Generating ordered list of Recommended Items: a Hybrid Recommender System of Microblog*

Yingzhen Li, Ye Zhang,
School of Mathematics & Computational Science, Sun Yat-Sen University

BACKGROUND
Observing the rise of twitter services’ popularity in 2007, Tencent[www.qq.com], one of China’s leading Internet service portal, launched microblog(board)China’s Twitter in 2010. Based on the large user group of its instant messaging service QQ(711.7 million)[1], Tencent Microblog has attracted large amount of registered users (425 million and 67 million daily active[2]) and became one of the dominant microblog platforms. Tencent invites celebrities and organizations to register and interact with users directly. Users can enjoy fun of Microblog directly on the website of Tencent Microblog or via the third-party port and related platforms. The service is embedded in Tencent’s other leading platforms like QQ signature, Qzone(blog platform), Qun/SNS service) and Weixin/mobile messenger.

PROBLEM
While Tencent has the biggest microblog user groups, Sina Microblog took a commanding lead with 56.5% of China’s microblog market[2]. Tencent Microblog(t.qq.com) has attracted large amount of registered users and generation of ordered item list.

SOLUTION: HYBRID RECOMMENDER SYSTEM
Recommender systems can be categorized into content-based algorithm[6], collaborative filtering[7], and influential ranking algorithm[8]. However each single algorithm has its unavoidable disabilities. Hence we design a hybrid approach considering user preference variance and similar interests among linked users, including keyword analysis, user taxonomy and generation of ordered item list.

The grade of recommending item i(belong to category h) to user u(specified to its user class) is computed by:

\[
\text{grade}(u, i) = \frac{2 \cdot \text{fond}(u, h)}{1 + \text{sim}(u, i)}
\]

where \(\alpha\) is trained in training process and identical to u.

Keyword Analysis
Noticing the existence of synonyms, we group the keywords into classes to extract user’s (and item’s) interests. But mining keyword classes directly in the huge user-keywordset is unrealistic, so we parallel the candidate generation by adopting and revising FDM[9]. Evidently the choice of (local/global) support and confidence affect the precision and complexity tremendously. We sampled 1000 users’ keywords and found out that these users have their keyword weights average in 0.14, so we assign supp_local = supp_global = 0.2 and conf_local = conf_global = 0.7. Also we notice the ambiguity of keywords, ‘apple’ for instance, hence we insert these ambiguous keywords into different classes simultaneously.

User Taxonomy
According to the number of tweets(due to the lost of login data) and interactions with others we group the users into 3 excluded groups - active, inactive and fake - to apply different types of strategies. In fact, lot of Tencent microblog users actually seldom login, and the messages generate from the third-party port associated platforms are synchronized to their microblog, generating the fake illusion of their activeness. In addition we also classify the spammers as fake users. With statistics we conclude that only 33.2% of the users have written more than 100 tweets. We choose min_tweet = 100, min_interaction = 20, and separate the users into 3 groups.

Item Ranking(Computing hot and HOTxs)
An item is a specific user, which can be a famous person, an organization, or a group. Items are organized in different categories by Tencent according to their professional domains, which forms a hierarchy. Obviously the number of an item’s fellowusers reflect its popularity directly. We adopt that indicator and rank the items in categories.

Computing Similarity(u,i) and fond(u, h)
Recommending items with high similarity in preference is considerably effective to increase the percentage of acceptance. We extract the interests of users from their keyword classes. Noticing that few users actively write tweets thus not enough keywords, we design indirect collaborative filter to mine the potential interests of inactive users from their followees’s [fan] and apply KNN(u, i) to represent the familiarity between two users u and i to adjust the weights of potential interests. Then we define KKH mapping which maps the interests to the hierarchy of items’ professional domains, compute fondt(u, h) which indicate user’s preference of the category and obtain u’s candidate items to compute their similarity simt(u, i).

EXPERIMENT & RESULTS
We sampled 5938 users for experiment. All the data is encrypted by Tencent. After training we noticed the variance of \(\alpha\) among different user classes.

EXPERIMENT ALGORITHM

\[
\text{AP}(u, i) = \sum_{j \in \text{top}3} \frac{\text{HOT}(i)}{\text{HOT}(j)}
\]

where \(\text{top}3\) is the precision of the \(i\) recommended item and \(\text{HOT}(i)\) is the change in the recall form \(1\) to \(3\) and present the experiment’s result and example prediction:

\[
\text{AP}(u, i) = \sum_{j \in \text{top}3} \frac{\text{HOT}(i)}{\text{HOT}(j)}
\]

The result showed the high performance of our Hybrid Recommender System after training. To save time we can divide users into smaller classes and train the machine respect to each class.

REFERENCE

ABOUT US
Yingzhen Li
Undergraduate student(Junior Year),
School of Mathematics & Computational Science,
Sun Yat-Sen University, Guangzhou, China
E-mail: Syzhang2@mail2.sysu.edu.cn

Ye Zhang
Undergraduate student(Junior Year),
School of Mathematics & Computational Science,
Sun Yat-Sen University, Guangzhou, China
E-mail: zhangyey5@mail.sysu.edu.cn

*solution of Track 1 task, KDD Cup 2012