Information Sciences 181 (2011) 1552–1572

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Personalized recommendation of popular blog articles for mobile applications

Duen-Ren Liu *, Pei-Yun Tsai, Po-Huan Chiu

Institute of Information Management, National Chiao Tung University, Hsinchu, Taiwan

ARTICLE INFO

Article history:
Received 5 October 2009
Received in revised form 22 October 2010
Accepted 1 January 2011
Available online 9 January 2011

Keywords:
Mobile service
Blog recommenders
Time-sensitive topic
Collaborative filtering

ABSTRACT

Weblogs have emerged as a new communication and publication medium on the Internet for diffusing the latest useful information. Providing value-added mobile services, such as blog articles, is increasingly important to attract mobile users to mobile commerce, in order to benefit from the proliferation and convenience of using mobile devices to receive information any time and anywhere. However, there are a tremendous number of blog articles, and mobile users generally have difficulty in browsing weblogs owing to the limitations of mobile devices. Accordingly, providing mobile users with blog articles that suit their particular interests is an important issue. Very little research, however, has focused on this issue.

In this work, we propose a novel Customized Content Service on a mobile device (m-CCS) to filter and push blog articles to mobile users. The m-CCS includes a novel forecasting approach to predict the latest popular blog topics based on the trend of time-sensitive popularity of weblogs. Mobile users may, however, have different interests regarding the latest popular blog topics. Thus, the m-CCS further analyzes the mobile users' browsing logs to determine their interests, which are then combined with the latest popular blog topics to derive their preferred blog topics and articles. A novel hybrid approach is proposed to recommend blog articles by integrating personalized popularity of topic clusters, item-based collaborative filtering (CF) and attention degree (click times) of blog articles. The experiment result demonstrates that the m-CCS system can effectively recommend mobile users' desired blog articles with respect to both popularity and personal interests.

1. Introduction

Weblogs have emerged as a new communication and publication medium on the Internet for diffusing the latest useful information. Blog articles represent the opinions of the populace and constitute a reaction to current events (e.g., news) on the Internet [13]. Accordingly, looking for the latest popular issues discussed by blogs and attracting readers' attention is an interesting subject. Moreover, providing value-added mobile services, such as blog articles, is increasingly important to attract mobile users to mobile commerce, in order to benefit from the proliferation and convenience of using mobile devices to receive information anytime and anywhere. There are, however, a tremendous number of blog articles, and mobile users generally have difficulty in browsing weblogs owing to the inherent limitations of mobile devices, such as small screens, short usage time and poor input mechanisms. Accordingly, providing mobile users with blog articles that suit their interests is an important issue. Very little research, however, has focused on this issue.

* Corresponding author. Tel.: +886 3 5131245; fax: +886 3 5723792.
E-mail addresses: dliu@mail.nctu.edu.tw, dliu@iim.nctu.edu.tw (D.-R. Liu).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2011.01.005

There are three main types of research regarding blogs. The first type of research focuses on analyzing the link structure between blogs to form a community [19,20]. Through the hyperlinks between blogs, people can communicate across blogs by publishing content related to other blogs. Nakajima et al. [31] proposed a method to identify the important bloggers in the conversations, based on their roles in preceding blog threads, and identify "hot" conversation. The second type of research focuses on content analysis to derive the propagation of topics and trends in the blogsphere. Gruhl et al. [11,12] modeled the information propagation of topics among blogs based on blog text. With the analysis of tracking topic and user drift, Hayes et al. [13] examined the relationship between blogs over time. Mei et al. [28] proposed a method to discover the distributions and evolution patterns across time and space. Although existing studies have investigated the evolution of blog topics, they have not considered how to predict the degree of popularity of blog topics. The last type of research focuses on how to model the bloggers and derive their interests in order to generate personal recommendations [38,40]. A variety of methods has been proposed to model the blogger's interests and provide recommended content which is similar to their earlier experiences [15,24].

The majority of previous studies on blogs have ignored the hot topics and popular articles discussed by mass groups of readers, who engage in browsing actions related to the blog articles. Moreover, existing studies do not consider recommending blog articles to mobile readers in mobile environments. With more and more blog articles continually being published on the Internet, the scale and complexity of blog contents are growing rapidly, resulting in information overload for blog readers. Mobile readers could only browse a very limited number of blog articles because of the restrictions of mobile devices. Accordingly, traditional recommendation methods, such as the collaborative filtering approach [1,2,5,17,25,35], may suffer the sparsity problem of finding similar users or items due to insufficient historical records of browsing blog articles by mobile readers. To address the sparsity issue and blog information overload, it is essential to design an appropriate mechanism for recommending blog articles in mobile environments. Blog readers are often interested in browsing emerging and popular blog topics, from which the popularity of blogs can be inferred according to the accumulated click times on blogs. Popularity based solely on click times, however, cannot truly reflect popularity trends. For example, a new event may trigger emerging discussions such that the number of related blog articles and browsing actions is small at the beginning and rapidly increases as time goes on. Thus, it is important to analyze the trend of time-sensitive popularity of blogs to predict emerging hot blog topics. In addition, blog readers may have different interests regarding the emerging popular blog topics. Nevertheless, existing researches have not addressed such issues of how to predict the popularity trend of blog topics and personalized popular topics.

More specifically, several studies have been proposed to model the blogger's interest and provide personal recommendations [15,24,38,40]. Traditional approaches of recommender systems can also be adopted to recommend blog articles to mobile users. However, existing researches have not addressed the issue of recommending personalized popular blog articles, which is especially important for mobile environments where mobile users can not freely browse a tremendous amount of blog articles on the Internet due to the restriction of mobile devices, and therefore must rely on service providers' recommendations to browse a small and feasible subset of blog articles. Many blog articles are new articles to the system, since they have not been viewed by any mobile user in the system due to the limitation of mobile devices. Traditional recommendation methods may suffer from the new item problem, in which there is no rating record on new items by which to derive the prediction [1]. It means that most new articles, which are popular on the Internet and to which the masses of Internet users pay attention, may be ignored by conventional recommendation methods. Accordingly, the recommended feasible set of blog articles should contain those articles which are new articles to the system but are popular with Internet users and also suit mobile users' personal interests. Existing recommendation approaches have neither addressed such issues nor considered the popularity degree of blog articles.

In this work, we propose a novel Customized Content Service on a mobile device (m-CCS) to recommend personalized and popular blog articles to mobile users. Conventional recommender systems mainly employ the users' behavior logs recorded in the systems to make recommendations. Differing from existing recommender systems, we use an additional data source collected from the Internet, i.e., the Internet users' click times on blog articles, to identify the popularity degree of blog articles which are integrated with recommendation approach to improve the recommender quality in mobile recommender services.

First, we propose a novel approach to predict the trend of time-sensitive popularity of blog topics. We analyze blog contents retrieved by co-RSS to derive topic clusters, i.e., blog topics. We define a topic as a set of significant terms that are clustered together based on aspects of similarity. By examining the clusters, we can extract the salient features of topics. Moreover, we analyze the click times of Internet readers accessing articles. For each topic cluster, we modified a double exponential smoothing method [6,7] to predict the popularity degree of the topic according to the variation in trends of click times by Internet readers. Second, mobile users may have different interests regarding the latest popular blog topics. Thus, we further propose a novel approach to infer mobile users' preferred (personalized) popular blog topics based on the predicted popularity degree of blog topics and mobile users' personal interests, derived by analyzing their browsing logs. Third, a novel hybrid recommendation approach is proposed to recommend blog articles by integrating personalized popularity of topic clusters, item-based collaborative filtering (CF) and attention degree (click times) of blog articles. The major novel ideas are as follows. The hybrid prediction is derived according to the clarity of personal preference derived from collaborative filtering, based on the historical behavior of the mobile user. With clear preference, i.e. more browsing records of mobile users, the hybrid prediction will be influenced more by user preference prediction based on collaborative filtering. The hybrid prediction is, however, dominated by Internet attention degree of articles for the mobile users who have very few browsing records with which to infer their preferences. Moreover, hybrid prediction considers the predictive personalized popularity degree of the topic cluster to which each article belongs; the more popular the topic of an article is, the more numerous the users who are interested in the article.

The filtered articles are sent to the individual's mobile device via a WAP Push service. This allows the user to receive personalized and relevant articles, satisfying the demand for instant information. Finally, we conduct on-line experiments to compare different strategies: unified push of articles selected by experts and personalized push of articles selected by the m-CCS system's novel recommendation service. The experiment result shows that our proposed approach considering customized predictive popularity degree can increase the click rates of blog articles to enhance the quality of recommendation. The proposed m-CCS system can effectively recommend desirable blog articles to mobile users based on popularity and personal interests.

The remainder of this paper is organized as follows. Section 2 introduces works related to blogs, forecasting and recommendations; a brief introduction to our system is given in Section 3; detailed descriptions of the processing module of our system are presented in Sections 4 and 5; Section 6 illustrates how to integrate different modules of our system to develop recommendation methods; the system architecture is illustrated in Section 7; Section 8 presents the evaluation of the usefulness of m-CCS empirically and practically; and the conclusions and suggestions for future work are presented in Section 9.

2. Literature review

2.1. Discovering the trend of blog topics

Blog content represents the opinions of the populace and reactions to current events (e.g., news) on the Internet [13]. With Web 2.0, blogs have become such a powerful force that mainstream media cannot help but take notice [9]. Several researches focus on analyzing blog content to derive the propagation of topics and trends in the blogsphere. Gruhl et al. [11,12] modeled the information propagation of blog topics, based on blog texts. The patterns they proposed for topic propagation were useful for predicting sales forecasts. In addition, more and more researches have recently been paying attention to studies on blog content. Blog text analysis focuses on eliciting useful information from blog entry collections, and determining certain trends in the blogosphere. A Natural Language Processing (NLP) algorithm has been used to determine the most important keywords within a definite time period; it can automatically discover trends across blogs [9]. Nevertheless, the above mentioned researches emphasize assigning blog articles to only one topic, while blogs, in fact, contain many topics. Mei et al. [28] focused on a mixture of subtopics and recognize the spatiotemporal topic patterns within blog documents. They proposed a probabilistic method to model the most salient topics from a text collection, and discover the distributions and evolution patterns across time and space. To track topic and user drift, Hayes et al. [13] examined the relationship between blogs over time. Some studies have investigated the evolution of blog topics. However, most researches have not considered how to predict the popularity degree of blog topics. In addition, researches mainly analyze the content of blog articles to discover the evolution and trend of blog topics without considering the Internet readers' perspective, i.e., the click times of Internet readers on blog articles. Differing from other studies, we identify blog topics by clustering similar blog articles into clusters (topics), and then use the accumulated Internet readers' click times of blog articles for generating topic clusters by which to predict the popularity degree of blog topics.

2.2. Recommending blog articles

Several studies investigated user modeling and personal recommendation in blog space. A variety of methods [38,40] has been proposed to model bloggers' interest, such as classifying articles into predefined categories to identify the author's preference [24], and thereby automatically recommend the blog articles which suit their interest, by analyzing the contents to which bloggers have reacted. Huang et al. [15] proposed an approach to extract terms relevant to users from blog articles, and then recommend blog articles explored by Google's search engine. While bloggers can receive recommended content which is similar to that their earlier experiences, the method ignores the hot topics and popular articles discussed by the bulk of readers which can attract mobile users' interest. These studies mainly examined the interests of bloggers and identified which topics were widely discussed by the bloggers without considering the perspectives of Internet readers. They did not address the issue of how to predict the popularity trend of blog topics. Moreover, existing approaches on recommending blog articles did not investigate the recommendation of popular blog articles by considering the popularity degree of blog topics. Differing from existing studies, we recommend personalized and popular blog articles by considering Internet readers' click times on blog articles and the predictive popularity degree of blog topics.

2.3. Forecasting

Forecasting methods mainly use historical data to infer future development trends. Time series prediction uses a set of observation values by time order to construct a suitable model to forecast future trends. Within the variety of methods, the exponential smoothing method [6] is easy to understand and highly reliable; this method can also use less data to make short term predictions. The exponential smoothing method assumes stability and regularity in the trend of time series.

A standard exponential smoothing method [30] assigns exponentially decreasing weights to previous observations. In other words, recent observations are given relatively more weight in forecasting than are the older observations. The exponential smoothing method has been widely used in short term or medium term economic development trend forecasting. In the simple exponential smoothing method, the current prediction value is derived from the prediction value and actual value of the preceding time period. Simple exponential smoothing is suitable for stationary time series which do not exhibit trend effect.

The double exponential smoothing approach is usually used to process the time series data with trend effect, and is predicted using Eq. (1) [7]. For preceding time series, $x(t)$ is the actual value at time t, and $\hat{x}(t)$ is the prediction value at time t; and $b(t)$ represents the trend effect at time t. To forecast the current value for time $t+1, \hat{x}(t+1)$ is the average value between two parameters, $x(t)$ and $[\hat{x}(t)+b(t)]$, weighted by α which is a smoothing constant. Therefore, the difference of smoothing constant would determine which parameter has greater influence in affecting the prediction value. Learning from the formula, each prediction value is weighted from the series value within the past period. The more recent the historical data, the greater the weight of the prediction:

$$\hat{x}(t+1) = \alpha x(t) + (1-\alpha)[\hat{x}(t) + b(t)], \tag{1}$$

$$b(t) = \beta[\hat{x}(t) - \hat{x}(t-1)] + (1-\beta)b(t-1). \tag{2}$$

The trend effect at time t, $b(t)$ is calculated as Eq. (2). The value β is used to weight the difference between two prediction values: $\hat{x}(t)$ and $\hat{x}(t-1)$, belonging to adjacent days and the preceding trend effect, $b(t-1)$. For the double exponential smoothing method, the value of $\hat{x}(t)$ and $b(1)$ have to be assigned in the initial stage. The simplest way is to make an assumption for $\hat{x}(2) = x(1)$ and $b(1) = 0$. Some research has also suggested that the selection of the initial value is not important toward the stationary [7], since it does not have a significant effect on the prediction result. In this work, we modified a double exponential smoothing method to predict the popularity degree of the topic according to the variation in trends of click times by Internet readers.

2.4. Recommendation approaches

The recommender system is widely used to provide suitable personalized information to users according to their needs and preferences [1–3,17,18,22,29,35]. The recommender system has been applied in many different areas [36], such as products [8,23], movies [32], books [10] and music [37], and not only offers personalized recommendation service for each customer, but also benefits business marketing strategies. Generally, the recommender system mainly includes content-based filtering and collaborative filtering.

The content-based filtering (CBF) approach analyzes customers' preferences regarding the item's attribute features to build up a personal feature profile, and then predict which items the customer will like [14,41]. In other words, this approach recommends items with similar attribute features to the customer profiles according to their past preferences; it is more likely to be used for document webpage and news article recommendations. However, this method still has some restrictions which need to be improved; it is not easy to analyze the features of items, and users can only receive recommended items which are similar to past ones [21].

The collaborative filtering (CF) approach is one of the most popular recommending approaches, and it has been successfully applied in many areas [4,32]. This method can solve some problems of content-based method mentioned before. There is no need to analyze the contents of an item; the recommended items are identified for target users solely based on the similarities to the historical profiles of other users. Furthermore, it can deal with items with content dissimilar to those in the past.

Based on the relationship between items or users, the CF method can be classified into two types [35]: user-based CF and item-based CF. User-based CF calculates the similarity between users, and predicts the target user's preference regarding different items; GroupLens is an example of such a system [32]. The CF approach involves two steps: neighborhood formation and prediction. The neighborhood of a target user is selected according to his/her similarity to other users, and is computed by Pearson correlation coefficient or the cosine measure. Either the k-NN (nearest neighbor) approach or a threshold-based approach is used to choose k users who are most similar to the target user.

With the numbers of users and items exploding, determining how to quickly produce high quality recommendations and search a large amount of potential neighbors in real time are important issues, especially for commercial systems. The item-based CF method has been proposed to identify the relationships between different items that users had already rated and then ranking recommended items each user has not viewed before; this method has already been applied on the Amazon platform [10], achieving good performance.

The item-based collaborative filtering (ICF) algorithm [34] first analyzes the relationships between items (e.g., documents), rather than the relationships between users. Then, the item relationships are used to compute recommendations for users indirectly, by finding items that are similar to other items which the user has previously accessed. Thus, the prediction for item j for user u is calculated by the weighted sum of the ratings given by the user for items similar to j and weighted by item similarity, as shown in Eq. (3):

$$p_{u,j} = \frac{\sum_{i=1}^{n} w(j,i) \times r_{u,i}}{\sum_{i=1}^{n} |w(j,i)|}, \tag{3}$$

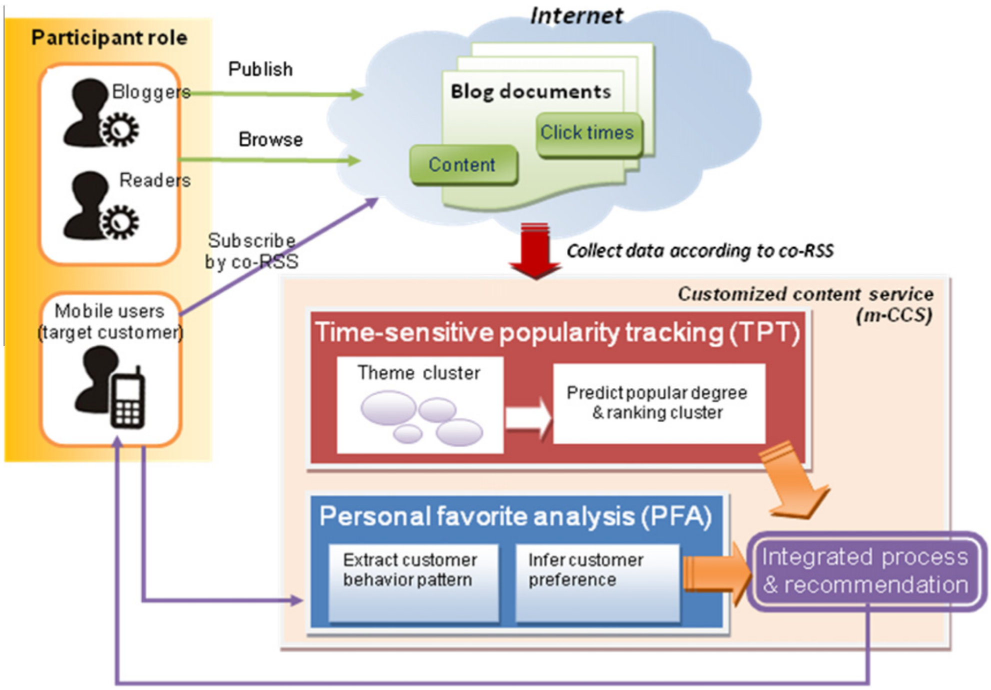

Fig. 1. System overview for m-CCS.

where $p_{u,j}$ represents the predicted rating of item j for user u; $w(j,i)$ is the similarity between two items j and i; and $r_{u,i}$ denotes the rating of user u for item i. A number of methods can be used to determine the similarity between items e.g., cosine-based similarity, correlation-based similarity, and adjusted cosine similarity methods. Since the adjusted cosine similarity method performs better than the others [34], we used it as the similarity measure for the ICF method. The adjusted cosine similarity between two items i and j is given by Eq. (4):

$$AdjSim(i,j) = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_u)^2}}, \tag{4}$$

where $r_{u,i}/\ r_{u,j}$ is the rating of item i/j given by user u; and $\bar{r}_u$ is the average item rating of user u.

The CBF method is limited in being unable to provide serendipitous recommendations since the recommendation is based solely on the content features of items that the user has preferred. The success of collaborative filtering relies on the availability of a sufficiently large set of quality preference ratings provided by users. Accordingly, finding users with similar preferences is difficult if the user rating matrix is very sparse (few preference ratings), causing the sparsity problem for the CF method. In addition, the CF method may suffer from the new item problem, in which there is no rating record on new items by which to derive the prediction [1].

3. System process overview

We propose a novel value-added mobile service, namely Customized Content Service on mobiles (m-CCS), to provide customized blog articles for mobile users based on the time-sensitive popularity of topics and personal preference patterns, as shown in Fig. 1.

The first step of our system is to collect blog articles from the Internet. The RSS mechanism is a useful way to capture the latest articles automatically without visiting each site. RSS is an abbreviation for Really Simple Syndication, which is an XML document to aggregate information from multiple web sources. Any mobile user can subscribe to RSS feeds. However, there may be a shortage of information caused by insufficient RSS feeds subscribed to individuals. Thus, we propose a *co-RSS* method to solve this problem. The co-RSS method gathers all RSS feeds from users such that RSS flocks, called crows-RSS, are formed to enrich information sources. After this preliminary procedure, the system can automatically collect desirable contents from diverse resources. Moreover, we use information retrieval technology (e.g. *tf-idf* approach) [33] to pre-process articles which are trawled every day from blog websites according to *crows-RSS* feeds. After extracting the features (term vectors) of blog articles, the *time-sensitive popularity tracking* (TPT) module groups articles into topic clusters and automatically predicts their trend of popularity. The details of the TPT module are presented in Section 4.

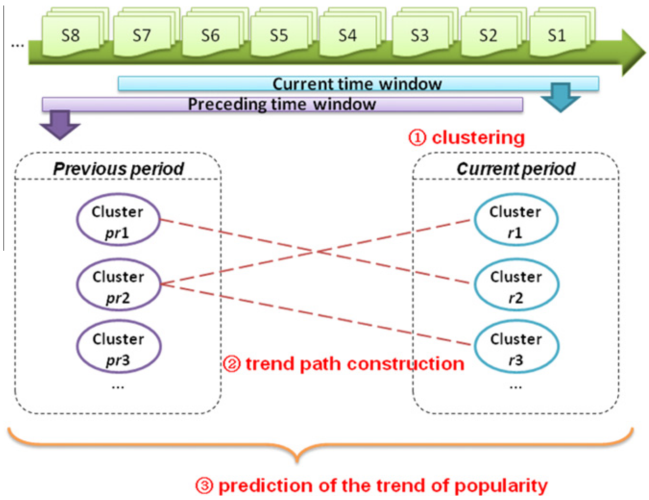

Fig. 2. Time-sensitive popularity tracking process.

Since the viewable content on mobile device screens is limited, designing a personalized service for filtering articles is particularly desirable. The m-CCS can monitor the click rates of articles daily and log user viewing records to infer implicit preference of mobile users. Without the effort of user rating, the implicit interest of a user regarding an article is inferred by comparing the time spent on reading the article with the average time spent on articles of the same size. The browsing records of users are analyzed to discover their behavior patterns and then their personal preferences are deduced through a *personal favorite analysis* (PFA) module. Moreover, the m-CCS predicts a user's preferred topics by deriving his/her customized popularity degree of topic clusters according to the predicted popularity of topic clusters and his/her preferences. Section 5 presents the details of the PFA module.

Finally, the system recommends blog articles based on the customized popularity degree of topic clusters and the preference of mobile users. The recommended articles are then sent to the user's mobile device via a WAP Push service. This allows users to instantly receive personalized and relevant blog articles. The proposed recommendation process of the m-CCS mainly integrates content analysis and collaborative filtering to improve the shortcomings of pure collaborative filtering (CF), including sparsity and cold start issues, as well as aspects such as: (1) the prediction of popular topic cluster of concern to bloggers and readers on the Internet, (2) the prediction of users' preference score by item-based collaborative filtering, and (3) attention degree (click times) of blog articles obtained from Internet users. The detailed descriptions of the recommendation process are presented in Section 6.

In general, the effectiveness of the CF recommendation approach mostly depends on the set of historical data. There are still potential limitations, such as sparsity and cold start issues [2,39]. Low-quality recommendation results may be delivered due to the sparsity issue, namely when the system only has very few rating records of users to measure the similarity between users or items. For the cold start issue of new items or new users, the system will present weak performance in recommendation because of the lack of active records viewed by users.

In our research, we focus on mobile users and blog articles. We apply clustering techniques to first group the articles into topic clusters and then form neighborhoods of items from the topic clusters, which can reduce the sparsity problem and improve the scalability of recommender systems. Additionally, many blog articles have not been viewed by any mobile user in our system due to the limitations of mobile devices. It means that most articles, which are popular on the Internet and are attractive to the masses of Internet users, may be ignored in the process of recommendation. Thus, our proposed recommendation approach not only considers mobile users' preferences concerning the articles which have been pushed to them on the mobile devices, but also considers the perspectives of Internet readers to identify the popularity of articles, in order to improve the quality of recommendation.

4. Time-sensitive popularity tracking

In this section, we present a novel approach to predict the trend of time-sensitive popularity of blog topics. We identify the blog topic clusters and their popularity according to the perspectives of writers and readers on the Internet, and then trace the trend of popularity temporally. In the following subsections, we illustrate the details of the tracking process shown in Fig. 2.

4.1. Forming topic clusters of blog articles

Articles in blogs are free and usually contain different opinions so that it is difficult to categorize articles into their appropriate categories as defined by bloggers. That is to say, the existing category in a blog website is insufficient to fully represent

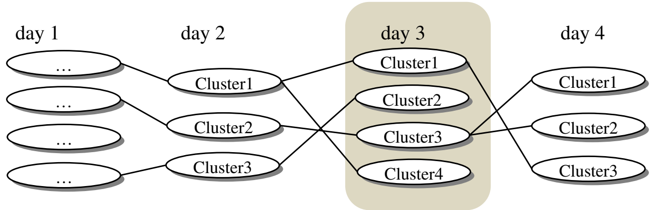

Fig. 3. The trend path of topic clusters.

the blog. In our research, we use article features, i.e., term-weight vector, derived from the pre-processing to deal with blog articles which are published within a given time window on the Internet. We collect blog articles from bog websites as the training corpus to construct the dictionary by applying one of the statistical methods, the log likelihood ratio, to extract meaningful phrases and terms. In addition, blog articles are trawled every day from blog websites according to the crowed-RSS feeds. Note that the blog training data is periodically updated and trained to update the dictionary. Significant terms/phrases are extracted from the content of an article according to the dictionary derived from the blog training data. In addition, each article is represented as a term vector by using the *tf-idf* approach [33] to calculate the weight of term i in an article j, as defined in Eq. (5):

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i}; \quad f_{i,j} = \frac{freq_{i,j}}{\max_l(freq_{l,j})}, \tag{5}$$

where N is the number of articles; n_i is the number of articles that contain term i; $f_{i,j}$ is the normalized frequency of term i in article j; $freq_{i,j}$ is the frequency of term i in article j; and $\max_l(f_{lj})$ is the frequency of term l which has the maximum frequency in article j.

The size of the time window is set as seven days. That is, all the articles posted in the past seven days will be categorized and recommended to individual users.

A hierarchical agglomerative algorithm with group-average clustering approach [16] is applied to implement the clustering step. It treats each article as a cluster first and then successively merges the pairs of clusters with highest cluster similarity. The similarities between two articles can be calculated by means of the cosine similarity measure, as shown in Eq. (6):

$$sim(d_i, d_j) = \cos(\vec{d}_i, \vec{d}_j) = \frac{\vec{d}_i \bullet \vec{d}_j}{\|\vec{d}_i\| \cdot \|\vec{d}_j\|}. \tag{6}$$

The cluster similarity between two clusters is defined as the average pairwise similarities of all pairs of articles from different clusters. The cluster similarity between two clusters r_i and r_j is calculated by Eq. (7), where d_i/d_j is a blog article belonging to the set of blog articles S_{ri}/S_{rj} in Cluster r_i/r_j; $|S_{ri}|/|S_{rj}|$ is the number of blog articles of S_{ri}/S_{rj} and $sim(d_i, d_j)$ denotes the cosine similarity between the articles d_i and d_j:

$$similarity(r_i, r_j) = \frac{\sum_{d_i \in S_{ri}} \sum_{d_j \in S_{rj}} sim(d_i, d_j)}{|S_{ri}||S_{rj}|}. \tag{7}$$

We stop merging the pairs of clusters when the highest cluster similarity is below a threshold during the merge process. The number of clusters each day is not constant; it depends on the density of the discussed topic. If the density of the topic which people discuss is high, the diversity of the article is low and the numbers of clusters decrease.

4.2. Constructing the trend path between clusters belonging to adjacent days

To reveal the path of the trend which predicts the popularity degree of current clusters, we measure the cluster similarity between the target Cluster r and all the Clusters pr belonging to the preceding period, and then select the one with maximum values to construct the link with one of the preceding clusters.

As blog articles are usually composed of unstructured words, to obtain similarity between two clusters appertaining to two days, we average the value of cosine similarity between articles crossing clusters. The similarity between two clusters (r, pr) in adjacent days is calculated by Eq. (7). After establishing the linkages, the trend of each current cluster can be derived from the preceding related cluster. As shown in Fig. 3, all of the clusters receive a trend path from the preceding cluster. The topic of *Cluster*1 in *day* 3 is evolved from *Cluster*1 in *day* 2, and so on, and we can use the relationship and similarity between them to calculate the popularity degree.

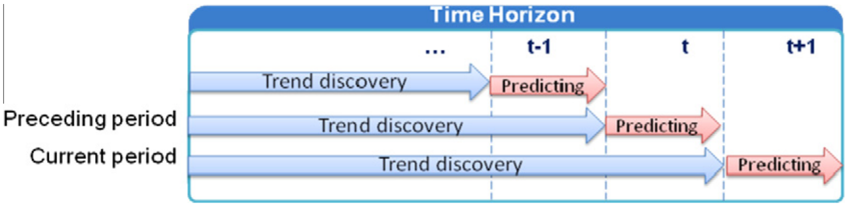

Fig. 4. The time series of popularity trend.

4.3. Acquisition of actual popularity degree for each preceding cluster

After clustering blog articles to form topic clusters (e.g. theme groups) and constructing the trend path, we mainly use reader attention, namely the click times of topic clusters, to derive the popularity degree of each cluster. To help predict the popularity degree of a current cluster, we consider the click times in proportion to the reader attention causing a topic to rise and flourish. After clustering blog articles to form a topic group and constructing the trend path, the actual popularity degree for each preceding cluster can be acquired from the times the articles have been clicked during a previous period. Let S_{pr} denote the set of blog articles in Cluster *pr*. For each preceding Cluster *pr*, we obtain $CT_t(S_{pr})$, the total click times of the articles in S_{pr} on the Internet within the preceding time period *t*, as defined in Eq. (8):

$$CT_t(S_{pr}) = \sum_{d_i \in S_{pr}} ClickTimes_t(d_i), \tag{8}$$

where the actual click times for blog article d_i in time *t* can be represented by $ClickTimes_t(d_i)$. Then, the click times can be converted to the actual popularity degree, $APD_{pr}(t)$, which is a normalized value based on the maximum *ClickTimes* over all S_k in the preceding period *t*, as defined in Eq. (9):

$$APD_{pr}(t) = \frac{CT_t(S_{pr})}{Max\{ClickTimes_t(S_k)\}} \times 100\%. \tag{9}$$

4.4. Predicting popularity degree of current cluster

We analyze the trend evolution of attention from Internet readers to predict the popularity degree of current cluster. The time series of popularity trend is a set of serial observation values by time order, as shown in Fig. 4. We modified the *double exponential smoothing* method described in Section 2.3 to forecast the degree of popular trend for each cluster of blog topic. We only give brief explanations of some equations of the double exponential smoothing method. Readers can refer to the references [6,7] for further details.

For each Cluster *r*, we use the weighted average method that combines the *actual popularity degree* (APD) and *predicted popularity degree* (PPD) of the preceding period to predict the popularity degree of current clusters on the assumption that the effect of popularity degree decays as days pass, as defined in Eq. (10):

$$PPD'_r(t+1) = \alpha \times APD_{pr}(t) + (1-\alpha) \times [PPD_{pr}(t) + b_{pr}(t)], \tag{10}$$

where we use Cluster *pr* at preceding time *t* to predict the initial popularity degree of Cluster *r* at time *t* + 1 which is denoted by $PPD'_r(t+1)$. For the preceding Cluster *pr* at time *t*, $APD_{pr}(t)$ is the actual popularity degree as mentioned above; $PPD_{pr}(t)$ denotes the predictive popularity degree of Cluster *pr* at time *t*. The $b_{pr}(t)$ represents the trend effect for the previous period. Note that the value of initial predictive popularity degree for current cluster, $PPD'_r(t+1)$, is between zero and one. The parameter α is a smoothing constant between zero and one, which is used to determine the relative importance of actual popularity degree and the predictive popularity degree with trend effect in the preceding period.

We combine the difference of the predictive popularity degrees at time *t* and at time $t-1$, and the trend effect at time $t-1$ to calculate the trend effect at time *t*, $b_{pr}(t)$, using the weighted average, as defined in Eq. (11):

$$b_{pr}(t) = \delta \times [PPD_{pr}(t) - PPD_{ppr}(t-1)] + (1-\delta) \times b_{ppr}(t-1). \tag{11}$$

Note that the Cluster *pr* is the preceding cluster of *r*, while the Cluster *ppr* is the preceding cluster of *pr*. The $PPD_{ppr}(t-1)$ and $b_{ppr}(t-1)$ are the predictive popularity degree and trend effect of Cluster *ppr* at time $t-1$, respectively. The parameter δ is a smoothing constant between zero and one, which is used to adjust the relative importance of the difference between the predictive popularity degrees at time *t* and at time $t-1$, and the trend effect at time $t-1$.

The values of α and δ in Eqs. (10) and (11), respectively, can be decided by experts or experimental analysis. The double exponential smoothing approach [7] is usually applied to analyze time series data; however, it does not consider the relation between topic clusters belonging to adjacent time periods. In our research, we concentrate on topic clusters in different time periods and construct the topic linkage from the preceding time to the current time as a topic trend path with a popularity degree. Therefore, to link topic clusters, the maximal similarity between adjacent clusters, i.e., current Cluster *r* and

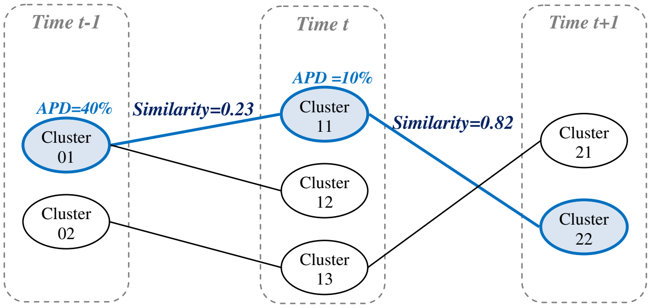

Fig. 5. The time series of topic clusters.

preceding Cluster pr, as described in Section 4.2, is selected to adjust the predictive popularity degree of Cluster r, as shown in Eq. (12). Notably, the smaller similarity leads to the lower reliability of the prediction path between two clusters:

$$PPD_r(t+1) = PPD'_r(t+1) \times similarity(r, pr). \tag{12}$$

In Fig. 5, we take one path of trend which belongs to three-day time periods as an example and set both parameters, α and δ, as 0.3. We use the popularity of *Cluster*11, which belongs to *Time t*, to predict the popularity degree of *Cluster*22 in *Time* $t+1$. In the same way, *Cluster*01 is useful to infer *Cluster*22. In the initial stage, the actual popularity degree for *Cluster*01 is assumed to be 40%. It is reasonable to assume $PPD'_{pr}(t) = APD_{ppr}(t-1)$, $PPD_{ppr}(t-1) = 0$, and $b_{ppr}(t-1) = 0$, at the starting time 0. Likewise, we also assume that predictive popularity degree $PPD_{Cluster01}(t-1)$ and the trend effect $b_{Cluster01}(t-1)$ for *Cluster*01 is zero, respectively. Thus, the initial predictive popularity degree of *Cluster*11 could be derived, and the value is 40%. Then the similarity across adjacent clusters should be considered to calculate the predictive popularity degree.

Suppose that the value of similarity between *Cluster*01 and *Cluster*11 is 0.23; we can obtain the predictive popularity degree of *Cluster*11 after adjustment as: 40% × 0.23 = 9.2%. Next, we use the values which were derived previously to predict the initial popularity degree of *Cluster*22 according to Eq. (10):

$$PPD'_{Cluster22}(t+1) = 0.3 \times APD_{Cluster11}(t) + 0.7 \times [PPD_{Cluster11}(t) + b_{Cluster11}(t)].$$

The value of trend effect, $b_{Cluster11}(t)$, is derived using Eq. (11):

$$b_{Cluster11}(t) = 0.3 \times [PPD_{Cluster11}(t) - PPD_{Cluster01}(t-1)] + 0.7 \times b_{Cluster01}(t-1) = 2.76\%.$$

Thus, $PPD'_{Cluster22}(t+1) = 0.3 \times 10\% + 0.7 \times [9.2\% + 2.76\%] = 11.37\%$. The value of similarity between *Cluster*11 and *Cluster*22 is 0.82. We obtain the final predictive popularity degree as follows:

$$PPD_{Cluster22}(t+1) = PPD'_{Cluster22}(t+1) \times similarity(Cluster11, Cluster22) = 9.32\%.$$

5. Personal favorite analysis

In this section, we present a novel scheme that models the interests of users who browse blog articles on mobile devices. Our proposed methods are implemented to enhance an existing system running in a real mobile business environment. Because of the limited features of mobile devices, it is inconvenient to give explicit relevance ratings of blog articles for mobile users. Thus, the existing system does not provide the function of explicit rating of articles. Providing explicit feedback such as rating items may bring users extra burden; because it would disturb the normal browsing process, it would usually be ignored by users [26]. Accordingly, we analyze the browsing patterns of mobile users as implicit feedback information to derive their preferences for blog articles.

5.1. Analysis of user browsing behavior

We model browsing patterns within session time by analyzing the log data of mobile users. A user's browsing pattern is derived by calculating his/her average reading time per word for browsing blog articles within session time. The system records the browsing time of blog articles requested by mobile users to derive the session interval and browsing time for each article. A timeout mechanism is used to terminate a session automatically when a user does not make any request in a time period. Calculating the time interval between user requests on articles within each session could estimate a user's browsing (stick) time on an article.

In order to acquire the browsing pattern for the user u, we analyze the browsing speed, $H_{u,s}$, to get the average browsing time per word in this session s, as shown in Eq. (13):

$$H_{u,s} = \frac{1}{|D_{u,s}|} \times \sum_{d_i \in D_{u,s}} \frac{Time_u(d_i)}{DocSize(d_i)}, \tag{13}$$

where d_i is an article i that the user u had browsed within session s; $D_{u,s}$ is a set of articles browsed by user u in session s; $|D_{u,s}|$ denotes the number of articles in $D_{u,s}$; $DocSize\ (d_i)$ identifies the number of words of the article; and $Time_u(d_i)$ denotes the user u's browsing time on blog article d_i.

After obtaining a user's current browsing behavior, $H_{u,s}$, which is viewed as the user's recent pattern within one session, we use a weighted approach to predict a user's future browsing pattern by an incremental approach, which incrementally modifies the former browsing pattern employing the user's current browsing behavior. The parameters β can be adjusted in order to set one as more important than the other. We believe that recent browsing behavior has a greater effect upon the future behavior of the mobile user, so we set the parameter β to give recent patterns more weight.

The predicted browsing pattern is calculated by using Eq. (14), where $H'_{u,s}$ denotes former browsing pattern which has been accumulated till session s for mobile user u. Then we can use the new browsing pattern at session s, i.e., $H_{u,s}$, to predict the future behavior at new session $s + 1$:

$$H'_{u,s+1} = \beta \times H_{u,s} + (1 - \beta) \times H'_{u,s}. \tag{14}$$

5.2. Inferring user preference for articles

In this step, we infer user preferences for articles based on their browsing behavior that is considered as implicit feedback information. Previous studies [27] have also found that reading time is indicative of interest. By analyzing a user's browsing time on an article, we can infer how interested the user is in the article and its corresponding preference score. If the browsing time is longer than usual, we can estimate that the user has a high preference level for the article.

According to the user's browsing behavior in usual time, we employ the user's browsing pattern mentioned in Section 5.1 to estimate the browsing time for the article and calculate the *Predict Browsing Time*, $PBT_u(d_i)$, to compare with *Actual Browsing Time*, $ABT_u(d_i)$, of the user. The predict browsing time $PBT_u(d_i)$ is equal to $DocSize(d_i) \times H'_{u,s+1}$, where $DocSize\ (d_i)$ is the size (number of words) of blog article d_i and $H'_{u,s+1}$ denote the average browsing time per word for user u as described in Section 5.1. Then, we calculate the *preference score* (PS) for target user u on blog article d_i as follows:

$$PS_u(d_i) = \frac{1}{1 + \frac{PBT_u(d_i)}{ABT_u(d_i)}}. \tag{15}$$

We can observe that the value of this function is in the range $(0,1)$; the higher value of preference score means that the user has more interest in the article.

6. Hybrid recommendation

In this section, we propose a novel hybrid method that combines user preference prediction by collaborative filtering, Internet attention degrees of articles, and customized popularity degree of topic cluster, in order to recommend personalized blog articles to mobile users.

The basic idea of this process is to integrate the different viewpoints of mobile users and Internet users. We use an item-based collaborative filtering approach to recommend the latent articles of interest according to the actual browsing behavior of mobile users. However, the CF approach suffers from the sparsity and cold start issues. Because of the limitations of the mobile device, the mobile user cannot easily surf blog articles and a lot of articles are never browsed by mobile users. It means that most popular articles on the Internet, attractive to the masses of Internet users, may be ignored in the process of recommendation. Thus, our proposed recommendation approach not only considers the mobile users' preferences concerning the articles which have been pushed to them on the mobile devices, but also considers the viewpoints of Internet readers to identify the attention degree of articles, in order to improve the quality of recommendation. We also consider the predictive popularity degree of the topic cluster to which each article belongs. The more popular the topic of an article is the more users there will be who are interested in the article.

6.1. Topic-based collaborative filtering

Research has demonstrated that the item-based CF approach can efficiently produce high-quality recommendations. The item-based CF method usually computes item similarity based on the whole set of items. However, user preferences on items of different clusters may vary, since the items of different clusters have different characteristics. Mobile users with similar preferences on a topic cluster (e.g. movies) may have different interests in other topics. As mentioned previously, we apply clustering techniques to group the articles into topic clusters first and then form neighborhoods of items from the topic clusters; this can improve the scalability of recommender systems. For each topic cluster, we adopt the item-based CF method to predict mobile users' preferred articles, due to the efficiency concern for commercial systems.

We use the adjusted cosine [34] to measure the similarity between two articles, d_i and d_j, which belong to Cluster r, as defined in Eq. (16). The set of users who co-rate both d_i and d_j is denoted by U_{ij}. The $PS_u(d_i)$ is the preference score of the user u on article d_i; $\overline{PS_u}$ is the average preference score of mobile user u:

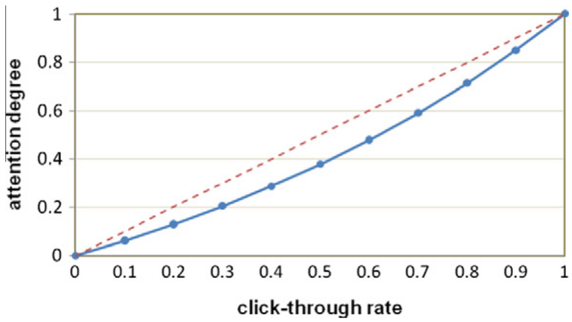

Fig. 6. The variation of attention degree.

$$sim_r^{adj}(d_i, d_j) = \frac{\sum_{u \in U_{ij}} (PS_u(d_i) - \overline{PS_u})(PS_u(d_j) - \overline{PS_u})}{\sqrt{\sum_{u \in U_{ij}} (PS_u(d_i) - \overline{PS_u})^2} \sqrt{\sum_{u \in U_{ij}} (PS_u(d_j) - \overline{PS_u})^2}}. \tag{16}$$

To predict the preference score of target user u on article i within Cluster r, the next step is to select a set of articles most similar to the target article and generate a predicted preference for the d_i using a weighted sum, as shown in Eq. (17), which is adopted from Eq. (3). $PPS_u^{cf}(d_i)$ denotes the predicted preference score of target user u on article i based on the item-CF method; d_j is the nearest neighbors of the target article d_i; and $\widehat{I}$ is the set of top-N articles most similar to the target article and had been browsed by user u:

$$PPS_u^{cf}(d_i) = \frac{\sum_{j \in \widehat{I}} PS_u(d_j) \times sim_r^{adj}(d_i, d_j)}{\sum_{j \in \widehat{I}} |sim_r^{adj}(d_i, d_j)|}. \tag{17}$$

6.2. The degree of attention for blog article

Mobile users are usually interested in those articles to which the majority of Internet readers pay attention. Within a topic Cluster r, we obtain the attention degree of an article, $attention_r(d_i)$, which is the accumulated click times indicating how much attention from Internet readers, as defined in Eq. (18):

$$attention_r(d_i) = \frac{e^{\frac{ACCT(d_i)}{Max_{d_j \in D_r}\{ACCT(d_j)\}}} - 1}{e - 1}. \tag{18}$$

The attention degree is derived from the *click-through rate*, which is calculated as $ACCT(d_i)/Max_{d_j \in D_r}\{ACCT(d_j)\}$. $ACCT(d_i)$ denotes the accumulated click times for article d_i; D_r is the set of articles in Cluster r; and $Max_{d_j \in D_r}\{ACCT(d_j)\}$ means the maximum accumulated click times of articles in Cluster r. We assume that the attention degree has the property of network externality. The larger the click-through rate of an article, the more attractive the article is to the mobile users. Fig. 6 shows that the value of attention degree, $attention_r(d_i)$, rises as the click-through rate increases, and it is between zero and one.

6.3. Customized predictive popularity degree

In the process of *time-sensitive popularity tracking* (TPT), we apply a *modified exponential smoothing* method to predict the general popularity degree of topic cluster. In this section, we further consider each user's preference obtained from the *personal favorite analysis* (PFA) step to derive the customized (personalized) popularity degree of topic cluster. An article can be included in different topic clusters belonging to successive time periods. Once a mobile user has read an article, his/her preference score is inferred from the browsing behavior. The customized popularity degree of topic cluster can then be derived using a user's preference scores for a certain article belonging to this cluster.

Two methods are designed to derive the *customized predictive popularity degree* (CPPD) of topic Cluster r for a specific user u. The first one is called *weighted customized predictive popularity degree* (WCPPD) method and is presented in Eqs. (19) and (20):

$$WCPPD_{u,r} = \omega_{u,r} \times PPD_r + (1 - \omega_{u,r}) \times \frac{\sum_{d_j \in D_{u,r}} PS_u(d_j)}{|D_{u,r}|}, \tag{19}$$

$$\begin{cases} \omega_{u,r} = \frac{1}{|D_{u,r}|}, & if |D_{u,r}| > 1 \\ \omega_{u,r} = 1, & \text{otherwise}. \end{cases} \tag{20}$$

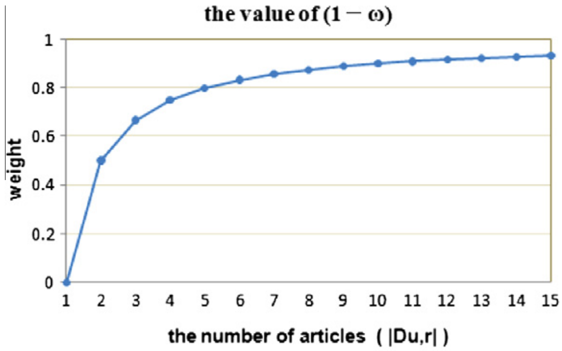

Fig. 7. The value of 1-ω with different number of browsed articles.

We use the average *preference score* (PS) of user u for those articles that have been read by u and contained in the target Cluster r to adjust the *predictive popularity degree*, PPD_r, for user u. $D_{u,r}$ denotes the set of articles that user u has browsed in Cluster r. The parameter ω is used to adjust the relative importance of PPD_r and the average *preference score*. The value of ω is smaller if user u has browsed more number of articles in Cluster r; thus more weight $(1-\omega)$ is assigned to the average *preference score* (PS) of user u.

The derivation of personalized *CPPD* is according to the reliability of the user's personal preference derived from the historical records of the user. If user u has browsed more articles, the system is more capable of predicting the user's preference score (PS), and thus the user's personal preference is more reliable. On the contrary, if user u has browsed fewer articles, then the user's personal preference is less reliable since the system may not be able to predict the user's preference score (PS) based on insufficient browsing records. With more reliable personal preference, i.e. more browsing records, the *CPPD* is influenced more by the average preference score (PS) of user u. The *CPPD* is, however, dominated by the general predictive popularity degree, PPD_r for the users who have very few browsing records to infer their preferences.

For those mobile users who have sufficient browsing records in our system, the popularity degree of topic cluster provided by m-CCS would be more customized. The system would assign more weight to personal characteristics of users who have sufficient historical behavior records, and give less weight to the general popularity degree of topic clusters. Conversely, if a user has very few behavior records to be analyzed, the degree of modification of topic clusters is smaller. That is, the less the browsing history of users, the less the personal ranking of clusters. The system will recommend the more general and popular topic clusters.

The second approach is called the *harmonic customized predictive popularity degree* (HCPPD) method. The basic idea of this method is to apply the harmonic mean approach to combine the *predictive popularity degree* (PPD_r) and the *adjusted average preference score* $\left(\overline{PS}^{adjusted}_{u,r}\right)$ for each topic cluster as in Eq. (21). We derive the *adjusted average preference score* according to Eq. (22). The weight value, $1-\omega_{u,r}$ defined in Eq. (20) is used to adjust the average preference scores. The *adjusted average preference score* would be high if a user browses more articles within topic cluster and shows higher preference for those articles. Moreover, according to the characteristic of harmonic mean, the customized predictive popularity degree of Cluster r for user u will be high if both the *predictive popularity degree* of Cluster r and *the adjusted average preference score* of user u are high:

$$HCPPD_{u,r} = \frac{2 \times PPD_r \times \overline{PS}^{adjusted}_{u,r}}{PPD_r + \overline{PS}^{adjusted}_{u,r}}, \tag{21}$$

$$\overline{PS}^{adjusted}_{u,r} = (1-\omega_{u,r}) \times \frac{\sum_{d_j \in D_{u,r}} PS_u(d_j)}{|D_{u,r}|}. \tag{22}$$

The adjusted average preference score is derived from the user's average preference score by using the weight value, $1-\omega_{u,r}$ to conduct the adjustment. The weight value, $1-\omega_{u,r}$, varies according to the number of articles that had been browsed by user u. The weight value $1-\omega_{u,r}$ is larger if the user had browsed more articles.

Fig. 7 identifies a tendency of the weight value, $1-\omega_{u,r}$, with regard to the different number of browsed articles ranging from one to fifteen. From the plots in Fig. 7, we observe that the value of weight, $1-\omega_{u,r}$, increases as the number of browsed articles increases. With more articles browsed by a user, his/her personal preference of historical records would become more important to affect the value of the customized predictive popularity degree of the topic cluster; thus, the average preference score is adjusted by multiplying it with a higher weigh value to derive the adjusted average preference score. Moreover, the weight value, $1-\omega_{u,r}$, increases rapidly for fewer browsed articles, while the curve trends to be flat for more numerous browsed articles. Generally, the adjusted average preference score would be high if a user browses more articles within topic cluster and shows higher preference for those articles.

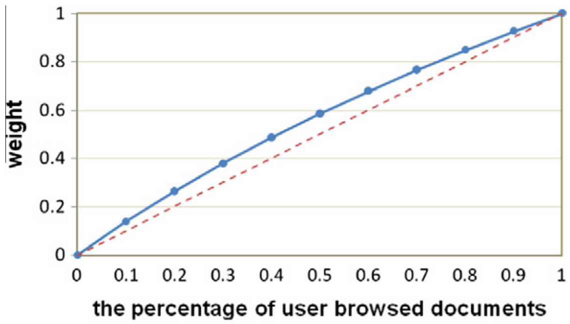

Fig. 8. Weight values in different percentages of user-browsed articles.

Fig. 9. m-CCS interface to display recommended contents on mobile device.

6.4. Article selection and recommendation

In this section, we propose a hybrid model that integrates the previous processes to recommend articles to mobile users. We derive the predictive preference score of a mobile user u on article d_i, $PPS^{hybrid}_{u,r}(d_i)$, as a hybrid of $PPS^{cf}_{u,r}(d_i)$, the predictive preference score by collaborative filtering, and $attention_r(d_i)$, the Internet readers' attention degree on the article within the Cluster r. $PPS^{hybrid}_{u,r}(d_i)$ can be expressed as Eq. (23). The parameter τ_u, which is used to adjust the relative importance of $PPS^{cf}_{u,r}(d_i)$ and $attention_r(d_i)$, is defined in Eq. (24). $\left|D^{push}_{u,t}\right|$ is the number of articles pushed to user u within time period t and $\left|D^{browsed}_{u,t}\right|$ denotes the number of articles that the user has browsed within time period t. The more articles a user has browsed, the more personal interest is emphasized when the historical records of the mobile user are sufficient to predict his/her preference (e.g. $PPS^{cf}_{u,r}(d_i)$). In contrast, the attention degree, which represents the opinion of the masses on the Internet, is more important to compute the prediction (for recommendation) when very few records of browsing articles exist and the system cannot effectively infer the mobile user's preference:

$$PPS^{hybrid}_{u,r}(d_i) = \tau_u \times PPS^{cf}_{u,r}(d_i) + (1 - \tau_u) \times attention_r(d_i), \tag{23}$$

$$\tau_u = \log_2\left(\frac{\left|D^{browsed}_{u,t}\right|}{\left|D^{push}_{u,t}\right|} + 1\right). \tag{24}$$

The computation of $PPS^{hybrid}_{u,r}(d_i)$ is according to the reliability of personal preference derived from the historical behavior of the user. With more reliable personal preference, i.e. more browsing records, the system is more capable of inferring the user's preference based on sufficient browsing records; thus, $PPS^{hybrid}_{u,r}(d_i)$ is influenced more by $PPS^{cf}_{u,r}(d_i)$. $PPS^{hybrid}_{u,r}(d_i)$, however, is dominated by $attention_r(d_i)$ for the users who have very few browsing records to infer their preferences since the personal preference derived from historical behavior of the user may be unreliable due to insufficient browsing records for analysis.

The value of τ_u is between zero and one, and the plots at different percentages of browsed articles, calculated by $\left|D_{u,t}^{browsed}\right|/\left|D_{u,t}^{push}\right|$ distribution, are shown in Fig. 8. When the browsing records are insufficient, τ_u tends to zero; $PPS_{u,r}^{cf}(d_i)$ is ignored and the final preference is mainly decided by using $attention_r(d_i)$. In contrast, with τ_u approaching the maximum value one, $PPS_{u,r}^{hybrid}(d_i)$ is mainly derived by $PPS_{u,r}^{cf}(d_i)$. The upward curve is slightly convex. That is to say, the value of the weight increases rapidly for smaller percentage of browsed articles, while the curve tends to the flat for larger percentage of browsed articles. We consider that user preference appears significant in the beginning of browsing behavior.

So far, we have generated the predictive preference on articles within clusters. To select the recommended articles from different clusters, we have to consider the priority (ranking) of topic clusters according to the customized popularity degree. In Section 6.3, we derived the customized predictive popularity degree, $CPPD_{u,r}$($WCPPD$ and $HCPPD$), to denote user u's personalized ranking of topic clusters. We derived the final predicted preference score of user u on article d_i, $PPS_u^{rec}(d_i)$ by further applying $CPPD_{u,r}$ to adjust user u's latent interest for articles cross topic cluster, as shown in Eq. (25). The articles with top-N predicted preference score are selected for recommendations:

$$PPS_u^{rec}(d_i) = PPS_{u,r}^{hybrid}(d_i) \times CPPD_{u,r}. \tag{25}$$

After the processing above, the selected articles are transformed into XHTML format for mobile devices and then pushed to the handsets via WAP. The system will push no more than ten titles of articles (see Fig. 9), due to the limitation of mobile devices and short user browsing time compared with that of PC users. Then, users can click the title in which they are interested to view the full contents.

7. System architecture

This research was conducted in collaboration with the CAMEO InfoTech Inc., provider of the WAP Push service for mobile phone users of Chunghwa Telecom (CHT), the biggest telecom company in Taiwan. We are implementing an m-CCS system of the proposed mobile service based on the CHT mobile customer database stored in the MySQL database; it was developed using the Python programming language. The operating web GUI uses a Django Web Framework (see Fig. 10).

The system adopts two IBM 1U servers to process the load balance computing and provide the browsing service for mobile phones. The system adopts the WAP (Wireless Application Protocol) push service, which is a SMS (Short Message Service) message containing a link to a WAP page. Users can access the WAP content, receiving a WAP Push message on compatible mobile handsets. On the mobile carrier site, the WAP Gateway is built in the machine room of the system operator. With the WAP Gateway, the system can reduce the traffic of wireless transmissions by encoding the mobile WAP page which contains the message and URL. The system then transforms the WAP Push message to the SMS format of GSM and dispatches the message to the mobile phone through SMSC, which is a device belonging to the system operator. Thus, the titles and URL links of articles can be shown on the mobile phone. The implementation of the m-CCS system is targeted for thousands of real users in practice. Therefore, the system must overcome the issues of efficiency and scalability.

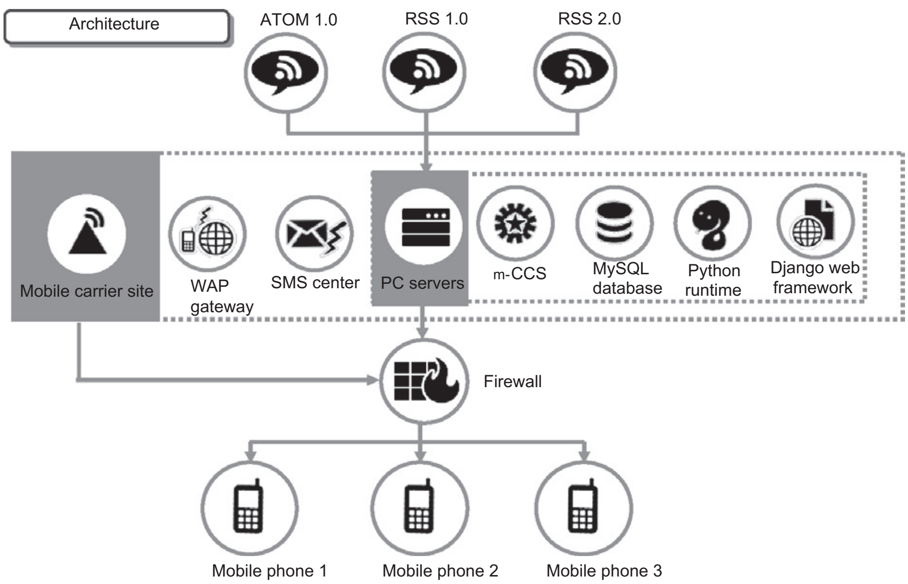

Fig. 10. The system architecture of m-CCS.

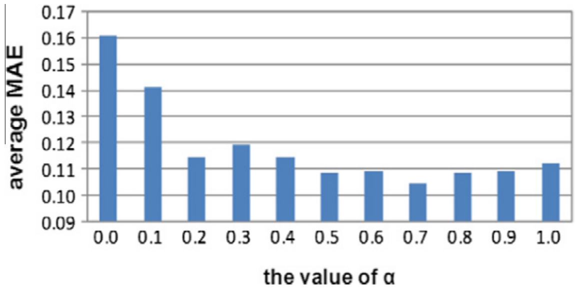

Fig. 11. The average MAEs under different $\alpha(\delta = 0.5)$.

We not only adopted the load-balancing architecture, but also carefully chose the algorithm and caching technology, in order to apply the system in a real business environment.

8. Applications and experimental evaluation

In this section, we evaluate the effectiveness of our proposed time-sensitive popularity tracking module and personalized recommendation service in Sections 8.1 and 8.2, respectively.

8.1. The evaluation of time-sensitive popularity tracking

In this section, we evaluate the performance of time-sensitive popularity tracking by comparing the difference between predicted popularity and the actual popularity of topic clusters.

8.1.1. Data sets and experimental design

In our research, we processed the latest data from Internet every day. System robots automatically Crawl the net for the newest blog articles according to the co-RSS feeds in real time. Since RSS is a well-structured format, it is easy to detect new posts. When there is a new post, the system will trigger the process of capturing articles. However, the RSS usually contains partial information on the articles. In order to get the whole content, m-CCS needs to capture the primitive HTML through the URL of the blog. Furthermore, we need to parse the HTML to get the article title, content, and publish time from a variety of websites. Finally, the well-structured data are stored into database.

The total number of new published articles collected from co-RSS feeds is around two thousand daily. To conduct the popularity prediction, it is necessary to capture the daily click times of captured articles within time window from Internet. We chose the blog sites providing information about click-times to conduct our evaluation. Accordingly, four popular blog sites in Taiwan, including Wretch (http://www.wretch.cc), Pixnet (http://www.pixnet.net), Mobile01 (http://www.mobile01.com), and Mypaper on PChome (http://mypaper.pchome.com.tw), were selected to conduct our experiments. There are around 150 new articles published daily from the blogs of these four blog sites and subscribed by co-RSS.

Time window is set as seven days. Articles published within the time window were processed to predict the popularity degree of topic clusters. About one thousand articles were chosen for analysis. The data set with click times of articles was collected form blog websites during the two-week period starting from the 10th of May 2009. In the topic clustering phase, we set a threshold value of 0.002 as the condition to stop the grouping of articles.

To evaluate the prediction model, the mean absolute error (MAE) [5] is used as the evaluation metric. As shown in Eq. (26), the MAE is calculated by the average absolute deviation between the predicted result and the actual result at particular time t, where S^t is the topic cluster set which was derived at time t, and $|S^t|$ denotes the number of topic clusters. The larger the MAE, the greater the error in the prediction model; thus, a model which presents a lower MAE can be regarded as a better model:

$$MAE_t = \frac{\sum_{r \in S^t} |PPD_r - APD_r|}{|S^t|}. \tag{26}$$

8.1.2. Evaluation result

The experiment was conducted for two weeks. During the period from the 10th of May to the 24th of May 2009, the PPD_r and APD_r of each topic cluster were derived every two days, and then we calculated the value of MAE to examine the quality of prediction model. We expect that the error rate of prediction decreases, i.e., the predictive popularity degree of topic cluster is improved, as time evolves.

This section presents the experimental result of prediction models based on different weight settings of parameters. As mentioned in Section 4, the parameter α used in Eq. (10) is used to determine the relative importance of the actual popularity

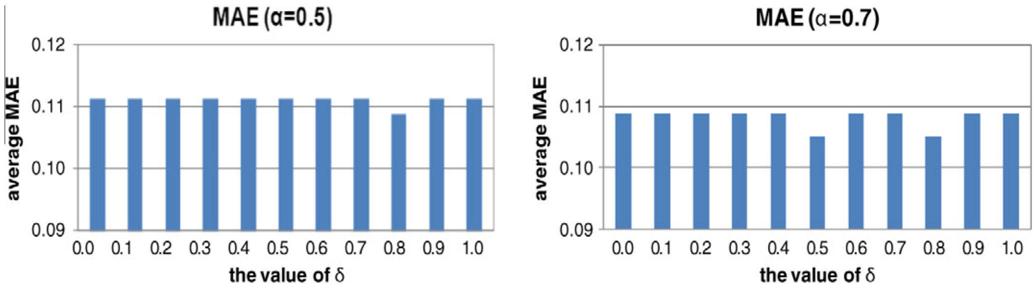

Fig. 12. The average MAE under different values of $\delta(\alpha = 0.5;\ \alpha = 0.7)$.

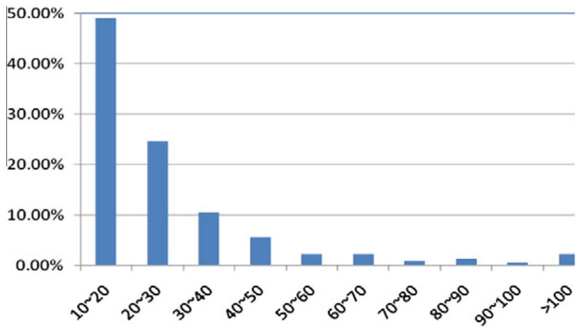

Fig. 13. The distribution of click counts for testing users.

degree and the predictive popularity degree with the trend effect in the preceding period. Parameter δ used in Eq. (11) is used to adjust the relative importance of the difference of the predictive popularity degrees at time t and $t-1$, and the trend effect at time $t-1$.

To determine the sensitivity of weight between the actual popularity degree and the variation trend in the preceding period, we performed an experiment by varying the value of α from 0.0 to 1.0 with an increment of 0.1, and setting the default value of δ as 0.5. Fig. 11 presents the average of MAE over seven prediction periods under various values of parameter α. The prediction model has the lowest MAE under $\alpha = 0.7$; this means that predicting the popularity degree of topic cluster can be more accurate when the system puts more weight on the preceding actual popularity degree.

To examine whether the value of δ would affect the result of MAE, we varied the value of δ under two fixed values of α: 0.5 and 0.7. The averages of MAE are plotted in Fig. 12. The result shows that there is no significant effect on the prediction errors (MAE) under different δ. In general, the best prediction accuracy is achieved under $\boldsymbol{\delta} = 0.8$, and it implies that the differences between successive predictive popularity degrees, which are derived at time t and time $t-1$, respectively, has an important impact on deriving the trend effect at time $t+1$. Moreover, the prediction accuracy under $\boldsymbol{\alpha} = 0.7$ is better than the accuracy under $\boldsymbol{\alpha} = 0.5$. Based on the above findings, the best parameter settings to predict the popularity degree of topic cluster are $\boldsymbol{\alpha} = 0.7$ and $\boldsymbol{\delta} = 0.8$, and such parameter settings are used in the rest of our experiments.

8.2. Evaluation of recommending blog articles

In this section, we conduct on-line experiments to evaluate our proposed approach in an online business environment. The experiments are conducted in collaboration with the CAMEO InfoTech Inc., the provider of the WAP Push service for mobile phone users of Chunghwa Telecom in Taiwan.

8.2.1. Data sets

Mobile users use the blog-service provided by Chunghwa Telecom for free 30 day trials; then, if they feel satisfied, they can become formal paid subscribers to enjoy of the use of this service. Prior to the implementation of our proposed m-CCS system, all the blog articles have been selected by the human experts, and then sent to all customers without considering users' personalized preferences. Currently, there are 18,136 users in the trial period of the system; the number of formal paid subscribers is 4967 persons. We only select formal paid users for evaluation, since free-trial users could stop using the service when their trial period ended.

Within the last one month, there were 4104 articles published on those four Internet blog websites mentioned above. Each mobile user, on average, only browsed 27.93 articles, i.e., 0.68% articles published on Internet. According to this observation, the number of articles browsed by mobile users is lower than that of Internet users because of the limitations of the mobile environment. Therefore, it is important to increase the click rates of mobile users by recommending the latest and interesting articles to mobile users.

We randomly selected 300 former paid customers with historical records of click times over ten times within the latest month as *testing users* to conduct the experiment. Among them, the highest record of click time was 257 times in one month; the lowest, 12 times. Fig. 13 illustrates the distribution of click times collected from the historical records of testing users. The amount of testing users who browse the blog articles from 10 to 20 times within one month, i.e., 3 to 5 times per week in average on mobile phone, is around 50%. About 25% testing users browse articles from 20 to 30 times within one month.

8.2.2. Design of the experiments

The item-based CF method was adopted to predict the preference scores of articles based on the article-similarity analyzed from the browsing log of mobile users. However, as mentioned previously, most blog articles published on the Internet are infrequently read by mobile users due to the limitations of the mobile environment. The deficiency of historical browsing data would result in poor performance for traditional recommendation methods, especially for the collaborative filtering methods. Moreover, in mobile environments, it is important to recommend new articles which have not been read by any mobile user but are attractive to Internet users. The CF methods also suffer from the cold-start problem of recommending new items (articles). In order to solve this problem, we have proposed a time-sensitive popularity-tracking module to predict the emerging trend of topic popularity in which most mobile users will be interested. Moreover, a customized approach is further developed to predict the customized predictive popularity of topic, and integrated with item-based CF for personalized recommendations of blog articles. With this approach, mobile users can timely receive the latest hot topic articles by mobile device any time and any place.

There are several factors which affect the quality of the recommendations. They include the personalized degree of the system, the predictive popularity degree of topic cluster, and the recommendation approach. Through the experiments, we will discuss the issues listed below.

- Does the system method's customized recommendation based on personal preferences of mobile users perform better than the expert method with human selection of articles?
- Does the method with customized predictive popularity degree of topic cluster perform better than the non-customized one?
- What is the effect of different approaches on deriving the customized predictive popularity degree of topic cluster?

To experimentally verify the effectiveness of our proposed methods, we compared different recommendation approaches. The *expert method* selected articles by human experts and then pushed the identical articles to all customers without considering mobile users' personal preferences. The system methods analyzed user preferences and then pushed the customized articles to mobile users automatically. The system methods include *non-CPPD* method, *weighted-CPPD* method and *harmonic-CPPD* method.

The *non-CPPD* method uses a formula similar to Eq. (23) to predict the preference score of an article by combining the predictive preference score derived from collaborative filtering and the attention degree of the article without considering *CPPD*. The non-CPPD method can be regarded as an enhancement of the conventional CF method by considering the attention degrees of articles. As mentioned in Section 6.3, there are two approaches to derive *CPPD*: the weighted method and the harmonic mean method. Both the *weighted-CPPD* and *harmonic-CPPD* methods use Eq. (25) to derive the final predictive preference score by considering the preference score derived from collaborative filtering, attention degree of the article and the customized predictive popularity degree of topic clusters.

The impacts of those methods on recommendation effectiveness are investigated in this experiment. We compared different recommendation models and evaluated their recommendation quality in Section 8.2.3. Note that the parameters, $\boldsymbol{\alpha}$ and δ had been experimentally determined in a previous experiment and were set as $\boldsymbol{\alpha} = \mathbf{0.7}$ and $\delta = \mathbf{0.8}$.

Moreover, the m-CCS system not only recommends articles matching personal interests of mobile users according to the analysis of behavior log but also recommends the latest and new articles which have not been browsed by any mobile user. The new articles were selected based on the time-sensitive popularity prediction and the attention degree (click times) of Internet users. Section 8.2.4 presents the experiment result on comparing the effect of system methods on recommending new articles. We would demonstrate that recommending new articles based on customized predictive popularity degree (CPPD) of topic cluster is more effective than the method without considering CPPD.

8.2.3. Comparing different recommendation methods

We conducted **on-line experiments** by recommending blog articles to 300 testing users selected, as described in Section 8.2.1. The recommendations are pushed to testing users in an on-line real business environment with the cooperation of CAMEO InfoTech Inc. To avoid disturbing customers, we could not send the recommended articles to users every day; instead, the frequency of on-line recommendation was three times a week. In other words, the system pushes blog articles once per two days on average, and only ten articles are pushed to a user each time because of the limitations of the small screen of mobile device. Moreover, the cooperation company cautiously agreed with a limited scope of on-line experiments to avoid disturbing and losing customers. Due to such limitations on conducting on-line experiments in a real mobile

Table 1
The number of recommended articles which are clicked by customers.

Recommendation method	No. of clicked articles	Hit ratio
Expert selection	219	7.30%
Non-CPPD	284	9.47%
Weighted-CPPD	302	10.07%
Harmonic-CPPD	336	11.20%

Table 2
The hit ratio of different types of articles for *non-CPPD* and *weighted-CPPD* methods.

Recommendation method	Type of articles	The number of pushed articles	The number of clicked articles	Hit ratio
Non-CPPD	New	2255	190	8.43%
	Existing	745	93	12.48%
Weighted-CPPD	New	1957	195	9.96%
	Existing	1043	124	11.89%

experiment, we used equations to derive the values of parameters in our proposed approaches without comparing our approaches with different parameter settings.

Most studies on recommender systems conducted off-line experiments which usually compared various methods off line without the risk of disturbing and losing customers. Differing from the off-line experiments, on-line evaluations in a real business environment need to be conducted cautiously in comparing methods with reasonable quality to avoid disturbing and losing customers. Due to such limitations, we conducted on-line experiments to compare the expert method, the non-CPPD method and the proposed CPPD method (weighted-CPPD and harmonic-CPPD). As described in Section 8.2.2, the non-CPPD method is an enhancement of the conventional CF method, by considering the attention degrees of articles.

The recommendation message can be sent by WAP Push service which is the most suitable way to deliver information to mobile users. The message to be pushed is limited within about 100 Chinese characters at one time, so it cannot display all the contents of articles. Therefore, the combination of article titles and URL link would be used as a message to be forwarded to the mobile phone users. Although the system can push messages merely within 100 Chinese characters at first, it can flexibly arrange the picture and word in XHTML format after connecting to the Internet.

While the WAP-Push messages catch the attention of customers, the mobile phone browser can be applied to receive the complete recommended articles by the attached URL link. Once the URL link is opened, the system first generates a set of titles which belong to the top-N recommended articles (here we set $N = 10$); then, if mobile users have any interest in those articles by reading the titles, they can click the articles (URLs) to request the full text and browse the details of articles.

We employed the hit ratio to evaluate the experiment results. The hit ratio is the fraction of recommended items (predicted to be interesting) that are clicked by the target users. The concept of hit ratio is similar to the standard precision formula. For mobile users, we consider click behavior as preference expression. A click on an article made by a mobile user indicates that the user has a certain degree of interest on the article after reading its title. The hit ratio, presented in Eq. (27), is defined as the ratio of the number of articles clicked (requested) by the target users to the number of articles recommended to the target users. The higher hit ratio implies higher recommendation quality. For each run of the experiment, there were 300 testing (target) users and all of them received 10 recommended articles, so that the size of recommended set was 3000:

$$\textit{Hit ratio} = \frac{\text{the number of clicked articles}}{\text{the number of recommended articles}}. \tag{27}$$

The number of clicked articles and the hit ratio for each method are listed in Table 1. We can observe that the hit ratio of the expert method is the lowest; the system methods perform better than the expert method does. The *non-CPPD* method performs worse than the *CPPD*-based methods, including the *Weighted-CPPD* and *Harmonic-CPPD* methods. Thus, the customized predictive popularity degree is effective in improving the quality of recommending blog articles to mobile users. Moreover, the *Harmonic-CPPD* recommendation has the highest hit ratio. The harmonic-mean approach is more effective than the weighted approach in deriving the customized predictive popularity degree of topic cluster for target users. We note that in harmonic-mean approach, the *CPPD* of Cluster r for user u is high if both the predictive popularity degree of Cluster r and the adjusted average preference score of user u are high.

For system methods, the customized articles would be recommended to satisfy individual interest. The customized recommendation delivered by the m-CCS system can provide a better hit ratio than both the human selection method and the enhanced conventional CF method (Non-CPPD). We expect that the result of system prediction toward user preferences can be further improved after delivering our m-CCS system on the real business environment for a certain period of time.

Table 3
The hit ratio of different types of articles for *non-CPPD* and *harmonic-CPPD* methods.

Recommendation method	Type of articles	The number of pushed articles	The number of clicked articles	Hit ratio
Non-CPPD	New	2540	167	6.57%
	Existing	460	98	21.30%
Harmonic -CPPD	New	2280	224	9.82%
	Existing	720	150	20.83%

8.2.4. The effect on recommending new articles

In mobile environments, it is important to recommend new articles which have not been browsed by any mobile user but are attractive to Internet users. Traditional CF methods can be adopted to recommend existing articles which have been read by other mobile users. The CF methods suffer the cold-start problem of recommending new items (articles). Our proposed mechanism deals with new articles by recommending the latest and diversified articles to mobile users that may satisfy their interests. This section presents the experiment result for comparing the effect of system methods on recommending new articles.

We selected two portions of the experiment process to investigate the effect of various approaches on different types of articles: the new articles and existing articles, by comparing the *non-CPPD* method with the CPPD-based methods, including the *weighted-CPPD* and the *harmonic-CPPD* method. Note that the *non-CPPD* method uses the item-CF and attention degrees of articles to make predictions without considering *CPPD*.

We derive the hit ratio for the new and existing articles, respectively. The comparison of the *non-CPPD* method with the *weighted-CPPD* method is listed in Table 2, while the comparison of the *non-CPPD* method with the *harmonic-CPPD* method is listed Table 3. The total number of recommended articles for 300 testing users is 3000. The result shows that no matter which method is applied, the number of new articles recommended is greater than the number of existing ones. For existing articles, the hit ratio of the *non-CPPD* method is slightly higher than those of the *CPPD*-based methods, but the number of clicked articles of the *non-CPPD* method is lower than those of the *CPPD*-based methods.

For new articles, both the hit ratio and the number of clicked articles of the *CPPD*-based methods are higher than those of the *non-CPPD* method. Accordingly, the *CPPD*-based methods perform better than the *non-CPPD* method does. Thus, recommending new articles based on customized predictive popularity degree (*CPPD*) is more effective than the method without considering *CPPD*.

9. Conclusion and future work

Owing to their dramatic growth in recent years, blogs have become a dominant medium on the Internet. Our study contributes to developing a new value-added service for mobile phone applications by proposing a novel Customized Content Service on a mobile device (m-CCS) to recommend personalized and popular blog articles to mobile users. In the past, studies on blogs centered on bloggers (the authors) but ignored the views of mass readers. Existing recommendation approaches did not address the issue of how to predict the popularity trend of blog topics. We have proposed a novel approach to predict the trend of time-sensitive popularity of blogs. The proposed m-CCS system is grounded on the topic clusters of the blog articles which represent the perspectives of the authors. The m-CCS also considers Internet readers' click rates to trace the popularity trends of the topic clusters from the affluent blog contents. For mobile phone users, the m-CCS will analyze their browsing behaviors and personal preferences to recommend their preferred popular blog topics and articles. Existing recommendation approaches do not consider the popularity degrees of blog articles. We contribute to proposing a novel hybrid approach combining personalized popularity of topic clusters and attention degree (click times) of blog articles with the item-based collaborative filtering (CF) to enhance the quality of recommending personalized and popular blog articles. In addition, our proposed approach serves as a novel attempt to adopt a recommendation service in a real mobile business environment.

Our evaluations were conducted in an on-line mobile business environment with the cooperation of a company that provides the WAP Push service for mobile phones. The cooperation company restricted the scope of the on-line experiments to avoid disturbing and losing customers. Due to such limitations, our proposed CPPD (weighted-CPPD and harmonic-CPPD) methods have been compared with the expert method and the non-CPPD method which is an enhancement of a convention CF method. The evaluation results show that our proposed methods can provide better recommendation quality (hit ratio) than the expert method and the conventional CF method enhanced with the attention degrees of articles. The proposed methods can effectively increase the hit ratio of customers who use their mobile phones to read blog articles. Considering customized predictive popularity degree is effective in improving the quality of recommending blog articles to mobile users. Moreover, the harmonic-mean approach is more effective than the weighted approach in deriving the customized predictive popularity degree of topic cluster for target users.

A recommended article list is arranged according to the predicted user's preference scores on articles. The order of the recommended article list will affect the mobile users' reading behaviors on the mobile phone. Generally, the top ranking article may have a higher click rate, since the mobile phone with small screen is difficult to scroll. Mobile users may have different degrees of preferences in browsing articles. In addition, users usually show more interest in those articles being clicked earlier. Our future work will consider user feedback on browsing recommended articles to adjust the prediction scores. For example, if a mobile user clicks the lower ranking articles first, it denotes that the user may have more interest in those articles; thus, we should put more weight on those articles during the process of inferring user preferences and making predictions.

Moreover, our recommendation approach considers item-based CF, attention degrees of articles and customized predictive popularity degrees of topic clusters, along with an attempt to balance the trade-off in recommending existing articles and new articles. Some mobile users may always pursue the newest and hottest articles, while some mobile users may be interested in those articles which are worth reading even though they are not new articles. Further study is required to investigate effective recommendation approach that considers mobile users' preferences toward browsing existing and new articles.

Acknowledgement

This research was conducted in collaboration with CAMEO InfoTech Inc., and was supported in part by the National Science Council of Taiwan under Grant No. NSC99-2410-H-009-034-MY3.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge Data Engineering 17 (6) (2005) 734–749.
[2] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Information Sciences 178 (1) (2008) 37–51.
[3] A.B. Barragáns-Martínez, E. Costa-Montenegro, J.C. Burguillo, M. Rey-López, F.A. Mikic-Fonte, A. Peleteiro, A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition, Information Sciences 180 (22) (2010) 4290–4311.
[4] D. Billsus, M.J. Pazzani, Learning collaborative information filters, in: Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., 1998, pp. 46–54.
[5] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence, 1998, pp. 43–52.
[6] R.G. Brown, Smoothing, Forecasting and Prediction of Discrete Time Series, Dover Publications, 2004.
[7] C. Chatfield, M. Yar, Holt-Winters forecasting: some practical issues, Journal of the Royal Statistical Society. Series D (The Statistician) 37 (2) (1988) 129–140.
[8] L.-S. Chen, F.-H. Hsu, M.-C. Chen, Y.-C. Hsu, Developing recommender systems with the consideration of product profitability for sellers, Information Sciences 178 (4) (2008) 1032–1048.
[9] N. Glance, M. Hurst, T. Tomokiyo, BlogPulse: automated trend discovery for weblogs, in: WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 2004.
[10] Greg Linden, Brent Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Computing (2003) 76–80.
[11] D. Gruhl, R. Guha, R. Kumar, J. Novak, A. Tomkins, The predictive power of online chatter, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, ACM, New York, NY, USA, Chicago, Illinois, USA, 2005, pp. 78–87.
[12] D. Gruhl, R. Guha, D. Liben-Nowell, A. Tomkins, Information diffusion through blogspace, in: Proceedings of the 13th International Conference on World Wide Web, ACM, New York, NY, USA, 2004, pp. 491–501.
[13] C. Hayes, P. Avesani, S. Veeramachaneni, An analysis of bloggers and topics for a blog recommender system, in: Proceedings of the Workshop on Web Mining, 7th European Conference on Machine Learning and the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD), Berlin, Germany, 2006, pp. 18–22.
[14] W. Hill, L. Stead, M. Rosenstein, G. Furnas, Recommending and evaluating choices in a virtual community of use, in: Proceedings of the SIGCHI Conference on Human factors in Computing Systems, ACM Press/Addison-Wesley Publishing Co., Denver, Colorado, United States, 1995, pp. 194–201.
[15] T.C. Huang, S.C. Cheng, Y.M. Huang, A blog article recommendation generating mechanism using an SBACPSO algorithm, Expert Systems with Applications 36 (7) (2009) 10388–10396.
[16] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.
[17] B. Jeong, J. Lee, H. Cho, Improving memory-based collaborative filtering via similarity updating and prediction modulation, Information Sciences 180 (5) (2010) 602–612.
[18] P. Kazienko, M. Adamski, AdROSA – Adaptive personalization of web advertising, Information Sciences 177 (11) (2007) 2269–2295.
[19] R. Kumar, J. Novak, P. Raghavan, A. Tomkins, On the bursty evolution of blogspace, in: Proceedings of the 12th International Conference on World Wide Web, 2003, pp. 568–576.
[20] R. Kumar, J. Novak, P. Raghavan, A. Tomkins, Structure and evolution of blogspace, Communications of the ACM 47 (12) (2004) 35–39.
[21] M. Kwak, D.-S. Cho, Collaborative filtering with automatic rating for recommendation, in: Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on, 2001, pp. 625–628.
[22] S.K. Lee, Y.H. Cho, S.H. Kim, Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations, Information Sciences 180 (11) (2010) 2142–2155.
[23] D.-R. Liu, C.-H. Lai, W.-J. Lee, A hybrid of sequential rules and collaborative filtering for product recommendation, Information Sciences 179 (20) (2009) 3505–3519.
[24] K. Liu, W. Chen, J. Bu, C. Chen, L. Zhang, User modeling for recommendation in blogspace, in: IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology Workshops, 2007, pp. 79–82.
[25] D.-R. Liu, Y.-Y. Shih, Integrating AHP and data mining for product recommendation based on customer lifetime value, Information & Management 42 (3) (2005) 387–400.
[26] C. Mark, L. Phong, W. Makoto, B. David, Implicit interest indicators, in: Proceedings of the 6th International Conference on Intelligent User Interfaces, ACM, Santa Fe, New Mexico, United States, 2001.
[27] M. Masahiro, S. Yoichi, Information filtering based on user behavior analysis and best match text retrieval, in: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Springer-Verlag New York, Inc., Dublin, Ireland, 1994.
[28] Q. Mei, C. Liu, H. Su, C.X. Zhai, A probabilistic approach to spatiotemporal theme pattern mining on weblogs, in: Proceedings of the 15th International Conference on World Wide Web, ACM, Edinburgh, Scotland, 2006, pp. 533–542.
[29] B. Mobasher, R. Cooley, J. Srivastava, Automatic personalization based on Web usage mining, Communications of the ACM 43 (8) (2000) 142–151.
[30] D.C. Montgomery, L.A. Johnson, J.S. Gardiner, Forecasting and time series analysis, second ed., McGraw-Hill, 1990.

[31] S. Nakajima, J. Tatemura, Y. Hino, Y. Hara, K. Tanaka, Discovering important bloggers based on analyzing blog threads, in: Proceedings of the WWW '05 workshop on the weblogging ecosystem: aggregation, analysis and dynamics, 2005.
[32] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, GroupLens: an open architecture for collaborative filtering of netnews, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, ACM, Chapel Hill, North Carolina, United States, 1994, pp. 175–186.
[33] G. Salton, M.J. McGill, Introduction to modern information retrieval, McGraw-Hill, Inc., New York, NY, USA, 1983.
[34] B. Sarwar, G. Karypis, J. Konstan, J. Reidl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, ACM, Hong Kong, 2001.
[35] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Analysis of recommendation algorithms for e-commerce, in: Proceedings of the 2nd ACM Conference on Electronic Commerce, ACM, Minneapolis, Minnesota, United States, 2000, pp. 158–167.
[36] J.B. Schafer, J.A. Konstan, J. Riedl, E-Commerce recommendation applications, Data Mining and Knowledge Discovery 5 (1–2) (2001) 115–153.
[37] U. Shardanand, P. Maes, Social information filtering: algorithms for automating "word of mouth, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995, pp. 210–217.
[38] J. Shen, Y. Zhu, H. Zhang, C. Chen, R. Sun, F. Xu, A content-based algorithm for blog ranking, in: Proceedings of the 2008 International Conference on Internet Computing in Science and Engineering, IEEE Computer Society, 2008.
[39] J.-H. Su, B.-W. Wang, C.-Y. Hsiao, V.S. Tseng, Personalized rough-set-based recommendation by integrating multiple contents and collaborative information, Information Sciences 180 (1) (2010) 113–131.
[40] T.M. Tsai, C.C. Shih, S.C. Chou, Personalized blog recommendation using the value, semantic, and social model, in: Proceedings of Innovations in Information Technology, 2006, pp. 1–5.
[41] P.S. Yu, Data mining and personalization technologies, in: Proceedings of the 6th International Conference on Database Systems for Advanced Applications, 1999, pp. 6–13.