Dual Sparse Attention Network for Session-based Recommendation – Technical Appendix

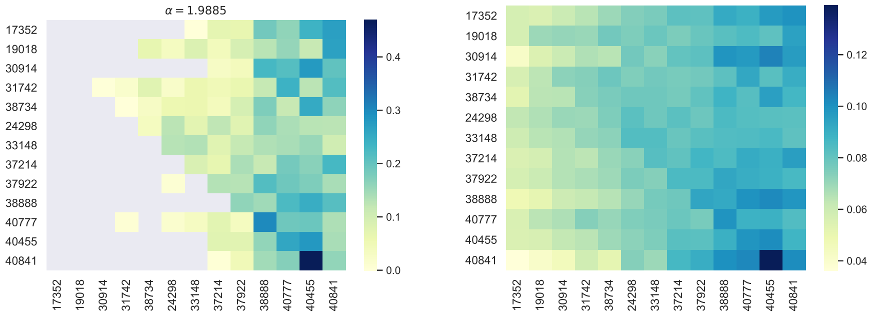
A. Parameter Setup Details

In this paper, all hyper-parameters are optimized via grid search on each dataset, respectively. Specially, we set the hyper-parameters range of grid search as follows: learning rate η in $\{0.0003, 0.0005, 0.001, 0.003\}$, dropout rate ε is in $\{0, 0.3, 0.5\}$, normalized weight w_k is in$\{1, 5, 10, 15, 20, 25, 30\}$. According to the experimental results with early-stopping strategy, the optimal hyper-parameters are $\{\eta : 0.001, \varepsilon : 0.5, w_k : 20\}$ on both two datasets. We used Adam as the model optimizer and for a fair comparison, we explore the case that embedding dimension $d = 100$, which is the optimal hyper-parameter in the previous work. We implement the proposed model by Pytorch 1.2 with Ubuntu 18.04, GPU of GeForce RTX 2080, CPU of Intel(R) Core(TM) i7-9700K and 64G Memory. We also use the Back-Propagation Through Time (BPTT) algorithm to train the model, and the mini-batch settings are {batch size: 512, epoch: 50}. All baselines use the same embedding dimension as the proposed model, and other hyper-parameters refer to the setting of the original paper and have been fine-tuned for best results. We also upload our source code and datasets in CodeAndData Appendix, which contain all details for reproducibility.

B. Example for different transformation functions

To demonstrate the utility of the proposed adaptive α-entmax, we select an example to exhibit the different effects of this function and softmax on DN dataset. We randomly choose a long session, which ranks the ground truth item 1st out of the testing set. Figure 1 illustrate the attention weights produced by the self-attention layer. This heatmap demonstrates that the proposed function does dwindle the value of some items to 0 and assigns the weight intensively, which lets us clearly distinguish the possible unrelated items. In Figure 1(a), we mark the zero value as pure gray, and the biggest item index 40481 denotes the learned target item index. We observe that some items are assigned with zero scores, indicating that these items have no correlation with the corresponding target embedding. Nevertheless, these items still have rather small weights in Figure 1(b) yet. This experiment confirms that the adaptively sparse transformation does have better item discrimination.

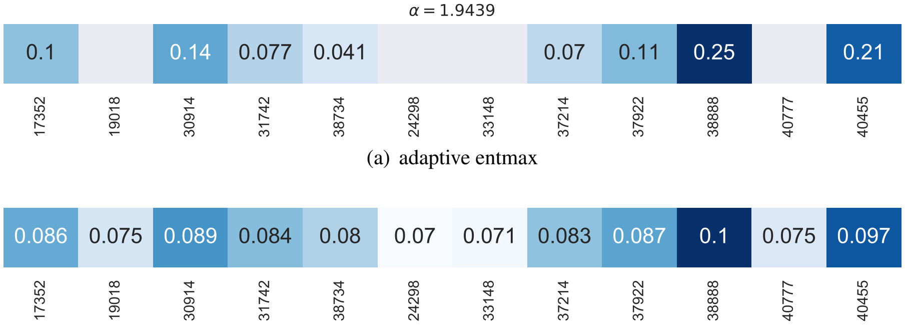
The attention weights produced by the target attention is illustrated in Figure 2. Similarly, we can use the adaptively sparse transformation function to find that four



(a) adaptive entmax

(b) softmax

Figure 1: Attention weights produced by the self-attention layer. The horizontal axis is the *key* item, and the vertical axis is the *query* item.



(a) adaptive entmax

(b) softmax

Figure 2: Attention weights produced by the target attention layer.

items in the session have no effect on the final result. Although we only report the observation in DN dataset, we also find the same phenomenon in RR dataset.