Aspect Re-distribution for Learning Better Item Embeddings in Sequential Recommendation

Wei Cai[1], Weike Pan[2], Jingwen Mao[1], Zhechao Yu[1] and Congfu Xu[1]*

cai_wei@zju.edu.cn, panweike@szu.edu.cn, jingwenmao@zju.edu.cn, 22121240@zju.edu.cn, xucongfu@zju.edu.cn

[1]College of Computer Science and Technology
Zhejiang University, Hangzhou, China

[2]College of Computer Science and Software Engineering
Shenzhen University, Shenzhen, China

Problem Definition

In the problem setting of sequential recommendation, the dataset contains a sequence set $\mathcal{S}$ and an item set $\mathcal{I}$. Each $s \in \mathcal{S}$ is a chronological item sequence $(i_1^s, i_2^s, \ldots, i_{|s|}^s)$, where $i_t^s \in \mathcal{I}$. To a given item sequence s, our task is to predict the next item $i_{|s|+1}^s$ from $\mathcal{I} \backslash s$ and provide an item list sorted by the estimated preferences.

Motivation (1/5)

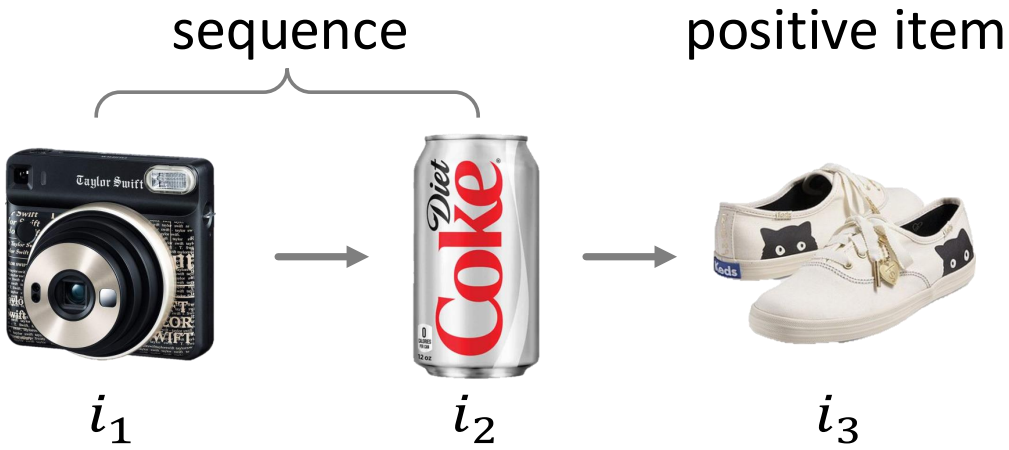


Figure: A training sample where i_1, i_2 and i_3 are all endorsed by Taylor Swift. The co-occurrence of two items only indicates their similarity in terms of endorser, and is independent of the other aspects such as category and color.

Motivation (2/5)

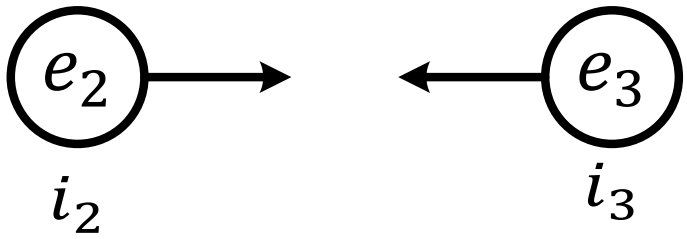


Figure: For the item i_2 in the sequence and the positive item i_3, existing models [Tang and Wang, 2018, Kang and McAuley, 2018] draw their embeddings closer in a high-dimensional space.

Motivation (3/5)

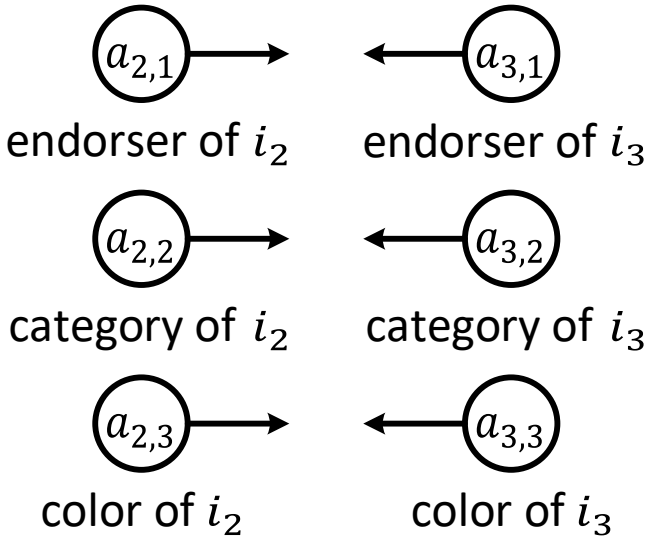


Figure: Some recent works [Ma et al., 2020, Wang et al., 2020] represent an item with several embeddings. Similarly, they make the embeddings of the two items close to each other.

Motivation (4/5)

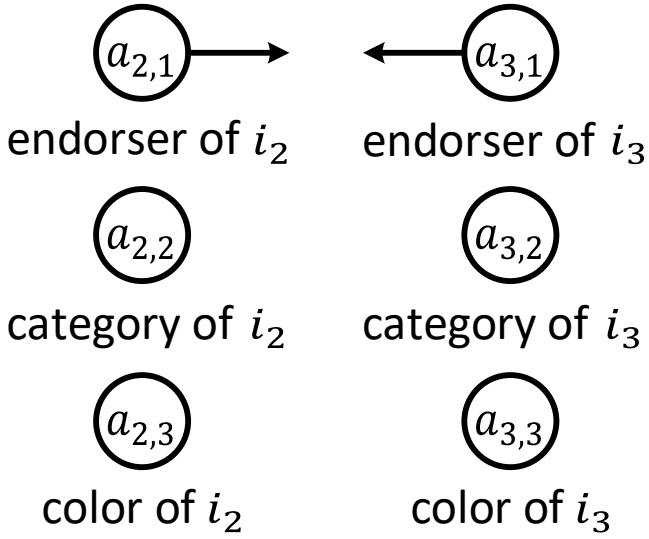
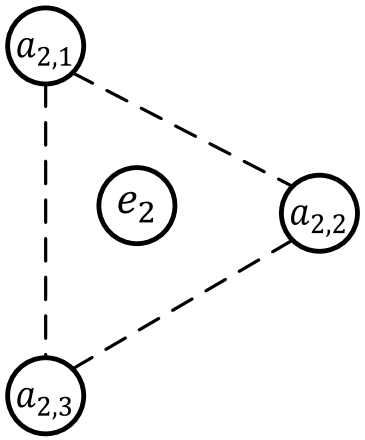


Figure: Nevertheless, recalling the training sample, it can be inferred from the sequence that both items i_2 and i_3 appear in the sequence because the user likes Taylor Swift. In other words, their co-occurrence is only because of the endorser and not related to the other aspects. Therefore, it is more accurate to update the important aspects of an item, leaving the other aspects unchanged.

Motivation (5/5)

Co-occurrence of items may only be due to some aspects of them and not related to the other aspects. Therefore, we aim to design a novel method to update the important aspects of an item.

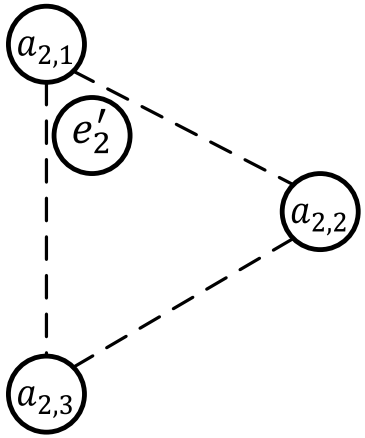
Overall of Our Solution (1/3)



$$e_2 = \frac{1}{3}a_{2,1} + \frac{1}{3}a_{2,2} + \frac{1}{3}a_{2,3}$$

Figure: For convenience, we focus on the update of item i_2. Before considering the aspect distribution (i.e., the importance of aspects), we first decompose the item embedding into several aspect embeddings. Initially, we consider each aspect to be equally important, i.e., the initial aspect distribution is uniform. Correspondingly, we make the center of the decomposed aspect embeddings coincide with the initial item embedding.

Overall of Our Solution (2/3)



$$e'_2 = \frac{4}{6}a_{2,1} + \frac{1}{6}a_{2,2} + \frac{1}{6}a_{2,3}$$

Figure: Given a sequence, we re-calculate the importance of each aspect. Specifically, we consider an aspect important if it is similar to the preceding items in the same sequence. Then, we aggregate the aspect embeddings into a new item embedding according to the aspect distribution. The new item embedding is called aspect-aware item embedding.

Overall of Our Solution (3/3)

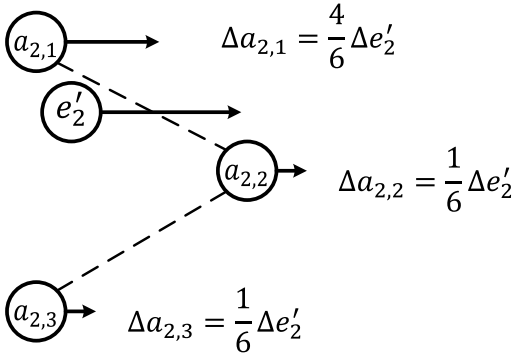


Figure: Finally, we directly provide the learned aspect-aware item embeddings to a successor model for training. The advantage of this approach is that updates of an aspect-aware item embedding are assigned to each aspect embedding according to the aspect distribution, so it focuses on updating the important aspects.

Notations

Table: Notations and descriptions.

Notation	Description
$\mathcal{I}$	item set
$\mathcal{S} = \{s\}$	sequence set
$s = (i_1^s, i_2^s, \ldots, i_{\lvert s \rvert}^s)$	a sequence
$s_t = (i_1^s, i_2^s, \ldots, i_t^s)$	a sequence containing the first t items of s
$\mathcal{P} = \{1, 2, \ldots, \lvert\mathcal{P}\rvert\}$	aspect set
$p \in \mathcal{P}$	an aspect
$d \in \mathbb{N}$	embedding dimensionality
$\boldsymbol{e}_i \in \mathbb{R}^d$	embedding of item i
$\boldsymbol{a}_{i,p} \in \mathbb{R}^d$	embedding of aspect p of item i
$w_{s,i,p} \in \mathbb{R}$	importance of aspect p of item i in s
$\boldsymbol{e}'_{s,i} \in \mathbb{R}^d$	aspect-aware item embedding of item i in s
$\hat{r}_{s_t,j} \in \mathbb{R}$	relevance of item j being the next item of s_t

Contributions

- We propose to focus on updating the important aspects of an item, which can more accurately capture the reason for the co-occurrence of the item and other items.
- We propose a novel aspect re-distribution method, which provides a solution for our insight by re-calculating the aspect distribution and reconstructing the item embeddings.
- We conduct extensive experiments on real-world datasets and show the effectiveness of our method.

Architecture (1/2)

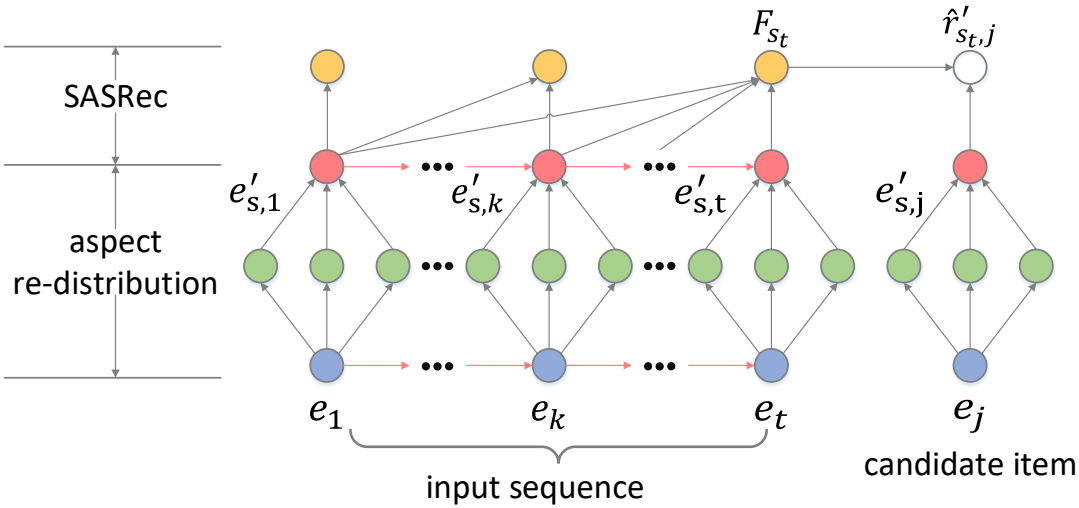


Figure: First, the embedding (blue node) of each item is decomposed under some constraints into the aspect embeddings (green nodes). Next, the aspect embeddings are aggregated into an aspect-aware item embedding (red node). Finally, the sequence embedding (yellow node) is calculated from the aspect-aware item embeddings using SASRec, with which the relevance of the candidate item being the next item can be calculated.

Architecture (2/2)

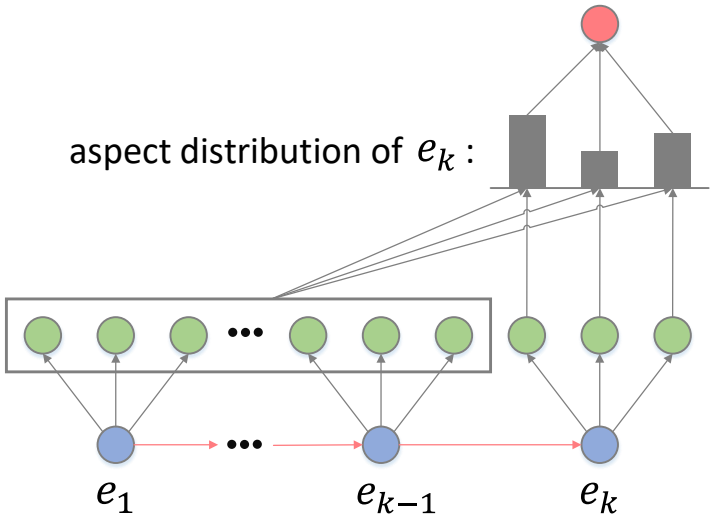


Figure: Take the generation of aspect-aware item embedding of item k as an example, all aspect embeddings of the previous items (i.e., items $1, 2, \ldots, k-1$) are accumulated (gray box). Then, the aspect distribution (grey bars) of item k is calculated. Finally, the aspect embeddings of item k are aggregated into an aspect-aware item embedding according to the aspect distribution.

Aspect Embedding Generation (1/3)

We decompose an initial item embedding into different aspect embeddings. Assuming $\mathcal{P}$ is the set of aspects that $\mathcal{P} = \{1, 2, \ldots, |\mathcal{P}|\}$ and $|\mathcal{P}|$ is a hyper-parameter, we use a projection matrix to project the item embedding $\boldsymbol{e}_i \in \mathbb{R}^d$ into embedding of aspect $p \in \mathcal{P}$:

$$\boldsymbol{a}_{i,p} = \frac{\boldsymbol{W}_p \boldsymbol{e}_i}{\|\boldsymbol{W}_p \boldsymbol{e}_i\|_2}, \tag{1}$$

where $\boldsymbol{a}_{i,p} \in \mathbb{R}^d$ is the embedding of aspect p of item i, and $\boldsymbol{W}_p \in \mathbb{R}^{d \times d}$ is the projection matrix of aspect p that is shared by all items.

Aspect Embedding Generation (2/3)

We introduce an independence loss [Wang et al., 2020] to encourage different aspects to contain different information. For item i, the mutual information of the aspects is:

$$L_{ind}^{i} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} -log \frac{\exp(\frac{s(\boldsymbol{a}_{i,p}, \boldsymbol{a}_{i,p})}{\tau})}{\sum_{p' \in \mathcal{P}} \exp(\frac{s(\boldsymbol{a}_{i,p}, \boldsymbol{a}_{i,p'})}{\tau})}, \tag{2}$$

where the $s(\cdot, \cdot)$ measures the similarity of two aspect embeddings and the τ denotes the temperature in softmax function which is a hyper-parameter. We use cosine similarity here:

$$s(\boldsymbol{a}_1, \boldsymbol{a}_2) = \frac{\boldsymbol{a}_1 \boldsymbol{a}_2^T}{\|\boldsymbol{a}_1\|_2 \|\boldsymbol{a}_2\|_2}. \tag{3}$$

Aspect Embedding Generation (3/3)

For items that frequently appear in different sequences, the independence of different aspects helps capture the reasons for its co-occurrence with the other items. Therefore, the proportion of an item in the loss function is proportional to the number of times it appears in different sequences:

$$L_{ind} = \frac{1}{|S|} \sum_{s \in \mathcal{S}} \frac{1}{|s|} \sum_{i \in s} L_{ind}^{i}, \tag{4}$$

where s is a sequence in $\mathcal{S}$.

Initial Aspect Distribution (1/2)

In the initial case, an intuitive assumption is that each aspect of an item is equally important. We define the initial aspect distribution, i.e., the importance of aspects of item i as $\mathcal{D}_i = \left[w_{i,1}, w_{i,2}, \ldots, w_{i,|\mathcal{P}|}\right]$, where $w_{i,p} = \frac{1}{|\mathcal{P}|}$ denotes the importance of aspect p of item i.
We then introduce a center loss to constrain the item embedding and the aspect embeddings:

$$L^i_{cent} = \left\| \boldsymbol{e}_i - \sum_{p \in \mathcal{P}} w_{i,p} \boldsymbol{a}_{i,p} \right\|_2, \tag{5}$$

where $\boldsymbol{e}_i$ is the embedding of item i, and $\boldsymbol{a}_{i,p}$ is the embedding of aspect p.

Initial Aspect Distribution (2/2)

Similar to Eq.(4), we use the times of an item appearing in different sequences to determine its proportion in the loss function:

$$L_{cent} = \frac{1}{|S|} \sum_{s \in \mathcal{S}} \frac{1}{|s|} \sum_{i \in s} L^{i}_{cent}, \tag{6}$$

where s is a sequence in $\mathcal{S}$.

Aspect Distribution Re-calculation (1/3)

For item i in sequence s, we re-calculate the aspect distribution according to the items in the same sequence s. We define the range of items that influence the distribution as all items before item i in the sequence s:

$$\mathcal{C}_{s,i} = \{j | j \in s, pos_{s,j} < pos_{s,i}\}, \tag{7}$$

where $pos_{s,i}$ represents the position of the item i in the sequence s.

Aspect Distribution Re-calculation (2/3)

The information of the items in $\mathcal{C}_{s,i}$ is accumulated into $\boldsymbol{q}_{s,i} \in \mathbb{R}^d$:

$$\boldsymbol{q}_{s,i} = \frac{1}{|\mathcal{C}_{s,i}|} \sum_{j \in \mathcal{C}_{s,i}} \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \boldsymbol{a}_{j,p}, \tag{8}$$

where $\frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \boldsymbol{a}_{j,p}$ contains the information of all aspects of the item j.

Aspect Distribution Re-calculation (3/3)

For the aspect p of item i, if $\boldsymbol{q}_{s,i}$ is more similar to $\boldsymbol{a}_{i,p}$, we consider that the aspect is more important. The fully connected layer is used to learn the importance $w'_{s,i,p} \in \mathbb{R}$ of aspect p of item i in sequence s from $\boldsymbol{q}_{s,i}$ and $\boldsymbol{a}_{i,p}$:

$$w'_{s,i,p} = softmax\left\{[\boldsymbol{q}_{s,i}, \boldsymbol{a}_{i,p}]\boldsymbol{W} + b\right\}, \tag{9}$$

where $[\cdot,\cdot]$ denotes the concatenation operation, and $\boldsymbol{W} \in \mathbb{R}^{2d\times 1}$ and $b \in \mathbb{R}$ denote the weights and bias of the fully connected layer, respectively. So far, we obtain a new aspect distribution $\mathcal{D}'_{s,i} = \left[w'_{s,i,1}, w'_{s,i,2}, \ldots, w'_{s,i,|\mathcal{P}|}\right] \in \mathbb{R}^{1\times|\mathcal{P}|}$.

Aspect-aware Item Embedding Generation (1/2)

After re-calculating the aspect distribution, we need to make the sequential model focus on updating the important aspects. We use the re-calculated aspect distribution $\mathcal{D}'_{s,i}$ as weights and aggregate all aspect embeddings into one single embedding:

$$\boldsymbol{e}'_{s,i} = \sum_{p \in \mathcal{P}} w'_{s,i,p} \boldsymbol{a}_{i,p}, \tag{10}$$

where $\boldsymbol{e}'_{s,i} \in \mathbb{R}^d$ contains the information of all aspects. Thus, we call it aspect-aware item embedding. Note that the item embedding $\boldsymbol{e}'_{s,i}$ is dependent on both the item i and the sequence s, which is different from that of most previous works.

Aspect-aware Item Embedding Generation (2/2)

This approach has the following advantages:

- Update of $\boldsymbol{e}'_{s,i}$ is passed back to $a_{i,p}$ according to $w'_{s,i,p}$, which allows the model to focus on updating the important aspects.
- $\boldsymbol{e}'_{s,i}$ adapts to arbitrary sequential models and thus has excellent generality.
- In the previous sequential models, an item has the same embedding in different sequences. With the introduction of the aspect-aware item embeddings, the embedding of an item can be different according to the sequence, which also improves the capability of the sequential model.

Next Item Prediction (1/2)

We use one of the most well-known models, SASRec [Kang and McAuley, 2018], as the successor recommendation model. Given a sequence of aspect-aware item embeddings $(\boldsymbol{e}'_{s,i_1^s}, \boldsymbol{e}'_{s,i_2^s}, \ldots, \boldsymbol{e}'_{s,i_t^s})$, SASRec uses the self-attention layers and the fully connected layers to aggregate the item embeddings into a subsequence embedding $\boldsymbol{F}_{s_t} \in \mathbb{R}^d$:

$$\boldsymbol{F}_{s_t} = f_{SASRec}\left((\boldsymbol{e}'_{s,i_1^s}, \boldsymbol{e}'_{s,i_2^s}, \ldots, \boldsymbol{e}'_{s,i_t^s})\right), \tag{11}$$

where f_{SASRec} denotes the self-attention layers and fully connected layers used in SASRec. Notice that the successor model can be replaced by almost any deep learning-based sequential models.

Next Item Prediction (2/2)

To predict the next item, SASRec adopts an MF layer on the aggregated sequence embedding $\boldsymbol{F}_{s_t}$ and the aspect-aware item embedding $\boldsymbol{e}'_{s,j}$ of the target item j:

$$\hat{r}_{s_t,j} = \boldsymbol{e}'_{s,j} \boldsymbol{F}_{s_t}^T, \tag{12}$$

where $\hat{r}_{s_t,j}$ denotes the relevance that the next item is item j.

Object Function

For a subsequence $s_t = (i_1^s, i_2^s, \ldots, i_t^s)$, the positive item is i_{t+1}^s. Binary cross entropy loss is adopted as the loss function here:

$$L_{rec} = -\sum_{s \in \mathcal{S}} \sum_{t \in \{1,2,\ldots,|s|-1\}} \left[\log\left(\sigma(\hat{r}_{s_t, i_{t+1}^s})\right) + \sum_{j \notin s} \log\left(1 - \sigma(\hat{r}_{s_t, j})\right) \right], \tag{13}$$

where item j is a negative item and $\sigma(\cdot)$ is the activation function. Combining Eq.(13) with Eq.(4) and Eq.(6), we have:

$$L = L_{rec} + \lambda_1 L_{ind} + \lambda_2 L_{cent}, \tag{14}$$

where λ_1 and λ_2 are hyper-parameters to control the weights of L_{ind} and L_{cent}, respectively.

Datasets

Table: Statistics of the datasets after preprocessing.

Dataset	#users	#items	#records	avg. records / user	avg. records / item
Music	20,165	20,356	132,595	6.58	6.51
Tmall	232,909	97,677	2,048,857	8.80	20.98
Baby	27,626	18,750	216,360	7.83	11.54
Twitter	7,964	4,272	157,530	19.78	36.88

Baselines

- BPR (Bayesian personalized ranking) [Rendle et al., 2012]
- FPMC (factorizing personalized markov chains) [Rendle et al., 2010]
- TransRec (translation-based recommendation) [He et al., 2017]
- GRU4Rec+ (recurrent neural networks) [Hidasi and Karatzoglou, 2018]
- Caser (convolutional sequence embedding) [Tang and Wang, 2018]
- BERT4Rec (bidirectional encoder representations) [Sun et al., 2019]
- SASRec (self-attentive sequential recommendation) [Kang and McAuley, 2018]

Parameter Configurations

- We select the embedding dimensionality d from $\{10, 20, 30, 40, 50\}$ and then set d to 50 for all models to make a fair comparison.
- For all the baselines, we select the parameters from the ranges suggested in the original papers, such as the Markov order $\in \{1, 2, \ldots, 5\}$ for Caser, the dropout rate $\in \{0.1, 0.2, \ldots, 0.9\}$ for BERT4Rec and so on.
- For our method, we set the batch size to 128, the learning rate to 0.001, the weight of the independence loss λ_1 to 0.5, the weight of the center loss λ_2 to 0.5 and select the dropout rate from $\{0.1, 0.2, \ldots, 0.9\}$ and the number of aspects $|\mathcal{P}|$ from $\{1, 2, 4, 8, 16\}$.

Main Results (1/2)

Table: Recommendation performance of our aspect re-distribution (ARD) with SASRec and seven baselines on four datasets. Notice that the best results and the second best results are marked in bold and underlined, respectively.

Dataset	Metric	BPR	FPMC	TransRec	GRU4Rec+	Caser	BERT4Rec	SASRec	ARD
Music	Rec@10	0.3769	0.4087	0.5406	0.4446	0.5180	0.5467	<u>0.5526</u>	**0.5890**
	NDCG@10	0.2217	0.2505	0.3813	0.2980	0.3341	0.3630	<u>0.3888</u>	**0.4184**
Tmall	Rec@10	0.4272	0.4675	0.5666	0.5561	0.5525	<u>0.6155</u>	0.6121	**0.6459**
	NDCG@10	0.2514	0.2862	0.3791	0.3791	0.3496	<u>0.4052</u>	0.4046	**0.4373**
Baby	Rec@10	0.4145	0.4307	0.4834	0.4232	0.4496	<u>0.5137</u>	0.5092	**0.5265**
	NDCG@10	0.2299	0.2436	0.2874	0.2494	0.2702	<u>0.3131</u>	0.3054	**0.3159**
Twitter	Rec@10	0.7143	0.6977	0.7362	0.6279	0.7384	0.7363	<u>0.7403</u>	**0.7579**
	NDCG@10	0.4853	0.4690	0.4907	0.3824	0.5317	0.5384	<u>0.5399</u>	**0.5547**

- Our ARD with SASRec achieves the best performance in all cases. In particular, the performance of our ARD with SASRec improves SASRec by 5.30% in terms of Rec@10 and 4.83% in terms of NDCG@10 on average of the four datasets. We attribute the improvement to learning item embeddings more accurately.

Main Results (2/2)

- SASRec and BERT4Rec using self-attention networks achieve the second best performance, which indicates that the self-attention networks are the state-of-the-art architectures for modeling sequential dependencies.
- The traditional models (BPR, FPMC, and TransRec) perform poorly on Tmall while achieving competitive performance on the other three datasets. The results show that the performance gap between the traditional models and the deep learning-based models is more pronounced on large datasets.

Ablation Study (1/3)

Table: Recommendation performance (Rec@10 and NDCG@10) of our ARD and its seven variants on four datasets. Notice that the best results and the second best results are marked in bold and underlined, respectively.

Dataset	Metric	Default	W/o DC	W/o IL	W/o CL	W/o DP	W/o RC
Music	Rec@10	0.5890	0.5768	0.5667	0.5815	0.5855	**0.5951**
	NDCG@10	**0.4181**	0.4081	0.4015	0.4107	0.4080	0.4132
Tmall	Rec@10	**0.6459**	0.6353	0.6363	0.6334	0.6252	0.6438
	NDCG@10	0.4373	0.4276	0.4333	0.4229	0.4146	**0.4380**
Baby	Rec@10	**0.5265**	0.5166	0.5120	0.5105	0.5158	0.5237
	NDCG@10	**0.3159**	0.3119	0.3082	0.2979	0.2989	0.3143
Twitter	Rec@10	**0.7579**	0.7487	0.7535	0.7516	0.7504	0.7550
	NDCG@10	0.5547	0.5481	0.5511	0.5447	0.5450	**0.5564**

Ablation Study (2/3)

- Without DC: We remove the decomposition operation in Eq.(1) in this variant. The degradation of the performance indicates that it is more effective to decompose an item embedding into several aspect embeddings than to simply use different embeddings to represent an item.
- Without IL: We remove the independence loss L_{ind} in Eq.(4) and find that the performance becomes worse. Without the independence loss, different aspects are more similar, which leads to the difference of aspect importance being unable to influence the prediction results.
- Without CL: We find that the performance degrades without the center loss L_{cent} in Eq.(6). Tthe center loss makes each aspect contain the same amount of information and forces each aspect to have the same initial importance. The experimental results also show that this intuitive preparation work is necessary and useful.

Ablation Study (3/3)

- Without DP: Without the dropout techniques, the performance is poor because of overfitting. In order to alleviate the overfitting caused by the model parameters introduced by our ARD, it is necessary to adopt the dropout techniques.
- Without RC: Without residual connections, the performance changes slightly. We find that in some situations, removing residual connections improves the performance. With residual connections, the initial item embeddings can propagate to the aspect-aware item embeddings. Therefore, a possible reason for the results is that the initial embeddings are less important than the aspect-aware embeddings.

Models Representing Item with Several Embeddings

- MacridVAE [Ma et al., 2019], DGCF [Wang et al., 2020] and Disen-GNN [Li et al., 2022] disentangle each user into concepts. SINE [Tan et al., 2021] and KA-MemNN [Zhu et al., 2020] disentangle each sequence into some concepts. The difference between these models and our ARD is that they treat each aspect or concept equally in prediction and training. Equivalently, they update different aspects of an item without distinction.
- A new seq2seq training framework [Ma et al., 2020] disentangles each sequence into several intentions. During training, it tries to make the input sequence and the target sequence similar within the same intention. Although it abandons some intentions with a too large loss, there is no difference in update strength for the different intentions involved in training. This leads to the model still updating different intentions with the same strength in essence.

Thank you!

- We thank the anonymous reviewers for their expert and constructive comments and suggestions.
- This paper is supported by the National Key R&D Program of China under grant (2022ZD0208605), and partially supported by the National Natural Science Foundation of China (NSFC) under grant No.62172283 and No.61672449.

He, R., Kang, W.-C., and McAuley, J. (2017).
Translation-based recommendation.
In *Recsys'17*, pages 161–169.

Hidasi, B. and Karatzoglou, A. (2018).
Recurrent neural networks with top-k gains for session-based recommendations.
In *CIKM'18*, pages 843–852.

Kang, W.-C. and McAuley, J. (2018).
Self-attentive sequential recommendation.
In *ICDM'18*, pages 197–206.

Li, A., Cheng, Z., Liu, F., Gao, Z., Guan, W., and Peng, Y. (2022).
Disentangled graph neural networks for session-based recommendation.
arXiv preprint arXiv:2201.03482.

Ma, J., Zhou, C., Cui, P., Yang, H., and Zhu, W. (2019).
Learning disentangled representations for recommendation.
arXiv preprint arXiv:1910.14238.

Ma, J., Zhou, C., Yang, H., Cui, P., Wang, X., and Zhu, W. (2020).
Disentangled self-supervision in sequential recommenders.
In *KDD'20*, pages 483–491.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2012).
Bpr: Bayesian personalized ranking from implicit feedback.
arXiv preprint arXiv:1205.2618.

Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. (2010).
Factorizing personalized markov chains for next-basket recommendation.
In *WWW'10*, pages 811–820.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., and Jiang, P. (2019).

Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer.
In *CIKM'19*, pages 1441–1450.

Tan, Q., Zhang, J., Yao, J., Liu, N., Zhou, J., Yang, H., and Hu, X. (2021).
Sparse-interest network for sequential recommendation.
In *WSDM'21*, pages 598–606.

Tang, J. and Wang, K. (2018).
Personalized top-n sequential recommendation via convolutional sequence embedding.
In *WSDM'18*, pages 565–573.

Wang, X., Jin, H., Zhang, A., He, X., Xu, T., and Chua, T.-S. (2020).
Disentangled graph collaborative filtering.
In *SIGIR'20*, pages 1001–1010.

Zhu, N., Cao, J., Liu, Y., Yang, Y., Ying, H., and Xiong, H. (2020).
Sequential modeling of hierarchical user intention and preference for next-item recommendation.
In *WSDM'20*, pages 807–815.