A Brand-level Ranking System with the Customized Attention-GRU Model

Yu Zhu12, Junxiong Zhu3, Jie Hou3, Yongliang Li3, Beidou Wang1, Ziyu Guan4, Deng Cai12

1 State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, China
2 Alibaba-Zhejiang University Joint Institute of Frontier Technologies
3 Alibaba Group, Hangzhou, China
4 School of Information and Technology, Northwest University of China

{zhuyu.cad, dcai}@zju.edu.cn, xike.zjx@taobao.com, muhuo.hj@alibaba-inc.com, anthony.lee.liyl@tmall.com, beidouw@sfu.ca, ziyuguan@nwu.edu.cn

Abstract

In e-commerce websites like Taobao, brand is playing a more important role in influencing users’ decision of click/purchase, partly because users are now attaching more importance to the quality of products and brand is an indicator of quality. However, existing ranking systems are not specifically designed to satisfy this kind of demand. Some design tricks may partially alleviate this problem, but still cannot provide satisfactory results or may create additional interaction cost. In this paper, we design the first brand-level ranking system to address this problem. The key challenge of this system is how to sufficiently exploit users’ rich behavior in e-commerce websites to rank the brands. In our solution, we firstly conduct the feature engineering specifically tailored for the personalized brand ranking problem and then rank the brands by an adapted Attention-GRU model containing three important modifications. Note that our proposed modifications can also apply to many other machine learning models on various tasks. We conduct a series of experiments to evaluate the effectiveness of our proposed ranking model and test the response to the brand-level ranking system from real users on a large-scale e-commerce platform, i.e. Taobao.

1 Introduction

In e-commerce websites such as Taobao, the brand is having a growing impact on customers’ decision of click/purchase [Zipser et al., 2016]. There are numerous reasons behind this phenomenon. On one hand, users may prefer items with high quality, and brand is a good indicator of product quality. On the other hand, users may have preference bias to certain brands because of the brand image built from brand marketing and campaigns, e.g. basketball fans may prefer clothes under the brand endorsed by NBA stars.

*Deng Cai is the corresponding author

Figure 1: Results of different ranking systems given the query jeans. Red lines highlight the important brand-related information.

(a) Results of existing ranking systems

(b) Results after clicking the checkbox Levi’s

(c) Results of our brand-level ranking system

However, as shown in Figure 1(a), the ranking results from most existing retrieval/recommender systems aggregate the items of different brands together. This cannot well satisfy the users with strong preference towards certain brands, since they have to waste time browsing through large numbers of
products from other brands. There exist several technical tricks to help with this issue: (1) Set the brand as a feature for the underlying ranking model, and then items with preferred brands would rank in the top of the list. However, items of different brands are still mixed together. (2) Set checkboxes/buttons to filter brands. As shown in Figure 1(b), on the result page of query “jeans”, users can choose to only browse products under brand Levi’s by clicking the checkbox filter for Levi’s. However, only a few brands can be displayed here and there is no personalization. Moreover, users have to click the checkboxes multiple times if they want to browse several brands, which increases the interaction cost and creates negative user experience.

Due to the drawbacks of existing systems mentioned above, we design a new brand-level ranking system, where items of the same brand are grouped together and brand groups are ranked based on users’ brand preference. A demo of our system is demonstrated in Figure 1(c). Jeans are first grouped by brands, e.g. Levi’s and Wrangler. The system will then rank Levi’s at a higher position, if it learns the user prefers Levi’s to Wrangler.

The core of the brand-level ranking system is to solve the personalized brand ranking problem. Various important information in e-commerce websites should be exploited to rank brands. Firstly, features such as brand pricing contribute to brand profiling. Second, the sequential information in user action (click or purchase) sequences and time intervals between actions are valuable in modeling users’ interests on brands [Zhu et al., 2017]. In addition, different types of actions (click and purchase) reflect users’ different preferences on items (and in turn the corresponding brands), e.g. generally a user purchasing an item indicates he is more interested in the item than if he clicks it. The key challenge is how to well exploit these characteristics of user action sequences for brand ranking. We formulate it as a point-wise ranking problem. Specifically, a classifier is trained based on the various information in e-commerce websites. Then given a user, his probability of preferring each brand is predicted by the classifier and brands are ranked based on their probabilities. Items within the same brand group can be ranked by traditional ranking methods.

Two sub-tasks need to be accomplished for the brand-level ranking system. One is the feature engineering for the brand ranking problem. We propose a series of brand features that are important for brand ranking (details in section 4.1). The other one is the design of the ranking model. We propose an adapted Attention-GRU [Chorowski et al., 2015] model to predict users’ probabilities of preferring different brands. RNN (Recurrent Neural Network) methods have achieved state-of-the-art performance in capturing the sequential information in user action sequences [Zhu et al., 2017]. GRU (Gated Recurrent Unit) [Cho et al., 2014] is among the best RNN architectures and the attention mechanism [Chorowski et al., 2015] helps to distinguish the influence of different previous actions on the current prediction. Therefore, Attention-GRU is chosen as the base model. We propose three major modifications to Attention-GRU for our task. (1) Combine the heuristically designed brand features and the model-learned brand embedding to better represent the brand. A brand embedding refers to a vector representation for the brand, which is learned from training data by our model [Zhu et al., 2016]. (2) Consider different types of actions. (3) Integrate the time-gate [Zhu et al., 2017] to model time intervals between actions, which can better capture users’ short-term and long-term interests. The effectiveness of our adapted model and the brand-level ranking system is evaluated in our offline and online experiments on a large-scale e-commerce platform. This paper’s contributions are outlined as follows.

- We propose the first brand-level ranking system, which provides explicit personalized brand-level ranking and better helps users to make click/purchase decisions based on their brand preference.
- We perform feature engineering tailored for the brand ranking task and propose an adapted Attention-GRU model as our ranking model. Specifically, we contribute three modifications which effectively improve the model performance.
- We conduct extensive offline and online experiments on a large-scale e-commerce platform. The results and feedbacks from real users prove the effectiveness of our adapted model and the brand-level ranking system.

2 Related Work

2.1 RNN, GRU and Attention-GRU

RNN [Elman, 1990] has been proved to perform excellently when modeling sequential data. It is formally defined as:

\[
s_m = f(W_x s_m + Us_m - 1),
\]

\[
o_m = \text{softmax}(Vs_m),
\]

where \(s_m, x_m\) and \(o_m\) are the hidden state, input and output at the \(m\)-th step. \(f\) is a non-linear activation function. \(W, U\) and \(V\) are the corresponding weights learned from training.

GRU [Cho et al., 2014] is an important ingredient of RNN architectures, which can avoid the problem of gradient vanishing. It replaces Eq. (1) with:

\[
z_m = \sigma(W_z x_m + U_z s_{m-1}),
\]

\[
r_m = \sigma(W_r x_m + U_r s_{m-1}),
\]

\[
s_m = z_m \odot \tanh(W_h x_m + U_h (r_m \odot s_{m-1}))
\]

\[
+ (1 - z_m) \odot s_{m-1},
\]

where \(z_m\) and \(r_m\) are the update and reset gates. \(\sigma\) and \(\tanh\) are sigmoidal and tanh nonlinearities. \(W_z, W_r, W_r, W_h\) and \(U_h\) are the weights. \(\odot\) denotes the element-wise product.

Attention-GRU [Chorowski et al., 2015] refers to GRU with the attention mechanism. RNN methods with the attention mechanism have been successfully applied to handwriting synthesis [Graves, 2013], machine comprehension [Pan et al., 2017] etc. It typically focuses on the task that generates an output sequence \(y = (y_1, \ldots, y_T)\) from an input sequence \(x = (x_1, \ldots, x_L)\). \(x\) is often processed by an encoder to output a sequential representation \(h = (h_1, \ldots, h_L)\). At the
3.2 Adaptations of Traditional RNN Models

We first transform $H_u$ to $\tilde{H}_u = [b_{1u}, h_{1u}, t_{2u} - t_{1u}], (b_{2u}, h_{2u}, t_{3u} - t_{2u}), \ldots, (b_{nu}, h_{nu}, T_u - t_{(n-1)u})]$. The mathematical representations of $h_{nu}$, $h_{nu}$ and $t_{nu-1} - t_{nu}$ are denoted as $R(h_{nu})$, $\hat{R}(h_{nu})$ and $\hat{R}(t_{nu-1} - t_{nu})$, respectively. $R(h_{nu})$ is defined with two alternative ways: (1) a brand feature vector or (2) a one-hot vector. $R(h_{nu})$ is defined as a one-hot vector. Since there are two types of actions, i.e. click and purchase, in our task, thus they are represented as $[0, 1]$ and $[1, 0]$, respectively. $R(t_{nu-1} - t_{nu})$ is set to be a scalar $t_{nu-1} - t_{nu}$ (in seconds). The label is 1 if $u$ actually clicks/purchases $B_q_u$ at time $T_u$ and 0 otherwise.

For Time-LSTM [Zhu et al., 2017], the input at each step is the concatenation $[R(h_{nu}), R(h_{nu})]$ and $R(t_{nu-1} - t_{nu})$. For the other RNN models, the input (i.e. $x_m$ in section 2.1) is the concatenation $[R(h_{nu}), R(h_{nu}), R(t_{nu-1} - t_{nu})]$. For RNN with no attention mechanism, the output (i.e. $\alpha_m$ in Eq. (2)) is a probability distribution over all brands, from which we can obtain $p(B_q_u)$. For Attention-GRU, $T$ and $y_m$ (i.e. $y_{m-1}$ in Eq. (8) when $m = 1$) in section 2.1 are set to be 1 and $B_{q_u}$, respectively. We replace Eq. (9) with:

$$\tilde{o}_m = \text{softmax}(\tilde{V} s_m),$$

where $\tilde{V}$ is the weight and $\tilde{o}_m$ is a probability distribution over labels 1 and 0. $p(B_{q_u})$ is equal to the probability of label 1. These models are trained by AdaGrad [Duchi et al., 2011] with the log loss calculated by $p(B_{q_u})$ and the label.

4 Proposed System

4.1 Feature Engineering

According to the market research conducted by the authors’ company, the price range feature demonstrates great importance in user and brand profiling. Therefore, the brand features are carefully engineered by slicing into 7-level price ranges, with the details as follows (also shown in Figure 2).

(1) All items belonging to a certain category (e.g. “clothing”) are sorted based on their prices. The sorted item list is denoted as $\{i_1, i_2, \ldots, i_n\}$. The price for $i_k$ is denoted as $p_t$. We define 7 price levels as 7 price ranges, with Level 1 $= (0, p_2]$, Level 2 $(p_2, p_3], \ldots$, Level 7 $= (p_6, p_7]$. (2) Given a certain brand, items belonging to this brand are grouped based on the price levels defined above. Specifically, an item is grouped to a price level if this item’s price locates in the corresponding price range.

(3) Given a certain brand, 8 most important e-commerce metrics are aggregated within each price level, generating 8-dimensional features for each price level. Features from all 7 price levels are then concatenated into the final 56-dimensional brand features. Specifically, the 8 most important brand-related e-commerce metrics used in this paper is summarized in Table 1.

4.2 The Design of the Ranking Model

An adapted Attention-GRU model is proposed to predict users’ probabilities of preferring different brands.

### Table 1: 8 Most Important E-commerce Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTR (Click-Through Rate)</td>
<td>Clicks/Impressions ratio</td>
</tr>
<tr>
<td>CVR (Conversion Rate)</td>
<td>Conversions/Clicks ratio</td>
</tr>
<tr>
<td>GMV (Gross Merchandise Value)</td>
<td>Total sales for a specific period</td>
</tr>
<tr>
<td>ATIP (Average Transacted Item Price)</td>
<td>Mean price of items purchased</td>
</tr>
<tr>
<td>Search Time</td>
<td>Time users spend searching for a product</td>
</tr>
<tr>
<td>Click Times</td>
<td>Time users spend clicking on a product</td>
</tr>
<tr>
<td>Add-To Cart Times</td>
<td>Time users spend adding to cart items</td>
</tr>
<tr>
<td>Transaction Times</td>
<td>Time users spend completing a transaction</td>
</tr>
</tbody>
</table>

(Due to page limit, please refer to the corresponding Wikipedia pages for definitions of CTR, CVR and GMV. ATIP = GMV Number of Transacted Items)
(1) The Choice of Attention-GRU
RNN solutions have achieved state-of-the-art performance when modeling users’ sequential actions [Hidasi et al., 2016; Zhu et al., 2017]. LSTM and GRU are two important ingredients of RNN architectures, both of which are able to avoid the problem of gradient vanishing. Compared to LSTM, GRU can yield comparable performance but has fewer parameters, thus it could train a bit faster and need less data to generalize [Chung et al., 2014]. Figure 3 is an example to show the motivation to apply the attention mechanism in our task. When we predict whether the user would click/purchase the jacket with the brand Adidas, we want to increase the influence of the first and third items on the prediction since compared to the cellphone and computer, the trousers and shoes are much more related to the jacket. In the attention mechanism, this will be reflected with larger weights on the first and third inputs.

(2) Modifications to Attention-GRU
To achieve better performance, we propose three modifications to Attention-GRU (also explained in Figure 4):

Modification 1: Combining the brand features and brand embedding to better represent the brand
In a traditional Attention-GRU model, the brand can be represented by either a vector consisting of brand features defined in section 4.1, or a one-hot vector based on brand ID. However, the brand features are heuristically designed based on expert experience and may lack some important information useful for the brand ranking task. For example, two brands may have similar brand features but are of different interest for the same user. Using one-hot vectors can explicitly distinguish different brands but lacks the brand content information. Here we propose to combine these two representations. Specifically, following the notations in section 3.1, assume \( b^u_m \) equals to \( B_k \) and the vector consisting of brand features for \( B_k \) is denoted as \( v_k \in \mathbb{R}^{56 \times 1} \). Its one-hot vector is defined as \( o_k = \{0, 1\}^{N \times 1} \), with the \( k \)-th entry equal to 1 and the other entries equal to 0. Matrix \( \mathbf{M}^\text{embed} \in \mathbb{R}^{56 \times 56} \) is defined to contain the embeddings of all brands. Then the combined mathematical representation considering both brand features and the brand embedding is defined as \( R^b(b^u_m) \), with

\[
R^b(b^u_m) = \mathbf{M}^\text{embed} \times o_k + v_k. \tag{11}
\]

\( \mathbf{M}^\text{embed} \) is learned from the training data. \( o_k \) is used to look up the embedding of \( B_k \) from \( \mathbf{M}^\text{embed} \).

We interpret Eq. (11) from three perspectives. (1) Heuristic initialization + Fine tuning. While \( v_k \) is heuristically designed, by learning from the training data, we fine tune the brand’s representation to be \( \mathbf{M}^\text{embed} \times o_k + v_k \), which would much better fit our task. (2) Prior information + Posterior modification. This perspective is similar to the first perspective, but in a Bayesian context [Berger, 2013]. Specifically, \( v_k \) is obtained based on our “prior” understanding of brands, then the training data enhances our understanding of them and allows us to give “posterior” estimations of their representations. (3) Content information + Collaborative information. \( v_k \) contains brands’ content information, while \( \mathbf{M}^\text{embed} \times o_k \) is learned from users’ click/purchase data in our task and thus captures the collaborative information. For some brands \( B_k \) that rarely appear in the training data, \( v_k \) can provide extra information to help with this “cold-start” situation [Wang et al., 2016a; 2016b].

Modification 2: Considering different types of actions
When modeling user behaviors by RNN models, previous works such as [Hidasi et al., 2016; Zhu et al., 2017] usually consider single type of action. However, users can perform multiple types of actions on items, e.g. click and purchase. One way to handle different types of actions is to model each of them as a one-hot vector \( R(h^u_m) \) (defined in section 3.2) and concatenate it to our combined brand representation \( R^b(b^u_m) \). However, this method allows no feature interaction between \( R^b(b^u_m) \) and \( R(h^u_m) \) (or implicit interaction by the non-linear activation function), and thus would not well capture the interactive relation between the brand and the type of action performed on the brand [Rendle, 2012; Chen et al., 2017].

Instead, we propose to define a matrix for each type of action (the total number of action types is usually not large). In our task, we define \( \mathbf{M}^\text{click} \in \mathbb{R}^{56 \times 56} \) and \( \mathbf{M}^\text{purchase} \in \mathbb{R}^{56 \times 56} \) for click and purchase, respectively. Then the representation considering both of the brand and the type of action is defined as:

\[
R(b^u_m) = \begin{cases}
\mathbf{M}^\text{click} \times R^b(b^u_m), & \text{if } b^u_m \text{ is clicked}, \\
\mathbf{M}^\text{purchase} \times R^b(b^u_m), & \text{if } b^u_m \text{ is purchased}.
\end{cases} \tag{12}
\]

In this way, we explicitly model the interaction between the brand and the type of action by matrix multiplication.

Modification 3: Integrating the time-gate to model time intervals between actions
The time gate in Time-LSTM [Zhu et al., 2017] is effective to capture users’ short-term and long-term interests. We propose that the time gate is a basic component and adapt it to our model as follows. A time gate \( T_m \) is defined as:

\[
T_m = \sigma(W_t x_m + \sigma(Q_t \Delta t_m)), \tag{13}
\]

where

\[
x_m = R(b^u_m), \tag{14}
\]

\[
\Delta t_m = R(t^u_{m+1} - t^u_m) = t^u_{m+1} - t^u_m.
\]

\( W_t \) and \( Q_t \) are the weights of \( x_m \) and \( \Delta t_m \) (time intervals), respectively. \( \Delta t_m \) may be very large, thus we impose a sigmoidal function \( \sigma \) on \( Q_t \Delta t_m \). Then we modify Eq. (5) to:

\[
s_m = s_m \odot T_m \odot \tanh(W_h x_m + U_h (r_m \odot s_{m-1} - 1) \odot s_{m-1}) + (1 - s_m) \odot s_{m-1}. \tag{15}
\]

When modeling user behaviors by our model, \( x_m \) in Eq. (15) represents the user’s most recent action, thus we can exploit \( x_m \) to learn his/her current short-term interest. \( s_{m-1} \) models this user’s previous actions, thus \( s_{m-1} \) reflects his/her long-term interest. \( T_m \) is helpful in two ways, i.e. (1)
\[ \text{tanh}(W_h x_m + U_h (r(m) \odot s_{m-1})) \] is filtered by not only \( z_m \), but also \( T_m \). So \( T_m \) can control the influence of \( x_m \) on current prediction. (2) \( \Delta t_m \) is firstly stored in \( T_m \), then transferred to \( s_m \), and would be transferred to \( s_{m+1}, s_{m+2}, \ldots \). Thus \( T_m \) helps to better model users’ long-term interest (\( s_m, s_{m+1}, \ldots \)) by incorporating \( \Delta t_m \) for later predictions. We denote Attention-GRU with the time gate as Time-Attention-GRU.

It is notable that Modification 1 is rather general and can apply to many machine learning models where the designed features and one-hot vectors exist. Modification 2 can be generalized to handle tasks involving various interaction types with the input. In Modification 3, we propose that the time gate can be integrated into not only LSTM but also other RNN models, just as how we integrate it into Attention-GRU.

**3 Loss and Training**

Similar to Attention-GRU, \( p(B_{uq}) \) in our model is obtained from \( o_m \) in Eq. (10). We define the loss for user \( u \) as:

\[
\text{loss}_u = \begin{cases} 
- \log(p(B_{uq})), & \text{if label}_u = 1, \\
-w \times \log(1 - p(B_{uq})), & \text{otherwise}, 
\end{cases} 
\]

(16)

where \( \text{label}_u = 1 \) if \( u \) actually generates an action on brand \( B_{uq} \) at time \( T_u \). We multiply the loss by \( w \) (\( < 1 \)) for negative instances because some unlabeled positive instances may be mixed in the training data [Pan et al., 2008]. Our model is optimized by AdaGrad [Duchi et al., 2011].

5 Offline Experiments

**5.1 Dataset**

A large-scale dataset is collected from Taobao. Specifically, we extract tuples \( \langle \text{user}_id, \text{brand}_id, \text{action}_type, \text{timestamp} \rangle \), with each tuple representing that user \( \text{user}_id \) has an action \( \text{action}_type \) on brand \( \text{brand}_id \) at time \( \text{timestamp} \). Users and brands with few interactions are filtered. The final data set in our offline experiments consists of \( M = 3, 591, 372 \) users, \( N = 90, 529 \) brands and 82, 960, 693 actions.

For each user \( u \), his/her action sequence is cut into short ones with the length equal to 11. The first 10 actions form \( H^u \). The brand and timestamp in the last action are assigned to \( B_{uq} \) and \( T_u \) respectively, serving as the positive instances (\( \text{label}_u = 1 \)) in training. The negative instances (\( \text{label}_u = 0 \)) are generated by replacing \( B_{uq} \) with another random brand.

**5.2 Compared Models and Evaluations**

Our proposed model, Attention-GRU with three modifications, is denoted as Attention-GRU-3M, and is compared with the following baselines.

**GRU** [Cho et al., 2014] is among the best RNN architectures. Thus we choose GRU as a representative of the original RNN models.

**Attention-GRU** Similar to Attention-GRU [Chorowski et al., 2015] is selected to represent RNN models with the attention mechanism.

**Time-LSTM** Time-LSTM [Zhu et al., 2017] has achieved state-of-the-art performance for sequential behavior modeling. Thus it would be a competitive baseline in our task.

**Session-RNN** Session-RNN [Hidasi et al., 2016] exploits RNN to capture users’ short-term interest based on sequential actions within a session. We use the publicly available python implementation\(^2\) of Session-RNN, with the session identified by timeout [Huang et al., 2004].

**libFM** Our task can also be treated as a brand recommendation problem. Therefore, many hybrid recommendation methods that simultaneously capture the content and collaborative information can be applied. We use the popular libFM model [Rendle, 2012] as a representative of these methods.

**GRU, Attention-GRU, Time-LSTM and Session-RNN** are adapted to our task as described in section 3.2. For libFM, we extract tuples \( \langle \text{user}_id, \text{brand}_id, \text{label} \rangle \), where label is 1 if user \( \text{user}_id \) has an action on brand \( \text{brand}_id \) and 0 otherwise. Then the one-hot vector for \( \text{user}_id \), the one-hot vector for \( \text{brand}_id \) and the corresponding brand feature vector are concatenated to be \( x \) and label is \( y \) in libFM. The number of units is empirically set to 256 for RNN models. The other hyperparameters in all models are tuned via cross-validation or set as in the original paper. Our code is public \(^3\).

AUC and F1 score [Kim and Leskovec, 2013] are used to evaluate the prediction performance of different models.

**5.3 Results and Discussions**

**Model Comparison**

As shown in Table 2, our proposed model significantly outperforms all baselines and we attribute it to the adoption of

\(^2\)https://github.com/hidasib/GRU4Rec

\(^3\)https://github.com/zyody/Attention-GRU-3M
the attention mechanism and three proposed modifications. In comparison, baselines GRU and Session-RNN, exploiting none of these merits, perform the worst. Compared with GRU, Attention-GRU adopts additional attention mechanism and Time-LSTM adopts Modification 3 proposed in section 4.2. Thus, they both outperform GRU and Session-RNN. As a hybrid recommendation approach, libFM captures content and collaborative filtering features at the same time, but fails to capture the sequential information of user behaviors, while our model captures all these information. This explains why libFM performs worse than our model.

The Effects of Three Modifications

As described in section 4.2, three modifications are proposed to enhance Attention-GRU. We now remove them one at a time to evaluate how each modification affects the prediction performance. We also change the size of data used in training to evaluate its effect on the performance. As shown in Figure 5 (The results of AUC and F1 are similar. Due to page limit, we only show the results of AUC), AUC declines when we remove each type of modification and declines the most when removing Modification 1, which indicates that all the proposed modifications contribute to the performance improvement and Modification 1 is more effective. As the size of training data increases, the influence of Modification 1 becomes larger (i.e. the gap between “Attention-GRU-3M” and “No Modification 1” becomes larger). One explanation may be that, compared to the other two modifications, Modification 1 brings much more parameters (i.e. entries of \( M_{embed} \)), thus it needs more training data to learn their optimal values. This also explains that in Table 2 and Figure 5, our model has a larger performance improvement than the other models as the data size increases.

### Table 2: Model Comparison

<table>
<thead>
<tr>
<th></th>
<th>50% training data</th>
<th>100% training data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>F1</td>
</tr>
<tr>
<td>GRU</td>
<td>0.583</td>
<td>0.557</td>
</tr>
<tr>
<td>Attention-GRU</td>
<td>0.591</td>
<td>0.569</td>
</tr>
<tr>
<td>Time-LSTM</td>
<td>0.592</td>
<td>0.570</td>
</tr>
<tr>
<td>Session-RNN</td>
<td>0.583</td>
<td>0.556</td>
</tr>
<tr>
<td>libFM</td>
<td>0.589</td>
<td>0.566</td>
</tr>
<tr>
<td>Attention-GRU-3M</td>
<td>0.605</td>
<td>0.580</td>
</tr>
</tbody>
</table>

Figure 5: Performance measured by AUC when we remove three modifications one at a time and vary the training data size. No Modification i refers to our model without Modification i.

### Table 3: Performance of Online Experiments

<table>
<thead>
<tr>
<th></th>
<th>CTR</th>
<th>ATIP</th>
<th>GMV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>50.77%</td>
<td>CNY 140.13</td>
<td>CNY 499,466.85</td>
</tr>
<tr>
<td>New Version</td>
<td>50.39%</td>
<td>CNY 144.90</td>
<td>CNY 499,466.85</td>
</tr>
<tr>
<td>Improvement</td>
<td>0.39%</td>
<td>3.40%</td>
<td>3.51%</td>
</tr>
</tbody>
</table>

Figure 6: The original ranking system of the platform is shown in the left. Users can switch to our brand-level ranking system by clicking the button “Brand” as in the right.

### 6 Online Experiments

To better evaluate users’ real response to our brand-level ranking system, we conduct a series of online experiments on the Tmall search scenario in Taobao. In this scenario, there are about \( 5 \times 10^8 \) clicks and \( 2 \times 10^6 \) purchases on items of nearly \( 10^5 \) brands from over \( 10^7 \) customers within one normal day. A standard A/B test is conducted online. As shown in Figure 6, we design two user interfaces, one adopts the original ranking system (left), and a button “Brand” is added to the other one (right), where users can switch to our brand-level ranking system by clicking the button. We preserve the original ranking system in the right for users who have no brand preference. For each setting, the same number (about \( 5 \times 10^5 \) per day) of users are randomly selected for A/B test. We perform the online experiments for seven days, and the average CTR, ATIP and GMV (described in section 4.1) per day are reported.

The results are shown in Table 3. We can see that compared with the baseline, CTR and ATIP in the proposed new ranking system are both improved, which indicates that by incorporating the brand-level ranking system, users are more likely to click the items and at the same time, they tend to purchase items with higher price (likely to be of higher quality). As a result, the key metric to optimize in this platform, i.e. GMV, has an improvement of 3.51%. Considering the traffic of the platform, it would result in a significant boost in revenue. Our brand-level ranking system has already gone into production on the platform, currently serving as a feature requiring user’s proactive opt-in to be activated. Nearly \( 4 \times 10^5 \) users actively use this system per day.

### 7 Conclusion

In this paper, we propose a brand-level ranking system to better satisfy the demand of users who have preference bias to certain brands. The core of this system is to solve the personalized brand ranking problem. In our solution, we firstly carefully design the brand features, then rank the brands by...
an adapted Attention-GRU model. In future work, we will explore the effectiveness of our modifications on other machine learning models (e.g. Modification 1 on Convolutional Neural Network [Krizhevsky et al., 2012], Modification 2 on Matrix Factorization [Qian et al., 2016]). In addition, we would improve the design of our ranking model (e.g. considering the exploration of new brands by reinforcement learning).

Acknowledgements
This work was supported in part by the National Nature Science Foundation of China (Grant Nos: 61751307, 61522206, 61373118, 61672409), the grant ZJU Research 083650 of the ZJUI Research Program from Zhejiang University, the National Young Top-notch Talent Support Program, the Major Basic Research Project of Shaanxi Province (Grant No. 2017ZDJC-31), and the Science and Technology Plan Program in Shaanxi Province of China (Grant No. 2017KJXX-80). Also thanks to the colleagues in Alibaba Group.

References


