Conceptualize and Infer User Needs in E-commerce

Xusheng Luo*
Alibaba Group
lxs140564@alibaba-inc.com

Yonghua Yang
Alibaba Group
huazai.yyh@alibaba-inc.com

Kenny Q. Zhu †
Shanghai Jiao Tong University
kzhu@cs.sjtu.edu.cn

Yu Gong
Alibaba Group
gongyu.gy@alibaba-inc.com

Keping Yang
Alibaba Group
shaoyao@taobao.com

ABSTRACT

Understanding latent user needs beneath shopping behaviors is critical to e-commercial applications. Without a proper definition of user needs in e-commerce, most industry solutions are not driven directly by user needs at current stage, which prevents them from further improving user satisfaction. Representing implicit user needs explicitly as nodes like "outdoor barbecue" or "keep warm for kids" in a knowledge graph, provides new imagination for various e-commerce applications. Backed by such an e-commerce knowledge graph, we propose a supervised learning algorithm to conceptualize user needs from their transaction history as "concept" nodes in the graph and infer those concepts for each user through a deep attentive model. Offline experiments demonstrate the effectiveness and stability of our model, and online industry strength tests show substantial advantages of such user needs understanding.

1 INTRODUCTION

Intuitively, knowing what users need in their mind when they come to the shopping platform is vital to e-commerce giants like Alibaba and Amazon. However, user needs in e-commerce are not well defined, making it difficult for various e-commerce applications to truly understand their users, which gradually becomes the bottleneck to further improve user satisfaction in e-commerce. For example, item recommendation, one of the major applications in e-commerce, widely adopts the idea of item-based *collaborative filtering (CF)* [? ?]. The recommender system uses user's historical behaviors as triggers to recall a small set of most similar items as candidates, then recommends items with highest weights after scoring with a ranking model. A critical shortcoming of this framework is that it is not driven by user needs in the first place, which inevitably makes it hard for the recommender system to jump out of historical behaviors to explore other implicit user needs. Besides,

items recommended are hard to be explained except for trivial reasons such as "similar to those items you have already viewed or purchased". Therefore, despite its widespread use, the performance of current recommendation systems is still under criticism. Users are complaining that some recommendation results are redundant or lack of novelty, since current recommender systems can only satisfy very limited user needs such as the needs for a particular category or brand. Without the ability of inferring user needs comprehensively and accurately, it is difficult for current systems to recommend items which a user may never think of but potentially have interests on, or provide convincing recommendation reasons to help users make shopping decisions.

In this paper, we attempt to conceptualize various implicit user needs in e-commerce scenarios as explicit nodes in a knowledge graph, then infer those needs for each user. By doing that, our platform is able to suggest a customer "other items you will need for outdoor barbecue next week" after he purchases a grill and clicks on charcoals, or remind him of preparing clothes, hats or scarfs that can "keep warm for your kids" as there will be a snowstorm coming next week. Different from most e-commerce knowledge graphs, which only contain nodes such as categories or brands, a new type of node, e.g., "Outdoor Barbecue" and "Keep Warm for kids", is introduced as bridging concepts connecting user and items to satisfy some high-level user needs or shopping scenarios. We call these nodes "**e-commerce concepts**", whose structure represents a set of items from different categories with certain constraints (more details in Section 2) . These e-commerce concepts, together with categories, brands and items, form a new kind of e-commerce knowledge graph, called **"E-commerce Concept Net"** (Figure 1 (a)). For example, "Outdoor Barbecue" is one such e-commerce concept, consisting of product categories such as charcoal, forks and so on, which are items required to host a successful outdoor barbecue party.

There are several possible practical scenarios in which inference of such e-commerce concepts from user behaviors can be useful. The first scenario is coarse-grained recommendation, where inferred concepts can be directly recommended to users together with its associated items. Figure 2(a) shows the real implementation of this idea in *Taobao* [1] App. Among normal recommended items, concept "Tools for Baking" is displayed to users as a card with its name and the picture of a representative item (left). Once a user clicks on it, he will enter into another page (right) where different items needed for baking are displayed. In this way, the
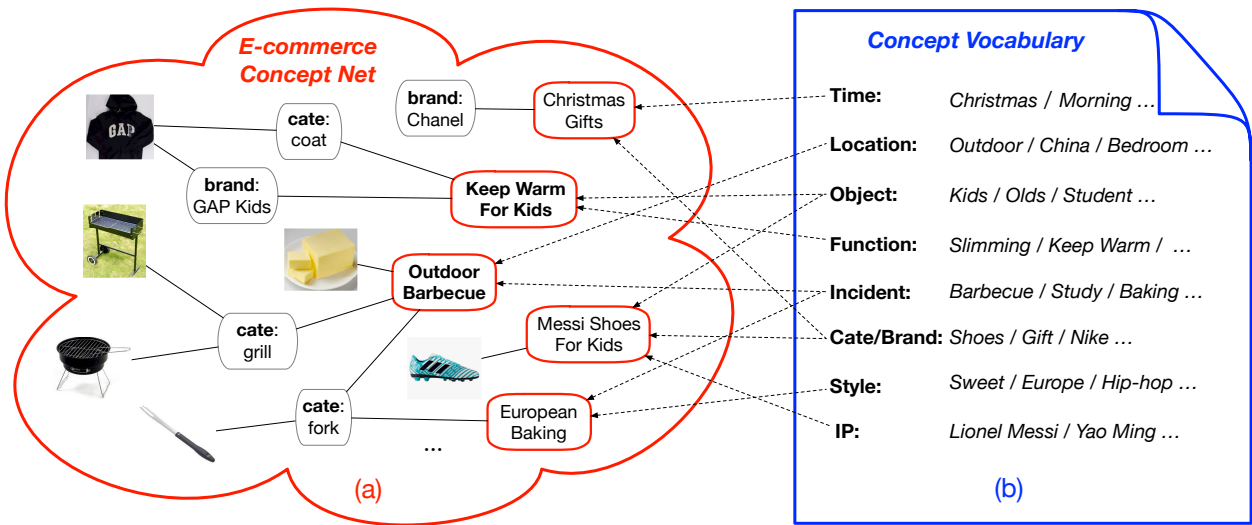
[1]http://www.taobao.com

Figure 1: (a) Overview of "E-commerce Concept Net", where concepts are marked by red rectangles and pictures are example items. (b) Overview of concept vocabulary, where each concept can be expressed using the values from eight different domains.

recommender system is acting like a salesperson in a shopping mall, who tries to guess the needs of his customer and and then suggests how to satisfy them. If their needs are correctly inferred, users are more likely to accept the recommended items. The second scenario is providing explanations for item recommendation as shown in Figure 2(b). While explainable recommendation attracts much research attention recently [?], most existing works are not practical enough for industry systems, since they are either too complicated (based on NLG [? ?]), or too trivial (e.g., "how many people also viewed" [? ?]). Our proposed concepts, on the contrary, precisely conceptualize user needs and are easy to understand. This idea is currently experimented in Taobao at the time of writing. Other possible scenarios can be query rewriting or query suggestion in e-commerce search engine.

User needs inference backed by a knowledge graph (KG) is a relatively new problem. The most related work is incorporating KG into recommendation [? ? ?]. Prior efforts are mainly categorized into two types. Path based methods [? ?] explore the various patterns of connections among items in KG, providing rich *meta-path* based features for user-item recommendations. Those methods generally treat KG as a heterogeneous information network (HIN) and rely on manually crafted meta-paths. The other line of research [? ?] leverage knowledge graph embedding (KGE) such as TransE [?], to bring extra information from KG to enhance the representation of items and users. However, KGE based methods usually lack the ability to reason across multiple hops and have not shown to be scalable on large-scale dataset. Different from most existing works targeting item (or movie/news), the target (concept) in our problem is a set of items, which itself has a non-trivial structure and contains much more information than a single item. In order to handle the informative input and provide more interpretability, we further extend the direction of path based works by proposing a deep interpretable model with a specially designed module called "attention cube", which aims to explore the mutual influences among

Figure 2: Two real examples of user-needs driven recommendation. (a) Display concepts directly to users as cards with a set of related items. (b) Concepts act as explanations in item recommendation.

users, concepts and paths connecting user-concept pairs within the concept net.

The contributions of this paper are summarized below:

- We formally define user needs in e-commerce and introduce "e-commerce concept net", a new genre of knowledge graph in e-commerce, where "concepts" can explicitly express various shopping needs for users.

- Based on the e-commerce concept net, we propose a path-based deep model with attention cube to infer user needs. We evaluate our model in both offline and online settings. Offline results show the model outperforms several strong baselines by a substantial margin of **2.4%** on AUC. Online testing deployed on a real recommender system in Taobao also achieves largest improvement on CTR and Discovery. **20.5%** improvements on User Satisfaction Rate further indicates the value of such user needs inference.
- Our model has already gone into production of Taobao, the largest e-commerce platform in China. We believe the idea of user needs understanding can be further applied in more e-commerce productions. There is ample room for imagination and further innovation in "user-needs driven" e-commerce.

2   E-COMMERCE CONCEPT NET

User needs in e-commerce, are not formally defined previously. Hierarchical categories and browse nodes [2] are ways of managing billions of items in e-commerce platforms and are usually used to represent user needs or interests [? ?]. However, user needs are far broader than categories or browse nodes. Imaging a user who is planning an outdoor barbecue, or who is concerned with how to get rid of a raccoon in his garden. They have a situation or problem but do not know what products can help. Therefore, tree-like structures such as hierarchical categories and browse nodes are not enough to represent those user needs.

In our e-commerce concept net [3], user needs are conceptualized as various shopping scenarios, also known as "e-commerce concepts". We **define** a proper concept being a short, fluent and reasonable phrase which naturally represents a set of items from different categories. In order to cover as many user needs as possible, a thorough analysis on query logs, product titles and other e-commercial text is conducted. Based on years of experience in e-commerce, each concept is expressed using values drawn from 8 different domains of an "e-commerce concept vocabulary", which is shown in Figure 1 (b). For example, "Outdoor Barbecue" can be written as "*Location*: outdoor, *Incident*: barbecue", and "Breakfast for Pregnancy" can be written as "*Object*: pregnant women, *Cate/Brand*: breakfast".

To form the complete e-commerce concept net, concepts are related to their representative items, categories, brands respectively, mainly adopting the idea of semantic matching [? ?]. It should be noticed that there is a hierarchy within each domain. For example, "Shanghai" is a city in "China" in the domain of *Location* and "pregnancy" is a special stage of a "woman" in the domain of *Object*. Vocabulary terms at different levels can be combined and result in different concepts. Accordingly, those concepts are naturally related to form a hierarchy as well. Besides the vocabularies to describe concepts, there are constraints to each concept. The aspects of concept *schema* include *gender*, *life stage* [4], etc. which actually corresponds to user profile. For example, the schema of

"Breakfast for Pregnancy" will be "*gender*: female, *life stage*: pregnancy", which indicates the group of users who are most likely to need this concept.

Ontology Vocab.	# Time	# Location	# Object	# Func.
	127	7,052	247	3,693
	# Inci.	# Cate/Bra.	# Style	# IP
	9,884	44,860	1,182	21,230
# Concepts (Raw)	35,211	# Concepts (Online)		7,461
# Items	1 billion	# Categories/Brands		19K/5.5M

Table 1: Statistics of E-commerce Concept Net.

Table 1 shows the statistics of the concept net used in this paper [5]. There are 35,211 concepts in total at current stage, among which 7,461 concepts are already deployed in our online recommender system, covering over 90% categories of Taobao and each concept is related with 10.4 categories on average.

Inspired by the construction of open-domain KGs such as Freebase [?] and DBpedia [?] which benefit various downstream applications [? ?], different kinds of KGs in e-commerce are constructed to describe relations among users, items and item attributes [? ? ?]. One famous example is the "Product Knowledge Graph" [6] of Amazon. Their KG mainly supports semantic search, aiming to help users search for products that fit their need with search queries like "items for picnic". The major difference is that they never conceptualize user needs as explicit nodes in KG as we do. In comparison, our e-commerce concept net introduces a new node to explicitly represent user needs. Besides, it becomes possible to link our e-commerce KG to open-domain KGs through the concept vocabulary, making our concept net even more powerful.

3   PROBLEM

In this section, we formally define the problem of user needs inference. Let U, V denote the sets of users, items respectively. The inputs of our problem are as follows:

1) User behavior on items. For each $u \in U$, a behavior sequence $b = \{b_1, b_2, \cdots, b_n\}$ is a list of behaviors in time order, where b_i is the ith behavior and b_n is the latest one. Each user behavior contains a user-item interaction, detailed as $b_i = <v_i, type_i, time_i>$, where $v_i \in V$, $type_i$ is the type of behavior, such as click or purchase, and $time_i$ denotes the specific time of the behavior.

2) E-commerce concept net. Concept net G consists of massive triples (h, r, t), where $h, t \in E$, $r \in R$ denote the head, tail and relation. E and R are entities and relations in the concept net. While most items in V can be linked to entities in E, some items may not, since the item pool in e-commerce platforms changes frequently. The set of all concepts in G is denoted as C.

3) Side information. For each user $u \in U$, we have corresponding profile information h, such as *gender*, *kid's life stage* and long-term preferred categories, etc. For each concept $c \in C$, we have its schema s introduced in Section 2;

---

[2]https://www.browsenodes.com/

[3]This section only gives a brief introduction of the e-commerce concept net, while more details will be discussed in a separate paper which will be released in the near future at https://github.com/angrymidiao/concept_net.

[4]Life stage is divided into: pregnancy, infant, kindergarten, primary school, middle school and high school in Taobao.

[5]Preview of concept data can be found at https://github.com/angrymidiao/concept_net.

[6]https://blog.aboutamazon.com/innovation/making-search-easier

Xusheng Luo, Yonghua Yang, Kenny Q. Zhu, Yu Gong, and Keping Yang

Given above inputs, the goal of user needs inference is to predict potential need in concept c for each user u. We aim to learn a prediction function $\hat{y}_{uc} = F(u, c; \theta)$, denoting the probability concept c is needed by user u, and θ is the model parameters.

4 APPROACH

Figure 3 gives an overview of the proposed model, which is a three-way architecture: a user, a candidate concept, and paths from the user to the concept. Given a user and a candidate concept, the model leverages rich features extracted from user behavior and profile, candidate concept schema and path context, then outputs a score, representing the probability of the user needs the candidate concept.

4.1 User Embedding

The representation for each user comes from two parts: user behavior sequence and user profile.

User Behavior Sequence

Each behavior consists of three things: the item, the behavior type and the behavior time. Due to enormous amount of items (over 1 billion) in e-commerce platform, we represent each item in behavior sequence using its description such as category, brand and shop, instead of directly using its id. This is for two reasons: to save memory for storing large amount of id embeddings and to avoid sparsity problem when encountering long-tail or new items while predicting. We consider four types of behavior: *click*, *bookmark*, *add to cart* and *purchase*. In addition, the day gap between the behavior and current time is also taken into account. Therefore, each behavior b_i can be represented as a multi-hot vector $[b_i^1, b_i^2, \cdots, b_i^F]$, where each one-hot vector b_i^f corresponds to one of the above mentioned feature and F is the total number. Then an embedding lookup layer shown in Figure 4 maps sparse behavior vector into a low-dimensional dense vector b_i:

$$b_i = [W_{lk}^1 b_i^1; W_{lk}^2 b_i^2; \cdots; W_{lk}^F b_i^F], \quad W_{lk}^f \in R^{d^f \times V^f} \quad (1)$$

where W_{lk}^f are parameters for embedding lookup layer, d^f is the dimension of dense vector and V^f is the vocabulary size.

Recurrent neural Networks (RNN) based models [? ?] assume a rigidly ordered sequence over data which is not always true for user behaviors in real-world applications such as e-commerce. Such left-to-right architectures may restrict the power of the historical sequence representations. Thus, we believe bidirectional model such as Transformer [?] with self-attention architecture is a more reasonable choice for modeling user behavior sequences. The embedding of user behavior sequence u_b is calculated as:

$$u_b = \textbf{Transformer}(b_1, b_2, \cdots, b_n) \quad (2)$$

User Profile

The aspects of user profile include *gender, age level, kid's gender, kid's life stage*, etc. We use a simple lookup layer similar to Eq. (1) to obtain the corresponding embedding for each profile aspect. Then we apply a function f_u to map the embedding list $u_h = [h_{\text{gender}}, h_{\text{age}}, \cdots]$ to a single vector as the representation of user profile u_p:

$$u_h = f_u(u_h) = f_u([h_{\text{gender}}, h_{\text{age}}, \cdots]) \quad (3)$$

where the simplest f_u is average pooling. Optimizations for f_u will be discussed in Section 4.5.

Finally we get the user embedding u by concatenation plus a fully connected layer:

$$u = \textbf{FC}([u_b; u_h]) \quad (4)$$

4.2 Concept Embedding

Similar to user embedding which comes from user behavior and user profile, we use two components to encode the candidate concept: concept id and concept schema. The representation of concept id c_i is obtained simply by lookup. For concept schema, we use embedding lookup layer to map one-hot vectors (aspects of concept schema) to dense vectors, then apply function f_c to obtain the representation of the concept schema c_s. Similar to the encoding of user profile, we then obtain concept embedding c:

$$c_s = f_c(c_s) = f_c([s_{\text{gender}}, s_{\text{age}}, \cdots]) \quad (5)$$

$$c = \textbf{FC}([c_i; c_s]) \quad (6)$$

4.3 Path Embedding

In order to leverage rich semantic features from the e-commerce concept net, we explore paths connecting users and concepts within the graph. We adopt the idea of *meta-path* [?], due to the fact KGs in e-commerce are usually extremely large. If we let the model discover possible paths from a behaved item to a concept freely as described in RippleNet [?], the computational overhead is unacceptable. Besides, empirical experience is valuable in e-commerce. Therefore, we believe manually crafted meta-paths are able to reduce noises and improve efficiency. A meta-path is a path in the form of "$T_1 \rightarrow T_2 \rightarrow \cdots T_n$", where each node (exclude user in our case) is a type of entity in the concept net, such as "$User \rightarrow Item \rightarrow Category \rightarrow Concept$". We mainly consider two types of meta-path in our concept net: *behavior* path and *preference* path. Behavior paths are triggered by items which a user clicks or purchases, such as "**UIC**" (User-Item-Concept), and "**UITC**"("**T**" for "Ca**T**egory"). Preference paths are triggered by long-term preferred categories or brands, such as "**UBC**"("**B**" for "**B**rand").

Within each meta-path, there are multiple specific paths called *path instance*s. For each meta-path, we sample a fixed number of path instances with highest priority scores. Calculation of the priority score for each edge in a path instance is based on heuristics. In the concept net, one item may belong to several concepts, while each concept also contains many items. So we mainly adopt *tf-idf* score to measure the importance of each "item-concept" edge and other types of edge. The score of the whole path instance is then calculated as the product of all the edge scores. Then we use a Convolution Neural Network (CNN) to encode each sampled instance pi and followed by a max-pooling operation to get the embedding of that meta-path (take "**UITC**" as an example):

$$pi_{\text{UITC}} = \textbf{CNN}([u_b, i, cate, c_i]) \quad (7)$$

$$p_{\text{UITC}} = \textbf{MaxPooling}(\{pi_{\text{UITC}}\}) \quad (8)$$

where i is the item embedding which only uses item description, and $cate$ is id lookup embedding of category. As for head and tail node, u_b is the behavior embedding and c_i is the lookup embedding of concept id. Comparing to RNN, CNN is much faster dealing with
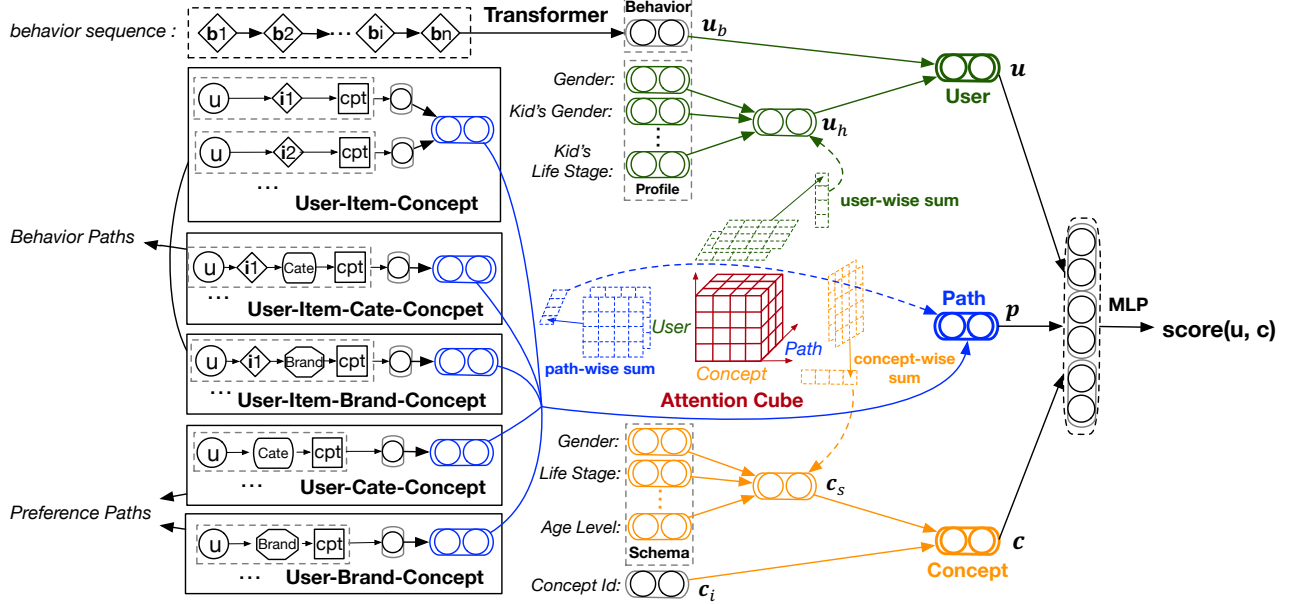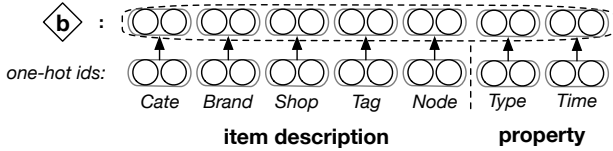
Figure 3: Overview of proposed model.

Figure 4: Encoding of user behavior

large amount of data and able to extract sequence dependency when sequence length is relatively short. Then the representation of meta-path context is calculated as:

$$\boldsymbol{p} = f_p(p) = f_p(\boldsymbol{p}_{\text{UIC}}, \boldsymbol{p}_{\text{UITC}}, \cdots) \tag{9}$$

4.4 The Whole Model

After getting the embedding for the user, the candidate concept and the paths connecting them, we concatenate the three embeddings and feed it into a MLP and the final output indicates the probability user u will need concept c:

$$\hat{y}_{uc} = \textbf{MLP}([\boldsymbol{u}; \boldsymbol{p}; \boldsymbol{c}]), \tag{10}$$

where the MLP module consists of two hidden layers with ReLU activation function and an output layer with sigmoid function.

We interpret user needs inference as a binary classification problem, where an observed user-concept interaction is assigned with a target value 1, otherwise 0. We use point-wise learning with the negative log-likelihood objective function to learn the parameters of our model:

$$\mathscr{L} = -\sum_{(u,c)\in D^+} \log \hat{y}_{uc} + \sum_{(u,c)\in D^-} \log(1 - \hat{y}_{uc}) \tag{11}$$

where D^+ and D^- are the positive and negative user-concept interaction pairs.

4.5 Attention Mechanism

If we define f_u, f_c and f_p as average pooling functions, each element contributes equally all the time. It is obviously suboptimal since different meta-paths are likely to effect users' decision making differently. Even for the same user, the preference on the same path may change targeting different concepts. Similarly, different aspects of user profile and concept schema can contribute to the final decision differently as well. For example, a pregnant female user is more likely to be in need of a concept like "Snacks for Pregnancy" rather than "Primary School". Some concepts are originally designed for a target group of users, so that different aspects should be weighted differently when encoding the user or the concept in each interaction. It is vital to investigate the mutual influence among user profile aspects, concept schema aspects and different mate-paths, since the input of our problem are much more informative comparing to previous works [? ? ?].

Attention mechanism has been been widely used to handle weighted sum of embeddings in recent years [? ?]. We proposed a novel attention module called "**Attention Cube**" to model the mutual influence of a three way interaction simultaneously in our problem. Attention cube is a three-dimensional tensor with x, y, z axis corresponding to $user$, $path$ and $concept$. We extend Luong's attention equation [?] to three-dimension and define the values of attention cube $\boldsymbol{Att}$ as below:

$$att_{i,j,k} = u_{h_i}{}^T \boldsymbol{W_1} p_j + p_j{}^T \boldsymbol{W_2} c_{s_k} + u_{h_i}{}^T \boldsymbol{W_3} c_{s_k} \tag{12}$$

where u_{h_i} is i^{th} embedding of user profile embedding list, c_{s_k} is k^{th} embedding of concept schema embedding list, and p_j is j^{th} embedding of meta-path embedding list. $\boldsymbol{W_1}$, $\boldsymbol{W_2}$, $\boldsymbol{W_3}$ are parameter matrices.

Then the weights of user profile aspects, concept schema aspects and different meta-paths are obtained by first calculating axis-wise

sum and then normalization:

$$\boldsymbol{\alpha_{u\,i}} = \frac{\exp(\sum_j \sum_k att_{i,j,k})}{\sum_i exp(\sum_j \sum_k att_{i,j,k})} \qquad (13)$$

We can get $\boldsymbol{\alpha_{p\,j}}$ and $\boldsymbol{\alpha_{c\,k}}$ in a similar way. Finally, the mapping functions f_u (similar for f_c and f_p) are defined to get $\boldsymbol{u_h}$ (similar for $\boldsymbol{c_s}$ and $\boldsymbol{p}$) as below:

$$\boldsymbol{u_h} = f_u(u_h) = \boldsymbol{\alpha_{u\,1}}\boldsymbol{h}_{\text{gender}} + \boldsymbol{\alpha_{u\,2}}\boldsymbol{h}_{\text{age}} + \cdots \qquad (14)$$

Since the attention weights $\boldsymbol{\alpha_{u\,i}}$, $\boldsymbol{\alpha_{p\,j}}$ and $\boldsymbol{\alpha_{c\,k}}$ are generated for each user-concept interaction separately, they are able to capture the complex mutual influence among the three components and result in better representations.

5 OFFLINE EVALUATION

In this section, we first introduce the dataset and experiment setup, including evaluation metrics and baselines. Then we present the offline results and give some discussions. Finally, we perform ablation tests to complete our experiments.

5.1 Datasets

Inferring e-commerce concepts a user potentially needs is a relatively new problem, there is no such public datasets for experiments. To create large amounts of gold standard data to train our model, we collect daily log of our online system, where concepts are already integrated in the recommender system. In a module called "Guess What You Like" at the front page of Taobao app, concepts are displayed as cards to users among the recommended items. There will be one concept card every ten items on average. In the snapshot shown in Figure 2(a), concept "Tools for Baking" is displayed as a card, with the picture of a representative item. Once users click on this card, it jumps to a page full of related items such as egg scrambler and strainer. In order to alleviate the potential influence of the item picture on users' decision making, we collect positive samples from those user-concept clicks only if that user continues to click at least two related items after entering the concept card. For the same reason, negative samples come from at least two exposes of the same concept (but different item pictures) without any clicks. We collect samples for continuous four days during January 11 to January 14, 2019, and use the data of first three days for training and validation. We randomly select 10% samples of the last day for testing. The ratio of negative and positive is around 37 : 1. For user-item interaction data, we collect 30-days transaction records on Taobao platform for each user in our data. Detailed statistics of our dataset is illustrated in Table 2.

	Training	Validation	Testing
# of samples	32,496,827	328,251	1,237,506
# of users	16,120,600	323,544	1,121,475
# of concepts	4,760	2,935	3,176
# of items	438M	76M	141M
# of categories	15,257	11,799	14,590
# of brands	1,434,659	428,036	1,088,480

Table 2: Statistics of Taobao's dataset.

Based on years of e-commerce experience, we mainly select five meta-paths (Figure 3) in our experiments: "**UIC**", "**UITC**" and "**UIBC**" for behavior paths; "**UTC**" and "**UBC**" for preference paths. Longer paths are not selected since they are likely to bring noises.

5.2 Experiment Setup

Evaluation Metrics

We perform evaluation of different models in two experiment scenarios. 1) In click-through-rate (CTR) prediction, we apply the trained model to each sample of test set and calculate *AUC* based on the output score to evaluate the overall performance; 2) In top-N recommendation scenario, we use the trained model to select N concepts with highest predicted scores for each user in the test set. we evaluate the results by Hit Ratio (*HR@N*), and Normalized Discounted Cumulative Gain (*NDCG@N*), which are widely used in recommendation tasks having very few ground-truth results [? ?]. In order to make sense under the second scenario, we augment the test set mentioned above by removing samples where the user does not have any positive clicks, and report averaged *HR* and *NDCG* across users.

Baselines

We compare with the following baselines:

- **BPR** [?] is the Bayesian Personalized Ranking model that minimize the pairwise ranking loss for implicit feedback.
- **Wide&Deep** [?] is the widely used recommendation framework, which jointly trains wide linear models and deep neural networks. We use embeddings of users, concepts and other entities to feed Wide&Deep.
- **MCRec+** is based on MCRec[?], which is a state-of-the-art HIN based model for recommendation. It treats the KG as HIN and extracts meta-path based features for modeling user-target interaction. We feed e-commerce concept net as the HIN for MCRec. For fair comparison, extra information appeared in our problem such as sequential user behaviors, user profile and concept schema, are also fed into MCRec in a compatible way.
- **KPRN+** is based on KPRN [?], another state-of-the-art knowledge-aware recommendation model, which aims to reason over KG by composing both entities and relations. Similar to MCRec+, we feed extra information to KPRN to get KPRN+.

Implementation Details

We implement our model using the python library of TensorFlow [7]. We set the length of user behavior sequence to 15, and sampled path instance within each meta-path to at most 50. The dimension of entity embeddings (item, concept, category, etc.) is set to 20, and the dimension of output layer is set to 32. The hidden state size of GRU is set to 40. All parameters are randomly initialized with Gaussian distribution. We perform a mini-batch log-likelihood loss training with a batch size of 512 for 5 training epochs. We use Adam optimizer [?], and the learning rate is initialized to 0.001. For all the comparison models, we refer to their original papers and tune the parameters using the validation set as well. With the help of a powerful distributed TensorFlow machine learning system in

---

[7]www.tensorflow.org

Taobao, we use 4 parameter servers and 20 workers, and the whole training process can be finished in 4 hours.

5.3 Results

We report the experimental results in Table 3 and Figure 5. Our model outperforms all the baselines, improving the result by up to 2.4% in *AUC*. Improvements in *HR* and *NDCG* also reveal the superiority of our model. BPR and Wide&Deep perform comparably poorly than other baselines, since they do not incorporate extra knowledge from e-commerce concept net into the model, failing to leverage rich features from paths between users and concepts. For knowledge-aware baselines, the main difference is the encoding of paths and attention mechanism. MCRec+ performs best among all baselines, since it also try to characterize a three-way interactions among user, paths and the concept. Our model substantially outperforms MCRec+ to achieve the best performance, which indicates the importance of modeling mutual attentive influence of three components simultaneously. KPRN+ performs worse than MCRec+, since the relation name matters in their problem is relatively trivial in our concept net. The last two lines of Table 3 further demonstrate the effectiveness of our proposed attention module. By comparing to a degenerated version of our model, which replaces attention cube with average pooling in each component, our full model achieves better performance.

Model	AUC
BPR	0.6005
Wide&Deep	0.6137
MCRec+	0.6447
KPRN+	0.6417
Ours (- att. cube)	0.6403
Ours (full)	**0.6612**

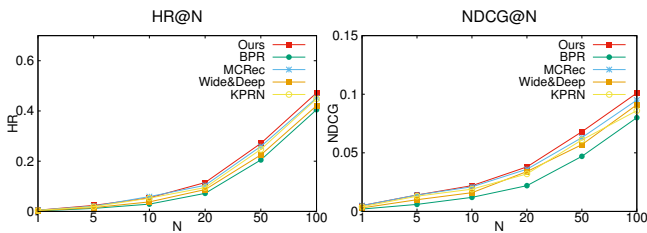**Table 3: AUC in CTR prediction on Taobao's dataset.**

Figure 5: HR and NDCG in Top-N recommendation.

5.4 Ablation Study

In this subsection, we explore the contribution of various components of our model. We report AUC on evaluation set to compare different variations in Table 4.

Behavior Paths vs Preference Paths

We first evaluate how different types of meta-path between users and concepts effect final performance. If we remove all paths, AUC drops by 6.8%, revealing the huge benefits brought by the concept

net. Between behavior paths and preference paths, we can observe that AUC drops more severely when removing the former ones, which indicates that behavior paths are more important than preference paths in our model. It appears that recent clicks or purchases of items play a larger role in reflecting user needs than long-term preferences, which may inflect that user needs are changeable and unstable, and they can be easily influenced.

Variation	AUC	Decrease (%)
- behavior paths	0.6826	4.03
- preference paths	0.6934	2.41
- all paths	0.6694	6.08
- user behavior sequence	0.7010	1.30
- user profile	0.6986	1.65
- concept schema	0.7031	1.00
Full	**0.7101**	0.0

Table 4: Ablation tests on validation set.

Behavior Sequence vs Side Information

Now we investigate the influence of user behavior sequence and side information in our problem, where side information further includes user profile and concept schema. Ablation towards these three components shows that they all contributes to the final inference result, while user profile information matters most (1.65% decrease in AUC). It is observed that user profile seems more important than user behavior sequence. The possible reason is that the attention cube degenerates to a matrix, if we remove user profile from the model. This may lead to a decrease in final performance.

6 ONLINE APPLICATION

The above offline experimental results have shown superiority of our proposed model for accurately inferring user needs. Now, we deploy our model online and integrate it into a recommender system in Taobao with a standard A/B testing configuration to answer the following three questions:

(1) Does our inference model still perform the best at online setting regarding both accuracy and novelty?
(2) Does user needs inference actually improve user satisfaction?
(3) Comparing to traditional item recommendation, does user-needs driven recommendation with concept cards bring extra value to e-commerce platforms?

6.1 Experiment Setup

The experiments are conducted in the online module introduced in Figure 2 (a). We integrate the inferred user needs (a.k.a. concepts) for each online user to our item recommender system, making recommendations of concept cards (one concept plus one representative item). Two online metrics are used to measure the performance: click-through-rate (**CTR**) and category-discovery (**Discovery**). Detailed definitions are as follows:

$$\mathbf{CTR} = \frac{\# \text{ concept card clicks}}{\# \text{ concept card exposes}}, \qquad (15)$$

$$\mathbf{Discovery} = \text{Avg}_u \left(\frac{\# \text{ new clk-cates in 15d}}{\# \text{ clk-cates}} \right) \qquad (16)$$

where Discovery is a measurement of how many distinct categories of representative items in concept cards a user clicked today are newly discovered (not clicked in the past 15 days in Taobao platform). It is a temporary[8] metric used in Taobao to evaluate the novelty of recommendation results.

We deploy the user needs inference module online and daily update our model. When recommending a concept card, online recommender system first output a list of items as usual, then we pair the items in the list with inferred top concepts, and filter out those items which are not related to any concepts. In the meantime, items within top concepts will complement the list. Followed by another ranking module, concept cards with highest scores will then be displayed to users.

6.2 Results

To answer the first question, we compare our model to the former strategy based on rules[9] and the strongest baseline MCRec+ in offline setting. Online results of A/B testing show that our model achieves highest CTR, which demonstrates that it can infer user needs more accurately. On the other hand, largest improvement on Discovery shows our model is able to bring more novelty.

Strategy	CTR	Discovery
Rule-based	-	-
MCRec+	+5.1%	+3.4%
Ours	**+6.0%**	**+5.6%**

Table 5: Improvements on CTR and Discovery.

To answer the second question, we conduct a real in-app user survey on Taobao since standard metrics like CTR and Discovery may not directly represent user satisfaction. Due to limited resource and time, we can only finish three rounds of survey. In each round, we randomly select 50,000 users and send them top 3 concepts inferred by model or by rule-based strategy. Each selected user is asked to answer a simple question: "Are you satisfied with [X] as a recommended shopping need for you?", where [X] is replaced by one inferred concept and the answer is YES or NO. Around 9k users out of 50k actually answered at least one question in each round of survey. The satisfaction rate is then calculated as the percentage of questions whose answer is YES in all answered questions, which is shown in Table 6. User satisfaction rate is improved by **20.6%** if we use the proposed inference model (14.7% for MCRec+), which demonstrates such user needs inference actually make users more satisfied. As we can notice, the absolute number of satisfaction rate is only 41%, which is clearly not a large number. In fact, it is hard to know the true upper bound of user satisfaction rate, meaning there is ample room for us to continuously explore user needs understanding.

To answer the last question, we compare recommending a concept card with recommending an item (traditional item recommendation) at the same position on "Guess What You Like". Online

Strategy	Satisfaction Rate	Improvement
Rule-based	34%	-
MCRec+	39%	+14.7%
Ours	**41%**	**+20.6%**

Table 6: Satisfaction rate from real user survey.

evaluation shows significant improvements [10] of recommending concept cards: **5.3%** in CTR and **9.6%** in Discovery. If we further consider the purchases of related items in the page guided by concept cards, total sales volume (GMV) is improved by **84.0%**, which demonstrates the great value and potential of such user-needs driven recommendation.

6.3 Case Study

A major contribution of our model is that we propose a attention cube to model three-way interactions simultaneously, aiming to distinguish different importances of different factors in an e-commerce interaction, which may inspire us to better understand user needs. Therefore, we analyze the attention values from several perspectives during online inference. All the following analysis is based on one day's user log.

During inference, if the gender of a user matches to the gender constraint of a concept, the attention weights of "gender" in both user profile and concept schema become nearly twice larger than not matching. This indicates our model can explicitly learn rules such as a young female user is more likely to need a concept "Party for girls" rather than "Party for boys".
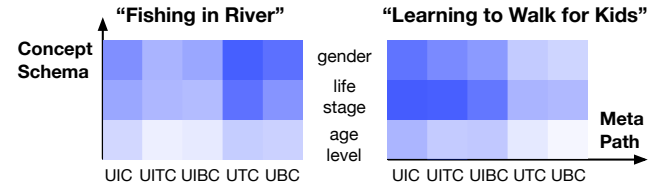

Figure 6: Visualization of attention weights for an anonymous user. Darker colors indicate higher weights.

To see if the same user has different preferences on meta-paths regarding different concepts, we randomly pick a user as illustrative example shown in Figure 6. The anonymous user has two positive interactions of concept cards: "Learning to Walk for Kids" and "Fishing in River". After digging into transaction data, we find that this user recently clicks a lot of kids related items, resulting high importance of behavior paths shown in his attention distribution when facing concept "Learning to Walk for Kids". On the contrary, he has few behaviors related to fishing. Accordingly, the attention weights of preference paths are much higher than average when facing concept "Fishing in River" since his long-term category preference is "fishing equipments".

---

[8]Designing a proper metric to evaluate novelty in industrial recommendation is a hard and unsolved problem.
[9]Concepts are ranked by the counting number of their related items which are behaved by the user.

[10]This comparison is not entirely fair due to the different display form between a concept card and an item. But the large improvements still indicate the value of user-needs driven recommendation. Integrating user needs understanding to general item recommendation is included in our future work.

7 CONCLUSION

In this paper, we point out that one of the biggest challenges in current e-commerce solutions is that they are not directly driven by user needs, which, however, are precisely the ultimate goal of e-commerce platform try to satisfy. To tackle it, we introduce a specially designed e-commerce knowledge graph practiced in Taobao, trying to conceptualize user needs as various shopping scenarios, also known as e-commerce concepts. We further proposed a deep attentive inference model to intuitively infer those concepts accurately. On our real-world e-commerce dataset, the proposed model achieved state-of-the-art performance against several strong baselines. After applying to online recommender system, great gain regarding both accuracy and novelty are achieved. A real user survey is conducted to demonstrate such user needs inference actually improves user satisfaction. More importantly, we believe that the idea of conceptualizing and inferring user needs can be applied to more e-commerce applications. In the future, we will continuously explore various possibilities of "user-needs driven" e-commerce.

8 ACKNOWLEDGEMENT

REFERENCES

[] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation. *arXiv preprint arXiv:1805.03352* (2018).

[] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. *Dbpedia: A nucleus for a web of open data.* Springer.

[] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*. 1247–1250.

[] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. 2787–2795.

[] Rose Catherine, Kathryn Mazaitis, Maxine Eskenazi, and William Cohen. 2017. Explainable entity-based recommendations with knowledge graphs. *arXiv preprint arXiv:1707.05254* (2017).

[] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *WSDM*. ACM, 108–116.

[] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.

[] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[] Sergio Cleger-Tamayo, Juan M Fernandez-Luna, and Juan F Huete. 2012. Explaining neighborhood-based recommendations. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1063–1064.

[] Felipe Costa, Sixun Ouyang, Peter Dolog, and Aonghus Lawlor. 2018. Automatic Generation of Natural Language Explanations. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*. ACM, 57.

[] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep Session Interest Network for Click-Through Rate Prediction. *arXiv preprint arXiv:1905.06482* (2019).

[] Yu Gong, Xusheng Luo, Yu Zhu, Wenwu Ou, Zhao Li, Muhua Zhu, Kenny Q Zhu, and Xi Chen Lu Duan. 2019. Deep Cascade Multi-task Learning for Slot Filling in Online Shopping Assistant. (2019).

[] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S Yu. 2018. Leveraging metapath based context for top-n recommendation with a neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1531–1540.

[] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 505–514.

[] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2333–2338.

[] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 345–354.

[] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 1 (2003), 76–80.

[] Kangqi Luo, Fengli Lin, Xusheng Luo, and Kenny Zhu. 2018. Knowledge Base Question Answering via Encoding of Complex Query Graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2185–2194.

[] Xusheng Luo, Kangqi Luo, Xianyang Chen, and Kenny Q Zhu. 2018. Cross-lingual entity linking for web tables. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.

[] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 373–374.

[] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 297–305.

[] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 417–426.

[] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. *arXiv preprint arXiv:1801.08284* (2018).

[] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2018. Explainable Reasoning over Knowledge Graphs for Recommendation. *arXiv preprint arXiv:1811.04540* (2018).

[] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics* 4 (2016), 259–272.

[] Markus Zanker and Daniel Ninaus. 2010. Knowledgeable explanations for recommender systems. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, Vol. 1. IEEE, 657–660.

[] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 353–362.

[] Yongfeng Zhang and Xu Chen. 2018. Explainable Recommendation: A Survey and New Perspectives. *arXiv preprint arXiv:1804.11192* (2018).

[] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Metagraph based recommendation fusion over heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 635–644.

[] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1059–1068.