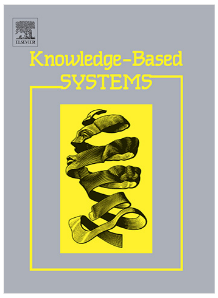
Journal Pre-proof




Learning aspect-level complementarity for intent-aware complementary recommendation

Rui Wang, Ning Yang, Philip S. Yu

PII: S0950-7051(22)01029-2
DOI: https://doi.org/10.1016/j.knosys.2022.109936
Reference: KNOSYS 109936

To appear in: *Knowledge-Based Systems*

Received date : 15 March 2022
Revised date : 20 September 2022
Accepted date : 20 September 2022

Please cite this article as: R. Wang, N. Yang and P.S. Yu, Learning aspect-level complementarity for intent-aware complementary recommendation, *Knowledge-Based Systems* (2022), doi: https://doi.org/10.1016/j.knosys.2022.109936.

# Learning Aspect-Level Complementarity for Intent-Aware Complementary Recommendation

Rui Wang[a], Ning Yang[a,*], Philip S. Yu[b]

[a]*School of Computer Science, Sichuan University, China*
[b]*Department of Computer Science, University of Illinois at Chicago, USA*

**Abstract**

Complementary recommendation aims to recommend items that are dissimilar but relevant to, and likely purchased together with, the items a user has purchased. Although many efforts have been made, the existing works for complementary recommendation still suffer from two shortcomings. First, the existing works often model the complementarity between items in terms of their co-occurrence patterns, which overlooks the influence of user intent. In fact, the intents of the users even with similar historical behaviors might be different and consequently need different complements. Second, the existing works often encode the complementary relationship at item level. In real world, however, different aspects of an item might contribute different complementarities to the same item, and the complementary information at aspect-level tends to be related with the intents of users. To overcome the two shortcomings, in this paper we propose a novel model called Aspect-level Complementarity Learning for Intent-aware Complementary Recommendation (AICRec). In particular, we propose a User Intent Perceiving (UIP) module, which enables AICRec to differentiate users' separate intents even though they are in similar scenarios. Meanwhile, we also devise an Aspect-level Complementarity Learning (ACL) module to infer an item's finer-grained complementarities to a user's intent at aspect-level, which helps AICRec personalize the recommended complementary

*Ning Yang is the corresponding author.
*Email addresses:* wangrui9295@stu.scu.edu.cn (Rui Wang), yangning@scu.edu.cn (Ning Yang), psyu@uic.edu (Philip S. Yu)

items with respect to the user's intent. At last, extensive experiments conducted on real datasets verify the superiority of AICRec over the state-of-the-art methods for complementary recommendation.

*Key words:* Intent-Aware Recommendation, Complementary Recommendation, Representation Learning

## 1. Introduction

Recommender systems have been playing an indispensable role in e-commerce platforms like Amazon and Alibaba, due to their ability to help users find out contents of interest from immense volume of items. Different from substitute recommendation where recommended items are similar and interchangeable [1–3], complementary recommendation aims to infer items that are dissimilar but relevant to, and may be purchased together with, the items that a user has already interacted with [4–7]. For example, a user who has added a laptop to the shopping cart will often choose a laptop bag or an adapter as a supplement rather than other kinds of laptops.

A variety of methods have been proposed for complementary recommendation, which often model the complementary relationships between items simply based on their frequent co-occurrence patterns discovered from user historical behavior data [8–12]. Although significant progress has been made, we argue that complementary recommendation is still far from being well solved partly due to the following challenges:

- **The Influence of User Intent** The existing methods often model the complementary relationships with frequent co-purchased patterns, which overlooks the influence of user intent, i.e., the need of a user or the reason driving a user's behaviors [13]. In real world, users may emphasize different aspects (also known as features) of the same item [14], due to their different intents. For example, Figure 1 shows that two users have purchased a laptop, but one pays more attention to the size and color of the complementary items while the other cares more about their function

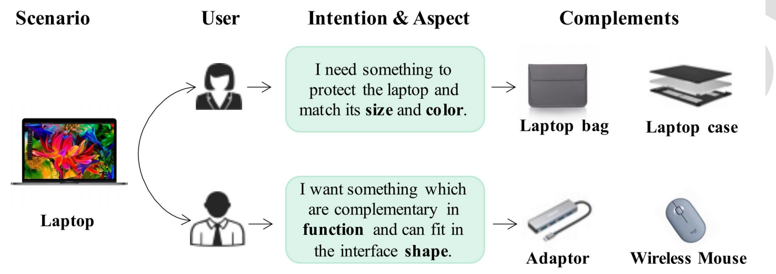



Figure 1: Illustration of motivation.

and shape, which makes their needs for complements different. As the users have different intents even in the same scenario, it is inappropriate to recommend the same complement items to them. Therefore, we need an effective method to personalize the recommended complements with respect to users' intents so that their diverse needs can be satisfied even they are in similar scenarios.

- **Fine-grained Complementarity** The existing methods often encode an item with only a single embedding to capture the item co-occurrence patterns, and evaluate the item-level complementarity in terms of the similarity between item embeddings. We argue that such coarse solution ignores the fact that items are composed of several aspects [15], and to the same item, the complementarities offered by different items are likely different on different aspects. Again taking Figure 1 as example, a laptop has the aspects of size, color, shape, and function, etc., and a laptop case might complement the laptop due to its size, while an adaptor complements it due to its function. Intuitively, the aspect-level complementary information provides rich discriminative signals to identify whether an item complements a user's intent. We hence need a method that is able to capture the fine-grained complementarity at aspect-level so that the recommended items can best serve the user intent.

In this paper, to address the above challenges, we propose two novel ideas, aspect-level complementarity learning and complementary score inference, for capturing and quantitatively evaluating the finer-grained complementarity between aspects rather than items with respect to the perceived user intent, respectively. Based on these ideas, we propose a novel model **A**spect-level Complementarity Learning for **I**ntent-aware **C**omplementary **Rec**ommendation (AICRec). In particular, to incorporate the influence of user intent, we first propose a **U**ser **I**ntent **P**erceiving (UIP) module for AICRec to identify a user's intent based on the category information of the items (which compose a scenario) that the user has interacted with. Especially, a unique intent embedding will be generated for a user via a fusion of the user's personal embedding with the scenario embedding, by which UIP is able to distinguish the different intents of users even though they are in similar scenarios. To address the challenge of fine-grained complementarity, we further devise an **A**spect-level **C**omplementarity **L**earning (ACL) module for AICRec. For an item, ACL will first break it down to $M$ latent aspect embeddings to characterize its aspect-level information, and then learn $M$ aspect-wise complementary embeddings to encode the aspect-level complementarity of a candidate item to the items that the user have interacted with. At last, we introduce a **C**omplementary **S**core **I**nference (CSI) to attentively figure out the complementary aspects under the guidance of the user intent, by which AICRec can infer the degree to which a candidate item serves a user's intent. The contributions of this paper can be summarized as follows:

- We propose a novel model called Aspect-level Complementarity Learning for Intent-aware Complementary Recommendation (AICRec), which is able to personalize the recommended complements with respect to a user's intent and the aspect-level complementarity.
- We propose two novel modules, the Aspect-level Complementarity Learning and the Complementary Score Inference, by which AICRec is able to identify and quantitatively evaluate an item's aspect-level complementarity to the user intent.
- The extensive experiments conducted on real-world datasets demonstrate

the superiority of AICRec over the state-of-the-art methods for complementary recommendation.

## 2. Related Works

In this section, we briefly review two domains of the works that are mostly related to our work, including complementary recommendation and intent-aware recommendation.

### *2.1. Complementary Recommendation*

In recent years, complementary recommendation has been widely applied in many applications, such as complementary clothing matching [16–19] and commodity bundling for e-commerce platforms [20, 21]. Zhang et al. [5] conduct a quality-aware complementary recommendation method in which the item with the best quality is recommended to the user. Nevertheless, the model ignores the asymmetric property of complementarity. McAuley et al. [4] and Zhang et al. [22] use LDA model to extract topics from item text informations. Rakesh et al. [23] propose LVAE, which links two variational auto-encoders to learn latent features over the reviews of items. Liu et al. [24] and Zhang et al. [25] leverage the graph structure to learn product representations in different relationship spaces. However, the user preferences and fine-grained semantic relationships are not taken into consideration in these works. He et al. [8] utilize multiple semantic spaces to encode fine-grained semantics of anchor items (items which have been purchased by a user). However, only mapping anchor items to multiple semantic spaces cannot match the representations of candidate items generated in a low-dimensional embedding space, resulting in semantic misalignment. Wang et al. [9] propose a multi-step path constraint algorithm, which violates the non-transitivity of complementarity. Wan et al. [26] devise Triple2vec which aims at capturing the relationships among (item, item, user) triples. However, Triple2vec neglects the higher-order complementarity when items are as a combination. Xu et al. [10] extend the Triple2vec method and propose a knowledge-aware complementary recommendation algorithm, which considers asymmetric, non-transitive, transductive and higher-order complementarity but neglects the granularity of complementarity. Hao et al. [27] propose a novel diversified complementary recommendation method, P-Companion. However, the complementary items recommended by P-Companion focus on diversity which may impair the accuracy of recommendations. The most recent work is proposed by Yang et al. [28], which can infer complementary relationships with an in-depth reasoning over a knowledge graph. By contrast with the existing works for complementary recommendation, our AICRec is able to simultaneously discern the unique intents of users and capture the finer-grained complementarities at aspect-level, which establish the superiority of AICRec.

### *2.2. Intent-aware Recommendation*

Intent-aware recommendation aims to discern the motivation of users' behaviors by learning latent intent representations [29]. Chen et al. [30] propose to learn a user intent embedding by applying attention mechanism to the categories of users' historical behaviors. Similarly, Tanjim et al. [31] encode the collaborative signals between similar users with self-attention mechanism to capture a user intent. Zhu et al. [32] propose to represent user intents with the embeddings that are generated through a key addressing over a memory network of a user's historical interactions. Wang et al. [33] point out that each item a user interacted with may serve a variety of purposes, and propose a Mixture Channel Purpose Routing Network to generate purpose-based specific representations for items. In [34], they further propose to capture a user's multiple intents with a neural intent-driven model.

Recently, researchers have also proposed to utilize side information for the modeling of user intent. For example, Wang et al. [35] emploit knowledge graph to capture users' motivation hidden in their complex historical behaviors. Yang et al. [36] propose a feedback interactive neural network to estimate user's potential intent which incorporates the negative feedback information. Liu et al. [37] realize that there exist multiple intents within a basket, and propose a framework named as Multi-Intent Translation Graph Neural Network

(MITGNN), which adopts GNN to model the multiple relations between items in a basket for capturing a user's multiple intents. Hao et al. [38] think of a user's intent as being time-evolving and propose a Dynamic Evolution based Deep Hierarchical Intention Network (Dy-HIEN), which applies a hierarchical dynamic embedding learning method to capture the drift of a user's intent over sessions. Basically, the existing methods for intent-aware recommendation lack finer-grained analysis on how an item matches a user's intent at aspect-level, which is in striking contrast with our work.

## 3. Basic Definition And Problem Formulation

Let $\mathcal{U}$ and $\mathcal{V}$ denote the user set and item set, respectively. For a user $u \in \mathcal{U}$, let $\mathcal{V}_t^u = \langle v_1^u, \cdots, v_t^u \rangle$ be $u$'s interaction sequence up to time step $t$, where $v_i \in \mathcal{V}$ is the item that $u$ interacts with at time step $i$, $1 \leq i \leq t$. Let $\mathcal{C}$ be the set of item categories, and $c(v) : \mathcal{V} \mapsto \mathcal{S}$ be the mapping function that returns the category of an item $v$. Based on the category mapping function, we define the interaction **scenario** (context) of a user $u$ till time step $t$ as the category sequence $\mathcal{C}_t^u = \langle c(v_1^u), \cdots, c(v_t^u) \rangle$. Then the target problem of this paper can be formulated as follow:

**Target Problem** Given a training set $\mathcal{D} = \{(u, \mathcal{V}_t^u, v_{t+1}^u)\}$ $(u \in \mathcal{U})$, where $v_{t+1}^u$ is the ground-truth item that user $u$ interacts with at time step $t+1$, we want to learn a ranking model $\mathrm{AICRec}(u, \mathcal{V}_t^u, v)$ to evaluate the complementarity $r_{u,v}$ of a candidate item $v$ to user $u$'s intent, such that

$$v_{t+1}^u = \underset{v \in \mathcal{V} \setminus \mathcal{V}_t^u}{\operatorname{argmax}} \mathrm{AICRec}(u, \mathcal{V}_t^u, v). \tag{1}$$

## 4. The Proposed Model

In this section, we first give a high-level view of the proposed AICRec model, then present its components in detail, and finally describe its learning. For simplicity, in the following text, we will omit the superscript $u$ when the context is unambiguous.

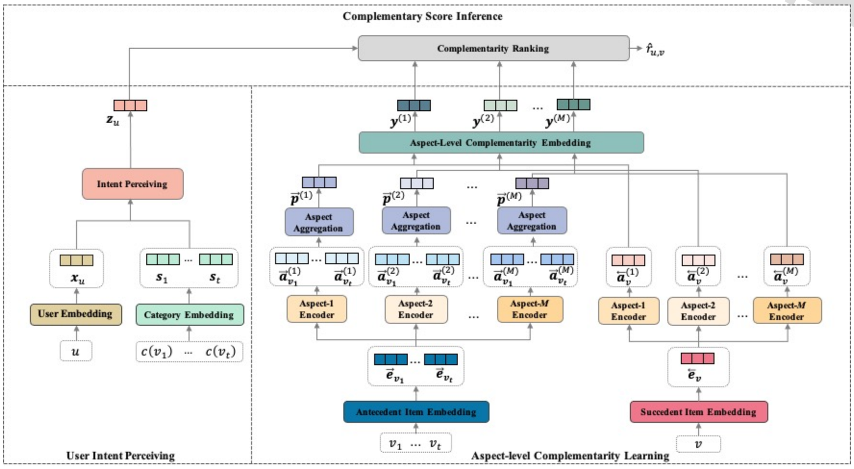



Figure 2: The architecture of AICRec.

### 4.1. Overview of AICRec

The architecture of AICRec is shown in Figure 2, from which we can see that AICRec consists of three modules, the User Intent Perceiving (UIP), the Aspect-level Complementarity Learning (ACL), and the Complementary Score Inference (CSI). We first use a toy example to illustrate the running of AICRec. Let's suppose a user $u$ has purchased an "iphone" and an "iphone case" in the first two time steps, i.e., $u$'s historical interaction set is $\mathcal{V}_2 = \{v_1, v_2\}$, where $v_1$ and $v_2$ are two one-hot encodings representing iphone and iphone case, respectively. Now we want AICRec to estimate the complementarity score $\hat{r}_{u,v}$, i.e., the degree to which a candidate item $v$, say "charger", is complementary to the user $u$'s interaction history $\mathcal{V}_2$. At first, AICRec will invoke UIP to encode the user $u$'s intent into the intent embedding $\boldsymbol{z}_u$. UIP will generate two scenario embeddings, $\boldsymbol{s}_1$ for $v_1$, and $\boldsymbol{s}_2$ for $v_2$, based on the the categories of the historical interaction items, $c(v_1)$ = "cell phone" and $c(v_2)$ = "accessory", respectively, and then generate $\boldsymbol{z}_u$ by fusing $\{\boldsymbol{s}_1, \boldsymbol{s}_2\}$ with $u$'s user embedding $\boldsymbol{x}_u$. Suppose we consider two latent aspects which correspond to "color" and "size", respectively. AICRec will invoke ACL to generate the aspect embeddings $\overrightarrow{\boldsymbol{a}}_{v_1}^{(1)}$ and $\overrightarrow{\boldsymbol{a}}_{v_2}^{(1)}$ to represent the color aspects of $v_1$ and $v_2$, respectively, and the aspect embeddings $\overrightarrow{\boldsymbol{a}}_{v_1}^{(2)}$ and $\overrightarrow{\boldsymbol{a}}_{v_2}^{(2)}$ to represent the size aspects of $v_1$ and $v_2$, respectively. Then ACL will generate the summary color aspect embedding $\overrightarrow{\boldsymbol{p}}^{(1)}$ for the whole history by aggregating the historical items' color aspect embeddings $\overrightarrow{\boldsymbol{a}}_{v_1}^{(1)}$ and $\overrightarrow{\boldsymbol{a}}_{v_2}^{(1)}$, and the summary size aspect embedding $\overrightarrow{\boldsymbol{p}}^{(2)}$ by aggregating the historical items' size aspect embeddings $\overrightarrow{\boldsymbol{a}}_{v_1}^{(2)}$ and $\overrightarrow{\boldsymbol{a}}_{v_2}^{(2)}$. With a matching between the summary color aspect embedding $\overrightarrow{\boldsymbol{p}}^{(1)}$ and the candidate item $v$'s color aspect embedding $\overleftarrow{\boldsymbol{a}}_{v}^{(1)}$, and a matching between the summary size aspect embedding $\overrightarrow{\boldsymbol{p}}^{(2)}$ and the candidate item $v$'s size aspect embedding $\overleftarrow{\boldsymbol{a}}_{v}^{(2)}$, ACL will generate the aspect-level complementarity embeddings $\boldsymbol{y}^{(1)}$ for color aspect and $\boldsymbol{y}^{(2)}$ for size aspect, respectively. At last, AICRec will invoke CSI to infer the complementary score $\hat{r}_{u,v}$ of the candidate item $v$ to the user $u$ based on the aspect-level complementarity embeddings $\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}\}$ and the user intent embedding $\boldsymbol{z}_u$. The whole procedure of AICRec can be formally described as follows:

(1) For a user $u$, UIP captures $u$'s unique intent with the intent embedding $\boldsymbol{z}_u$, which is generated with the fusion of the user embedding $\boldsymbol{x}_u$ and the scenario embeddings $\{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_t\}$. Here $\boldsymbol{s}_i$ is the embedding of the category $c(v_i)$ of $u$'s $i$-th interaction $v_i \in \mathcal{V}_t$, $1 \leq i \leq t$.

(2) At the same time, ACL encodes the aspect-level complementarities of the candidate item $v \in \mathcal{V} \setminus \mathcal{V}_t$ to the user $u$'s historical interactions $\mathcal{V}_t$, with the $M$ aspect-level complementarity embeddings $\{\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(M)}\}$. In order to capture the directional information of complementary relationship, ACL learns two embeddings for each item to reflect its different roles. If an item $v_i$ is a historical interaction of the user, ACL generates the antecedent item embedding $\overrightarrow{\boldsymbol{e}}_{v_i}$ to represent it, while if it is a candidate item, its succedent item embedding $\overleftarrow{\boldsymbol{e}}_{v_i}$ will be generated. With the antecedent item embedding $\overrightarrow{\boldsymbol{e}}_{v_i}$ of a historical interaction item $v$ as input, $M$ aspect encoders are invoked to generate $v_i$'s aspect embeddings $\{\overrightarrow{\boldsymbol{a}}_{v_i}^{(1)}, \cdots, \overrightarrow{\boldsymbol{a}}_{v_i}^{(M)}\}$. By an aspect-wise

aggregation of the latent aspect embeddings of the historical interactions in $\mathcal{V}_t$, ACL obtains $M$ summary aspect embeddings $\{\overrightarrow{\boldsymbol{p}}^{(1)}, \cdots, \overrightarrow{\boldsymbol{p}}^{(M)}\}$ for the whole interaction history $\mathcal{V}_t$. Similarly, for a candidate item $v$, ACL extracts its $M$ latent aspect embeddings $\{\overleftarrow{\boldsymbol{a}}_v^{(1)}, \cdots, \overleftarrow{\boldsymbol{a}}_v^{(M)}\}$ based on its succedent item embedding. At last, ACL generates the $M$ aspect-level complementarity embeddings $\{\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(M)}\}$ based on the summary aspect embeddings of $\mathcal{V}_t$ and the latent aspect embeddings of candidate item $v$.

(3) Finally, AICRec will invoke CSI to infer the complementarity score $\hat{r}_{u,v}$ to evaluate the degree to which a candidate item $v$ is complementary to the user $u$'s intent, by integrating the aspect-level complementarity embeddings $\{\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(M)}\}$ with respect to the intent embedding $\boldsymbol{z}_u$.

### *4.2. User Intent Perceiving*

The task of UIP module is to capture a given user unique intent based on the user's interaction history $\mathcal{V}_t = \langle v_1, \cdots, v_t \rangle$. Intuitively, a user's intent is often revealed by the category information of the items she has already interacted with. For example, if a user has purchased items of sports and dressing categories, she may want to choose sportswear next time, while if she is browsing household items, she may be preparing a new home. Meanwhile, even though two users interact with the same items, they also likely have different intents because user intent also depends on the user's individual information. Therefore, to accurately capture the user intent, we need to consider both the user itself and the scenario $\mathcal{C}_t = \langle c(v_1), \cdots, c(v_t) \rangle$, where $c(v_i) \in \mathcal{C}$ is the category of item $v_i$, $1 \le i \le t$.

Based on the above idea, we first obtain the user embedding $\boldsymbol{x}_u \in \mathbb{R}^{d \times 1}$ of $u$ and the scenario embedding $\boldsymbol{s}_i \in \mathbb{R}^{d \times 1}$ of $c(v_i)$, where $d$ is the embedding size, with the following transformations:

$$\boldsymbol{x}_u = \boldsymbol{W}_{\mathrm{u}} \boldsymbol{u}, \tag{2}$$

$$\boldsymbol{s}_i = \boldsymbol{W}_{\mathrm{c}} \boldsymbol{c}_i, \tag{3}$$

where $\boldsymbol{u} \in \mathbb{R}^{|\mathcal{U}|\times 1}$ and $\boldsymbol{s}_i \in \mathbb{R}^{|\mathcal{C}|\times 1}$ are two one-hot vectors representing the user $u$ and the category $c(v_i)$, respectively, $\boldsymbol{W}_{\mathrm{u}} \in \mathbb{R}^{d\times|\mathcal{U}|}$ and $\boldsymbol{W}_{\mathrm{s}} \in \mathbb{R}^{d\times|\mathcal{C}|}$ are learnable matrices. Then we generate the user intent embedding $\boldsymbol{z}_u \in \mathbb{R}^{d\times 1}$ by fusing the user embedding $\boldsymbol{x}_u$ and the scenario embeddings $\{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_t\}$ as follow:

$$\boldsymbol{z}_u = \boldsymbol{W}_{\mathrm{z}}^{(1)}\big[\boldsymbol{x}_u \oplus \sigma\big(\boldsymbol{W}_{\mathrm{z}}^{(2)}(\boldsymbol{s}_1 \oplus \boldsymbol{s}_2 \oplus \cdots \oplus \boldsymbol{s}_t) + \boldsymbol{b}_{\mathrm{z}}^{(2)}\big) + \boldsymbol{b}_{\mathrm{z}}^{(1)}\big], \tag{4}$$

where $\oplus$ is concatenation operation, $\boldsymbol{W}_{\mathrm{z}}^{(1)} \in \mathbb{R}^{d\times(d+d)}$, $\boldsymbol{W}_{\mathrm{z}}^{(2)} \in \mathbb{R}^{d\times td}$, $\boldsymbol{b}_{\mathrm{z}}^{(1)} \in \mathbb{R}^{2d\times 1}$, and $\boldsymbol{b}_{\mathrm{z}}^{(2)} \in \mathbb{R}^{d\times 1}$ are learnable parameters. Here $\sigma(\cdot)$ is a nonlinear activation function, for which we choose ReLU in this paper. The insight of Equation (4) is that the user intent represented by $\boldsymbol{z}_u$ is dependent on the scenario represented by the scenario embeddings $\{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_t\}$ as well as the user information encoded in the user embedding $\boldsymbol{x}_u$, by which we can distinguish the personalized intent for the users even though their scenarios are identical.

### *4.3. Aspect-level Complementarity Learning*

As we have mentioned before, ACL is responsible for encoding the directional complementarity offered by a candidate item $v$ to a user's historical interactions $\mathcal{V}_t$ at each latent aspect.

#### *4.3.1. Directional Item Embedding*

To preserve the asymmetric and non-transitive property of complementary relationships [10], we use two different transformations to learn item embeddings for the items in $\mathcal{V}_t$ and the candidate items in $\mathcal{V}\backslash\mathcal{V}_t$, respectively. Let $\boldsymbol{v} \in \mathbb{R}^{|\mathcal{V}|\times 1}$ be a one-hot vector representing item $v \in \mathcal{V}$. Then for an item $v \in \mathcal{V}_t$, we represent it with its antecedent embedding $\overrightarrow{\boldsymbol{e}}_v \in \mathbb{R}^{d\times 1}$ which is generated as follow:

$$\overrightarrow{\boldsymbol{e}}_v = \overrightarrow{\boldsymbol{W}}_{\mathrm{e}}\boldsymbol{v}, \tag{5}$$

where $\overrightarrow{\boldsymbol{W}}_{\mathrm{e}} \in \mathbb{R}^{d\times|\mathcal{V}|}$ is a learnable antecedent item embedding matrix. At the same time, for a candidate item $v \in \mathcal{V} \setminus \mathcal{V}_t$, we represent it with its succedent item embedding which is generated as follow:

$$\overleftarrow{\boldsymbol{e}}_v = \overleftarrow{\boldsymbol{W}}_{\mathrm{e}}\boldsymbol{v}, \tag{6}$$

where $\overleftarrow{\boldsymbol{W}}_{\mathrm{e}} \in \mathbb{R}^{d\times|\mathcal{V}|}$ is a learnable succedent item embedding matrix.

*4.3.2. Latent Aspect Embedding*

To capture the fine-grained complementarity at aspect-level, we first decouple an item embedding into $M$ latent aspect embeddings by invoking a set of aspect encoders. Given the antecedent item embedding $\overrightarrow{\boldsymbol{e}}_v$ of an item $v \in \mathcal{V}_t$, we choose an MLP as the aspect encoder due to its simplicity, to extract $v$'s $k$-th latent aspect embedding $\overrightarrow{\boldsymbol{a}}_v^{(k)} \in \mathbb{R}^{d\times 1}$, which is defined as

$$\overrightarrow{\boldsymbol{a}}_v^{(k)} = \boldsymbol{Q}_1^{(k)}\sigma(\boldsymbol{Q}_2^{(k)}\overrightarrow{\boldsymbol{e}}_v + \boldsymbol{b}_2^{(k)}) + \boldsymbol{b}_1^{(k)}, \tag{7}$$

where $\boldsymbol{Q}_1^{(k)} \in \mathbb{R}^{d\times d}$, $\boldsymbol{Q}_2^{(k)} \in \mathbb{R}^{d\times d}$, $\boldsymbol{b}_1^{(k)} \in \mathbb{R}^{d\times 1}$, and $\boldsymbol{b}_2^{(k)} \in \mathbb{R}^{d\times 1}$ are learnable parameters of the $k$-th aspect encoder. Similarly, for a candidate item $v \in \mathcal{V}\backslash\mathcal{V}_t$, we can obtain its $M$ latent aspect embeddings $\{\overleftarrow{\boldsymbol{a}}_v^{(k)}\}$ $(1 \le k \le M)$ by applying the same aspect encoders (Equation (7)) to its succedent item embedding $\overleftarrow{\boldsymbol{e}}_v$. It is noteworthy that for the same item $v$, its latent aspect embeddings will be different when it plays different roles. When $v$ is a historical interaction item of some user, its latent aspect embeddings are extracted from its antecedent item embedding $\overrightarrow{\boldsymbol{e}}_v$, while when it acts as a candidate item, they come from its succedent item embedding $\overleftarrow{\boldsymbol{e}}_v$.

Intuitively, it is desirable that the decoupled latent aspect embeddings can carry aspect information as independent as possible. For this purpose, we impose the following orthogonal loss on the latent aspect embeddings during the training stage:

$$\mathcal{L}_{\mathrm{a}} = \sum_{v\in\mathcal{V}} \| \overrightarrow{\boldsymbol{A}}_v^{\mathrm{T}}\overrightarrow{\boldsymbol{A}}_v - \boldsymbol{I} \|_{\mathrm{F}}^2 + \| \overleftarrow{\boldsymbol{A}}_v^{\mathrm{T}}\overleftarrow{\boldsymbol{A}}_v - \boldsymbol{I} \|_{\mathrm{F}}^2, \tag{8}$$

where $\|\cdot\|_{\mathrm{F}}$ is Frobenius norm, $\overrightarrow{\boldsymbol{A}}_v \in \mathbb{R}^{d\times M}$ and $\overleftarrow{\boldsymbol{A}}_v \in \mathbb{R}^{d\times M}$ are two matrices with $\{\overrightarrow{\boldsymbol{a}}_v^{(k)}\}$ and $\{\overleftarrow{\boldsymbol{a}}_v^{(k)}\}$ as columns, respectively, and $\boldsymbol{I}$ is an identity matrix.

#### 4.3.3. Summary Aspect Embedding

Once we obtain the $M$ latent aspect embeddings $\{\overrightarrow{\boldsymbol{a}}_v^{(k)}\}$ for each historical interaction item $v \in \mathcal{V}_t$, we will generate $M$ summary aspect embeddings $\{\overrightarrow{\boldsymbol{p}}^{(k)} \in \mathbb{R}^{d\times 1}\}$ $(1 \leq k \leq M)$ for the whole historical interaction set $\mathcal{V}_t$. Intuitively, the $k$-th summary aspect embedding $\overrightarrow{\boldsymbol{p}}^{(k)}$ is supposed to be an aggregation over the $k$-th latent aspect embeddings of all the items in $\mathcal{V}_t$, and different items in $\mathcal{V}_t$ contribute unequally. Therefore, to quantitatively differentiate the contributions of the latent aspect embeddings of different items, we employ the following attention mechanism to generate $\overrightarrow{\boldsymbol{p}}^{(k)}$:

$$\overrightarrow{\boldsymbol{p}}^{(k)} = \sum_{v\in\mathcal{V}_t} \alpha_v^{(k)} \overrightarrow{\boldsymbol{a}}_v^{(k)}, \tag{9}$$

$$\alpha_v^{(k)} = \frac{\exp\left(\frac{\boldsymbol{q}^{(k)\mathrm{T}}\overrightarrow{\boldsymbol{a}}_v^{(k)}}{\sqrt{d}}\right)}{\sum_{v'\in\mathcal{V}_t}\exp\left(\frac{\boldsymbol{p}^{(k)\mathrm{T}}\overrightarrow{\boldsymbol{a}}_{v'}^{(k)}}{\sqrt{d}}\right)}, \tag{10}$$

where $\alpha_v^{(k)}$ is the attention coefficient of $v$ to the $k$-th summary aspect embedding, and $\boldsymbol{q}^{(k)}$ is a learnable query vector.

#### 4.3.4. Aspect-level Complementarity Embedding

At last, we embed the complementary relationship from a candidate item $v \in \mathcal{V} \setminus \mathcal{V}_t$ to a user's historical interactions $\mathcal{V}_t$ at each latent aspect. Intuitively, the complementarity at $k$-th aspect depends on the aspect embedding $\overleftarrow{\boldsymbol{a}}_v^{(k)}$ of the candidate item $v$ and the summary aspect embedding $\overrightarrow{\boldsymbol{p}}^{(k)}$ of $\mathcal{V}_t$. Therefore, we straight generate the $M$ aspect-level complementarity embeddings $\{\boldsymbol{y}^{(k)} \in \mathbb{R}^{d\times 1}\}$ $(1 \leq k \leq M)$ with a fusion of $\overleftarrow{\boldsymbol{a}}_v^{(k)}$ and $\overrightarrow{\boldsymbol{p}}^{(k)}$ conducted by the following MLP:

$$\boldsymbol{y}^{(k)} = \boldsymbol{W}_{\mathrm{y}}\sigma\big(\boldsymbol{W}_{\mathrm{y}}^{'}(\overrightarrow{\boldsymbol{p}}^{(k)} \oplus \overleftarrow{\boldsymbol{a}}_v^{(k)}) + \boldsymbol{b}_{\mathrm{y}}^{'}\big) + \boldsymbol{b}_{\mathrm{y}}, \tag{11}$$

where $\boldsymbol{W}_{\mathrm{y}} \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_{\mathrm{y}}^{'} \in \mathbb{R}^{d\times 2d}$, $\boldsymbol{b}_{\mathrm{y}} \in \mathbb{R}^{d\times 1}$, and $\boldsymbol{b}_{\mathrm{y}}^{'} \in \mathbb{R}^{d\times 1}$ are learnable parameters.

### 4.4. Complementary Score Inference

So far, we have obtained the $M$ complementarity embeddings $\{\boldsymbol{y}^{(k)}\}$ $(1 \leq k \leq M)$, each of which encodes the complementarity of the candidate item $v$

to the user's historical interactions at a latent aspect. Now we will infer the final complementary score $\hat{r}_{u,v}$ as the estimate of the likelihood that the user $u$ will interact with $v$ because of the intent of the user. The idea here is that different aspects serve the user's intent with different weights. For this purpose, we adaptively weigh the contributions of the different aspects with the attention coefficients learned by an attention mechanism. In particular, we take the user's intent embedding $\boldsymbol{z}_u$ as the query vector and aggregate the aspect-level complementarity embeddings $\{\boldsymbol{y}^{(k)}\}$ into one intent-aware complementarity embedding $\boldsymbol{y}_{u,v} \in \mathbb{R}^{d\times 1}$ with the following attention mechanism:

$$\boldsymbol{y}_{u,v} = \sum_{k=1}^{M} \beta_k \boldsymbol{y}^{(k)}, \tag{12}$$

$$\beta_k = \frac{\exp(\frac{\boldsymbol{z}_u{}^{\mathrm{T}} \boldsymbol{y}^{(k)}}{\sqrt{d}})}{\sum_{i=1}^{M} \exp(\frac{\boldsymbol{z}_u{}^{\mathrm{T}} \boldsymbol{y}^{(i)}}{\sqrt{d}})}. \tag{13}$$

Finally, we obtain the complementary score $\hat{r}_{u,v}$ with the following two-layer MLP:

$$\hat{r}_{u,v} = \sigma\big(\boldsymbol{W}_{\mathrm{r}}^{'}(\boldsymbol{W}_{\mathrm{r}}\boldsymbol{y}_{u,v} + \boldsymbol{b}_{\mathrm{r}}^{'}) + \boldsymbol{b}_{\mathrm{r}}\big), \tag{14}$$

where $\boldsymbol{W}_{\mathrm{r}} \in \mathbb{R}^{d\times d}$, $\boldsymbol{W}_{\mathrm{r}}^{'} \in \mathbb{R}^{d\times d}$, $\boldsymbol{b}_{\mathrm{r}} \in \mathbb{R}^{d\times 1}$, and $\boldsymbol{b}_{\mathrm{r}}^{'} \in \mathbb{R}^{d\times 1}$ are learnable parameters.

### 4.5. Learning of AICRec

A training set $\mathcal{D}$ is a set of quads $(u, \mathcal{V}_t, v_+, v_-)$, where $v_+$ is the positive item with which user $u$ interacts after $t$ and $v_-$ is a negative item. For a positive item $v_+$, we utilize the popularity based sampling strategy [39] to generate three negative items, where the more frequent an item $v$ ($v \neq v_+$) occurs in the dataset, the higher its sampling probability. Then following the popular pairwise ranking criteria of BPR [40], we define the complementary loss as follow:

$$\mathcal{L}_{\mathrm{c}} = \frac{1}{|\mathcal{D}|} \sum_{(u,\mathcal{V}_t,v_+,v_-)\in\mathcal{D}} \log(\hat{r}_{u,v_+} - \hat{r}_{u,v_-}), \tag{15}$$

where a pair $(v_+, v_-)$ will incur penalty if $\hat{r}_{u,v_+} < \hat{r}_{u,v_-}$. Finally, by integrating the orthogonal loss $\mathcal{L}_{\mathrm{a}}$ defined in Equation (8) and the complementary loss $\mathcal{L}_{\mathrm{c}}$,

| Datasets | #Users | #Items | #Categories | #Interactions |
|---|---|---|---|---|
| Tmall | 154,216 | 319,727 | 1,297 | 2,288,397 |
| TaoBao | 39,363 | 228,060 | 5,547 | 442,660 |
| Superstore | 793 | 1,862 | 17 | 9,994 |

Table 1: The statistics of datasets.

the overall loss can be defined as

$$\mathcal{L} = \mathcal{L}_{\mathrm{c}} + \lambda \mathcal{L}_{\mathrm{a}} + \|\Theta\|_2, \tag{16}$$

where $\Theta$ represents all learnable parameters of AICRec and $\lambda$ is a factor controlling the contribution of the orthogonal constraint. In experiments, we adopt Adam [41] as the optimizer to minimize $\mathcal{L}$, due to its ability to adaptively determine the learning rate during the gradient descent.

## 5. Experiments

The goal of the experiments is to verify the superiority of AICRec by answering the following research questions:

- **RQ1**: How does AICRec perform as compared to the state-of-the-art complementary recommendation methods?
- **RQ2**: How does AICRec benefit from its components UIP and ACL?
- **RQ3**: How do the hyper-parameters affect the performance of AICRec?
- **RQ4**: How can the superiority of AICRec be illustrated with intuitive and visualizable case studies?

### *5.1. Experiment Setting*

#### *5.1.1. Datasets*

We conduct the experiments on the following three real datasets, which are summarized in Table 1.

- **Tmall** [1] Tmall consists of more than 2M interactions of more than 154K users with more than 319K items falling into 1,297 categories, which are collected within the six consecutive months until Nov. 11, 2015.
- **TaoBao** [2]: Taobao dataset contains more than 442K interactions of more than 39K users with more than 228K items falling into more than 5K categories during the week from November 25 to December 3, 2017.
- **Superstore** [3]: Superstore dataset is derived from American supermarkets, which contains more than 9K interactions of 793 users with more than 1,862 items falling into 17 categories.

For each dataset, we filter out the users or items with less than 3 interaction records. At the same time, we randomly partition each dataset into training set, validation set, and test set with proportions of 60, 20, and 20 percent, respectively.

*5.1.2. Baselines*

To verify the superiority of AICRec, we compare it with the following state-of-the-art complementary recommendation methods:

- **Monomer** [8]: Monomer is a complementary recommendation research designed to mine local compatibility relationships between pairs of items.
- **Triple2vec** [26]: Triple2vec is a representation learning method, which models the complementary relationships of triples (item, item, user) linked by the same basket.
- **KA-CRL** [10]: KA-CRL is also a representation learning method which models the complementary relationships between item sequences.

[1] https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
[3] https://tianchi.aliyun.com/dataset/dataDetail?dataId=93284

- **P-Companion** [27]: P-Companion is a diversity complementary recommendation method, which aims at recalling as many categories of items as possible with an item embedding as input.

- **ASLI** [31]: ASLI is a sequential recommendation model based on convolutional networks, which utilizes self-attention mechanism to capture the dependence between items within a sequence, and a time-sensitive convolutional neural network to capture users' potential intents.

*5.1.3. Evaluation Protocol*

We evaluate AICRec and the baseline methods with the widely adopted metrics HR (Hits Ratio) and NDCG (Normalized Discounted Cumulative Gain). HR@$k$ is the ratio of the users whose ground-truth is ranked in the first $k$ positions, while NDCG@$k$ accounts for the position of the hit, which assigns higher weight to the hits at higher positions.

Let $S_k^u$ be the number of the testing user $u$'s instances that a ground-truth item is ranked in the corresponding recommended top-$k$ item list, which is defined as

$$S_k^u = \sum_{i=1}^{N_u} \mathbb{I}\big(rank(v_i^u, \boldsymbol{l}_i^u) \leq k\big), \tag{17}$$

where $v_i^u$ is the ground-truth item of $u$'s $i$-th testing instance, $\boldsymbol{l}_i^u$ is the corresponding recommended top-$k$ item list, $N_u$ is the number of the testing instances of $u$, $rank(v, l)$ is the rank of item $v$ in list $l$, and $\mathbb{I}(x)$ is the indicator function that returns 1 if $x$ is true, otherwise 0. Then HR@$k$ and NDCG@$k$ can be defined as follows:

$$\text{HR@}k = \frac{1}{|\mathcal{U}_t|} \sum_{u \in \mathcal{U}_t} \mathbb{I}(S_k^u \geq 1), \tag{18}$$

$$\text{NDCG@}k = \frac{1}{|\mathcal{U}_t|} \sum_{u \in \mathcal{U}_t} \frac{1}{N_u} \sum_{i=1}^{N_u} \frac{\mathbb{I}\big(rank(v_i^u, \boldsymbol{l}_i^u) \leq k\big)}{\log_2\big(1 + rank(v_i^u, \boldsymbol{l}_i^u)\big)}. \tag{19}$$

where $\mathcal{U}_t$ is the set of testing users.

| Dataset | Method | HR@5 | HR@10 | HR@30 | NDCG@5 | NDCG@10 | NDCG@30 |
|---|---|---|---|---|---|---|---|
| | Monomer | 0.2830 | 0.4264 | 0.6402 | 0.1708 | 0.2236 | 0.3083 |
| | Triple2vec | 0.2901 | 0.4354 | 0.6577 | 0.1992 | 0.2306 | 0.3125 |
| | P-Companion | 0.2752 | 0.4150 | 0.6315 | 0.1701 | 0.2199 | 0.3002 |
| | KA-CRL | 0.3175 | 0.4631 | 0.6910 | 0.2428 | 0.2933 | 0.3419 |
| Tmall | ASLI | 0.3797 | 0.4843 | 0.7009 | 0.2762 | 0.3103 | 0.3681 |
| | AICRec | **0.4875** | **0.5957** | **0.7834** | **0.3830** | **0.4180** | **0.4625** |
| | %Improv. | 28.39% | 23.00% | 11.77% | 38.66% | 34.70% | 25.64% |
| | Monomer | 0.1192 | 0.1890 | 0.4083 | 0.0941 | 0.1118 | 0.1793 |
| | Triple2vec | 0.1269 | 0.1926 | 0.4208 | 0.0982 | 0.1161 | 0.1836 |
| | P-Companion | 0.1126 | 0.1837 | 0.4015 | 0.0920 | 0.1115 | 0.1737 |
| | KA-CRL | 0.1487 | 0.2237 | 0.4515 | 0.1077 | 0.1312 | 0.1943 |
| TaoBao | ASLI | 0.1543 | 0.2231 | 0.4692 | 0.1103 | 0.1340 | 0.2001 |
| | AICRec | **0.2352** | **0.3331** | **0.5896** | **0.1678** | **0.1992** | **0.2593** |
| | %Improv. | 52.43% | 46.90% | 25.66% | 52.13% | 48.65% | 29.58% |
| | Monomer | 0.0772 | 0.1457 | 0.3780 | 0.0470 | 0.0675 | 0.1277 |
| | Triple2vec | 0.0538 | 0.1158 | 0.3212 | 0.0271 | 0.0466 | 0.0940 |
| | P-Companion | 0.0655 | 0.1282 | 0.3253 | 0.0381 | 0.0578 | 0.1179 |
| | KA-CRL | 0.0647 | 0.1353 | 0.3294 | 0.0399 | 0.0616 | 0.1218 |
| Superstore | ASLI | 0.0764 | 0.1412 | 0.3588 | 0.0385 | 0.0583 | 0.1052 |
| | AICRec | **0.1078** | **0.2047** | **0.4865** | **0.0666** | **0.0976** | **0.1628** |
| | %Improv. | 39.63% | 40.49% | 28.70% | 41.70% | 44.59% | 27.48% |

Table 2: Performance Comparison.

### 5.2. *Performance Comparison (RQ1)*

Table 2 shows the performances of AICRec and the baseline methods over the three datasets, where the best runs per metric are marked in boldface, the best runs among baseline methods are underlined, and the performance improvements (%Improv.) in percent compared with the best baseline methods are also presented in the last line of the results on each dataset. We can see that AICRec achieves a significant improvement of the recommendation performance on all the three datasets. In particular, we have the following observations on the experimental results:

1. The methods that take user individual informations into account (AICRec, Triple2vec, KA-CRL, and ASLI) achieve better recommendation performance than non-personalized methods (P-Companion and Monomer). Usu-

ally, there are a large number of items on e-commerce platforms, and users' needs become more complex. The experimental results imply that the recommendation methods without encoding of user-specific informations are difficult to meet the personalized needs of users.

2. The methods that consider session-wise relationships (AICRec, KA-CRL, ASLI) perform better than the methods that focus only on pair-wise relationships (Monomer, Triple2vec, P-Companion). This is because the session-wise context offers more high order dependence information from which the user preferences can be captured more accurately.
3. AICRec performs much better than the most recent complementary recommendation method KA-CRL. The main reason is due to the ability of AICRec to model the aspect-level complementarities between items and capture more semantics based on the self-attention mechanism.
4. At last, we can see that P-Companion does not perform well on both TaoBao and Tmall, possibly because it is designed to recommend diverse items from different categories. However, complementary items usually belong to the same category, for example, shirts and pants are in dressing category, and computers and keyboards are in electronic product category. Therefore, pursuing diversity only may impair the accuracy of complementary recommendation.

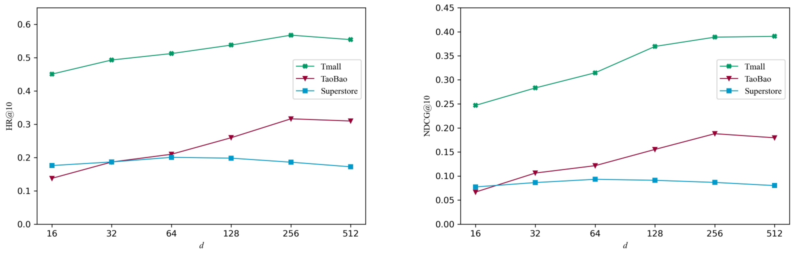
### *5.3. Ablation Experiments (RQ2)*

Now we investigate the effectiveness of the two modules of AICRec, User Intent Perceiving (UIP) and Aspect-level Complementarity Learning (ACL), by comparing AICRec with its two variants AICRec-U and AICRec-A. AICRec-U removes the UIP module by substituting the user original embedding $\boldsymbol{u}$ for the user intent embedding $\boldsymbol{z}_u$ in Equation (13), while AICRec-A removes the ACL module by substituting $\overrightarrow{\boldsymbol{e}}_v$ and $\overleftarrow{\boldsymbol{e}}_v$ for $\overrightarrow{\boldsymbol{p}}_v$ and $\overleftarrow{\boldsymbol{a}}_v$ in Equation 11, respectively. The results of ablation experiments are shown in Table 3, from which we can make the following observations:

| Dataset | Method | HR@5 | HR@10 | HR@30 | NDCG@5 | NDCG@10 | NDCG@30 |
|---|---|---|---|---|---|---|---|
| Tmall | AICRec-U | 0.4528 | 0.5577 | 0.7582 | 0.3502 | 0.3842 | 0.4318 |
| | AICRec-A | 0.3565 | 0.4709 | 0.6982 | 0.2638 | 0.3006 | 0.3543 |
| | AICRec | **0.4875** | **0.5957** | **0.7834** | **0.3830** | **0.4180** | **0.4625** |
| TaoBao | AICRec-U | 0.2287 | 0.3244 | 0.5647 | 0.1632 | 0.1890 | 0.2461 |
| | AICRec-A | 0.1609 | 0.2243 | 0.4747 | 0.1161 | 0.1405 | 0.2030 |
| | AICRec | **0.2352** | **0.3331** | **0.5896** | **0.1678** | **0.1992** | **0.2593** |
| Superstore | AICRec-U | 0.0882 | 0.1647 | 0.4294 | 0.0488 | 0.0735 | 0.1346 |
| | AICRec-A | 0.0705 | 0.1412 | 0.3647 | 0.0415 | 0.0671 | 0.1273 |
| | AICRec | **0.1078** | **0.2047** | **0.4865** | **0.0666** | **0.0976** | **0.1628** |

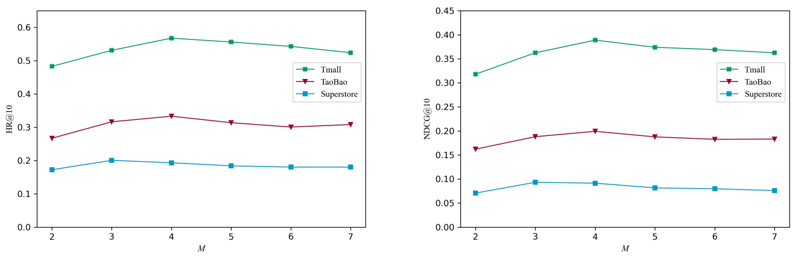
Table 3: Results of Ablation Experiments.

- Compared with AICRec, AICRec-U substantially degenerates in complementary recommendation performance on all cases. This result justifies our assumption that the complementary relationships depend on user intents, and shows that the user intent captured by AICRec can help personalize the complementary items.
- Similarly, on all cases AICRec also achieves better performances than AICRec-A. This result is consistent with our expectation that complementary recommendation will benefit from the modeling of aspect-level complementarity by which richer and finer-grained semantic information can be captured.
- Interestingly, we also note that AICRec-U performs worse than AICRec-A, which shows that the capturing of user intent plays a more important role than the modeling of aspect-level complementarity. We would not be surprised in this result if we take a close look at Equation (13), where the user intent embedding $\boldsymbol{z}_u$ works as the query vector to determine the attention coefficients of the aspect-level complementarity embeddings $\{\boldsymbol{y}^{(k)}\}$. Therefore, once the user intent embedding $\boldsymbol{z}_u$ degenerates to the original user embedding $\boldsymbol{u}$, the pooling of the aspect-level complementarity embeddings will lose the personalization offered by user intents.




(a) HR@10 (b) NDCG@10

Figure 3: Tuning of Embedding Size $d$.




(a) HR@10 (b) NDCG@10

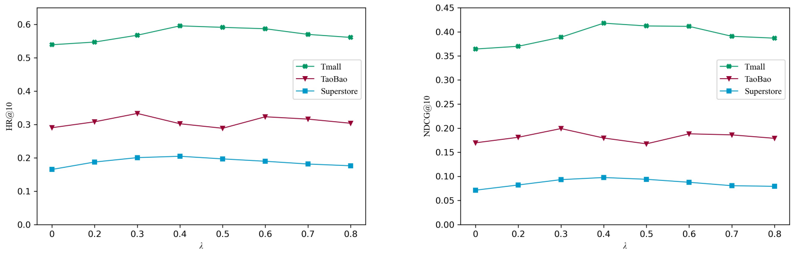
Figure 4: Tuning of the Numbers of Aspect-Encoders $M$.

### *5.4. Hyper-parameter Tuning (RQ3)*

In this section, we investigate the impact of the three hyper-parameters, the embedding size $d$, the number of aspect encoders $M$, and the orthogonal constraint coefficient $\lambda$ in Equation (16), on validation sets in terms of HR@10 and NDCG@10.

#### *5.4.1. Embedding Size d*

Figure 3 shows the results of the tuning of the embedding size $d$. We can see that basically, the curves rise first as the increasing embedding size leads to more encoding capacity, and then decline due to the overfitting incurred by




(a) Results on HR@10 (b) Results on NDCG@10

Figure 5: Tuning of Orthogonal Constraint Coefficient $\lambda$.

the excessive embedding dimensionality. According to the results, we set the embedding size $d$ to 256 on datasets Tmall and TaoBao, and 64 on Superstore.

### *5.4.2. Number of Aspect Encoders M*

Figure 4 shows the results of the tuning of the number of aspect encoders $M$. Recall that AICRec invokes the aspect encoders to decouple an item embedding into $M$ latent aspect embeddings so that we can evaluate the complementarity at aspect-level. From Figure 4 we can see that the curves rise as $M$ is increasing, i.e., factorizing into more latent aspects, which again verifies that learning complementarity at aspect-level benefits the performance of complementary recommendation. However, excessive $M$ will also cause overfitting which makes the curves go down after the optimal values of $M$. Therefore, we finally set $M$ to 4 on Tmall and TaoBao, and 3 on Superstore.

### *5.4.3. Orthogonal Constraint Coefficient λ*

Recall that the coefficient $\lambda$ in Equation (16) balances the contribution of the orthogonal constraint $\mathcal{L}_{\mathrm{a}}$ defined in Equation (8). As we can see from Figure 5, on all datasets the performances improve as $\lambda$ increases from 0, which confirms that the orthogonal constraint can serve the purpose to enforce the independence on the decoupled latent aspect embeddings for better capturing of aspect-level complementarity. However, when $\lambda$ exceeds a threshold, the performance curves

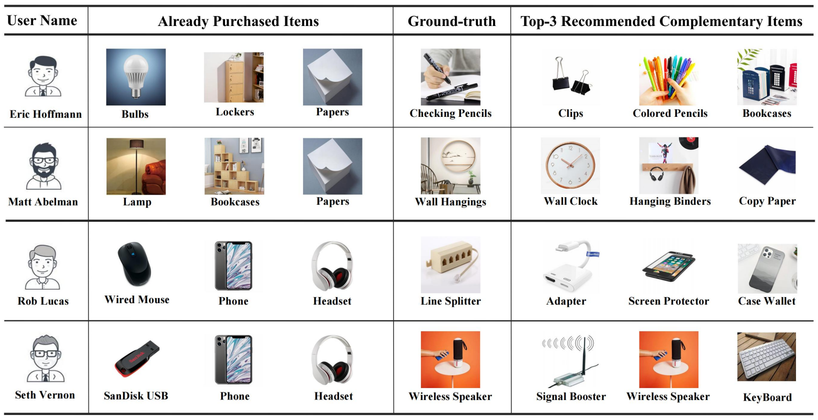


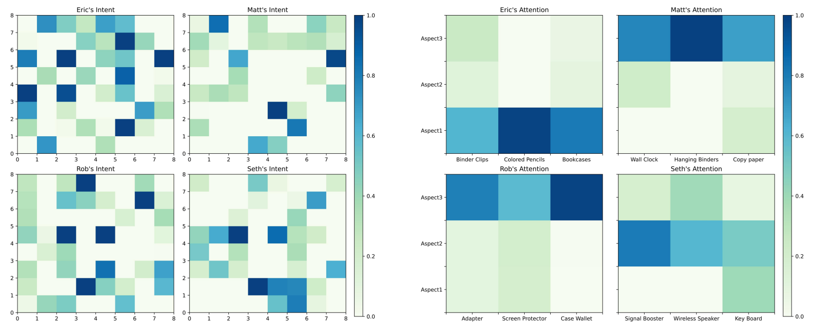

Figure 6: Case studies of AICRec's complementary recommendation.

begin to drop. This is because the excessive independence reduces the mutual information, and consequently degrades the modeling of the correlation, between the decoupled latent aspects. According to the results shown in Figure 5, we finally set $\lambda$ to 0.3 on TaoBao and 0.4 on Tmall and Superstore, so that an optimal trade-off can be achieved for the independence and correlation between the latent aspects.

### *5.5. Case Study (RQ4)*

In this section, we conduct visualizable case studies to illustrate AICRec's ability to personalize the complementary recommendation with respect to the intents of users. For this purpose, from the test set of Superstore we randomly select two pairs of users where each pair of users have similar purchasing history.

As shown in Figure 6, the first pair of users, Eric and Matt, both bought the similar items of lightings, storages and printer papers. However, the next items purchased by Eric and Matt are Checking Pencils and Wall Hangings, respectively, which shows that the intent of Eric is on office items while the intent of Matt is on home improvement items. In other words, they have different




(a) Personalized Intent Embeddings $\boldsymbol{z}_u$ (b) Aspect Attention Coefficients $\beta_k$

Figure 7: Visualization of personalized intents and aspect attention coefficients.

intents even though they have similar purchasing history. We can see that the top-3 items (Clips, Colored Pencils, and Bookcases) recommended by AICRec for Eric are of the category of office items, and most of the items (Wall Clock, Hanging Binders, and Copy Paper) recommended by AICRec for Matt are more consistent with the category of home improvement items. Similarly, the users of the second pair, Rob and Seth, have similar historical purchased items including Phone and Headset. However, the next item purchased by Rob is a phone line splitter, while that purchased by Seth is a wireless speaker, which implies that the intent of Rob is on the complements of Phone while the intent of Seth is more on the general electronics. Again we can see that the top-3 items recommended by AICRec for Rob and Seth are different so that their diverse needs can be satisfied. Basically, we argue that these cases demonstrate that AICRec is able to fulfill the personalized complementary recommendation due to its ability to capture the unique intents of the users from their historical behaviors even if they are similar.

To further explain why AICRec can make personalized complementary recommendations for users in similar scenarios, in Figure 7 we visualize the intent embeddings $\boldsymbol{z}_u$ (see Equation (4)) of the sample users and the aspect attention coefficients $\beta_k$ (see Equation (13)) of their recommended items. As the dimensionality of $\boldsymbol{z}_u$ is 64, so for clarity, we rearrange the embedding vector components to fill an $8 \times 8$ matrix column by column. As shown in Figure 7(a), the intent embeddings learned by AICRec for the sample users of each pair are obviously dissimilar from each other, which suggests that their different intents are discerned by AICRec due to the User Intent Perceiving module. At the same time, from Figure 7(b) we can observe that for a sample user, a recommended item has discrepant attention coefficients on different latent aspects because of the user's unique intent, which indicates that different aspects of an item contribute different complementarities to the user's intent. For example, AICRec discovers that for Eric, the complementarity on the latent aspect 1 should obtain more attention, while for Matt, the winner is the latent aspect 3.

## 6. Conclusion

In this paper, we propose a novel model called **A**spect-level Complementarity Learning for **I**ntent-aware **C**omplementary **Rec**ommendation (AICRec). Different from the existing methods for complementary recommendation, AICRec is able to differentiate users' intents even they are in similar scenarios, and infer an item's finer-grained complementarities to a user's intent at aspect-level, with the User Intent Perceiving (UIP) module and the Aspect-level Complementarity Learning (ACL) module, respectively. The results of extensive experiments conducted on real datasets demonstrate the superiority of AICRec over the state-of-the-art methods for complementary recommendation.

## References

[1] T. Chen, H. Yin, G. Ye, Z. Huang, Y. Wang, M. Wang, Try this instead: Personalized and interpretable substitute recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 891–900.

[2] S. Zhang, H. Yin, Q. Wang, T. Chen, H. Chen, Q. V. H. Nguyen, Inferring substitutable products with deep network embedding, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 4306–4312.

[3] J. McAuley, C. Targett, Q. Shi, A. van den Hengel, Image-based recommendations on styles and substitutes, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 43–52.

[4] J. McAuley, R. Pandey, J. Leskovec, Inferring networks of substitutable and complementary products, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 785–794.

[5] Y. Zhang, H. Lu, W. Niu, J. Caverlee, Quality-aware neural complementary item recommendation, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 77–85.

[6] Z. Yang, J. Ye, L. Wang, X. Lin, L. He, Inferring substitutable and complementary products with knowledge-aware path reasoning based on dynamic policy network, Knowledge-Based Systems 235 (2022) 107579.

[7] Y. Li, T. Chen, Z. Huang, Attribute-aware explainable complementary clothing recommendation, World Wide Web 24 (5) (2021) 1885–1901.

[8] R. He, C. Packer, J. McAuley, Learning compatibility across categories for heterogeneous item recommendation, in: 2016 IEEE 16th International Conference on Data Mining, 2016, pp. 937–942.

[9] Z. Wang, Z. Jiang, Z. Ren, J. Tang, D. Yin, A path-constrained framework for discriminating substitutable and complementary products in e-commerce, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 619–627.

[10] D. Xu, C. Ruan, J. Cho, E. Korpeoglu, S. Kumar, K. Achan, Knowledge-aware complementary product representation learning, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 681–689.

[11] C. P. Huynh, A. Ciptadi, A. Tyagi, A. Agrawal, Craft: Complementary recommendations using adversarial feature transformer, Computer Vision – ECCV 2018 Workshops abs/1804.10871 (2018) 54–66.

[12] T. Zhao, J. McAuley, M. Li, I. King, Improving recommendation accuracy using networks of substitutable and complementary products, in: 2017 International Joint Conference on Neural Networks, 2017, pp. 3649–3655.

[13] Y. Liu, Y. Tian, Y. Xu, S. Zhao, Y. Huang, Y. Fan, F. Duan, P. Guo, Tpgn: A time-preference gate network for e-commerce purchase intention recognition, Knowledge-Based Systems 220 (2021) 106920.

[14] Y. Hou, N. Yang, Y. Wu, P. S. Yu, Explainable recommendation with fusion of aspect information, World Wide Web 22 (1) (2019) 221–240.

[15] H. Chen, J. Li, Neural tensor model for learning multi-aspect factors in recommender systems, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020, pp. 2449–2455.

[16] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, S. Belongie, Learning visual clothing style with heterogeneous dyadic co-occurrences, in: Proceedings of the 2015 IEEE International Conference on Computer Vision, 2015, pp. 4642–4650.

[17] W.-C. Kang, E. Kim, J. Leskovec, C. Rosenberg, J. McAuley, Complete the look: Scene-based complementary product recommendation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10524–10533.

[18] X. Han, Z. Wu, Y.-G. Jiang, L. S. Davis, Learning fashion compatibility with bidirectional lstms, in: Proceedings of the 25th ACM International Conference on Multimedia, 2017, pp. 1078–1086.

[19] Z. Li, B. Wu, Q. Liu, L. Wu, H. Zhao, T. Mei, Learning the compositional visual coherence for complementary recommendations, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020, pp. 3536–3543.

[20] M. Zhang, J. Bockstedt, Complements and substitutes in product recommendations: The differential effects on consumers' willingness-to-pay, CEUR Workshop Proceedings 1679 (2016) 36–43.

[21] H. Zhao, L. Si, X. Li, Q. Zhang, Recommending complementary products in e-commerce push notifications with a mixture model approach, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 909–912.

[22] M. Zhang, X. Wei, X. Guo, G. Chen, Q. Wei, Identifying complements and substitutes of products: A neural network framework based on product embedding, ACM Transactions on Knowledge Discovery from Data 13 (2019) 1–29.

[23] V. Rakesh, S. Wang, K. Shu, H. Liu, Linked variational autoencoders for inferring substitutable and supplementary items, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 438–446.

[24] Y. Liu, Y. Gu, Z. Ding, J. Gao, Z. Guo, Y. Bao, W. Yan, Decoupled graph convolution network for inferring substitutable and complementary items, in: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, 2020, pp. 2621–2628.

[25] W. Zhang, Z. Chen, H. Zha, J. Wang, Learning from substitutable and

complementary relations for graph-based sequential product recommendation, ACM Transactions on Information Systems 40 (2021) 1–28.

[26] M. Wan, D. Wang, J. Liu, P. Bennett, J. McAuley, Representing and recommending shopping baskets with complementarity, compatibility, and loyalty, in: Proceedings of the 27th ACM International Conference on Information and Knoweldge Management, 2018, pp. 1133–1142.

[27] J. Hao, T. Zhao, J. Li, X. L. Dong, C. Faloutsos, Y. Sun, W. Wang, P-companion: A principled framework for diversified complementary product recommendation, in: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, 2020, pp. 2517–2524.

[28] Z. Yang, J. Ye, L. Wang, X. Lin, L. He, Inferring substitutable and complementary products with knowledge-aware path reasoning based on dynamic policy network, Knowledge-Based Systems 235 (2022) 107579.

[29] N. Zhu, J. Cao, X. Lu, H. Xiong, Learning a hierarchical intent model for next-item recommendation, ACM Transactions on Information Systems 40 (2) (2021) 1–28.

[30] T. Chen, H. Yin, H. Chen, R. Yan, Q. V. H. Nguyen, X. Li, Air: Attentional intention-aware recommender systems, in: 2019 IEEE 35th International Conference on Data Engineering, 2019, pp. 304–315.

[31] M. M. Tanjim, C. Su, E. Benjamin, D. Hu, L. Hong, J. McAuley, Attentive sequential models of latent intent for next item recommendation, in: Proceedings of the Web Conference 2020, 2020, pp. 2528–2534.

[32] N. Zhu, J. Cao, Y. Liu, Y. Yang, H. Ying, H. Xiong, Sequential modeling of hierarchical user intention and preference for next-item recommendation, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 807–815.

[33] S. Wang, L. Hu, Y. Wang, Q. Z. Sheng, M. Orgun, L. Cao, Modeling multi-purpose sessions for next-item recommendations via mixture-channel

purpose routing networks, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 3771–3777.

[34] S. Wang, L. Hu, Y. Wang, Q. Z. Sheng, M. A. Orgun, L. Cao, Intention2basket: A neural intention-driven approach for dynamic next-basket planning, in: Proceedings of the 29th International Joint Conferences on Artificial Intelligence, 2020, pp. 2333–2339.

[35] X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He, T.-S. Chua, Learning intents behind interactions with knowledge graph for recommendation, in: Proceedings of the Web Conference 2021, 2021, pp. 878–887.

[36] Y. Yang, B. Ma, J. Tan, H. Deng, H. Huang, Z. Zheng, Finn: Feedback interactive neural network for intent recommendation, in: Proceedings of the Web Conference 2021, 2021, pp. 1949–1958.

[37] Z. Liu, X. Li, Z. Fan, S. Guo, K. Achan, P. S. Yu, Basket recommendation with multi-intent translation graph neural network, in: Proceedings of 2020 IEEE International Conference on Big Data, 2020, pp. 728–737.

[38] Z. Hao, J. Hao, Z. Peng, S. Wang, P. S. Yu, X. Wang, J. Wang, Dy-hien: Dynamic evolution based deep hierarchical intention network for membership prediction, in: Proceedings of the 15th ACM International Conference on Web Search and Data Mining, 2022, pp. 363–371.

[39] T. Chen, Y. Sun, Y. Shi, L. Hong, On sampling strategies for neural network-based collaborative filtering, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 767–776.

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

[41] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations, 2015, pp. 1–13.

**Rui Wang** : Conceptualization, Methodology, Software, Writing- Original draft preparation
**Ning Yan**g : Conceptualization, Methodology, Writing- Original draft preparation, Writing- Reviewing and Editing, Supervision
**Philip S. Yu** : Conceptualization, Methodology, Writing- Original draft preparation

**Declaration of interests**

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: