

An End-to-end Model of Predicting Diverse Ranking On Heterogeneous Feeds

Zizhe Gao
Alibaba Group
zizhe.gzz@alibaba-inc.com

Zheng Gao
Indiana University Bloomington
gao27@indiana.edu

Heng Huang
Alibaba Group
gongchong.hh@taobao.com

Zhuoren Jiang
Sun Yat-sen University
jiangzhr3@mail.sysu.edu.cn

Yuliang Yan
Alibaba Group
yuliang.yyl@alibaba-inc.com

ABSTRACT

As an external assistance for online shopping, multimedia content (feed) plays an important role in e-Commerce field. Feeds in formats of post, item list and video bring in richer auxiliary information and more authentic assessments of commodities (items). In Alibaba, the largest Chinese online retailer, besides traditional item search engine (ISE), a content search engine (CSE) is utilized for feeds recommendation as well. However, the diversity of feed types raises a challenge for the CSE to rank heterogeneous feeds. In this paper, a two-step end-to-end model including Heterogeneous Type Sorting and Homogeneous Feed Ranking is proposed to address this problem. In the first step, an independent Multi-Armed bandit (iMAB) model is proposed first, and an improved personalized Markov Deep Neural Network (pMDNN) model is developed later on. In the second step, an existing Deep Structured Semantic Model (DSSM) is utilized for homogeneous feed ranking. A/B test on Alibaba product environment shows that, by considering user preference and feed type dependency, pMDNN model significantly outperforms than iMAB model to solve heterogeneous feed ranking problem.

KEYWORDS

e-Commerce Search, Heterogeneous Feed Ranking, Multi-Armed Bandit, Deep Neural Network

ACM Reference format:

Zizhe Gao, Zheng Gao, Heng Huang, Zhuoren Jiang, and Yuliang Yan. 2018. An End-to-end Model of Predicting Diverse Ranking On Heterogeneous Feeds. In *Proceedings of ACM SIGIR Workshop on eCommerce, Ann Arbor, Michigan, USA, July 2018 (SIGIR 2018 eCom)*, 6 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Search Engine plays a vital role in e-Commerce industry, which can navigate users' potential purchasing behavior. Therefore, designing an elaborate ranking algorithm is the key challenge for every search engine. Traditionally in e-Commerce, search engines are all item search engine (ISE), meaning the returned results for

a given query is a ranking list of items. However, with the boom of diverse types multimedia, it is not enough to recommend items solely on item direct information such as reviews. As the emergence of self-media on internet, more and more users are willing to share their expressions publicly. It is a way to advertise themselves while offer shopping tips to other users meantime. Those types of shared multimedia are various such as post (article), list and video which provide more detailed information about the items, like item introduction, maintenance guideline, usage demonstration and so on. In this paper, a piece of multimedia content is named as a piece of "feed". As the external information offered in feeds can help users make better purchasing choices, content search engine (CSE) is developed in need to recommend high quality feeds back to users. In recent years, jointly managing item search engine (ISE) and content search engine (CSE) together in e-Commerce has been proved to be able to better attract user attentions and increase the click-through rate on item pages already. One example can be found in Figure 1. A user can issue queries in both ISE and CSE to retrieve an item ranking list as well as a feed ranking list. As items and feeds can be mapped based on feed contents (i.e. feed 1 and feed k both are related to item 1 based on their content. And feed k also talks about item k as well), the user can take advises from feeds to make purchase decisions. Hence, improving the quality of heterogeneous feed ranking in CSE has a great meaning for e-Commerce.

Currently, there are two challenges remained for heterogeneous feed ranking. First, cross-domain knowledge between ISE and CSE needs to be explored and uncovered to support better CSE ranking performance. Second, as the types of content are heterogeneous (including post, list and video) in CSE, novel algorithms needs to be designed to deal with heterogeneous type sorting problem.

Although heterogeneous feed ranking is a new topic, some of previous studies have offered possible solutions to deal with similar problems. To solve cross-domain challenge, HEGS model proposed by [12] uses a two step approach by sampling data from all domains based on a designed clustering algorithm constrained by KL divergence first, and taking a regression model to learn a optimized label for input data afterwards. [14] introduces a novel cross-domain ranking by transferring the preference order between domains. [6] uses matrix factorization methods on heterogeneous dataset and utilize user reviews to generate item ranking list back to users. [11] uses cross-domain collaborative filtering techniques to figure out the most important heterogeneous data resource from all domains.

Previously, Multi-armed bandit (MAB) framework is widely used to deal with heterogeneous ranking problem [1]. [7] develops a fast

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGIR 2018 eCom, July 2018, Ann Arbor, Michigan, USA
© 2018 Copyright held by the owner/author(s).
ACM ISBN 123-4567-24-567/08/06...\$15.00
https://doi.org/10.475/123_4

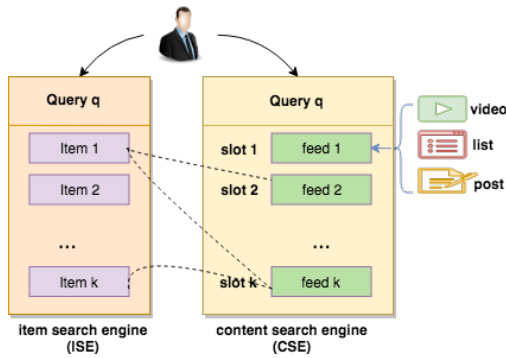


Figure 1: Relationship between Item Search Engine and Content Search Engine

MAB algorithm by considering past user behaviors. [10] introduces two online learning algorithms based on user click records to rank diverse documents on the web. [8] applies MAB model on user side information and develop an epoch-greedy model to recommend the most relevant ad on a web page. While [9] formalizes two MAB models on e-Commerce targeting on e greedy solution and independent solution. In recent years, deep learning techniques becomes popular in recommendation and information retrieval domain. [3] proposes a content-based recommendation system and uses a rich feature set to represent users, according to their web browsing history and search queries so as to map users and items into a latent space via a deep learning approach. [15] presents a deep model to learn item properties and user behaviors jointly from review text. And [13] develops a RNN model to predict user shopping behavior leveraging features from clickstream data.

Alibaba, the largest Chinese online retailer, offers both ISE and CSE to users on mobile application. Derived from all theoretical and practical foundations mentioned above, in this paper, we aim to solve heterogeneous feed ranking problem in Alibaba CSE and recommend users adequate feeds to benefit their purchasing choices. We divide the heterogeneous feed ranking problem in CSE into two phases: Heterogeneous Type Sorting and Homogeneous Feed Ranking. In this paper, we mainly focus on solving the first heterogeneous type sorting problem, and formulate the second step with an existing well known model called Deep Structured Semantic Model (DSSM) [5]. In this paper, the i_{th} “slot” means the i_{th} position in the ranking list to hold a feed. Its type is learned and determined in the first step. In the second step, proper type of feeds will be selected and filled into relevant slots. The two steps are learned and trained together via a proposed end-to-end model. To solve heterogeneous type sorting task, two novel models are proposed based on different assumptions about user preference and slot dependency. An independent Multi-Armed Bandit (iMAB) model is designed to rank feeds assuming slots independent with each other and generates a global model for feed ranking. While a personalized Markov Deep Neural Network (pMDNN) model is designed to jointly select feed types for all slots and offers a personalized feed type result for each user. The iMAB model solely relies on CSE record and generates the feed type for each slot independently in a statistical estimation method. While pMDNN model integrates both ISE and

CSE historical record to build up user profile and uses a three-layer neural network to generate feed types for each slot at the same time. The generated feed types of both models will be utilized in a Deep Structured Semantic Model (DSSM) [5] to predict relevant feeds for each slot. Results based on A/B test on Alibaba product environment shows that by considering user preference and feed type dependency, pMDNN model outperforms than iMAB model for heterogeneous feed ranking in the CSE.

The contribution of this paper is fourfold. First, we integrate cross domain knowledge from both ISE and CSE to generate feed ranking list in CSE. Second, an end-to-end model is designed to solve the heterogeneous feed ranking problem which avoids involving extra parameters during the model training process. Third, two novel models are designed and compared for solving heterogeneous feed type selection problem. User preference and feed type dependency are both considered in the second model. Fourth, A/B test of the two models are conducted in Alibaba product environment to generate a convincing comparison between the two models.

The paper is organized in following sections: Section 2 introduces basic concepts of our paper including Alibaba business background and the data information used in our approach; Section 3 mainly introduces how our model is designed to deal with heterogeneous feed ranking problem; Section 4 shows the details of the experiment result and Section 5 draws a conclusion and points out the future work.

2 BASIC CONCEPTS

2.1 Business Background

In our line of business, Alibaba owns both CSE (Content Search Engine) and ISE (Item Search Engine) which are highly interacted with each other to create an online shopping environment for users. All items in ISE are associated with a bunch of feeds in CSE. And users can travel and search between two search engines freely without boundary.

The mixture usage of Alibaba ISE and CSE can benefit user online shopping. Generally, users logging in Alibaba and interacting with search engines are always with intentions. However, if they only search items in ISE, they might get lost when facing numerous items. For example, in Alibaba official website, hot categories like clothing may contain thousands of items. Each item is labelled under dozens of keywords such as fashionable style, slim cut and Korean-like style, etc. It challenges users to distinguish appropriate items from a huge set of item candidates without any instruction.

Consequently, to help users avoid hesitation and offer them reliable shopping suggestions, CSE is came into being on behalf of a shopping guide for users. Given queries from users, CSE organizes a proper feed ranking list as a returned result instead of item ranking list. And feeds are represented in format of post (article), list (item list) and video. They are produced by “Daren”s who are experts of a certain e-Commerce field (clothing, travelling, or cosmetics, etc.). In their feeds, “Daren”s introduce pros and cons of certain items and raise personal advises to specific subjects based on their domain knowledge. A post feed is an article to describe the properties of particular items; A list feed is a set of recommended items offered for a specific field; A video feed is a short video shot to demonstrate

Slot	post		list		video	
	pv	ipv	pv	ipv	pv	ipv
slot 0	1731732	230592	73540	5460	0	0
slot 1	1704866	177854	49348	4288	0	0
slot 2	785993	63730	696546	51032	234	0
slot 3	949621	57600	693031	36384	0	0
slot 4	755625	36886	825816	63276	0	0

Table 1: Feed historical data in CSE top 5 slots

the suggested items. By taking suggestions from “Daren”, users can make better choices for purchasing items online.

Daily data in product environment empirically shows that user travel rate between the two search engines are frequent. Before users jump into CSE, they always have searching records in ISE already. It implies that users actually are willing to pursue advises from “Daren”s. And offering better CSE searching result can help users to target suitable items more easily and make purchases afterwards, which is the primary goal of e-Commerce. However, it is still confronted with challenges. First, as feed types are heterogeneous, the fitness of different types of feeds are incomparable towards a given query. For example, whether a list feed is better than a post feed heavily depends on user preference. Second, majority of users entered Alibaba CSE carry with user behaviors in ISE. And how to deal with this cross-domain information and build up user profiles to form a personalized feed ranking in CSE also needs to be explored.

2.2 Data Preparation

In our approach, we aim to return a heterogeneous feed ranking list $R_l(feed)|u, q$ given a query q issued by a user u . Each of the Top K ranked feed will be located and displayed in a “slot” in CSE from top to bottom. To learn the independent Multi-armed Bandit (iMAB) model and personalized Markov Deep Neural Network (pMDNN) model which will be introduced in the next section, both slot related statistical data (global information) and user click streaming data (personalized information) need to be obtained.

2.2.1 Slot Related Statistical Data. One assumption of user preference is that feed type in each slot is independent with each other. And for each slot, the probability of three candidate feed types (post, list, video) follow their own *Beta Distributions*. Therefore, to estimate the prior distribution $p(\theta|\alpha, \beta, T, s)$ of a feed type θ given all candidate types T in a slot s , it is necessary to know all related statistical data of each slot so that α and β can be estimated. The slot related statistical data contains two parts: online real-time data and offline historical data. Online real-time data refers to the streaming data about the number of clicks and displays for a particular slot type produced by users each day. And offline historical data refers to the past N days total number of page view (pv) and item page view (ipv). The online daily data is streaming data that can only observed in real time. While offline historical data can be tracked and obtained from repository. We calculate and illustrate the statistics of top 5 slots in Table 1.

As we can see from Table 1, the total number of pv and ipv in each relevant slots varies, and video feeds hardly appear in the

user_feature	query_feature	feed type
0.0073, 0.4694, ..., 0.0809	-0.0135, -0.0278, ..., 0.00613	100

Table 2: A example of user personalized data for each slot

top 5 slots in CSE. It indicates that, in a global view, users prefer different feed types in different slots.

2.2.2 User Click Streaming Data. User behavior sequence data from both ISE and CSE comprehensively is also useful to train a personalized feed ranking result. To build up user profile, we set a window size w and only consider the latest w behaviours users take in ISE. The behaviors can be represented as two types of triplet as $\langle user, issue, query \rangle$ and $\langle user, click, item \rangle$. And the number of times users click on items shows the relationship strength between users and items, while the number of times users issue the same query shows the relationship strength between users and queries. Based on that, a given dimensional embedding can be learned for each user/query under the same latent space later on. Moreover, feed type in each slot is encoded via one-hot encoder. In the end, all users, queries and feed types can be represented as vectors. An example is shown in Table 2. The first two columns refer to the learned representation for each user f_u and an issued query f_q . The third column refers to the one-hot representation feed type f_t in each slot.

3 METHODS

We are willing to observe better heterogeneous feeds ranking for users’ preference. The whole process contains Heterogeneous Type Sorting step and Homogeneous Feed Ranking step. For the first step, an independent Multi-Armed Bandit (iMAB) model is designed for slot independent scenario and an improved personalized Markov Deep Neural Network (pMDNN) model is designed for slot dependent scenario. Section 3.1 and 3.2 introduce the two models individually. And for the second step, a DSSM model is utilized to assign proper type of feeds in each slot. The details is introduced in Section 3.3. pMDNN model can be trained together with DSSM to formalize an end-to-end model.

3.1 independent Multi-Armed Bandit

In iMAB model, the evaluation metric of heterogeneous feed ranking is the ratio θ between ipv and pv . Higher θ means when a user browse a feed in CSE, the user is more likely to click the feed. So θ can be used to evaluate the fitness of the heterogeneous feed ranking towards users’ real need. Hence, for each independent slot, we estimate a prior ratio θ distribution of each feed type, and are willing to choose the feed type that is able to generate the highest θ value.

Theoretically, as Beta distribution can naturally represent any kind of distributions controlled by two parameters α and β , it makes sense to assume the ratio θ of each type has a prior distribution following $\theta_i \sim B(\alpha_i^0, \beta_i^0)$ where $i \in \mathcal{U} = \{post, list, video\}$. α_i^0 is the type i historical ipv number and β_i^0 is the difference between type i historical pv number and ipv number. It is because the expectation

Algorithm 1 independent Multi-Armed Bandit

```

1: Slots Type Selection:
2: for all slot in slots do
3:   for all type in types do
4:      $\alpha_0 \leftarrow$  historical pv number
5:      $\beta_0 \leftarrow$  historical pv - ipv number
6:      $D^{pv} \leftarrow$  online streaming pv number
7:      $D^{ipv} \leftarrow$  online streaming ipv number
8:      $\lambda \leftarrow$  time impact factor
9:     draw  $\theta_{type} \sim \mathcal{B}(\alpha_0 + \lambda D^{ipv}, \beta_0 + \lambda(D^{pv} - D^{ipv}))$ 
10:    store  $\theta_{type}$ 
11:  end for
12:   $p(type) = \frac{\exp(\theta_{type})}{\sum_{j \in types} \exp(\theta_j)}$ 
13:  draw type based on  $p(type)$ 
14:  set type in slot
15: end for

```

of $B(\alpha_i^0, \beta_i^0)$ is $\frac{\alpha_i^0}{\alpha_i^0 + \beta_i^0}$, which is the historical ratio between ipv and pv. Hence the posterior ratio distribution can be updated by online real-time stream data each day, and represented as $\theta_i | D_i \sim B(\alpha_i^0 + \lambda D_i^{ipv}, \beta_i^0 + \lambda(D_i^{pv} - D_i^{ipv}))$ where D_i refers to the coming data each day of feed type i and λ is a time impact factor as new data should have more influence to update ratio distribution.

In the end, we apply a two step sampling strategy to choose the type of each slot. First, for each feed type i , a value θ_i is randomly generated as the estimation of the ratio between pv and ipv followed by the probability distribution below.

$$p(\theta_i | D_i) = \frac{(\theta_i)^{\alpha_i - 1} (1 - \theta_i)^{\beta_i - 1}}{\mathcal{B}(\alpha_i, \beta_i)} \quad (1)$$

where $\mathcal{B}(\alpha_i, \beta_i)$ is a constant given α_i , and β_i . $\alpha_i = \alpha_0 + \lambda D_i^{ipv}$ and $\beta_i = \beta_0 + \lambda(D_i^{pv} - D_i^{ipv})$.

Second, a Softmax function is applied on all feed types to generate a normalized selection probability for each feed type.

$$p(i) = \frac{\exp(\theta_i)}{\sum_{j \in \mathcal{U}} \exp(\theta_j)} \quad (2)$$

where i refers to one of the three feed types and θ_i is the random value generated following posterior probability distribution $\mathcal{D}(\theta)$ showed in Formula 1.

In this way, feed types in all slots are selected independently. The pseudo code of the whole procedure is showed in Algorithm 1.

3.2 personalized Markov Deep Neural Network

Dependent heterogeneous feed type selection is determined by three factors: user, query and previous slot feed types in the same page. First, different users may express various preferences on items under same query. For example, when one user searches “dress”, she may be willing to see more posts about the description of the dress. While for other users, they might prefer lists because they want to see more item choices rather than single item introduction. Second, user preference of feed types on current slot may be potentially influenced by previous feed types, which can be regarded as a *Markov process*. For example, no user is willing to see a single

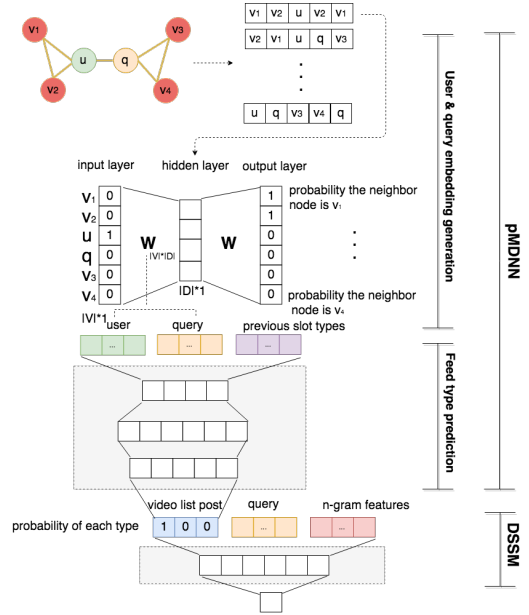


Figure 2: The end-to-end model with pMDNN learned from cross-domain knowledge and user preference

type of feeds in all slots, they more or less expect to see diverse types of feeds. And third, different queries should also result in different feed type allocation in all slots. To integrate user preference, query and previous recommended feed types together, we propose a personalized Markov Deep Neural Network (pMDNN) model to generate the recommended feed type $t_i | (user, query, t_1, \dots, t_{i-1})$ for the i_{th} slot. The whole model can be decomposed into two sub tasks including an user & query representation learning task and a personalized slot type prediction task, which is demonstrated in Figure 2.

3.2.1 Representation of User and Query. Based on statistics, more than 80 percent of users in CSE come from ISE. Hence, by using their user behavior sequence data, we can construct a graph to describe the relationship between users, queries and items. After that, node2vec [4] applies a skip gram model to learn embeddings of users and queries in the end. Detailed pipeline is shown in the upper part of Figure 2 and the objective function is listed below:

$$O(\vec{f}_v) = \log \sigma(\vec{f}_t \cdot \vec{f}_v) + k \mathbb{E}_{u \in P_{noise}} [-\log \sigma(\vec{f}_u \cdot \vec{f}_v)] \quad (3)$$

\vec{f}_v is the embedding of current node v . t is a positive neighbour node of node v and u is a negatively sampled node of v . It means that given a node v , we need to learn a node embedding representation that can maximize the probability to generate its positive neighbor node u and minimize the probability to generate its negative node sets P_{noise} .

The middle part of Figure 2 shows how to train node embedding representation. The input layer is one-hot encoding of node. The weight matrix W is the all nodes embedding, it can help to project the input one-hot encoding node into a $|D|$ dimension latent space.

And then maximize the probability to generate neighbour nodes of the node u .

In the end, all users and queries can have embedding representations with a given length dimension. And we use the user & query embeddings as the input for slot feed type prediction.

3.2.2 Type prediction. We are willing to predict feed types in each slot given users, queries and previous slot feed types information. Hence, the objective function of our goal is showed as:

$$\operatorname{argmax}_{\Phi} \prod_{i=1}^K p(\Phi(\mathcal{X}_i) = c | u_i, q_i, f_i) \quad (4)$$

where \mathcal{X}_i is input feature vectors for the i_{th} slot, which is related to users u_i , queries q_i and previous slot feed types f_i . Φ is the transformation function for input feature vectors to the output feed type. c is true feed type of current slot. Our goal is to maximize the joint probability of successfully predicting slot feed types.

To simplify our pMDNN model and accelerate the running speed, only one-order Markov process of slot feed type is applied in this model. It means that to predict the i_{th} slot feed type, only the $(i-1)_{th}$ slot feed type has latent impact on that. While it brings a problem to predict the first slot feed type for a user u . Because there will be no previous slot feed type information. To generate a pseudo information for the first slot, the favorite item i of user u is detected in ISE according to the number of viewed times and the length of stayed time. Then we map the item i in ISE to its related feed f in CSE and use the type of f as a substitution.

We build up the pMDNN model to recommend the feed type with given embedding of user and query as well as previous slot types. The input layer is the concatenation of user embedding (\mathcal{U}), query embedding (\mathcal{Q}) and previous slot types (\mathcal{T}). User and query embedding are learned via node2vec on constructed graphs. The whole input layer construction can be viewed as:

$$\mathcal{X} = \mathcal{U} \oplus \mathcal{Q} \oplus \mathcal{T} \quad (5)$$

After that, three fully connected hidden layers are attached to the input layer. Every layer utilizes linear classifiers and cross entropy as the loss function. Activation function in each hidden layer is set to *ReLU* and the output layer applies *Softmax* as the activation function. Throughout gradient descent and back propagation, we can train our model until convergence. The output layer is a vector which contains a probability distribution of three feed types on each specific slot after taking a Softmax activation function.

$$\begin{aligned} L_1 &= \operatorname{ReLU}(w_0 \cdot \mathcal{X}) \\ L_2 &= \operatorname{ReLU}(w_1 \cdot L_1) \\ L_3 &= \operatorname{ReLU}(w_2 \cdot L_2) \\ \mathcal{L} &= \operatorname{Softmax}(w_3 \cdot L_3) \end{aligned} \quad (6)$$

\mathcal{L} represents the true label of current slot feed type. the pMDNN model will be trained in offline phase, and we could manage the trained model to predict real-time user request. And L_1, L_2, L_3 refer to three hidden layers respectively. The first part of Figure 2 illustrate this workflow.

3.3 Homogeneous feed ranking

The next step is to rank homogeneous feeds and fill in related slots. For example, if $slot_i, slot_j, slot_k$ are chosen to have ‘‘post’’ feed type, we need to rank all post feeds and select the top 3 feeds with highest relevance score towards the issued query. As all types of feeds are all associated with textual information such as title, an existing Deep Structured Semantic Model (DSSM) [5] is applied to rank all post feeds to fill in the three slots.

In DSSM, instead of encoding each word with one-hot representation, a *Word Hashing* method is raised to leverage n-gram model to decompose each word. It leads to a dimension reduction of word representation.

Afterwards, a Deep Neural Network (DNN) model uses query and feeds as input layer, and train the model parameters by maximizing the likelihood of the clicked documents given the queries across the training set. Equivalently, the model needs to minimize the following loss function:

$$L(\Lambda) = -\log \prod_{Q, D^+} p(D^+ | Q) \quad (7)$$

where Λ denotes the parameter set of the neural networks. D^+ is the true labelled feed and Q is the user-issued query. The model is trained readily using gradient-based numerical optimization algorithms.

In the end, given a query, all candidate feeds can be ranked by the generative probability calculated from this model. It can be trained with pMDNN at mean time to formalize an end-to-end model, which is showed in Figure 2. While it still needs to be trained separately from iMAB model.

4 EXPERIMENT

We conduct our experiments on item search engine (ISE) and content search engine (CSE) of Alibaba product environment with data introduced in Section 2.2 which has been partitioned into 80% training and 20% testing randomly split. User behavior sequences collected from logs of Alibaba during $N = 90$ days are constructed to a behaviors graph which are in favor of representing users and queries to dimensional embeddings.

4.1 Model Setup

For iMAB model, in online part, we implement a real-time Flink[2] job which parses user behavior logs and extracts a series of status that represent whether user click or browse displayed feeds on different slots. Then the user behaviors are synchronized as online rewards to iMAB model. As user behavior logs of Alibaba are of huge amounts, to make sure Flink job are in low latency, we assign 256 workers to do parsing and joining then 64 workers to do aggregating. As we expected, online rewards are transferred to iMAB model under 3 seconds which makes it possible to select arm represents a probability distribution based on latest user behaviors. While in offline part, more than 100 million ipv and pv records are aggregated to estimate Beta Distribution. Based on empirical study, we set $\lambda = 10$ as a time impact factor in iMAB model.

Besides, pMDNN model needs a training phase in which loss function, optimizer and parameters are set as followed:

in (%)	bucket 1		bucket 2		bucket 3	
	iMAB	pMDNN	iMAB	pMDNN	iMAB	pMDNN
pv	-0.1515	1.1114	0.7024	2.9562	0.5250	0.4617
uv	-0.0381	0.4435	0.1670	1.9201	0.0733	-0.4121
pv click	2.1613	2.2994	1.6571	1.8403	-0.4227	0.0591
uv click	0.3986	3.9676	-0.0682	2.1698	-0.3437	1.8449
uv CTR	0.4369	3.5095	-0.2348	0.2418	-0.4167	2.2703

Table 3: Incremental ratio after two models are deployed

- User and Query representation: 128 dimensional graph embedding
- Feed type representation: One-hot vector
- Activation function: ReLu and Softmax
- Loss Function: Cross entropy
- Optimizer: Gradient descent optimizer
- Learning rate: 0.0001
- epochs: 100,000

Well trained pMDNN model exported in specific saved_model format will be serving in CSE, which receives real-time online requests that contain user, query and preceding feed type and then predict next slot type in order with converted embedding vectors. And the default settings in original DSSM is applied in our model.

4.2 A/B Test

We deploy our proposed models to three buckets. Each of them equally handle user requests via a hash partition function. We select 5 major indices to compare performance between iMAB and pMDNN. pv stands for the number of displayed items, while pv click is how many displayed items are clicked; Similarly, uv is the total number of distinct users entered CSE and uv click represents the number of users who clicked feeds; As to uv CTR, it is the ratio of users who clicked or not.

Table 3 shows experimental results, in which pMDNN generally outperforms iMAB in comparison to primitive offline ranking method. Especially uv click and uv ctr, they are essential to our scenario, because the increase of uv click shows that more users tend to CSE so that it facilitates their shopping experiences, in the meanwhile, the boost of uv ctr shows users entered CSE are really interested in model ranking results. As to pv click, it also shows that our proposed model works fine since more users are willing to click feeds after issued queries.

Based on pv click and uv CTR, we can conclude that pMDNN is superior to iMAB by applying cross-domain knowledge and optimizing ranking results in whole page. Besides, combining user preference information could increase the probability of user clicking as shown by uv click.

5 CONCLUSION

To facilitate user purchasing behavior, content search engine (CSE) emerges as the supplement of item search engine (ISE). Item introduction, Shopping guide and expert advises in post, list and video type could be offered in CSE which makes it critical to users' shopping choices especially confronting hundreds of item candidates.

Provide a diverse and personalized feed ranking result can benefit users on item selection.

In this paper, we presented an end-to-end model of predicting diverse ranking on heterogeneous feeds that is a two-step approach - Heterogeneous Type Sorting and Homogeneous Feed Ranking. In the first step, two models independent Multi-Armed bandit (iMAB) and personalized Markov Deep Neural Network (pMDNN) are proposed to tackle heterogeneous data sorting. Being an online learning algorithm, iMAB combines historical statistics and online rewards which could quickly converge in each slot but fail to optimize the whole page results. Consequently, we put forward pMDNN based on ISE to CSE cross-domain knowledge and formalize it as a one-order Markov process which not only provide user preferred feeds on specific slot but fix the problem of slot independence. An existing DSSM model leverages deep learning techniques to rank same-type feeds afterwards. Via A/B test on Alibaba product environment, result shows pMDNN outperforms than iMAB on most of well known metrics used in e-Commerce field. Future work will involve more cross-domain knowledge like purchasing intention to affect ranking result as well as more analysis on user sequential data in ISE.

REFERENCES

- [1] Róbert Busa-Fekete and Eyke Hüllermeier. A survey of preference-based online learning with bandit algorithms.
- [2] Paris Carbone, Asterios Katsifodimos, Stephan Ewen, Volker Markl, Seif Haridi, and Kostas Tzoumas. 2015. Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 36, 4 (2015).
- [3] Ali Mamdoh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 278–288.
- [4] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.
- [5] 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 Conference on information & knowledge management*. ACM, 2333–2338.
- [6] Mohsen Jamali and Laks Lakshmanan. 2013. HeteroMF: recommendation in heterogeneous information networks using context dependent factor models. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 643–654.
- [7] Pushmeet Kohli, Mahyar Salek, and Greg Stoddard. 2013. A Fast Bandit Algorithm for Recommendation to Users With Heterogenous Tastes.. In *AAAI*.
- [8] John Langford and Tong Zhang. 2008. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*. 817–824.
- [9] Jonathan Lou  dec, Max Chevalier, Josiane Mothe, Aur  lien Garivier, and S  bastien Gerchinovitz. 2015. A Multiple-Play Bandit Algorithm Applied to Recommender Systems.. In *FLAIRS Conference*. 67–72.
- [10] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. 2008. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*. ACM, 784–791.
- [11] Shaghayegh Sahebi and Peter Brusilovsky. 2015. It takes two to tango: An exploration of domain pairs for cross-domain collaborative filtering. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 131–138.
- [12] Xiaoxiao Shi, Qi Liu, Wei Fan, Qiang Yang, and Philip S Yu. 2010. Predictive modeling with heterogeneous sources. In *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM, 814–825.
- [13] Arthur Toth, Louis Tan, Giuseppe Di Fabbrizio, and Ankur Datta. 2017. Predicting Shopping Behavior with Mixture of RNNs. (2017).
- [14] Bo Wang, Jie Tang, Wei Fan, Songcan Chen, Zi Yang, and Yanzhu Liu. 2009. Heterogeneous cross domain ranking in latent space. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 987–996.
- [15] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 425–434.