ISSN: 2472-3703 DOI: 10.6911/WSRJ.202101_7(1).0005

A Collaborative Filtering Schema on Second-hand Commodity Online Trading

Mingruo Shi

Beijing Wuzi University, Beijing, China

Abstract

Second-hand commodity trading is getting even more popular due to the online shopping convenience. It's, however, is big challenge to pay high delivery cost even if a buyer just buys a very cheap item so that in most cases the potential deal fails to make eventually. A collaborative filtering schema is proposed to recommend relevant commodities available nearby the locations of seller to buyer so that the deal could be. Essentially, our proposal is to make shipment cost acceptable to buyer by combining multiple items together from multiple sellers. More specifically, low rank matrix factorization is leveraged to recommend top n similar commodities for each user selected commodity with in same geo-location.

Keywords

Second-hand commodity; Collaborative filtering; Delivery cost; Low rank matrix factorization.

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, social tags, search engine and online shopping in general. There are also recommender systems for experts [1], collaborators [2], jokes, restaurants, garments, financial services [3], life insurance, online dating, and Twitter pages [4].

Recommender systems typically produce a list of recommendations in one of two ways – through collaborative and content-based filtering (or the personality-based) approach [5,8,9]. 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 [6]. Content-based filtering approaches utilize a series of discrete characteristics of an item to recommend additional items with similar properties [7]. These approaches are often combined (called Hybrid Recommender Systems).

A recommender system usually includes two major parts. One is recommendation which will create a recommendation item candidate list based on user's activity history. The other part is to create top n recommendation results by ranking the recommended candidates. To our best knowledge, no literature gives the detailed introduction how to combine commodities from multiple sellers into a single shipment to buyer to save the shipping cost on second-hand shopping. In this case the user's activity history majorly is of shopping history.

In this paper, we proposed a collaborative filtering recommender for second-hand commodity recommendation to combine commodities from multiple sellers to single shipment to achieve higher deal made rate by saving overall shipping cost. The paper organizes as follows. Section 2 introduces similar commodity recommendation algorithm based on low rank matrix

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202101_7(1).0005

factorization. Section 3 introduces the experiment results. We summarize our possible ongoing work in section4.

2. ALGORITHM

Problem statement for recommendation is as follows. There are n_u users and n_m commodity categories. The input data are user click history with the fixed period (e.g. 3 months) from some online shopping website. We could figure out an estimated rating $y^{(i,j)}$ for user j to commodity category i by mining the giving historical log. For example, we could set rating from 0 to 5 according to if user j clicked job i and how long-time user j focus on job i. If there is a rating from input data, $r(i,j) = 1$. Otherwise, $r(i,j) = 0$. The output is the learned rating $\bar{y}^{(i,j)}$ for user j to commodity category i which is used to recommend jobs for user i.

Our solution is to develop a collaborative filtering recommender system. The target to learn features for each commodity category i and each user j. Given the features to be learned for commodity category 1, ..., n_m are $x^{(1)}, \dots, x^{(n_m)}$ and the features to be learned for user 1, ..., n_u are $\theta^{(1)}, \dots, \theta^{(n_u)}$ respectively.

One possible way is given commodity category's features to estimate users' features. Then given users' features to learn commodity category's features. And then iteratively to refine users' features and commodity category's features separately. The details are as bellow. By this way, the learning rate is slow.

Given $x^{(1)}, \dots, x^{(n_m)}$, estimate $\theta^{(1)}, \dots, \theta^{(n_u)}$:

$$\min_{\theta^{(1)},\dots,\theta^{(n_u)}} \frac{1}{2}\sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}\left(\left(\theta^{j}\right)^{T}x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2$$

Given $\theta^{(1)}, \dots, \theta^{(n_u)} x^{(1)}, \dots, x^{(n_m)}$, estimate $x^{(1)}, \dots, x^{(n_m)}$:

$$\min_{x^{(1)},\dots,x^{(n_m)}} \frac{1}{2}\sum_{i=1}^{n_m}\sum_{j:r(i,j)=1}\left(\left(\theta^{j}\right)^{T}x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

This paper is to minimize $x^{(1)}, \dots, x^{(n_m)}$ and $\theta^{(1)}, \dots, \theta^{(n_u)}$ simultaneously:

$$J_{x^{(1)},\dots,x^{(n_m)},\theta^{(1)},\dots,\theta^{(n_u)}} = \min_{\substack{\theta^{(1)},\dots,\theta^{(n_u)} \\ x^{(1)},\dots,x^{(n_m)}}} \frac{1}{2}\sum_{(i,j):r(i,j)=1}\left(\left(\theta^{j}\right)^{T}x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

Our collaborative filtering algorithm:

1. Initialize $x^{(1)}, \dots, x^{(n_m)}$ and $\theta^{(1)}, \dots, \theta^{(n_u)}$ to small random values.

2. Minimize $J_{x^{(1)},\dots,x^{(n_m)},\theta^{(1)},\dots,\theta^{(n_u)}}$ using gradient descent (or an advanced optimization algorithm). E.g. for every $j = 1, \dots, n_u, i = 1, \dots, n_m$:

$$x_k^{(i)} := x_k^{(i)} - \alpha\left(\sum_{j:r(i,j)=1}\left(\left(\theta^{(j)}\right)^{T}\right)x^{(i)} - y^{(i,j)}\right)\theta_k^{(j)} + \lambda x_k^{(i)}\Bigg)$$

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202101_7(1).0005

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha\left(\sum_{i:r(i,j)=1}\left(\left(\theta^{(j)}\right)^T x_k^{(i)} - y^{(i,j)}\right)\theta_k^{(j)} + \lambda\theta_k^{(j)}\right)$$

3. For a user with parameters θ and a job with (learned) features x, predict a star rating of $\theta^T x$.

$$\begin{bmatrix} \left(\theta^{(1)}\right)^T\left(x^{(1)}\right) & \cdots & \left(\theta^{(n_u)}\right)^T\left(x^{(1)}\right) \\ \vdots & \ddots & \vdots \\ \left(\theta^{(1)}\right)^T\left(x^{(n_m)}\right) & \cdots & \left(\theta^{(n_u)}\right)^T\left(x^{(n_m)}\right) \end{bmatrix}$$

Mean normalization is introduced here to recommend jobs for users never clicked any job:

$$\mu_i = \underset{j:r(i,j)=1}{\text{avg}} \; y^{(i,j)}$$

$$\bar{Y} = Y - \mu$$

$$\begin{bmatrix} \left(\overline{\theta}^{(1)}\right)^T\left(\overline{x}^{(1)}\right) & \cdots & \left(\overline{\theta}^{(n_u)}\right)^T\left(\overline{x}^{(1)}\right) \\ \vdots & \ddots & \vdots \\ \left(\overline{\theta}^{(1)}\right)^T\left(\overline{x}^{(n_m)}\right) & \cdots & \left(\overline{\theta}^{(n_u)}\right)^T\left(\overline{x}^{(n_m)}\right) \end{bmatrix}$$

For user j on commodity category i, predict a star rating as below:

$$\left(\overline{\Theta}^{(j)}\right)^T\left(\overline{X}^{(i)}\right) + \mu_i$$

Each commodity on online shopping website is related to one or more commodity categories. When user click one second-hand commodity and then add it to shopping cart, we will try to recommend multiple commodities available to sell which are close enough to the seller of the clicked commodity.

How do we recommend such relevant nearby commodities? We could get the seller's geo location by seller's shipping address. Then we could pick all commodities available around the seller within 10 miles. Then find top n commodities ranking j by $\|x^i - x^j\|$ in ascending order which $\|x^i - x^j\|$ is the distance between commodity i and commodity j.

In such way, the buyer has more opportunity to pick more commodities he or she need so that more likely to make the deal. Once deal is made, single shipment is much more possible especially for cheap and small-size commodities.

3. EXPERIMENTS AND EVALUATION

nDCG [10] is a measure of ranking quality. So, we use it to measure the performance of our proposed collaborative filtering algorithm. We firstly take a brief introduction to nDCG. Then illustrate the experiment result which indicates that our algorithm is promising in terms of nDCG.

3.1. Evaluation Method

Three-month job click history log in our test second-hand online shopping website is taken for evaluation. The data are divided into two sets i.e. training set and test set. The test set is the latest day log (one-day log). We use the training set to get recommendation for every user and compare the recommendation list with commodity list that users bought in the test set. When measuring, we remove users/commidities in the test set that is NOT in the training set. Note that the offline evaluation does not truly reflect the performance of the online system (assume

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202101_7(1).0005

the performance would not be worse than the real case) since we haven't made the recommendation to end users yet. But it helps us determine if the ideas can potentially improve the performance of the online system.

Training: our data sets covers around 3,000 users and about 10,000 commodities. The training can be done in 4 hours, using about 3 computers in a Hadoop cluster. So, we mainly discuss metric nDCG later in this paper.

nDCG (Normalized Discounted Cumulative Gain). nDCG @ k is defined 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 jobs to present to users. Basically, nDCG assign higher score to visited items that in the top of the recommendation list than the visited items in the bottom of the list.

3.2. Evaluation Result

Besides our algorithm, we also developed a co-click-based algorithm to generate non-personized top n commodity list. However, our algorithm takes important signals from users' bought history. Note that we only take top 12 commodities are evaluated. The evaluation results are shown in Figure 1. As Figure 1 suggests, our collaborative filtering algorithm outperforms.

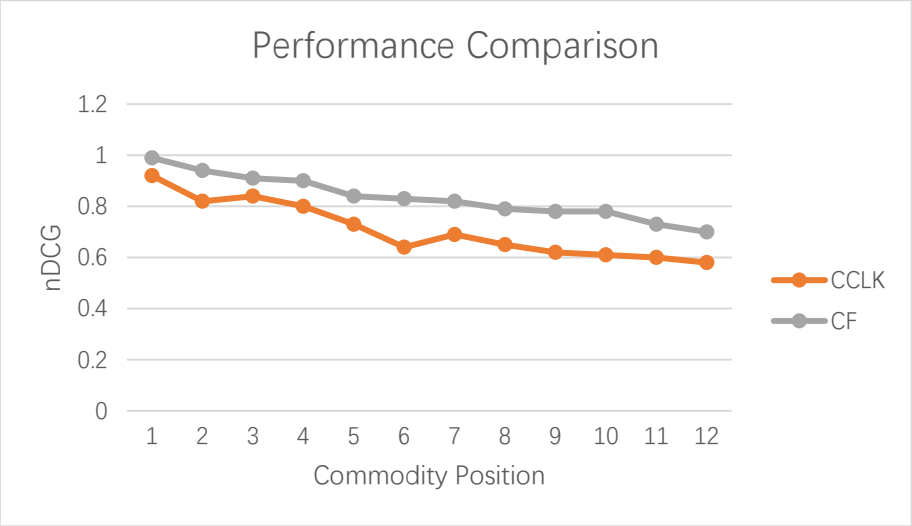

Figure 1. Our collaborative filtering (CF) algo vs co-click-based (CCLK) algo

4. CONCLUSION AND FUTURE WORK

By recommending relevant commodities close to seller based on low rank matrix factorization, buyer get more opportunity to buy commodities from seller in single shipment. So overall of shipping cost is saved. The higher deal make rate is achieved by experiment evaluation. In future more methods could be tried to get even better relevant recommending commodity. Firstly, we could recommend batch of related commodities through emails or in-site messages for potential buyers in case the buyer had not made a deal before. Secondly, recommend single commodity recommendation for seller close enough to the potential buyer with much less shipping cost.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202101_7(1).0005

ACKNOWLEDGMENTS

This paper was 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] Bouadjenek, Mohamed Reda, et al. "Evaluation of personalized social ranking functions of information retrieval." ICWE'13.

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

[10] Discounted cumulative gain. https://en.wikipedia.org/wiki/Discounted_cumulative_gain#Normalized_DCG.