A House Recommendation System

Mingruo Shi

Beijing Wuzi University, Beijing 101149, China.

Abstract

A recommendation system is increasing important in more and more domains. House search is even more important for our daily life. Our proposed filtering solution is based on content match to handle large number of users and houses for large scale house serving website.

Keywords

Recommendation, Content based filtering, Ranking, Hadoop.

1. Introduction

Recommender systems have become extremely common in recent years, and are utilized in a variety of areas: some popular applications include movies, music, news, books, research articles, search queries, expert system, collaborators, social tags, financial services, Twitter pages, and products in general [1,2,3,4].

Recommender systems typically produce a list of recommendations in two possible ways – through collaborative and content-based filtering [5,11,12]. Collaborative filtering approaches building a model from a user's past behavior (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in [6]. Content-based filtering approaches utilize a series of discrete characteristics of an item to recommend additional items with similar properties [7]. These approaches might be combined to produce better recommendation results.

A recommendation system usually including two major parts. One is to create a recommendation item candidate list based on user's visit/click history. The other part is to create top n recommendation results by ranking the recommended candidates. There are some papers discussing some scalable recommendation to users including Google news recommendation [8,10]. To our best knowledge, no literature gives the detailed introduction how a large-scale recommendation system is designed and implemented scaled both to number of users and items on house recommendation.

In this paper, we proposed a content-based filtering recommendation algorithm scalable to both users and items which can be applied to house recommendation esp. on house websites.

The paper is organized as follows: in section 2, we describe in detail our algorithm to support large number of users and houses. In section 3 introduces our experiments. We summarize our ongoing work in section 4.

2. Algorithm

2.1 Basic Idea

Given a list of N users U, a list of M houses H and the click history of a given periods e.g. 60 days, our algorithm will output a list of house recommendations for each user he/she might prefer where U are users {U_1,U_2, …, U_n } with user profile and H are houses {H_1,H_2, …, H_m}.

Our solution is to develop a content-based collaborative filtering recommendation system. We firstly extract the side information for user to build user profile and house information for match. Then use the click history and side information including user profile and house information in both training and rank phases of the algorithm.

2.2 Build User Profile

The basic user profile information includes age, location, gender etc. which could get for registered users. Besides the mentioned basic user information, we could leverage the information inferred from user's daily activities i.e. searched keywords and click activities.

We build user profile by leveraging house detailed information. The most important information will be extracted from house are address, zip code, listing price, sold price, city, built year, square feet, number of bedrooms. We can also leverage more house entity information such lot size, estimated price. Then we build (feature, feature value) key value pairs by aggregating a fixed period click history for example 60 days.

House features for user profiles could be built by different ways for a specific recommendation target. Propose one way including features: country, city, zip code, zip code | listing price bucket, built year, zip code | sold price, square feet. The feature value is percentage by aggregating last 60-day click history.

2.3 Extract House Features

Our target is to recommend houses to users. we therefore only consider active listing houses. And extract the same features as user profile. For example, we could extract features such as country, city, zip code, zip code | listing price, built year and square feet.

2.4 Rank Recommendation Result

We could calculate match score between users and houses with equal country and city. For matched user and house pair, we could get match score by the following way:

$$s_{u,h} = \sum_{i=1}^{n} c_i f_i \ . \tag{1}$$

$s_{u,h}$ is the match score. n is the number of matched features for user u and house h. f_i is the feature value in user's profile for a give user u, we get top k houses ranked by match score satisfy $s_{u,h_1} \geq s_{u,h_2} \geq \cdots \geq s_{u,h_k}$ where k is a predefined number.

2.5 Decide Coefficients

There are a few ways to decide the coefficients c_i during user and house matching process. We use logistic regression machine learning method on user click history to figure out the optimized coefficients c_i.

Finally, there are multiple ways to recommend house list to users in some ways. For example, a user could get recommended by web page or by email.

3. Experiments

3.1 Evaluation Methodology

We take 60-day click log in our test house website and split the data into the training set and the test set. The test set is the latest day click log. The training set are used to get recommendation for every user and compare the recommendation list with the list of houses that users visited in the test set. When we measure, we remove users/entities in the test file that is NOT in the training file.

Running time: In our log, the number of distinct users is up to ~2 million and houses is ~1 million. We filtered out the traffic visited by user with age less than 18 years old. The training can be done in 1 hours, using about 5 computers in a Hadoop cluster. So, in the following, we mainly discuss about the prediction accuracy.

Prediction accuracy: NDCG (Normalized Discounted Cumulative Gain) are used for this purpose in this paper. We define NDCG @ k as $NDCG_k = \frac{DCG_k}{DCG_k^*}$ where $DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i}-1}{\log_2(i+1)}$ and DCG_k^* is the ideal DCG_k. We assign rel_i to equal 1 if there was a click on item ranked in position i. Here, k is the recommendation length, i.e, select top k houses to users. Basically, NDCG assign higher score to clicked items that in the top of the recommendation list than the clicked items in the bottom of the list.

Evaluation Result Besides our algorithm, we also developed a Collaborative Filtering algorithm which doesn't use any house entity information at all. However, our algorithm takes important signals from users' click history. The evaluation result is shown in Table 1. As Table 1 suggest, our algorithm clearly outperforms collaborative filtering algorithm.

Table 1. NDCG

Algorithm	NDCG @1	NDCG @2	NDCG @3
CF	3.5	4.2	4.5
Our Algo	5.7	6.3	6.8

4. Future Work

Firstly, personalized ranking methods from Information Retrieval could be leveraged. Researchers came up with various ranking strategies to present relevant information to users. There is a line of research that incorporate user’s interests into the ranking, called personalized ranking [11]. The same idea can be applied in ranking the recommendation result. Finally, probabilistic recommendation algorithm might be helpful. pLSI (probabilistic latent semantic indexing)-based algorithm may outperform our hashing-based algorithm [12].

Acknowledgments

This paper is supported by Technological innovation service capacity improvement, Research base construction, Collaborative Innovation Center for Intelligent Logistics System (No. 0351800901).

References

[1] H. Chen, A. G. Ororbia II, C. L. Giles ExpertSeer: a Keyphrase Based Expert Recommender for Digital Libraries, in arXiv preprint 2015.

[2] H. Chen, L. Gou, X. Zhang, C. Giles Collabseer: a search engine for collaboration discovery, in ACM/ IEEE Joint Conference on Digital Libraries (JCDL) 2011.

[3] Alexander Felfernig, Klaus Isak, Kalman Szabo, Peter Zachar, The VITA Financial Services Sales Support Environment, in AAAI/IAAI 2007, pp. 1692-1699, Vancouver, Canada, 2007.

[4] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Bosagh Zadeh WTF: The who-to-follow system at Twitter, Proceedings of the 22nd international conference on World Wide Web.

[5] Hosein Jafarkarimi; A.T.H. Sim and R. Saadatdoost A Naïve Recommendation Model for Large Databases, International Journal of Information and Education Technology, June 2012.

[6] Prem Melville and Vikas Sindhwani, Recommender Systems, Encyclopedia of Machine Learning, 2010.

[7] R. J. Mooney & L. Roy (1999). Content-based book recommendation using learning for text categorization. In Workshop Recom. Sys.: Algo. and Evaluation.

[8] Das, Abhinandan S., et al. "Google news personalization: scalable online collaborative filtering." WWW’07.

[9] Indyk, Piotr. "A small approximately min-wise independent family of hash functions." Journal of Algorithms 38.1 (2001): 84-90.

[10] Jure Leskovec, Anand Rajaraman, Jeff Ullman, “Mining of Massive Datasets”, Chapter 3.

[11] Bouadjenek, Mohamed Reda, et al. "Evaluation of personalized social ranking functions of information retrieval." ICWE’13.

[12] Liu, Jiahui, et al. "Personalized news recommendation based on click behavior." ICIUI’10.