ICMR 2024

The 14th International Conference on Multimedia Retrieval

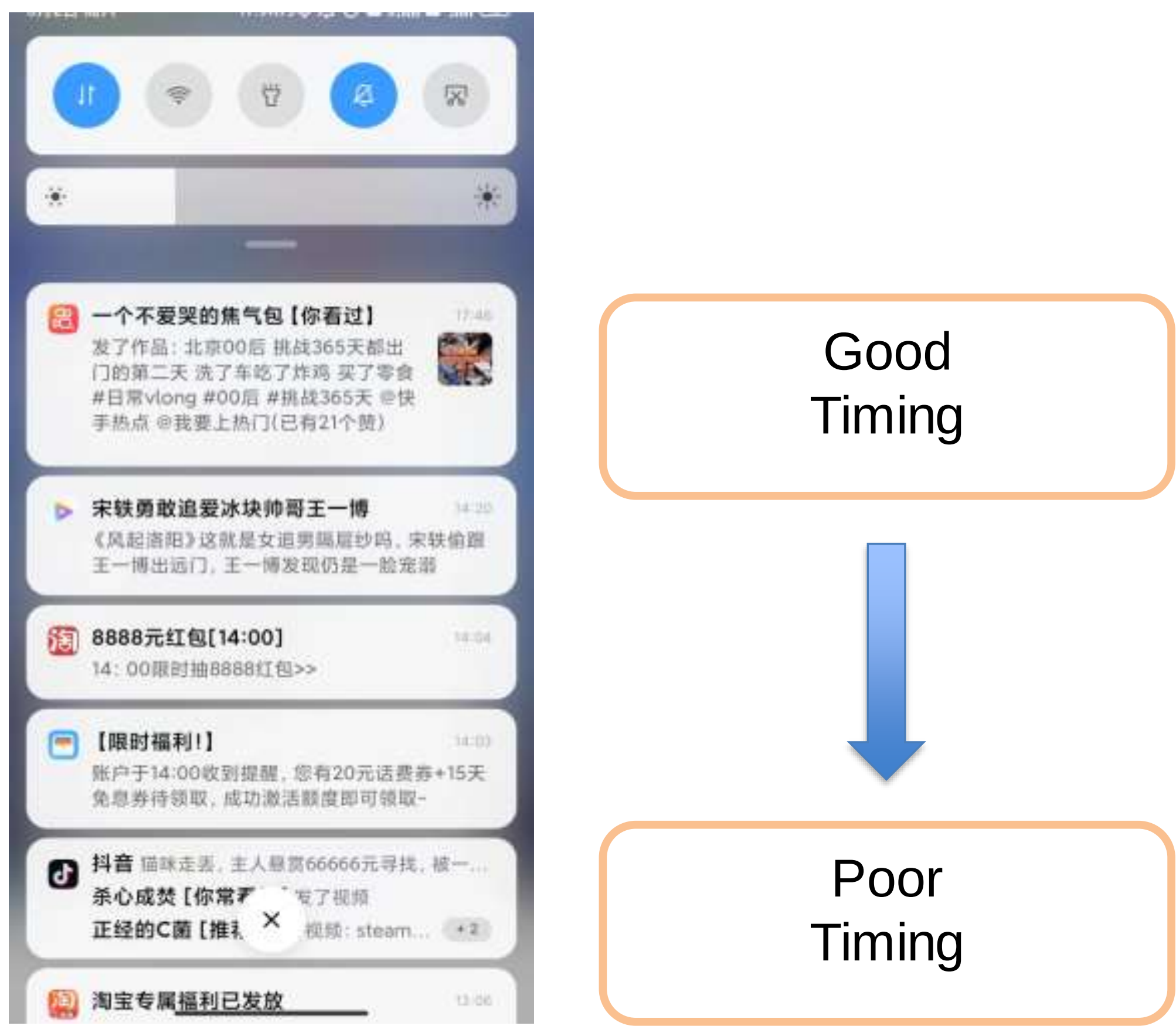
TIM: Temporal Interaction Model in Notification System

Huxiao Ji [1], Linchuan Li [1], Haitao Yang [1] , Shunyu Zhang [1], Cunyi Zhang [1]
Xuanping Li [*], Wenwu Ou [2]
1. Kuaishou Technology Co., Ltd. Beijing 100085, China
2. unaffiliated China
Corresponding author: lixuanping@kuaishou.com

Introduction

Different from traditional recommendation systems, notification recommendation requires active selection of the timing for pushing. If the delivery of notifications is poorly timed, users may miss these alerts on their phones. In contrast, if multiple notifications are sent continuously while users are using their phones, they may feel disturbed and consequently disable the notification function.

To model the engagement habits of users, we propose the Temporal Interaction Model (TIM), which estimates users' slot-wise CTR over a day. Moreover, we put forward the first strategy that optimizes the timing of multiple notifications in a day, with the ultimate goal of maximizing the notification click volume of Kuaishou.


一个不爱哭的焦气包【你看过】 17:46
发了作品：北京00后 挑战365天都出门的第二天 洗了车吃了炸鸡 买了零食 #日常vlong #00后 #挑战365天 @快手热点 @我要上热门(已有21个赞)

宋轶勇敢追爱冰块帅哥王一博 14:20
《风起洛阳》这就是女追男隔层纱吗，宋轶偷跟王一博出远门，王一博发现仍是一脸宠溺

8888元红包[14:00] 14:04
14：00限时抽8888红包>>

【限时福利！】 14:03
账户于14:00收到提醒，您有20元话费券+15天免息券待领取，成功激活额度即可领取~

抖音 猫咪走丢，主人悬赏66666元寻找，被一…
杀心成焚 [你常看 … 发了视频
正经的C菌 [推荐 … 视频：steam… +2

淘宝专属福利已发放 13:06

Fig.1 Comparison on good timing and poor timing on notifications

Temporal Interaction Model

We collected various interaction features of users during K time slots in the past L days, including notification receipts, clicks, active duration, video browsing counts, etc. The feature vectors forms a $K \times L \times d_1$ tensor $X = (X_1, X_2, \dots, X_K)$. We introduce the Temporal Attention Unit (TAU) to extract user behavior pattern embeddings from these interaction features. The fully connected network that follows integrates user contextual features and behavior pattern embeddings.

TAU consists of two attention layers. The first layer is a target attention layer with date type embedding (DTE), which can alleviate the difference in user behavior habits between work days and rest days. This layer outputs one embedding for each of the K slots by extracting user interaction features of the same time slot in previous L days. The second attention layer is a linear self-attention layer with rotary position embedding (RoPE), in which the embeddings of each slot fuses with each other.

Finally, for each slot k, the output of TAU is concatenated with users' contextual embeddings, and fed into the fully connected network to output the predicted CTR p_k.

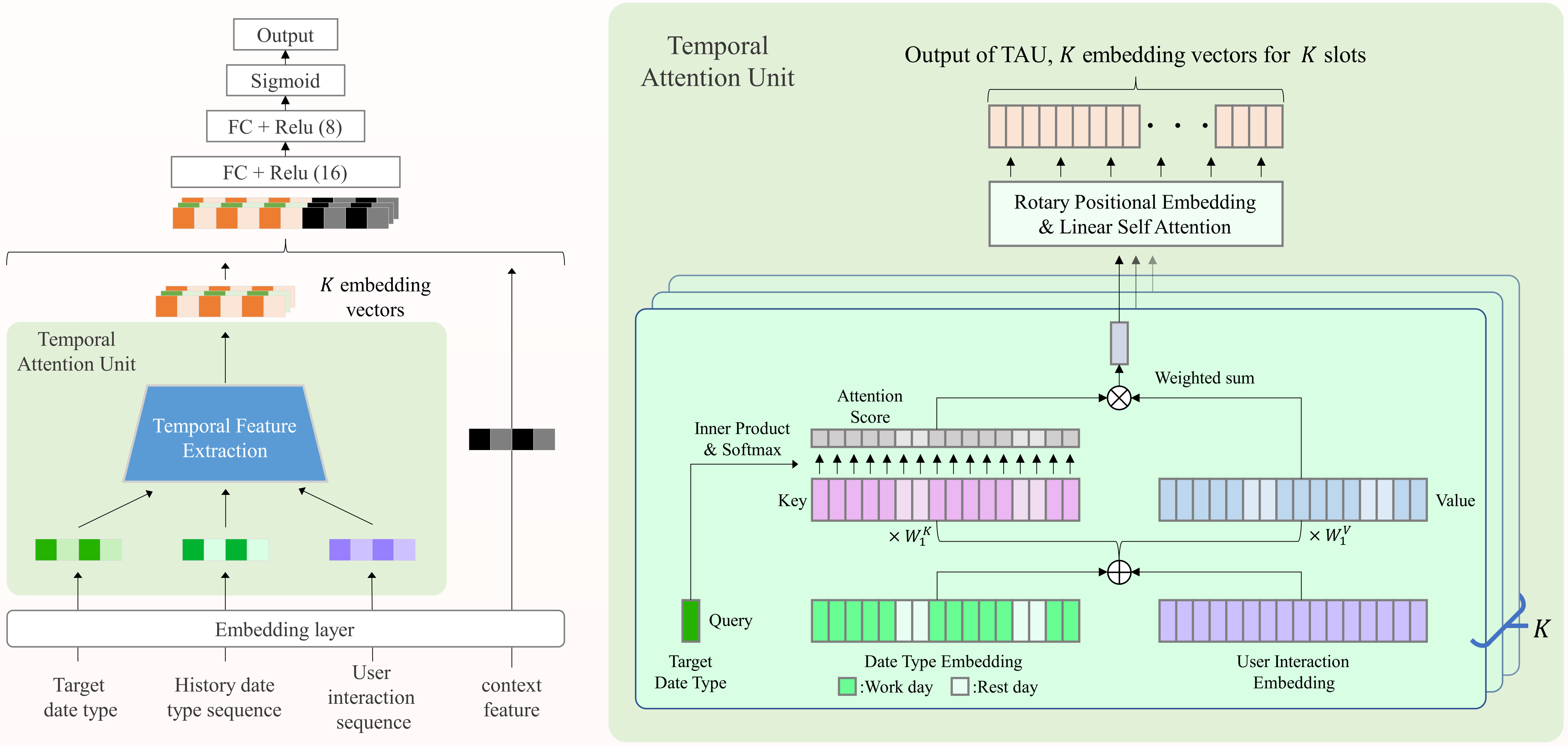


Fig.2 Overview of TIM, with TAU as the central structure for extracting user behavioral patterns. TAU extracts the feature of the target date from the day dimension and then fuses the features in the slot dimension.

Maximizing Click Count

With the predicted slot-wise CTR of TIM, it remains challenging to coordinate these results in practical implementation. In the real-world scenario of Kuaishou, a user is allocated a given quota of notifications per day. We model the relationship between slot-wise CTR and the business goal of maximizing user's overall click count in one day by

$$\begin{aligned} \underset{\boldsymbol{n}}{\text{minimize}} \quad & -\boldsymbol{p}^{\top}\boldsymbol{n} + \lambda\|\boldsymbol{n}\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{n}\|_1 = q \\ & n_i \geq 0 \qquad i = 1,2,\dots,K \end{aligned}$$

where $\boldsymbol{n} = (n_1, n_2, \dots, n_K)$ represents the expected numbers of notifications sent in each slot, $\boldsymbol{p} = (p_1, p_2, \dots, p_K)$ is the vector of predicted slot-wise CTR and q denotes the quota of notifications assigned to the user. Solving the problem, we obtain that

$$n_i = \begin{cases} \frac{1}{2\lambda}(p_i - \varphi), & \mu_i = 0 \\ 0, & \mu_i \neq 0 \end{cases}$$

Note that while $\boldsymbol{\mu}_i = \boldsymbol{0}$ and $\lambda = \frac{\|\boldsymbol{p}\|_1}{2q}$, we have $\|\boldsymbol{n}\|_1 = \frac{\|\boldsymbol{p}\|_1 - K\varphi}{2\lambda} = \frac{q\|\boldsymbol{p}\|_1 - Kq\varphi}{\|\boldsymbol{p}\|_1} = q$, implying that φ=0 and $n_i \propto p_i$. With such observation, we are inspired to set $n_i = q \cdot \frac{p_i}{\|\boldsymbol{p}\|_1}$.

To implement the result, we propose to control the sending with cumulative CTR. Assuming that slot s begins at t_{s-1} and ends at t_s, and the CTR is consistent throughout the slot. The expected number of notifications sent before time $t_{s-1} + \Delta_t (0 \leq \Delta_t < T)$ is

$$\mathcal{P}(t_{s-1} + \Delta_t) = \left(\sum_{i=0}^{s-1} p_i + \frac{\Delta_t}{T} \cdot\ p_s\right) \cdot \frac{q}{\|\boldsymbol{p}\|_1}$$

When a trigger occurs at time $t_{s-1} + \Delta_t$, the probability to send a notification is

$$min(\mathcal{P}(t_{s-1} + \Delta_t) - \mathcal{H}(t_{s-1} + \Delta_t), 1)$$

where $\mathcal{H}\ (T)$ represents the notification count sent before time T.

Results

For offline experiments, We randomly select 1 million users' notification data and features during a week as the training dataset and use the data from the same set of users on the next day as the testing dataset. We sampled two datasets according to whether the target date is a work day or a rest day. We compute the area under the receiver operating characteristic curve (AUC), top-k hit ratio (HR@k) and top-k Accuracy (A@k) on the testing dataset.

Method	AUC	HR@1	HR@5	HR@9	A@1	A@5	A@9
Work day							
XGB	0.5907	0.0201	0.113	0.22	0.181	0.204	0.221
XGB$_{int}$	0.6673	0.0207	0.118	0.227	0.187	0.212	0.227
MLP	0.7704	0.0318	0.159	0.312	0.287	0.286	0.281
MLP$_{int}$	0.8485	0.0391	0.183	0.347	0.352	0.33	0.313
TIM	**0.8614**	**0.0403**	**0.188**	**0.356**	**0.363**	**0.339**	**0.32**
Rest day							
XGB	0.5917	0.0196	0.109	0.216	0.188	0.208	0.229
XGB$_{int}$	0.6644	0.0201	0.114	0.224	0.192	0.218	0.238
MLP	0.7723	0.031	0.154	0.306	0.295	0.294	0.291
MLP$_{int}$	0.8398	0.0369	0.174	0.334	0.351	0.332	0.318
TIM	**0.8559**	**0.0379**	**0.179**	**0.341**	**0.36**	**0.34**	**0.325**

Tab.1 Performance comparison between TIM and baseline models..

Results show that TIM outperforms all baseline models on all metrics, both on work day and rest day, which shows the strong power of TIM in extracting user behavior patterns. Moreover, XGB$_{int}$ and MLP$_{int}$ surpass XGB and MLP respectively by a large margin, indicating the significant effect of user history interaction features.

For online A/B tests, users are randomly divided into three groups. For users in the control group, notifications are distributed evenly over time. Users in another group receive notifications based on the overall slot-wise CTR. The last group uses TIM to control the delivery.

Metrics	overall	TIM
DAU	+0.026%(0.37)	+0.100%(0.00)
send volume	-0.214%(0.00)	+0.023%(0.48)
CTR	+0.437%(0.00)	+1.217%(0.00)
watch time	+0.014%(0.73)	+0.066%(0.11)
switch close rate	+0.408%(0.53)	+0.448%(0.47)

Tab.2 TIM's business gain in online A/B test, compared with uniform delivery. In parentheses are p-values.

Results show that TIM guarantees both precise timing of notifications and business health, providing users with a strong sense of awareness.