Two-stage Interest Calibration Network for Reranking Hotels

Denghao Ma[1], Yueguo Chen[2], and Xiaoyong Du[2]

[1] Meituan, Beijing, China
[2] Renmin University of China, Beijing, China
madenghao@ruc.edu.cn, chenyueguo@ruc.edu.cn, duyong@ruc.edu.cn,

Abstract. As one key task of hotel search, interest modelling is to capture users' interests over their historical clicks. To this end, it is important yet challenging to understand the searching intention behind each click (interest focus), i.e., what did the user like about the hotel when clicked a hotel? Besides, the query input by a user represents the user's interests to some extent. The differences between historical queries and the current query result in the gap between historical interests and the current interest (interest gap). Because historical clicks stem from historical interests, using them to model the current interests inevitably suffers from the interest gap challenge. To capture the interest focus and address the interest gap, we propose the two-stage interest calibration network (TCN), i.e., search-internal and search-external. In the search-internal calibration, we propose new insights of using the divergences among clicks and unclicks to model interest focus, and then develop a divergence-based calibration network. In the search-external calibration, inspired by the smoothing techniques for language models, we propose the interest smoothing principle to bridge interest gap: the interests learnt from historical clicks + smoothing factor $\approx$ current interests. To implement this principle, we develop an interest smoothing network by reusing the query data. In the network, the interest domination unit is developed to learn the interest representation from historical clicks, and the interest smoothing unit is developed to construct the smoothing factor. Extensive offline experiments and online A/B testing are performed and show that TCN significantly outperforms the baselines. Besides, our model has been deployed in a hotel e-commerce platform and brought 2.90% CTCVR and 1.53% CTR lifts.

1 Introduction

In recent years, users (guests) have become accustomed to searching accommodations through hotel e-commerce platforms, such as *Airbnb*, *Ctrip*, and *Meituan-Hotel*. As well, hosts, who provide places to stay, have taken hotel e-commerce platforms as key tools to publish their accommodation information. As a two-sided marketplace, a hotel e-commerce platform needs to satisfy the requirements of both guests and hosts— guests ask it to recommend cost-effective rooms and hosts ask to bring in more guests. To this end, a growing number of researchers [1–3] pay attention to the hotel search task. A typical pipeline of hotel search systems contains three main stages: 1) Recall, retrieving relevant hotels from a large database of hotels; 2) Rank, selecting the top thousand from the relevant hotels as candidates; 3) Rerank, sorting the candidates by fine-grained features (e.g., behavior-based interest preference), and returns the top hundred as output.

Interest modelling aims to construct users' interest representations over their behavior sequences. It has been verified very important to product recommendation [4–6]. However, the interest modelling solutions proposed for product recommendation can not work well in hotel search because of the characteristics of hotel search— In hotel search, 1) users' interests are more concentrated than those in product recommendation. The targets of hotel search belong to only one domain, i.e., hotel, while the targets of product recommendation belong to multiple domains, e.g., book and food; 2) Users' interests are highly dependent on their queries, e.g., a check-in location, number of guests, price and check-in/checkout dates; 3) Users' behavior sequences are more sparse than those in product recommendation [7]. These motivate us to study the interest modelling task for reranking hotels (at the rerank stage).

We identify two challenges for interest modelling in hotel search:
Challenge one: Interest focus. In hotel e-commerce platforms, a hotel has many attributes and each attribute has multiple values. When a user clicked the hotel, she might only like some attributes and values (interest focus). Which attributes and values did the user like? Modelling the interest focus on a click is critical for modelling the user's interests on all clicks, but is very challenging. For a hotel, different users who clicked it may have different interest focuses, because they care about different attributes and values. How to use the same clicked hotel to model different interest focuses for different users? To the best of our knowledge, few studies have explored this problem.

Challenge two: Interest gap. A user searches hotels by entering a query, e.g., the check-in location, number of guests and check-in/checkout dates. The query represents user's interests to some extent. For instance, if the number of guests is two and the check-in date is Valentine' s day, the user's interests in the room type may be double rooms, while that may be single rooms if the number of guests is one. For a user, the current query is different from her historical queries, and thus the current interests may be different from historical interests (interest gap). Because historical clicks stem from historical interests, using them to model the current interest inevitably suffers from the interest gap challenge. How to bridge the interest gap challenge?

In hotel search, the interest gap has not been explored. In product recommendation, one approach is modelling the interest evolving process [8]. However, this approach can not work well in the hotel domain because 1) the click sequences in the hotel domain are much shorter than those in the product domain. The short sequences may not provide sufficient statistics for the interest evolution learning; 2) Time intervals between clicks in the hotel domain are much longer than those in the product domain. The long time intervals severely weaken the relationship between clicks, and thus hinder the modelling of interest evolution.

Solution. We focus on addressing the above two challenges and propose a two-stage interest calibration network to address them.

Search-internal Calibration. In a search, user's interests are often specific and focus on some attribute values of hotels. Therefore, there is high similarity among hotels that were clicked in the search, while there is high divergence between clicked hotels and unclicked hotels in the search. To capture and model the interest focus on a click, we propose three insights by using the search data. In a search, for an attribute (e.g., price), 1) if its attribute values on clicked hotels are more similar, the user cares more about this attribute, and thus it is more important for constructing the interest focus; 2) if its values on the clicked hotels are more different from those on the unclicked hotels, the attribute is more important; 3) if one attribute value is more similar to the values of clicked hotels and more different from the values of unclicked hotels, the value is more important (details in Section 4.1). To capture these insights, we develop a divergence-based calibration network to model the interest focus on a click.

Search-external Calibration. Inspired by the smoothing techniques for language models [9], we propose an interest smoothing principle to bridge the interest gap: *the interests learnt from historical clicks* $+$ *smoothing factor* $\approx$ *current interests*. To implement this principle, we develop the interest smoothing network by reusing the query data. Specifically, the interest domination unit is developed to learn the interest representation from historical clicks. It uses a click-extension approach and query-extension approach to estimate the relevance between the current query and historical clicks, The click-extension approach uses the historical query corresponding to a click as a bridge between the click and current query, and the query-extension approach uses the relevant hotels of the current query as the bridge. The interest smoothing unit is developed to construct the smoothing factor by reusing the query data. Like the query-extension approach, it derives the smoothing factor from both the current query itself and the relevant hotels of the current query.

Our contributions are concluded as follows:

- We define the interest modelling task by emphasizing the search and query data and identifying two new challenges, i.e., interest focus and interest gap.
- We are the first to explore the interest focus in the task of interest modelling, propose three insights based on search data, and develop the divergence-based calibration network to capture the interest focus.
- To bridge the interest gap challenge, we propose the interest smoothing principle, and then develop the interest smoothing network by reusing the query data.
- Both offline experiments on two large-scale industrial datasets and online A/B testing have been conducted to validate the effectiveness of our model. Our model has been deployed in a major hotel e-commerce platform and brought 2.90% CTCVR and 1.53% CTR lifts, which is significant to the hotel business.

2 Related Work

2.1 Hotel Search

We make a brief survey about the solutions and datasets for hotel search.

Solutions for Hotel Search. Previous solutions for hotel search may be grouped into four classes, i.e., the feature engineering based solutions, machine learning based solutions and deep learning based solutions. In the feature engineering based solutions, many features of both hotels and users are designed, such as hotel popularity, hotel star, brand, price, location, ratings, user age, the numbers of clicks, orders and refunds, and so on [7, 1]. In the machine learning based solutions [10, 12], the Boosting Trees, Logistic Regression, SVMRank and Random Forests techniques are widely used for ranking candidate hotels. Recently, many deep learning solutions have been proposed to rank candidate hotels. For example, the study [3] uses the Recurrent Neural Networks to produce an embedding of the entire query context, and uses the embedding to manage the diversity of hotel results; The study [2] proposes a new ranking neural network to address the positional bias and cold start problems.

Datasets for Hotel Search. To support the research of hotel search, some studies and organizations publish the datasets with different emphases. The dataset [1] emphasizes the customer reviews and scores. The datasets [3] emphasize the metadata, such as category, description, facilities, star rating, image count, location and others. Recently, the study [7] publishes a session-based hotel recommendations dataset, which emphasizes the interactions of users with hotel e-commerce platforms and is sampled from the trivago website between November 1, 2018, and November 8, 2018. Besides, Alibaba group releases user behaviour data from Fliggy platform [4], which covers all behaviors (including clicks, favorites, adding, and purchases) of approximately 2 million random users from June 3, 2019 to June 3, 2021. However, these datasets don't contain the search-based click sequences and query sequences, and the candidate hotel list of the user's current query, which are the input of our model.

2.2 Interest Modelling for Product Recommendation

Since the interest modelling over users' behavior sequences in hotel search has not been deeply explored, we survey the solutions of interest modelling in product recommendation. To model the diverse interests of a user, DIN [13] uses the target to estimate the importance of each click, and MIND [4] uses a capsule network and a dynamic routing method to construct multiple vectors for the user's interests. The study [8] designs an interest extraction layer by GRU with auxiliary loss, and an interest evolving layer to model the interest evolving process. In study [14], a hierarchical periodic memory network with personalized memorization is designed for modelling the multi-scale sequential patterns of user interests. To model the long-term interest, the study [15] proposes an attentive Asymmetric-SVD paradigm, and the study [6] proposes a self-attention based sequential model to capture long-term semantics and estimate the relevance of each historical click. In study [16], a novel deep feedback network is proposed to model users' interests over click, unclick and dislike behaviors. To reduce the effect of noise clicks on the interest representations, the study [17] proposes an all-MLP model with learnable filters. Considering the cost of computation of

[3] https://data.world/datafiniti/hotelreviews, www.kaggle.com/PromptCloudHQ/

[4] https://tianchi.aliyun.com/dataset/dataDetail?dataId=113649

Table 1: Notations and Descriptions.

Notation	Description
u,d,q	A user, the candidate hotel and the current query.
S, Q, X, Y	The search, query, click and unclick sequences.
X_k, Y_k, q_k	The click and unclick sequences, query in the search s_k.
$\overline{u}$, $\overline{d}$	Features of user u and hotel d.
$\hat{u}$, $R(u)$	The profile features and interest feature of a user u.
E_{x_i},E_{y_z}, E_q	Embeddings of click x_i, unclick y_z, query q.
$E^j_{x_i}$,$E^j_{y_z}$	The j-th dimension in the embedding of click x_i and unclick y_z.
R_X, R_q	Interest representation from X and q.
y_c,y_r	Ground truth label of CTCVR and CTR.
σ_*, W_*, b_*	MLP with multiple layers, weight matrix and bias matrix.
$\oplus$, $\circ$	Concat operation and Hadamard product.

long term historical behavior sequences, the study [18]searches the important clicks of the target, and then models users' interests over the important clicks.

Different from the above work, this paper studies interest modelling for reranking hotels. First, we identify two new challenges for interest modelling (interest focus and interest gap), and highlight the importance of search and query data for the task. Second, to capture interest focus, we propose three insights based on search data, and develop a divergence-based calibration network. Third, to bridge the interest gap, we propose an interest smoothing principle, and then develop an interest smoothing network by reusing the query data.

3 Preliminaries

3.1 Hotel Search Architecture

Recall. In this stage, the user's query is used for matching all hotels from a large database of hotels and the matched hotels are returned as candidate hotels. Specifically, if the check-in location is a geographical area, all hotels in the area are returned as candidates; If the location is a geographical point, the hotels within 10km from the check-in location are returned. Secondly, we check the status from the check-in date to checkout date, and filter candidates with closed status. Finally, we filter hotels that can not meet the number of guests.
Rank The recall stage retrieves all hotels that meet the user's query as candidates. Too many candidates can not be directly transferred to the rerank stage, because the rerank stage usually uses a complex and effective model to sort candidates, and too many input candidates will dramatically increase the computational cost. To solve this problem, the rank stage is designed to quickly select the top thousand from candidates retrieved by the recall stage, and transfers these thousand candidates to the rerank stage. In the rank stage, XGBoost [19] is widely applied to select candidates, because of its simplicity and efficiency.
Rerank Because the hotel business cares about both click through rate (CTR) and click through&conversion rate (CTCVR), we adopt a multi-task learning approach to optimize CTR and CTCVR prediction tasks, i.e., MMoE [22]. MMoE uses a mixture-of-experts structure and gating network to learn multi-tasks. In the training phase, both CTR and CTCVR are estimated and then used for computing the loss. In the inference phase, only CTCVR is estimated, and all candidate hotels are ranked by their CTCVR scores.

The CTCVR prediction is to estimate the probability $p_{cr}(d|u,q)$ of a user u clicking and reserving a hotel d under query q. CTR prediction is to estimate the probability $p_c(d|u,q)$ of u clicking d under q. We estimate them as follows:

$$p_{cr}(d|u,q) = p_c(d|u,q) \cdot p_r(d|u,q) \tag{1}$$

$$p_c(d|u,q) = \sigma_c(\overline{d}, \overline{u}, q, \Psi_c); \quad p_r(d|u,q) = \sigma_r(\overline{d}, \overline{u}, q, \Psi_r) \tag{2}$$

where $p_r(d|u,q)$ is the probability that a user u reserves a hotel d, after clicked it. The $\overline{d}$ and $\overline{u}$ are the features of d and u. The $\sigma_*()$ with parameters Ψ_* is a $m \times n \times 1$ MLP with ReLU [27], ReLU and Sigmoid [28] activation functions.

In a search q, given a candidate hotel $d \in D$, a user $u \in U$ may click d, reserve or not. We can use a tuple to record the response of u on d, i.e., (u, d, q, y_c, y_{cr}) where $y_c = 1$ if u clicks d, otherwise $y_c = 0$; $y_{cr} = 1$ if u clicks and reserves d, otherwise $y_{cr} = 0$. Based on examples C, we apply the cross-entropy loss function [20] to optimize models, and formulate the loss as follows:

$$L_{ctcvr} = -\frac{1}{|C|} \sum_{u,d,q,y_c,y_{cr} \in C} \{y_{cr} \log p_{cr} + (1 - y_{cr}) \log(1 - p_{cr})\} \tag{3}$$

$$L_{ctr} = -\frac{1}{|C|} \sum_{u,d,q,y_c,y_{cr} \in C} \{y_c \log p_c + (1 - y_c) \log(1 - p_c)\} \tag{4}$$

where the final loss is estimated as $L_{ctr} + L_{ctcvr}$.

The user feature $\overline{u} = (\hat{u}, R(u))$ consists of the profile features $\hat{u}$ and the sequential feature $R(u)$. The $\hat{u}$ may include gender, age level, purchase power, preferred categories and brands. The item features $\overline{d}$ contain category, brand, cumulative order number, the lowest price, and so on. In this paper, our contributions focus on constructing the sequential feature $R(u)$.

3.2 Problem Definition

Because the task of interest modelling is in the rerank stage and the stage receives candidates T_q from the rank stage, the task can take T_q as input. Since the selection of T_q in rank stage is based on the current query q, T_q are relevant to q and represent user's interests to some extent. We define the task as follows:
The input consists of 1) a user $u \in U$; 2) the sequence S, Q, X and Y where each $s_k \in S$ contains $q_k \in Q$, $X_k \in X$ and $Y_k \in Y$; 3) the current query q and the relevant candidate hotels T_q of q. Both historical query q_k and the current query q consist of the check-in location, check-in and checkout dates, price, star and the number of guests. The output is an embedded representation $R(u)$ that represents the current interest of the user u.

Different from the definitions of interest modelling in product recommendation, we first use the searches to organize clicks and unclicks. In a search, the user's needs and interests are consistent, and the search-based organization can provide effective semantics to model relations among clicks and unclicks. Secondly, we use the historical query to extend semantics of historical clicks, and use the current query to supervise interest modelling. Thirdly, the candidate hotels T_q are taken as input, because T_q represent user's interest to some extent.

According to the definition, we formulate interest modelling as follows:

$$R(u) = f(S, Q, X, Y, q, T_q, \psi) \tag{5}$$

where f is a function of generating the interest representation, and ψ is a set of parameters. We sought to design an effective function f with parameters ψ.

4 Solution: Two-stage Interest Calibration Network

We propose a two-stage interest calibration network to learn $R(u)$, i.e., search-internal calibration for modelling the interest focus and search-external calibration for bridging the interest gap.

4.1 Search-internal Calibration

We analyze a lot of searches from a major hotel e-commerce platform, and find that in a search, 1) clicked hotels often share similar attributes and values; 2) the attributes and values of clicked hotels are usually distinct from those of unclicked hotels. Accordingly, we propose three insights to model interest focus.

Insight one. In a search, for an attribute, if its values on the clicked hotels are more similar, the user cares more about this attribute and the attribute is more important for modelling the interest focus. When searching hotels, a user usually has clear needs on several attributes. The needs lead to user's behaviors— only clicking the hotels that meet their needs. Therefore, the attribute values of clicked hotels are similar. So we can use the similarities of attribute values on clicked hotels to discover the liking attributes of a user.

Insight two. In a search, for an attribute, if its values on clicked hotels are more different from those on unclicked hotels, the attribute is more important for modelling the interest focus. In other words, if we can distinguish the clicked hotels from the unclicked hotels only by the values of an attribute, the attribute is the interest focus. This insight tells us— the differences of attribute values between clicked hotels and unclicked hotels can be used for identifying the liking attributes and estimating the liking degrees on the attributes.

Insight three. In a search, if an attribute value is more similar to the attribute values of clicked hotels, but more different from the attribute values of unclicked hotels, the value is more likely to be the interest focus. For example, given a price sequence of clicked hotels $X_k = \{150\$, 135\$, 145\$, 260\$\}$ and that of unclicked hotels $Y_k = \{90\$, 300\$, 250\$, 240\$, 230\$\}$, a price value 145\$ is more important for modelling interest focus than a price value 260\$, because 145\$ is more similar to X_k and more different from Y_k than 260\$.

According to the above three insights, we propose the divergence-based calibration network, and formulate the network as follows:

$$v(x_i) = W \circ E_{x_i} = f_1(q_k \in Q, X_k \in S_k, Y_k \in S_k, x_i \in X_k, E) \circ E_{x_i} \tag{6}$$

where $\circ$ denotes the Hadamard dot and E is a trainable embedding dictionary. In E, every feature is initialized as an embedding. The specific initialization of

Table 2: Feature and embedding initialization.

Feature Group	Feature	Feature Dimension	Feature Type	Embedding Dimension
User features	gender	2	one-hot	16
	age level	150	one-hot	32
	education level	20	one-hot	16
	vip level	8	one-hot	16
	...	...	...	...
Sequence features	click sequence	10^9	multi-hot	128
	unclick sequence	10^9	multi-hot	128
	...	...	...	...
Hotel features	id	10^9	one-hot	128
	category id	20	one-hot	16
	brand id	10^4	multi-hot	64
	...	...	...	...

features and embeddings is shown in Table 2. For a feature of the one-hot type, we initialize its embedded representation as a single embedding vector; For a feature of the multi-hot type, we initialize its embedded representation as a list of embedding vectors. The E_{x_i} is the embedding of click x_i and is consistent for all users. To get E_{x_i}, we concatenate the embeddings of all features on x_i. The W is a learnable weight vector and has the same dimensions of E_{x_i}. It ensures that different users who clicked the same hotel may have different interest focuses.

To capture the insight one, we use the element divergence of a dimension to estimate the importance of the dimension. Let X_k^j=$[E_{x_1}^j; \cdots; E_{x_n}^j]$ denote the elements of j-th dimension of the clicked hotels X_k. We use the relative divergence of X_k^j to estimate the importance of the j-th dimension as follows:

$$W_{X\to X}(X_k^j) = \frac{1}{RD(X_k^j)+\varepsilon}, \quad RD(X_k^j) = \frac{1}{n}\sum_{i=1}^{n}\frac{|E_{x_i}^j - \overline{X_k^j}|}{|\overline{X_k^j}|+\varepsilon} \tag{7}$$

where ε is a very small numeric constant for preventing the denominator from being zero, e.g., $1e-6$. $\overline{X_k^j}$ is the average value of the elements in X_k^j, and $E_{x_i}^j$ is the j-th element in E_{x_i}. The $|\cdot|$ is a sign of computing the absolute value.

According to the insight two, we estimate the importance of a dimension by using the element distribution divergence between the clicked hotels and the unclicked hotels. We estimate the element distribution divergence as follows:

$$W_{Y\to X}(X_k^j) = \sum_{i=1}^{n}\sum_{z=1}^{m}\frac{|E_{x_i}^j - E_{y_z}^j|}{n*m} \tag{8}$$

where $E_{y_z}^j$ is the j-th dimension of E_{y_z}; n and m are the sizes of X_k and Y_k.

According to the insight three, we estimate the importance of each embedding element to the interest focus. To estimate the similarity between $E_{x_i}^j$ and X_k^j, we firstly compute the difference between $E_{x_i}^j$ and X_k^j, and then use the reciprocal of the difference as the similarity. We combine the similarity and difference to estimate the importance of $E_{x_i}^j$, and formulate the importance as follows:

$$W_{XY\to X}(E_{x_i}^j)=\frac{\sum_{z=1}^{m}|E_{x_i}^j-E_{y_z}^j|/m}{|E_{x_i}^j-\overline{X_k^j}|+\varepsilon} \tag{9}$$

According to the importance $W_{X\to X}(X_k^j)$ and $W_{Y\to X}(X_k^j)$ of the dimension j, and element importance $W_{XY\to X}(E_{x_i}^j)$, we rewrite f_1 in Equation 6 as follows:

$$v(x_i)=f_1(q_k\in Q,X_k\in S_k,Y_k\in S_k,x_i\in X_k,E)\circ E_{x_i}=\begin{bmatrix}w(E_{x_i}^1)\\ \cdots\\ w(E_{x_i}^n)\end{bmatrix}\circ\begin{bmatrix}E_{x_i}^1\\ \cdots\\ E_{x_i}^n\end{bmatrix} \tag{10}$$

$$\begin{aligned}w(E_{x_i}^j)=\sigma_1(w_1[&W_{Y\to X}(X_k^j)\oplus W_{X\to X}(X_k^j)\oplus W_{XY\to X}(E_{x_i}^j)\\ &\oplus E_{x_i}^j\oplus\overline{X_k^j}\oplus E_{q(x_i)}]+b_1)\end{aligned} \tag{11}$$

where σ_1 is a $64\times32\times1$ MLP with activation function ReLU [27]. The w_1 and b_1 are the weight matrix and the bias vector, respectively. The $q(x_i)$ is the corresponding query of x_i, and $E_{q(x_i)}$ is the embedding of $q(x_i)$. To construct $E_{q(x_i)}$, we concatenate the embeddings of all components in $q(x_i)$.

4.2 Search-external Calibration

To bridge interest gap, we propose an interest smoothing principle: the interests learnt from historical clicks + smoothing factor $\approx$ current interests. To implement it, we develop the interest smoothing network and formulate it as follows:

$$R(u)=R_X+\lambda R_q \tag{12}$$

where R_X is the interests learnt from historical clicks X and R_q is the smoothing factor. We design the interest domination unit to construct R_X and interest smoothing unit to construct R_q. The λ is a learnable weight.

Interest domination unit. We hope the important clicks to dominate the user's current interest. To estimate the importance, we argue 1) if a click is more relevant to the current query, it is more important for representing the current interest; 2) if a click is more relevant to the whole click sequence, it is more important. Therefore, we derive the current interest representation from the click sequence X and current query q as follows:

$$R_X(u,q,X)=\sum_{x_i\in X}r(q,x_i)r(X,x_i)v(x_i) \tag{13}$$

where $r(q,x_i)$ is the relevance of x_i to q and $r(X,x_i)$ is the relevance of x_i to X.

To estimate $r(X,x_i)$, there are three intuitions: 1) if x_i appears in X more frequently, x_i is more relevant to X; 2) if x_i is more relevant to the other clicks in X, x_i is more relevant to X; 3) if the relevant clicks of x_i appear in X more frequently, x_i is more relevant to X. Accordingly, we estimate $r(X,x_i)$ as follows:

$$r(X,x_i)=\sum_{x_j\in X}n(x_i,X)n(x_j,X)r(x_j,x_i) \tag{14}$$

where $n(x_i, X)$ is the number of x_i appearing in X. The $r(x_j, x_i)$ is the relevance of x_j to x_i, and we estimate it as follows:

$$r(x_j, x_i) = \sigma_2(w_2[v(x_j) \oplus v(x_i) \oplus |v(x_j) - v(x_i)|] + b_2) \tag{15}$$

where σ_2 is a three-layer MLP with same activation functions of σ_1 in Equ. 11.

To estimate $r(q, x_i)$, we propose two approaches—click extension and query extension. The first approach uses the corresponding historical query $q(x_i)$ of x_i to extend the semantics of x_i, and then uses the relevance between q and $q(x_i)$ to estimate $r(q, x_i)$. The second approach extends the query q by using top-k relevant hotels T_q of q, and uses the relevance between T_q and x_i to estimate $r(q, x_i)$. Because our model is in the rerank stage and thus can take the candidates sorted by rank stage as input, we can use the top-k candidates as the relevant hotels of q (T_q) (default value of k is 20). We estimate $r(q, x_i)$ as follows:

$$r(q, x_i) = r(T_q, x_i) r(q, q(x_i)) \tag{16}$$

where $r(q, q(x_i))$ is the relevance between the current query q and the historical query $q(x_i)$. We estimate it as $\sigma_3(w_3[E_q \oplus E_{q(x_i)} \oplus |E_q - E_{q(x_i)}|] + b_3)$, where σ_3 is a three-layer MLP with the same activation functions of σ_1 in Equation 11. Similar to Equation 14, we estimate $p(T_q|x_i)$ as follows:

$$r(T_q, x_i) = \sum_{t_j \in T_q} n(t_j, T_q) p(t_j|x_i) \tag{17}$$

where $r(t_j, x_i)$ is estimated by using Equation 15.

Interest smoothing unit. We construct the smoothing factor R_q from two perspectives. On the one hand, the current query contains multiple important information, such as the check-in location, number of guests and check-in/checkout dates. These information can represent user's current interests to same extent. For example, if the number of guests is two and the check-in date is Valentine's day, the user' s interests in room type may be a double room, and that may be a single room when the number of guests is only one. On the other hand, we can use the rank stage to retrieve relevant hotels T_q of the current query q (interest modelling is in the rerank stage), and these hotels represent user's current interests to same extent. So we formulate R_q as follows:

$$R_q(u, q, X) = E_q \oplus \sum_{t_i \in T_q} r(T_q, t_i) r(X, t_i) E_{t_i} \tag{18}$$

where E_q is the embedding of the current query q. We map the query q to attribute values by text matching and concatenate the embeddings of the mapped values as E_q. The $r(T_q, t_i)$ is the relevance of t_i to T_q and estimated by Equation 17. The $r(X, t_i)$ is the relevance of t_i to X and estimated by Equation 14.

To learn a good value for λ in Equation 12, there are two intuitions: 1) If R_q is more relevant to R_X, it is more likely to represent user's current interest and thus λ should be larger; 2) If X contains more clicks, i.e., the click sequence is longer, R_X is more likely to represent user's current interest and thus λ should be smaller. This is because the longer click sequence can reflect more historical interests, and the more historical interests are more likely to cover the current interest. Accordingly, we design a function to automatically estimate λ as follows:

$$\lambda = \sigma_4(w_4[R_X \oplus R_q \oplus |R_X - R_q| \oplus n(X)] + b_4) \tag{19}$$

where $n(X)$ is the size of click sequence X. The σ_4 is a three-layer MLP with the same activation functions of σ_1 in Equation 11.

5 Experiments

5.1 Experiment Setup

Comparison Solutions. Since interest modelling in hotel search has not been explored, we select SOTA solutions in product recommendation as baselines:
• Base. The user and hotel profile features are applied to the CTR and CTCVR prediction tasks, and any online solutions of interest modelling are not applied.
• DNN. It encodes all clicks by using a MLP with three layers, and averages the representations of all clicks as the user's interests.
• DIN [13]. It first estimates the relevance between each click and the target, and then uses the relevance to sum the representations of all clicks.
• SIM [18]. It searches the important clicks and then uses DIN [13] to model the user's interests over the important clicks.
• DIEN [8]. It uses an interest extractor layer to capture temporal interests and an interest evolving layer to capture the interest evolving process.
• SASRec [6]. It uses the attention mechanism to capture long-term semantic and estimate the relevance of each historical click.
• DFN [16]. It uses an internal feedback and external feedback components to capture the fine-grained interactions among behaviors.
• FMLP [17]. It is a MLP model with learnable filters and designed for reducing the effect of noise clicks.
Datasets.
• *Indu*. The dataset is sampled from the user behavior logs in a major hotel e-commerce platform [5]. According to the method of sampling [21], we take four weeks' samples as the training data from November 1, 2021 to November 28, 2021, and use the following two days' samples as the evaluation data and testing data, respectively. The format is introduced in Section 3.1. The dataset contains approximately 5.5 million random users and 1.3 million hotels. It includes $2,103,945,115$ negative examples, $136,225,279$ positive examples with click behaviors and $85,665,472$ positive examples with order behaviors. The maximum length of the behavior sequences is 500 and minimum length is 0. On average, a search contains 3.5 clicks and and 9.7 unclicks; A user has 32.3 clicks and 110.6 unclicks. Note that every user is equipped with a click sequence of six months.
• *Fliggy*[6] (public dataset). According to the survey about datasets of hotel search

[5] The real name of hotel e-commerce platform is not shown because of double-blind review policy.
[6] https://tianchi.aliyun.com/dataset/dataDetail?dataId=113649

in Section 2, there are no public datasets including the search-based click sequences and query sequences. To further verify the effectiveness of our model, we adapt our model to fit the public dataset *Fliggy*. Specifically, only the interest smoothing network is applied because the dataset does not contains search information; In Equation 13, we only use $r(X, x_i)$ in Equation 14 to estimate the weight of $v(x_i)$; In Equation 18, we derive the smoothing factor R_q only from the candidate d, i.e., $R_q(u, q, X) = r(X, d)E_d$. The *Fliggy* dataset covers all behaviors (including clicks, favorites, adding, and purchases) of approximately 2 million random users from June 3, 2019 to June 3, 2021. It is the largest recommender system dataset from real industrial scenes, and contains user historical behavior data of users and basic attribute data of hotels.

Evaluation Metrics. For the offline experiments, we use the following metrics:

• AUC [23]. AUC is a widely used metric in IR and data mining tasks.

• QAUC. Studies [24, 13] introduce a user-centric weighted AUC, which measures the goodness of intra-user order by averaging AUC values over users. In the task of hotel search, we highlight the AUC values of queries, and adapt the user-centric weighted AUC to the query-centric weighted AUC:

$$QAUC = \frac{\sum_{i=1}^{n} num_i \cdot AUC_i}{\sum_{i=1}^{n} num_i}. \tag{20}$$

where n is the number of queries, AUC_i is AUC of query i, and num_i is the number of exposed hotels for query i.

• QNDCG [25]. NDCG is a measure of ranking quality considering both the relevance and ranked positions. QNDCG is the average NDCG of all queries.

• Step/sec. It is a metric for measuring the running efficiency of a model, i.e., the executed steps per second.

We use the CTR and CTCVR as metrics for online A/B testing. For estimating the online computational cost, we use the TP99 latency as a metric, which is the time consumption of the top 99.99% queries [26].

Reproducibility. The hyperparameters of all models are assigned some default values for fair comparisons. Specifically, the batch_size is 32, the optimizer is the Adam with $1e-6$ learning_rate, embed_optimize is the Adagrad optimizer with 0.8 learning_rate. All models are developed in Python 2.7 and Tensorflow 1.10. Tensorflow is distributed on 40 computers, and each computer has a 4-core CPU and $10G$ memory. For the online A/B testing, our model and baseline models are deployed on the same computation servers so that they have the same online conditions to respond users' queries. The statistical significance is tested against TCN by a two-tailed paired t-test, and † denotes the difference at 0.05 level.

5.2 Experimental Results

Comparison with baselines. We perform all models over the datasets, and report their achieved metrics in Table 3. The notation *Indu without profile features* denotes that the profile features of *Indu* are not applied to all models. It can be seen that 1) DIN and SIM models perform better than other baseline models, i.e., DIEN, SASRec, DFN. In hotel search, users' click sequences are

Table 3: Metric comparison. Since AUC is a number, not a list, the significance of AUC metrics is not tested. Since *Fliggy* does not contain query data, QAUC and QNDCG are not calculated for the *Fliggy* dataset.

Model	AUC	QAUC	QNDCG	AUC	QAUC	QNDCG	AUC
	Indu			*Indu without profile features*			*Fliggy*
Base	0.92373	0.83687†	0.78718†	—	—	—	—
DNN	0.92522	0.83912†	0.78855†	0.68052	0.60316†	0.59816†	0.68662
DIN	0.92559	0.83937†	0.78877†	0.69145	0.61444†	0.61284†	0.69779
SIM	0.92570	0.84019†	0.78947†	0.68995	0.61018†	0.61261†	0.70041
DIEN	0.92482	0.83817†	0.78788†	0.67976	0.59894†	0.60376†	0.68127
SASRec	0.92513	0.83823†	0.78799†	0.66862	0.58795†	0.59711†	0.69575
DFN	0.92492	0.83842†	0.78807†	0.64839	0.59296†	0.59982†	0.69377
FMLP	0.92517	0.83861†	0.78798†	0.63819	0.59765†	0.58921†	0.67851
TCN	**0.92716**	**0.84139**	**0.79059**	**0.80165**	**0.66252**	**0.64431**	**0.71723**

very sparse—the sequences are very short and the time intervals between clicks are very long. The sparsity can not provide sufficient semantics to model the relations among clicks. So the solutions of modelling the relations among clicks i.e., DIEN, SASRec and DFN, can not make full use of their advantages on the sparse sequences; 2) TCN outperforms all baselines on all metrics. Note that the metric gains achieved by TCN on the *Indu* dataset are not higher than those on dataset *Indu without profile features*. This is because *Indu* contains 700+ profile features and these features can effectively portray users' interests; And thus metric gains stemmed from interest modelling are not high on the *Indu* dataset. But on the *Indu* dataset, 0.146% AUC improvement over the best baseline SIM is very significant to the hotel business because it brings 2.17% increase in CTCVR (see online A/B test in Table 6) and an increase of about 5500 orders. Besides, the metric differences between TCN and DIN as well as TCN and SIM are much larger than those between Base and DIN as well as Base and SIM. This illustrates that the metric improvements are non-trivial to achieve. The metric improvements verify that TCN model user's interests better than baselines.

In principle, TCN first models the interest focus on each click, and then construct the user's interest representation based on the interest focuses, while baselines don't consider and model the interest focus. Besides, the interest smoothing network in TCN can effectively address the interest gap challenge, while baselines can not well address the challenge. Although DIEN uses an interest evolving layer to capture the interest evolving process, DIEN can not well address the interest gap, because in hotel domain, 1) users' click sequences are so short that they can not provide sufficient statistics for interest evolution learning; 2) the time intervals between clicks are so long that they severely weaken the relationships among clicks and hinder the modelling of interest evolution. The experimental results in Table 3 illustrates that TCN can address the interest focus and interest gap challenges more effectively than baseline models.

Ablation study. We conduct an ablation study to test the effectiveness of components in TCN and show the results in Table 4. The notation *Indu without profile features* denotes that the profile features are not applied to these models. The notation TCN-DCN, TCN-ISN, TCN-IDU and TCN-ISU denote that the divergence-based calibration network, the interest smoothing network, the

Table 4: Effectiveness of components in TCN. "-DCN", "-ISU", "-IDU" and "-ISU" respectively denote that divergence-based calibration network, interest smoothing network, interest domination unit and interest smoothing unit are not applied.

Model	AUC	QAUC	QNDCG	AUC	QAUC	QNDCG
	Indu			*Indu without profile features*		
TCN-DCN	0.92635	0.83968	0.78913	0.78395	0.63712	0.62696
TCN-ISN	0.92619	0.83952	0.78906	0.68994	0.60547	0.59930
TCN-IDU	0.92683	0.84053	0.78965	0.74216	0.62953	0.61845
TCN-ISU	0.92695	0.84091	0.78981	0.75674	0.64517	0.62759
TCN	**0.92716**	**0.84139**	**0.79059**	**0.80165**	**0.66252**	**0.64431**

Table 5: Improving the baselines by using components of TCN. "+DCN" and "+ISU" respectively denote that the divergence-based calibration network and interest smoothing unit are applied to improve baselines.

Model	AUC	QAUC	QNDCG	Model	AUC	QAUC	QNDCG
DIN	0.69145	0.61444	0.61284	SASRec	0.66862	0.58795	0.59711
DIN+DCN	0.69532	0.61647	0.61433	SASRec+DCN	0.67922	0.59672	0.60490
DIN+ISU	0.72409	0.62344	0.61700	SASRec+ISU	0.70374	0.60322	0.60902
SIM	0.68995	0.61018	0.61261	DFN	0.64839	0.59296	0.59982
SIM+DCN	0.69451	0.61602	0.61507	DFN+DCN	0.65702	0.59706	0.60229
SIM+ISU	0.72143	0.62476	0.62204	DFN+ISU	0.69844	0.60790	0.60702
DIEN	0.67976	0.59894	0.60376	FMLP	0.63819	0.59765	0.58921
DIEN+DCN	0.68354	0.60233	0.60741	FMLP+DCN	0.64707	0.60508	0.60381
DIEN+ISU	0.71290	0.61544	0.61398	FMLP+ISU	0.68703	0.60821	0.60603
TCN	**0.80165**	**0.66252**	**0.64431**	—	—	—	—

interest domination unit, and the interest smoothing unit are not applied to TCN, respectively. In TCN-ISN, we average the embedded representations of the interest focuses of all clicks as the final user's interest representation. From Table 4, we can see that 1) the metrics achieved by TCN-DCN is lower than that achieved by TCN. This illustrates those the divergence-based calibration network is important to interest modelling, and verifies the effectiveness of the divergence-based calibration network; 2) The metrics achieved by TCN-ISN are lower than those achieved by TCN, which verifies the importance of the interest smoothing network. As well, it demonstrates that the interest smoothing network can well bridge the interest gap; 3) The metrics achieved by TCN-DCN are higher than that achieved by TCN-ISN, which illustrates that the interest smoothing network is more important to interest modelling than the divergence-based calibration network; 4) TCN achieves the better metrics than TCN-IDU and TCN-ISU. This directly illustrates that not using any one of TCN-IDU and TCN-ISU results in a decrease in metrics, and then verifies the effectiveness of the interest domination unit and interest smoothing unit. The above ablation studies verify the rationality of TCN and the effectiveness of components.

Improving baselines. We use the divergence-based calibration network and the interest smoothing unit to improve the baselines, and denote the improved baselines by +DCN and +ISU, respectively. Specifically, we first use the divergence-based calibration network to model the interest focus on each click, and then take the interest focuses as the input of baseline models; In Equation 12, we use the interest representations learned by baselines to replace R_X. The experimental results are presented in Table 5. It can be seen that the improved baselines

Table 6: Online A/B test (relative improvements).

Metric	DNN	SIM	TCN
CTCVR	0.00%†	+0.73%†	**+2.90%**
CTR	0.00%†	+0.36%†	**+1.53%**

Table 7: Comparisons of computational cost.

Metric	DNN	DIN	SIM	DIEN	SASRec	DFN	FMLP	TCN
Step/sec	55	53	48	25	43	38	41	45
TP99	37 ms	—	41 ms	—	—	—	—	42 ms

with notation +DCN achieve better metrics than the original baselines. This illustrates that the divergence-based calibration network is important to interest modelling and can effectively capture the interest focus. Besides, the improved baselines with notation +ISU perform better than the original baseline, which verifies that the interest smoothing unit is effective to address the interest gap challenge. The metric improvements illustrate that the components in TCN can improve baseline models.

Online A/B test. We perform the A/B tests and show the results in Table 6. Considering the computational cost and economic benefits, we select DNN, SIM and TCN to conduct the A/B tests for two weeks. Each model is assigned with 20% traffic. It can be seen that TCN achieves the best online CTCVR and CTR metrics. Compared to DNN, TCN achieves the +2.91% CTCVR improvement and +1.53% CTR improvement. These metric improvements are much higher than those achieved by SIM, and significant to hotel business.

Comparisons of computational cost. We show the computational of all models in Table 7. It can be seen that 1) the Step/sec of DNN is higher than that of other models, and the TP99 of DNN is lower than that of other models. This illustrates that the training cost and online predicting cost of DNN are smaller than those of other models; 2) the computational cost of TCN is slightly higher than that of DNN and SIM. By A/B testing, the slight cost gain does not significantly decrease the efficiency of the online hotel search system.

6 Conclusion

We study the interest modelling task for reranking hotels, and identify two new challenges, i.e., interest focus challenge and interest gap challenge. To capture interest focus, we propose three insights based on search data, and develop a divergence-based calibration network. To bridge the interest gap, we propose an interest smoothing principle, and then develop an interest smoothing network by reusing the query data. By experiments, we find that 1) modeling the interest focus on each click is important for constructing the user's interest representation over all clicks; 2) About addressing the interest gap challenge, the combination of the interest domination unit and the interest smoothing unit performs better than any single one; 3) Users' click sequences are very sparse in the hotel domain, the dependency relationships among clicks are very weak, and thus the solutions of modeling the relationships can not work well.

References

1. Diego Antognini and et al.: HotelRec: a Novel Very Large-Scale Hotel Recommendation Dataset, LREC, 2020, 4917–4923
2. Haldar, Malay and et al.: Improving deep learning for airbnb search, SIGKDD, 2020, 2822–2830
3. Mustafa Abdool and et al.: Managing Diversity in Airbnb Search, SIGKDD, 2020, 2952–2960
4. Chao Li and et al.: Multi-Interest Network with Dynamic Routing for Recommendation at Tmall, CIKM, 2019, 2615–2623
5. Massimo Quadrana and et al.: Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks, RecSys, 2017, 130–137
6. Fei Sun and et al.: BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer, CIKM, 2019, 1441–1450
7. Jens Adamczak and et al.: Session-based Hotel Recommendations Dataset, Trans. Intell. Syst. Technol., 2021 ,1:1–1:20
8. Guorui Zhou and et al.: Deep Interest Evolution Network for Click-Through Rate Prediction, AAAI, 2019, 5941–5948
9. Zhai, Chengxiang and et al.: A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval, SIGIR, 2001, 334–342
10. Kai Zhang and et al.: Hotel recommendation based on user preference analysis, ICDE, 2015,134–138
11. Bin Liu and et al.: Learning geographical preferences for point-of-interest recommendation, SIGKDD, 2013, 1043–1051
12. Vittoria Cozza and et al.: Mining Implicit Data Association from Tripadvisor Hotel Reviews, EDBT, 2018, 56–61
13. Guorui Zhou and et al.: Deep Interest Network for Click-Through Rate Prediction, SIGKDD, 2018, 1059–1068
14. Kan Ren and et al.: Lifelong Sequential Modeling with Personalized Memorization for User Response Prediction, SIGIR, 2019, 565–574
15. Zeping Yu and et al.: Adaptive User Modeling with Long and Short-Term Preferences for Personalized Recommendation, IJCAI, 2019, 4213–4219
16. Ruobing Xie and et al.: Deep Feedback Network for Recommendation, IJCAI, 2020, 2519–2525
17. Kun Zhou and et al.: Filter-enhanced MLP is All You Need for Sequential Recommendation, WWW, 2022, 2388–2399
18. Qi Pi and et al.: Search-based User Interest Modeling with Lifelong Sequential Behavior Data for Click-Through Rate Prediction, CIKM, 2020, 2685–2692
19. Tianqi Chen and et al.: XGBoost: A Scalable Tree Boosting System, SIGKDD, 2016, 785–794
20. Zhilu Zhang and et al.: Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels, NeurIPS, 2018, 8792–8802
21. Tianqi Chen and et al.: MIND: A Large-scale Dataset for News Recommendation, ACL, 2020.
22. Jiaqi Ma and et al.: Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts, SIGKDD, 2020, 1930–1939
23. Tom Fawcett: An introduction to ROC analysis, Pattern Recognit. Lett., 2006, 861–874
24. Han Zhu and et al.: Optimized Cost per Click in Taobao Display Advertising, SIGKDD, 2017, 2191–2200

25. Yining Wang and et al.: A Theoretical Analysis of NDCG Type Ranking Measures, The 26th Annual Conference on Learning Theory, 2017, 25–54
26. Hu Liu and et al.: Kalman Filtering Attention for User Behavior Modeling in CTR Prediction, Advances in Neural Information Processing Systems, 2020
27. Djork-Arné Clevert and et al.: Fast and Accurate Deep Network Learning by Exponential Linear Units, ICLR, 2016
28. Sudhir Kumar Sharma and et al.: An Adaptive Sigmoidal Activation Function Cascading Neural Networks, Soft Computing Models in Industrial and Environmental Applications, 2011, 105–116