

國立交通大學

資訊管理研究所

碩士論文

結合聲望與內容式過濾之書籤網站
部落格文章推薦

Combining Reputation and Content-Based Filtering for Blog Article
Recommendation in Social Bookmarking Websites

研究生：彭其捷

指導教授：劉敦仁 博士

中華民國九十九年七月

結合聲望與內容式過濾之書籤網站部落格文章推薦

Combining Reputation and Content-Based Filtering for Blog Article

Recommendation in Social Bookmarking Websites

研究生：彭其捷

Student: Chi-Chieh Peng

指導教授：劉敦仁

Advisor: Duen-Ren Liu



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

July 2010

Hsinchu, Taiwan, the Republic of China

中華民國九十九年七月

結合聲望與內容式過濾之書籤網站部落格文章推薦

研究生：彭其捷

指導教授：劉敦仁 博士

國立交通大學資訊管理研究所

摘要

Web2.0 是一個新興的網路社群，提供一個平台讓網友互動、管理與分享資訊，像是部落格文章、書籤網站、網路影片、書評、產品意見等。書籤網站提供讓網友發表自己的文章或是推薦別人文章的功能，讓大家能更方便搜尋與分享熱門的文章或是自己感興趣的文章。但是隨著網路快速的發展，過多的網路訊息造成資訊過載的問題。以書籤網站為例，即使已經過濾處理，但每天還是有大量的文章推薦至書籤網站，而無法順利的消化如此龐大的資訊量。本研究提出了以文章熱門度為基礎，整合使用者聲望與內容式過濾之個人化部落格文章推薦方法，透過分析使用者過去的文章推薦情形，進一步推薦使用者感興趣的文章。實驗結果顯示本研究所提出的方法比傳統方法能更有效的針對使用者的興趣來推薦適合的部落格文章。

關鍵字：Web 2.0、部落格、社交書籤網站、推薦系統、聲望、內容導向式過濾

Combining Reputation and Content-Based Filtering for Blog Article

Recommendation in Social Bookmarking Websites

Student : Chi-Chieh Peng

Advisor : Duen-Ren Liu

Institute of Information Management

National Chiao Tung University

ABSTRACT

The new generation of web-based communities, Web2.0, represents an innovative spirit in sharing and managing contents. Social bookmarking is a portal for users to share, organize, search, and manage bookmarks of web resources. However, with the rapid growth of web documents that are produced every day, people are facing the problem of information overload. The Social bookmarking web site provides the push (user recommendation) counts of articles indicating the recommended popularity degrees of articles. Thus, users can refer the push counts to find popular and interesting articles. Popularity based solely on push counts, however, cannot truly reflect the trend of popularity. In this paper, we propose to derive the popularity degree of an article by considering the reputation of users that push the article. Moreover, we propose a personalized blog article recommendation approach, which combines the reputation-based popularity with content based filtering, to recommend popular blog articles to users that satisfy their personal preferences. Our experimental results show that the proposed approach outperforms conventional approaches.

Keywords: Web2.0, blog, Social bookmarking, Recommender System, Reputation,

Content-based filtering

誌謝

終於阿，終於等到可以寫誌謝的這一天，不想寫的太過矯情，因此我想到什麼，就寫什麼！這兩年來深刻的體悟就是，研究的路真的不好走，一步一步踏實地走下去，才可以積沙成塔，才能蓋出完整的研究理論，進而找到更好的解決問題的方法！而這一路上真的要感謝很多人的幫助。

首先最感謝的當然是我們實驗室的大家長，劉敦仁老師啦，感謝你花那麼多時間忍耐我的破英文寫作，還有糾正我許多因為粗心大意所犯的錯誤，幫助我寫出更高品質的論文，甚至讓我成功的完成了出國發表的目標，還得到了論文比賽的全國優勝，這一切真的多虧了您對我論文的指導，您真的是我研究上的啟蒙恩師！

第二必須要感謝的就是大頭，這一年來的陪伴，教會了我許多人生的態度與方法，而你也總在我辛苦的時候扶我一把，在我苦悶的時候帶來我歡笑，希望我們能夠永遠就這樣快樂下去。

再來是親愛的 DBIS 同學們，小黑、雅婷、卉芳，兩年的研究生活中因為與你們在一起，真的多了很多歡笑，而且你們也總在我需要幫助的時間給予我關心，希望我們畢業後都能順順利利的。其他人像是錦慧、宇軒、Hani、Linda、瓊瑤、韋孝、純和學長、志偉學長、秀玉學姐、蔡佩芸、邱璇、偉珍、瑋哥、文婷、國瑄、采馨、正和、珮榕等等，都讓我碩班的時光變得如此多采多姿，誠懇的祝福你們未來都能一帆風順。

同時也要感謝交大羽球隊的支持，因為你們願意讓我同時兼顧研究與練球，讓我學會如何在團體生活與自己的研究上面找到平衡點與時間分配的能力，同時也認識了很多可以一起打拼的戰友，謝謝你們。

其他還有許多想要感謝的人，但是真的是要感謝的人太多了，不如謝天吧，謝謝所有幫助我一路走來完成論文的所有貴人！

最後，僅將這份榮譽與我的家人共享，因為有你們的支持，我才能順利的完成研究所的學業，謝謝！

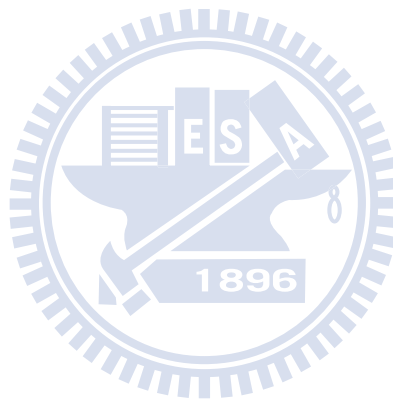
Index

| | |
|---|----------|
| 摘要..... | ii |
| ABSTRACT..... | iii |
| 誌謝..... | iv |
| Index..... | v |
| List of Tables | vii |
| List of Figures..... | viii |
| | |
| 1. INTRODUCTION..... | 1 |
| | |
| 2. RELATED WORK..... | 5 |
| 2.1 Web2.0..... | 5 |
| 2.1.1 Blogs | 5 |
| 2.1.2 Social Bookmarking..... | 6 |
| 2.2 Recommender Systems..... | 7 |
| 2.2.1 Content-based Filtering..... | 7 |
| 2.2.2 Collaborative Filtering | 8 |
| 2.2.3 Reputation System | 8 |
| | |
| 3. REPUTATION-BASED APPROACH..... | 9 |
| 3.1 Overview..... | 9 |
| 3.2 Data Preprocessing | 11 |
| 3.2.1 TFIDF..... | 11 |
| 3.2.2 Article Clustering | 12 |
| 3.2.3 User Grouping..... | 13 |
| 3.3 Reputation Analysis | 14 |
| 3.4 Content-based Filtering with Reputation Popularity..... | 15 |
| 3.4.1 Reputation-based article popularity..... | 15 |

| | | |
|-----------|---|-----------|
| 3.4.2 | Reputation-based Group Preference Score..... | 16 |
| 3.4.3 | Predict Preference Score | 17 |
| 3.5 | Hybrid Recommendation | 19 |
| 4. | EXPERIMENT EVALUATION | 21 |
| 4.1 | funP Company | 21 |
| 4.2 | Experiment Design | 21 |
| 4.2.1 | Data Set..... | 22 |
| 4.2.2 | Evaluation Metrics | 25 |
| 4.2.3 | Methods compared in this experiment | 26 |
| 4.3 | Experimental results | 28 |
| 4.3.1 | Comparison of the weighting methods in calculating reputation | 28 |
| 4.3.2 | Evaluation of time factor..... | 29 |
| 4.3.3 | Evaluation of Reputation Analysis..... | 30 |
| 4.3.4 | Comparison of CBF and ICF | 32 |
| 4.3.5 | Comparison of the weighting methods in our hybrid method..... | 33 |
| 4.3.6 | Comparison of our proposed approach | 34 |
| 4.3.7 | Comparison of all methods | 35 |
| 5. | CONCLUSIONS | 36 |
| | REFERENCES..... | 37 |

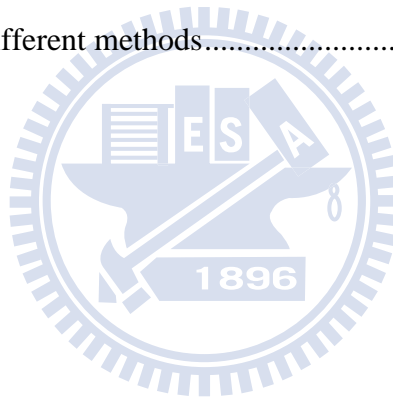
List of Tables

| | |
|--|----|
| Table 1 The statistics of the articles..... | 23 |
| Table 2 The number of interest about article categories..... | 24 |
| Table 3 The statistics of users' push-count..... | 25 |



List of Figures

| | |
|---|----|
| Fig. 1 Overview of proposed recommendation system | 9 |
| Fig. 2 Overview of CBFRRGP | 18 |
| Fig. 3 The value precision under different β | 28 |
| Fig. 4 Average push count | 29 |
| Fig. 5 The F1-metric under different number of top N articles | 30 |
| Fig. 6 The F1-metric between RGPS and PRankRGPS | 31 |
| Fig. 7 The F1 of ICF, CBF, ICFRRGP, and CBFRRGP method | 32 |
| Fig. 8 The value precision under different α | 33 |
| Fig. 9 The F1 of CBF, ICF, RGPS, CBFRRGP, and Hybrid method | 34 |
| Fig. 10 The comparison of different methods | 35 |



1. INTRODUCTION

Web2.0 represents a new generation of web-based communities for internet innovation. Through Web2.0, information is delivered more collaboratively in a social-related way [34]. Web2.0 services, including social bookmarking, social tagging, blog, and Wikipedia, are valuable collections of human knowledge that are created by users in a collaborative manner. For example, people can share their daily lives in Blogger, chat little things in Facebook, search collaborative editing knowledge in Wikipedia, and tag funny pictures in Flickr, etc. In other words, Web 2.0 social relationships bring us a new way of sharing.

Blog is a web page that serves as a publicly accessible note for an individual or a group of people. With the rapid growth of bloggers and blog articles, the vast amount of information brings the phenomenon of information overload [16]. As a result, it is an ideal place to provide recommendation service in the blog platform, especially for the purpose of finding valuable blog articles.

Social bookmarking provides the service of article recommendation for popular blogs based on the number of people that like the blog articles. funP (<http://funp.com/>) is a popular social bookmarking Web site in Taiwan. This site enables users to discover and share contents from blogs on the web. Users can share their own blog articles or recommend other people's articles. The ease of sharing for mass users in social bookmarking site, results an overwhelming amount of articles, making the selection of desirable articles increasingly difficult for users.

A recommender system is a solution to the problem of information overload [2]. Recommender systems are widely used to provide suitable personalized information to

users according to their preferences [13]. Generally, recommender systems mainly include Content-based filtering (CBF) and Collaborative filtering (CF) approaches. The CBF approach analyzes the users' preferences on the attribute features of item to build up a personal feature profile, and then predict which items the user prefer. The CF exploits historical data expressing preferences to form user neighbors or item neighbors, and makes recommendations based on those similar users' opinions or similar items [6].

Moreover, reputation systems have been integrated with recommender systems to enhance recommendation quality [22]. Reputation systems generally analyze user interactions to derive the reputation scores of users from his/her past behaviors [28]. There are two categories of reputation systems, one computes the reputation scores based on users' past ratings on items [23], whereas in social bookmarking site, there are no ratings. The other category of reputation systems considers human relationships to derive user preferences by presuming a user's preference similar to his/her friends' preferences [21]. However, human relationships need to be explicitly specified and are difficult to obtain.

In a social bookmarking web site, the users usually have two roles. The first role, namely the publishers can publish and push (recommend) their own articles or other users' articles to the web site. The second role, namely the followers who also like the published articles, can push the published articles to express their recommendation. In this paper, we use the push-follower relationships to form a reputation network and derive the reputation scores of users. Generally, a user with more followers will have higher reputation scores. The web site provides the push counts of articles indicating the recommended popularity degrees of articles. Thus, users can refer the push counts to find

popular and interesting articles. Popularity based solely on push counts, however, cannot truly reflect the trend of popularity. The articles pushed by highly reputed users are more likely to become popular than those articles pushed by users with lower reputation scores. Thus, we propose to derive the popularity degree of an article by considering the reputation scores of users that push the article. In addition, users may have different interests in the emerging popular articles. Accordingly, we propose a personalized blog article recommendation approach, which combines the reputation-based popularity with content based filtering, to recommend desirable articles to users that satisfy popularity and personal interests.

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 [19]. Bloggers can receive the recommended content which is similar to their earlier experiences.

Although existing researches have proposed content-based filtering or collaborative filtering approaches to recommend desirable blog articles that satisfy user preferences, they did not address the issue of recommending personalized popular articles in social bookmarking web sites. Existing recommendation approaches did not consider the recommended popularity degrees of articles, and did not investigate the issue of deriving recommended popularity degrees of articles by considering user reputations. Accordingly, our approach combines reputation-based popularity with the content-based filtering to enhance the quality of recommending personalized and popular blog articles. Our experimental results show that the proposed approach outperforms conventional approaches.

The remainder of this paper is organized as follows: Session 2 introduces the related works about web2.0 and recommender systems. Our proposed method is given in Section 3. Section 4 shows our experiment results. The conclusions are finally described in Section 5.



2. RELATED WORK

2.1 Web2.0

Web2.0 is a web technology that facilitates information delivery through a collaborative and social-related manner. Web2.0 created several new business models [24], such as social bookmarking, social tagging, and blogs. In this research, we implement personalized blog article recommendations in a social bookmarking website.

2.1.1 Blogs

Nowadays, blogs have already become a social media for people to express themselves. People use blogs to share their findings, communicate with friends, and express their opinions [11].

Some studies focused on link structure analysis on blogs, for example, Kritikopoulos et al. proposed a method to find the social relationship between bloggers by analyzing link structure [15]. Agarwal et al. identified the influential bloggers in a community by analyzing blog cross-links [3].

Several researches focused on analyzing blog content, to discover valuable information, including categorizing blogger's interests based on short snippets of blog posts [19], and identifying bloggers' emotion ratings from short blog posts [8]. Blog post had been analyzed to recommend suitable tags [32], and automatically predict trends [9]. A variety of methods has been proposed for user modeling and personalized recommendation in blog space. For example, Liu et al. [20] classified articles into predefined categories to identify authors' preferences, and thus to automatically recommend blog articles which are suitable for their interests by analyzing the contents

which bloggers have acted on. Tsai and Liu [35] recommended blog articles for mobile applications by analyzing the popularity trend of blog topics. Tsai et al. [36] combined semantic tagging and personal social model to recommend blogs. Huang et al. [10] proposed an approach to extract relevant terms from blog articles associated with users, and then recommend blog articles explored by Google's search engine.

Nevertheless, most research did not consider user reputations and popularity degree of blog articles. We analyze user's push (recommend) behavior in social bookmarking website to derive user reputations. The reputation-based popularity degrees of blog articles are derived based on user reputation, and are used as the kernel of our recommendation approach.

2.1.2 Social Bookmarking

Social bookmarking via web-based systems enables users to manage their bookmarks of web pages. Famous social bookmarking sites like Slashdot.org and digg.com have their own model of reputation where users with extensive authorship and recommendation are promoted to being moderators and super-moderators [33]. Current research include Klaisubun et al. [14], who proposed an analysis of users' behaviors in discovering useful information resources through social bookmarking services. Puspitasari et al.[27] applied social bookmarking in digital library system, combining comments and ratings to help people find objects of quality.

funP (<http://funp.com>) [18] is a company that provides bookmarking service in Taiwan, allowing people to share their articles and recommend other people's articles through the web platform. In funP, hot articles are separated and organized into different

categories. When users publish their articles, they have to choose one main category, such as “travel”, and then they will be asked to provide some tags describing the topics of their articles. funP only provides ‘hot’ article recommendations, such that every user receives the same article recommendations regardless of his/her interest. Traditional social bookmarking sites only promote the hottest article, without considering user interests and reputation.

2.2 Recommender Systems

As e-commerce prospers, an overwhelming amount of information flows through the Internet has caused the problem of information overload. Given this problem, recommender systems have emerged in various applications to provide assistance [22, 28].

2.2.1 Content-based Filtering

The Content-based filtering (CBF) approach analyzes users’ preferences on the attribute features of item to build up a personal feature profile, and then predict which items the user will like [6]. Content-based filtering (CBF) has been used mainly in the context of recommending items such as web pages and news articles, etc, by analyzing their content descriptions. The content is parsed, and item features are extracted to establish a characteristic profile. Items that were previously liked by a user are used to generate a user profile. Therefore, to pre-process the item content, the content-based recommender systems depend heavily upon the techniques of information retrieval. The limitation of the CBF approach is that users can only receive the recommended items which are similar to the past.

2.2.2 Collaborative Filtering

Based on the relationship between items or users, CF method can be classified into two types [30], user-based CF (UCF) and item-based CF (ICF). The UCF exploits historical data expressing preferences to form user neighbors and make recommendations based on those similar users' opinions. ICF [31] analyzes the similarities between items, which are based on user's ratings among items. Then, the item similarities are used to compute recommendations for a user by finding items that are similar to those items the user has liked previously. A famous example is Amazon.com that recommends similar items to the customer based on past records [17].

2.2.3 Reputation System

A reputation system collects, distributes, and aggregates feedback about participants' past behavior [28], allowing users to maintain trust in a dynamic environment. Many researchers proposed to use reputation as an auxiliary factor in the recommending phase. Opinion leader in a group can be identified by using the reputation system [26]. Adler and Alfaro [1] presented a content-driven reputation system to derive the Wikipedia author's reputation.

In addition, link analysis algorithm such as PageRank algorithm has been applied to derive users' reputation in a user interactive question-answering system [7]. Google's PageRank algorithm derives the importance web pages by computing the PageRank score of a webpage, which is basically derived from the PageRank scores of those web pages pointing to the web page [4].

3. REPUTATION-BASED APPROACH

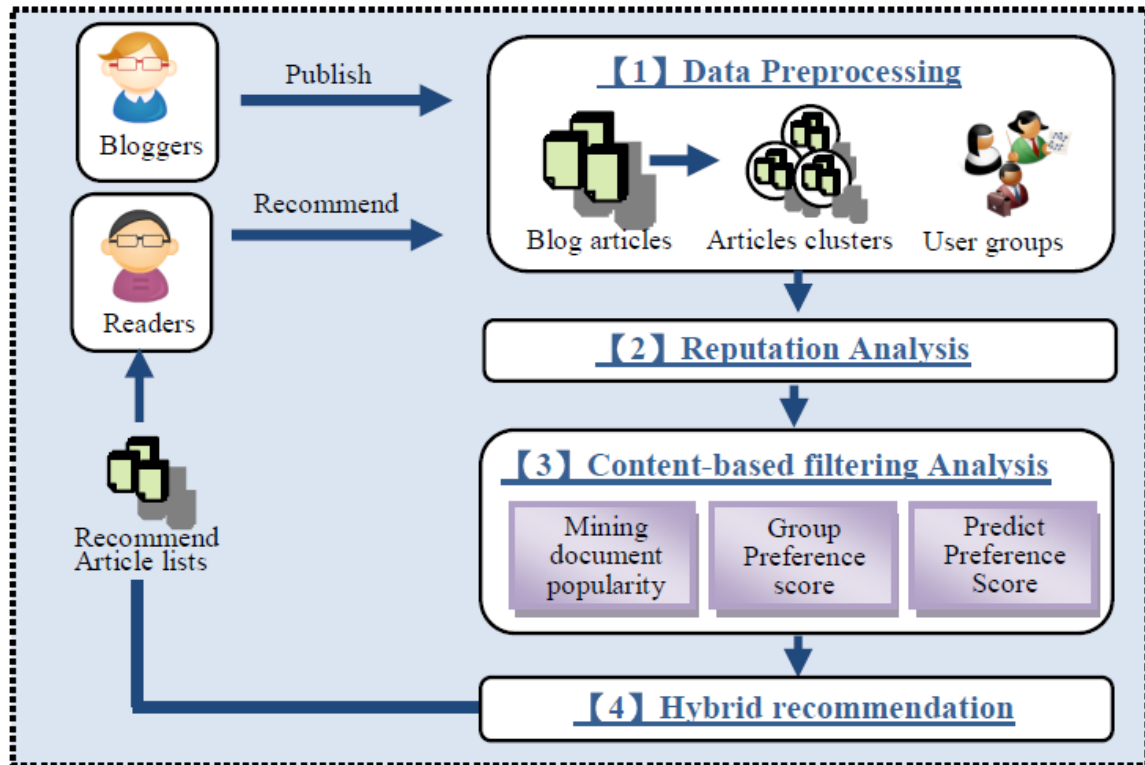


Fig. 1 Overview of proposed recommendation system

3.1 Overview

In a social bookmarking web site, funP, the users (publishers) can publish and push (recommend) their own articles or other users' articles to the web site. The users (followers) who also like the published articles can push the published articles to express their recommendation. Fig. 1 shows a framework of our proposed approach, which derives users' interests by analyzing the articles that users had pushed, and derives users' reputation scores based on the push-follower relationships between users. Our proposed approach combines the reputation-based popularity with a modified content-based filtering to recommend desirable articles to users that satisfy popularity and personal interests. The reputation-based popularity degree of an article is derived by considering

the reputation scores of users that push the article. Our reputation-based approach can perform better than the approach only based on the push counts, since the articles pushed by highly reputed users are more likely to become popular than those articles pushed by users with lower reputation scores.

In data-preprocessing step, we not only cluster blog articles, but also group users into user groups based on their article preferences. We captured and performed data preprocessing to articles that were published or pushed by the users. Each article is represented as an article profile, which is formed according to user specified tags and the term vector extracted by the *tf-idf* approach [29]. Then, we use hierarchical agglomerative clustering method [12] with group-average approach to form article clusters based on article profiles, where similar articles with similar tags or term vectors, are categorized into the same cluster. In this step, we also cluster users into groups, based on their user profiles. The user profile of a user is derived by computing the centroid (average) of the term vectors of articles that the user had pushed previously. Similar users are clustered into the same group. Note that we use the cosine measure to compute the similarity between article profiles as well as the similarity between user profiles.

The second step is reputation analysis. A user may have different reputation in different kinds of articles. For example, a user may have high reputation in ‘travel’ category, but he/she may have low reputation in ‘finance’ category. We use the push-follower relationships between users to derive the reputation scores of users in an article category.

The third step is content-based filtering Analysis. We first compute the reputation-based popularity degrees of articles, and then use a modified content-based

filtering (CBF) to derive the recommendation score of an article based on the article's neighboring articles and the preferences of the target user's group.

The final step is hybrid recommendation. By combining the reputation-based group preference score and above content-based filtering enhanced preference score, we proposed a hybrid method to recommend final article list to users. The items with top-N highest prediction scores will be recommended to the target user.

3.2 Data Preprocessing

The data was collected from funP, which is a social bookmarking website that provides online blog article recommendation service. We aggregate all the user information and article information, such as article content, post time, tags, editors, etc. We also collect the user behavior records, as for example, the records of who published which article to the platform, or who pushed which article as his/her interested article.

3.2.1 TFIDF

Each article is represented as an article profile, which is formed according to user specified tags and the term vector extracted by the *tf-idf* approach [29]. In this process, we compute the importance (term weight) of each word in their article sets. *tf* means the term frequencies which count the appearance of each word, and *idf* means the relative importance of each word in distinguishing the articles. The weight of a term *i* in a document *j*, denoted by $w_{i,j}$, is expressed as Eq. 1:

$$w_{i,j} = tf_{i,j} \cdot idf_i = tf_{i,j} \cdot (\log_2 \frac{N}{n} + 1) \quad (1)$$

where $tf_{i,j}$ is the frequency of term i in document j , idf_i is measured by $(\log_2 N/n + 1)$, N is the total number of documents in the collection, and n is the number of documents in which term i occurs at least once.

We extract the results of *tf-idf* and the tags of articles from funP Company, and use three ways to organize the profile; the top-N terms are selected as an article's profile, the top-N tags are selected as its profile; and the combination of top-N terms and tags is used as the article profile. We use a centroid approach to derive the profile of a user from the article profiles that the user had pushed previously.

3.2.2 Article Clustering

Articles from blogs are actually versatile; we might simply separate the articles into categories such as “travel”, “food”, or “sports”. However, a category may be too general to distinguish the characteristics of articles. In addition, a category may contain a lot of articles. Making the predictions of users' interests on articles based on neighboring articles from a category may result in poor recommendation quality and inefficiency. Thus, we apply clustering technique to group blog articles published within a given time window into topic clusters. The size of the time window is set as ten days. The articles published within the time window are the candidate articles for making predictions. User preferences on articles of different clusters may vary, since the articles of different clusters have different characteristics. Our recommendation approach can improve the quality and scalability of recommendation by forming neighborhoods of items from the topic clusters. A hierarchical agglomerative algorithm with group-average clustering approach [12] is applied to implement the clustering step. It treats each article as a cluster

first and then successively merges the pairs of clusters with highest cluster similarity. The similarities between two articles can be calculated by means of the cosine similarity measure, as shown in Eq. 2

$$\text{sim}(d_i, d_j) = \cos(\vec{d}_i, \vec{d}_j) = \frac{\vec{d}_i \cdot \vec{d}_j}{\|\vec{d}_i\| \cdot \|\vec{d}_j\|} \quad (2)$$

The cluster similarity between two clusters is defined as the average pairwise similarities of all pairs of articles from different clusters. We stop merging the pairs of clusters, if the highest cluster similarity is below a threshold during the merge process.

3.2.3 User Grouping

Our recommendation approach adopt the modified CBF by considering the target user's group preferences to alleviate the limitation of CBF, which recommends only those items which are similar to the items user liked previously. We cluster users into groups based on their user profiles. The user profile of a user is derived by computing the centroid (average) of the term vectors of articles that the user had published or pushed previously. Similar users are clustered into the same group. The cosine measure is used to compute the similarity between user profiles. We also use the hierarchical agglomerative algorithm as our grouping method, which stops merging the pairs of groups when similarity is below a threshold during the merge process.

3.3 Reputation Analysis

In this section, we derive users' reputation scores based on the push-follower relationships between users. A publisher is a user who published and pushed his/her own articles or other users' articles to the web site. User u_b is a follower of user u_a if u_b pushed an article after u_a has pushed the article. Our approach derives the reputation score of a user by accumulating his/her followers. Detailed steps are explained in the following.

In general, a user u will have a higher reputation score if u has more followers, implying that more users have agreed upon u 's recommendations, and thus u has higher reputation. For a given article d_i pushed by user u , we count the number of followers of u , denoted as $follower(u, d_i)$. The number of followers indicates that when a user push an article, usually how many other users will follow this user's recommendation to push the same article.

Since users may have different reputations in different article categories, we derive the category reputations of users based on article categories. Let $AveFC_u^{ca}$ denote the average number of followers of user u for pushing (recommending) articles in category ca , as defined in Eq. 3.

$$AveFC_u^{ca} = \frac{1}{|D_u^{ca}|} \sum_{d_i \in D_u^{ca}} follower(u, d_i), \quad (3)$$

where D_u^{ca} denotes the set of articles pushed by user u in category ca . Besides the average number of followers, we also consider the number of articles pushed by user u to derive the reputation score of user u in category ca , $Reputation_u^{ca}$, shown as Eq. 4. A

user u will have higher reputation if u 's average number of followers is higher. Moreover, user u will have higher reputation in category ca , if u had pushed more number of articles in category ca .

$$Reputation_u^{ca} = AveFC_u^{ca} \left(1 - \frac{1}{|D_u^{ca}| + 1} \right) \quad (4)$$

3.4 Content-based Filtering with Reputation Popularity

We use a modified content-based filtering (CBF) to derive the recommendation score of an article based on the article's neighboring articles and the reputation-based group preference scores (RGPS) of the target user's group. We note that traditional CBF only uses the similarity of user profile and article profile to derive the recommendation score, thus CBF has a limitation of recommending only those items which are similar to the items user liked previously. RGPS is derived from the article popularity considering user reputations. We separate popularity into Global popularity (GP) and Local popularity (LP), where GP is the popularity derived among all users and LP is the popularity derived from a user group.

3.4.1 Reputation-based article popularity

The popularity degree of an article d is derived by accumulating the reputation scores of users that had pushed article d . We derive the popularity degree based on the category of article d , since users have different reputations in different categories. Let

GP_d^{ca} denote the global popularity of an article d in category ca among all users. GP_d^{ca} is calculated using Eq. 5.

$$GP_d^{ca} = \sum_{u \in U_d^{push}} Reputation_u^{ca}, \quad (5)$$

where U_d^{push} is the set of users who had published or pushed article d ; $Reputation_u^{ca}$ is the reputation score of user u in category ca .

Users have been clustered into different groups based on their interests (user profiles). Accordingly, we can also derive the local popularity of an article d in category ca for a user group g , $LP_{g,d}^{ca}$, as defined in Eq. 6.

$$LP_{g,d}^{ca} = \sum_{u \in U_g \text{ and } u \in U_d^{push}} Reputation_u^{ca}, \quad (6)$$

where U_g is the set of users who belong to user group g .

3.4.2 Reputation-based Group Preference Score

To predict the reputation-based group preference score of a user group g on an article d , $RGPS_{g,d}$, we leverage the relative importance of the article's global popularity and local popularity, as defined in Eq. 7.

$$RGPS_{g,d} = [\beta \times GP_d^{ca} + (1 - \beta) \times LP_{g,d}^{ca}] \times TF_d \quad (7)$$

where GP_d^{ca} is the global popularity of an article d in category ca ; $LP_{g,d}^{ca}$ is the local popularity of an article d in category ca for user group g ; β is used as a parameter to adjust the relative importance of global popularity and the local popularity; TF_d is a time factor of article d .

It is reasonable to assume that the article published in the recent past is more interesting to users. That is to say, the older articles should be given a lower score. We consider the time decay effect of the articles. Each article is assigned a time weight according to the time it was published. Thus, higher time weights are given to articles published in the recent past. The time weight of each article is defined as Eq. 8. We adopt the formula in [38] to compute our time factor.

$$TF_d = e^{-\tau(t_{now} - t_d)}, \quad (8)$$

where t_{now} means the date now, t_d means the date time that the article was published, and τ is the tunable parameter, which we set at 1/10 to avoid it from dropping too fast.

3.4.3 Predict Preference Score

CBF derives the user profile of target user u from the article profiles of the articles that u liked, and then makes recommendations based on the profile similarity of the candidate article and the target user. We use a modified content-based filtering (CBF)

to predict the target user u 's preference score on an article d based on d 's neighboring articles and the group preferences of u 's group g . The process is shown as Fig. 2.

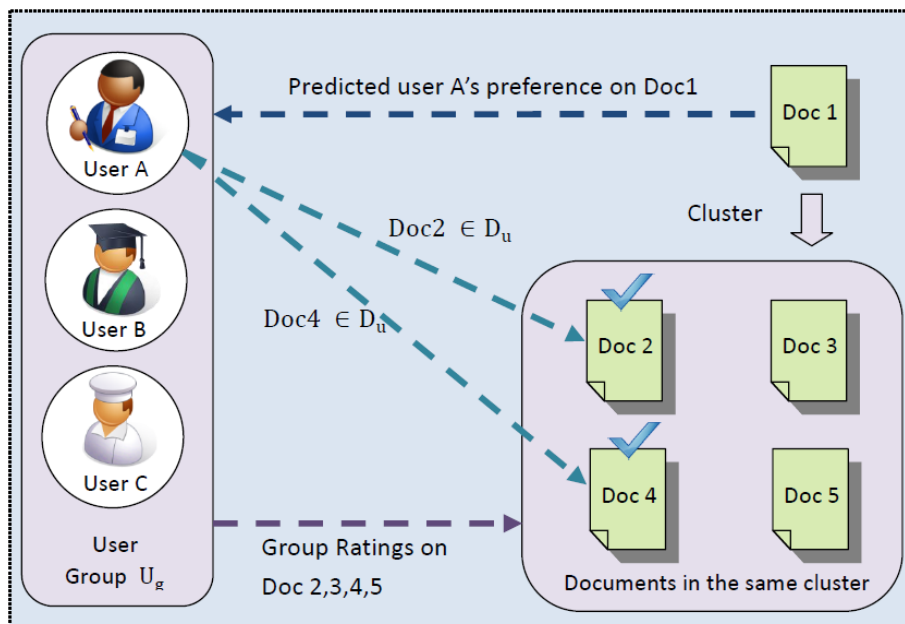


Fig. 2 Overview of CBFGRP

Let $PPS_{u,d}^{CBFRGP}$ denote the predicted preference score of target user u on the target article d ; and u belong to user group g . The articles in d 's cluster are chosen to compute their similarities with article d based on the cosine measures of their article profiles. Those articles d_j with similarity higher than a predefined threshold θ are selected as the neighboring articles of article d , which is denoted as $neighbors(d)$. Let $PS(u,d_j)$ be the preference score of target user u on article d_j . If article d_j had been pushed by u , then $PS(u,d_j)$ equals 1; otherwise, if article d_j had not been pushed by u , we use $RGPS_{g,d_j}$, the group g 's group preference score on article d_j , as u 's preference score on d_j . The content-similarities between article d and the neighboring articles are used as weights to compute the weighted average of preference scores. $PPS_{u,d}^{CBFRGPS}$ is derived using Eq. 9.

$$PPS_{u,d}^{CBFRGP} = \frac{\sum_{d_j \in \text{neighbors}(d)} PS(u,d_j) \times \text{sim}(d,d_j)}{\sum_{d_j \in \text{neighbors}(d)} |\text{sim}(d,d_j)|} \quad (9)$$

$$PS(u, d_j) = \begin{cases} 1, & \text{if } u \text{ had pushed } d_j \\ RGPS_{g,d_j}, & \text{otherwise} \end{cases}$$

CBF mainly uses the content of articles that u liked before to make recommendations, and has a limitation of recommending only those items which are similar to the items user liked previously. The quality of recommendation is even worse if very few items were liked by the target user previously. Our approach adopts the CBF by further considering the target user's group preferences on neighboring articles to alleviate the limitation. The group preference scores of neighboring articles that had not been pushed by the target user are also considered in deriving the prediction scores. The reputation-based group preference score, $RGPS_{g,d_j}$ is derived by accumulating the reputation scores of users who had pushed article d_j .

3.5 Hybrid Recommendation

In Sections 3.3.2, we have derived $RGPS_{g,d}$, the group preference score of group g on article d . In section 3.3.3, we compute $PPS_{u,d}^{CBFRGP}$, the predicted preference score of user u on article d by content-based filtering enhanced with reputation-based group preferences. In this section, we propose a hybrid method that integrates $RGPS_{g,d}$ and $PPS_{u,d}^{CBFRGP}$ to recommend articles to users. The approach derives $PPS_{u,d}^{hybrid}$, the hybrid

prediction score of user u on article d by a linear combination of the reputation-based group preference on article d , $RGPS_{g,d}$ and the prediction score $PPS_{u,d}^{CBFRGP}$, as defined in Eq. 10. The parameter α is used to adjust the relative importance of $RGPS_{g,d}$ and $PPS_{u,d}^{CBFRGP}$

$$PPS_{u,d}^{hybrid} = \alpha \times RGPS_{g,d} + (1 - \alpha) \times PPS_{u,d}^{CBFRGP} \quad (10)$$



4. EXPERIMENT EVALUATION

In this section, we conduct experiments to evaluate the recommendation quality of our proposed method in comparison with other recommendation methods. We describe funP company in Section 4.1, the experiment design is detailed in Section 4.2, followed by the experiment result, demonstrated in Section 4.3.

4.1 funP Company

We implement our method on a social bookmarking site, funP. funP is a company that provides bookmarking service in Taiwan [18], allowing people to share their articles and recommend other people’s articles through the web platform. In funP, hot articles are separated and organized into different categories. In our dataset, there are 40121 users, 19836 articles, 11 main categories, and 17987 tags (published in three weeks). Each article providing its information, including topic, category, picture, content, tags, and push counts.

Article categories include ‘food’, ‘travel’, ‘technology’, ‘sports’, ‘entertainment’, ‘political’, ‘art’, ‘life’, ‘finance’, and ‘fashion’. When users publish their articles, they have to choose one main category, such as “travel”, and then they will be asked to provide some tags describing the topics of their articles. funP only provides ‘hot’ article recommendations, such that every user receives the same article recommendations regardless of his/her interest.

4.2 Experiment Design

The dataset collected from funP is composed of two periods of ten days, from Jan

1, 2010 to Jan 20, 2010. The first ten days are used as training period, and the remaining ten days are used for testing. We analyze users' past behavior, such as, article push record and article publish record, and predict the user's future request. We recommend articles that users didn't push during the training period, and if users push those articles in the testing period, the recommendations are considered successful. Accordingly, we evaluate the performances of our proposed methods and compare them with other recommendation methods.

There are several factors that affect the quality of the recommendations. They include the recommendation approach, reputation analysis, and time factor. Through the experiments, we will discuss the issues listed below.

- What is the best relative weight between article global popularity and article local popularity, and the relative importance of group preference score and predicted preference score?
- Does the method with time factor perform better than the non-time factor one?
- Does our proposed reputation approach perform better than previous studies?
- Does content-based filtering (CBF) approach perform better than the item-based collaborative filtering (ICF) in our dataset?
- What is the effectiveness on different approaches for deriving the personalized article recommendation?

4.2.1 Data Set

There are a total of 40121 users registered, 19836 articles, 17987 tags, and 11

main article categories in funP. Table 1 shows the number of articles in each category with the number of pushes in training period. The result indicates that ‘food’, ‘travel’, and ‘life’ are more popular than other kind of articles, which means that these article can attract more audiences.

Table 1 The statistics of the articles

| Number of pushes \ Categories | More than 50 | 40~50 | 30~40 | 20~30 | 10~20 | 0~10 | 0 | Sum |
|-------------------------------|--------------|-------|-------|-------|-------|------|-------|-------|
| Food | 21 | 10 | 7 | 15 | 40 | 521 | 1719 | 2333 |
| Travel | 13 | 6 | 14 | 14 | 49 | 464 | 1552 | 2112 |
| Technology | 2 | 0 | 3 | 14 | 46 | 293 | 1597 | 1955 |
| Comic | 3 | 3 | 8 | 7 | 21 | 165 | 749 | 956 |
| Sports | 0 | 0 | 2 | 0 | 9 | 124 | 559 | 694 |
| Entertainment | 0 | 2 | 8 | 11 | 33 | 419 | 1703 | 2176 |
| Political | 6 | 2 | 1 | 5 | 20 | 154 | 702 | 890 |
| Art | 9 | 7 | 8 | 10 | 42 | 355 | 1428 | 1859 |
| Life | 13 | 6 | 11 | 14 | 39 | 556 | 2302 | 2941 |
| Finance | 3 | 0 | 5 | 4 | 17 | 127 | 992 | 1148 |
| Fashion | 0 | 0 | 0 | 1 | 6 | 164 | 1015 | 1186 |
| advertisemet | 0 | 2 | 1 | 3 | 34 | 366 | 1179 | 1585 |
| Sum | 70 | 38 | 68 | 98 | 356 | 3708 | 15497 | 19836 |

We also investigated the number of category interest of users. Table 2 shows that there were 53.47% users that are only interested in one category, which means that most users like only one kind of article; hence, we assume that users may have different reputations in different article categories. We derived the category reputations of users based on article categories, as defined in Eq. 4.

Table 2 The number of interest about article categories

| Number of category interest | User percentage |
|------------------------------------|------------------------|
| 1 | 53.47% |
| 2 | 17.90% |
| 3 | 8.23% |
| 4 | 4.92% |
| 5 | 2.60% |
| 6 | 1.83% |
| 7 | 2.42% |
| 8 | 1.61% |
| 9 | 1.48% |
| 10 | 1.66% |
| 11 | 1.74% |
| 12 | 2.15% |
| Sum | 100% |

Since many users didn't provide sufficient information, which means that they do not push articles often in funP, we only selected 200 users as our testing candidates. These candidates consist of users that participated more actively in funP. Table 4.2 shows their article pushed-count records in the testing period. The result shows that most candidates push more than 50 articles in the testing period.

Table 3 The statistics of users' push-count

| Number of pushes | Number of candiates | Percentage (%) |
|------------------|---------------------|----------------|
| More than 50 | 118 | 59% |
| 41 ~50 | 9 | 5% |
| 31~40 | 14 | 7% |
| 21~30 | 16 | 8% |
| 11~20 | 19 | 10% |
| 1~10 | 24 | 12% |
| 0 | 0 | 0% |
| Sum | 200 | 100% |

4.2.2 Evaluation Metrics

Two metrics, precision and recall, are used to measure the quality of recommendation. These are also commonly used as measures in information retrieval [5]. *Precision* is the fraction of recommended articles (predicted to be interesting) that are really found to be interesting:

$$Precision = \frac{\text{number of correctly recommended articles}}{\text{number of recommended articles}}$$

Recall is the fraction of interesting articles that can be located:

$$Recall = \frac{\text{number of correctly recommended articles}}{\text{number of interesting articles}}$$

Articles interesting to user u were those articles pushed by u in the test period. Correctly recommended articles were those that matched interesting articles. However, increasing the number of recommended items tended to reduce the precision and increase the recall. An F1-metric [37] could be used to balance the trade-off between precision and recall. F1 metric assigned equal weight to precision and recall.

$$F1 = \frac{2 \times \textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}}$$

4.2.3 Methods compared in this experiment

Our proposed methods mainly recommend articles by combining content-based filtering with reputation-based group popularity of articles. However, there are some other existing methods used to generate the recommendation list. Therefore, we compared these methods to evaluate the effectiveness of our method. The methods compared in our experiments are as follows.

CBF (Traditional Content-based filtering): CBF only considers the content similarities between the target article and neighboring articles that were pushed by the target user. CBF does not consider reputation-based group preference on neighboring articles, as shown in Eq. (11). Note that D_u is the set of articles that have been pushed by user u in the training set.

$$PPS_{u,d}^{CBF} = \frac{\sum_{d_j \in \textit{neighbors}(d) \cap D_u} \textit{sim}(d, d_j)}{|\textit{neighbors}(d) \cap D_u|} \quad (11)$$

RGPS (Reputation-based group preference): RGPS uses the reputation-based group preference score of target article by Eq. 7 to make recommendations.

CBFRGP (Content-based filtering enhanced with reputation-based group preference): CBFRGP is the method proposed in this research. CBFRGP adopts CBF by considering both the target user's preference and reputation-based group preference on neighboring articles, as calculated by Eq. 9.

Hybrid (Combining RGPS and CBFRGP): Hybrid method combines RGPS and CBFRGP, the predicting preference score of articles by content-based filtering enhanced with reputation-based group preferences, shown by Eq. 10.

ICF (Traditional Item-based collaborative filtering): ICF only consider the cross interest between users. Once an article is pushed by some users, we regard these people as having similar interest, therefore, recommend similar articles to them. ICF does not consider article's content, and derives the prediction score by replacing $sim(d, d_j)$ in Eq. 11 with $ItemSim(d, d_j)$, where we define two items to be similar if they have similar pushers, as calculated by the Jacard similarity. Note that U_{d_i} / U_{d_j} is the set of users that had pushed the article d_i/d_j .

$$ItemSim(d_i, d_j) = \frac{|U_{d_i} \cap U_{d_j}|}{|U_{d_i} \cup U_{d_j}|} \quad (13)$$

ICFRGP (Item-based collaborative filtering enhanced with reputation-based group preference): ICFRGP enhances ICF through our proposed reputation-based group preference on neighboring articles. ICFRGP derives the prediction score by replacing the $Sim(d, d_j)$ in Eq. 9 with $ItemSim(d, d_j)$.

4.3 Experimental results

4.3.1 Comparison of the weighting methods in calculating reputation

As mentioned in section 3, the relative weight between article global popularity and article local popularity in Eq. 7 is used to adjust the relative importance of group preference score. We performed an experiment by varying the value β from 0.0 to 1.0 with an increment of 0.1. F1 was derived from average scores of six different numbers of recommendations: top 40, top 60, top 80, top 100, top 120, and top 140. Fig. 3 presents the average F1 variation under different values of β . Here we used our proposed RGPS as the recommendation method. The result shows that the highest F1 is obtained when $\beta = 0.2$, which means that predicting the recommendation result can be more accurate when the system puts more weight on the local preference.

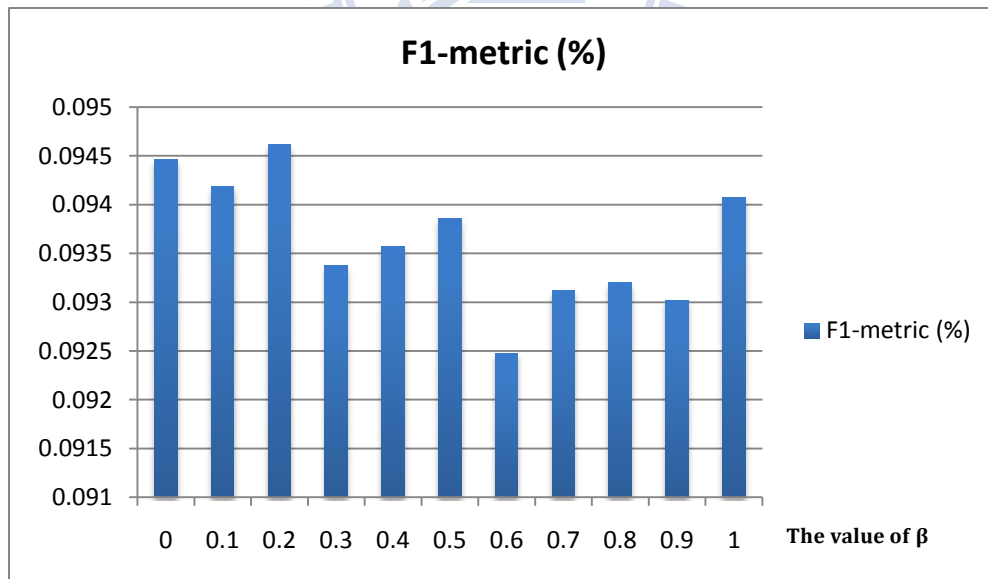


Fig. 3 The value precision under different β

4.3.2 Evaluation of time factor

Time factor plays an important role while recommending articles. Since we assume that the article published in the recent past is more interesting to users. In Eq. 8, we add a time factor TF to be a tunable parameter. That is to say, the older articles should be given a lower score, as we consider the time decay effect of the articles. We picked all articles that were published in January 1, 2010 and prove that the average push count often decreases through time by. Fig.4 shows the average push count in each day.

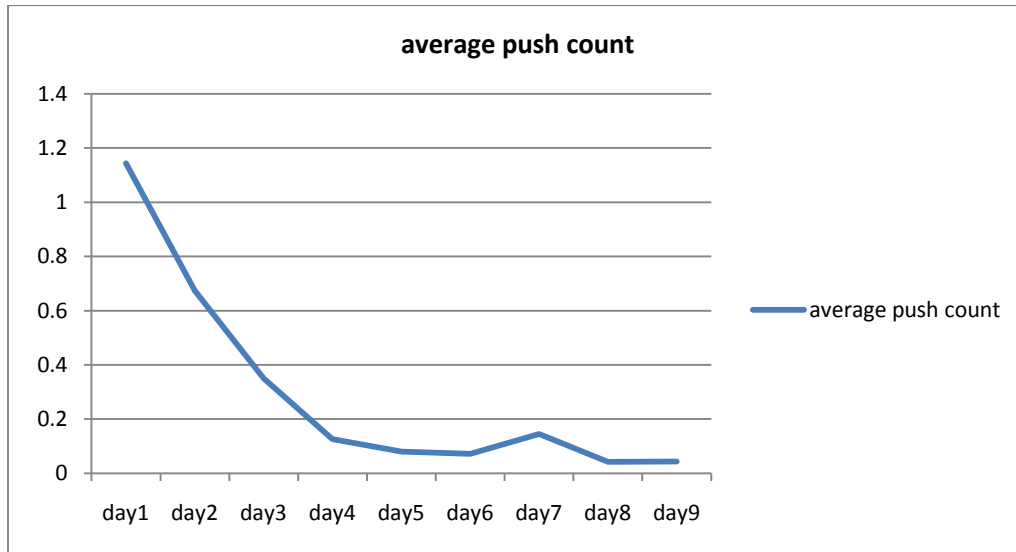


Fig. 4 Average push count

The push count rate was decreased as time goes by; therefore we perform an experiment to evaluate the effect of time factor in recommendation. F1 values are compared under different number of recommendations, shown in Fig. 5. We remove TF_d factor in Eq. 7 as a comparable method, which is named as ‘RGPS without TF’.

The F1 values of our proposed RGPS methods are higher than RGPS without TF under different number of recommendations. This indicates that the time factor contributes in deriving more precise recommendations for target users.

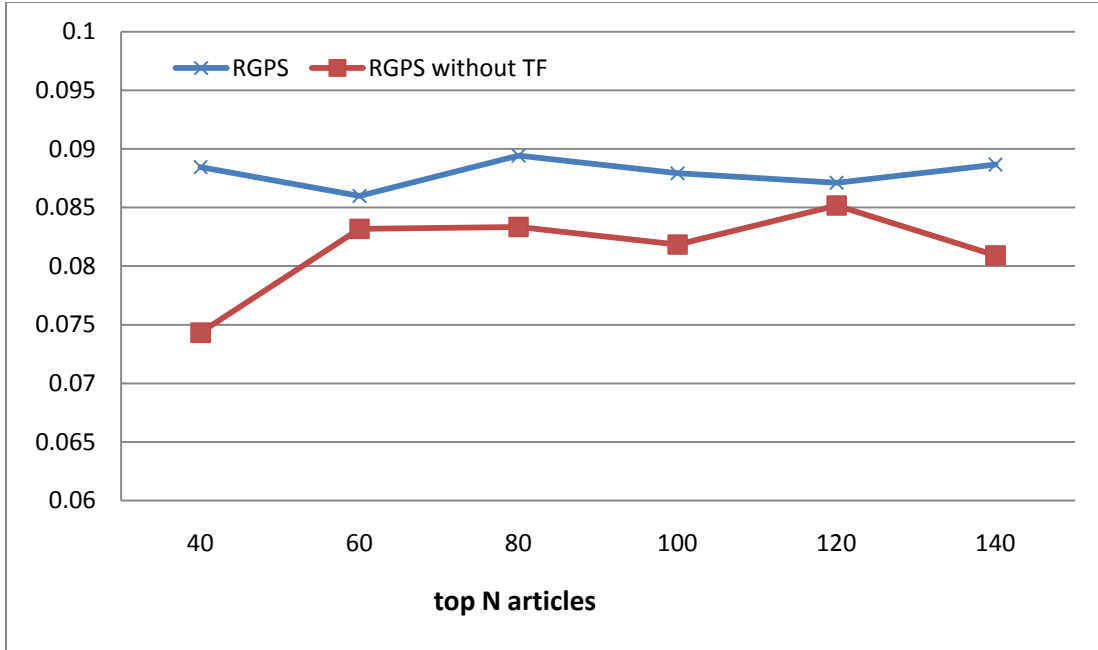


Fig. 5 The F1-metric under different number of top N articles

4.3.3 Evaluation of Reputation Analysis

In Eq. 4, we derive users' reputation scores based on the follower relationships between users. By accumulating his/her followers in different article categories, we can get the reputation scores from the number of followers. However, reputation scores can also be derived from Pagerank scores. Therefore, we compared the performance of our proposed approach with Pagerank. Since RGPS is derived from the popularity degree in Eq. 7, which in term is derived from the reputation score in Eq. 5 and Eq. 6. We separate two ways to get RGPS values. Firstly, RGPS is derived from recommendation scores of followers' relationship, and then PRankRGPS computes its recommendation scores by replacing $Reputation_u^{ca}$ in Eq. 4 with Pagerank scores, shown as Eq. 14 [25].

$$PRankRepu_{u_i} = \theta \sum_{u_o \rightarrow u_i} \frac{PRankRepu_{u_o}}{Outlink(u_o)} + (1 - \theta) \frac{1}{N} \quad (14)$$

Where $PRankRepu_{u_i}$ and $PRankRepu_{u_o}$ are the Pagerank score of user u_i and user u_o ; θ is the damping factor, generally set around 0.85; $u_o \rightarrow u_i$ means user u_o had pushed user u_i 's article before; $Outlink(u_o)$ is the total number of outlinks from user u_o ; N is the total number of users.

For comparing the performance of RGPS and PRankRGPS, F1 values are compared under different number of recommendations. The result is shown in Fig.6, which indicates that RGPS outperforms PRankRGPS.

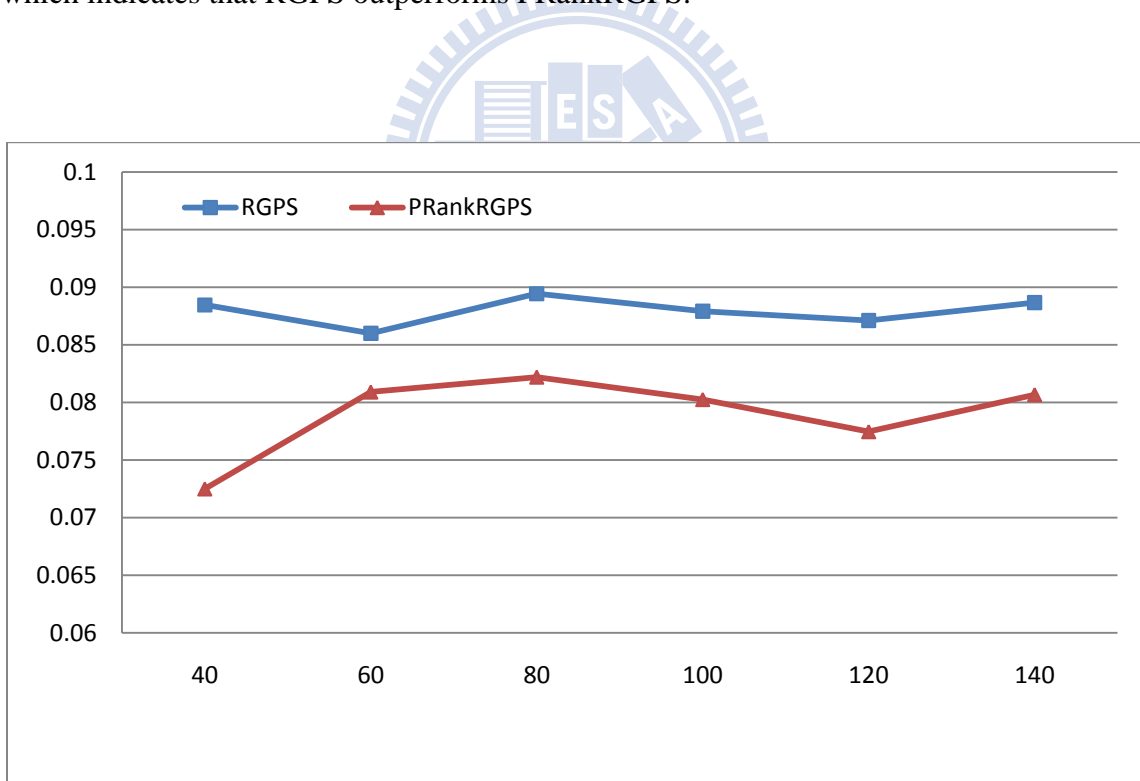


Fig. 6 The F1-metric between RGPS and PRankRGPS

4.3.4 Comparison of CBF and ICF

Our method mainly uses the Content-based filtering approach. However, collaborative filtering (CF) is also a popular recommendation method, and thus we also compared our method with one kind of CF, namely item-based CF (ICF). ICF predicts user's article preference by considering neighboring users' preference. Since users already form groups based on their article preference, if an article has not been pushed by the user before, we could predict his/her preference by counting the number of neighboring users that had pushed the article before, and at the same time compare their similarity based on their interests. We compare our proposed CBFRGP with ICF, CBF, and ICFRGP, defined at Section 4.2.3. Fig. 7 shows that our proposed reputation-based group preference factor is useful in improving the recommendation quality, because the performance of ICFRGP is better than ICF, and the performance of CBFRGP is better than CBF. The result also implies that the performance of ICF and CBF are similar, but CBFRGP has slightly higher F1 scores than ICFRGP under different top N recommendations.

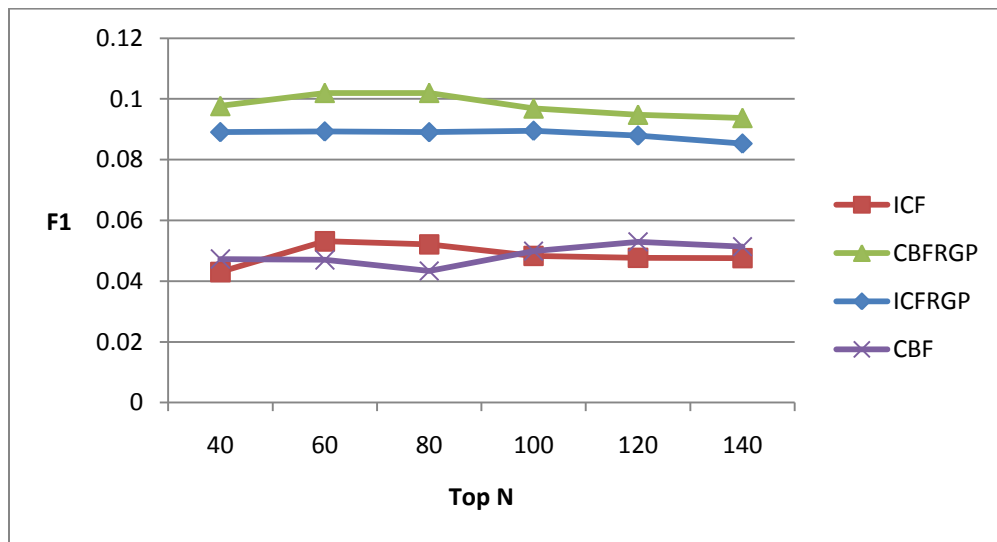


Fig. 7 The F1 of ICF, CBF, ICFRGP, and CBFRGP method

4.3.5 Comparison of the weighting methods in our hybrid method

For each user, we can obtain the group preference score (RGPS) and predicted preference score (CBFRGP) on target article. RGPS denotes the direct preference on target item, and CBFRGP denotes the predicted preference derived from neighboring items. We proposed a hybrid method in section 3.5 that integrates $RGPS_{g,d}$ and $PPS_{u,d}^{CBFRGP}$ to recommend articles to users.

The relative weight in Eq. 10 is used to adjust the relative importance of RGPS and CBFRGP of user u on article d . To examine whether the value of α would affect the result of recommendation, we varied the value of α between 0 and 1. F1 was derived from the average scores of six different numbers of recommendations: top 40, top 60, top 80, top 100, top 120, and top 140. The F1 values are illustrated in Fig. 8. The result shows that the highest F1 is obtained from $\alpha = 0.9$. The result shows that predicting the recommendation result can be more accurate when the system puts more weight on the predicted preference score (CBFRGP).

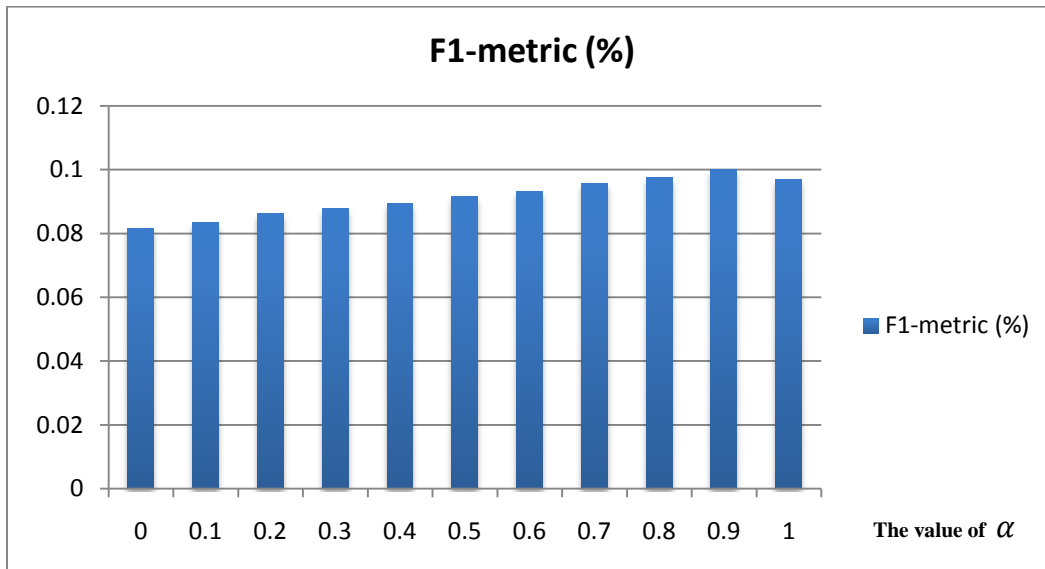


Fig. 8 The value precision under different α

4.3.6 Comparison of our proposed approach

We proposed three methods to find the recommended articles, RGPS, CBF, and Hybrid, as defined in Section 4.2.2. Here we compared these method with traditional CBF and ICF methods. Fig. 10 shows the experimental results under different top N recommendations, and the result shows the hybrid method get the highest scores. The experiment result shows that the hybrid method with $\alpha = 0.9$ is the best approach to recommend personalized and popular blog articles.

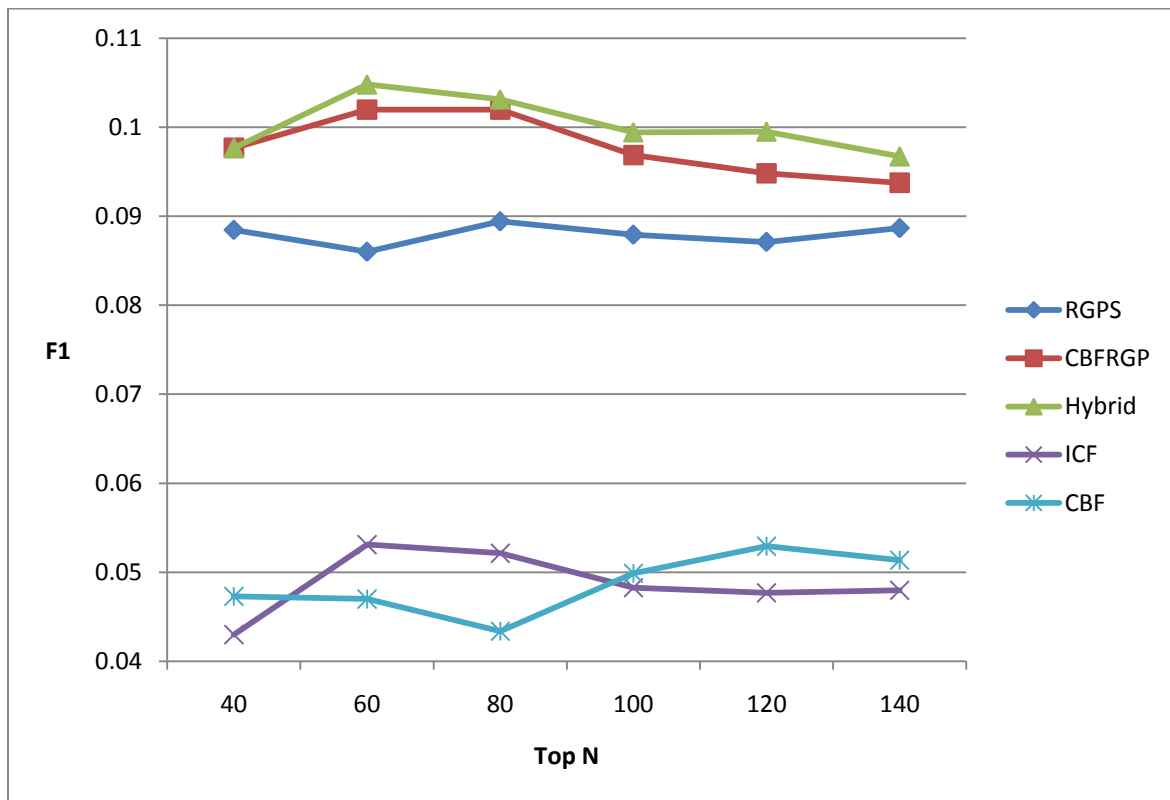


Fig. 9 The F1 of CBF, ICF, RGPS, CBF, and Hybrid method

4.3.7 Comparison of all methods

In the experiment, we compared various recommendation methods. Fig. 11 shows the F1 measures of each method under top 140 recommended articles. This ranking implies that, compared with other methods, our proposed approach has better improvements on recommendation quality.

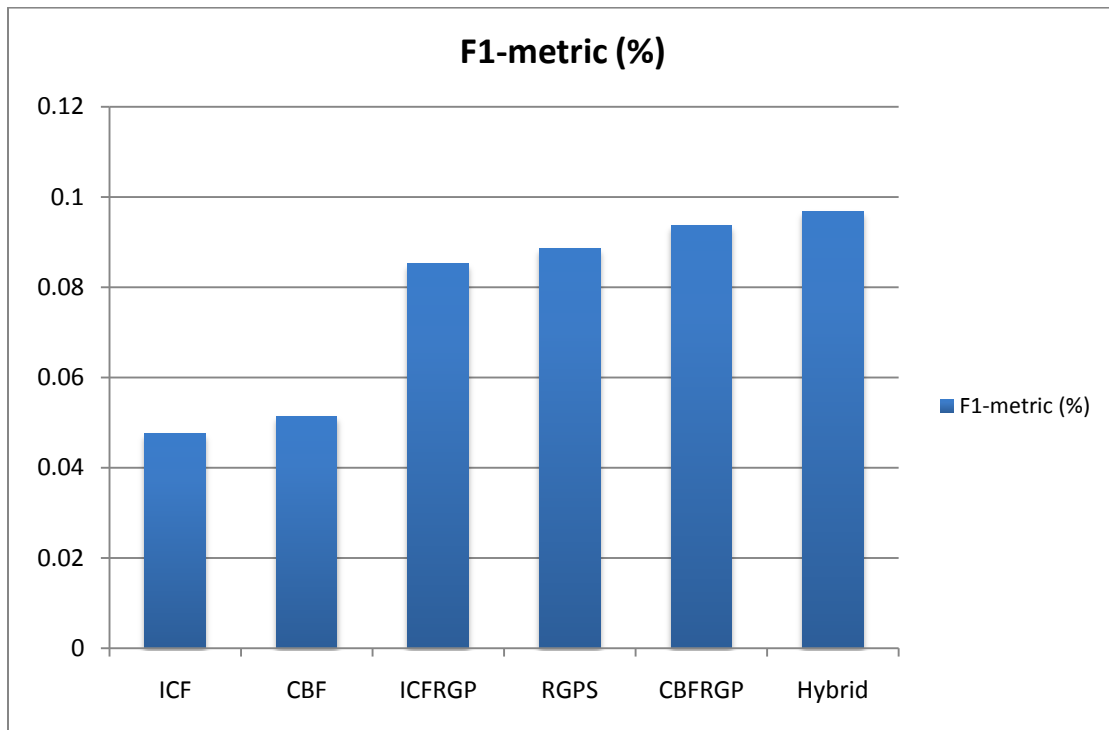


Fig. 10 The comparison of different methods

According to the experimental results, generally, the performance ranking of those methods are Hybrid > CBFRGP > RGPS > ICFRGP > CBF > ICF. This implies that CBF enhanced with reputation-based popularity can achieve higher recommendation quality. Overall, the experimental results suggest that the Hybrid method is better than other methods.

5. CONCLUSIONS

Owing to the dramatic growth of Web2.0 in recent year, blogs have become an important sharing media on the internet. Social bookmarking was invented to be a portal to collect blog articles, and bookmark lists are ranked by recency or popularity. However, people might have their own interests, and they would prefer to read articles which match their preferences. In this research, we have proposed a personalized blog recommendation service on a social bookmarking site, recommending desirable blog articles based on user preferences and reputation-based popularity of articles. We contribute to proposing a novel approach for deriving the reputation-based popularity of articles in a social bookmarking site. Moreover, our recommendation approach adopts content-based filtering (CBF) by considering the target user's group preferences to alleviate the limitation of CBF that recommends only those similar items user liked previously. Our experiment results show that the proposed method outperforms traditional CBF and ICF methods, and can effectively improve the quality of recommendation.

Future works will address in two themes. First, solving the cold-start problem; new users or new articles are hard to analyze due to lack of data. Therefore, future studies are needed to create a way to evaluate new user's preference and new article's popularity. Secondly, people often change their preference while surfing on the internet, but the present work only focuses on analyzing past data and predicting future preference. Hence further investigation include providing recommendation list based on different article categories in a real-time manner.

REFERENCES

- [1] Adler, B. T., and Alfaro, L. d. 2007. A content-driven reputation system for the wikipedia, in Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, 261-270.
- [2] Adomavicius, G., and Tuzhilin, A. 2005. 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, no. 6, 734-749.
- [3] Agarwal, N., Liu, H., Tang, L., and Yu, P. S. 2008. Identifying the influential bloggers in a community, in Proceedings of the international conference on Web search and web data mining, Palo Alto, California, USA, 207-218.
- [4] Altman, A., and Tennenholtz, M. 2005. Ranking systems: the PageRank axioms, in Proceedings of the 6th ACM conference on Electronic commerce, Vancouver, BC, Canada, 1-8.
- [5] Baeza-Yates, R. A., and Ribeiro-Neto, B. 1999. Modern Information Retrieval: Addison-Wesley Longman Publishing Co., Inc.
- [6] Burke, R. 2002. Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-Adapted Interaction, vol. 12, no. 4, 331-370.
- [7] Chen, W., Zeng, Q., and Wenyan, L. 2006. A User Reputation Model for a User-Interactive Question Answering System, in Proceedings of the Second International Conference on Semantics, Knowledge, and Grid, 40.
- [8] Gill, A. J., Gergle, D., French, R. M., and Oberlander, J. 2008. Emotion rating from short blog texts, in Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, Florence, Italy, 1121-1124.
- [9] Glance, N., Hurst, M., and Tomokiyo, T. 2004. BlogPulse: Automated Trend Discovery for Weblogs, in WWW 2004 Workshop on the Weblogging Ecosystem.
- [10] Huang, T. C., Cheng, S. C., and Huang, Y. M. 2009. A blog article recommendation generating mechanism using an SBACPSO algorithm, Expert Systems with Applications, vol. 36, no. 7, 10388-10396, September.
- [11] Isaías, P., Miranda, P., and Pifano, S. 2009. Critical Success Factors for Web 2.0 A Reference Framework, Online Communities and Social Computing, A. Ozok and P. Zaphiris, eds., pp. 354-363: Springer Berlin Heidelberg.

- [12] Jain, A. K., Murty, M. N., and Flynn, P. J. 1999. Data Clustering: A Review, *ACM Computing Surveys*, vol. 31, no. 3, pp. 264 - 323.
- [13] Kazienko, P., and Adamski, M. 2007. AdROSA--Adaptive personalization of web advertising, *Information Sciences*, vol. 177, no. 11, 2269-2295.
- [14] Klaisubun, P., Kajondecha, P., and Ishikawa, T. 2007. Behavior Patterns of Information Discovery in Social Bookmarking Service, in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, 784-787.
- [15] Kritikopoulos, A., Sideri, M., and Varlamis, I. 2006. BlogRank: ranking weblogs based on connectivity and similarity features, in *Proceedings of the 2nd international workshop on Advanced architectures and algorithms for internet delivery and applications*, Pisa, Italy, 8.
- [16] Li, H., Lee, F. M., and Chan, S. C. 2008. The Blog-Article Recommendation System (BARS). 771-776.
- [17] Linden, G., Smith, B., and York, J. 2003. Amazon.com recommendations: item-to-item collaborative filtering, *IEEE Distributed Systems Online*, vol. 4, no. 1, 2003.
- [18] Liu, E. Z. F., and Chang, Y. F. 2008. The learning opportunities of social bookmarking service: an example of FunP, *WSEAS TRANSACTIONS on SYSTEMS*, vol. 7, no. 10, 1196-1205.
- [19] Liu, J., Birnbaum, L., and Pardo, B. 2008. Categorizing blogger's interests based on short snippets of blog posts, in *Proceeding of the 17th ACM conference on Information and knowledge management*, Napa Valley, California, USA, 1525-1526.
- [20] Liu, K., Chen, W., Bu, J., Chen, C., and Zhang, L. 2007. User Modeling for Recommendation in Blogspace, in *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops*, 79-82.
- [21] McNally, K., O'Mahony, M. P., Smyth, B., Coyle, M., and Briggs, P. 2010. Towards a reputation-based model of social web search, in *Proceeding of the 14th international conference on Intelligent user interfaces*, Hong Kong, China, 179-188.

- [22] O'Donovan, J., and Smyth, B. 2005. Trust in recommender systems, in Proceedings of the 10th international conference on Intelligent user interfaces, 167-174.
- [23] O'Donovan, J., and Smyth, B. 2006. Is trust robust?: an analysis of trust-based recommendation, in Proceedings of the 11th international conference on Intelligent user interfaces, Sydney, Australia, 101-108.
- [24] Oreilly, T. 2007. What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software, *Communications & Strategies*, First Quarter, SSRN, 17.
- [25] Page, L., Brin, S., Motwani, R., and Winograd, T., *The PageRank Citation Ranking: Bringing Order to the Web*, Technical Report, Stanford InfoLab, 1999.
- [26] Peng, T. C., and Chou, S. C. T. 2009. iTrustU: a blog recommender system based on multi-faceted trust and collaborative filtering, in Proceedings of the 2009 ACM symposium on Applied Computing, Honolulu, Hawaii, 1278-1285.
- [27] Puspitasari, F., Lim, E.-P., Goh, D. H.-L., Chang, C.-H., Zhang, J., Sun, A., Theng, Y.-L., Chatterjea, K., and Li, Y. 2007. Social bookmarking in digital library systems: framework and case study, in Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, Vancouver, BC, Canada, 488-488.
- [28] Resnick, P., Kuwabara, K., Zeckhauser, R., and Friedman, E. 2000. Reputation systems, *Communications of the ACM*, vol. 43, no. 12, 45-48.
- [29] Salton, G., and McGill, M. J. 1986. *Introduction to Modern Information Retrieval*: McGraw-Hill, Inc.
- [30] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. 2000. Analysis of recommendation algorithms for e-commerce, in Proceedings of the 2nd ACM conference on Electronic commerce, Minneapolis, Minnesota, United States, 158-167.
- [31] Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. 2001. Item-based collaborative filtering recommendation algorithms, in Proceedings of the 10th international conference on World Wide Web, Hong Kong, Hong Kong.

- [32] Sood, S., Owsley, S., Hammond, K., and Birnbaum, L. 2007. TagAssist: Automatic Tag Suggestion for Blog Posts, in International Conference on Weblogs and Social (ICWSM).
- [33] Sundaresan, N. 2007. Online trust and reputation systems, in Proceedings of the 8th ACM conference on Electronic commerce, San Diego, California, USA, 366-367.
- [34] Tanaka, K. 2009. Web search and information credibility analysis: bridging the gap between Web1.0 and Web2.0, in Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, Suwon, Korea, 39-44.
- [35] Tsai, P.-Y., and Liu, D.-R. 2009. Personalized Popular Blog Recommender Service for Mobile Applications, in Proceedings of the 10th International Conference on E-Commerce and Web Technologies, Linz, Austria, 2-13.
- [36] Tsai, T. M., Chia, C. S., and Chou, S.-C. T. 2006. Personalized Blog Recommendation Using the Value, Semantic, and Social Model, in Proceedings of the innovations in information Technology, Dubai 1-5.
- [37] Van Rijsbergen, C. J. 1979. Information Retrieval, 2nd ed., London: Butterworth-Heinemann.
- [38] Zhang, J., Ackerman, M. S., Adamic, L., and Nam, K. K. 2007. QuME: a mechanism to support expertise finding in online help-seeking communities, in Proceedings of the 20th annual ACM symposium on User interface software and technology, Newport, Rhode Island, USA, 111-114.