Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 19 (2013) 412 – 419

Procedia
Computer Science

The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013)

LINK RECOMMENDER: Collaborative-Filtering for Recommending URLs to Twitter Users

Nazpar Yazdanfar, Alex Thomo

Department of Computer Science, University of Victoria, Victoria, Canada, V8P 5C2

Abstract

Twitter, the popular micro-blogging service, has gained a rapid growth in recent years. Newest information is accessible in this social web service through a large volume of real-time tweets. Tweets are short and they are more informative when they are coupled with URLs. Due to tweet overload in Twitter, we believe that an accurate URL recommender system is a beneficial tool for information seekers. In this paper, we focus on a neighborhood-based recommender system that recommends URLs to Twitter users. We consider one of the major elements of tweets, *hashtags*, as topic representatives of URLs in our approach. We propose methods for incorporating hashtags in measuring the relevancy of URLs. Our experiments show that our neighborhood-based recommender system outperforms the matrix factorization-based system significantly. We also show that the accuracy of URL recommendation in Twitter is time-dependent. A higher recommendation accuracy is obtained when more recent data is provided for recommendation.

Keywords: Collaborative-filtering algorithms, Recommendation systems, Social network

1. Introduction

Micro-blogging web sites such as Twitter offer an opportunity to investigate large-scale social systems where preferences of users are traceable from their activities. Despite Twitter's simplicity, it provides many advantages for its users in the form of social short messaging. Twitter allows users to constantly update their public time lines and follow the post history of their favorite topics and people. Due to the popularity and worldwide usage of Twitter, a growing body of research critically analyzes it from different aspects. However, from the perspective of recommender systems, there is not much work done to use Twitter data as a target for applying diverse recommendation techniques and be able to recommend various items, such as tweets, URLs, followees, or hashtags to twitterers.

In this paper, we focus on the recommendation of URLs (occurring in tweets) using collaborative-filtering approaches. To this end, we propose a neighborhood-based (NB) approach that utilizes user-URL-hashtag connections. The advantage of an NB approach is that it provides immediate recommendations for users with newly entered records. This ability becomes important in our Twitter recommendation system

Email addresses: ynazpar@cs.uvic.ca (Nazpar Yazdanfar), thomo@cs.uvic.ca (Alex Thomo)

doi:10.1016/j.procs.2013.06.056

because users tend to retweet URLs constantly. Intuitively, a neighborhood approach produces recommendations based on correlations between either pairs of items or pairs of users. In an item-based approach, the preference prediction (of a user for an unknown item) is based on the ratings of similar items by the same user, while in a user-based approach, the preference prediction is based on the ratings of similar users for the same item.

Recommending URLs that match Twitter users' is of great importance. In recent years people benefit from social networks such as Facebook and Twitter to share their interesting URLs with their friends and followers. URL sharing gained popularity in Twitter since tweets are brief and sometimes noisy. As a result, twitterers add URLs to their tweets to expand them. Therefore, very often, published messages in Twitter contain URLs. Notifying users of URLs fitting to their interest in the "sea" of tweets assists them find recent updates more quickly.

However, discovering twitterers' favorite URLs based on collaborating filtering is a challenging problem. Every minute, hundreds of URLs are published in Twitter referring to various topics, events, and news. Twitterers' tastes in topics might vary constantly. Due to such data sparsity, naïve collaborating filtering techniques show poor performance when the correlations between pairs of items or users are small.

In order to alleviate data sparsity, a proposed solution is to reduce the rank of the user-item matrix using matrix factorization methods such as Singular Value Decomposition (SVD) [1]. Unlike neighborhood approaches, matrix factorization methods transform both user and item vectors to a latent factor space, where similarities between items and users are generated by lower-dimensional hidden factors automatically inferred from data. Matrix factorization methods have attracted significant attention in recent years due to their success in the Netflix Prize competition. However, applying SVD on incomplete matrices, as in the case of collaborative-filtering, might produce results that are not easy to characterize. On the other hand, filling missing ratings is expensive and considerably increases the complexity. In the Twitter URL recommendation case we focus on, we show that matrix factorization methods used in the Netflix competition did not produce results of high quality.

Therefore, we turn our attention to devising a neighborhood-based approach using a three dimensional matrix connecting users, URLs, and hashtags found in the tweets. Thus, instead of using raw text-based approaches, we consider hashtags, which are words marked as important by the authors of the tweets. The hashtags of a tweet can be regarded as either topics of the URLs or summary words of the tweet. For instance, a user might post a tweet saying "I enjoy following #election news + URL". Most likely, "election" is the descriptor of the URL she is posting.

In order to utilize hashtags as approximate topic indicators of a URL, we propose an approach based on collaborative-filtering to discover the user's favorite URLs. The assumption here is that different URLs presented with identical hashtags are about the same topics. Therefore, besides measuring the correlation of URLs based on the users who published them, we also exploit hashtags to boost the similarity determination of URLs. The predicted preference score of each user to each unseen item (URL) is computed by both the correlation between pairs of users and pairs of hashtags. The two scores extracted for each item are then combined to form an individual score for each user for her unobserved items. This predicted score determines whether an item should be recommended to a user or not.

More specifically, the main contributions of this paper are:

1. We propose a representation of the extracted data as a three-dimensional matrix of user-item-hashtag to solve the problem of data sparsity when measuring the correlation between items.
2. We use weighted mean and maximum functions to combine the similarities of items over user and hashtag dimensions to predict the preference score of users to each item. We show that our methods outperform the SVD method that is known as a successful method for CF recommendation.
3. We also perform a more realistic evaluation taking the timestamp of the postings of URLs into consideration. Using the timestamps, we only predict URLs using past URL postings, not future ones. This is a special feature not found typically in the evaluation of other recommender systems. Our experiments show that more data does not mean better results: considering URLs posted more than seven days in the past for recommending URLs on the present day has a clear trend of higher error rate.

The rest of the paper is organized as follows. We start from existing work in collaborative-filtering recommender systems in Section 2. Then, we describe data collection and preparation in Section 3. We define our ternary relationship between users, URLs, and hashtags in Section 4. Section 5 is devoted to present our methods and we discuss how we can incorporate hashtags in measuring item similarities. In Section 6, we compare a matrix factorization method with our new methods. Moreover, we show how the timestamps of the postings of the URLs affect the quality of our recommendations. We conclude our findings in Section 7.

2. Related work

Collaborative-filtering is regarded as a promising recommendation strategy which has attracted a great body of research from academia and industry. Google news [2] and Amazon [3] have applied this conventional recommendation approach in their systems. In recent years, there were many attempts for optimizing the performance of collaborative-filtering recommendation systems. Wang et al. [4] showed that combining user-based and item-based collaborative-filtering methods produces robust recommendations. Bogers et al. [5] used the content-based filtering to improve the precision of recommendation for social book marking websites.

With the growing attention toward Twitter as a social network, many researchers have applied recommendation techniques in Twitter. Twopics was introduced in [6] to find topics of interest for Twitter users by disambiguating and categorizing elements of a tweet. Using social graphs, a followee recommender system was also implemented for Twitter [7, 8]. By analyzing information diffusion patterns of tweets, [9] tried to recommend emergency news to users.

Collaborative-filtering is put to work to recommend various features in Twitter. Chen et al. [10] added some factors including tweet topic level, social relation and authority of the tweet's publisher to enhance Twitter-based recommendations. Hannon et al. [7] took advantage of both content-based and collaborative-filtering approaches to recommend followee to each user. Regarding to the effectiveness of collaborative-filtering, we chose this method as our target recommendation method. Inspired from [5], we investigate the effect of adding hashtags as the summary or the topic indicator of the URLs without employing content-based filtering.

3. Data

We collected public tweets through the Twitter streaming API, setting the default access level. The default access sends the same tweets if two different clients connect to this endpoint and provides approximately 1% of all public tweets flowing through Twitter [11]. From May 1 to May 22, 2012, we obtained these accessible sample tweets comprising of about 8 million tweets submitted by 4 million users. On average, 362,717 tweets were collected daily. Tweets were distributed almost the same in each day of data collection. Each tweet, also known as "status", includes: (1) a tweet identifier, (2) a user identifier, (3) date and time of creation, (4) textual content, (5) URLs, and (6) hashtags.

As the streaming sample API delivers a feed of tweets without any constraints on the language of the tweet, our tweet corpus contains tweets published in different languages and countries. Some of them were only pure text, meaning that they had no URL or hashtag. Since we aim at recommending URLs to users, those tweets were not useful for us. We only focused on tweets that contain at least one URL in their text. Next, we narrow down our filtering to *"active users"* and *"active URLs"* as defined in the following.

3.1. User and URL Selection

We observed that there were users who had only one tweet in our collection, or users who had posted the same tweet many times. In order to reduce the data sparsity and remove spam, we constructed a profile for each user. If a user has more tweets with URLs than a specific threshold, then it is labeled as an *"active user"*. Also, if a URL is shared by more active users than a specific threshold, it is called an *"active URL"*. Here, we choose ten and three as thresholds for active users and active URLs, respectively. Restricting our data to active users and active URLs, we finally obtained 63,080 unique users and 8,905 unique URLs.

Table 1. General statistics about our tweet collection

Number of tweets	7,979,777
Number of users	4,285,186
Number of URLs	6,666,457
Number of active users	63,080
Number of active URLs	8,905
Number of hashtags	5,998

3.2. Hashtag Selection

Twitter allows users to emphasize on the keywords of their tweets by creating hashtags. Twitterers use the hashtag symbol # before a relevant keyword or a phrase to categorize tweets. Intuitively, hashtags can be assumed as topic indicators of both tweets and URLs. We extracted the hashtags of all tweets regardless of the tweet language.

4. Data representation

Unlike other works (cf. [12, 13, 14]) that model a topic for each item from the textual content of posts, we directly benefit from hashtags by considering them as the topics of items (URL). Our major goal here is to improve the quality of collaborative-filtering recommendations by creating another layer of correlation between pairs of items that employs hashtags. In this way, each URL not only has a vector of users who have tweeted it, but also has another vector of hashtags that are assigned to it.

Assuming a ternary relationship between items, users, and hashtags, we define our relationship as a 3D matrix

$$\mathbf{R}(i_k, u_l, h_m),$$

where:

- an item (URL) is referred as i_k, where $k \in [1, K]$, and K is the number of items,
- a user is referred as u_l, where $l \in [1, L]$ and L is the number of users,
- a hashtag is referred as h_m, where $m \in [1, M]$, and M is the total number of distinct hashtags.

Typically, in item-based, collaborative-filtering recommender systems, there is an item-user matrix that is used to keep the records about how users and items are connected using users' ratings for items. However, in the context of Twitter, the ratings should be inferred from users' behavior as Twitter does not support explicit ratings. Therefore, we form our $K \times L \times M$ user-item-hashtag matrix with binary values. If URL i_k is posted by user u_l who has also defined hashtag h_m, we fill cell $\mathbf{R}[i_k, u_l, h_m]$ with 1, otherwise 0.

From the initial 3D matrix, we create the following two 2D matrices. The first is the $K \times L$ item-user matrix IU, where each item i_k is represented as a row vector of users who submitted the item, and each user u_l is represented as a column vector of items the user posted. The second matrix is the $K \times M$ item-hashtag matrix IH that we obtain it by aggregating R over users as follows:

$$IH(i_k, h_m) \quad = \quad \sum_{l=1}^{L} R(i_k, u_l, h_m)$$

This matrix incorporates hashtags in the collaborative-filtering approach describing the connection between items and hashtags. In this matrix, each item i_k is represented as a row vector of hashtags assigned to the tweets containing the item, and each hashtag is represented as a column vector of items posted in tweets having the hashtag. The cells of matrix IH contain the number of times a certain hashtag is assigned via a

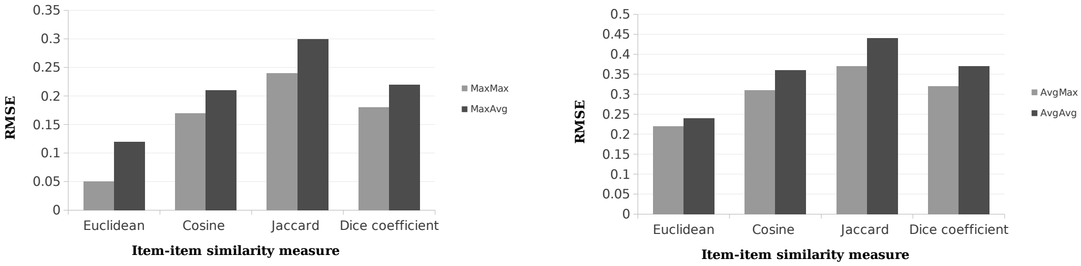

Fig. 1. Results of four different methods for incorporating hashtags and producing recommendations.

tweet to an item. In our method, each item i_k is described by two vectors: a vector of users who have posted i_k in their tweets, and a vector of hashtags that were assigned to the tweets containing i_k.

The main problem we focus is to predict whether a user u_l will like an item i_k. For this, our method computes a score $\hat{r}_{k,l} \in [0, 1]$, and based on this score we produce the recommendation.

5. Our Method

Unlike [5] that uses only an item-tag matrix for calculating similarities between items, we improve item-item similarity computations by benefiting from both item-hashtag and item-user matrices. For instance, assume we have two item vectors $\vec{I_k} = \langle u_1, u_2, u_3 \rangle$ and $\vec{I_j} = \langle u_4, u_5, u_6 \rangle$ in matrix IU, with corresponding hashtag vectors $\vec{I_k} = \langle h_1, h_2, h_3 \rangle$ and $\vec{I_j} = \langle h_2, h_3, h_4 \rangle$ in matrix IH, respectively. Considering IU alone, as in the conventional collaborative-filtering, we get 0 as the correlation of $\vec{I_k}$ and $\vec{I_j}$, since they have no user overlap. On the other hand, following the approach of [5] that measures item similarities based on only an item-hashtag matrix we suffer information loss for the items that have overlaps in both user and hashtag dimensions. Consequently, the relevance score $\hat{r}_{k,l}$ of such items should be boosted when measuring item-item similarity based on hashtag and user correlations simultaneously.

In order to calculate the correlation between two items, i_k and i_j, a similarity measure is required. We use four similarity measures: *Euclidean, cosine similarity, Jaccard, and Dice coefficient*.

Formally, the similarities between two items i_k to i_j for the above measures are as follows:

$$\begin{aligned}
sim_{Eucliean}(i_k, i_j) &= -\sqrt{\sum_{w=1}^{W} (i_{k_w} - i_{j_w})^2} \\
sim_{cosine}(i_k, i_j) &= -\frac{i_k \cdot i_j}{||i_k|| \cdot ||i_j||} \\
sim_{Jaccard}(i_k, i_j) &= \frac{|i_k \cap i_j|}{|i_k \cup i_j|} \\
sim_{Dice}(i_k, i_j) &= \frac{2 \cdot |i_k \cap i_j|}{|i_k| + |i_j|}
\end{aligned}$$

where $||.||$ denotes the length of a vector (square root of the sum of squares of the components), whereas $|.|$ denotes the cardinality of a vector considered as a set of elements.

Since obtaining item-item similarity scores by focusing only on the hashtag dimension or only on the user dimension causes loss of information, once the similarities between the items in item-user and item-tag matrices are measured via a similarity function, we need a technique to combine the two scores so that the similarity between items i_k and i_j is generated as a single score. To this end, we use two methods of weighted mean and maximum over similarities of items based on the hashtag and user vectors of i_k and i_j.

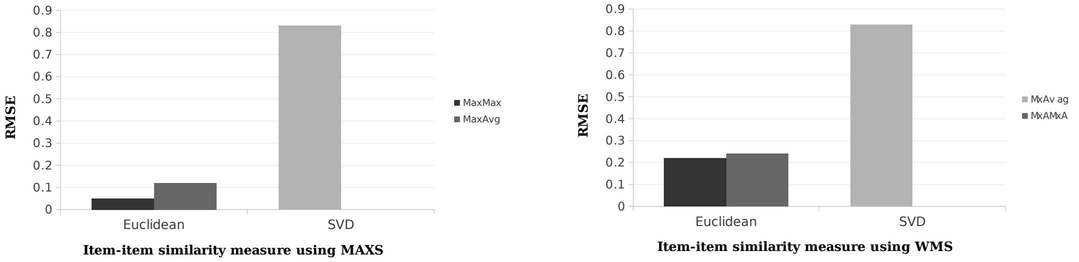

Fig. 2. Comparing the RMSE of matrix factorization-based method with our neighborhood-based method.

Formally, the weighted mean method is defined as:

$$WMS(i_k, i_j) = \frac{\alpha \cdot sim_u(i_k, i_j) + \beta \cdot sim_h(i_k, i_j)}{(\alpha + \beta)} \tag{1}$$

where $sim_u(i_k, i_j)$ is the similarity between two items in item-user matrix IU, $sim_h(i_k, i_j)$ is the similarity between two items in item-hashtag matrix IH, and α and β are the weights we give to the importance of each one of these two similarities that are determined via testing.

We also use the maximum function $MAXS(i_k, i_j)$ that outputs the maximum value of $sim_u(i_k, i_j)$ and $sim_h(i_k, i_j)$ as:

$$MAXS(i_k, i_j) = max\{sim_u(i_k, i_j), sim_h(i_k, i_j)\} \tag{2}$$

After combining the similarity scores generated from two matrices of item-user and item-hashtag, we find the prediction scores $\hat{r}_{k,l}$ (to recommend item i_k to user u_l) following an item-item collaborative-filtering approach.

Specifically, the top N similar items to an item i_k that user u_l has posted are ranked in a descending order and inserted into a list, called $L_{k,l}$. For predicting score $\hat{r}_{k,l}$, one way is to consider the *mean* or the *maximum* similarity to i_k of the items in list $L_{k,l}$. If the calculated score $\hat{r}_{k,l}$ passes a threshold, we recommend i_k to u_l. In summary, our algorithm for predicting the score $\hat{r}_{k,l}$ of i_k for user u_l is described in three major steps:

- Compute the list of the top N similar items to i_k which are posted by u_l, called $L_{k,l}$. The item-item similarities are computed using the *WMS* or *MAXS* described earlier.
- Then, compute the mean or the maximum similarity to i_k of the items on $L_{k,l}$. This is score $\hat{r}_{k,l}$.
- If $\hat{r}_{k,l}$ is greater than a threshold value, we recommend item i_k to u_l.

In our evaluation we use an unbiased threshold of 0.5 regardless of $\hat{r}_{k,l}$ distributions.

6. Evaluation

We experimented two hypotheses on our URL recommendation system. In the first experiment, we test the impact of considering hashtags as metadata for each item in collaborative-filtering. To show how metadata affects the accuracy of collaborative-filtering, we consider hashtag similarities as well as user similarities when computing the correlations between pairs of items. Next, we investigate whether our Twitter-based recommender system is time-sensitive. As Twitterers' topics of interest constantly change over time, recommendation quality should also vary by time. To answer these questions, we run a second set of experiments to observe temporal variability in the recommendation accuracy of our URL recommender system. We evaluated the quality of our methods on Twitter data using the standard root mean squared error as:

$$RMSE = \sqrt{\frac{\sum_{(K,l) \in TestSet} (r_{k,l} - \hat{r}_{k,l})^2}{|Testset|}} \tag{3}$$

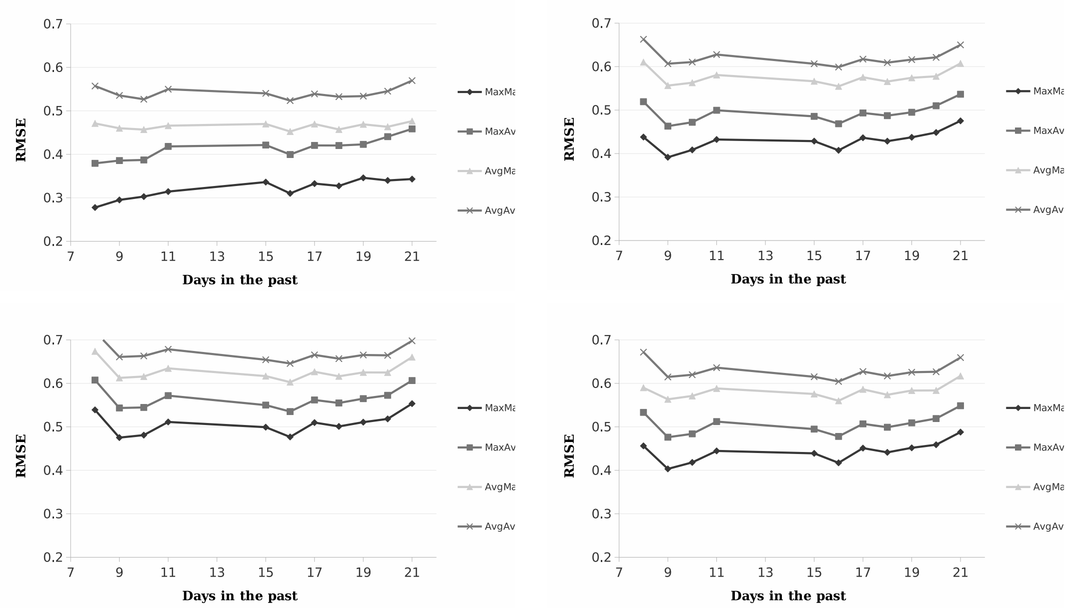

Fig. 3. RMSE when using: Euclidean similarity **[Top-Left]**, Cosine similarity **[Top-Right]**, Jaccard similarity **[Bottom-Left]**, Dice similarity **[Bottom-Right]**.

where $r_{k,l}$ is 1 if user u_l has posted item (URL) i_k. Since we are doing evaluation here, we follow the hide-one out approach, in which we hide one URL posting and then try to predict it using our method. Based on Figure 1, the best evaluation result (*RMSE* = 0.05) is reported when we used Euclidean measure on IH and IU matrices with *MAXS*.

6.1. Comparison with the Matrix Factorization method

The effect of hashtags as metadata can be also seen when we compare our methods against a best-in-class collaborative-filtering approach based on matrix factorization from the Neflix competition [15]. As we show in Figure 2, applying SVD on Twitter data results in poor recommendation quality compared to both methods of *MAXS* and *WMS* explained in Section 5 for incorporating hashtag similarities with user similarities when measuring item correlations. Our methods show a significant performance in terms of having smaller RMSE than the SVD method. The huge RMSE difference is shown when we compute either mean or maximum similar items to each item that we try to predict.

6.2. Temporal variation in recommendation accuracy

According to previous studies, taking the time variance into account when implementing a collaborative-filtering recommendation system may generate more precise recommendations. Based on [16, 17] the time-incorporated recommender systems produce more accurate recommendations than the pure collaborative-filtering systems. Here, we run our experiment to show that our Twitter-based recommender system is time-sensitive, meaning that the RMSE differs as tweets become more recent from the time of first tweet collected.

We define a time window indicating the number of days we move backward in time from the time of URL post which we hide and try to predict. Figure 3 shows the effect of increasing the time window. The temporal RMSE variation follows an upward trend for RMSE as the size of window grows. In other words, the more days we consider in the past, the worst the recommendations get. This conforms to an explanation that the Twitters' typically care more for fresh, recent URLs.

Regardless of the approach for merging user and hashtag similarities or the method for selecting the similar items, RMSE rises when the time lag between the posting time of the hidden URLs and the rest of

URLs considered for recommendation grows from seven days. This finding matches the result of [18] that explains sharing URLs increases to reach a peak and then starts to fall after a period. As the result, in our case it can be inferred that item similarities based on their common users and hashtags start to decline as old URLs will eventually have fewer users or hashtags in common with more recent ones.

7. Conclusions

We have proposed a neighborhood-based approach for recommending URLs to Twitter users. Our methods suggest that the accuracy of the collaborative-filtering recommender system can improve if we aggregate hashtag similarities and user similarities. Our experiments have shown that our solutions for Twitter-based recommender system can achieve better performance in term of RMSE. We have studied the performance of matrix factorization methods (SVD) as a prize-winner method for Netflix competition on our Twitter data and observed a huge RMSE (0.83) compared to our RMSE for our best approach. We have also shown that providing more posting history of users will not result in more accurate performance and it will damage recommder accuracy in Twitter. Our temporal RMSE variation experiment demonstrated that taking the timestamps of the postings of the URLs into account is quite beneficial in recommendation performance.

References

[1] Y. Koren, R. M. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, IEEE Computer 42 (8) (2009) 30–37.

[2] A. S. Das, M. Datar, A. Garg, S. Rajaram, Google news personalization: scalable online collaborative filtering, in: Proceedings of the 16th international conference on World Wide Web, WWW '07, ACM, New York, NY, USA, 2007, pp. 271–280.

[3] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, Internet Computing, IEEE 7 (1) (2003) 76 – 80.

[4] J. Wang, A. P. de Vries, M. J. T. Reinders, Unifying user-based and item-based collaborative filtering approaches by similarity fusion, SIGIR '06, ACM, New York, NY, USA, 2006, pp. 501–508.

[5] T. Bogers, A. van den Bosch, Collaborative and Content-based Filtering for Item Recommendation on Social Bookmarking Websites, in: Proceedings of the ACM RecSys'09 Workshop on Recommender Systems & the Social Web, New-York, NY, USA, 2009, pp. 9–16.

[6] M. Michelson, S. A. Macskassy, Discovering users' topics of interest on twitter: a first look, in: AND, ACM, 2010, pp. 73–80.

[7] J. Hannon, M. Bennett, B. Smyth, Recommending twitter users to follow using content and collaborative filtering approaches, RecSys '10, ACM, New York, NY, USA, 2010, pp. 199–206.

[8] J. Hannon, K. McCarthy, B. Smyth, Finding useful users on twitter: Twittomender the followee recommender, in: ECIR, 2011, pp. 784–787.

[9] J. Cheng, A. Sun, D. Hu, D. Zeng, An information diffusion-based recommendation framework for micro-blogging, J. AIS 12 (7).

[10] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yoa, Y. Yu, Collaborative personalized tweet recommendation, SIGIR '12, ACM, Portland, Oregan,USA, 2012.

[11] https://dev.twitter.com/docs/streaming-apis.

[12] M. J. Pazzani, J. Muramatsu, D. Billsus, Syskill & webert: Identifying interesting web sites, in: AAAI/IAAI, Vol. 1, 1996, pp. 54–61.

[13] M. Balabanović, Y. Shoham, Fab: content-based, collaborative recommendation, Commun. ACM 40 (3) (1997) 66–72.

[14] R. J. Mooney, L. Roy, Content-based book recommending using learning for text categorization, in: Proceedings of the fifth ACM conference on Digital libraries, DL '00, ACM, New York, NY, USA, 2000, pp. 195–204.

[15] Y. Koren, R. M. Bell, Advances in collaborative filtering., in: F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011, pp. 145–186.

[16] T. Q. Lee, Y. Park, Y.-T. Park, A time-based approach to effective recommender systems using implicit feedback, Expert Systems with Applications 34 (4) (2008) 3055 – 3062.

[17] Y. Koren, Collaborative filtering with temporal dynamics, Commun. ACM 53 (4) (2010) 89–97.

[18] S. Wu, C. Tan, J. M. Kleinberg, M. W. Macy, Does bad news go away faster?, in: ICWSM, 2011.