

An Attribute-aware Neural Attentive Model for Next Basket Recommendation

Ting Bai^{1,2,3}, Jian-Yun Nie³, Wayne Xin Zhao^{1,2,*}, Yutao Zhu^{1,2}, Pan Du³, Ji-Rong Wen^{1,2}

¹ School of Information, Renmin University of China

² Beijing Key Laboratory of Big Data Management and Analysis Methods

³ Department of Computer Science and Operations Research, University of Montreal

{baiting,ytzu}@ruc.edu.cn, {batmanfly,jirong.wen}@gmail.com, {nie,pandu}@iro.umontreal.ca

ABSTRACT

Next basket recommendation is a new type of recommendation, which recommends a set of items, or a basket, to the user. Purchase in basket is a common behavior of consumers. Recently, deep neural networks have been applied to model sequential transactions of baskets in next basket recommendation. However, current methods do not track the user's evolving appetite for items explicitly, and they ignore important item attributes such as product category. In this paper, we propose a novel Attribute-aware Neural Attentive Model (ANAM) to address these problems. ANAM adopts an attention mechanism to explicitly model user's evolving appetite for items, and utilizes a hierarchical architecture to incorporate the attribute information. In specific, ANAM utilizes a recurrent neural network to model the user's sequential behavior over time, and relays the user's appetite toward items and their attributes to next basket through attention weights shared across baskets on the two different hierarchies. Experiment results on two public datasets (*i.e.*, Ta-Feng and JingDong) demonstrate the effectiveness of our ANAM model for next basket recommendation.

CCS CONCEPTS

• Information systems → Recommender systems; • Computing methodologies → Neural networks;

KEYWORDS

Attribute-aware model; Hierarchical attentive architecture; Next basket recommendation

1 INTRODUCTION

Recommender systems provide great help for users to find their desired items from a huge number of offers. Most studies have focused on item recommendation, where each item is

recommended separately. In most real scenarios, users often purchase a basket of items at a visit of an online store. A basket contains several items the user purchases together. The next basket recommendation is to predict the next few items that the user would likely buy. The key difference with item recommendation is that items in a basket can be dependent. For example, it is more likely that a user puts bread and beer in the same basket than bread and wrenches. Next basket recommendation is also different from session-based recommendation because the order to put items in the basket is not as important as in a session. The items that a user would put in his basket are certainly dependent on the general interests of the user, but are also dependent on the items that the user has purchased in his previous baskets. Both elements reflect the user's appetite for items, which often evolves over time.

Detecting the purchase appetite of users and their evolution in time has been an active research topic in recent years [1, 3–5]. Three main approaches have been proposed to model the sequential behaviors of a user in next basket recommendation, which are respectively based on: purchase pattern, Markov Chains (MC) and Recurrent Neural Network (RNN). Pattern-based method [1] considers the correlation among items within the same basket, and incorporates different product factors (*e.g.*, co-occurrence, periodicity) into the decision process. Factorizing Personalized Markov Chains (FPMC) [3] models both user's sequential behavior and general taste by conducting a tensor factorization over the transition cube. Hierarchical Representation Model (HRM) [4] improves FPMC by employing a two-layer architecture to construct a no-linear hybrid aggregation of the user vector and the transaction representation. Notice that these two MC-based methods model the sequential behaviors of users only between adjacent transactions, which is insufficient to capture the long-term trend of baskets. To address this problem, Dynamic REcurrent bAasket Model [5] (DREAM) adopts RNN to model global sequential features which reflect interactions among baskets, and uses the hidden state of RNN to represent user's dynamic interests over time.

However, the previous RNN methods ignore the attributes of item, *e.g.*, category and price, which are crucial in the user's purchase decision. For instance, if a user begins to purchase some products for babies, he or she is more likely to purchase other products in that category in the near future. Based on the above observations, we propose an Attribute-aware Neural Attentive Model (ANAM) for next basket recommendation.

* Corresponding Author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5657-2/18/07...\$15.00

<https://doi.org/10.1145/3209978.3210129>

Instead of using the combined representation of item and its attributes as the input of recommendation model, we propose a novel hierarchical architecture to apply the independent attention mechanism to items and attributes respectively. Then we apply a joint learning function to combine users varying appetite towards items and attributes. ANAM utilizes an attentive RNN to model the user’s sequential behavior over time, and relays the user’s appetite for items and their attributes to next basket through attention weights shared across baskets in the two different hierarchies.

Our contributions are summarized as follows: (1) We propose a hierarchical attentive architecture to explicitly model the user’s appetite for any attributes of items (*e.g.*, category); (2) Using a joint learning function combining the attentive information of items and attributes, our model is more effective to capture user’s varying appetite towards items; (3) We demonstrate through two real-world datasets the effectiveness of our model for the next basket recommendation task.

2 ATTRIBUTE-AWARE NEURAL ATTENTIVE MODEL

2.1 Problem Statement

Assume that we have a set of users and items, denoted by U and I respectively. Let $u \in U$ denote a user and $i \in I$ denote an item. The number of users and items is denoted as $|U|$ and $|I|$ respectively. Given a user u , his or her purchase records sorted by time is a sequence of baskets $B^u = \{B_1^u, B_2^u, \dots, B_t^u\}$. t is the step of the sequence of baskets. $B_t^u \subseteq I$ consists of a set of items. Each item i has some attributes, such as the category and price. Currently we use the category information as item attributes, but our model could be easily extended to characterize other attribute information. The attribute of item i is denoted as $c_i \in C$, where C is the set of categories. The category information of items in basket B_t^u is denoted as C_t . Based on the above notations, given a user u ’s purchase history, the next basket recommendation task can be defined as a prediction problem which aims to infer a set of items that u would probably buy in the next basket. Such a prediction problem can be reformulated as a ranking problem of all items for each user. With the ranking list of all items, we recommend top K items to the user.

2.2 The Proposed Model

In this paper, we propose a unified Attribute-aware Neural Attentive Model (ANAM) using the architecture shown in Fig. 1. ANAM utilizes a hierarchical attentive RNN to model the user’s sequential behavior over time, and relays the user’s appetite for items and their attributes to next basket through attention weights shared across baskets. In the following, we first model the information of a basket: encoding information of items and item attributes in each basket; and learning the joint function to integrate the corresponding attention weights of items and attributes. Then we model user’s sequential behavior by RNN.

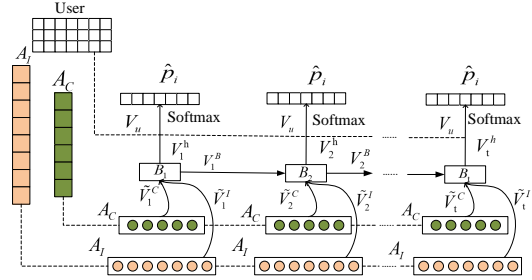


Figure 1: Overview of the architecture of ANAM

2.2.1 Encoding items and item attributes. For a basket B_t^u at step t , we represent the information of item set I_t using a $|I_t|$ -dimensional one-hot representation, denoted by $\mathbf{e}_t^I \in \mathbb{R}^{|I_t| \times 1}$, only the entry corresponding to item which exists in basket B_t^u will be set to 1. The same for the set of categories C_t , denoted by $\mathbf{e}_t^C \in \mathbb{R}^{|C_t| \times 1}$. Then we apply a concatenation-based lookup layer to transform the one-hot vectors of I_t and C_t into latent vectors

$$\mathbf{v}_t^I = \text{CONCAT-LOOKUP}(\mathbf{P}^\top, \mathbf{e}_t^I), \quad (1)$$

$$\mathbf{v}_t^C = \text{CONCAT-LOOKUP}(\mathbf{Q}^\top, \mathbf{e}_t^C), \quad (2)$$

where $\mathbf{P} \in \mathbb{R}^{D \times |I|}$ and $\mathbf{Q} \in \mathbb{R}^{D \times |C|}$ are the transformation matrices for lookup and D is the embedding dimension of each item and category. $\mathbf{v}_t^I \in \mathbb{R}^{D \times |I_t|}$ and $\mathbf{v}_t^C \in \mathbb{R}^{D \times |C_t|}$ are the latent vector of items and item categories in basket B_t^u . $|I_t|$ is the number of items in B_t^u and $|C_t|$ is the number of categories in B_t^u . For each item $i_t \in I_t$, the corresponding embedding vector is $\mathbf{v}_t^I \in \mathbb{R}^{D \times 1}$; and for each category $c_t \in C_t$, it is $\mathbf{v}_t^C \in \mathbb{R}^{D \times 1}$. Since the number of items in each basket changes, we use a masked zero-padding value in the embedding layer to convert each basket to a fixed-dimension of representation vector.

2.2.2 Integrating attention weights. We employ attention mechanism to capture user’s varying appetite toward items and categories upon all baskets. For each item $i \in I$ and each category $c \in C$, we assume user’s appetite for the item and category is $\mathbf{a}_i \in \mathbf{A}_I$ and $\mathbf{a}_c \in \mathbf{A}_C$ respectively. $\mathbf{a}_i \in \mathbb{R}^{D \times 1}$ and $\mathbf{a}_c \in \mathbb{R}^{D \times 1}$ are initialized randomly and learned automatically as the training process over all baskets. $\mathbf{A}_I \in \mathbb{R}^{|I| \times D}$ and $\mathbf{A}_C \in \mathbb{R}^{|C| \times D}$ are the attention matrices of all items and categories (see Figure 1). For an item i in basket B_t^u , we obtain the attentive representation of i by $\tilde{\mathbf{v}}_t^I = \mathbf{v}_t^I \odot \mathbf{a}_i$, where “ \odot ” denotes the element-wise product of vectors. Now we have the latent vector \mathbf{v}_t^I of all items (see Eq. 1) and \mathbf{v}_t^C of all categories (see Eq. 2) in B_t^u , we integrate attention weights with the latent vectors of items and categories as follows

$$\tilde{\mathbf{v}}_t^I = \mathbf{v}_t^I \odot \text{CONCAT}(\mathbf{a}_i | i \in I_t), \quad (3)$$

$$\tilde{\mathbf{v}}_t^C = \mathbf{v}_t^C \odot \text{CONCAT}(\mathbf{a}_c | c \in I_t), \quad (4)$$

where “CONCAT” function concatenates the attentive vectors of the items or categories in the basket.

Intuitively, higher attention to a category makes the product in it more likely to be purchased. We adopt a joint learning function by applying an element-wise product, which incorporates the attentive vectors of items and categories into a unified vector $\mathbf{v}_t^B \in \mathbb{R}^{D \times |I_t|}$ to represent the basket.

$$\mathbf{v}_t^B = \tilde{\mathbf{v}}_t^I \odot \tilde{\mathbf{v}}_t^C. \quad (5)$$

2.2.3 Modeling user’s sequential behavior. Given a sequential baskets records $B^u = \{B_1^u, B_2^u, \dots, B_t^u\}$ of a user u , we obtain the basket representation \mathbf{v}_t^B of B_t^u in Eq. 5. The sequence of baskets of user u can be represented as $\mathbf{v}^B = \{\mathbf{v}_1^B, \mathbf{v}_2^B, \dots, \mathbf{v}_t^B\}$. For a user u , we represent it using a $|U|$ -dimensional one-hot representation, denoted by $\mathbf{e}_u \in \mathbb{R}^{|U| \times 1}$, only the entry corresponding to u will be set to 1. Then we apply a lookup layer to transform the one-hot vectors of u into latent vectors

$$\mathbf{v}_u = \text{LOOKUP}(\mathbf{W}^\top, \mathbf{e}_u), \quad (6)$$

where $\mathbf{W} \in \mathbb{R}^{D \times |U|}$ is the transformation matrices for lookup.

In order to model user’s sequential behavior, we adopt Long Short-Term Memory (LSTM), which has proven effective at modeling sequential data. The input of LSTM at step t is \mathbf{v}_t^B . The output of LSTM at step t (*i.e.*, hidden state) is represented as $\mathbf{v}_t^h \in \mathbb{R}^{D \times 1}$, and it is applied later to construct our loss function.

2.3 The Loss Function for Optimization

For a user u and his or her previous baskets $B_{1:t}^u$, we define the probability of an item i being purchased in the next basket B_{t+1}^u by softmax function

$$p(i \in B_{t+1}^u | u, B_{1:t}^u) = \frac{\exp(\mathbf{v}_t^I \cdot (\mathbf{v}_u \odot \mathbf{v}_t^h))}{\sum_{j=1}^{|I|} \exp(\mathbf{v}_j^I \cdot (\mathbf{v}_u \odot \mathbf{v}_t^h))}, \quad (7)$$

where $\mathbf{v}_u \in \mathbb{R}^{D \times 1}$ is the embedding vector of user u , and $\mathbf{v}_t^h \in \mathbb{R}^{D \times 1}$ is the hidden vector of LSTM at step t .

To effectively learn from the training data, we adopt a weighted cross-entropy as the optimization objective at each step of LSTM, which is defined as

$$L = \sum_{u \in U} \sum_{B_t^u \in B^u} \sum_{i \in I_t} (-m \cdot y_i \cdot \log \hat{p}_i - n \cdot (1 - y_i) \cdot \log(1 - \hat{p}_i)), \quad (8)$$

where \hat{p}_i is the probability of an item i being purchased in the next basket in our model. If item i is purchased in the next basket, $y_i = 1$, otherwise, $y_i = 0$. m and n are the weights of positive and negative instances (purchased or not in the next basket). The reason of using different weights is to cope the fact that there are usually much more negative instances than positive instances in a dataset.

We take the last basket of each user as the testing data, the penultimate basket as the validation set to optimize parameters, and the remaining baskets as the training data. We implement our models in PYTHON using the library KERAS. The loss function in Eq. 8 is optimized by Adam with a batch size of 200 in Ta-Feng and 500 in JingDong datasets. Due to the disparity of the amount of positive and negative instances, we set m 500 times larger than n in our experiments to punish the error of mistaking the positive instances. The

Table 1: Statistics of the evaluation datasets.

Datasets	# Users	#Items	# Transactions	# Category
Ta-Feng	9,238	7,973	464,118	1,074
JingDong	4,832	3,283	41,932	165

learning rate is set to 0.001 and embedding size in the input layer and the units in LSTM are set to 50 (*i.e.*, $D = 50$).

After training, given a user’s historical transaction records, we can obtain the probability of each item i being purchased in the next basket according to Eq. 7. We then rank the items according to their probability, and select top K results as the final recommended items to the user.

3 EXPERIMENTS

Dataset. We experiment with two real-world datasets, namely Ta-Feng¹ and JingDong. Ta-Feng dataset contains 4 months (November 2000 to February 2001) of shopping transactions of the Ta-Feng supermarket. JingDong dataset contains product reviews records of users in 4 months (January 2012 to April 2012), and is shared in [6]. The users having less than 10 and 25 purchases in Ta-Feng and JingDong are removed, so are the items purchased less than 10 and 20 times. The average number of baskets for a user in Ta-Feng and JingDong datasets is 8.4 and 8.7, and the average number of products in each basket is 6.6 and 4.1. The statistics of the two datasets are summarized in Table 1.

Evaluation metrics. Following [4, 5], we choose the top K (*i.e.*, $K = 5$) items in the ranking list of all items as the recommended set. To evaluate the performance of our model, we adopt the widely used F1-score and Normalized Discounted Cumulative Gain (NDCG).

Baseline methods compared. We consider the following baselines for performance comparisons. (1) *TOP*: It ranks the items according to their popularity. (2) *NMF* [2]: It is the state-of-the-art method in traditional model based collaborative filtering methods. It uses nonnegative matrix factorization on user-item matrix. For implementation, we use the public code NMF: DTU Toolbox². (3) *FPMC* [3]: It learns a transition matrix based on underlying Markov chains. It models the sequential behavior but only between adjacent transactions. (4) *HRM* [4]: It employs neural network to implement a nonlinear operation to integrate the representation of users and item purchase history in last transactions. (5) *DREAM* [5]: It incorporates both general customers’ preferences and sequential information by using RNN. It is the state-of-the-art method in next basket recommendation task.

Among all the above method, NMF does not model the basket information. FPMC and HRM only use the sequence information between two adjacent baskets. DREAM uses RNN to capture the global sequential information, but does not use attention mechanism to effectively model the sequence

¹<http://www.bigdatalab.ac.cn/benchmark/bm/dd?data=Ta-Feng>

²<http://cogsys.imm.dtu.dk/toolbox/nmf/>

information. Besides, all of the above methods do not consider item attributes. Our ANAM employs a hierarchical attentive architecture to apply attention mechanism to items and attributes respectively, which can effectively model users' varying appetite towards items. To empirically evaluate the effectiveness of the attention mechanism and attribute information respectively, we compare our ANAM with DREAM - an RNN model without attention and attribute information, and a degenerated Neural Attentive Model (NAM) - our attentive RNN model without attribute information.

Results and analysis. We present the results of F1-score@5 and NDCG@5 on the next basket recommendation performance in Table 2.

Table 2: Performance comparisons of different methods on the next basket recommendation task.

Datasets	Ta-Feng		JingDong	
Models	F1-score@5	NDCG@5	F1-score@5	NDCG@5
TOP	0.051	0.084	0.0066	0.0094
NMF	0.052	0.072	0.0069	0.0097
FPMC	0.059	0.087	0.0078	0.0099
HRM	0.062	0.089	0.0095	0.0174
DREAM	0.065	0.084	0.0122	0.0123
DREAM*	0.133	0.173	0.1046	0.1542
NAM	0.142*	0.187*	0.1283*	0.1826*
ANAM	0.146*	0.190*	0.1313*	0.1842*

Note: “*” indicates the statistically significant improvements (*i.e.*, two-tailed *t*-test with $p < 0.01$) over the best baseline (*i.e.*, DREAM*).

In our experiments, the embedding dimension in HRM and DREAM is set to 50. For JingDong dataset, we set the learning rate to 0.001 in Dream, 0.0003 in HRM and 0.001 in FPMC which yield the best results. For Ta-Feng, we use the same training and testing data as HRM, so we report the baseline results as in [4]. We re-implement the DREAM, and modify the objective function as Eq. 8. This change leads to a huge improvement of the original results in [5]. The parameters of our modified DREAM (denoted as DREAM*) are the same as in our ANAM model. We report the results of DREAM* in Table 2.

We can make the following observations: (1) TOP is the weakest baseline, since it is a non-personalized method. NMF performs better than TOP, but it does not consider any sequential information of users. FPMC outperforms slightly NMF by taking into account adjacent baskets. (2) HRM further improves the effectiveness by using neural network. This shows the ability of neural network to model complex interactions between user's general taste and their sequential behavior. Compared with HRM, DREAM achieves better effectiveness on F1-score@5 and NDCG@5 due to use of whole sequential information. (3) DREAM*, which uses a modified loss function, leads to a large boost in effectiveness. This indicates the great importance to weigh the training examples in the training process. (4) Our degenerated model NAM consistently and significantly outperforms all baseline

methods, showing the effectiveness of our attentive mechanism on item to capture user's evolving appetite for items. (5) ANAM performs better than NAM, This indicates the contribution of the attribute information of item.

4 CONCLUSION

This paper presented a novel attribute-aware neural attentive model for next basket recommendation, which utilizes a hierarchical attentive architecture to integrate the attribute information of items. ANAM effectively captures the user's evolving appetite for the item by using a joint learning function combining the attentive information of items and attributes. Experimental results on two public datasets (*i.e.*, Ta-Feng and JingDong) demonstrated the effectiveness of our ANAM model for next basket recommendation. This work shows the necessity to model the baskets in the purchase history of a user, and to incorporate the attribute information about items. As future work, we will investigate more attributes of item (*e.g.*, price) and explore the effects of multiple factors on the user's purchase decision. This series of experiments confirms the previous results that the sequential information about baskets provides some useful information for next basket prediction. The superior performance of our model shows that attribute information can further boost the effectiveness.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61502502, the National Basic Research 973 Program of China under Grant No. 2014CB340403 and the Beijing Natural Science Foundation under Grant No. 4162032. Ting Bai was supported by the Outstanding Innovative Talents Cultivation Funded Programs 2016 of Renmin University of China, and partly by an NSERC discovery grant.

REFERENCES

- [1] Riccardo Guidotti, Giulio Rossetti, Luca Pappalardo, Fosca Giannotti, and Dino Pedreschi. 2017. Next Basket Prediction using Recurring Sequential Patterns. *arXiv preprint arXiv:1702.07158* (2017).
- [2] Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*. 556–562.
- [3] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 811–820.
- [4] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. ACM, 403–412.
- [5] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 729–732.
- [6] Xin Wayne Zhao, Yanwei Guo, Yulan He, Han Jiang, Yuexin Wu, and Xiaoming Li. 2014. We know what you want to buy: a demographic-based system for product recommendation on microblogs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1935–1944.