ISSN 2278-3083
Volume 3, No.5, September - October 2014
International Journal of Science and Applied Information Technology
Available Online at http://warse.org/pdfs/2014/ijsait01352014.pdf

A Recommendation System based on Big Data Technology

Minsoo Lee

Dept. Computer Science and Engineering, EwhaWomans University, Korea, mlee@ewha.ac.kr

Abstract: Big data created by Social network contains all sorts of information of the real world such as human relations, time, space and etc. Now it is possible to collect huge amount of data and store it. But the more data we get, the more difficult it is to get the meaningful and requisite information for each person. Thus, it is necessary for us to have a customized recommendation system with a high degree of accuracy which reflects personal characteristics using big data. In this paper,I organized key factors that affect the recommendation by analyzing the characteristics of big data provided by SNS. On the basis of these key factors and relations, I designed a big data model and embodied it for information recommendation systems using MongoDB.The recommendation algorithm can also be parallelized by using the map-reduce approach.

Key words: recommendation, big data, MongoDB, map-reduce

1.INTRODUCTION

Our daily lives leave behind data over the last few years. Social networks really are changing our daily life, and they are enabling technology to bring out the interesting information (relationship of people, time, and space). It's hard to retrieve information that is neededfrom this data. We need a model that incorporates factors related to social networks and can be applied to information recommendation with respect to various social behaviors that can increase the reliability of the recommended information. So I introduce a big data model for recommender systems using social network data.

2. RELATED RESEARCH

Recommendation Systems are a technology that automate the process of suggesting items (such as music, book, films, advertising, club, etc.) that can be interesting for a specific user of the system.

2.1 User Familiarity-based Techniques

In general, users share a variety of information with others who areconnected with a friendship relationship in a social network [1]. Previous work has shown that a user's number of followers is not a good measure of her capacity to propagate content on Twitter, and influence can be defined as the capacity to affect the behavior of others[2]. Also, friendship in social network does not assure the familiarity between the two users. In order to determine the familiarity in social networks,I consider personal informationsuch as relationship, gender, etc.

2.2 The Expert Recommendation Technique

The Expert is one who we can trust to have produced consistent and reliable evaluations (ratings)of items in a given domain. The expert's dataset has different features from regular user's dataset. The expert's dataset of sparse data for item is less than user's datasetand can solve the data scarcity problem. Experts produce consistent and recognized evaluations, so it's expected to reduce noise [3]. Experts have the motivation to participate in the evaluation when there is a target of the new items. This can minimize the problems of coldstart. Because of these advantages, finding experts that can perform recommendation to users in recommendation systems can reduce problems arising fromonly using user's datasets [4].

2. RECOMMENDATION DATA MODEL

I selected the data which includes the necessary information for the recommendation. The selected data is based on a data model. This data model consists of five layers which are User, Club, Content, Item and Category Layer. The data model shows the relationship among the layers as shown in Figure 1. The data are from SNS.

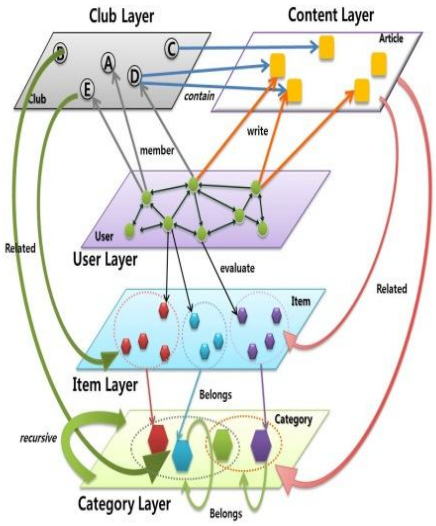

Figure1: Data model for recommendation based on big data

Adatabase schema suitable for the algorithm for recommendation is designed. So a simple algorithm is made and the necessary attributes for recommendation were selected. All evaluation values are not given equal importance. The weights are used to "up-weight" or "down-

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(No. 2012R1A1A2006850).
This work was also supported by the National Research Foundation of Korea grant funded by the government (Ministry of Education, Science and Technology) (NRF-2012M3A9D1054744).

weight" the importance given to the individual ratings. I calculate weights using the similarity between users, user's expertise and evaluation history. I consider users who meet the conditions to be an expert on the subject. The user join a lot of clubs of interest. She/he writes a lot of content for the items of interest and evaluate the item many times. Other users' response could be very positive and other users would usually share the content written by a user.

2.1 MongoDB

I designed a database schema for recommendation using MongoDB [5]. MongoDB is a document-oriented databases. Document-oriented databases are one of the main categories of so-called NoSQL databases. A document-oriented database is designed for storing, retrieving, and managing document-oriented information. Data in MongoDB has a flexible schema. This means collections do not enforce a document structure. Collection in MongoDB are similar to tables, document are similar to rows. So we can insert a single document or an array of documents into MongoDB. As suchMongoDBenables storing of embedded documents. This is the reason why MongoDB was chosen.

2.2 Table Schema

A. User table

User table based on the information about User layer stores basic information of users, user's clubs and user's relationship with friends. This table also stores evaluation history and expert level for recommended items.

The _id field must have a unique value in Mongo DB. You can think of the _id field as the document's primary key. The primary key of User table is user ID. Attributes in the User Table are name, id, password, birthday, gender, nationality, national origin, address, telephone number, e-mail address and marital status,which only havea single value. The club IDs in which the user joined and the interesting field have multiple values in a single cell. Therefore these data are stored in a single array.

The user's alma mater, user's friendship, evaluation of an item and expert information are stored in a table using an embedded document because they have complex hierarchies and multiple values. As I recommend an item, I consider the user's relationship with friends on SNS as important information. The embedded document friendship stores user's friends' ID, the date which they made friends and relationship.

Table 1: User table

_id	attribute 1	attribute 2	attribute 3	attribute 4	attribute 5	attribute 6
user_id	name	id	pw	birthday	gender	nationality

attribute 7	attribute 8	attribute 9	attribute 10	attribute 11	attribute 12	attribute 13
national _origin	address	tel	e-mail	marital _status	club_id	interesting _field
					club_id	
					club_id	interesting _field
					club_id	
					club_id	interesting _field
					:	

embedded document 1				embedded document 2		
attribute 14	attribute 15	attribute 16	attribute 17	attribute 18	attribute 19	attribute 20
alma_mater				friendships		
elementary _school	middle _school	high _school	university	user_id	relationship	date
			university	user_id	relationship	date
			:	user_id	relationship	date
				:	:	:

embedded document 3				embedded document 4		
attribute 21	attribute 22	attribute 23	attribute 24	attribute 25	attribute 26	attribute 27
item_scores				category_experts		
item_id	score	item _expert_level	date	category _id	category _expert_level	date
item_id	score	item _expert_level	date	category _id	category _expert_level	date
:	:	:	:	category _id	category _expert_level	date
				:	:	:

B. Club table

The Club table stores group's prestige, member's visit history, information of member and basic information about group, which are club name, group's birthday and topic(category_id) for group. The primary key of club table is club_id. The table is used to judge user similarities and compute the expert level. This table has category_id to know what the topic for group is and the IDs of contents which are owned by group. The members embedded document stores each member's ID, join date and last visit time. The 'visit ' embedded document has the number of visitors per day. The data included in the'visit' document is used to compute the prestige of the club. I consider the total visitors and the recent visitors, when deciding the prestige of the group. If the number of total visitors and the recent visitors are high would mean that the club members were very active.

Table 2: Club table

_id	attribute 1	attribute 2	attribute 3	attribute 4	attribute 5
club_id	name	date	category_id	club _prestige	members _count

attribute 6	embedded document 1			embedded document 2	
	attribute 7	attribute 8	attribute 9	attribute 10	attribute 11
content_id	members			visit	
content_id	user_id	join_date	last_visit	date	visit_count
:	user_id	join_date	last_visit	date	visit_count
	:	:	:	:	:

C. Content and History table

The Content Layer is divided into the Content table and History table for the sake of convenient reference. I store the basic information ofthe content and reply, writer's information, user's preference(like_user_count) and concern degree(view_count) in the Content table using content_id as the primary key of Contenttable.

The information which is stored in Content table is described as follows. As the written content is in regard to category and not in regard to the given item, the value of item_id is null. If the content is open, the value of security_level is 0. If the content is nondisclosure, the value of security_level is 0. The embedded document 'replies' includes the ID of the user who has written a comment,the comment, and the comment creation date. The

like_user_idsis a single array which stores the ID of the user who like the content.like_user_ids is used to compute the public popularity. The embedded document 'sharing' stored the user's ID or club's ID which take the content. 'sharing_count' which is summary data shows the power of the content. The user (or the club) who wrote popular content has a high level of expertise.

To decide a user's expert level, History table stores the data about the contents written by users and the information of reply using user_id as primary key. The'sharing_count'represents how many people shared the contents and reply written by user.If the 'sharing_count'which is summary data is high would mean thatother user's response was very positive.

Table 3: History table

_id	attribute 1	embedded document 1				embedded document 2		
		attribute 2	attribute 3	attribute 4	attribute 5	attribute 6	attribute 7	attribute 8
user _id	sharing_ count	contents				replies		
		content _id	item_id	count	date	item_id	count	date
		content _id	item_id	count	date	category _id	count	date
		content _id	NULL	count	date	item_id	count	date
		content _id	item_id	count	date	item_id	count	date
		:	:	:	:	:	:	:

D. Item table

The Item Table stores the basic information about item, which are the name of the item, the item creation date and maker, and summary data about evaluation using 'item_id' as the primary key. The evaluation value of the 'item' and the information on user evaluating the item are stored in the embedded document 'item_scores'. The summary data includingscore_expert_count and score_expert_averageare used to compute the evaluation value of the category which the item belongs to.

Table 4: item table

_id	attribute 1	attribute 2	attribute 3	attribute 4	attribute 5	attribute 6	attribute 7
item_id	name	date	maker	score _count	score _average	score _expert _count	score _expert _average

attribute 6	embedded document 1			
	attribute 7	attribute 8	attribute 9	attribute 10
category_id	item_scores			
category_id	user_id	score	item_expert_level	date
category_id	user_id	score	item_expert_level	date
category_id	user_id	score	item_expert_level	date
:	:	:	:	:

E. Category table

The Category Layer is divided into theCategorytable which contains the basic information, and theCategory structuretable which represents the hierarchical relationship of categories. TheCategorytablewhich contains the basic informationconsist ofcategory name, category score and category creation date.

Table 5: Category table

_id	attribute 1	attribute 2	attribute 3
category_id	name	category_score	date

The Category structure table is used to update the evaluation value of the category and to understand the category structure. The category is a recursive structure which contains other categories and the evaluation value of the category is calculated by adding the items' value. Once the evaluation value of one item is updated, the evaluation value of the categories which include the item must be updated as well. To do this effectively I can find the parent categories which are the parent of the item using the 'parent_id' and update the values of the categories. I repeat this work until it is over.

Table 5: Category structure table

_id	attribute 1	attribute 2
category_id	parent_id	child_id
	parent_id	child_id
	parent_id	child_id
	:	child_id
		:

3. RECOMMENDATION FRAMEWORK

In this section, I discuss the recommendation technique which considers the various elements of the social network. An analysis on the elements composing information recommendation in a social network environment is carried out and an approach to integrate these elements for information recommendation is discussed. The recommendation can enhance the credibility and better reflect the personal preference of users.

3.1 Proposed Recommendation Technique

The following formula represents the basic integration concepts of information recommendation element

$$R = \frac{P1\,Rank \times W1 + P2\,Rank \times W2 + \cdots + Pn\,Rank \times Wn}{\sum_{i=0}^{n} w_i} \quad (1)$$

Where *w* is the weight of each person and *PiRank* is the rank value for each person. Therefore, I can calculate *R* by *PiRank* multiplied by the weight which depends on the reliability.

I consider weight in social networks for providing individuals with personalized recommendations.

$$w_n = SNSw_n(p_n) + EVALw_n(p_n) \quad (2)$$

The $w_n$denotes weight in a social network. When the value of weight is high the evaluation is more reliable.$SNSW_n(p_n)$is the weight based on social network and $\mathrm{EVAL}W_n(p_n)$is the weight based on evaluation.

I propose two factors about the weight based on social network. Friends' or experts' preference information is more credible and plausible. For example, if someone wants to try a restaurant which he didn't go to before, he would accept his friend's or expert's recommendation easily due to the trust on them. In general, most normal users trust a power user's opinion, and accept the items recommended by them with ease. Motivated by this example, I identify experts

from users and consider the preference of these users as important.

$$SNSw_n = conf(P_n, u) + \exp(P_n) \quad (3)$$

The $conf(P_n)$ is the confidence which express the relationship between users in SNS, $exp(P_n)$is the expert on the topic.

The confidence is calculated by considering four main factors such as friendship between people, the proportion of co-joining the same groups on the topic, the relationship based on the content and the similarity between the two users. The following equation can integrate the confidences together which are used in the weight computation.

$$conf(p,u) = friendship(p,u) + \frac{n(group(p) \cap group(u))}{n(group(p) \cup group(n))} + \frac{n(content(p) \cap content(u))}{n(content(p) \cup content(n))} \quad (4)$$

In Equation (4), *friendship(p,u)* will use the value of interaction in a social network. I consider three cases where both user *p*and *u* both mutually consider each other as friends, only *p* considers *u* as a friend, and only *u* considers *p* as a friend.

4. RECOMMENDATION PROCESSING

In Equation (4), to calculate the ratio of the content which users participate in, I have to know the number($\cup$C) of contents which user *p* and *u* participated in and the number($\cap$C) of contents which user *p* or *u* participated in. It is possible to calculate the numbers($\cup$C and $\cap$C) using Map and Reduce functions [6].

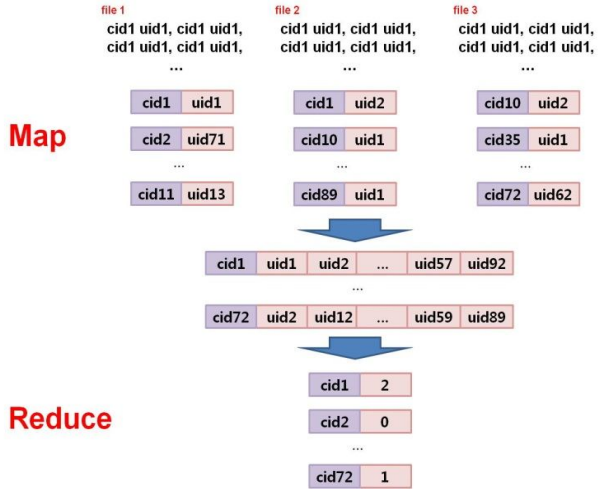

Figure2: Map-Reduce Processing of Recommendation

The overall process is shown in Figure 2. Map functions take the input data, which contain the identifiers of the contents and the identifiers of the user who participate in the contents, use the IDs of content as key and use the IDs of user who participated in the content as value, produces a set of intermediate key/value pairs. It is then submitted to the Reduce function. The Reduce function accepts IDs of the content as an intermediate key and list of the users'ID as a set of values for that key. It merges together these values to form a possibly smaller set of values($\cup$C or $\cap$C). If two users participated in a content then the value is 2. If only one user participated in the content, the value is 1. If no user participated in a content then the value is 0. The Reduce function will show the results of the corresponding value from each key. The value of $n(content(p) \cap content(u))$is the total of the one's and The value of $n(content(p) \cup content(u))$is the total of the two's. The weight of the group is processed in the same way.

5. CONCLUSION AND FUTURE WORK

In this paper, Iproposed various elements related to recommendation of big data provided by SNS and proposed the recommendation framework as well as the processing mechanism. I designed a big data model and implemented it using MongoDBfor information recommendation systems. The recommendation framework provides calculation of weightson user's ratings related to SNS friendships and common interest, and expertise. The processing algorithm is provided as a map-reduce algorithm.

Further work could be done on enhancing the recommendation framework with more detail elements fine-tuned for specific applications.

REFERENCES

1. B. Liu, Z. Yuan.**Incorporating Social Networks and User Opinions for Collaborative Recommendation: Local Trust Network based Method**, CAMRa2010, pp. 53-56,Sep. 2010.
2. A. Cheng, N. Bansal, N.Koudas.**Perckalytics: Analyzing Experts and Interests on Twitter**, In Proc. of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD13), pp.973-976, 2013.
3. S. Song, S. Lee, S. Park, S. Lee. **Determining User Expertise for Improving Recommendation Performance**, ICUIMC'12, Article No. 67, Feb.2012,
4. X. Amatriain, N. Lathia, J. M. Pujol, H. Kwak, and N. Oliver.**The Wisdom of the Few: A Collaborative Filtering Approach Based on Expert Opinions from the Web**, In Proc. of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR09), pp. 532-539, 2009.
5. D. Michael, MongoDB: The Definitive Guide, O'REILLY, September 2010
6. J. Dean and S.Ghemawat, **MapReduce: Simplified Data Processing on Large Clusters**, Communications of the ACM, Vol. 51 Issue 1, pp. 107-113, January 2008