國立交通大學

資訊管理研究所

碩　士　論　文

行動應用之熱門部落格個人化推薦服務

Personalized Popular Blog Recommender Service for Mobile Applications

研 究 生： 蔡佩芸

指導教授： 劉敦仁 博士

中 華 民 國 九 十 八 年 六 月

行動應用之熱門部落格個人化推薦服務

Personalized Popular Blog Recommender Service for Mobile Applications

研 究 生：蔡佩芸　　Student : Pei-Yun Tsai

指導教授：劉敦仁　　Advisor: Duen-Ren Liu

A Thesis

Submitted to Institute of Information Management

College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science in Information Management

June 2009

Hsinchu, Taiwan, the Republic of China

中華民國九十八年六月

行動應用之熱門部落格個人化推薦服務

研究生： 蔡佩芸 **指導教授： 劉敦仁 博士**

國立交通大學資訊管理研究所

摘 要

部落格是一種網路寫作、出版的新模式，並可與其他部落格進行資料的串接、互動；由於操作使用簡單，任何人都可以在網路上發表言論。若能善用行動裝置特色，在手機上提供部落格文章推薦的加值服務，將是一個重要的應用，並能創造出更廣大的閱讀族群。因為設備的限制，使用者較難於行動裝置上隨意地瀏覽內容。面對為數眾多的部落格，「如何挑選出適合每位使用者的文章」是一個重要的議題。目前為止，較少研究針對行動裝置上的部落格內容個人化推薦服務做探討。

本研究提出一個行動裝置上的個人化內容服務的系統，為手機用戶挑選並推薦部落格文章。該系統可以由偵測網路上部落格內容的熱門趨勢，即時預測文章的熱門程度變化。再者，手機用戶有不同的喜好，因此需依據使用者個人偏好進行文章過濾；本研究所建置的系統由手機用戶的歷史行為紀錄進行分析，以找出該使用者的興趣特徵。最後，推薦使用者最新、熱門，且符合個人偏好的部落格文章。本研究實際將系統上線使用進行實驗評估，以驗證所提的系統方法確實可以提升行動裝置上的部落格文章點閱率。

關鍵字： 行動裝置服務、部落格推薦系統、時間敏感主題

Personalized Popular Blog Recommender Service for Mobile Applications

Student: Pei-Yun Tsai **Advisor: Duen-Ren Liu**

Institute of Information Management
National Chiao Tung University

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, to benefit from the proliferation and convenience of using mobile devices to receive information any time and anywhere. There is, however, 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 interests is an important issue. Very little research, however, focuses on this issue.

In this work, we propose a Customized Content Service on a mobile device (m-CCS) to filter and push blog articles to mobile users. The m-CCS can predict the latest popular blog topics by forecasting the trend of time-sensitive popularity of weblogs. Mobile users may, however, have different interests in the latest popular blog topics. Thus, the m-CCS further analyzes the mobile users' browsing logs to derive their interests, which are then combined with the latest popular blog topics to derive their preferred blog topics and articles. The experiment demonstrates that the m-CCS system can effectively recommend mobile users' desired blog articles with respect to both popularity and personal interests.

Keywords: Mobile service, Blog Recommender System, Time-Sensitive Topic.

致 謝

非常開心終於寫到致謝的部分了。雖然每天一直期盼著趕快完成論文，朝思暮想那張可愛的畢業證書，但在真正要離開的時候，卻又開始有點捨不得了。

首先最感謝的當然是指導老師劉敦仁教授，謝謝您終於放我走了(痛哭流涕中…)。謝謝劉老師對學生論文的指導，每逢週末都會抽空來學校與學生個別討論，留意並了解我們的進度與內容。除了課業上，老師也常帶我們出去玩、生日party、各種假借名義的慶祝活動、好樂迪唱歌；除了想盡辦法讓我們有個快樂的研究所生活，還要應付學生在生活上的瑣碎問題。跟老師相處，可以當他是一個像朋友般的爸爸，沒有教授的嚴肅架勢，還三不五時主動開玩笑，講一些自己覺得幽默的笑話。劉老師真的是我見過最盡職的指導老師與所長了。能如此順利畢業，更要很感謝口試委員魏志平老師、李永銘老師的細心審查，並在口試過程中給予指導與建議。

感謝所有交大資管所老師兩年來的指導。也謝謝大學時期的老師，感謝您們在過去的熱心教導，讓我有機會進入交大就讀，尤其吳帆老師、阮金聲老師、吳榮訓老師、熊博安老師、劉立頌老師，即使在我畢業後離開了中正大學，仍不斷給予學生關心與勉勵。

論文的完成，要感謝CAMEO公司提供實驗用的資料，愛芸、利同的全力協助與配合，讓實驗可以順利進行。更感謝泊寰學長的指導，除了論文之外，還教導我許多書本上所學不到的實務經驗，讓我受益良多。真的很幸運可以跟學長合作兩年。感謝實驗室仍在苦戰中的博班學長姐們：純和、志偉、韋孝、秀文、Hani，以及一起趕案子的宇軒大隊長，謝謝你在過去兩年的協助，一起留守實驗室看日出。謝謝錦慧學姐在我最後衝刺論文的幾個月，每天跟我一起在實驗室相依為命，妳是位耐心又溫柔的好學姐，我會努力幫妳找個好對象嫁掉的。祝福大家早日畢業，成為台灣最優秀的博士。

感謝實驗室同學：常常被我欺負的偉珍，謝謝妳總是保留乾淨的座位借我放東西；Lab大總管子瑋，雖然你總愛自以為是黑瑋帥；還有兩年內給與我無數協助的邱圈圈。謝謝已經畢業的學長姐：盈妹、佩君、鈺婷、振東、健誠，讓我從碩一就愛上這個Lab。碩一學弟妹：瓊瑤、雅婷、卉芳、其捷、榮笙，謝謝你們總是很搞笑的跟我們一起玩樂，讓Lab總是充滿歡笑，祝福你們明年一樣可以快快地順利畢業唷！另外，還要感謝傑克大叔以及France兄妹倆在論文期間的許多協助。大學至今同班六年的同學兼好友子鳳、建勳，共患難九年的安眉，謝謝你們這麼多年來對我的照顧與協助，一起彼此加油打氣。

感謝我親愛的孟哲阿爸、素珍阿母以及眾多親戚長輩們二十多年來的照顧，讓我無憂無慮快快樂樂的在交大上學讀書。我愛你們~

最後，謝謝所有資管所的成員(族繁不及備載)，是你們讓我的交大生活多采多姿！

好棒喔！真的寫完了，我想，連做夢都會偷笑了~

蔡佩芸

中華民國九十八年六月 謹於 交通大學

Lab剛換新冷氣，這裡實在是個舒服又令人流連的好地方

Index

List of Tables

List of Figures

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 population and react to current events (e.g., news) on the Internet [17]. Most people read blogs because it is a new source of news [24]. Looking for what is the latest popular issue discussed by blogs and attracting readers' attention is an interesting subject. Another issue is the lack of channel to receive blog information passively for user. This is a major disadvantage over traditional media, which broadcasts content right to your eyes over readily available channels (i.e. TV, newspapers, etc.). Moreover, providing value-added mobile services such as blog articles is increasingly important to attract mobile users to mobile commerce, 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 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, focuses on this issue.

There are mainly three types of research regarding blogs. The first type of research focuses on analyzing the link structure between blogs to form a community [21, 22]. Through the hyperlinks between blogs, people can communicate across blogs by publishing content relating 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 contents analysis to derive the propagation of topics and trends in blogsphere. Gruhl *et al.* [15, 16] modeled the information propagation of topics among blogs based on blog text. Mei *et al*. [27] proposed a method to discover the distributions and evolution patterns over time and space. With the analysis of tracking topic and user drift, Hayes *et al.* [17] examine the relationship between blogs over time. Most researches have not

considered how to predict the popularity degree of blog topics. The last type of research focuses on how to model the blogger and derive their interests for personal recommendation [20, 40]. A variety of methods has been proposed to model the blogger's interest, such as classifying articles into predefined categories to identify the author's preference [26]. Bloggers can receive the recommended content which is similar to their earlier experiences.

However, the majority of previous studies about blogs ignore the hot topics and popular articles discussed by the reader mass, who take browsing actions on the blog articles. Moreover, existing studies do not consider recommending blog articles to mobile readers in mobile environments. With more and more blog articles published on the Internet, the scale and complexity of blog contents are growing rapidly and result in information overload for bog readers. Mobile readers could only browse very few blog articles because of the restriction of mobile device. Accordingly, traditional recommendation methods, such as the collaborative filtering approach, 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 the trend of popularity. 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 the emerging blog topics. In addition, blog readers may have different interests in the emerging popular blog topics. Nevertheless, very few researches have addressed such issues.

In this work, we propose a Customized Content Service on a mobile device (m-CCS) to recommend blog articles to mobile users. The m-CCS can predict the trend of time-sensitive popularity of blogs. First, 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 similarity. By examining the clusters, we can extract the features of topics from the viewpoints of the authors. Moreover, we analyze the click times the readers give to articles. For each topic cluster, from the variation in trends of click times we can predict the popularity degree of the topics from the readers' perspectives.

Second, mobile users may have different interests in the latest popular blog topics. Thus, the m-CCS further analyzes mobile users' browsing logs to derive their interests, which are then used to infer their preferred popular blog topics and articles. We scrutinize the browsing behaviors and dissect the interests of the mobile users, then modify the ranking of topic clusters according to their preferences. Moreover, the m-CCS recommends blog articles by integrating personalized popularity of topic clusters, item-based CF and attention degree (click times) of blog articles. The filtered articles are then sent to the individual's mobile device immediately via a WAP Push service. This allows the user to receive personalized and relevant articles and satisfies the demand for instant information. Finally, the system of m-CCS demonstrates that the system can effectively recommend desirable blog articles to mobile users that satisfy popularity and personal interests.

We summarize the contributions of this paper as follows:

We propose a value-added mobile service to provide customized blog articles for mobile users, and the basic idea is to combine the estimated popularity of articles in the topic cluster and the predicted interest of the user in the articles. Without the effort of user rating, the implicit interest of customer in an article is inferred by comparing the time spent on reading the article with the average time spent on articles with the same size.

The proposed recommendation process mainly integrates contents analysis and collaborative filtering to improve the shortcoming of pure collaborative filtering such as sparsity and cold start problem, including aspects as (1) the prediction of popular topic cluster concerned by blogger 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.

In general, the effectiveness of CF recommendation approach mostly depends on the set of historical data. There is still potential limitation such as sparsity and cold start problem. It may deliver low-quality recommendation results when the system only has a few rating records of users, for measuring the similarity between users. For new items or new users, because of no active records viewed by users, the system will present weak performance in recommendation.

In our research, we focus on the mobile user and blog articles. For dimensionality reduction, we apply clustering techniques to group the data set first and then forming neighborhoods form the partitions, which can reduce the sparsity and improve scalability of recommender systems. Previous studies [4, 41] had also indicated the benefits of clustering application in recommender systems.

Additionally, many blog articles have not been viewed by any mobile user in our system due to the limitation of mobile device. It makes that most articles, which are popular on the Internet and the masses of Internet users pay attention to, may be ignored in the process of recommendation. Thus, we consider the activities of bloggers and the Internet readers as different viewpoints to identify the popularity of each article, so as to improve the recommender performance.

This study implements m-CCS which is suitable for thousands of real users in practice. To recommend the latest and the most popular blog articles instantly, the system needs to timely process the article contents and analyzes browsing behavior of thousands of online mobile users within two hours. Therefore, it must overcome the issue of efficiency and scalability. We not only adopt the load balancing architecture, but also carefully choose the algorithm and caching technology, in order to apply the system in a real business environment.

In the experiment, we compare different strategies: unified push of articles selected by experts and personalized push of articles selected by system recommendation service with m-CCS. The experiment result shows that the m-CCS can increase the click rate of blog articles to enhance customer satisfaction.

The remainder of this paper is organized as follows. Section 2 introduces the related works about blog, forecasting, and the recommendation. A brief introduction to our system is given in Section 3. The detailed descriptions about the processing module of our system are presented in Section 4 and Section 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. The conclusions and future work are finally made in Section 9.

2 Literature review

2.1 Blog

Blogging is a new method of publishing articles on the Internet. Due to its ease of use, anyone can publish and maintain a blog article on the Internet via a publishing software tool. Blog has emerged as an important type of web page with a set of dated entries, and is usually associated with profile of writers. In general, bloggers can freely voice their views on any subject of interest and share their knowledge with others. Therefore, blog content represents the opinions of the population and reacts to current events (e.g., news) on the Internet [17].

Under the notion of Web 2.0, in which everyone is encouraged to participate in public discussions, the process to gather different assemblies will thus, empower public attentions to off-stream occurrences. By definition and in practice, blogs have a distinction over traditional media, with even broader diversification as opposed to the relatively narrow view of the media, dictated by a handful of journalists and editors. Blogs have become such a force that mainstream media cannot help but take notice [12]. Moreover, one of the reasons for people to read blogs is that it is a new source of news [24].

The blog fever is accompanied by increasing interests from research and industrial communities to harness this important information source. There are three stems of research regarding blogs. The first type of research focuses on analyzing the link structure between blogs to form a community. The leading academic research [21, 22] on the weblog community

proposed a way to discover bursty evolution of blogspace by applying the hyperlink among blogs to cluster blogs, form communities, and inspect the changes of the communities. Subsequently, researches were introduced about the distribution of blogs over locations and how to form communities.. Through the hyperlinks between blogs, people can communicate across blogs by publishing content relating 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. Moreover, it has been argued, by Herring and others, that most blogs are less-connected as few bridging hyperlinks are available on the Internet [17]. For this reason, blog analysis may not perform well if they are viewed as typical web pages for page rank algorithms.

The second type of research focuses on analyzing blog content. Gruhl et al. [15, 16] modeled the information propagation of topics among blogs based on blog text. The patterns they proposed for topic propagation were useful to predict the ranks of sales forecasts. In addition, more and more researches pay attention to studies about blog content recently. Blog text analysis focuses on eliciting useful information from blog entry collections, and determining certain trends in the blogosphere. Natural Language Processing (NLP) algorithm has been used to determine the most important keywords within a definite time period, and it can automatically discover trends across blogs [12]. Nevertheless, above researches emphasized on assigning blog articles to only one topic, while blogs, in fact, contain many topics. Mei et al. [27] focus 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 over time and space. For tracking topic and user drift, Hayes et al. [17] examine the relationship between blogs over time . However, most researches have not considered how to predict the popularity degree of blog topics.

The last type is about user modeling and personal recommendation in blog space. A variety of methods [20, 40] has been proposed to model the blogger's interest, such as classifying articles into predefined categories to identify the author's preference [26], and thus to

automatically recommend the blog articles which are suitable for their interest by analyzing the contents which bloggers have acted on. Bloggers can receive the recommended content which is similar to their earlier experiences, but the methods ignore the hot topics and popular articles discussed by the reader mass that can attract mobile user's interest.

Nevertheless, the preceding studies, no matter what type of researches, were all observed from the viewpoints of bloggers. They mainly examined the interests of bloggers and identified which topics were widely discussed by the bloggers.

2.2 Forecasting

Forecasting [8] is to estimate what is going to happen. It mainly uses the history to reflect the developing statement in the future. There are mainly three types of forecasting approach. The fist is subjective judgments with experts which depend on the professional knowledge for a specific domain. The second method is to construct the relation models which mostly use explanatory variables to explain the predictor variable with inductive reasoning and hypothesis test. The third type is time series prediction. Time series is a set of observation value by time orders. The time series prediction build up a suitable model to forecast the future trend from the past observation value.

Within the variety of methods, Exponential smoothing method [7] is easy to understand and highly reliable, and this method can use few data to make a short term prediction. Exponential smoothing method assumes that there exists stability and regularity in the trend of time series which can be reasonably postponed with the drift. Since the latest trend will last to the nearest future in some level, we put the latest information in higher weight.

Exponential smoothing method has been widely used in the production forecast, and also used in the short term or medium term economic development trend forecasting. The exponential smoothing method gives historical data a dynamic weight fading by time to converge to zero.

2.2.1 Simple exponential smoothing method

In simple exponential smoothing method, to get the current prediction value, we weight the past values including the prediction value and the actual value belonging to the preceding time period. In Eq. (1), for preceding time series, $x(t)$ is the actual value at time t, and $\hat{x}(t)$ is the prediction value 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 $\hat{x}(t)$, weighted by α which is a smoothing constant. Therefore, the difference of smoothing constant would determine which parameter has more power of influence to affect the prediction value.

$$\hat{x}(t+1) = \alpha x(t) + (1-\alpha)\hat{x}(t) \tag{1}$$

Learning from the formula, each prediction value is weighted from the series value within past period. The more recent history data is. The more important weight of prediction is.

Here the smoothing constant can be decided in a subjective way or by minimizing ESS as Eq. (2) [9]. In usual, the smaller smoothing constant is suitable for the time series data which is change more violently, or we can say which is anomalous obviously. In contrast, the larger value of smoothing constant is suitable for the stable time series data.

$$ESS = \sum_{t=1}^{n} \left[x(t) - \hat{x}(t) \right]^2 \tag{2}$$

Simple exponential smoothing is suitable for stationary time series which don't have trend effect. Moreover, if there is a trend of time series, we can predict it by implementing double exponential smoothing method.

2.2.2 Exponential smoothing method with trend effect

In this section, we introduce the double exponential smoothing approach to process the time series data which has trend effect and is predicted as follow [10].

$$\hat{x}(t+1) = \alpha x(t) + (1-\alpha)\left[\hat{x}(t) + b(t) \right] \tag{3}$$

$$b(t) = \beta\left[\hat{x}(t) - \hat{x}(t-1)\right] + (1-\beta)b(t-1) \quad (4)$$

The basic concept is similar to the simple exponential smoothing method. But the most distinction between them is to consider the value of *b(t)*, which represents the trend effect at time *t* and is calculated as Eq. (4). Apply *b(t)* 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)*.

Using double exponential smoothing method in prediction, the value of $\hat{x}(t)$ and *b(1)* have to be assigned in the initial stage. The simplest way is to make a assumption for $\hat{x}(2) = x(1)$ and *b(1)=0* [13]. Some research has also suggested that the selection of initial value is not important toward the stationary [10], since it does not have a significant effect on the prediction result.

2.3 Recommendation

Due to the flourish development of the Internet, information grows and circulates very fast. For solving the problems of information overload, the recommender system is needed to provide suitable personalized information to the users according to their needs and preferences [28, 30, 37]. The recommender system has been highly used in many different areas [38], such as news [25], movie [32], book [14] and the music [39], and gives not only personalized recommendation service for each customer, but also brings the benefit to business marketing strategies.

Generally, the recommender system mainly has two types, including content-based filtering, collaborative filtering [1, 2].

2.3.1 Content-based recommendation

Content-based Recommendation approach analyzes the customers' preferences on item attribute feature to build up a personal feature profile, and then predict which items the customer will like [18, 42]. In other words, this approach is used to recommend items with similar attribute features to the customers according to their preference in the past. It is more

likely to be used for document recommendation, and also been used to recommend webpage and news articles. However, this method still has some restrictions to be improved, such as it is not easy to analyze the features of items, users can only receive the recommended items which are similar to the past [23].

2.3.2 Collaborative filtering recommendation

Collaborative filtering (CF) approach is one of the most popular recommending approach, and it has been successfully applied in many areas [3, 32]. This method can solve some problem of content-based method mentioned before. There is no need to analyze the contents of item; the recommended items are identified for a target user solely based on similarities of historical profile to other users. Furthermore, it can deal items with dissimilar content to those seen in the past.

Based on the relationship between items or users, CF method can be classified into two types [37], user-based CF and item-based CF. User-based CF is to calculate the similarity between users, and predict the target user preference toward different items. GroupLens is an example of such systems [32]. With the number of user and item being exploded, how to quickly produce high quality recommendation and search a large amount of potential neighbors in real time are important issues, especially for commercial systems. Considering from the aspect of items, item-based CF method has been proposed to identify the relationships between different items that users had already rated and then rank recommended items each user has not viewed before. The application has already been used on the Amazon platform [14] with a good performance.

3 System process overview

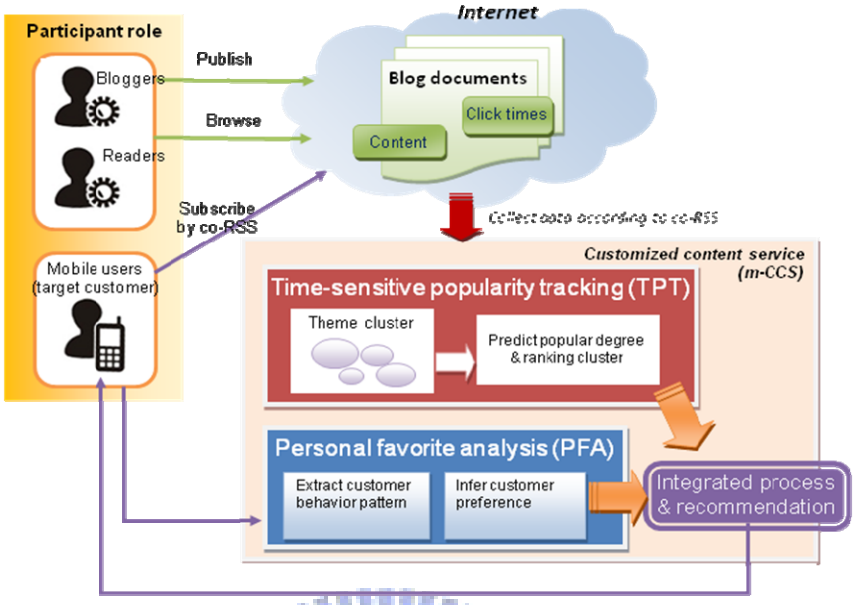

Fig. 1. System overview for m-CCS.

Because of their dramatic growth during recent years, blogs have become a dominant medium on the Internet. In order to provide blog contents to mobile users immediately, we propose a Customized Content Service on mobile (m-CCS) to combine two sources, time-sensitive popular topics and personal preference pattern. This is based on the idea of social navigation [11], i.e., everyone is interested in realizing what others like. Some research [34] has considered that observing the behavior of other users could create value for individuals, and it has already been applied extensively to large-scale websites. The simplest instance of social navigation is to forward the most popular items to readers such as New York Times, and the advanced technology is about complex recommender engines such as *Amazon.com* and *Epinion*.

In Fig. 1, the first step of our system is to get some blog articles from the Internet. The RSS mechanism is a useful way to capture the latest articles automatically without visiting each site. There is a shortage of information caused by insufficient RSS feeds subscribed to individuals, so we propose *co-RSS* method to solve this problem. The *co-RSS* process is similar to the concept of Web 2.0; anyone on the Internet can subscribe to RSS feeds. By gathering all feeds from users, RSS flocks called *crows-RSS* are formed to enrich information sources. After this

preliminary procedure, the system can automatically collect attractive contents from diverse resources.

We use information retrieval technology [35] to pre-process articles which are crawled everyday from blog websites according to *crows-RSS* feeds. After extracting features of blog articles by the process of *time-sensitive popularity tracking* (TPT), m-CCS groups articles into topic clusters and automatically traces the trend of popularity. Then, m-CCS can rapidly respond to the topics that most readers may be interested in.

Since the viewable content on mobile phone screens is limited, it is particularly desirable to design a personalized service for filtering article. The m-CCS can instantly monitor daily service transfer rates and log user viewing records to infer implicit preference of mobile users. The browsing records of users are analyzed to find behavior patterns and then the personal preferences are deducted through *personal favorite analysis* (PFA).

Finally, the system pushes the blog articles to the mobile users by integrating the above message sources, including the popular topics from the discussions of all blog articles, and the preference of readers. The m-CCS sorts out the most popular topics of the day and filters related articles based on users' implicit preference. The filtered articles are then sent to the user's mobile device via a WAP Push service. This allows users to receive personalized and relevant blog articles in real time.

Take the Olympic game, for example. The Olympics were held in Beijing in August 2008. They were the subject of international focus and discussion. Even someone with little interest in sports could not fail to keep an eye on the Olympic Games. Therefore, readers who do not possess clear preferences such as new users or occasional users are likely to receive contents of popular topics from the system. Conversely, to readers with distinct preferences, the system will grant higher weights to their preferences when the recommender operation is progressed; when the popularity degree of a topic is high enough, the system will also send related articles to the readers.

4 Time-sensitive popularity tracking module

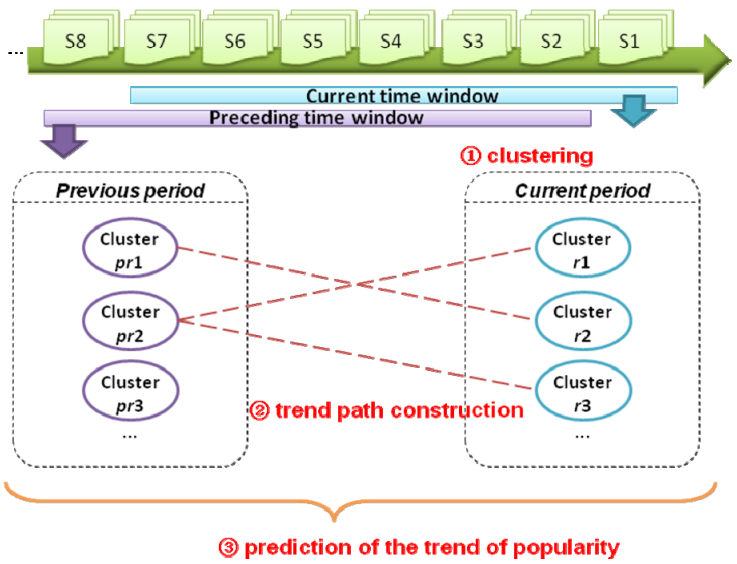

Fig. 2. Time-sensitive popularity tracking process.

We identify the blog topic clusters and their popularity according to the viewpoints of writers and readers on the Internet, and then trace the trend of popularity with time. Therefore, we can forecast the popularity degree for each topic cluster. In the following subsections, we will illustrate the details of the tracking process in Fig. 2.

4.1 Forming topic clusters of blog articles

We consider online blog diaries published and maintained by individuals which can represent personal experiences, thoughts and concerns. Therefore, we can obtain bloggers' intentions by analyzing their blog contents.

Articles in blogs are free and usually contain different opinions so that it is difficult to categorize articles into their appropriate categories which are defined by bloggers. That is to say, the existing category in a blog website is insufficient to represent the blog. In our research, we use article features, i.e., term-weight vector, derived from the pre-processing process to deal with blog articles which are published within a given time window on the Internet. 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 [19] is applied to implement the clustering step. It treats each article as a cluster first and then successively merges pairs of clusters. During the process, we calculate cluster distance to decide whether we should stop merging or not, and the similarities between two articles can be calculated by means of the cosine similarity measure, as shown in Eq. (5).

$$sim(d_m, d_n) = \cos(\vec{d}_m, \vec{d}_n) = \frac{\vec{d}_m \bullet \vec{d}_n}{\|\vec{d}_m\| \cdot \|\vec{d}_n\|} \quad (5)$$

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 similarity between the target cluster *r* and all the clusters *pr* belonging to the preceding period, and then select the one with max values to construct the link with one of the preceding clusters.

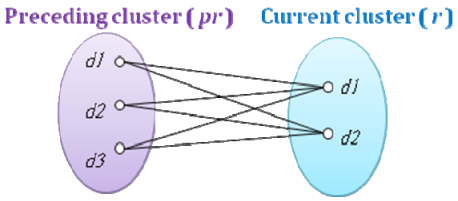

Fig. 3. Calculation of the similarity between two clusters.

As blog articles are usually composed of unstructured words, to obtain similarity between two clusters which belong to two days we average the value of cosine similarity between articles crossing clusters. Then we can identify the differences between two clusters clearly.

In Fig. 3, the similarity between two clusters (r, pr) in adjacent days is calculated by Eq. (6), where d_i / d_j is a blog article belonging to the set of blog articles S_r / S_{pr} in cluster *r/pr*; | S_r |/| S_{pr} | is the number of blog articles of S_r / S_{pr} and $Sim(d_i,d_j)$ denotes the cosine similarity between the articles d_i and d_j, as mentioned in Section 3.1.

$$similarity(r, pr) = \frac{\sum_{d_i \in S_r} \sum_{d_j \in S_{pr}} Sim(d_i, d_j)}{|S_r||S_{pr}|} \tag{6}$$

After the establishment of linkages, the trend of each current cluster can be derived from the preceding related cluster. Phenomenally, it may happen that there is no out-link for a preceding cluster; this means that the popularity of the topic in the cluster decreases gradually and almost vanishes.

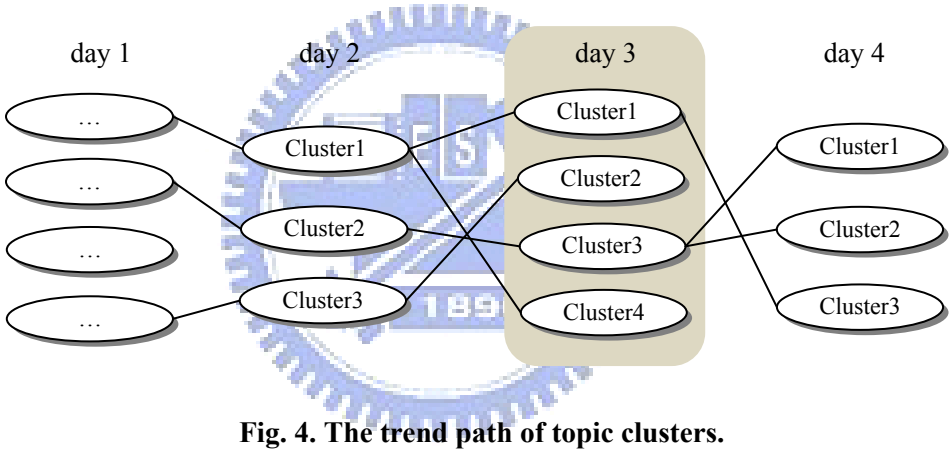

Fig. 4. The trend path of topic clusters.

We take *day 3* in Fig. 4 as an example to explain the trace of trend. There is no outbound linking starting at *cluster 4*, which indicates that very few blog articles in *day 4* contain similar topic of *cluster 4* in *day 3*. The popularity degree of the topic that *cluster 4* presents has become much weaker than before, because there are not enough articles to support the topic.

Nevertheless, one path must connect with the current cluster. Every topic has its own initial state which implies a previous related topic. As mentioned above in Fig. 4, all 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 we can use the relationship and similarity between them to calculate the popularity degree, and so on.

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 attention from readers, 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 attention from readers who make a topic rising and flourishing. After clustering blog articles to form 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 in previous period. For each preceding cluster *pr*, we obtain the total click times of the articles, $CT_t(S_{pr})$, on the Internet within preceding time period, as defined in Eq. (7).

$$CT_t(S_{pr}) = \sum_{d_i \in S_{pr}} ClickTimes_t(d_i) \tag{7}$$

where, S_{pr} denotes the set of blog articles d_i in cluster *pr*, and the actual click times for blog article d_i in time *t* can be represented by $ClickTimes_t(d_i)$ $d_i \in S_{pr}$. Then, the click times can be transferred to actual popularity degree, $APD_{pr}(t)$, which is a normalized value based on the maximum *ClickTimes* over all S_i in the preceding period, as follows.

$$APD_{pr}(t) = \frac{CT_t(S_{pr})}{Max\{ClickTimes_t(S_i)\}} \times 100\% \tag{8}$$

4.4 Predicting popularity degree of current cluster

We analyze the trend evolution of attention from readers to predict the popularity degree of current cluster. The time series is a set of serial observation values by time order as in Fig. 5. Forecasting mainly uses the history to respond to the development trend in the future. A standard exponential smoothing method [29] assigns exponentially decreasing weights to the previous observations. In other words, recent observations are given relatively more weight in forecasting than the older observations. Unlike traditional predictive models, it can estimate

time series with a small amount of data. We modified the *double exponential smoothing* method [5] to forecast the degree of popular trend for each cluster of blog topic.

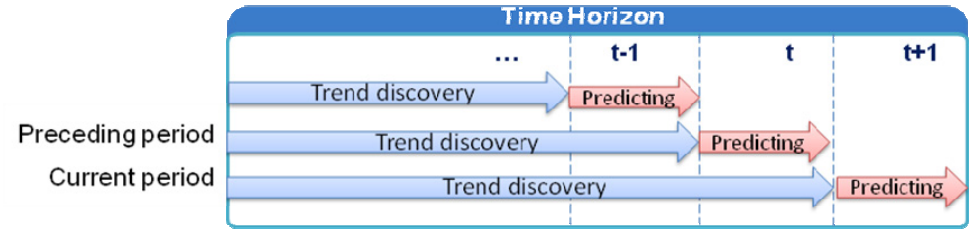

Fig. 5. The time series of popularity trend.

Here are the two equations associated with *double exponential smoothing*. 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. (9).

$$PPD'_r(t+1) = \alpha \times APD_{pr}(t) + (1-\alpha) \times \left[PPD_{pr}(t) + b_{pr}(t) \right] \tag{9}$$

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 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 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* using the weighted average.

$$b_{pr}(t) = \delta \times \left[PPD_{pr}(t) - PPD_{ppr}(t-1) \right] + (1-\delta) \times b_{ppr}(t-1) \tag{10}$$

Note that the cluster *pr* is the preceding cluster of *r*, while the cluster *ppr* is the preceding cluster of *pr*. The $PPD_{ppr}(t\text{-}1)$ and $b_{ppr}(t\text{-}1)$ 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 Eq. (9) and Eq. (10) respectively can be decided by experts or experimental analysis.

In the prediction model, we adopt a double exponential smoothing approach [10] to predict the topic popularity. The double exponential smoothing approach 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 as a topic trend path with a popularity degree. Therefore, to make a link between topic clusters, the maximal similarity between adjacent clusters, i.e., current cluster *r* and preceding cluster *pr*, as described in Section 4.2, is selected to adjust the predictive popularity degree of cluster *r*. Notably, the smaller similarity leads to the lower reliability of the prediction path between two clusters.

$$PPD_r(t+1) = PPD'_r(t+1) \times sim(r, pr) \tag{11}$$

In Fig. 6, 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 *cluster11*, which belongs to *Time t*, to predict the popularity degree of *cluster22* in *Time t+1*. In the same way, *cluster01* is useful to infer *cluster22*. In the initial stage, the supposed value of actual popularity degree for *cluster01* is 40%. Previous research has suggested that it is reasonable to assume $PPD'_r(t) = APD_{pr}(t\text{-}1)$ and $b_{pr}(t\text{-}1)=0$, at the starting time *0*, so the initial predictive popularity degree of *cluster11* could be reasoned and the value is 40%. Likewise, we also assume the value of trend effect for *cluster01* is zero. Then the similarity across adjacent clusters should be considered to calculate the predictive popularity degree. Suppose the value of similarity between *cluster01*

and *cluster11* is 0.23; we can obtain the predictive popularity degree of *cluster11* after adjustment as $40\% \times 0.23 = 9.2\%$.

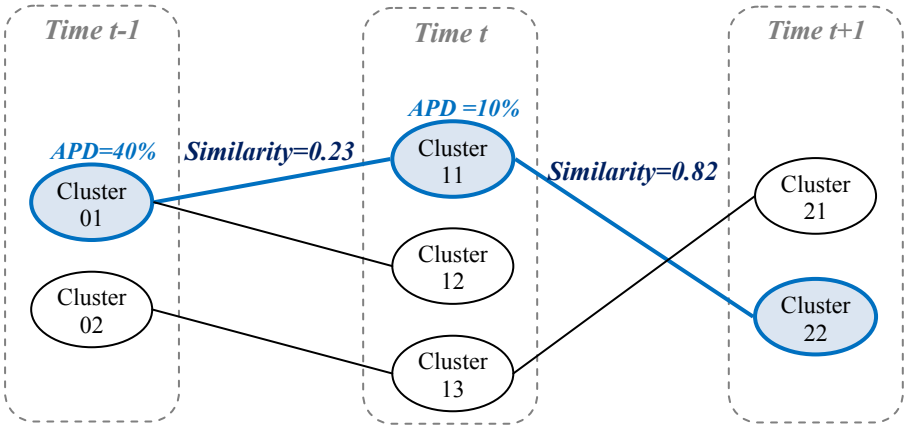

Fig. 6. The time series of topic clusters.

Next, we use the values which were derived previously to predict the popularity degree of *cluster22* in Eq.(9) before adjusting for path-link similarity.

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

According to the double exponential smoothing method mentioned, the value of trend effect, $b_{cluster11}(t)$, is derived in Eq. (10).

$$b_{Cluster11}(t) = 0.3 \times \left[PPD_{Cluster11}(t) - PPD_{Cluster01}(t\text{-}1) \right] + 0.7 \times b_{Cluster01}(t\text{-}1)$$

$$b_{Cluster11}(t) = 0.3 \times \left[9.2\% - 0 \right] + 0.7 \times 0 = 2.76\%$$

And then Eq. (9) is as follows,

$$PPD'_{Cluster22}(t+1) = 0.3 \times 10\% + 0.7 \times \left[9.2\% + 2.76\% \right] = 11.37\%$$

Taking similarity into account is the last step.

$$PPD_{Cluster22}(t+1) = PPD'_{Cluster22}(t+1) \times Similarity(Cluster11, Cluster22)$$

In Fig. 6, the value of similarity between *cluster11* and *cluster22* is 0.82. We obtain the final predictive popularity degree as follows.

$$PPD_{Cluster22}(t+1) = 11.37\% \times 0.82 = 9.32\%$$

4.5 Normalizing predictive popularity degree of clusters

We have taken into account different aspects from bloggers who post the articles and readers who click blogs to take browsing action on the Internet. In this step, to predict the popularity degree of each topic cluster, we have clustered similar topics based on the feature of content and analyzed the trend of the click ratio in different topics.

We normalize the value of predictive popularity degree above, and arrange the cluster in percentage order. The formula is as below:

$$NPPD_r(t) = \frac{PPD'_r(t)}{\max\left\{PPD'_i(t)\right\}} \tag{12}$$

5 Personal favorite analysis module

Modeling user preference is useful for many personalized services, such as recommender systems [33]. In this paper, we propose a novel scheme to model the interests of users who browse blog articles in mobile devices.

This study extracts a set of keywords to represent blog articles; then a cluster of articles which contain keyword sets is regarded as a topic. In addition, a blog article is regarded as a preference unit. Because of the limited features of mobile devices, it is inconvenient to give explicit relevance ratings of blog articles for mobile users. We analyze browsing behavior to get individuals' favorites and thus do not require any extra user effort. First, we analyze the browsing pattern of each user to calculate user preference scores for blog articles which have been read by mobile users.

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 give any request in a time period. Fig. 7 shows that the time period between *r3* and *r4* exceeds the length of the timeout interval, so it is separated into different sessions. Calculating the time interval between user requests within each session could estimate the user's stick time of the article. In session 1, the stick time of *r1* is the difference between *r1* and *r2*, and the stick time of *r2* is the difference between *r2* and *r3*.

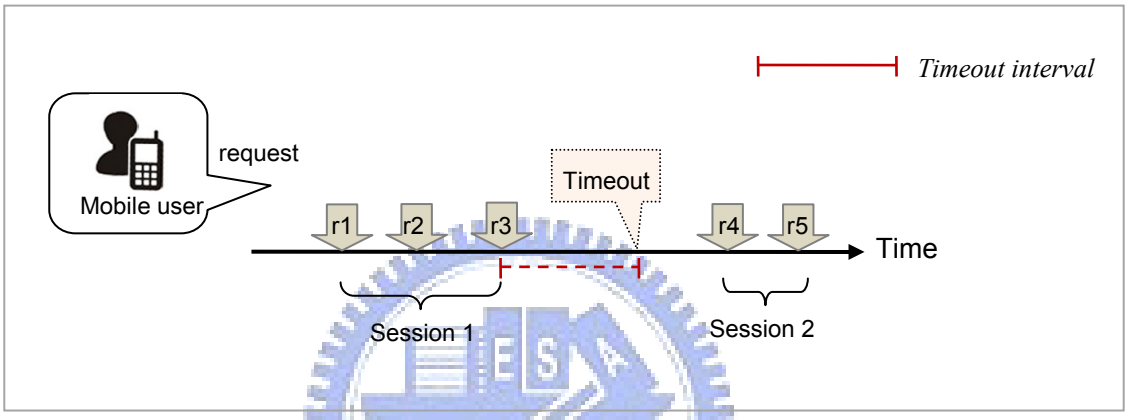

Fig. 7. User behavior for requesting articles.

In order to acquire the browsing pattern for the user u, we analyze the browsing speed, $H_{u,s}$, to get the average reading 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 a article i that the user had browsed within session s; $D_{u,s}$ is a set of artcles browsed by user u in session s; and $|D_{u,s}|$ denotes the number of articles in $D_{u,s}$. $DocSize(d_i)$ identifies the size number of words of the article; $Time_u(d_i)$ denotes the browsing time for the user u in 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 using 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 greater effect upon the future behavior of the mobile user, so we set the parameter β to give recent pattern 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. By analyzing a user reading time on an article, we can infer how interested the user is in the article and its corresponding preference score. If the stick time is longer than usual, we can estimate that the user has a high preference level for the article. That is, the user is addicted to the article and is willing to spend more time on it.

According to the user's reading behavior in usual time, we use 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 denoted by $PBT_u(d_i) = DocSize(d_i) \times H'_{u,s+1}$, where $DocSize(d_i)$ is the size of blog article d_i and $H'_{u,s+1}$ means the browsing pattern for user u as mentioned in Section 5.1. Then, we calculate the *preference score* (PS) for target user u in 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), and the higher value of preference score means that the user has more interest in the article.

6 Integrated process and recommendation module

In this section, we propose a hybrid method, which combines user preference prediction of collaborative filtering, attention degree for each article on the Internet, and customized popularity degree of cluster to recommend personalized blog contents for mobile users.

The basic idea of this process is to integrate the different viewpoints of mobile customers and Internet users. A collaborative filtering approach considers actual browsing behavior of mobile users as implicit preference in order to recommend the latent articles of interest. Because of the limitation of the device, however, the mobile user cannot easily surf any content, and a lot of articles have never been browsed on mobiles. Therefore, it is necessary for our proposed system automatically to pursue those articles to which most users have paid close attention on the Internet.

Finally, the predictive popularity degree of topic clusters that each article belongs to is taken into account. The more popular the topic is, the more users have an interest in it. We now give full details of procedure.

6.1 Topic-based collaborative filtering

In the collaborative filtering (CF) method, computing similarity is the most important process, and it is usually calculated on the whole set of items. But the fact is that there are a lot of differences between items which belong to disparate clusters. That is to say, items in different clusters are often at variance with the characteristic of user preference since the diversity between items becomes higher. Therefore, it does not necessarily reflect the appropriate similarity, and users who have similar preference in one category of items may turn out to have totally different judgment when it comes to another kind of item. For example, Alice and Bob have expressed a similar preference for a topic about movies, but we cannot directly affirm they are also interested in others.

The clustering algorithm mentioned before is based on the threshold of similarity to generate partitions of varying size. Clustering techniques work by identifying groups of articles which

appear to have similar topics. In our research, within each topic cluster we apply the CF method to predict the latent preference article for mobile users. Moreover, with the increase of blog articles and mobile users, how to raise efficiency of computing is the primary consideration for commercial systems. The item-based CF approach has proved that it can quickly produce high-quality recommendation.

The function of item-based collaborative filtering can be defined as follow. First, in Eq. (16), we use adjusted cosine [36] to measure the similarity between two items, d_i and d_j, which belong to cluster r. Here 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.

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

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). d_j is the nearest neighbors of the target article d_i. $\hat{I}$ denotes the set of N articles that are most similar to the target article and have been browsed by user u.

$$PS_u^{cf}(d_i)=\frac{\sum_{j\in\hat{I}}PS_u(d_j)\bullet sim_r^{adj}(d_i,d_j)}{\sum_{j\in\hat{I}}|sim_r^{adj}(d_i,d_j)|} \tag{17}$$

After the previous step, we obtained the predictive preference scores by using a collaborative filtering algorithm in each cluster for mobile users.

6.2 The degree of attention for each blog article

Except for the prediction scores of mobile users derived from the collaborative filtering method, the viewpoint of Internet users is important. According to the theory of social

navigation mentioned before, looking for the kind of articles the majority of people pay attention to is an available reference.

Within every topic cluster, we obtain the attention degree for each article with the accumulated click times that represent how much attention people pay. It is defined as follows.

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

Here we apply the *click-through rate* to derive the attention degree, and it is calculated as $ACCT(d_i)/Max_{d_j \in D_r}\{ACCT(d_j)\}$, where $ACCT(d_i)$ denotes the accumulated click times for article, d_i, and $Max_{d_j \in D_r}\{ACCT(d_j)\}$ means the most accumulated click times of articles in the cluster *r*.

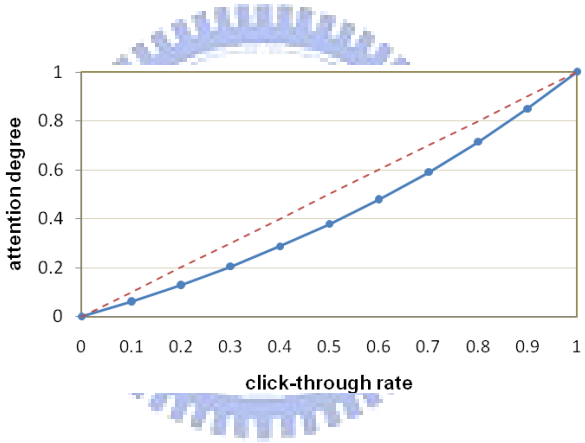

Fig. 8. Exponential changes in the variation of attention degree.

We suppose that the attention degree has the property of network externality. As increasing of click-through rate, one article would be more attractive to mobile user. Fig. 8 shows that the value of attention degree, $attention_r(d_i)$, rises exponentially 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 provide equal ranking score of topic cluster for every mobile user. Next,

we consider each user's preference in a *personal favorite analysis* (PFA) step to modify the order of topic cluster and derive the personal ranking cluster.

The different topic clusters belonging to successive days may be covered by the same articles. Once a mobile user has read one article, his/her preference score is inferred from browsing behavior and can be applied to modify the ranking score of the topic cluster in which the article is included. Therefore, cluster ranking can be modified 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 as follows:

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

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

In the formula above, we use the average *preference score* (PS) of user *u* for those articles that have been read 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.*

As regards those mobile users who have expressed their own preferences significantly in our system, the ranking of topic clusters provided by m-CCS would be more customized. The system would assign more weight on personal characteristics of users who have presented an amount of historical behavior records and give less weight on the general popularity degree of topic clusters. On the contrary, 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 first 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* ($\overline{PS}_{u,r}^{adjusted}$) for each topic cluster as in Eq. (21). Here we derive the *adjusted average preference score*, $\overline{PS}_{u,r}^{adjusted}$, as shown in Eq. (22). The weight value, $1\text{-}\omega_{u,r}$ shown 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, 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.

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

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

Here we illustrate the effect of weight value, $1\text{-}\omega_{u,r}$, on deriving the customized predictive popularity degree. Since users have different characteristics of interest for each topic cluster, there is discrepancy in the value of $\omega_{u,r}$ suggested in Eq. (20). For example, Alice, Bob and Charlie have browsed blog articles which belong to the same topic cluster *x*, and their preference score for each article is derived as in Table.1. Alice has read ten articles; the average preference score is (0.7+0.7+0.7+0.7+0.7+0.7+0.7+0.7+0.7+0.7)/10=0.7. Bob has only browsed two articles, d_2 and d_6, and the average preference score is (0.7+0.7)/2=0.7. For Charlie, the average preference score is (0.7+0.7+0.6+0.6)/4=0.65. But it is not enough to represent user preference for the cluster by using average preference score. The number of articles which have been browsed should be taken into account. Generally, once mobile users are willing to click on more articles, it means they have more interest in this topic.

Table.1. User-browsed articles and the preference score.

Preference score	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
Alice	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Bob	--	0.7	--	--	--	0.7	--	--	--	--
Charlie	--	0.7	0.7	0.6	0.6	--	--	--	--	--

Therefore, for Alice, Bob and Charlie, the *adjusted average preference score* ($\overline{PS}_{u,r}^{adjusted}$) can then be calculated as follows, and the number of articles that users have browsed would influence the difference in the results.

$$\omega_{Alice,x} = \frac{1}{10} = 0.1$$

$$\overline{PS}_{Alice,x}^{adjusted} = (1-0.1)\times 0.7 = 0.63$$

$$\omega_{Bob,x} = \frac{1}{2} = 0.5$$

$$\overline{PS}_{Bob,x}^{adjusted} = (1-0.5)\times 0.7 = 0.35$$

$$\omega_{Charlie,x} = \frac{1}{4} = 0.25$$

$$\overline{PS}_{Charlie,x}^{adjusted} = (1-0.25)\times 0.65 = 0.49$$

Fig. 9 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. 9, 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 personal preference of historical records would become more important to affect the value of customized predictive popularity degree of topic cluster. Moreover, the weight value, 1-$\omega_{u,r}$, increases rapidly for smaller number of browsed articles, while the curve trends to the flat for larger number of browsed articles. We consider that user preference appears significant in the beginning of browsing behavior.

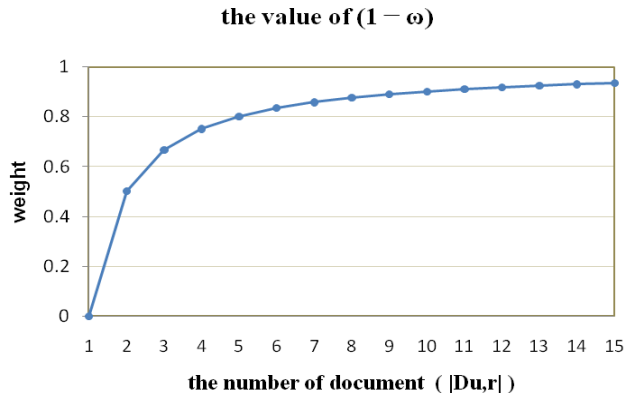

Fig. 9. The value of *1-ω* with different number of browsed articles.

6.4 Content selection and recommendation

In Sections 6.1 and 6.2, we obtained the predictive preference score by collaborative filtering and the attention degree on the Internet for articles within every cluster. Then, for mobile users, Section 6.3 identified the WCPPD/ HCPPD method to determine customized predictive popularity degree in topic clusters. Here we will propose a model to integrate the previous processes to recommend the latent interest articles for mobile users.

We derive the predictive preference score of articles as a hybrid of $PS_{u,r}^{cf}(d_i)$, the predictive preference score by collaborative filtering and $attention_r(d_i)$, the attention degree on the Internet for articles within cluster, and it can be expressed as in Eq. (23).

For every mobile user u, we suggest the value of τ_u, which denotes the relative importance of $PS_{u,r}^{cf}(d_i)$ and $attention_r(d_i)$, is defined as in Eq. (24), where $\left|D_{u,t}^{push}\right|$ is the number of articles which have been pushed to user u within time period t and $\left|D_{u,t}^{browsed}\right|$ denotes how many articles the user has browsed. The more articles a user has browsed, the more personal interest is emphasized, which means that the history records of the mobile user are sufficient and their preference is clear. 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 few records of browsing articles exist to infer a mobile user's preference.

$$PS_{u,r}^{predict}(d_i) = \tau_u \times PS_{u,r}^{cf}(d_i) + (1-\tau_u) \times attention_r(d_i) \quad (23)$$

$$\tau_u = \log_2\left(\frac{\left|D_{u,t}^{browsed}\right|}{\left|D_{u,t}^{push}\right|} + 1\right) \quad (24)$$

The purpose of computing $PS_{u,r}^{predict}(d_i)$ is to infer the user preference in terms of the information collected so far within topic clusters, and it is adjusted according to the clarity of personal preference which is composed of historical behavior. With the clear preference, $PS_{u,r}^{predict}(d_i)$ is more influenced by $PS_{u,r}^{cf}(d_i)$. $PS_{u,r}^{predict}(d_i)$ is, however, dominated by $attention_r(d_i)$ for the users who have an unclear preference.

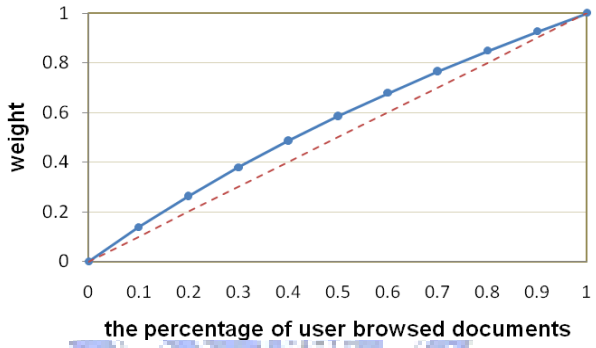

Fig. 10. Weight values in different percentages of user-browsed articles.

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. 10. When the characteristic of user preference is insufficient, τ_u becomes zero, where $PS_{u,r}^{cf}(d_i)$ is ignored and the final preference is decided using only $attention_r(d_i)$. In contrast, with τ_u approaching the maximum value one, $PS_{u,r}^{predict}(d_i)$ would be decided by $PS_{u,r}^{cf}(d_i)$ entirely. The upward curve is slightly convex. That is to say, the value of weight increases rapidly for smaller percentage of browsed articles, while the curve trends to the flat for larger percentage of browsed articles. We consider that user preference appears significant in the beginning of browsing behavior, which also represents the user's greater degree of satisfaction with the recommended contents.

So far, we have generated the predictive article preference within clusters. To select the recommended articles from different clusters, we have to consider the different priority (ranking) of topic clusters. In Section 6.3, for every user u, we have derived the customized predictive popularity degree, $CPPD_{u,r}$, of cluster to denote personalized ranking of topic clusters. Here we apply $CPPD_{u,r}$ to adjust the user latent interest for articles cross topic cluster, as in Eq. (25). Finally, the article with high estimated preference will be selected for mobile users.

$$PS_u^{select}(d_i) = PS_{u,r}^{predict}(d_i) \times CPPD_{u,r} \tag{25}$$

After the processing above, the picked articles are transformed into XHTML for mobiles and then pushed to handsets via WAP. Because the resolution of handsets allows only scroll browsing and user time is short compared with that of PC users, the system will only push the titles of no more than ten articles (see Fig. 11). Then users can click the title in which they are interested to view full contents.

Fig. 11. m-CCS interface to display recommended contents on mobile device.

7 System architecture

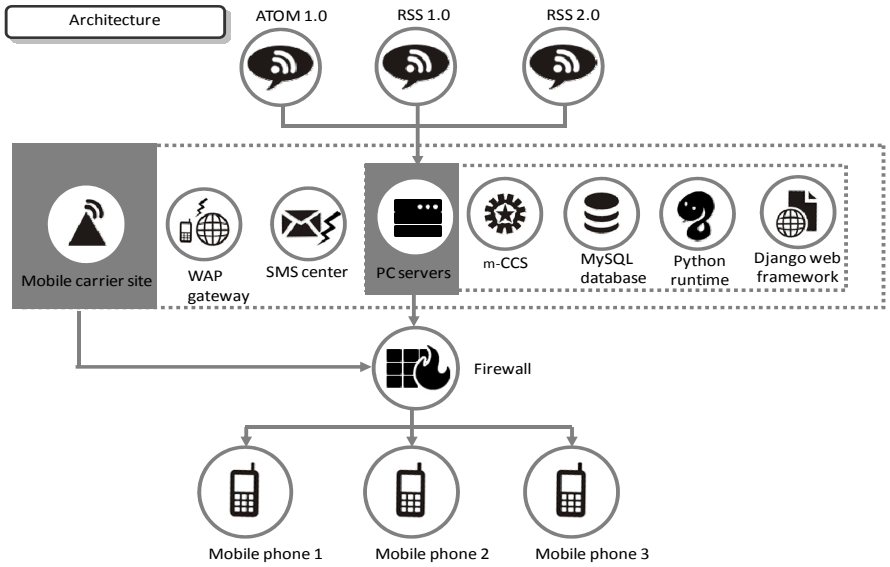

Fig. 12. The system architecture of m-CCS.

This research is conducted in collaboration with CAMEO InfoTech Inc., provider of the WAP Push service for mobile phones. Currently, there are 200 000 users of Chunghwa Telecom (CHT), the biggest telecom company in Taiwan, with more than 4000 users officially subscribing to the service and paying a regular fee.

We are implementing a m-CCS system of the proposed mobile service based on the CHT mobile customer database stored in the MySQL database; it is developed using the Python programming language. The operating web GUI uses a Django Web Framework (see Fig. 12).

The m-CCS module has to be set up in the machine room of CHT located at Taiwan, and provide the computation of blog recommendation. Adopt two IBM 1U servers to proceed the load balance computing and provide the browsing service for mobile phone.

On the mobile carrier site, WAP Gateway is built in the machine room of system operator. With the WAP Gateway, it can lower the flow of wireless transmission by encoding the mobile WAP page which contains message and URL. And then, transform the WAP Push message to the SMS format of GSM and dispatch the message to the mobile through SMSC, which is

device belonging to system operator. Therefore, the words and URL links could be showed on the mobile phone.

Finally, taking advantage of m-CCS, we look forward to improving customers' satisfaction by providing customized blog contents.

8 Applications and experimental evaluation

We have implemented a commercial system m-CCS to push blog articles to mobile users. In this section, we evaluate the effectiveness of our proposed time-sensitive popularity tracking module and personalized recommendation service in Section 8.1 and Section 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 process the latest data from Internet every day. System robots automatically crawl the net for the newest blog articles according to co-RSS feeds in real time. Since RSS is a well-structured format, it's easy to detect new posts. When there is a new post, the system will trigger the process of capturing articles. However, RSS usually contains only partial information of the articles. In order to get the whole content, m-CCS needs to capture the primitive HTML through the URL of the blog.

Because the HTML format may contain not only irrelevant HTML tags and Java Script language but also the advertisement of websites, the redundant content will be deleted through the String process algorithm after capturing the article from the web page. Furthermore, since the layout format of each website is different, we need to parse HTML to get the article title,

content, and publish time from a variety website. Finally, the well-structured data are stored into database, and the fields of articles are listed in Table.2.

Table.2. The field and description of blog article stored in database

	Field	Description
1	strId	Article identification
2	datetimePublish	The date of blog article to be published and clawed from web site
3	strTitle	The title of blog article which is headlined by blogger
4	strPlainText	The plain text of blog article before preprocessing
5	strUrl	the url link of this blog articles on the Internet

The total number of new published articles collected from co-RSS feeds is around two thousands daily. To make the popularity prediction, it is necessary to fetch the daily click times of captured articles within time window from Internet. We can only choose the blog sites providing information about click-through 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), are selected to conduct our experiment. 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 time window were processed to predict popularity degree of topic clusters. About one thousand articles were chosen for analysis. The data set with click times of articles is collected form blog websites during two-week period starting from 10 May 2009. In the topic clustering phase, we set a threshold value 0.002 as the condition to stop grouping articles.

To evaluate the prediction model, the mean absolute error (MAE) [6] 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 a set of topic clusters derived at time *t* and $N\left(S^t\right)$ denotes the number of topic clusters. The larger

the MAE, the more error the prediction model will be. Thus, a model which presents a lower MAE can be regarded as a better model.

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

8.1.2 Evaluation result

The experiment was conducted for two weeks. During the period from 10 May to 24 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 evolved.

This section presents the experimental result of prediction models based on different weight settings of parameters. As mentioned in Section 4, there are two smoothing constants, α and δ, which are used to weight the diverse status mentioned in Eq. (9) and Eq. (10), respectively. The parameter α is used to determine the relative importance of the actual popularity degree and the predictive popularity degree with trend effect in preceding period. The parameter δ is used to adjust the relative importance of the difference of the predictive popularity degrees at time *t* and at time *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 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. Table.3 presents the MAE under various values of parameter α. The result shows that generally, the value of MAE decreases with time no matter what the value of α is. In addition, the table also shows the average of MAE under different values of α, and the result is also plotted in Fig. 13. The prediction model has the lowest MAE under ***α=0.7***. It means that predicting the popularity degree of topic cluster can be more accurate when the system put more weight on the preceding actual popularity degree.

Table.3. The MAE for the sets of cluster $S_{t,r}$ with different values of α (δ = 0.5)

α =	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
May 12	0.1724	0.1724	0.1724	0.1726	0.1724	0.1726	0.1724	0.1726	0.1724	0.1724	0.1724
May 14	0.1844	0.1715	0.1350	0.1355	0.1350	0.1112	0.1195	0.0960	0.1126	0.1108	0.1130
May 16	0.1567	0.1434	0.1318	0.1312	0.1318	0.1256	0.1360	0.1285	0.1477	0.1565	0.1671
May 18	0.1501	0.1265	0.0912	0.1005	0.0912	0.0828	0.0751	0.0703	0.0604	0.0547	0.0544
May 20	0.1639	0.1333	0.0935	0.0971	0.0935	0.0863	0.0878	0.0866	0.0886	0.0899	0.0916
May 22	0.1578	0.1283	0.0993	0.1057	0.0993	0.0974	0.0936	0.0939	0.0905	0.0900	0.0898
May 24	0.1445	0.1148	0.0815	0.0944	0.0815	0.0852	0.0819	0.0867	0.0888	0.0935	0.0979
Average	**0.1614**	**0.1414**	**0.1150**	**0.1196**	**0.1150**	**0.1087**	**0.1095**	**0.1050**	**0.1087**	**0.1097**	**0.1123**

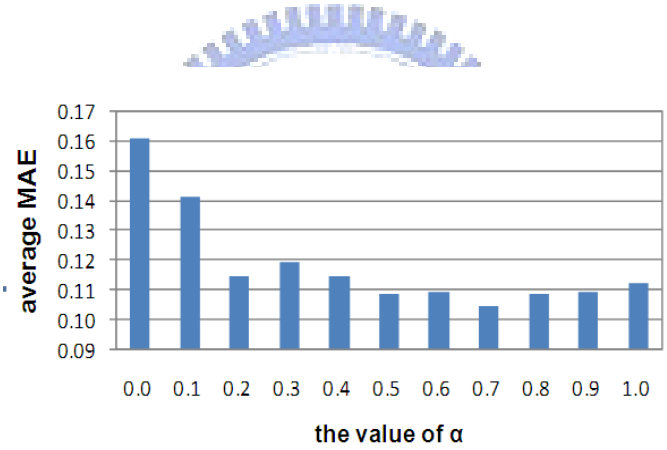

Fig. 13. The performances of the compared smoothing constant under different α (δ = 0.5)

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 corresponding MAE is shown in Table.4 and Table.5. The averages of MAE are also plotted in Fig. 14 and Fig. 15. 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 ***δ =0.8***, and it implies that the difference 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 ***α=0.7*** is better than the accuracy under ***α=0.5***.

Based on the above findings, the best parameter settings to predict the popularity degree of topic cluster are ***α=0.7*** and ***δ =0.8***, and such parameter settings are used in the rest of our experiments.

Table.4. The MAE for the sets of cluster $S_{t,r}$ with different values of δ (α = 0.5)

δ =	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
May 12	0.1724	0.1724	0.1724	0.1724	0.1724	0.1724	0.1724	0.1724	0.1726	0.1724	0.1724
May 14	0.1249	0.1249	0.1249	0.1249	0.1249	0.1249	0.1249	0.1249	0.1112	0.1249	0.1249
May 16	0.1330	0.1330	0.1330	0.1330	0.1330	0.1330	0.1330	0.1330	0.1256	0.1330	0.1330
May 18	0.0830	0.0830	0.0830	0.0830	0.0830	0.0830	0.0830	0.0830	0.0828	0.0830	0.0830
May 20	0.0892	0.0892	0.0892	0.0892	0.0892	0.0892	0.0892	0.0892	0.0863	0.0892	0.0892
May 22	0.0958	0.0958	0.0958	0.0958	0.0958	0.0958	0.0958	0.0958	0.0974	0.0958	0.0958
May 24	0.0802	0.0802	0.0802	0.0802	0.0802	0.0802	0.0802	0.0802	0.0852	0.0802	0.0802
Average	**0.1112**	**0.1112**	**0.1112**	**0.1112**	**0.1112**	**0.1112**	**0.1112**	**0.1112**	**0.1087**	**0.1112**	**0.1112**

Table.5. The MAE for the sets of cluster $S_{t,r}$ with different values of δ (α = 0.7)

δ =	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
May 12	0.1724	0.1724	0.1724	0.1724	0.1724	0.1726	0.1724	0.1724	0.172648	0.1724	0.1724
May 14	0.1152	0.1152	0.1152	0.1152	0.1152	0.0960	0.1152	0.1152	0.096043	0.1152	0.1152
May 16	0.1409	0.1409	0.1409	0.1409	0.1409	0.1285	0.1409	0.1409	0.128533	0.1409	0.1409
May 18	0.0677	0.0677	0.0677	0.0677	0.0677	0.0703	0.0677	0.0677	0.070275	0.0677	0.0677
May 20	0.0877	0.0877	0.0877	0.0877	0.0877	0.0866	0.0877	0.0877	0.086607	0.0877	0.0877
May 22	0.0918	0.0918	0.0918	0.0918	0.0918	0.0939	0.0918	0.0918	0.093863	0.0918	0.0918
May 24	0.0847	0.0847	0.0847	0.0847	0.0847	0.0867	0.0847	0.0847	0.086734	0.0847	0.0847
Average	**0.1086**	**0.1086**	**0.1086**	**0.1086**	**0.1086**	**0.1050**	**0.1086**	**0.1086**	**0.1050**	**0.1086**	**0.1086**

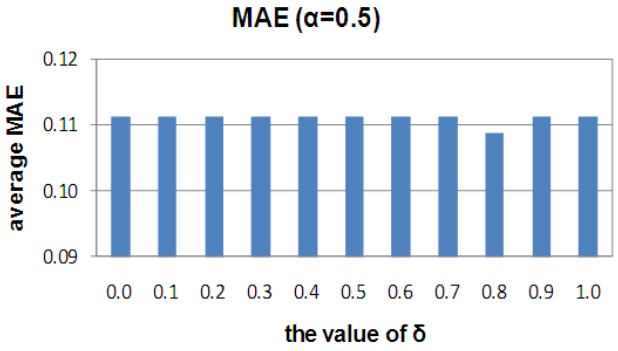

Fig. 14. The average MAE for the sets of cluster $S_{t,r}$ with different values of δ (α = 0.5)

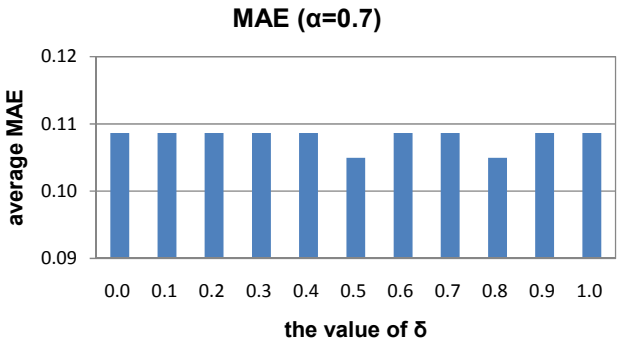

Fig. 15. The average MAE for the sets of cluster St,r with different values of δ (α = 0.7)

8.2 Evaluation of recommending blog articles

Chunghwa Telecom in Taiwan has officially provided the service of browsing blog contents on the mobile phone for eight months. All the blog articles have been selected by the human experts, and then sent to all customers without considering users' personalized preferences. Even if the experts continuously adjust the strategy of content selection to satisfy the preference of most users, the click rates for articles on mobile phone are still very low. Therefore, how to raise the click rates of customers is the major objective for our research.

A personalized popular blog recommender service has been implemented in such real business environment to increase the click rates for blog articles and improve the satisfaction of mobile customers. Through the proposed m-CCS mechanism, we can predict the popularity

degrees of topic clusters, and further analyze user preferences for personalized article recommendation. In this section, we conduct experiments to evaluate our proposed approach.

8.2.1 Data sets

We use a real-world data set to examine the performance of the proposed approach, and the experiments are conducted in an online business environment. Mobile users use the blog-service provided by Chunghwa Telecom for free 30 days trial, and then if they feel satisfied, they can become formal paid subscribers to enjoy of the use of this service. Currently, there are 18136 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 periods ended.

Within the latest one month, there are 4,104 articles published on those four Internet blog websites mentioned above. Each mobile user has on average browsed only 27.93 articles, i.e., 0.68% articles published on Internet. According to this observation, the amount of articles browsed by mobile users is lower than that of Internet users because of the limitation of 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 select 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 is 257 times in one month; the lowest is 12 times. Fig. 16 illustrated 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, 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.

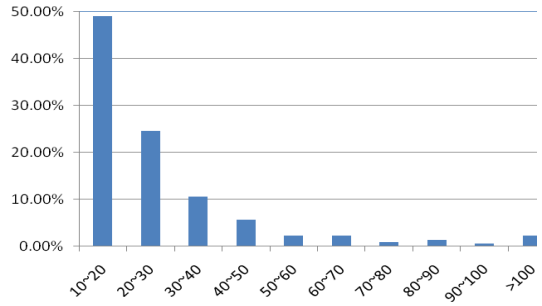

Fig. 16. The distribution of click counts for testing users.

8.2.2 Design of the experiments

The item-based CF method has been adopted to predict the preference scores of articles based on the article-similarity analyzed from the browsing log of mobile users. However, as mentioned in previous section, most blog articles published on Internet are seldom read by mobile users due to the limitation of mobile environments. 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 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 recommendation of blog articles. With this approach, mobile users can timely receive the latest articles of hot topic by mobile device anytime and anyplace.

There are several factors to 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. Trough the experiments, we will discuss the issues listed below.

- Does the system method with 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 on different approaches for deriving the customized predictive popularity degree of topic cluster?

To experimentally verify the effectiveness of our proposed methods, we compare different recommendation approaches shown in Fig. 17. The *expert method* selected articles by human expert 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.

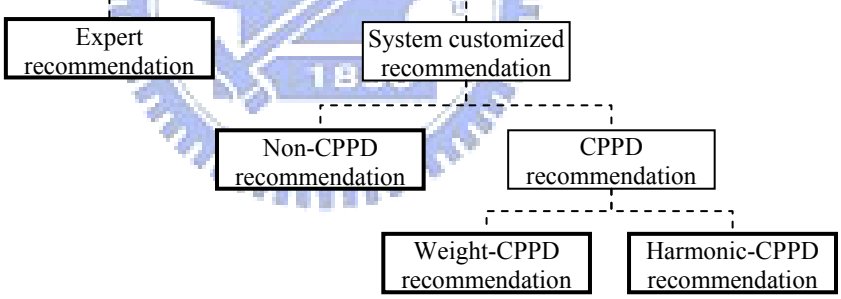

Fig. 17. The classification of recommendation methods

The *non-CPPD* method uses formula similar to the 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 CPDD and which cluster the article belong to. As mentioned in Section 6.3, there are two approaches to derive CPPD, one is the weighted method, and the other is 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.

The impacts of those methods on recommendation effectiveness are investigated in this experiment. We compare different recommendation models and evaluate their recommendation quality in Section 8.2.3. Note that the parameters, α and δ, had been experimentally determined in previous experiment and set as α**=0.7** and δ **=0.8**.

Moreover, the m-CCS system not only recommends articles fitting 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 are 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 for comparing the effect of system methods on recommending new articles. We will 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 conduct on-line experiments by recommending blog articles to 300 testing users selected as described in Section 8.2.1. To avoid disturbing customers, we cannot send the recommended articles to users every day; instead, the frequency of on-line recommendation is 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 limitation on small screen of mobile device.

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 message merely within 100 Chinese

characters at first, but 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). And then, if mobile users have interest in those articles, they can request the full text to browse the details of articles.

Any activity of customers through mobile phone will be logged, including the phone number, mobile phone type and what time to request which contents, etc. This information is recorded in database for further analysis.

We employ the hit ratio to evaluate the experiment results. The hit ratio, presented in Eq. (27), is defined as the ratio of the size of clicked set to the size of recommended set, where the size of clicked set is the number of articles clicked (requested) by testing users in our experiment, and the size of recommended set is the number of articles recommended to all target testing users. The higher hit ratio implies higher recommendation quality. For each run of experiment, there are 300 testing users and all of them receive 10 recommended articles, so that the size of recommended set is 3000.

$$\text{Hit ratio} = \frac{\text{size of clicked set}}{\text{size of recommend set}} \tag{27}$$

The number of clicked articles and the hit ratio for each method are listed in Table. 6. We can observe that the hit ratio of expert method is the lowest; the system methods perform better than the expert method. The *non-CPPD* method performs worse than the CPPD-based methods including the *Weighted-CPPD* and *Harmonic-CPPD* methods. Thus, considering 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 testing user. 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.

Table. 6. 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 %

For system methods, the customized articles would be recommended to satisfy individual interest. 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. The customized recommendation delivered by the m-CCS system is helpful for mobile business to enhance customer satisfactions.

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 mobile users the latest and diversified articles that may satisfy their interests. This section presents the experiment result for comparing the effect of system methods on recommending new articles.

We select 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 prediction without considering CPPD.

Table. 7. The hit ratio in different type of articles for non-CPPD and weighted-CPPD recommendation 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%

Table. 8. The hit ratio in different type of articles for non-CPPD and harmonic-CPPD recommendation 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%

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. 7, while the comparison of the *non-CPPD* method with the *harmonic-CPPD* method is listed Table. 8. 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. 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

Our study suggests a new value-added service for mobile phone applications. Dispatches of customized blog articles not only enable mobile phone users to enjoy reading blog articles without time and venue limitations but also provide a proactive channel to deliver the blog contents instead of passive browses from the users.

Owing to their dramatic growth in recent years, blogs have become a dominant medium on the Internet. Although RSS feed is a common solution to receive the latest contents automatically, there are still some problems, including lack of subscriptions and overloaded information. For that reason, filtering the articles so they are suitable for each user is an important issue. In the past, studies about blogs centered on bloggers (the authors) but ignored the views of mass readers. Our study applies the co-RSS method based on the notion under Web 2.0. Our proposed m-CCS can predict the trend of time-sensitive popularity of blogs. The m-CCS is grounded on the topic clusters of the blog articles which represent the perspectives of the authors. The m-CCS also considers the aspects of the readers' click rates to trace the popularity trends of the topic clusters from the affluent blog contents. Moreover, as regards mobile phone users, the m-CCS will analyze their browsing behaviors and personal preferences to recommend their preferred popular blog topics and articles.

Our experiment evaluations show that 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 user.

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 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; 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, and thus we should put more weight in 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, 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.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. on Knowl. and Data Eng.,* vol. 17, pp. 734-749, 2005.

[2] M. Balabanovi and Y. Shoham, "Fab: content-based, collaborative recommendation," *Commun. ACM,* vol. 40, pp. 66-72, 1997.

[3] D. Billsus and 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.

[4] A. Borchers, D. Leppik, J. Konstan, and J. Riedl, "Partitioning in Recommender Systems," University of Minnesota - Computer Science and Engineering Technical Report 1998.

[5] B. L. Bowerman and T. O. C. Richard, *Forecasting and time series: an applied approach*: Duxbury Press, 1993.

[6] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence,* pp. 43-52, 1998.

[7] R. G. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series*: Dover Publications.

[8] C. Chatfield, "What is the 'best' method of forecasting?," *Journal of Applied Statistics,* vol. 15, pp. 19-38, 1988.

[9] C. Chatfield, A. B. Koehler, J. K. Ord, and R. D. Snyder, "A new look at models for exponential smoothing," *The Statistician in press,* 2001.

[10] C. Chatfield and M. Yar, "Holt-Winters forecasting: some practical issues," *The Statistician,* vol. 37, pp. 129-140, 1988.

[11] A. Dieberger, "Supporting social navigation on the World Wide Web," *International Journal of Human-Computers Studies,* vol. 46, pp. 805-825, 1997.

[12] N. Glance, M. Hurst, and T. Tomokiyo, "BlogPulse: Automated trend discovery for weblogs," in *WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2004.

[13] C. W. J. Granger and P. Newbold, "Forecasting Economic Time Series," *New York: Academic Press,* 1986.

[14] Greg Linden, Brent Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing,* pp. 76-80, 2003.

[15] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins, "The predictive power of online chatter," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, Chicago, Illinois, USA 2005, pp. 78-87.

[16] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, "Information diffusion through blogspace," in *Proceedings of the 13th International Conference on World Wide Web*, 2004, pp. 491-501.

[17] C. Hayes, P. Avesani, and 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.

[18] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," in *Proceedings of the SIGCHI conference on Human factors in computing systems* Denver, Colorado, United States: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 194-201.

[19] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys,* vol. 31, pp. 264–323, 1999.

[20] S. Jie, Z. Yan, Z. Hui, C. Chen, S. Rongshuang, and X. Fayan, "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.

[21] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, "On the bursty evolution of blogspace," in *Proceedings of the 12th International Conference on World Wide Web*, 2003, pp. 568-576.

[22] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, "Structure and evolution of blogspace," *Communications of the ACM,* vol. 47, pp. 35-39, 2004.

[23] M. Kwak and 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

[24] J. D. Lasica, "Weblogs: A new source of information," *We've got blog: How weblogs are changing our culture, John Rodzvilla (ed). Perseus Publishing, Cambridge, MA,* 2002.

[25] H. J. Lee and S. J. Park, "MONERS: A news recommender for the mobile web," *Expert Systems with Applications,* vol. 32, pp. 143-150, 2007.

[26] K. Liu, W. Chen, J. Bu, C. Chen, and 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.

[27] Q. Mei, C. Liu, H. Su, and C. X. Zhai, "A probabilistic approach to spatiotemporal theme pattern mining on weblogs," in *Proceedings of the 15th international conference on World Wide Web*, Edinburgh, Scotland, 2006, pp. 533 - 542.

[28] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic personalization based on Web usage mining," *Commun. ACM,* vol. 43, pp. 142-151, 2000.

[29] D. C. Montgomery, L. A. Johnson, and J. S. Gardiner, *Forecasting and time series analysis*, 2nd ed.: McGraw-Hill, 1990.

[30] M. D. Mulvenna, S. S. Anand, and A. G. Büchner, "Personalization on the Net using Web mining: introduction," *Commun. ACM,* vol. 43, pp. 122-125, 2000.

[31] S. Nakajima, J. Tatemura, Y. Hino, Y. Hara, and 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, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work* Chapel Hill, North Carolina, United States: ACM, 1994, pp. 175-186.

[33] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM,* vol. 40, pp. 56-58, 1997.

[34] L. Rosenfeld and P. Morville, *Information architecture for the world wide web*, 3rd ed.: O'Reilly & Associates, Inc. Sebastopol, CA, USA, 2007.

[35] G. Salton and M. J. McGill, *Introduction to modern information retrieval*: McGraw-Hill, Inc. New York, NY, USA, 1983.

[36] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web* Hong Kong, Hong Kong: ACM, 2001.

[37] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proceedings of the 2nd ACM conference on Electronic commerce* Minneapolis, Minnesota, United States: ACM, 2000, pp. 158-167.

[38] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-Commerce Recommendation Applications," *Data Min. Knowl. Discov.,* vol. 5, pp. 115-153, 2001.

[39] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating "word of mouth"," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1995, pp. 210-217.

[40] T. M. Tsai, C. C. Shih, and S. C. Chou, "Personalized Blog Recommendation Using the Value, Semantic, and Social Model," in *Innovations in Information Technology, 2006*, 2006, pp. 1-5.

[41] L. H. Ungar and D. P. Foster, "Clustering Methods for Collaborative Filtering," *Proceedings of the Workshop on Recommendation Systems at the 15th National Conference on Artificial Intelligence,* pp. 112-125, 1998.

[42] P. S. Yu, "Data mining and personalization technologies," in *Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference on*, 1999, pp. 6-13.