https://dx.doi.org/10.22161/ijaers/3.10.36 ISSN: 2349-6495(P) | 2456-1908(O)

Social Data Schema and Recommendation Method for Restaurant Information

Minsoo Lee

Department of Computer Science and Engineering, Ewha Womans University, Seoul, Korea

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 restaurant recommendation system with a high degree of accuracy which reflects personal characteristics using big data. In this paper, we organized key factors that affect the restaurant recommendation by analyzing the characteristics of big data provided by SNS. On the basis of these key factors and relations, we designed a big data model and embodied it for information recommendation systems using MongoDB. The restaurant recommendation technique uses a collaborative filtering approach by using the factors of the social network and the user's evaluation history.*

Keywords— Social data, Recommendation, Database, Schema, Restaurant Information

I. INTRODUCTION

Our daily lives leave behind data over the last few years. Social networks really are changing the way we live our lives, and they are enabling technology to bring out the interesting information (relationship of people, time, space). It's hard to retrieve information that we need in a large amount of data. We need a model which 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 we introduce a big data model for recommender systems using social network data.

II. RELATED WORK

Recommendation Systems are a technology that automates 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, user shares a variety of information with others who you make a friendship in social network. 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 network, we consider personal information (such as relationship, gender etc.), and activities.

2.2 The Expert Recommendation Technique

The Expert is 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 dataset and can solve the data scarcity problem. Expert produced consistent and recognition evaluations, so it's expected to reduce noise. Experts have the motivation to participate in the evaluation when there is the target of the new items, so it minimizes cold-start. Because of these advantages, finding experts from users to recommendation system can reduce problems that use user's dataset.

III. RECOMMENDATION ALGORITHM FOR RESTAURANT

We select the data which include the necessary information for the restaurant recommendation. The selected data based on this data model. This data model consists of five layers which are User, Club layer, Content, Item and Category Layer.

This data model describes the link among the layers. The data are from SNS.

https://dx.doi.org/10.22161/ijaers/3.10.36 **ISSN: 2349-6495(P) | 2456-1908(O)**

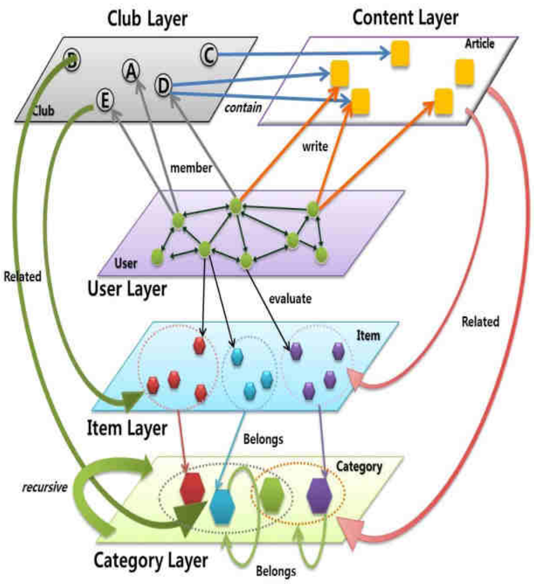

Fig. 1: Data model for Recommendation based on big data

We want to design database schema suitable for the algorithm to restaurant recommendation. So We made the draft of the algorithm and selected necessary attribute for recommendation. We do not give equal importance to all evaluation values. The weights are used to "up–weight" or "down–weight " the importance given to the individual rating. We calculate weights using the similarity between users, user's expertise and evaluation history. We consider users who meet the conditions to be an expert on the subject. The user joins a lot of clubs for restaurant. She or he writes a lot of content and its replies for restaurant and evaluate the item for restaurant many times. Other users response was very positive and other users usually share the content written by user. Reference is used to search what we want.

3.1 MongoDB

We designed database schema for recommendation using MongoDB. MongoDB is 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 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 you see, Mongo lets you store embedded documents. This is the reason why we chose MongoDB.

IV. TABLE SCHEMA

4.1 User Table

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

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. A single attribute in the User Table have user data -name, id, password, birthday, gender, nationality, national origin, address, telephone number, e-mail address and marital status, which has only a single value. The ID of clubs which user joined and interesting fields have multi value in a single cell. Therefore, these data are stored in single array.

User's alma mater, user's friendship, evaluation of a restaurant and evaluation-feeding are stored in table using embedded document because they have complex hierarchies and multi value. As we recommend a restaurant or food, we consider user's relationship with friends on SNS as important information. The embedded document friendship stores user's friend's 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
				:	:	:

4.2 Club Table

'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

https://dx.doi.org/10.22161/ijaers/3.10.36 ISSN: 2349-6495(P) | 2456-1908(O)

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 have 'category_id' to know what is the topic for group 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 'visit' document is used to compute club's prestige. We consider the total visitors and the recent visitors, when deciding group's prestige. The total visitors and the recent visitors are high means the club members acted very well.

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
	:	:	:	:	:

4.3 Content and History Table

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

The information which are stored in 'Content' table is described as follows. Of the written content is in regard to category and not in regard to given item, the vaue 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 was written a comment and comment and the comment creation date. 'like_user_ids' is single array which store the ID of the user who like the content. like_user_count 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 content. The user (or the club) who wrote the popular content is a high level of expertise.

Table. 3: Content table

_id	attribute 1	attribute 2	attribute 3	attribute 4	attribute 5
content_id	user_id	title	content	category_id	item_id

attribute 6	attribute 7	attribute 8	attribute 9	attribute 10	attribute 11
date	view_count	like_user _count	club_id	sharing_count	security_level

attribute 12	embedded document 1			embedded document 2		
	attribute 13	attribute 14	attribute 15	attribute 16	attribute 17	attribute 18
like_user _id	replies			sharing		
like_user _id	user_id	comment	date	user_id	club_id	date
like_user _id	user_id	comment	date	user_id	NULL	date
like_user _id	user_id	comment	date	user_id	club_id	date
:	:	:	:	:	:	:

To decide a user's expert level, 'History' table is stored the data about the contents written by users and the information of reply using user_id as primary key. 'Sharing_count' represents how many people shared the contents and reply written by a user. 'Sharing_count' which is summary data is high means the other user response was very positive.

Table. 4: 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
		:	:	:	:	:	:	:

4.4 Item Table

'Item' Table store 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, which are score_expert_count and score_expert_average, are use to compute the evaluation value of the category which the item belongs to.

Table. 5: 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
:	:	:	:	:

4.5 Category Table

The 'Category' Layer is divided the 'Category' table which contains the basic information and the 'Category structure' table which represents for the sake of convenient reference. The 'Category' table, which

https://dx.doi.org/10.22161/ijaers/3.10.36 ISSN: 2349-6495(P) | 2456-1908(O)

contains the basic information, stores category name, category score and category creation date.

Table. 6: 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 recursive structure which contains the others 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. To do this effectively work we can find the parent categories which are the parent of the item using the 'parent_id' and update the values of the categories. We repeat this work until it is over.

Table. 7: Category structure table

<table>
<tr><th>_id</th><th>attribute 1</th><th>attribute 2</th></tr>
<tr><td rowspan="5">category_id</td><td>parent_id</td><td>child_id</td></tr>
<tr><td>parent_id</td><td>child_id</td></tr>
<tr><td>parent_id</td><td>child_id</td></tr>
<tr><td rowspan="2">:</td><td>child_id</td></tr>
<tr><td>:</td></tr>
</table>

V. EVALUATION OF DATA MODEL

In this section, we discuss restaurant recommendation technique which considers the various elements of the social network. This method will be applied to our model for the purpose of verifying the usefulness of our model. 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.

5.1 Proposed recommendation technique

The following formula represents the basic integration concepts of information recommendation elements.

$$R = \frac{P_1 Rank \times w_1 + P_2 Rank \times w_2 + \dots + P_n Rank \times w_n}{\sum_{i=1}^{n} w_i} \quad (1)$$

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

We consider weight in social network for providing individuals with personalized recommendation on restaurant or food specifically.

$$w_n = SNSw_n(p_n) + EVALw_n(p_n)$$

w_n is the 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 $EVALw_n(P_n)$ is the weight based on evaluation.

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

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

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

The confidence is calculated by considering four main factors as shown in the above data model. They are the friendship between people, the proportion of co-joining the same groups on the topic about food, the relationship based on the content and the similarity between the two users. We obtain the user similarity through comparing a user Profile(gender, age, address, etc.). The following equation can integrate the confidences together which are used in the weight computation.

$$conf(p,u) = friendship(p,u) + \frac{n(group(p) \cap group(u))}{n(group(p) \cup group(n))} + \frac{n(content(p) \cap content(u))}{n(content(p) \cup content(n))} + similarity(p,u) \quad (2)$$

Equation (2) represents the value of the user's friendship intimacy, friendship(p,u) is the friend level of users, group(p) is the group where the user p is a member, and content(p) is the content that user p commented on or discussed.

$$friendship(p,u) = Max(L1(p,u) + L1(u,p), L2(p,*) + L2(*,u) + L2(*,p) + L2(u,*)) \quad (3)$$

In Equation (3), friendship(p,u) will use the value of interaction in social network. We consider three cases that user p, I are friends, I friend user p and user p friend me. When i friends user p, his recommendation has impact on me, so the weight is higher than the case that the other friend me. As you can see in the formula, depending on the level of intimacy, the value of the connection are different; if there is no connection in level 1 then the value is 0, and will select the value of level 2. If it is not level 1 or level 2(there are no relationship), the value of two levels are all 0, so that the value of friendship(p,u) is 0 too. In this study, the weight for the connection of level 1 is 1, and the weight for the connection of level 2 is 1/4.

We assume there are some more reliable and important users for recommendation process, who have deep and broad knowledge of specific domains. $\exp(p_n)$ means expert who have deep and broad knowledge of food.

https://dx.doi.org/10.22161/ijaers/3.10.36 ISSN: 2349-6495(P) | 2456-1908(O)

VI. CONCLUSION

In this paper, we found the various elements related to restaurant recommendation of big data provided by SNS and proposed the restaurant recommendation technique based on the elements. The various social are used to up–weight or down–weight the importance given to the individual rating while performing collaborative filtering, thereby improving the accuracy of the predictions. And we designed a big data model and embodied it for information recommendation systems using MongoDB.

REFERENCES

[1] Bin Liu, Zheng Yuan, "Incorporating Social Networks and User Opinions for Collaborative Recommendation: Local Trust Network based Method", CAMRa2010, pp. 53-56, Sep. 2010.

[2] Sang-il Song, Sangkeun Lee, Sungchan Park, Sang-goo Lee, "Determining User Expertise for Improving Recommendation Performance", ICUIMC'12, No. 67, Feb. 2012.

[3] 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 (SIGIR ’09), 2009.

[4] Alex Cheng, Nilesh Bansal, Nick Koudas, “Perckalytics: Analyzing Experts and Interests on Twitter”, In proc. of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD ‘13), pp.973-976

[5] Xiaoyue Han, Lianhua Tian, Minjoo Yoon, Minsoo Lee, "A Big Data Model supporting Information Recommendation in Social Networks", The Second International Conference on Cloud and Green Computing(CGC2012), pp. 810-813, Xiangtan, Hunan, China, November 2012

[6] Dirolf Michael, MongoDB: the Definitive Guide, O'REILLY, September 2010