Sparse Attention Network For Session-based Recommendation

Jiahao Yuan,[1] Zihan Song,[1] Mingyou Sun,[1] Xiaoling Wang,[1, 2*] Wayne Xin Zhao[3]

[1] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
[2] Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai, China
[3] Gaoling School of Artificial Intelligence, Renmin University of China
{jhyuan, zhsong, mysun}@stu.ecnu.edu.cn, xlwang@cs.ecnu.edu.cn, batmanfly@gmail.com

AAAI 2021
Code: https://github.com/SamHaoYuan/DSANForAAAI2021

Reported by liang li

Motivation

Details：

- One is to regard the last-click as the query vector to denote the user's current preference.
- And the other is to consider that all items within the session are favorable for the final result, including the effect of unrelated items (i.e., spurious user behaviors)

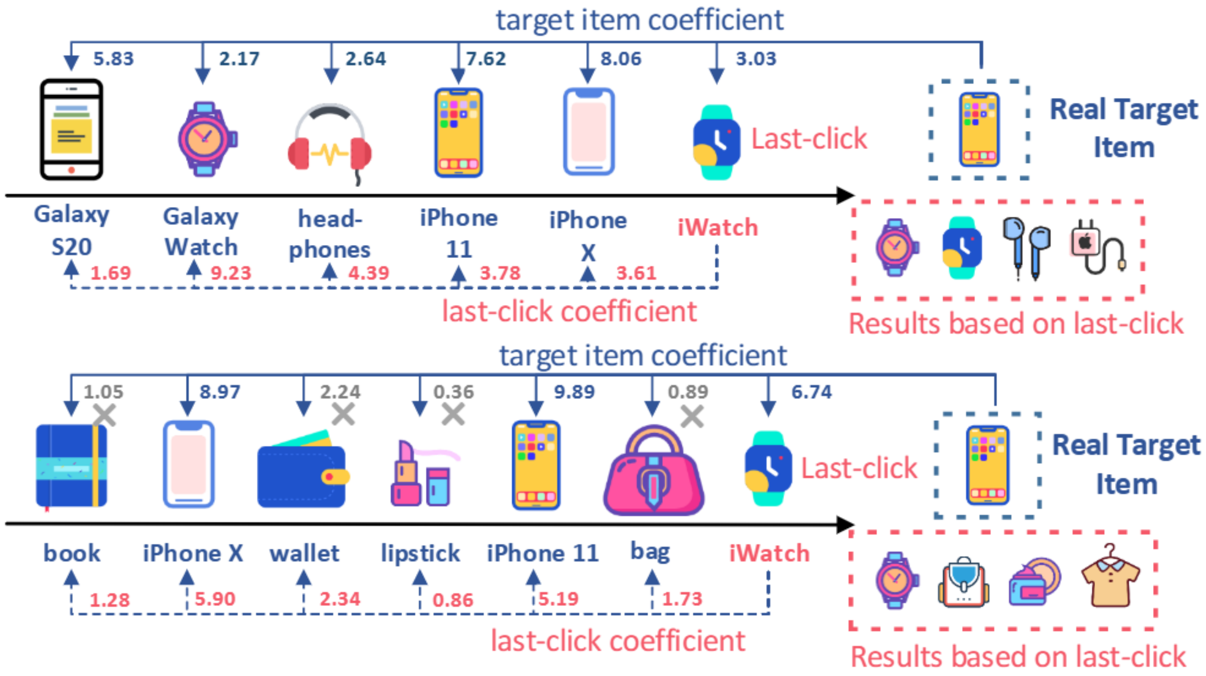

Figure 1: Motivating examples of session-based recommendation. This paper aims to directly model the real target item representation and alleviate the impact of unrelated items.

Problem Statement

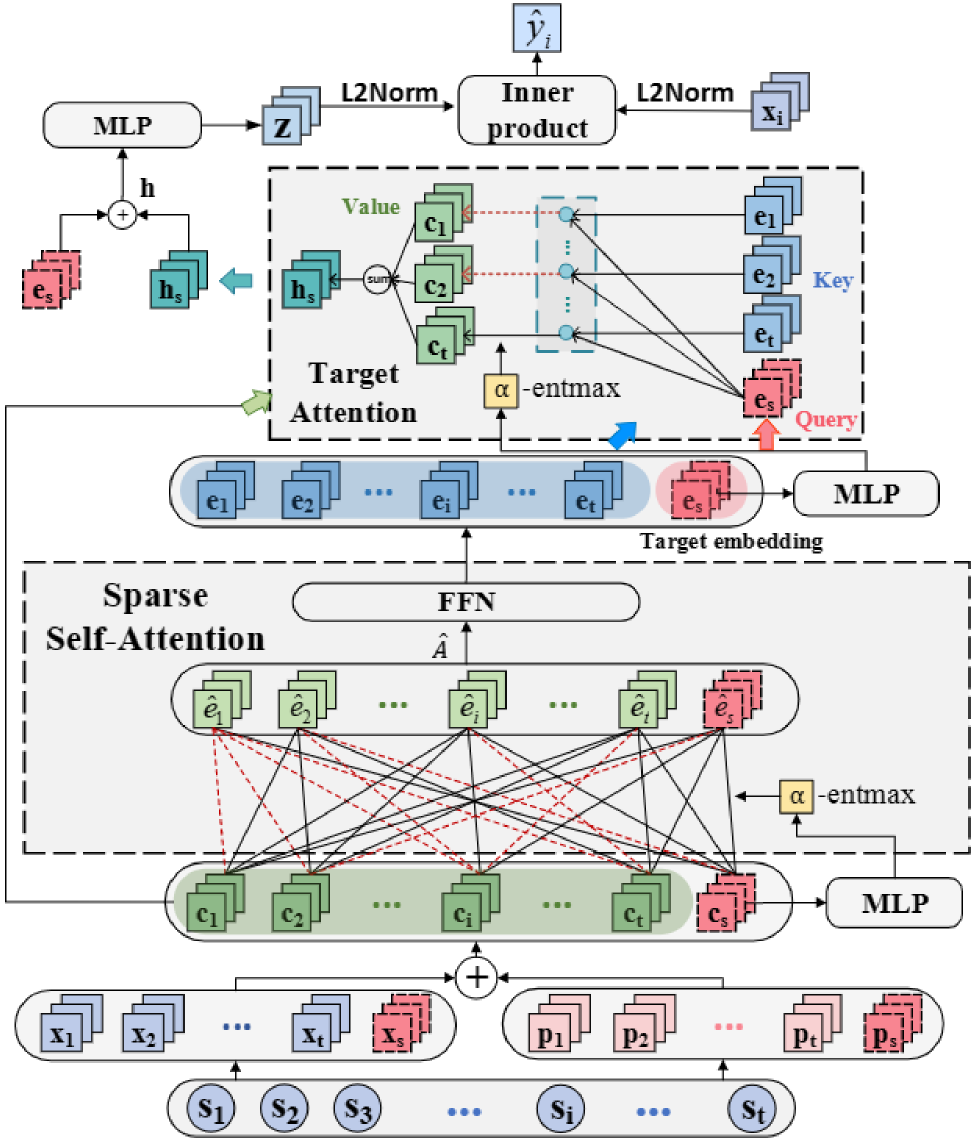

Figure 3: The general architecture of the proposed model. The red dot line indicates a possible zero weight value.

$$I = \{i_1, i_2, \dots, i_m\}$$

$$S = \{s_1, s_2, \dots, s_n\}$$

$$s_p \in I$$

$$S_t = \{s_1, s_2, \dots, s_t\} (1 < t < n)$$

Method

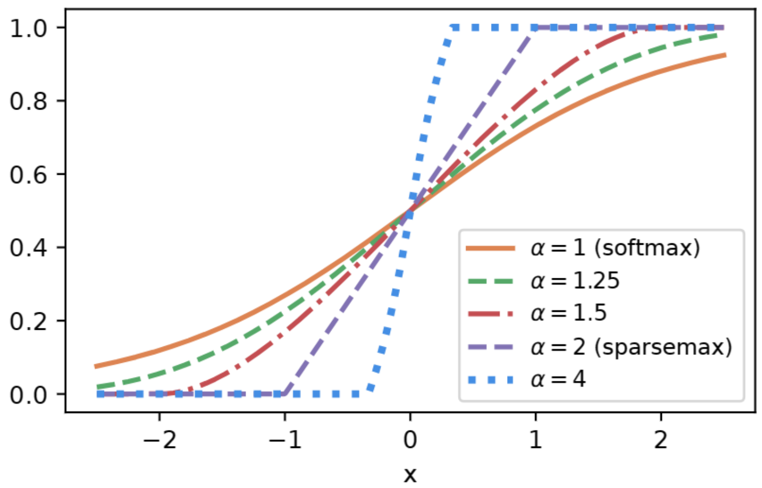

Figure 2: Illustration of α-entmax in the two-dimensional case.

$$sparsemax(x) = \underset{p \in \triangle^{d-1}}{\operatorname{argmin}} ||p - x||^2 \qquad (1)$$

$$\alpha\text{-entmax}(x) = \underset{p \in \triangle^{d-1}}{\operatorname{argmax}}\, p^T x + H_\alpha^T(p), where$$

$$H_\alpha^T(p) = \begin{cases} \frac{1}{\alpha(\alpha-1)} \sum_j (p_j - p_j^\alpha), & \alpha \neq 1 \\ H^S(p), & \alpha = 1 \end{cases} \qquad (2)$$

$$\triangle^{d-1} = \{p \in \mathbb{R}^d | 1^T p = 1, p \geq 0\}$$

where x is the input vector and p is the output vector.

	Origin	Sparse
Softmax	$p_i = \frac{e^{s_i}}{\sum_{j=1}^{n} e^{s_j}}$	$p_i = \begin{cases} \frac{e^{s_i}}{\sum_{j \in \Omega_k} e^{s_j}}, & i \in \Omega_k \\ 0, & i \notin \Omega_k \end{cases}$

其中Ω_k是将$s_1, s_2, \ldots, s_n$从大到小排列后前k个元素的下标集合。说白了，苏剑林大佬提出的Sparse Softmax就是在计算概率的时候，只保留前k个，后面的直接置零，k是人为选择的超参数

Method

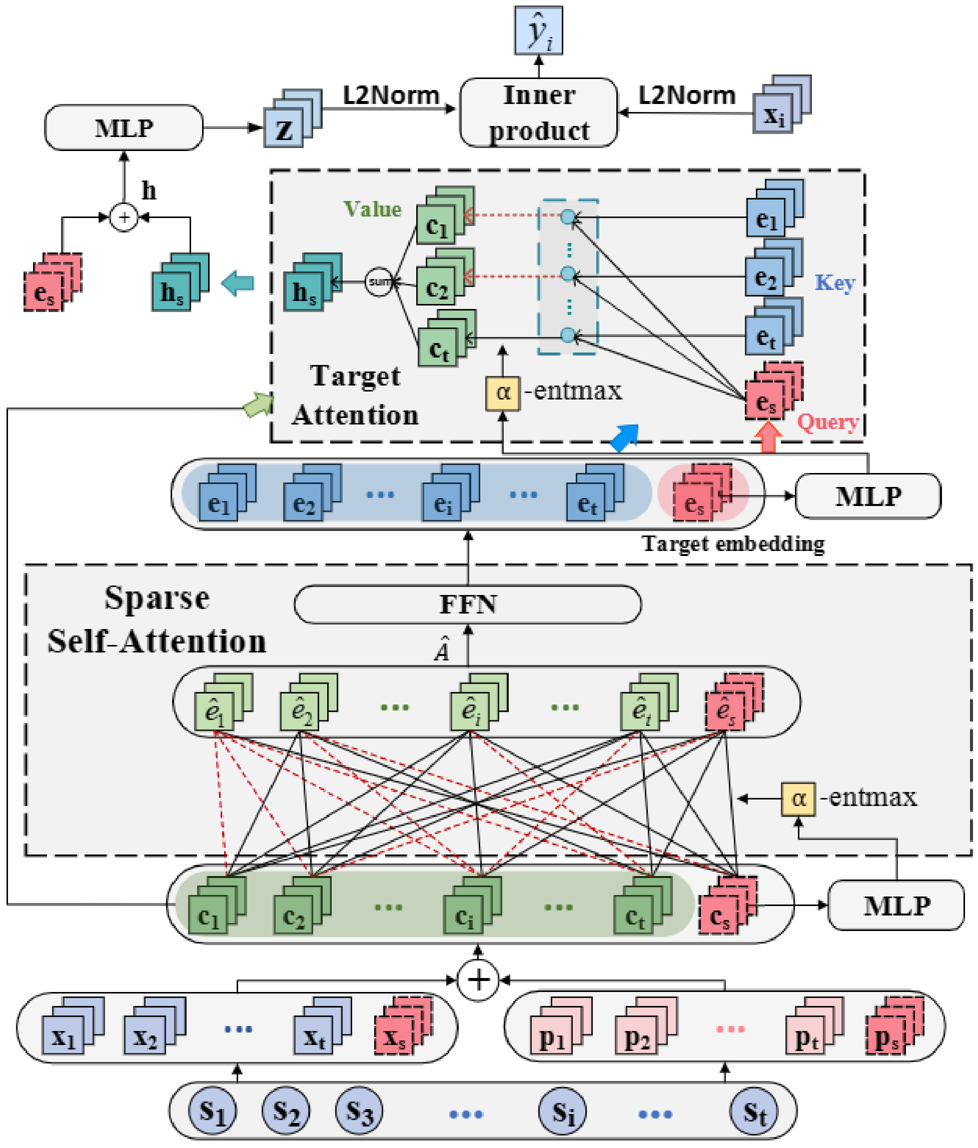

Figure 3: The general architecture of the proposed model. The red dot line indicates a possible zero weight value.

$$c_i = Concat(x_i, p_i) \tag{3}$$

$$\hat{C} = \{c_1, c_2, ..., c_t, c_s\}$$

$$\hat{A} = \alpha\text{-entmax}\left(\frac{QK^T}{\sqrt{2d}}\right)V \tag{4}$$

$$\alpha = \sigma(W_\alpha c_s + b_\alpha) + 1 \tag{5}$$

$$Q = f(\hat{C}W^Q + b^Q) \tag{6}$$

$$FFN(\hat{A}) = \max(0, \hat{A}W_1^{self} + b_1)W_2^{self} + b_2 \tag{7}$$

$$E = SAN(\hat{C}) \tag{8}$$

Method

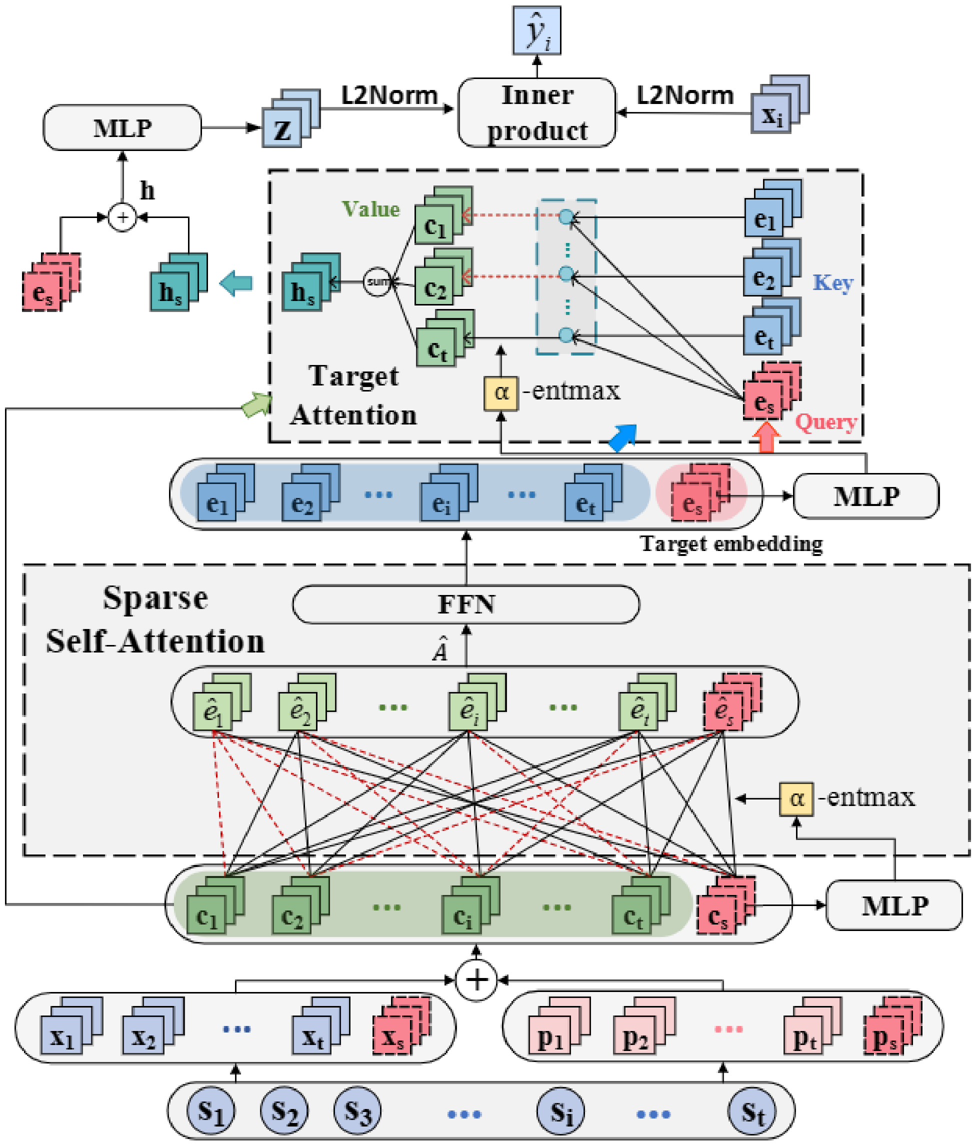

Figure 3: The general architecture of the proposed model. The red dot line indicates a possible zero weight value.

$$\beta_p = \alpha\text{-entmax}(W_0 f(W_1 e_p + W_2 e_s + b_a)) \tag{9}$$

$$\alpha = \sigma(W_\alpha e_s + b_\alpha) + 1. \tag{10}$$

$$\begin{aligned} h_s &= \textstyle\sum_{p=1} \beta_p c_p \\ h &= Concat(e_s, h_s) \\ z &= f(W_z h + b_z) \end{aligned} \tag{11}$$

$$\begin{aligned} \hat{z} = w_k L2Norm(z),\ \hat{x}_i = L2Norm(x_i) \\ \hat{y}_i = softmax(\hat{z}^T \hat{x}_i) \end{aligned} \tag{12}$$

$$L(y, \hat{y}) = -\sum_{i=1}^{m} y_i \log(\hat{y}_i) \tag{13}$$

Experiments

Datesets	# train	# test	# clicks	# items
Diginetica	526,135	44,279	858,108	40,840
Retailrocket	433,648	15,132	710,586	36,968

Table 1: Statistics of datasets used in the experiments

Experiments

Datasets	Diginetica				Retailrocket			
Metrics	HR@10	HR@20	MRR@10	MRR@20	HR@10	HR@20	MRR@10	MRR@20
S-POP	0.2389	0.2409	0.1392	0.1393	0.3578	0.3803	0.2468	0.2481
FPMC	0.1807	0.2571	0.0713	0.0765	0.2599	0.3237	0.1338	0.1382
SKNN	0.3661	0.4835	0.1561	0.1649	<u>0.4674</u>	<u>0.5428</u>	0.2593	0.2646
STAN	<u>0.3820</u>	0.4993	0.1678	0.1759	0.4656	0.5348	0.2633	0.2681
GRU4Rec	0.2617	0.3927	0.0969	0.1059	0.3835	0.4401	0.2327	0.2367
STAMP	0.3349	0.4647	0.1399	0.1489	0.4295	0.5096	0.2461	0.2517
SR-GNN	0.3772	0.5050	0.1675	0.1763	0.4321	0.5032	0.2607	0.2657
GC-SAN	0.3786	<u>0.5084</u>	<u>0.1689</u>	<u>0.1779</u>	0.4410	0.5118	<u>0.2692</u>	<u>0.2740</u>
Bert4Rec	0.3461	0.4878	0.1327	0.1425	0.4585	0.5419	0.2584	0.2642
CoSAN	0.3475	0.4834	0.1429	0.1522	0.4381	0.5247	0.2380	0.2440
DSAN	**0.4029***	**0.5376***	**0.1805***	**0.1899***	**0.4905***	**0.5654***	**0.3021***	**0.3074***
Improv.	5.47%	5.74%	6.87%	6.75%	4.75%	3.78%	10.70%	11.28%

Table 2: Performance of all recommendation models. The boldface is the best result over all methods, the underline is the best result of all baselines, and * denotes the significant difference for t-test.

Experiments

Datesets	DIGINETICA		RETAILROCKET	
Metrics	HR@20	MRR@20	HR@20	MRR@20
DSAN-NS	0.5199	0.1858	0.5322	0.3000
DSAN-NT	0.5348	0.1838	0.5646	0.3040
DSAN-DA	0.5340	0.1876	0.5600	0.3010
DSAN	**0.5376**	**0.1899**	**0.5654**	**0.3074**

Table 3: Impacts of the dual attention network.

(a) HR@20 (b) MRR@20

Figure 4: Experimental results with different transformation function on two metrics.

Experiments

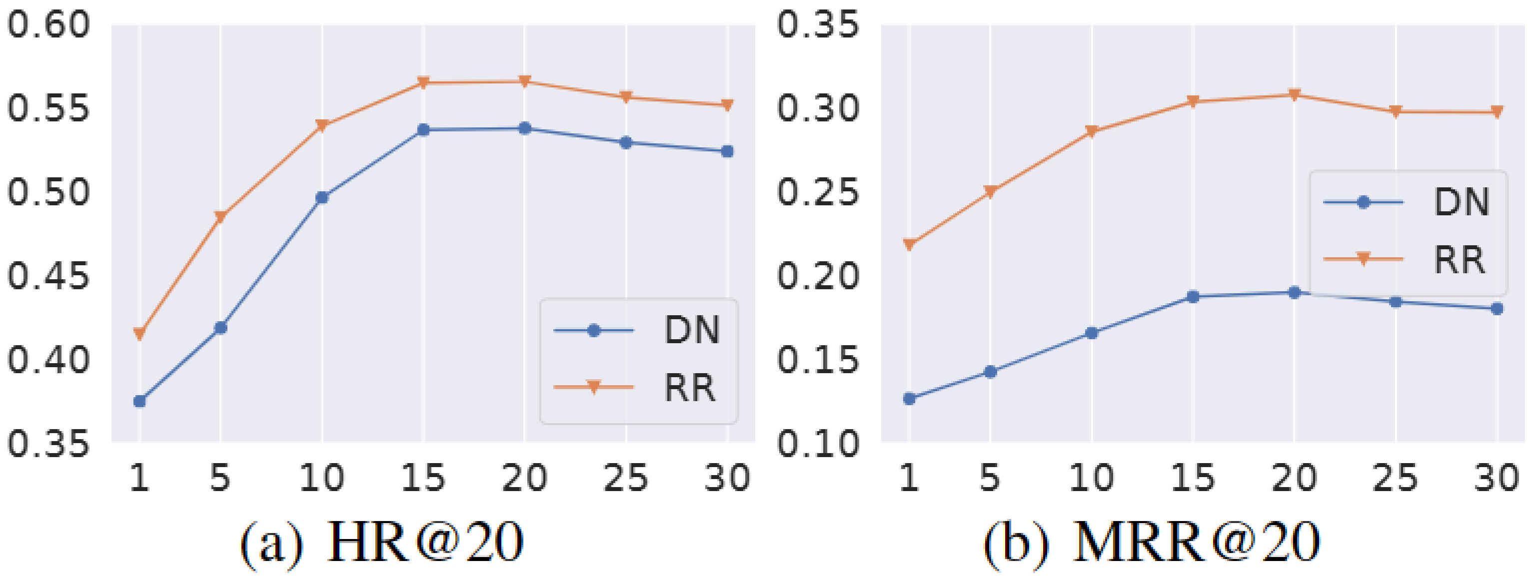

Figure 5: Performance with different normalized weight w_k.

Thanks