Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 437–445
Expert Systems with Applications
www.elsevier.com/locate/eswa

A location-aware recommender system for mobile shopping environments

Wan-Shiou Yang *, Hung-Chi Cheng, Jia-Ben Dia

Department of Information Management, National Changhua University of Education, No. 1, Jin-De Road, Changhua 500, Taiwan, ROC

Abstract

Exploring new applications and services for mobile environments has generated considerable excitement among both commercial companies and academics. In this paper we propose a location-aware recommender system that accommodates a customer's shopping needs with location-dependent vendor offers and promotions. Specifically, we propose a recommender system for recommending vendors' webpages – include offers and promotions – to interested customers. We present the developed system and the constructed approaches in this paper. The results of experiments using both synthetic and empirical data are also reported and discussed.

Keywords: Recommender system; Mobile computing; E-commerce; M-commerce; Information retrieval

1. Introduction

Mobile computing, where users equipped with PDAs, cellular phones, or laptops are free to move while staying connected to service networks, has proved to be a true revolution (Stafford & Gillenson, 2003). Exploring the promises of mobility in designing new applications and services that automatically accommodate customer's shopping needs with location-dependent vendor offers and promotions has generated considerable excitement among both commercial companies (e.g., HP, Nokia, and Samsung) and academics (e.g., Brunato & Battiti, 2003; Sun, 2003; Tewari, Youll, & Maes, 2002; Tsang, Ho, & Liang, 2004).

One of the most popular tools provided in e-commerce to accommodate customer shopping needs with vendor offers is recommender systems (CACM, 1992). Many such systems have emerged over the past decade, whose basic idea is to recommend products according to the preferences of customers. Two broad classes of recommendation approaches that are commonly used are the *content-based* and *collaborative* approaches (CACM, 1992). The content-based approach characterizes recommendable items as a set of content features and represents the profile of a customer's interests using a similar feature set. This approach selects recommendable items that have a high degree of similarity to the customer's profile. In contrast, the collaborative approach recommends items to a customer by taking into account the preferences of other customers (Shardanand & Maes, 1995).

Traditional recommendation techniques, however, are not fully suitable in mobile shopping environment. M-commerce is unique in its location-aware capability (Stafford & Gillenson, 2003). Mobile computing adds a relevant but mostly unexplored piece of information – the customer's physical location – to the recommendation problem. Personal recommendations in m-commerce provide the opportunity and the challenge to take location into account. Therefore, in this research, we propose a location-aware system for recommending vendors' webpages in a mobile shopping environment.

The proposed personal recommender (PR) system is designed to recommend vendors' webpages (including offers and promotions) to interested customers. The PR

* Corresponding author. Tel.: +886 4 7232105x7611; fax: +886 4 7211162.
E-mail address: wsyang@cc.ncue.edu.tw (W.-S. Yang).

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.09.033

system adopted a location-aware architecture so that customers can receive the information of their preferred vendors that are in their neighborhood. The core of the PR system is a recommendation mechanism that analyzes a customer's history and position so that vendor information can be ranked according to the match with the preferences of a customer. Various characteristics of mobile shopping environments are taken into account, and the resulting system can be applied to m-commerce.

This paper is structured as follows. Section 2 reviews related research efforts, and Section 3 describes the overall architecture of the PR system. Section 4 details the design and construction approaches of the PR system, and Section 5 reports our implementation experiences and evaluation results. Section 6 summarizes this work and suggests future directions for research.

2. Related work

Due to the increasing adoption of mobile devices, location-aware systems are becoming more widespread and the demand for them continues to grow. Such systems employ various means to obtain position estimates, such as GPS (Global Positioning System) (outdoor only, with a precision of about 10 m), active badges (Want, Hopper, Falcao, & Gibbons, 1992) (precisions ranging from a few centimeters to room size), or by exploiting the radio propagation properties of the wireless networking medium (Bahl & Padmanabhan, 2000) (with precisions of a few meters for Wi-Fi). These location-aware systems use the estimated position to provide various services. For example, the "Active Badge Location System" proposed by Want et al. (1992) is considered one of the first location-aware applications. This system used infrared technology to determine a user's current location, which was used to forward phone calls to a telephone close to the targeted user. The "Personal Shopping Assistant", proposed at AT&T in 1994 (Asthana, Cravatts, & Krzyzanouski, 1994), provided an indoor wireless system for personalized shopping assistance. The Cyberguide system (Long, Kooper, Abowd, & Atkeson, 1996), developed by the Future Computing Environments group at the Georgia Institute of Technology, used wireless transmissions and GPS to build mobile tour guides that provided information to tourists based on knowledge of their positions and orientation. This system has been extended to support conference attendees by providing them with appropriate information via a PDA as they navigate conference rooms (Dey, Salber, Abowd, & Futakawa, 1999). Several other tour guides (e.g., Davies, Cheverst, Mitchell, & Friday, 1999; Kreller et al., 1998; Cheverst, Davies, Mitchell, & Friday, 2000) have also been proposed. In the present study, we focused on recommending vendor's webpages.

E-commerce has seen the emergence of many types of recommender systems that are designed to provide personal recommendations about various types of products and services, including news and emails (Billsus & Pazzani, 1999; Goldberg, Nichols, Oki, & Terry, 1992; Konstan et al., 1997; Lang, 1995), webpages (Balabanović & Shoham, 1997; Terveen, Hill, Amento, McDonald, & Creter, 1997; Armstrong, Freitag, Joachims, & Mitchell, 1997), books (Mooney & Roy, 2000), music albums (Shardanand & Maes, 1995), and movies (Alspector, Kolcz, & Karunanithi, 1998; Ansari, Essegaier, & Kohli, 2000; Basu, Hirsh, & Cohen, 1998; Pennock, Horvitz, Lawrence, & Giles, 2000; Schafer, Konstan, & Riedl, 2001). The first type of recommendation techniques was called the *content-based approach* (CACM, 1992). The basic idea of content-based approach is to recommend products according to preferences of customers. Content-based approach characterizes recommendable items by a set of content features and represents customers' interest profile by a similar feature set. Then, the relevance of a given content item and the customer's interest profile is measured as the similarity of this recommendable item to the customer's interest profile. Content-based approach selects recommendable items that have a high degree of similarity to the customer's interest profile. Another type of recommendation technique, the *collaborative approach* (or sometimes called the *social-based approach*), takes into account the given customer's interest profile and the profiles of other customers with similar interests (Shardanand & Maes, 1995). The collaborative approach looks for relevance among customers by observing their ratings assigned to products in a relatively small training set. The nearest-neighbor users are those that are most relevant to the target customer. These customers then act as "recommendation partners" for the target customer, with the collaborative approach recommending to the target customer items that appear in the profiles of these recommendation partners but not in those of the target customer. Although these two types of recommender systems have achieved certain success in e-commerce, an important piece of information in m-commerce – the customer's location – has not been exploited in previous recommender systems.

As noted by Sun (2003), the low input and output capabilities of mobile devices make searches for supplier information inconvenient. Therefore, there is a pressing need to make it easier for mobile users to obtain their desired information at the right time and in the right place. Sun (2003) proposed the IRE (Information Requirement Elicitation) framework to help users specify their requirements and filter suppliers by their contextual relevancy to users. Websigns, proposed by HP in 2001 (Pradhan, Brignone, Cui, McReynolds, & Smith, 2001), sends the user's position to a central server that extracts all supplied items whose direction and distance fall within some item-dependent intervals. Brunato and Battiti (2003) assume that a mobile user is likely to interact with a PDA only for the time that is strictly needed to find certain interesting information, and hence proposed the PILGRIM system in which each webpage is associated with a location ellipse derived from previous user's access logs, with webpages recommended when the user is located within the corresponding ellipses.

Our work is influenced by the IRE framework, while we focus on and specify the approaches for recommending vendor's webpages. In the PR system, users receive the information of their preferred vendors that are in their neighborhood without the limitations of item-dependent interval or location ellipse.

3. System architecture

Position-detection systems fall into two categories. In *centralized methods*, such as Active Badge (Harter & Hopper, 1994), Active Bats (Ward, Jones, & Hopper, 1997), and PARCTAB (Want et al., 1996), the infrastructure consists of receivers deployed in some places, with end-users beaconing out data. Client's location is determined and held on servers. In contrast, in *decentralized methods*, such as GPS, Cricket (Priyantha, Chakraborty, & Balakrishnan, 2000), and RADAR (Bahl & Padmanabhan, 2000), the infrastructure consists of beacons deployed in some places, signaling to clients their locations. Hence, client's location is determined and held on a personal device.

There were numerous interviews (Barkhuus & Dey, 2003), reports (Weiser, Gold, & Brown, 1999), and books (Garfinkel, 2001) describing public unease over the potential for abuse of privacy-sensitive location-aware systems. It may be that privacy will be the greatest barrier to the adoption of location-aware services (Brunato & Battiti, 2003). Compared with the centralized approach, the decentralized approach gives end-users greater choice over whether to disclose their location to others. We have therefore adopted the decentralized approach here, whereby customers only send out their location data to the server when they need the recommendation service.

The overall architecture of the PR system is shown in Fig. 1. The client side comprises two components: (i) a standard Internet browser and (ii) the location manager that estimates the client's position. When a customer needs the recommendation service, a request is sent with the customer's position to the server. On the server side, the core is the recommendation engine consisting of off-line and on-line subsystems. The off-line subsystem maintains a database, WEB ACCESS, that logs data about what webpages have been visited by each customer. This subsystem also analyzes the logged data and derives the profile of each customer's interests. The on-line subsystem maintains the customer profiles plus a database, VENDOR DATA, containing vendor's information, such as name, physical address, and webpage link, which is directly registered by vendor. When receiving a service request, the on-line subsystem generates a list of possibly interesting webpages based on the customer's interests profile, vendor data, and the instantaneous position of the customer provided by the location manager.

A screenshot of the client-side subsystem is shown in Fig. 2. After sending a service request, the customer receives a recommendation list containing links to various vendors in the right-hand frame. The positions of the customer and recommended vendors are also marked in a map displayed in the left-hand frame. Once a link to a specific vendor in the right-hand frame is clicked, the corresponding webpage will be shown in the bottom frame.

4. Details of the approach

In this section, we describe the off-line learning performed in the recommendation engine of the PR system,

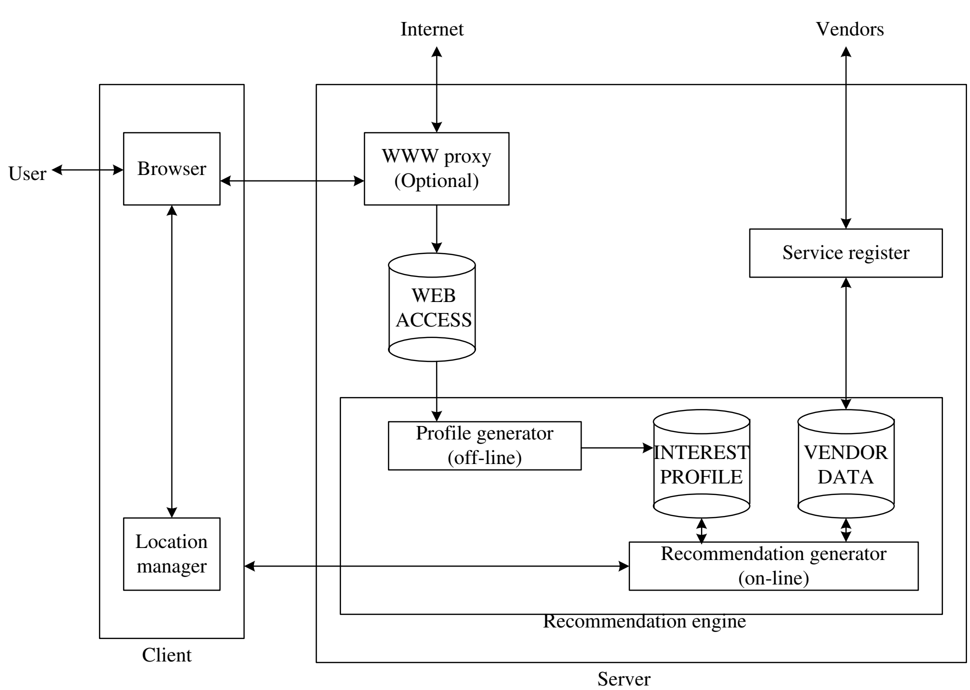

Fig. 1. Architecture of the PR system.

Fig. 2. A screenshot of the client-side PR system.

depict the on-line generation of a recommendation list, and then discuss how to improve the search efficiency.

4.1. Identification of interest profile

Once the WEB ACCESS database is populated with past customer accesses to webpages, its data can be used to estimate a customer's interest profile. The PR system applies a simple information extraction method (Kushmerick, Weld, & Doorenbos, 1997) to the visited webpages. Information extraction involves obtaining useful structured data from unstructured text. In our system it involves parsing the raw pages, removing punctuation and prepositions, and grouping stemming words into generalized terms. The system adopts the bag-of-words model and creates a vector of terms for each webpage, in which each cell indicates the frequency with which each term occurs in the webpage.

The profile of interest of each customer, which is learned from the webpages that the customer has visited, is also represented as a vector of terms that are created by the system by summarizing the webpages that he or she has visited. Therefore, if a customer c has visited a set S of webpages, the interest profile of customer c is estimated using

$$CP(c) = \sum_{w \in S} \mathrm{Vector}(w), \tag{1}$$

where $\mathrm{Vector}(w)$ denotes the term vector of webpage w.

4.2. Generation of recommendation list

Based on the generated profile of a customer's interest, the PR system estimates the customer's interest in the webpage of a particular vendor based on the similarity of the webpage to the customer's interest profile according to

$$\mathrm{Similarity}(c, w) = \mathrm{Cosin}(CP(c), \mathrm{Vector}(w)) \tag{2}$$

The physical distance between the customer and vendor is also considered by the PR system. Each webpage registered by a vendor is assigned a physical location. We posit that the likelihood of a vendor being visited by a customer falls exponentially with the distance between them:

$$\mathrm{DistanceDecay}(c, w) = \frac{1}{e^{\lambda \times \mathrm{Distance}(c,w)}} \tag{3}$$

where $\lambda \in [0, \infty]$ is a parameter representing the customer's sensitivity to location, and $\mathrm{Distance}(c, w)$ denotes the Euclidean distance between customer c and webpage w. Therefore, the interest of customer c in vendor's webpage w at the customer's present position is

$$\mathrm{Interest}(c, w) = \mathrm{Similarity}(c, w) * \mathrm{DistanceDecay}(c, w) \tag{4}$$

The PR system generates a recommendation list containing the top-N webpages for the requesting customer.

Parameter λ ranges from 0 for a customer who does not care about a vendor's location to ∞ for a customer who is infinitely sensitive to a vendor's location. Clearly, if λ is small, the exponential distance function decreases slowly, and the customer is more likely to choose a vendor in a distant location; if λ is large, the exponential distance function decreases quickly, and the customer is less likely to choose a vendor in a distant location. A procedure for determining the most suitable λ for each customer was also designed in this study.

The procedure for determining λ starts with a randomly chosen λ_0, and iteratively adjusts it to find the desired value using least-squares strategy (Mehrotra, Mohan, & Ranka, 1996). For an estimated setting of the parameter, the squared error between the estimated output and the desired output d of the exponential decay function is $E = (\mathrm{e}^{-\lambda\times\mathrm{Distance}(c,w)} - d)^2$, so that

$$\frac{\partial E}{\partial \lambda} = -2(\mathrm{e}^{-\lambda\times\mathrm{Distance}(c,w)} - d) \times \mathrm{e}^{-\lambda\times\mathrm{Distance}(c,w)} \times \mathrm{Distance}(c,w) \tag{5}$$

The magnitude of the change is hence computed as a negative multiple of $\frac{\partial E}{\partial \lambda}$:

$$\Delta\lambda = (\mathrm{e}^{-\lambda\times\mathrm{Distance}(c,w)} - d) \times \mathrm{e}^{-\lambda\times\mathrm{Distance}(c,w)} \times \mathrm{Distance}(c,w) \tag{6}$$

Parameter λ is learned by observing the customer's selection behavior. As shown in Fig. 2, when a recommended list is presented to a customer, the positions of recommended vendors are also marked in a map. The selection by the customer of a recommended page suggests that the corresponding vendor is within an acceptable distance from the customer. Therefore, once customer c requests the service and a set S of recommended webpages is generated, suppose webpage $w_f \in S$ is the farthest webpage that customer c selected. We can define a disjoint partitioning of S as $S_S \cup S_{NS}$, where S_S is the set of the webpages closer than w_f to customer c and S_{NS} is the set of remaining pages. Also, suppose w_n is the nearest webpage to customer c in S_{NS}. Since it is believed that webpage w_f is within the desired distance of customer c, we make a negative change to λ by setting the desired output of the exponential decay function to its maximal value 1:

$$\lambda = \lambda + (\mathrm{e}^{-\lambda\times\mathrm{Distance}(c,w)} - 1) \times \mathrm{e}^{-\lambda\times\mathrm{Distance}(c,w)} \times \mathrm{Distance}(c,w) \tag{7}$$

For page w_n, we make a positive change to λ by setting the desired output of the exponential decay function to its minimal value 0:

$$\lambda = \lambda + (\mathrm{e}^{-\lambda\times\mathrm{Distance}(c,w)} - 0) \times \mathrm{e}^{-\lambda\times\mathrm{Distance}(c,w)} \times \mathrm{Distance}(c,w) \tag{8}$$

The λ learning procedure repeatedly makes the above changes to the parameter value in order to determine the suitable setting.

Table 1 lists an example containing ten webpages listing the similarities and distances to a customer c. Suppose λ is initially set to 0 and the system generates a top-five recommendation list. Initially, five pages $\{w_2, w_3, w_4, w_7, w_8\}$ will be recommended to customer c since they have higher similarities to the profile of the customer's interest. If customer c only selects pages w_2 and w_4 from this recommendation list, the λ learning procedure partitions the five pages as two sets: $\{w_2, w_3, w_4\}$ and $\{w_7, w_8\}$. Because w_4 is the farthest page in page set $\{w_2, w_3, w_4\}$, λ is recomputed as 0 (i.e., $\lambda = 0 + (\mathrm{e}^{-0\times 0.5} - 1) * \mathrm{e}^{-0\times 0.5} * 0.5 = 0$). Because w_8 is the nearest page in the page set $\{w_7, w_8\}$, λ is recomputed as 0.8 (i.e.,$\lambda = 0 + (\mathrm{e}^{-0\times 0.8} - 0) * \mathrm{e}^{-0\times 0.8} * 0.8 = 0.8$). In the next iteration, λ will be set to 0.8. If the customer requires a recommendation service at the same position, five webpages $\{w_1, w_2, w_3, w_4, w_7\}$ will be recommended.

```
λ-learning(λ: real; S: a set of webpages): real
{
    Partition S into two sets S_S and S_NS;
    Let w be the farthest page in S_S;
    λ = λ + (e^(-λ×Distance(c,w)) - 1) × e^(-λ×Distance(c,w)) × Distance(c,w);
    Let w be the shortest page in S_NS;
    λ = λ + (e^(-λ×Distance(c,w)) - 0) × e^(-λ×Distance(c,w)) × Distance(c,w);
    Return λ;
}
```

Fig. 3. The pseudocode of the λ learning procedure.

Fig. 3 details the pseudocode of the λ learning procedure that is implemented and embedded in the client's location-manager component. Each time the customer requests the recommendation service, the current λ is send with the client's position to the server that generates a recommendation list. After receiving the generated webpages, λ will then be recomputed according to the customer's selection behavior and held on the client's device for the next recommendation service.

4.3. Improvement of search efficiency

In location-aware applications, customers often expect to receive services in real time. Therefore, in order to avoid the time-consuming task of scanning the entire webpage database for each service request to build a recommendation list, we also implement a locality-based data structure. Specifically, a data structure, R^+-trees, is implemented in the PR system. R^+-trees, which was originally proposed by Sellis, Roussopoulos, and Faloutsos (1987), is a height-balanced tree that consists of intermediate and leaf nodes. Data objects (each of which is contained in a split rectangular space) are stored in leaf nodes, and

Table 1
An example of ten vendor's pages

Pages	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
Similarity(c, w_i)	0.55	0.8	0.6	1.0	0.2	0.1	0.9	0.65	0.5	0.4
Distance(c, w_i)	0.5	0.2	0.3	0.5	0.7	1.0	0.9	0.8	0.8	1.2

intermediate nodes are built by grouping rectangles at the lower lever. Specifically, in R^+-trees, a leaf node is of the form (oid, RECT), where oid is an object identifier and is used to refer to an object in the database, and RECT describes the bounds of data objects. For example, in two-dimensional space, an entry RECT will be of the form (x_{low}, x_{high}, y_{low}, y_{high}), which represents the coordinates of the lower-left and upper-right corner of the rectangle. An intermediate node is of the form (p, RECT), where p is a pointer to a lower level node of the tree and RECT is a representation of the rectangle that encloses it.

Suppose the webpages listed in Table 1 are distributed over a two-dimensional space as shown in Fig. 4(a). A corresponding R^+-trees then can be constructed as shown in Fig. 4(b), where ten leaf nodes (N_6, N_7, N_8, N_9, N_{10}, N_{11}, N_{12}, N_{13}, N_{14}, and N_{15}) are built. Also, six intermediate nodes (N_0, N_1, N_2, N_3, N_4, and N_5) are constructed by grouping nodes at the lower level. The pointer to a lower level node and the enclosed rectangle are also stored in each intermediate node.

Such an R^+-trees, once constructed, can be used to improve the efficiency of searching webpages. A well-known tree search strategy, the best-first search (Lee, Chang, Tseng, & Tsai, 2001), is used in the PR system. The search strategy starts from the root node of the constructed R^+-trees. During the process of tree searching, the exponential decay function (i.e., Eq. (3)) is used as an evaluation function and the intermediate node with the

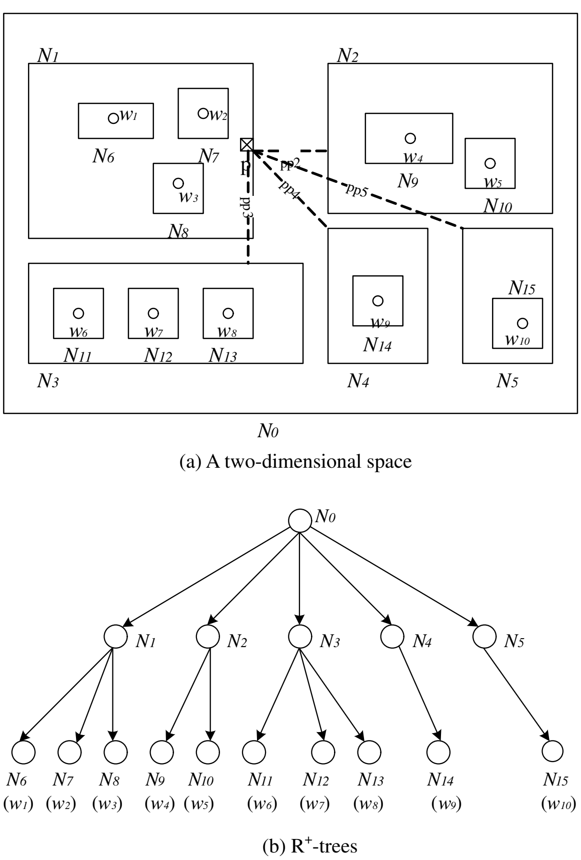

Fig. 4. An example of R^+-trees.

largest value among all intermediate nodes that have been examined so far is selected for further extension. Please note that, in order to compute the value of the exponential decay function, the distance between a customer and an intermediate node must be determined. In this study, if the customer falls in the rectangle range of the intermediate node, the distance is regarded as 0; otherwise the distance is computed as the shortest distance between the customer and the corresponding rectangle of the node. Take Fig. 4 as an example again. Suppose there is a customer requiring the recommendation service at position p. The shortest distances between customer c and intermediate nodes N_1, N_2, N_3, N_4, and N_5 are 0, pp_2, pp_3, pp_4, and pp_5, respectively. Therefore, among the second-level intermediate nodes, N_1 will be first examined.

Also, the interest of a customer in webpages that have been examined serves as a lower bound to the termination of branches. From Eq. (4), it is clear that the interest on a given webpage is equal to or smaller than the value of the exponential decay function. Hence, any intermediate node whose value of the exponential decay function is smaller than the minimum interest of currently examined top-N webpages cannot be a recommended item. The intermediate node and its descendants can thus be ignored. Considering the previous example again, suppose that λ is 1 and pp_2, pp_3, pp_4, and pp_5 are 0.3, 0.75, 0.7, and 1.1, respectively. If the system generates a top-three recommendation list and the interests on pages w_1, w_2, and w_3 have been evaluated as 0.33, 0.65, and 0.44, then node N_5 and its descendents can be ignored since they will never lead to any webpage for which the interest will be greater than w_1. The complete webpage-searching algorithm is shown in Fig. 5.

5. Evaluation

The different building blocks of the PR system shown in Fig. 1 were implemented and integrated with GPS technology in this study. The separate components, written in Java, interact with standard HTML to generate the recommendation list for customers as shown in Fig. 2. In this section, we present the results of applying our proposed approach to both synthetic and empirical data. The main purpose of using synthetic data is to evaluate the efficiency and scalability of our proposed approach. We then applied our proposed approach to empirical data and report here on the effectiveness.

5.1. Synthetic data

The first experiment concentrated on the on-line workload of the PR system, and tested it in a simulated environment. K sites, each associated with a webpage, were randomly placed in a 10×10 km square. We considered a service station in which customers, each with a random characteristic value and a vector of terms, arrive in accordance with a Poisson process with a rate β. Upon arrival, a customer either enters the service if the recommendation

```
WebpageSearching(Num: a real; c: a customer): a set of webpages
{
// Suppose Num is the number of webpages that would be selected and c represents
data about the customer, including the interests profile and position;

    SelectedPages = ∅;
    Bound = 0;
    Let H be a heap and first add the root node of the R+-trees into H;
    While (H ≠Null)
    {
        Let N = the root element in H;
        IF (DistanceDecay(c, N) ≥Bound) then    // Lower bound condition
        {
            If (N is a leaf node in the R+-trees) then
            {
                Let I = the interest on the page corresponding to N to customer c;
                If (I > Bound ) then
                {
                    SelectedPages = SelectedPages ∪N;
                    If (|SelectedPages| > Num) then
                    {
                    SelectedPages = SelectedPages – page with the smallest interest;
                    Bound = smallest interest on pages in SelectedPages;
                    }
                } // End of IF (I > Bound)
                H = H – N;
            } // End of IF (N is a leaf node in the R+-trees)
            ELSE
            {
                 H = H – N;
                 Add the children of N into H;
            }
        } // End of IF (DistanceDecay(c, N) ≥Bound)
    } // End of WHILE loop
    Return SelectedPages;
}
```

Fig. 5. Webpage-searching algorithm.

engine is free at that moment or else joins the queue. After the engine has generated a recommendation list for a customer, it then either serves the longest-waiting customer or, if there are no waiting customers, waits for the arrival of the next customer. In addition, there is a fixed time (1 h) after which no additional arrivals are allowed to enter

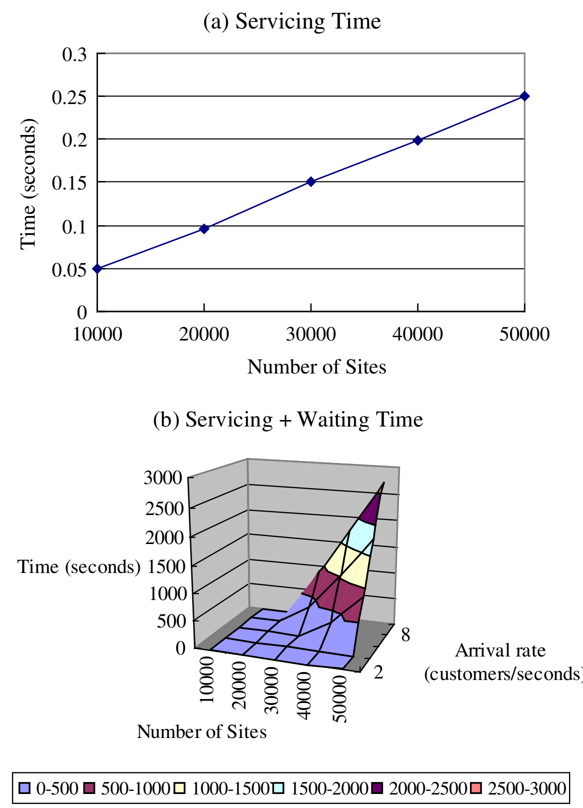

Fig. 6. The execution time of the experimental PR system.

the system, although the server completes servicing all those that are already in the system.

We first investigated the effects of varying the number of websites from 10,000 to 50,000 (in increments of 10,000) on the average request servicing time. Ten experiments were executed, and the average result is shown in Fig. 6(a). In general, the average servicing time grows linearly with the number of sites. This is as expected since the PR system must scan the webpage database for each service request to build a recommendation list. Also, we investigated the joint effects of the number of the websites and the request arrival rate β (ranging from 2 to 10 in increments of 2), on the average time a customer spends in the system (i.e., including servicing time and waiting time). Ten experiments were executed, and the average result is shown in Fig. 6(b). The lower-left area of Fig. 6(b) shows that the average time a customer spends in the system grows slowly. However, the upper-right area indicates that when the servicing time exceeds the time period between two requests (i.e., the inverse of the arrival rate β), the average time a customer spends in the system grows dramatically. This is because the accumulation of a large number of requests in the queue results in a huge increase in the waiting time.

5.2. Empirical data

The second experiment focused on the effectiveness of the PR system in a real-world environment. Between

January 2004 and August 2005, we collected data from 136 graduate and undergraduate students. The subjects were shown a brief, 2-min demonstration of the system; and then were provided with laptops or PDAs implementing the system which they used for 3 months. During the 1st month, only the browser component was activated to log what webpages had been visited by the subject. During the 2nd and 3rd months, all components were activated to provide the recommendation service. The proposed approach was used in the 2nd month to learning parameter λ, and during the last month one of the following approaches was chosen at random when a service request was received from a customer:

1. *Content-distance-based* approach, which is the approach described in Sections 3 and 4.
2. *Content-based* approach, which differs from the content-distance-based approach by ignoring the factor of distance. In this approach, the interest on a given webpage is only measured as the similarity of this webpage to the profile of the customer's interests.
3. *Distance-based* approach, which differs from content-distance-based approach by ignoring the factor of webpage content. In this approach, the interest on a given webpage is only measured by the factor of distance.
4. *Random* approach, which randomly chooses webpages.

During the last month, each subject evaluated all the different approaches. Therefore, for each recommendation service, we applied an approach at random to generate a top-five recommendation list. A total of 8266 recommendation lists were generated for all the subjects: 2044, 2085, 2036, and 2101 by the content-distance-based, content-based, distance-based, and random approaches, respectively. All the subjects were required to report their satisfaction level on a 7-point scale when they received each recommendation. The effectiveness of the various approaches is shown in Fig. 7, which plots the average performance.

Fig. 7 indicates that the satisfaction level is higher for the distance-based approach than for the random approach.

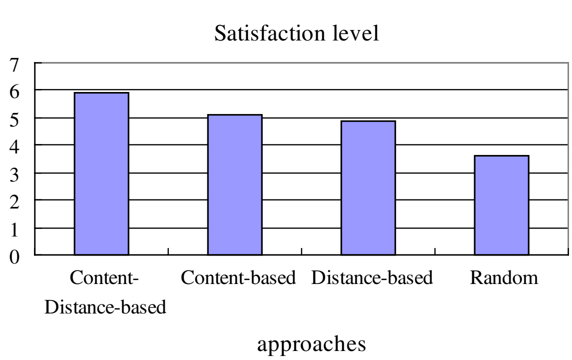

Fig. 7. The average reported satisfaction levels for the various approaches.

The satisfaction level is slightly better for a content-based approach than for the distance-based approach, which demonstrates the usefulness of webpage content. Finally, the proposed content-distance-based approach has the best performance, which we attribute to the usefulness of combining webpage content and the factor of distance. The performance of the content-distance-based approach was significantly better than that of the content-based approach ($p < 0.05$).

6. Conclusion

This study investigated location-aware personal recommendations by constructing a system that amalgamates the information abundance of the Internet with the tangible richness of physical shopping, in terms of location. We have discussed how the proposed approach is implemented to exploit the functionality afforded by a powerful location-aware architecture, and have evaluated its performance using both synthetic and empirical data. Our experimental results demonstrate that the proposed approach outperforms the content- and distance-only approaches.

The work described here can be extended in several directions. Firstly, more extensive real-world testing is desirable. Secondly, the PR system currently adopts a single-server architecture. For the improvement of efficiency and stability, multiserver architecture should be investigated. Finally, although the decentralized approach we adopted is considered to be more privacy-preserving, some subjects in the experiments still show their privacy concerns. Therefore, the integration of other privacy-preserving schemes and techniques into our architecture to improve the privacy level of our system is another interesting direction for future research.

References

Alspector, J., Kolcz, A., & Karunanithi, N. (1998). Comparing feature-based and clique-based user models for movie selection. In *Proceedings of the third acm international conference on digital libraries*.

Ansari, A., Essegaier, S., & Kohli, R. (2000). Internet recommendation systems. *Journal of Marketing Research, 37*(3).

Armstrong, R., Freitag, D., Joachims, T., & Mitchell, T. (1997). WebWatcher: a tour guide for the world wide web. In *Proceedings of the 15th international joint conference on artificial intelligence*.

Asthana, A., Cravatts, M., & Krzyzanouski, P. (1994). An indoor wireless system for personalized shopping assistance. In *Proceedings of the workshop on mobile computing systems and applications*.

Bahl, P., & Padmanabhan, V. N. (2000). RADAR: an in-building RF-based user location and tracking system. In *Proceedings of IEEE INFOCOM*.

Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM, 40*(3).

Barkhuus, L., & Dey, A. K. (2003). Location-based services for mobile telephony: a study of user's privacy concerns. In *Proceedings of the ninth IFIP TC13 international conference on human–computer interaction*.

Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation as classification: using social and content-based information in recommendation. In *Proceedings of AAAI symposium on machine learning in information access*.

Billsus, D., & Pazzani, M. J., 1999. A hybrid user model for news story classification. In *Proceedings of the seventh international conference on user modeling*.

Brunato, M., & Battiti, R. (2003). PILGRIM: a location broker and mobility-aware recommendation system. In *Proceedings of first IEEE annual conference on pervasive computing and communications*.

CACM (1992). Special issue on information filtering. *Communications of the ACM, 35*(12).

Cheverst, K., Davies, N., Mitchell, K., & Friday, A. (2000). Experiences of developing and deploying a context-aware tourist guide: the guide project. In *Proceedings of the sixth annual international conference on mobile computing and networking*.

Davies, N., Cheverst, K., Mitchell, K., & Friday, A. (1999). Caches in the air: disseminating information in the guide system. In *Proceedings of the second IEEE workshop on mobile computing systems and applications*.

Dey, A., Salber, D., Abowd, G., & Futakawa, M. (1999). The conference assistant: combining context-awareness with wearable computing. In *Proceedings of the third IEEE international symposium on wearable computers*.

Garfinkel, S. (2001). *Database nation: the death of privacy in the 21st century*. O'Reilly & Associates.

Goldberg, D., Nichols, D., Oki, B., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM, 35*(12).

Harter, A., & Hopper, A. (1994). A distributed location system for the active office. *IEEE Network, 8*(1).

Konstan, J., Miller, B., Maltz, D., Herlocker, J. L., Gordon, L., & Riedl, J. (1997). Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM, 40*(3).

Kreller, B., Carrega, D., Shankar, J., Salmon, P., Bottger, S., & Kassing, T. (1998). A mobile aware city guide application. In *Proceedings of the ACTS mobile communications summit*.

Kushmerick, N., Weld, D., & Doorenbos, B. (1997). Wrapper induction for information extraction. In *Proceedings of the 15th international joint conference on artificial intelligence*.

Lang, K. (1995). Newsweeder: learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*.

Lee, R. C. T., Chang, R. C., Tseng, S. S., & Tsai, Y. T. (2001). *Introduction to the design and analysis of algorithms*. Unalis Press.

Long, S., Kooper, R., Abowd, G.D., & Atkeson, C.G. (1996). Rapid prototyping of mobile context-aware applications: the cyberguide case study. In *Proceedings of the second ACM international conference on mobile computing and networking*.

Mehrotra, K., Mohan, C. K., & Ranka, S. (1996). *Elements of artificial neural networks*. The MIT Press.

Mooney, R., & Roy, L., (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM international conference on digital libraries*.

Pennock, D., Horvitz, E., Lawrence, S., & Giles, C. (2000). Collaborative filtering by personality diagnosis: a hybrid memory- and model-based approach. In *Proceedings of the 16th conference on uncertainty in artificial intelligence*.

Pradhan, S., Brignone, C., Cui, J.-H., McReynolds, A., & Smith, M. T. (2001). Websigns: hyperlinking physical locations to the web. *IEEE Computer, 34*(8).

Priyantha, N. B., Chakraborty, A., & Balakrishnan, H. (2000). The cricket location-support system. In *Proceedings of the sixth ACM international conference on mobile computing and networking*.

Schafer, J., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery, 5*(1).

Sellis, T. K., Roussopoulos, R., & Faloutsos, C. (1987). The R^+-tree: a dynamic index for multi-dimensional objects. In *Proceedings of the international conference on very large data bases*.

Shardanand, U., & Maes, P., 1995. Social information filtering: algorithms for automating 'word of mouth'. In *Proceedings of the SIGCHI international conference on human factors in computing systems*.

Stafford, T. F., & Gillenson, M. L. (2003). Mobile commerce: what it is and what it could be. *Communications of the ACM, 46*(12).

Sun, J. (2003). Information requirement elicitation in mobile commerce. *Communications of the ACM, 46*(12).

Terveen, L., Hill, W., Amento, B., McDonald, D., & Creter, J. (1997). PHOAKS: a system for sharing recommendations. *Communications of the ACM, 40*(3).

Tewari, G., Youll, J., & Maes, P. (2002). Personalized location-based brokering using an agent-based intermediary architecture. *Decision Support Systems, 34*(12).

Tsang, M. M., Ho, S.-C., & Liang, T.-P. (2004). Consumer attitudes toward mobile advertising: an empirical study. *International Journal of Electronic Commerce, 8*(3).

Want, R., Hopper, A., Falcao, V., & Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems, 10*(1).

Want, R., Schilit, B. N., Adams, N. I., Gold, R., Petersen, K., Goldberg, D., et al. (1996). The PARCTAB ubiquitous computing experiment. In H. F. Korth & T. Imielinski (Eds.), *Mobile computing*. Academic Press: Kluwer.

Ward, A., Jones, A., & Hopper, A. (1997). A new location technique for the active office. *IEEE Personnel Communications, 4*(5).

Weiser, M., Gold, R., & Brown, J. S. (1999). The origins of ubiquitous computing research at PARC in the late 1980. *IBM Systems Journal, 38*(4).