshpande, M. and Karypis, G. (2001), "Selective Markov models for predicting Web-page accesses", *Proceedings of the 1st International SIAM Conference on Data Mining*, available at: www.siam.org/meetings/sdm01/pdf/sdm01_04.pdf

Fischer, G. and Stevens, C. (1991), "Information access in complex, poorly structured information spaces", *Proceedings of the ACM International Conference on Human Computer Interaction*, pp. 63-70.

Fox, C. (1992), "Lexical analysis and stoplists", in Frakes, W.B. and Baeza-Yates, R. (Eds), *Information Retrieval: Data Structures & Algorithms*, Prentice-Hall, Englewood Cliffs, NJ.

Garey, M.R. and Johnson, D.S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman & Company, San Francisco, CA.

Huang, Z., Chung, W. and Chen, H. (2004), "A graph model for e-commerce recommender systems", *Journal of the American Society for Information Science and Technology*, Vol. 55 No. 3, pp. 259-74.

Hwang, S-Y., Hsiung, W-C. and Yang, W-S. (2003), "A prototype WWW literature recommendation system for digital libraries", *Online Information Review*, Vol. 27 No. 3, pp. 169-82.

Karypis, G., Aggarwal, R., Kumar, V. and Shekhar, S. (1997), "Multilevel hypergraph partitioning: applications in VLSI domain", *Proceedings of the 34th International*

Conference on Design Automation, pp. 526-9, available at: www.sigda.org/Archives/ProceedingArchives/Dac/ Dac97/ papers/1997/dac97/psfiles/32_4.ps

Korfhage, R.R. (1997), *Information Storage and Retrieval*, John Wiley & Sons, New York, NY.

Krulwich, B. and Burkey, C. (1997), "The infofinder agent: learning user-interests through heuristic phrase extraction", *IEEE Expert*, Vol. 12 No. 5, pp. 22-7.

Kuflik, T., Shapira, B. and Shoval, P. (2003), "Stereotype-based versus personal-based filtering rules in information filtering systems", *Journal of the American Society for Information Science and Technology*, Vol. 54 No. 3, pp. 243-50.

Lang, K. (1995), "Newsweeder: learning to filter Netnews", *Proceedings of the 12th International Conference on Machine Learning*, pp. 331-9.

Liu, B., Hsu, W. and Ma, Y. (1999), "Mining association rules with multiple minimum supports", *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 337-41.

Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A. and Cohen, M.D. (1987), "Intelligent information-sharing systems", *Communications of ACM*, Vol. 30 No. 5, pp. 390-402.

Mobasher, B., Cooley, R. and Srivastava, J. (1999), "Creating adaptive Web sites through usage-based clustering of URLs", *Proceedings of the IEEE Knowledge and Data Engineering Exchange Workshop*, pp. 19-25.

Mobasher, B., Dai, H., Luo, T., Sung, Y. and Zhu, J. (2000), "Integrating Web usage and content mining for more effective personalization", *Proceedings of the International Conference on E-Commerce and Web Technologies*, pp. 165-76.

Mooney, R. and Roy, L. (2000), "Content-based book recommending using learning for text categorization", *Proceedings of the ACM Conference on Digital Libraries*, pp. 195-204.

Myaeng, S.H. and Korfhage, R.R. (1990), "Integration of user profiles: models and experiments in information retrieval", *Information Processing and Management*, Vol. 26 No. 6, pp. 719-38.

NSYSU-ETD (n.d.), "ETD system at National Sun Yat-Sen University", available at: www.lib.nsysu.edu.tw/ethesys/ english/default_e.htm

Pazzani, M. (1999), "A framework for collaborative, content-based and demographic filtering", *Artificial Intelligence Review*, Vol. 13 No. 5/6, pp. 393-408.

Pazzani, M., Muramatsu, J. and Billsus, D. (1996), "Syskill & Webert: identifying interesting Web sites", *Proceedings of the National Conference on Artificial Intelligence*, Vol. 13 No. 5/6, pp. 54-61.

Pennock, D., Horvitz, E., Lawrence, S. and Giles, C. (2000), "Collaborative filtering by personality diagnosis: a hybrid memory- and model-based approach", *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 473-80.

Pitkow, J. and Pirolli, P. (1999), "Mining longest repeating subsequences to predict World Wide Web surfing", *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, pp. 139-50.

Rucker, J. and Polanco, M.J. (1992), "Siteseer: personalized navigation for the Web", *Communications of the ACM*, Vol. 35 No. 12, pp. 73-5.

Sarwar, B.M., Karypis, G., Konstan, J. and Riedl, J. (2000), "Analysis of recommender algorithms for e-commerce", *Proceedings of the 2nd ACM E-Commerce Conference*, pp. 158-67.

Shepherd, M., Duffy, J.F., Watters, C. and Gugle, N. (2001), "The role of user profiles for news filtering", *Journal of the American Society for Information Science and Technology*, Vol. 52 No. 2, pp. 149-60.

Srivastava, J., Cooley, R., Deshpande, M. and Tang, P. (2000), "Web usage mining: discovery and applications of usage patterns from Web data", *SIGKDD Explorations*, Vol. 1 No. 2, pp. 12-23.

Yan, T.W., Jacobsen, M., Garcia-Molina, H. and Dayal, U. (1996), "From user access patterns to dynamic hypertext linking", *Computer Networks*, Vol. 28 No. 7-11, pp. 1007-14.

Yang, Q., Zhang, H.H. and Li, T. (2001), "Mining Web usage logs for prediction models in WWW caching and prefetching", *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 473-8.