Mining User Interests for Personalized Tweet Recommendation on Map-Reduce Framework

Guanyao Du[1], Jianying Sun[1], Xinglong Huang[2] and Jianjun Yu[1]
[1] *Computer Network Information Center, Chinese Academy of Sciences, 4 Zhongguancun Nansijie, Haidian District, Beijing 100190, China*
[2] *MailBox No.28, 87th Xiangshannanlu, Beijing 100093, China*

Keywords: Recommender System, User Interests, Topic-STG, SVD, Big Data, Map-Reduce.

Abstract: The tremendous growth of micro-blogging systems in recent years poses some key challenges for recommender systems, such as how to process tweet big data under distributed environment, how to striking a balance between high accurate recommendations and efficiency, and how to produce diverse recommendations for millions of users. In our opinion, accurately, instantly, and completely capturing user preferences over time is the key point for personalized tweet recommendation. Therefore, we introduce three features to model personal user interests and its evolution for tweet recommendation, including textual information, user behaviors, and time. We then offer two enhanced recommendation models: Topic-STG (Session-based Temporal Graph) model and SVD (Singular Value Decomposition) model, combining these features to learn user preference and recommend personalized tweet. To further improve the algorithm efficiency for micro-blogging big data, we provide the parallel algorithm implementation for Topic-STG and SVD models based on Hadoop Map-Reduce framework. Experiments on a large scale of micro-blogging dataset illustrate the effectiveness of the proposed models and algorithms.

1 INTRODUCTION

As a convenient communication means, especially with smart phones, the micro-blogging systems not only act as the role of social relation between people, but also as important sources for people to obtain useful information. Currently, there are more than 400 million messages generated on Twitter from 500 million users, and 100 million Chinese messages on Sina Weibo (a Chinese Twitter) each day from 249 million users. Such enormous users ceaselessly chase and produce a large amount of information. It benefits the users but also can flood users and hence puts them at the risk of information overload.

Recommender system is a powerful tool to address the information overload problem (Xavier and Justin, 2016; Roberto et al., 2016). Much previous work has been proposed to recommend different objects on micro-blogging systems in recent years. Some of them investigated the content-based recommendation approaches, which were the fundamental mechanisms on micro-blogging systems, and were easy to be applied for hashtag recommendation (Hannon et al., 2010). Others considered the recommendation in social networks, in which the social structure model was very useful and helpful (Yigita et al., 2015). Currently, combining several features and correspondingly providing linear recommendation models is also a prevailing way on tweet recommendation (Yin et al., 2015). To easily incorporate information such as temporal dynamics, neighborhood relationship, and hierarchical information for recommendation, SVD series (Koren, 2010), SVDFeature (Chen et al., 2012), and other approaches (Jiang et al., 2014) provided scalable framework to efficiently solve large-scale collaborative filtering problems with auxiliary information using matrix factorization techniques for recommender systems.

Based on the probabilistic matrix factorization technique (Salakhutdinov and Mnih, 2007) which can offer a uniform and scalable framework to model explicit user interests, some significant work were presented to combine social factors (such as personal interest, interpersonal interest similarity, and interpersonal influence) together, and fused them into a unified personalized recommendation model. However, sometimes the user interests may

Du, G., Sun, J., Huang, X. and Yu, J.
Mining User Interests for Personalized Tweet Recommendation on Map-Reduce Framework.
DOI: 10.5220/0006274102010208
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 1*, pages 201-208
ISBN: 978-989-758-247-9

be implicit (like our circumstance), and should be inferred from user behaviors (such as @, hashtags, or retweeting). Also sometimes it's difficult to model the short-term and long-term user interests. For example, in (Xiang et al., 2010), Xiang et al. proposed a novel recommendation approach named STG to model users' long-term and short-term preferences over time, whereas, they neglected the textual information of tweets, which presented rich sentiment information to predict user preference. To solve this problem, (Yu and Shen, 2014; Yu et al., 2014; Yu and Zhu, 2015) involving tweet recommendation summarized three features to model user interests, and constructed the hybrid model or Topic-STG model to learn user preference for tweet recommendation consequently. In this paper, we would enhance the SVD model mainly to model explicit user interests, and provide the Topic-STG model for implicit and temporal tweet recommendation.

At the same time, recommending tweet based on massive micro-blogging datasets is a typical “Big Data” application since the recommender models are computation-intensive and time-consuming. Furthermore, the capability of micro-blogging datasets is ever-increasing that the traditional approaches would be difficult to process such large datasets. A parallel version for recommender models is expected since most of big data applications are developed with cloud computing technique which enables convenient, on-demand network access to a shared pool of configurable computing resources. As a result, a new platform of “Big Data” tools has arisen to handle sense making over large quantities of data, as in the Apache Hadoop. With Hadoop parallel Map-Reduce framework, recommender algorithms can be distributed in different computers to accelerate computation.

“Big Data” is a new term used to identify datasets that cannot be managed with current methodologies or data mining software tools due to their large size and complexity (Fan and Bifet, 2012; Kumar et al., 2013; Moens et al., 2014). Specifically, Kumar et al. gave a Hazy system to build and maintain big-data analytics with the latest statistical and machine-learning techniques (Kumar et al., 2013). Moens et al. introduced Frequent Itemset Mining (FIM) approaches on the Map-Reduce platform balancing data distribution and inter-communication costs (Moens et al., 2014).

As for the “Big Data” in domain applications, Roy et al. developed a system for end-to-end processing of genomic data, including alignment of short read sequences, variation discovery, and deep analysis (Roy et al., 2012). Chawla et al. provided a “Big Data” driven approach towards personalized healthcare, and demonstrated its applicability to patient-centered outcomes (Chawla and Davis, 2013). Zaiane built an agent that recommended on-line learning activities or shortcuts in a course web site based on learners' access history to improve course material navigation as well as assist the online learning process (Zaiane, 2012). However, most of the previous work involving cloud computing architectures for “Big Data” solutions only focused on common data mining platform or common services for data processing, and considered less about data mining based on recommender system, especially for micro-blogging data. This motivated us to do this work to provide special mining processes for recommender techniques.

In this paper, we consider the tweet recommendation with three features: the tweet textual information, the user's behavior, and the time factor, and focus the “Big Data” problem with massive micro-blogging dataset under parallel computation model. The contributions can be summarized as follows:

(1) We offer two models for tweet recommendation by considering the long-term and the short-term aspects to extract the top N tweets, one is the extended Topic-STG (Session-based Temporal Graph) model (Yu et al., 2014), and the other is the SVD (Singular Value Decomposition) model.

(2) To further improve the algorithm efficiency, we develop the parallel versions of Topic-STG and SVD models under Hadoop Map-Reduce framework, and conduct comprehensive experiments to evaluate the two techniques on a real large dataset, i.e., Sina Weibo.

(3) The experimental results illustrate that the introduced strategies outperform the state-of-the-art approach by a wide margin. It also shows that the Topic-STG model is more suitable for short-term user interests mining when users' behavior is not easy to capture, whereas the SVD model has more advantages for long-term user interests mining with explicit ratings.

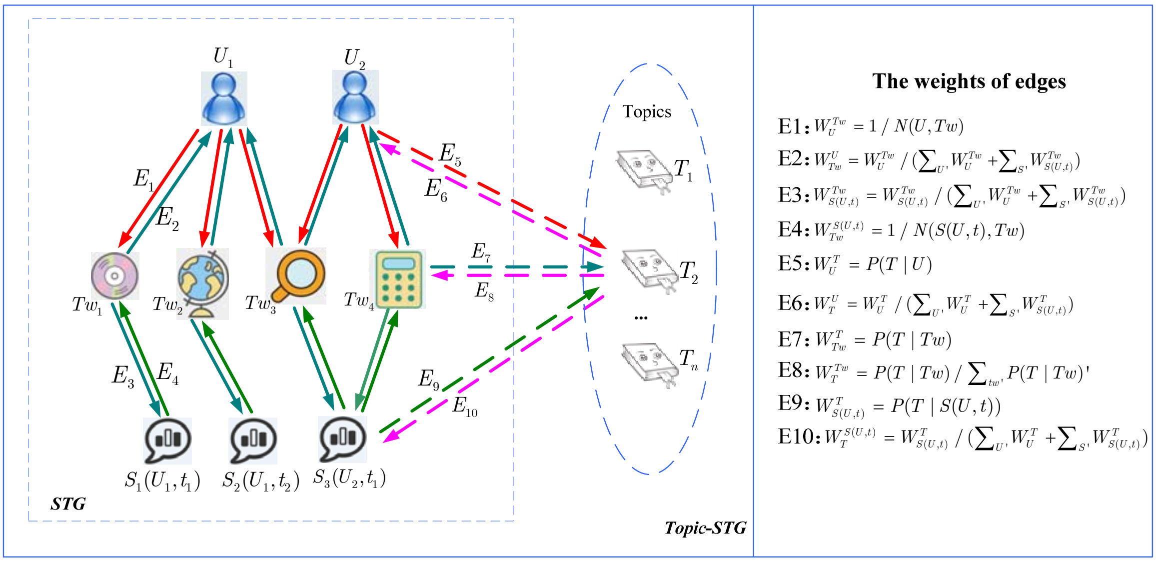

Figure 1: An example of Topic-STG and the weights of edges.

2 TWO ENHANCED RECOMMENDATION MODELS

2.1 Recommendation on Topic-STG Model

Topic-STG is an extended approach which bridges STG (Xiang et al., 2010) and textual information by adding a "topic-node"' into the existing bipartite graph.

Once a user visits or operates (e.g. "retweet", "comment", and "favorite") a tweet, he/she would show implicit interest to some extent. This implicit interest can be summarized as a "topic preference". Similar to STG approach, if a user U_1 operates on a tweet Tw_1, two pairs of edges (E_1 - E_4) which represent the connection between U_1 and Tw_1 with specific weights will be created as shown in Fig. 1. In Topic-STG, the topic-node is generated by LDA (Latent Dirichlet Allocation). We train the latent topics to infer tweets' topic distribution, long-term and short-term topic distribution of users, consequently six new correspondingly topic-related edges (E_5 - E_{10}) will be created which are the links between the topic-node and the corresponding user-node, tweet-node, and session-node. We should mention that a tweet may present several topics, here we just select the topic with the highest probability to present the implicit meaning of the tweet for simplicity. Given the weight of each edge in Fig. 1, we could recommend candidate tweets to a user *U* at a timestamp *t*. Due to the space limitations, the specific steps of the recommendation approach are omitted here, please refer to (Yu et al., 2014) and (Yu and Zhu, 2015) to get the details.

2.2 Recommendation on SVD Model

SVD is a matrix factorization technique commonly used for producing low-rank approximations, which produces results that are better than a traditional collaborative filtering algorithm most of the time when applied to the dataset with explicit ratings, such as movie, music ratings datasets. SVD also solves a general form of collaborative problems, and thus allows develop new models just by defining new features, which can easily incorporate information such as temporal dynamics, neighbourhood relationship, and hierarchical information into the SVD model.

SVD maps both users and items (tweets) to a joint latent factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space, which approximates user u's rating of item i, and can be denoted by $\hat{r}_{ui}$.

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u \quad (1)$$

Here, the observed rating is broken down into its four components: global average μ, item bias b_i, user bias b_u, and user-item interaction $q_i^T p_u$. SVD gives a basic model for producing product recommendations. We extend this model by considering three features of tweets.

As for textual information, there are two important sources to model user preference: 1) Hashtag, which covers almost the whole user interests; 2) the tweets' keywords, which can present the user interest on a special topic. Hence we get:

$$p_u^1 = W(u,kw)\sum_{kw\in K(u)} y_{kw}(kw) + |T(u)|^{-\frac{1}{2}} \sum_{n\in T(u)} y_{tg}(u) \quad (2)$$

$$q_i^1 = W(i,kw)\sum_{kw\in K(i)} y_{kw}(kw) + |T(i)|^{-\frac{1}{2}} \sum_{n\in T(i)} y_{tg}(i) \quad (3)$$

where $W(*,kw)$ gives the weights of keywords kw, $K(*)$ and $T(*)$ present the keywords and hashtags sets respectively.

As we observed, there are two types of operations on the micro-blogging systems: the attribute one and the non-attribute one. The attribute one is the operation that may affect user interests via the operation frequency, such as retweet, comment.

$$p_u^2 = 1/\left\|\alpha_u\right\|_2 \sum_{j\in A(u)} \alpha_{u,j}\cdot y_j \quad (4)$$

where $\alpha_{u,j}$ is the normalized operation count.

The non-attribute operation is the one that the operation frequency would not affect the user interests, such as favorite, add friends.

$$p_u^3 = \frac{1}{\sqrt{N(u)}}\sum_{j\in N(u)} y_j + \sum \lambda^{rd} \frac{1}{\sqrt{N(u)}}\sum_{j\in N(u)} y_j \quad (5)$$

Considering that a user's friends would also affect his/her interest, we define a decay factor λ to model this relationship, and set $rd \le 2$.

According to our previous work, we found that the short-term user interests may always change in one week, and the minimum unit of this change is one day. Thus the time factor can be modeled as follows:

$$b_{day} = \{b_{ti}\}, i \in [0,7) \quad (6)$$

Finally, we get an enhanced SVD model for recommendation:

$$\hat{r}_{ui} = b_{ui} + b_{day} + (q_i + q_i^1)^T (p_u + p_u^1 + p_u^2 + p_u^3) \quad (7)$$

3 PARALLEL COMPUTING IMPLEMENTATION

The two recommender parallel computing models are developed with Map-Reduce component, which is a programming model for processing large data sets and used to do distributed computing on clusters of computers. Since Map-Reduce is a common parallel computing model, we need to encapsulate data access interfaces and mining models to process key/value pairs. We acquire micro-blogging data stored in HDFS, and implement Map-Reduce functions separately. It needs to be emphasized that the mining algorithms are data-insensitive, which means that the map function can read each record randomly and do not affect the final result.

3.1 Topic-STG Map-Reduce Implementation

The Topic-STG parallel implementation based on Map-Reduce framework mainly develops map and reduce functions. The map function scans each records of micro-blogging datasets, calculates the score of each path of Topic-STG, and finally outcomes the preference of user on tweet. The reduce function summarizes all preferences, and ranks top *N* recommendation results. The details are described in **Algorithm 1.**

The sentence must end with a period. As we mentioned, Topic-STG would adopts LDA to generate topic node, so we need to realize LDA parallel version on Map-Reduce framework. The LDA parallel version would use Mahout toolkit since it builds a scalable machine learning library based on Hadoop.

Algorithm 1. Map-Reduce implementation for Topic-STG model.

1. **set** input file in HDFS and read into values
2. map(key, values, OuptputCollector output)
3. *{*
4. **for** score of each path: $\varphi(P)$ **do**
5. computing preference of u on tweet tw: p_U^{Tw} selecting shortest paths between U and Tw
6. output.collect(key,*{*values.user.ID, p_U^{Tw} *}*)
7. *}*
8.
9. reduce(key, values, OuptputCollector output)
10. *{*
11. **for** all users values.user.ID in *{values}* **do**
12. select Top *N* results from p_U^{Tw}
13. output.collect(key,*{*values.user.ID, p_U^{Tw} *}*)
14. *}*
15. **set** reduce result to output file

3.2 SVD Map-Reduce Implementation

Mahout provides a standard SVD implementation, we need to calculate a new $\hat{r}_{ui}$ with modified parameters: b_{ui}, b_{day}, q_i and p_u. The details are in **Algorithm 2**.

Algorithm 2. Map-Reduce implementation for enhanced SVD model.

set input file in HDFS and read into values
map(key, values, OuptputCollector output)
{
for each rating score **do**
calculating the difference between rating score and predict score: $e_{ui} = score - preScore$
using e_{ui} to update the increments of $b_u[uid], b_i[iid]$, $q_i[uid][k], p_u[iid][k]$
output.collect(*{uid,iid}*, flag(b_{ui} , b_{day} , q_i , p_u))
}

reduce(key, values, OuptputCollector output)
{
for each key in*{uid,iid}* **do**
$b_{ui} = \sum \mathrm{b_{ui}}$, $b_{day} = \sum b_{day}$, $q_i = \sum q_i$, $p_u = \sum p_u$
output.collect(*uid,iid*, b_{ui} , b_{day} , (q_i , p_u))
}
set reduce result to output file

4 EXPERIMENTAL EVALUATION

We crawled 811,586 original tweets with 100 initial users on Sina Weibo micro-blogging system after filtering out those inactive accounts and spammers to get a dense dataset from 2015/02/01 to 2015/11/01. There are 30,641 new-created tweets, 780,945 retweets, 1,852,562 comments, 68,327 favorites, 99,762 friendships, 41,127 hashtags, 19,023 2-degree users, 82,435 keywords in the collected dataset. We chose our training dataset from 2015/02/01 to 2015/07/31, and the remains as the test dataset. Considering the particularity of the Chinese micro-blogging system, we generate these Chinese terms from several basic corpuses, including Sogou Pinyin input dict, NLPIR micro-blogging corpus. Moreover, to avoid the possible bias of training user preference, we chose 30 users from our dataset, and "@" them with those personalized recommendations. These volunteers are active users in Weibo from different majors, jobs, and ages with different interest. The volunteers show their best effort helping to feedback whether they were really interested in the recommendations.

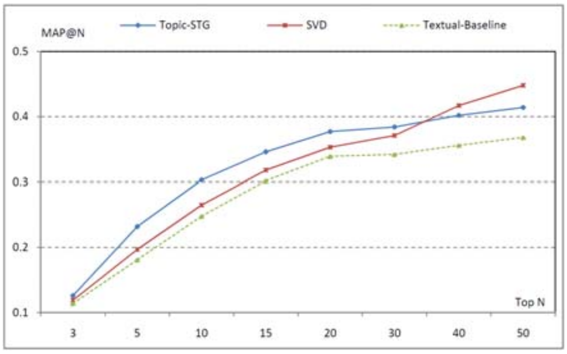

Figure 2: Tweet recommendation precision with different approaches.

When training the LDA parameters, we figure out 40-200 topics manually from the training tweets, and set the parameters $\alpha = 0.5$ and $\beta = 0.1$.

The Map-Reduce environment was constructed under a infrastructure with 128 virtual machines (each with 1 core of multi-core CPU, 16GB memory, 300G disk), 1PB distributed storage. We constructed two HDFS clusters for file-based micro-blogging datasets with 64 virtual machines respectively, one for Topic-STG, and the other for SVD model. We then built two parallel recommender models on virtual machines, and repeated each experiment with 100 times to calculate the average results.

4.1 The Precision of Recommendation

MAP@N (Mean Average Precision) evaluates the prediction accuracy of the top *N* recommendations for users, which is a popular rank evaluation method to evaluate the recommendation accuracy. We gave the *MAP@N* evaluation with the Topic-STG model, the SVD model, and a baseline--content similarity based approach. As shown in Fig.2, both the Topic-STG model and the SVD model outperform the baseline since we consider more features.

As we observed, *MAP@N* is positively correlated with the length *N* of the recommendations, and the *MAP@N* values of three approaches are very close when *N* is set to 50. We guess that though users have a wide range of interests, top 50 recommendations would cover almost all of the user interests. Also it means the time factor and the users' behaviors would reduce their effect when the recommendation list is enough long. Moreover, we found that the Topic-STG outperforms the SVD model when $N \in [3,30]$. Maybe it's due to the fact that the Topic-STG captures the long-term and short-term user interests better than the SVD's since users are always interested in those hot topics.

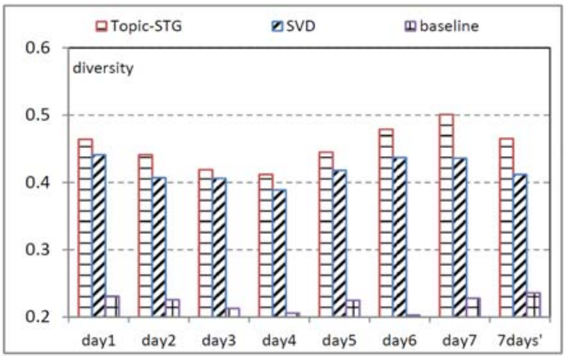

Figure 3: Diversity of top 20 recommendations.

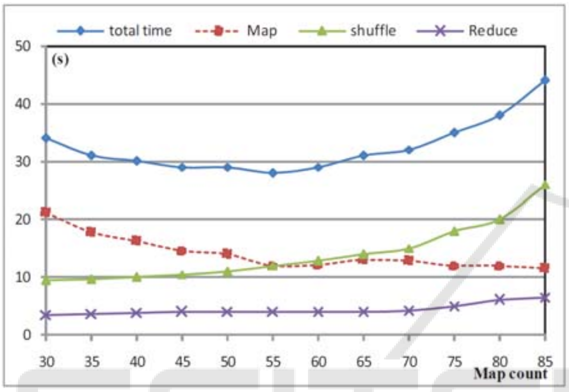

Figure 5: The time consumption comparison of different periods of Map-Reduce.

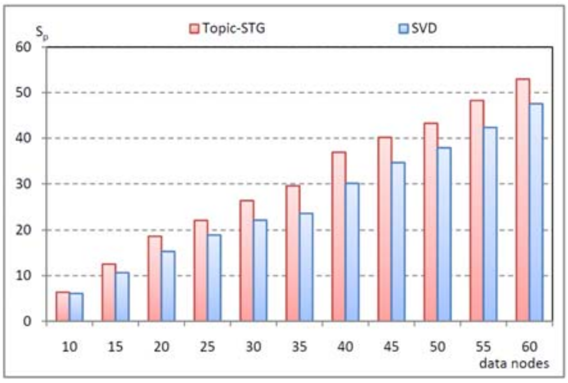

Figure 4: Efficiency comparison for the two Map-Reduce based models.

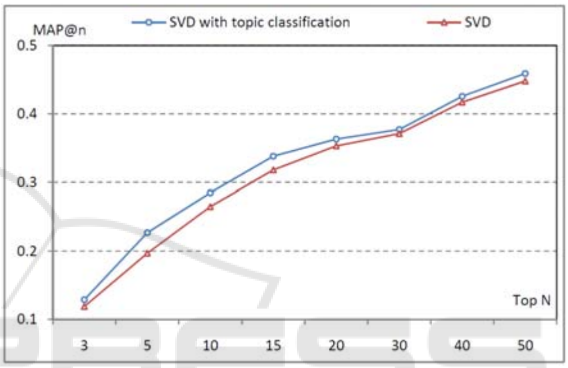

Figure 6: MAP@N comparisons with topic classification in SVD model.

Besides, Topic-STG considers the implicit tweet content to make the scores of the hot topics higher when the list is short. Whereas when the list becomes longer, i.e. $N > 30$, the effect of the hashtag textual feature is highlighted.

4.2 Diversity of Top N Recommendation

It often happens that the products on the recommendation list are highly similar to each other and lack of diversity. For example, when Tianjin Port explosion happened on Aug. 12, 2015, most of the recommended tweets were associated with the "Tianjin Port explosion" event. Consequently, those tweets contributed for long-term user interests would not be ranked for recommendation. Diversity would increase the probability of retrieving unusual or novel items which are relevant to the user. We use the metric introduced in (Hurley and Zhang, 2011) to evaluate the diversity.

We present the diversity results with the proposed approaches comparing with the baseline approach as shown in Fig. 3, and evaluate the diversity metric on Top 20 recommendation with 30 random users. Totally, the diversities of our proposed models are better than the baseline since both of them take into account the time factor. When it refers to 7 days' diversity, our models perform significantly better than the baseline. This is because that, for each day, our models give different recommendation results on different topics, which makes more diversity for 7 days totally, whereas the baseline approach uses the content similarity which would recommend relatively the same topics when time changes, and thus leads to fewer diversity for 7 days'. Also we find that Topic-STG's gets bigger variance than the SVD's, which means that Topic-STG shows more data fluctuation, and reflects that it's time-sensitive and better to learn short-term user interests.

4.3 Efficiency of Map-Reduce

We borrowed the concept of "speedup" to evaluate the efficiency of Map-Reduce based recommender models. In parallel computing, speedup refers to how much a parallel algorithm is faster than a corresponding sequential algorithm.

$$S_p = T_1 / T_p \tag{9}$$

where p is the number of map-reduce functions, T_1 is the execution time of the sequential mining algorithm, and T_p is the execution time of Map-Reduce based recommender models.

We present S_p efficiency comparison for Map-Reduce based Topic-STG and SVD models as shown in Fig.4. We set the experiment with data servers ranging from 10 to 60 with increment of 5. It can be shown that S_p increases almost linearly with the increase of data nodes, and the more data nodes are online, the higher is S_p.

Map-Reduce based Topic-STG has better S_p than Map-Reduce based SVD model, this is because that Map-Reduce based Topic-STG needs more execution time since it includes LDA map-reduce function, which would make the ratio of time for resource preparation smaller. On the contrary, Map-Reduce based SVD model needs less execution time than Topic-STG, whereas the time for resource preparation is the same.

We then present the experiment on time-consumption of different steps in Map-Reduce based Topic-STG process with map nodes ranging from 30 to 85 with increment of 5. As shown in Fig.5, the total execution time fist reduces when map nodes increase, and achieve the least when the number of map nodes is equal to 55. And then the total execution time would increase when map nodes increase. This phenomenon reveals that the efficiency of Map-Reduce is affected with three steps and two factors. It is also shown in Fig.5 that, the execution time of reduce function does not change significantly, which means that the reduce function is not the primary step affecting efficiency. Map and shuffle functions would change significantly when map nodes change. When more map nodes are added, the execution time of map function would decrease, and the execution time of shuffle function would increase, this is because when more map nodes are added, the task would be distributed to more map nodes, and reduce the execution time. Whereas this step needs more network transmission, and increases communication cost. We need to balance the parallel and the communication cost, which inspires us to set the number of map node as 55.

4.4 Discussion

4.4.1 The Effect of Topic Sensitiveness

General speaking, the evolution of tweet's topic may affect the user interests. As we observed, a topic can also be categorized into two classifications: time-sensitive (such as "Tianjin Port Explosion") and time-insensitive (such as "Data Mining"). The time-sensitive topic would help to model short-term user interests, whereas the time-insensitive topic captures the long-term user interests. The classifications would reduce the computational cost of SVD model since those time-insensitive hashtags and tweets are no longer calculated with b_{day} in Equation (6).

We addressed the concept of "topic velocity" which is measured by the increasing count of tweets or hashtags during a period of time to refine temporary time-sensitive tweets and hashtags from the time-insensitive ones. We checked the topic velocity for each hour, and set two thresholds $\xi_{ispeed} = 1000$ and $\xi_{dspeed} = -1000$ to find those temporary time-sensitive tweets and hashtag respectively. It can be observed from Fig.6 that higher precision for the final recommendation results will achieve when refine the temporary time-sensitive tweets and hashtag.

4.4.2 The Effect of Topic Number

As we observed, the number of topics would also influence *MAP@N* and diversity metrics of Topic-STG model. As shown in Table 1, if we use the Topic-STG model, the performance of *MAP@N* tends to be stable when $Num(topic) \in [150, 200]$, nodes play an important role to optimize the recommendation results.

As shown in Table 1, the more topics are added, the better performance of the 7 days' diversity would achieve. This is because that more topics would

Table 1: *MAP@N* and 7 days' diversity influenced by latent topics in Topic-STG model.

Metrics	**The number of latent topics**						
	40	50	80	100	120	150	200
MAP@N	0.258	0.319	0.333	0.367	0.385	0.412	0.425
Diversity	0.413	0.436	0.467	0.490	0.517	0.532	0.549

extend user interests, and bring more choices. Another reason is that those new generated topics may be sub topics of the original one. For example, topics “football”, “basketball”, and “NBA” can all be included in the topic “sports”, whereas the new topics would bring more details on user interests. Therefore, it's important to select suitable scope and theme of topics for personalized recommendation.

Indeed, different background, culture, and mutual influence among users, as potential and implicit features, may all affect the user interests, since different approaches capture user interests from different profiles and granularity. The results also reveal that the micro-blogging systems should select suitable length of N for personalized recommendation.

5 CONCLUSIONS

In this paper, we comprehensively considered three aspects of the information: the textual information, the users' behavior, and the time factor to model the user interests, and constructed Topic-STG model and SVD model for tweet recommendation. Also the parallel versions of Topic-STG and SVD models based on Map-Reduce framework were provided to achieve better performance. Experiments on massive Sina Weibo dataset show the effectiveness of the proposed models and algorithms. Still there are several issues should be solved. The first one is that the Topic-STG model brings more computational cost comparing with the original STG model. We should utilize some pruning strategies to improve the performance. The second problem is that retweets and comments present clear attitude to represent user's strong interest or hate. We need to adopt opinion mining approach to identify the subjective information for the SVD model.

ACKNOWLEDGEMENTS

This work is supported by the National Key Research Program of China (No.2016YFB0501900).

REFERENCES

Chawla, N. V., Davis, D. A., 2013. Bringing Big Data to Personalized Healthcare: a Patient-Centered Framework. *Journal of General Internal Medicine.*

Chen, T., et al, 2012. SVDFeature: A Toolkit for Feature-Based Collaborative Filtering. *Journal of Machine Learning Research* 13, 3619-3622.

Xavier, A., Justin, B., 2016. Past, Present, and Future of Recommender Systems: An Industry Perspective. *In Proc. RecSys 2016*, pp.211-214. ACM Press.

Fan, W., Bifet, A., 2012. Mining Big Data: Current Status, and Forecast to the Future. *ACM SIGKDD Explorations Newsletter* 14(2), pp.1-5.

Hannon, J., et al., 2010. Recommending Twitter Users to Follow Using Content and Collaborative Filtering Approaches. *In Proc. RecSys 2010*, pp.199-206. ACM Press.

Hurley, N., Zhang, M., 2011. Novelty and Diversity in Top-N Recommendation-Analysis and Evaluation. *ACM Transactions on Internet Technology* 10(4), 63-72.

Jiang, M., et al., 2014. Scalable Recommendation with Social Contextual Information. *IEEE Transactions on Knowledge and Data Engineering* 26(11), 2789-2802.

Koren, Y., 2010. Collaborative Filtering with Temporal Dynamics. *Communications of the ACM* 53(4), 89-97.

Kumar, A., Niu, F., Christopher, R., 2013. Hazy: Making it Easier to Build and Maintain Big-Data Analytics. *Communications of the ACM* 56(3), 40-49.

Roberto, P.,et al., 2016. The Contextual Turn: from Context-Aware to Context-Driven Recommender Systems. *In Proc. RecSys 2016*, pp.249-252.

Moens, S., Aksehirli, E., Goethals, B., 2014. Frequent Itemset Mining for Big Data. *In Proc. IEEE BigData 2014*, pp.111-118.

Roy, A. et al., 2012. Massive Genomic Data Processing and Deep Analysis. *VLDB Endowment* 5(12), 1906-1909.

Salakhutdinov, R., Mnih, A., 2007. Probabilistic Matrix Factorization. *In Proc. NIPS 2007*, pp.1257-1264.

Xiang, L., et al., 2010. Temporal Recommendation on Graphs via Long- and Short-Term Preference Fusion. *In Proc. SIGKDD 2010*, pp.723-732. ACM Press.

Yigita, M., et al., 2015. Extended Topology Based Recommendation System for Unidirectional Social Networks. *Expert Systems with Applications* 42(7) 3653-3661.

Yin, H. et al., 2015. Dynamic User Modeling in Social Media Systems. *ACM Trans. Inf. Syst*. 33(3), 1-44.

Yu, J., Shen, Y., 2014. Evolutionary Personalized Hashtag Recommendation. *In: Proc. WAIM 2014*, pp.34-37.

Yu, J., et al., 2014. Topic-STG: Extending the Session-Based Temporal Graph Approach for Personalized Tweet Recommendation. *In: Proc. WWW 2014*, pp.413-414. ACM Press.

Yu, J., Zhu, T., 2015. Combining Long-Term and Short-Term user interest for personalized hashtag recommendation. *Frontiers of Computer Science* 9(4), 608-622.

Zaiane, O. R., 2012. Building a Recommender Agent for E-Learning Systems. *In Proc. ICCE 2012*, pp.55-59.