The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

Learning Sequential Behavior for Next-Item Prediction

Na Lu Yukio Ohsawa Teruaki Hayashi

Department of System Innovation, School of Engineering, The University of Tokyo

A more precise recommendation plays an essential role in e-commerce. Representation learning has attracted many attentions in recommendation field for describing local item relationships. In this paper, we utilize the item embedding method to learn item representations and user representations. Our methods compute cosine similarity of user vector and recommended item vectors to achieve the goal of personalized ranking. Experiment on real-world dataset shows that our model outperforms baseline model especially when the number of the recommended item is relatively small.

1. INTRODUCTION

The sharp growth of e-commerce and the using mobile electronic device require a more precise prediction of next item that users would probably like to purchase. Data mining of users' behaviors aims at finding useful patterns from a large database. In this task, understanding users' history and features are one of the most critical parts.

To deal with this task, some models were developed based on last transaction information, which is mostly involving Markov chains[Chen 12]. This method mainly makes use of users' sequential transaction data to predict what will be the next item considering the last transaction event. The advantages of this method are that it could consider the time sequence and recommend a proper item for the next movement. Other general recommendation models would consider users' past purchase behavior as a whole to generate their overall taste (or features)[Rendle 10] . This method could generally grasp a user's interests. The most widely used method of general recommendation models is called collaborative filtering. The advantages of this method are that it could get users' interesting points. Thus, the recommendation could generate from users' whole behavior. However, this method discards subsequent information that may lack preciseness in next-item prediction.

Here a good recommendation model could consider not only the sequential information but also users' overall taste. A hierarchical representation model was proposed to combine both sequential information and user history transaction information [Wang 15]. The proposed hierarchical representation model used a two-layer model. One-layer aggregated all the sequential transactions, and in the second layer, this sequential information was aggregated with the user's overall taste. Then the combined information was used to predict item in the next transaction. This method was novel by setting different layers to combine two kinds of information. However, a better method has been proposed to learn item representations.

Contact: Na Lu, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656 Japan, Department of Systems Innovation, School of Engineering,The University of Tokyo, Bldg.No.8. 507 , 080-1241-0956, luna@g.ecc.u-tokyo.ac.jp

For understanding sequence data, we utilize the Skip-Gram model for word representation learning in natural language process field [Mikolov 13] named as word2vec. Skip-Gram model learns word representations by predicting the context of this word. More precisely, word2vec get a word vector in a lower dimensional space compared with one-hot representation. This method was later generalized as item2vec for learning item representations [Barkan 16]. Item2vec treats users' subsequent behavior as a sentence in word2vec and creates item vectors.

By learning users' sequential data to generate item representations, we proposed a method for aggregating users' history behavior and general taste to build a recommendation system.

2. RELATED WORKS

A good recommendation system could improve users' decision-making process in this information overload era. The widely used recommendation methods include collaborative filtering, content-based filtering, and hybrid filtering. Despite the use of traditional methods, many approaches are proposed to improve the quality of recommendations. We first review some related work in this field.

2.1 Sequential Pattern Mining

Pattern mining is an essential branch of data mining, which consists of discovering frequent itemsets, associations, sub-graphs, sequential rules, etc. [Chen 96]. The target of sequential pattern mining is to detect sequential patterns by analyzing a set of sequential data, in which occurrence frequency is one of the target [Pei 04]. Item2vec embeds items into a low-dimensional representation by accounting the item co-occurrence in user records. That is, this model could generally capture the co-occurrence patterns of items in each transaction data.

2.2 Personalized Ranking

From the target of the recommendation system, it can be treated as a rating prediction problem or a personalized ranking problem [Rendle 09]. The task of personalized ranking is to provide a user with a ranked list of items, which matches a real-life scenario. An example is that an online retailer wants to give a personalized ranking item list that a user may probably buy in the recent future. Former research for personalized ranking algorithms optimized through learning users' preferences on a set of items, which include BPR [Rendle 09], CliMF [Shi 12].

2.3 Item Representation Learning

The word embedding method [Mikolov 13] have attracted much attention from fields besides NLP. The recommendation is also to utilize this method for better performance, including clustering [Barkan 16] and regression. Representation learning in recommendation means getting relationships between items from a specific data set, which is called item embedding. Barkan and Koenigstein [Barkan 16] first proposed Item2Vec model which based on a neural item embedding model for collaborative filtering. In this method, item embedding is used to learn a better item representation but fail to give a personalized ranking recommendation. In this research, we propose an item embedding based method combined with users' history behaviors to provide a personalized next-item recommendation.

3. PROBLEM STATEMENT

In this section, we first introduce the problem formalization of recommendation based sequence behavior. We then describe the item embedding and recommendation for the next item in detail. After that, we talk about the learning and prediction procedure of this method.

3.1 Formalization

Let $U = \{u_1, u_2, ..., u_{|U|}\}$ be a set of users and $I = \{i_1, i_2, ..., i_{|I|}\}$ be a set of items, in which $|U|$ and $|I|$ denote to the total number of unique users and items, respectively. For each user u, the transaction history data T^u is given by $T^u = (T^u_1, T^u_2, T^u_3, ..., T^u_t)$, where $T^u_t \subseteq I$. The purchase history of all users is denoted as $T = \{T^1, T^2, T^3, ..., T^t\}$. Given the transaction data of all users, our task is to predict what the user will probably buy in the next time (eg. t-th), which is denoted as $R = \{R^1, R^2, R^3, ..., R^u\}$. Every R^i includes k items as recommendation: $R^i = \{R^i_1, R^i_2, ..., R^i_k, \}$.That is, we need to generate a personalized ranking R^i for user u_i in t-th transaction.

3.2 Item2Vec algorithm

Our purpose is to learn a recommendation model from a sequential transaction data which could also combine users' overall taste. In this section, we first explain Item2Vec algorithms in detail, which generate item embedding from sequential data. Then users' general taste will be concluded from one user's whole transaction data. At last, item representations and users' general taste will be combined to create a personalized ranking for a next-item recommendation.

To proposed our method for personalized ranking from a sequential user transaction data, we first need to have a look at Item2Vec specifically. Skip-gram with negative sampling (SGNS) was first introduced in word embedding by Mikolov et al. [Mikolov 13]. The neural embedding in natural language processing attempts to map words and phrases into a vector space of low-dimensional semantics and syntax. Skip-gram uses the current word to predict its context words. The item collection in Item2vec is equivalent to the sequence of words in word2vec, that is, the sentence. Commodity pairs that appear in the same collection are considered positive. For the set $w_1, w_2, ..., w_K$ objective function:

$$\frac{1}{K}\sum_{i=1}^{K}\sum_{j\neq i}^{K} log(w_j|w_i) \tag{1}$$

Same as word2vec, using negative sampling, define $p(w_j|w_j)$ as:

$$p(w_j|w_j) = \sigma(u_i^T v_j)\prod \sigma(-u_i^T v_k) \tag{2}$$

Finally, the SGD method is used to learn the max of the objective function and to obtain the embedding representation of each item. The cosine similarity between the two items is the similarity of items.

The cosine similarity between two vectors can be formalized as:

$$cos(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1||v_2|} \tag{3}$$

3.3 Proposed method

From Item2Vec method, all users' transaction data $T = \{T^1, T^2, T^3, ..., T^t\}$ is used to learn item representations. More specifically, Item2Vec algorithm inputs a large corpus of transactions and creates a vector space, in which every unique item is transformed as a vector in this space. Based on this, we produce item representations based on users' sequential transaction data.

The advantage of our methods is that we can introduce aggregation operations in forming user representations from their history transaction data. In this work, we propose two aggregation methods to get a user representation.

The first is average pooling. This method construct one vector by taking the average value from a set of vectors. Let $V = \{v_1, v_2, v_3, ..., v_l\}$ be a set of vectors. Average pooling of V can be formalized as:

$$f_{ave}(V) = \frac{1}{l}\sum_{i=1}^{l} v_i \tag{4}$$

Second is max pooling. This method construct one vector by taking the max value from a set of vectors. Thus, max pooling can be formalized as:

$$f_{max}(V) = \begin{bmatrix} max(v_1[1] & ... & v_l[1]) \\ max(v_1[2] & ... & v_l[2]) \\ ... & ... & ... \\ max(v_1[n] & ... & v_l[n]) \end{bmatrix} \tag{5}$$

From a user's transaction data T^i, we can get a user representation $\vec{u_i}$ from $f_{ave}(T^i)$ and $f_{max}(T^i)$ as $u_{i\vec{ave}}$ and $u_{i\vec{max}}$. Combine with top-K recommendation from item embedding, which is R^i, we re-rank R^i based on the weighted similarity with user u_i. The detail of re-ranking of recommendation R^i is in Algorithm 1.

In this way, we can combine u_i's general taste ($u_{i\vec{ave}}$ and $u_{i\vec{max}}$) and sequential prediction (R^i) to get a overall prediction.

Algorithm 1 Combination of user representation and top-K recommendation

Input: top-K recommendation R^i for u_i, user vector $u_{i\vec{ave}}$ and $u_{i\vec{max}}$, item set I

Output: R^i_{ave} and R^i_{max}

for $j \leq top - K * 2$ **do**
  if $R^i_j \subseteq I$ **then**
    $R^i_{ave-j} = cos(R^i_j, u_{i\vec{ave}})$ and $R^i_{max-j} = cos(R^i_j, u_{i\vec{max}})$
  else
    $test\_size - 1$
  end if
end for
sort R^i_{ave} and R^i_{max} from highest to lowest, choose top-K items from R^i_{ave} and R^i_{max}
return R^i_{ave} and R^i_{max}

Dataset name	# users	# items	# T
Online Retail	90,346	2553	397,923

Table 1: Basic Information about Online Retail dataset

4. EXPERIMENT AND DISCUSSION

In this section, we conduct empirical experiments to test the effectiveness of our method for a next-item recommendation. We first introduce the experimental data set, the baseline methods in our experiments. Then we compare our approach with the baseline model to study the effect of different aggregations. Finally, we make some analysis on the result of the experiments.

4.1 Dataset

We conduct our experiment on an open data set named 'Online Retail dataset' [UCI 15]. This data set includes transaction data from 2010.12.01 to 2011.12.09. Every row includes invoice number, product number, product name, sale quantity, sale time, unit price, customer ID, and customer's country. After deleting the row that has a default value, the data set basic information is in Table 1.

4.2 Evaluation and Discussion

We divided the dataset into train data and test data. Train data was used to train item2vec model to generate the item representations. Test data was used to evaluate the effectiveness of our method.

In the test data, we first remove the last transaction data from user u. So the remaining is $T^u_{n-1} = \{i_1, i_2, i_3, ..., i_{t-1}\}$. We use the learned model and remaining T^u to make a recommendation of top-K items located closer to each item in the learned vector space. Then these top-K items and user vector derived from T^u_{n-1} are combined to make the final top-K recommendation.

Here we use Recall as the prediction evaluation. The recall is formalized as below:

$$Recall(T^u_t, R^u_t) = \frac{T^u_t \bigcap R^u_t}{T^u_t}$$

Method / top-K	Ave	Max
1	14.98%	20.25%
3	6.41%	8.43%
5	8.95%	7.54%
10	4.57%	6.13%
15	1.47%	7.75%
20	3.37%	3.85%

Table 2: Recall percentage improvement compared with baseline method

In our experiment, we set top-K=1,3,5,10,15,20 as the number of items that would be recommended to user u. In this experiment, the baseline method is the prediction derived from the item2vec method, which was not combined with a user vector. The comparison of these methods are as follows.

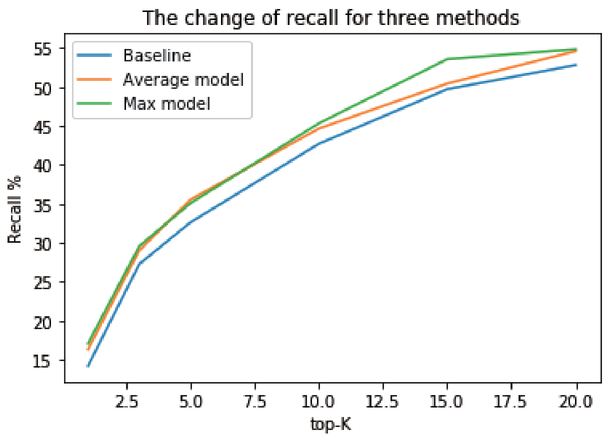

Figure 1: The change of recall for three methods

We can see that the average model and max model could improve over 10% of recall compared with the baseline model. That means if we recommend one item for a user, our model performed well by aggregating user's vector and item2vec prediction. However, this improvement declined with the increase in top-K, which means if we recommend a lot of items to a user at one time, our improvement is not as effective as recommending fewer items. Compared with the baseline model, the recall of the average model and max model are higher, and they get higher with the increase of top-K. If we provide more items for a user, the probability of correct prediction will be higher, just as Figure 1 shows above.

5. CONCLUSION

Representation learning has attracted many attentions in recommendation field for describing local item relationships. In this paper, we utilize the item embedding method to learn item representations from sequential transaction data. And we also constructed user representations to get a ranked list of items for a user. The experiment result demonstrated that our proposed method for next-item recommendation outperformed baseline model in prediction recall. Specifically, our models get 14.98% and 20.25% improvement compared with baseline model in a top-1 recommendation, which means we get a distinct improvement when the number of the recommended item is relatively small.

6. ACKNOWLEDGEMENT

This work was funded by JSPS KAKENHI, JP16H01836, JP16K12428, and industrial collaborators.

References

[Chen 12] Chen, Shuo, et al. "Playlist prediction via metric embedding." Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012.

[Rendle 10] Rendle, Steffen, Christoph Freudenthaler, and Lars Schmidt-Thieme. "Factorizing personalized markov chains for next-basket recommendation." Proceedings of the 19th international conference on World wide web. ACM, 2010.

[Wang 15] Wang, Pengfei, et al. "Learning hierarchical representation model for nextbasket recommendation." Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, 2015.

[Mikolov 13] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.

[Barkan 16] Barkan, Oren, and Noam Koenigstein. "Item2vec: neural item embedding for collaborative filtering." Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on. IEEE, 2016.

[Chen 96] Chen, Ming-Syan, Jiawei Han, and Philip S. Yu. "Data mining: an overview from a database perspective." IEEE Transactions on Knowledge and data Engineering 8.6 (1996): 866-883.

[Pei 04] Pei J, Han J, Mortazavi-Asl B, et al. Mining sequential patterns by pattern-growth: The prefixspan approach[J]. IEEE Transactions on Knowledge & Data Engineering, 2004 (11): 1424-1440.

[Rendle 09] Rendle, Steffen, et al. "BPR: Bayesian personalized ranking from implicit feedback." Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, 2009.

[Shi 12] Shi, Yue, et al. "CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering." Proceedings of the sixth ACM conference on Recommender systems. ACM, 2012.

[UCI 15] https://archive.ics.uci.edu/ml/datasets/Online+Retail