Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 13, no. 7, November 2017, pp. 1077-1086
DOI: 10.23940/ijpe.17.07.p10.10771086

Micro-blog Real Time Personalized Recommendation based on Partial Indexing

Dun Li, Meng Wang, Lun Li, Zhiyun Zheng*

School of Information Engineering,Zhengzhou University,Zhengzhou, China

Abstract

Micro-blog is a new social networking service platform and users are very concerned about real-time personalized information. However, the existing micro-blog platform does not fully consider the user's real-time personalized demands. The paper proposes a micro-blog real-time personalized recommendation model. We constructed partial index mechanism to maintain the latest release or update of micro-blog, and inferred the topic distribution of micro-blog and user interest vector based on the LDA model to meet the real-time personalized demands of users. Experimental results on real datasets show that the proposed method is real-time and effective.

Keywords: micro-blog; real-time recommendation; personalized recommendation; partial indexing

(Submitted on July 25, 2017; Revised on August 30, 2017; Accepted on September 15, 2017)

(This paper was presented at the Third International Symposium on System and Software Reliability.)

1. Introduction

Social Networking Services (SNS) has become an important way for people to interact, publish and query real-time information such as Facebook, Twitter, YouTube and Sina micro-blog. Users can publish, repost, comment, and like micro-blog. They are concerned about the real-time personalized micro-blog information. However, the huge number of Micro-blog information and continuous updates makes it difficult to find real-time information of interest. Therefore, real-time and personalized micro-blog recommendation is particularly important to users.

However, it is a huge challenge to take into account the real needs of users and personalization in the massive micro-blog information processing. In recent years, scholars in the micro-blog personalized recommendation and real-time response have been a popular area of research. [3,7,8,11,12] recommended the Micro-blog personally combining the user's impact and the popularity, but they did not take into account the real-time, usually provided the outdated information to the user. In real time of micro-blog, [1,2,4,9] considered the micro-blog release time and the return results to the real-time update in the social network, but these works did not fully consider the user needs of diversified personality and received the limited quality of service as a result.

In order to solve these problems, the paper combines indexing technology with the topic model LDA, and puts forward the micro-blog local indexing mechanism. It uses the topic model LDA to deduce the distribution of micro-blog topic, and describes the user's interest with the user interest similarity and user interaction trust. This work is faced with two major challenges: (1) the rapid release rate and the huge amount of data, which make it necessary to organize the updated data effectively, and (2) real-time and accurate response to user personalized service requests.

The rest of this paper is structured as follows. In Section 2, we review the techniques related to social networking services, and section 3 discusses the local indexing mechanism in detail. It uses the LDA model to infer the user's interest to personalize the ranking, and describes how to give users real-time recommendation of their interest in micro-blog. In section

* Corresponding author.
E-mail address: iezyzheng@zzu.edu.cn

4, we use the real data set to evaluate the local index scheme in the micro-blog system in real time personalized recommendation performance.

2. Related work

2.1. Real-time search service

In recent years, the development of social networking services is very rapid. In order to enable users to find real-time updates of information quickly, real-time search service was introduced to the micro-blog platform.

In [5], Gao et al. proposed a real-time recommendation method for micro-blog users. It built the stream data model to represent the massive micro-blog publishing and updating, uses the real-time top-k query based on the sliding window model to respond to the real-time micro-blog recommendation, and recommends the micro-blogs to users according to the user's main interests in the real-time hotspots. However, the main interests could not accurately reflect all the user's interests, and the main preferences of the recommendation could not fully meet the user's individual demands. In [6], Li et al. proposed two general frameworks to perform real-time personalized top-k queries based on two characteristics of social networks: frequent content updates and small world phenomena. They designed the three-dimensional cube index, and sorted the results based on a combination of micro-blog release time, social relevance and the sort function of text similarity. But the sorting function of the user's personalized service is not sufficient. In [7], Chen et al. studied the real-time retrieval problem on Twitter. They grouped the micro-blog based on the probability of micro-blog, optimized the Twitter index, and sorted the results of the user's interest combining the time factor with the user's influence. However, the return results were too solitary to take into account the diversification of user interests. In [10], Earlybird (Twitter's real-time search service) sorted the query results according to the release time of each Twitter. However, sorting only considered the time dimension and resulting in some users interested in Twitter being not recommended.

2.2. Social network personalized recommendation

With the extensive application of micro-blog, there are more and more researches on recommendation in the micro-blog system. Friends' personality recommendation system research has aroused great concern.

In [1], Chen et al. built the Collaborative Ranking model for Twitter personalized recommendation. The model took into account the relationship between Twitter content and users, and recommended personalized Twitter according to the micro-blog topic distribution factors, micro-blog content characteristics and user's influence characteristics in the social network. But the method did not make full use of users' interaction information so that the recommendation was not ideal. In [2], a heterogeneous collective network recommendation algorithm based on user clustering is proposed. The algorithm extracted the micro-blog content features, clustered the entire user group based on the content similarity, and then the clustering results were utilized to recommend the related friends. The algorithm reduces the impact of sparse and cold start problems to a certain extent on the recommended quality, but the user's interests were changing and the recommendation set was low. The algorithm did not propose an effective way to solve this problem and the user was less satisfied with the recommendation. [4] proposed the Learning to Rank (sorting learning) framework, selected three features, such as the length of the tweets, the quality of the text and the user influence, to sort the Twitter according to the query satisfaction. But the returned results could not be updated in real time to meet the demands of users' real-time personality. In [9], the recommendation algorithm of user similarity concept filtering was introduced. It improved the collaborative filtering algorithm, calculated the similarity between user attribute similarity and user interaction, obtained the absolute similarity of the user and recommended the users with the same interest. The algorithm was calculated based on micro-blog content similarity and the recommendation accuracy was not accurate enough.

In recent years, many scholars realized that a text may involve multiple potential topics at the same time, and each topic usually involved a series of specific vocabularies. Thus, some probability generation models on texts were proposed to reveal the potential topics of the texts, in which LDA was widely studied as an unsupervised probability generation model of the text topic, but this work did not take into account the user's real-time requirements.

In summary, the traditional recommendation method only took into account the user micro-blog content similarity and micro-blog release time. But micro-blog is a kind of typical social network services and reflects the characteristics of current social network services such as massive, real-time and personalized features. The user's personalized service in the micro-blog platform highlights the inadaptability of the traditional recommendation system in the social network services.

3. Real-time personalized micro-blog recommendation

The micro-blog real-time personalized recommendation framework based on the local index mechanism is shown in Figure 1. It contains four parts: data preprocess, index, personalized sorting and real-time recommendation.

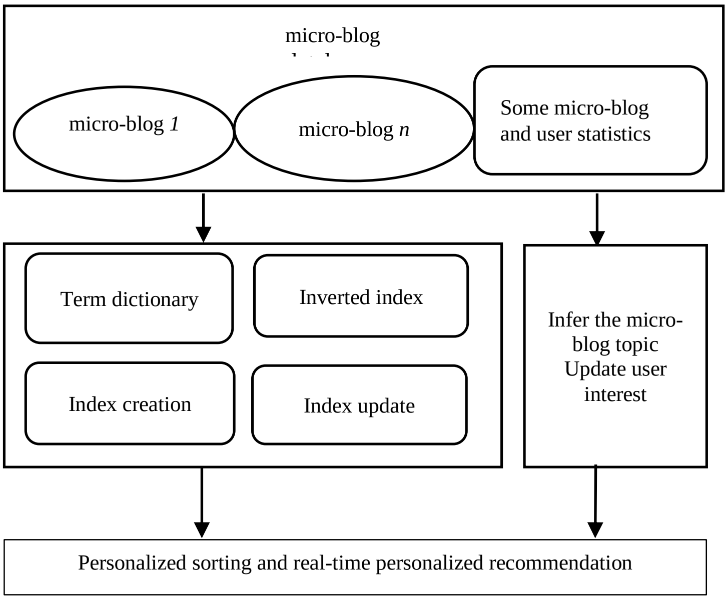


Figure 1. The micro-blog real-time personalized recommendation framework

a) Data preprocess

The data preprocess collects explicit information (such as micro-blog content, tags) and implicit information (such as user relationship information, interactive information) on the micro-blog platform. And then, it calculates the released-newly micro-blog and some of the user's features that imply the user's potential interest or current demands. User information is divided into two categories: the user's social relations information (including attention, activity) and micro-blog interactive information (including the number of times to be forwarded, the number of comments, the number of points).

b) Index

Index is to organize the mass and updated-rapidly micro-blog information, and improve the efficiency of the search micro-blog. The paper presents a micro-blog Partial Indexing Mechanism, which maintains the latest or updated micro-blog.

c) Personalized sorting

For the micro-blogging system social characteristics and media characteristics of this unique nature, personalized sorting takes into account the content characteristics of micro-blogs and the user's interaction characteristics. Personalized sorting organizes the user's micro-blogs into a collection of documents. The topic model is used to build the user's interest model, which draws the user's thematic feature vector. Interaction between users is used to calculate the interaction trust between micro-blog users and synthesize the user's interest vector. Then, the topic distribution of the micro-blog is introduced, by calculating the probability distribution of each keyword in the micro-blog on the subject; finally, the calculation method of micro-blogging and user interest vector similarity in index is introduced.

d) Real-time recommendation

Based on user-specific interest, real-time recommendation selects the corresponding micro-blog from the index list. Then it sorts them according to the micro-blog personalized recommendation sorting scheme to their popularity. Finally, it selects the micro-blog users which may be interested in the top-k popular micro-blog and builds the user personalized recommendation list.

3.1. User's interest and micro-blog topic distribution

1) User topic feature vector

Each micro-blog usually contains one or more implied topics, and a particular topic has a corresponding relationship with a particular term. The main method used to mine the underlying topic of corpus is the LDA subject probability model. In this paper, the LDA topic model is used to model the micro-blog content, and then the distribution probability of the subject characteristic words is calculated. Finally, the topic distribution vector of micro-blog is obtained. This paper calculates the correlation between user interest vector and micro-blog subject distribution vector, and proposes a personalized recommendation scheme based on micro-blog.

a) Micro-blog user interest calculation

Let C denotes a predetermined set of T topic, the topic set C={C_1,C_2， …， C_T}, t is a micro-blog, $P(C_i \mid t)$ denotes the t Twitter belonging posterior probability of C_i topic. The higher the value of $P(Ci \mid t)$ is, the higher the probability that the micro-blog t belongs to the subject C_i is.

Assuming that a user has published a micro-blog, let {t_1， t_2， …， t_d} denote a micro-blog collection published by a user, then the subject feature vector of the user can be represented by a T-dimensional vector $V_u=(v_1, v_2, \ldots, v_T)$description.

$$v_i=\frac{1}{d}\sum_{j=1}^{d} P\left(C_i|t_j\right) \tag{1}$$

When the user has not yet released any micro-blog, the user's topic feature vector is 0 vector. The larger the component v_i in the thematic feature vector indicates that the user is more interested in the subject C_i.

b) Micro-blog user trust calculation

① The degree of concern to users

The user's level of interest $W_f(u_o)$ is calculated in formula (2):

$$W_f(u_o)=\frac{|followers(u_o)|}{|followees(u_o)|} \tag{2}$$

Where $W_f(u_o)$ represents the ratio of the number of fans to the number of users concerned, followers(u_o) represents the number of fans of user u_o, and followees(u_o) indicates the number of objects of user u_o.

② The number of concerned objects in the fan users

The number of concerned objects in the fan users u_t is calculated in formula (3):

$$W_o(u_o)=|\text{followees}(u_t)\cap\text{followers}(u_o)| \tag{3}$$

Where followees(u_t) represents the object of the user u_t , followers(u_o) is the user u_o fans.

③ Tthe number of share friends to different users

The number of share friends concerned by user u_t and u_o is shown in formula (4):

$$W_t(u_o)=|\text{followees}(u_t)\cap\text{followees}(u_o)| \tag{4}$$

Where followees(u_o) represents the concerning object of the user u_o, and followees(u_t) represents the concerning object of the user u_t. Obviously, the more the number of users concerned with the object u_o and u_t, the greater the similarity between them.

④ Interactive trust between users

Let user set U={u_1,u_2 , …,u_n}, interactive trust of user u_o based on the user u_t is shown in the formula (5):

$$R(u_o)=\alpha W_f(u_o)+\beta W_o(u_o)+ \gamma W_t(u_o) \tag{5}$$

The weights α, β and γ are the most important model parameters in the trust calculation. The linear regression model can obtain the optimal solution for α, β and γ. In this paper, (5) is used as a linear regression model to evaluate the model parameters. The dependent variables of the linear regression are $y=R(u_o)$, and the three independent variables are $x1=\alpha(u_o)$, $x2=\beta(u_o)$ and $x3=\gamma(u_o)$. The coefficients corresponding to the independent variables are α, B and γ. Assuming that there are *n* sets of training data, the linear regression formula (6) can be derived from formula (5):

$$\begin{cases} y1=\beta 1x_{11}+\beta 2x_{12}+\beta 3x_{13} \\ y2=\beta 1x_{21}+\beta 2x_{22}+\beta 3x_{23} \\ \quad\quad\quad\ldots \\ yn=\beta 1x_{n1}+\beta 2x_{n2}+\beta 3x_{n3} \end{cases} \tag{6}$$

The group of observations was substituted into the linear regression model and the loss function (7) was introduced. The β1, β2, and β3 values corresponding to the minimum value of the loss function are obtained.

$$L(y,(x_1,x_2,x_3))=(y-f(x_1,x_2,x_3))^2 \tag{7}$$

c) User Interest Vector

According to the user's micro-blog content information and the trust between users, we improve the user interest model. User interest vector is shown in formula (8):

$$M_u = V_u * R(\mathrm{u_o}) \tag{8}$$

M_u is the user interest vector, which consists of two parts: the user's topic feature vector V_u and the social interaction based on the trust R (u_o).

2) Micro-blog topic distribution

We suppose that micro-blog *t* is composed of *n* words, denoted as $\{w_1, w_2, \ldots, w_n\}$. Let the random variable Z_{wi} denote the subject of the word w_i. The probability of the word w_i, Z_{wi} = j (the word wi belongs to the j-th topic) in the micro-blog *t* is given as follows:

$$P\left(Z_{wi} = j \middle| Z_{t,\neg i}, t, \varphi, \varepsilon\right) \propto \frac{P(Z_{wi} = j, Z_{t,\neg i}, t|\varphi, \varepsilon)}{P(Z_{t,\neg i}, t|\varphi, \varepsilon)} = \frac{n(j,t)+\varepsilon-1}{n(t)+T\varepsilon-1} \times \varphi_{wi}^{j} \tag{9}$$

Where ε denotes the parameter of the priori distribution to the user's interest, φ_{wi}^{j} denotes the probability that the word w_i belongs to the j-th topic, and $Z_{t,\neg i}$ denotes the subject of all the words except the subject of the i-th word (N, t) in the micro-blog t, n(j, t) represents the number of words belonging to the j-th topic in the micro-blog t, and the subject distribution based on the word w_i of the formula (10) S_{wi}= $(s_1, s_2, ..., s_T)$, where the component s_j is normalized in the probability (10).

$$s_j = \frac{P(Z_{wi}=j|Z_{t,\neg i},t,\varphi,\varepsilon)}{\sum_{j=i}^{T} P(Z_{wi}=j|Z_{t,\neg i},t,\varphi,\varepsilon)} \tag{10}$$

The subject of the word w_i is sampled from the distribution $S_{wi} = (s_1, s_2, ..., s_T)$, and the probability θ of the micro-blog belonging to the j-th topic can be estimated as:

$$\theta_j = \frac{n(j,t)+\varepsilon}{n(t)+T\varepsilon-1} \tag{11}$$

Finally, the micro-blog topic distribution is described as $P_t=(\theta_1, \theta_2, \ldots, \theta_T)$.

3) User interest orientation and micro-blog subject distribution

After the similarity calculation, we can sort the micro-blog and the similarity of interest in the index list. Assuming that t_1 and t_2 represent the two micro-blogs in the index list, Sim (P_t, M_u) represents the similarity between the user vector and the micro-blog topic vector as formula (4-12). If the user interest vector and micro-blog t_1 topic distribution are similar, it is easy to know that the user may be more interested in micro-blog t_1.

$$\mathrm{Sim}(P_t, P_u) = \frac{P_t.P_u}{|P_{t|} \times |P_{u|}} \tag{12}$$

Where P_t represents the subject distribution vector of micro-blog in the index list, and M_u represents the user's interest vector. The micro-blog and user interest similarity are evaluated by calculating the cosine of the two vectors.

3.2. Local index and real-time personalized recommendation

1) Index structure

Creating index is the most efficient way to retrieve data. However, the existing index cannot be directly applied to the micro-blog system. As the recently released or updated micro-blog is very popular, attention will continue to decline when time goes on. Therefore, this paper proposes a local indexing scheme to maintain the latest releases or updates of micro-blog. The local index is processed by the inverted index. We determine whether it is the latest release of micro-blog for the new arrival of micro-blog according to the micro-blog score function, and update the index list in real time. The local index structure has two main parts: the Term dictionary and the inverted index.

a) The Term dictionary

The Term dictionary is a simple hash table, that is, according to a given term, find the list of inverted document IDs for term. In order to save space, each term is assigned a unique and increasing-monotonically ID as key, the value of the data contains: record some information about the word itself and pointers to *reverse listing*.

b) Inverted index

The inverted index is almost the core of search engines, users are interested in micro-blogs that are saved in memory and modify the structure of inverted index. Given a Term, the inverted index returns a list of the corresponding micro-blogs and Figure 2 shows the inverted index structure. For each record in the index, the different micro-blogs are determined by ID. To facilitate sorting, we save a micro-blog score and timestamp. The record of the same keyword is saved in the same list, and the latest record is inserted into the list header according to the time stamp of the micro-blog.

ID	SCORE	TIME

Figure 2. Inverted index structure

2) The construction of the index

Lucene is a high-performance and full-text search open source engine architecture that provides a complete query engine and index engine, the index architecture of this paper is created on the basis of Lucene.

There is an assumption that the user is only interested in top-K results, and it can be easily verified by the search engine's statistics. Each index refers the micro-blog of interest to the user. During query processing, we only need to check the associated indexes to find the results. Different users have different concerns about hot events, some may focus on public events, some may only focus on the recent events. This paper uses the micro-blog scoring function to represent the popularity of micro-blogs. In general, the higher the number of micro-blog is forwarded, commented, praise, the higher the popularity of micro-blog. So the calculation formula of the micro-blog scoring function is shown in (13):

$$Score(t)=num_R+num_C+num_L \tag{13}$$

Where num_R indicates the number of micro-blogs commented, num_C indicates the number of micro-blogs being forwarded, num_L represents the number of points is micro-blogs being liked.

For a newly published micro-blog, the index construction is described as algorithm 1. M indicates the maximum number of micro-blogs in the index list, n indicates the number of micro-blogs in the index list, the initial value is 0, MIN_SCORE records the lowest score of the M micro-blog in the index list I. In the 6th line,2 the new release micro-blog scores are calculated. If the number of micro-blogs in the list is less than K, the MIN_SCORE value is zero. When there are exactly K micro-blogs in the list, MIN_SCORE updates the lowest score for these K micro-blogs and the algorithm terminates.

Algorithm 1. *Initializing partial indexing algorithm*

```
Input: Scoring function Score(t)
Output: Index list I
1.  Initializes the index list I to Null, MIN_SCORE is 0
2.  FOR(n=1; n<=M; n++)
3.  IF (a new micro-blog arrives) THEN
4.  { Add new micro-blog to I
5.  The function Score (t) of the call function is calculated as：SCORE
6.  IF (n=1) THEN
7.  MIN_SCORE=SCORE
8.  ELSEIF (MIN_SCORE>SCORE)
9.  MIN_SCORE=SCORE}
10. END FOR
11. RETURN I
```

3) Index updates

After initializing the inverted index, with the time goes on, the popularity of the micro-blog has been changing and the micro-blog version is also being updated. The index list should also be updated in real time. The update of the index list is described in algorithm 2.

Algorithm 2. *Update algorithm of index list I*

```
Input: Index List I
Output: After the index list I is updated
1.  WHILE (a new micro-blog arrived)
2.  Use the function Score (t) to calculate the micro-blog score
3.  IF (Score(t)>MIN_SCORE) THEN
4.  Update the micro-blog to I
5.  Delete the corresponding micro-blog from the inverted list；SORT(I)
6.  Update the MIN_SCORE
```

4) Recommendation of the users' interested micro-blog

Algorithm 3 describes how to choose the first k micro-blog which the user is most interested from the index list. At first, the index list I, micro-blog set S waiting for the update and the current personalized micro-blog recommendation list L (u) of the user u are inputted. When L (u) is empty, the first line calculates the interest degree of user u and the interest similarity of each micro-blog in I, and returns the k-like micro-blog with the highest similarity to user u. If L (u) is not empty, exp is marked as the expired micro-blog in L(u). The 5th line calculates the user u interest orientation and the similarity between each micro-blog topic distribution in S, replaces the micro-blog of less similarity with those of larger similarity after removing the expired micro-blogs, and gets the new list marked as L '(u). If there is no updated micro-blog, the 6th lines would calculate the user u interest and each micro-blog interest similarity after removing the expired micro-blogs in I, and return k micro-blogs of the most similarity to the user.

Algorithm 3. *Real-time personalized micro-blog recommendation algorithm*

```
Input: Index List I
       To be updated micro-blog set S
       User u top-k interest micro-blog list L (u)
Output: Update User u Personalized Recommendation List L '(u)
1.  IF (L(u) is empty) THEN
2.    The k micro-blog of the highest similarity to user u interest in L(u) are added to the L '(u)
3.  ELSE
4.    exp is an expired micro-blog in L (U)
5.  IF (S is not empty) THEN
6.    replaces the micro-blog of less similarity with those of larger similarity in S, marked as L'(U)
7.  ELSE
8.    the k micro-blogs of the most similarity to user interest in exp are added to L '(u).
9.  RETURN L′(u)
```

4. Experimental analysis

4.1. Experimental environment and data set

In order to verify the real-time performance and validity of our method, the real user data in micro-blog platform is selected as the experimental objects. Sina micro-blog is the most influential and the most watched micro-blog service platform; users can publish, forward, comment, like the micro-blog, and so on. Users in different areas are selected as the target users adopting the breadth priority algorithm. 30,000 user data was collected. In order to improve the accuracy of the experimental algorithm, we clean up and integrate the data set for pre-processing, get 18 385 user data, including 2 345 824 micro-blog and 1 543 754 interactive information. In this paper, we use the crossover experiment to divide the data set into 80% training set and 20% test set and carry out several experiments to verify the algorithm.

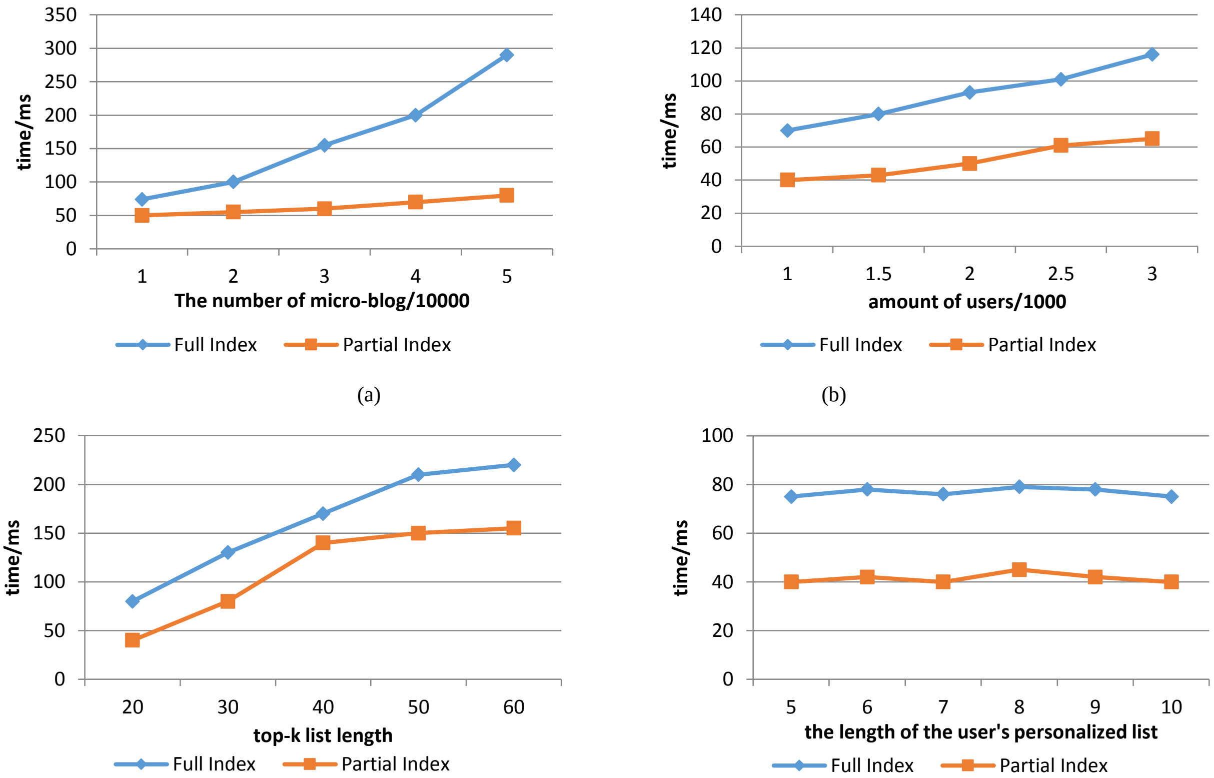
4.2. Evaluation parameters

In order to evaluate the personalized micro-blog real-time recommendation system, the two most important criteria is recommended efficiency and recommended effectiveness. We designed a benchmark method to validate the efficiency and effectiveness of the local real-time personalized recommendation method. Instead of creating an index for the micro-blogs data set, the time order is considered, not the user's interest, in the recommended micro blog phase. The top k micro-blogs are directly returned to the user in the global top-k list. The Full Index and Partial Index are used to represent the reference method and compared to our recommendation method based on the local index mechanism. The following measures are used to compare these two methods: (1) Time cost: measure the user's interest in micro-blog recommended method of time cost to verify the micro-blog recommended efficiency, (2) Memory cost: memory usage of a personalized recommendation method for local indexes, and (3) Mean of average accuracy: measure the mean of the average accuracy of personalized micro-blog recommendation.

4.3. Experiment and analysis

1) The efficiency of the real-time personalized recommendation method

In order to measure the efficiency of the recommended method and record the time cost of each list update, we use the updated time cost to measure the efficiency of the Partial Index method and the Full Index method.



(a) (b)

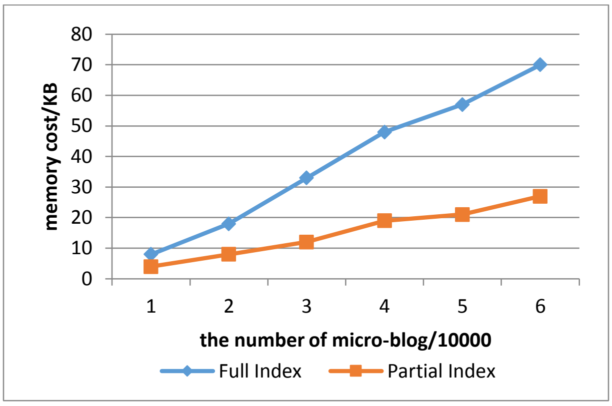
(c) (d)

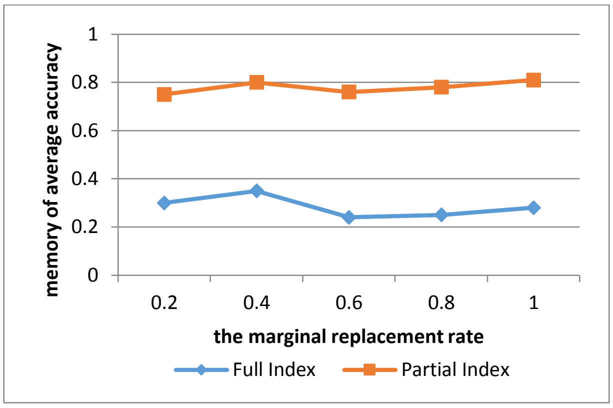
Figure 3. The time cost of the real-time personalized recommendation method

Figure 3 compares the time cost of the Full Index and Partial Index methods by changing some parameters. The horizontal axis in Figure 3(a) shows the change in the number of users, and the vertical axis is the time cost in milliseconds. It can be observed from the figure that as the number of user increases, the list of recommended micro-blogs is increasing, resulting in an increase in the time cost of the recommended method. The horizontal axis in Figure 2(b) represents the number of micro-blogs per second, and the vertical axis is the time cost measured in milliseconds. It can be seen from the figure that, with the increase of the number of micro-blogs updates, the frequency of micro-blogs updated in the global top-k list is also increasing, and leads to the increasing time cost of the recommended method. It can be seen that the Full Index method is much more efficient than the Partial Index method. The horizontal axis in Figure 3(c) shows the change in the length of the global top-k list. It can be seen that the length of the global top-k list increases as the number of return results to the user increased, and the number of micro-blogs to be indexed increases, resulting in the increase in the time cost of the recommended method. It can be seen that the Full Index method is much more efficient than the Partial Index method. The horizontal axis in Figure 3(c) shows the change in the length of the global top-k list. It can be seen that the length of the global top-k list increases as the number of return results to the user increases, and the number of micro-blogs to be indexed increases, resulting in an increase in the time cost of the recommended method. Figure 3(d) measure the time cost of the personalized recommendation by changing the length of the user's personalized recommendation list. The horizontal axis indicates the length of the user's personalized micro-blog list, and the vertical axis is the time cost in milliseconds. As it can be seen from the figure, since the algorithm maintains only a few popular micro-blogs, the comparison frequency of micro-blog ranking is basically unchanged; as the length of the user recommendation list continues to increase, the time cost of the recommended method is almost constant.

2) The effectiveness of the real-time personalized recommendation method

In order to verify the effectiveness, the average accuracy rate is measured by changing the marginal replacement rate of the score function, and the memory cost in different micro-blogs is tested.





(a) (b)

Figure 4. The effectiveness of the real-time personalized recommendation method

Figure 4(a) shows the change in the mean accuracy of the five recommended results when the marginal substitution rate of the scoring function changes. It is obvious that the micro-blog in the global Top-k list may not be same. The change of marginal substitution rate leads to the different results of the final recommendation. However, the Partial Index method is much more effective than the Full Index method according to the mean of the average accuracy rate. Figure 4(b) shows the memory cost comparison of the two methods. The horizontal axis represents the number of micro-blogs per second, and the vertical axis is the memory cost in KB. It is easy to find that memory cost increase with the increasing number of updated micro-blogs. The PI method optimizes the inverted index structure, and only stores the latest and most relevant micro-blogs in memory, which improves system throughput and memory change trends are relatively stable.

5. Conclusions

In this paper, a new index and personalized ranking method is proposed to support real-time personalized recommendation of micro-blog system that adopts micro-blog local index mechanism to optimize the inverted index structure. The main topic distribution and user interest of micro-blogs are pushed based on the topic model of LDA. Lastly, it recommends the micro-blog that users are most interested in the global Top-k list to accomplish the real-time personalized recommendation in micro-blog platform.

The contextual and emotional analysis of the user and the recommended object will have an impact on the recommended quality, contextual information analysis can be more accurately inferred the following, and we can get more accurate user's interest through the sentiment analysis in micro-blog. In future work, we will focus on how to add contextual information and emotional information to the recommendation methods.

Acknowledgments

The authors are grateful to the editors and reviewers for their helpful comments and suggestions. This research is partially supported by National Social Science Foundation project (17BXW065), Science and Technology Research project of Henan province (172102310628, 162102310616) and Science and Technology Research project of Zhengzhou (141PPTGG368).

References

1. K. Chen, T. Chen, G. Zheng, et al. "Collaborative Personalized Tweet Recommendation". *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012, 661-670
2. K. Chen, P. Han, J. Wu. "User Clustering Based Social Network Recommendation". *Chinese Journal of Computers*, 2013, 36(2):349-359
3. W. E. Diewertab, K. J. Foxb. "Decomposing Productivity Indexes Into Explanatory Factors". *European Journal of Operational Research*, 2017, 256(1):275-291
4. Y. Duan, L. Jiang, T. Qin, et al. "An Empirical Study on Learning to Rank of Tweets". *Proceedings of the 23rd IEEE, International Conference on Computational Linguistics*.，2010，295-303
5. M. Gao，C. Jin，W. Qian et al. "Real-time and Personalized Search over a Micro-blog System". *Computer Journal*, 2014, 57(9):1281-1295
6. Y. Li，Z. Bao，G. Li，et al. "Real Time Personalized Search on Social Networks". *IEEE International Conference on Data Engineering*, 2015，639-650
7. D. Nie, Y. Fu, J. Zhou, et al. "A Personalized Recommendation Algorithm via Biased Random Walk". *2014 11th International Joint Conference on Computer Science and Software Engineering*, 2014. 292-296
8. D Ramage, S Dumais, et al. "Characterizing Micro-Blogs with Topic Models". *Proceedings of the International AAAI Conference on Weblogs and Social Media*, AAAI, 2010,130-137
9. H. Rong, S. Huo, C. Hu, J. Mo. "User Similarity-Based Collaborative Filtering Recommendation Algorithm". *Journal on Communication*, 2014, 35(2):16-24
10. J. Xu, T. Xing, M. Schaar. "Personalized Course Sequence Recommendations". *IEEE Transactions on Signal Processing*, 2016, 160-177
11. J. Yao, B. Cui, Z. Xue. "Provenance-based Indexing Support in Micro-blog Platforms". *IEEE 28th International Conference on Data Engineering*, 2012, 558-569
12. R. Zhong, G. Li, K. Tan, et al. "G-tree: an Efficient Index for KNN Search on Road Networks". *ACM International Conference on Conference on Information & Knowledge Management*, 2013, 39-48