Adapting User Preference to Online Feedback in Multi-round Conversational Recommendation

Kerui Xu[1,4], Jingxuan Yang[1], Jun Xu[2,3,*], Sheng Gao[1,4], Jun Guo[1], Ji-Rong Wen[2,3]
[1]School of Artificial Intelligence, Beijing University of Posts and Telecommunications
[2]Gaoling School of Artificial Intelligence, Renmin University of China
[3]Beijing Key Laboratory of Big Data Management and Analysis Methods
[4]Guizhou Provincial Key Laboratory of Public Big Data, GuiZhou University
{xukerui,yjx,gaosheng,guojun}@bupt.edu.cn,junxu@ruc.edu.cn,jirong.wen@gmail.com

ABSTRACT

This paper concerns user preference estimation in multi-round conversational recommender systems (CRS), which interacts with users by asking questions about attributes and recommending items *multiple times* in one conversation. Multi-round CRS such as EAR [14] have been proposed in which the user's online feedback at both attribute level and item level can be utilized to estimate user preference and make recommendations. Though preliminary success has been shown, existing user preference models in CRS usually use the online feedback information as independent features or training instances, overlooking the *relation* between attribute-level and item-level feedback signals. The relation can be used to more precisely identify the reasons (e.g., some certain attributes) that trigger the rejection of an item, leading to more fine-grained utilization of the feedback information. To address aforementioned issue, this paper proposes a novel preference estimation model tailored for multi-round CRS, called Feedback-guided Preference Adaptation Network (FPAN). In FPAN, two gating modules are designed to respectively adapt the original user embedding and item-level feedback, both according to the online attribute-level feedback. The gating modules utilize the fine-grained attribute-level feedback to revise the user embedding and coarse-grained item-level feedback, achieving more accurate user preference estimation by considering the relation between feedback. Experimental results on two benchmarks showed that FPAN outperformed the state-of-the-art user preference models in CRS, and the multi-round CRS can also be enhanced by using FPAN as its recommender component.

CCS CONCEPTS

• **Information systems → Recommender systems**; • **Human-centered computing → Interactive systems and tools**.

KEYWORDS

Multi-round conversational recommendation; User preference

* Corresponding author

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441791

ACM Reference Format:
Kerui Xu, Jingxuan Yang, Jun Xu, Sheng Gao, Jun Guo, Ji-Rong Wen. 2021. Adapting User Preference to Online Feedback in Multi-round Conversa-tional Recommendation. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 10 pages. https://doi.org /10.1145/3437963.3441791

1 INTRODUCTION

Conversational recommender systems (CRS), which elicit user's current preference through a multi-turn dialogue with the user, have attracted increasing research efforts in recent years. CRS resolve the difficulties of reliably estimating the user's current intent through collecting user online feedback instead of solely based on the past user-item interactions [10]. A variety of conversational recommendation task formulations have been proposed [5, 6, 14, 16, 23, 37, 38]. Among these studies, the multi-round CRS [14, 15, 17] have the ability to collect rich online feedback and achieve promising results.

A multi-round CRS usually consists of two components: the conversational component (CC) that interacts with the user, and the recommender component (RC) that estimates the user's preference [14, 23]. At each turn, CRS can *ask* whether the user likes a given attribute or *recommend* a list of items. Compared to other CRS settings (e.g., single-round CRS [23]), the conversation in multi-round CRS continues when the user rejects the recommended item list. Therefore, it can collect rich online feedback at both attribute level and item level. The attribute-level signals reflect the user's positive or negative preference to specific attributes, derived from the user's binary feedback to the action *ask*; the item-level signals reflect the user's negative preference to those recommended items, derived from the user's rejection[1] to the action *recommend*.

Utilizing these online feedback information in multi-round CRS, however, is not trivial. The EAR framework [14] adopts factorization machine (FM) [21] as the recommender component where the attribute-level feedback is encoded as the input features and the item-level feedback is treated as the training instances for online update. Though promising results have been observed, the FM model is originally developed under the static recommendation setting and overlooks the *relation* between fine-grained attribute-level feedback and coarse-grained item-level feedback collected in multi-round CRS. Item-level feedback is hard to utilize since the reason for rejection can be varied [2], indicating that though RC usually makes recommendations based on preferred attributes [10],

[1]Please note that only the negative item-level signals are considered because an acceptance to *recommend* usually ends the conversation with success.

the user may still reject the items. For example, given negative item-level feedback "reject a red iPhone", the reason for rejection could be the attribute "red color", or the attribute "Apple brand ", or both. Directly using the feedback as training instances will lower the affinity score of all attributes associated with the item. However, if the user has expressed "like red color"(i.e., positive attribute-level feedback) in the current conversation, the system could infer that "Apple brand " may be the unpreferred attribute. Hence, utilizing the relation between the item-level and attribute-level feedback can achieve more precise estimation of user preference and help to improve the performance of multi-round CRS.

In this paper, we propose a novel user preference estimation model tailored for multi-round CRS, referred to as Feedback-guided Preference Adaptation Network (FPAN). FPAN first represents the users, items, and attributes as the nodes in a heterogeneous graph and then applies Graph Neural Network to learn the node embeddings. To capture the relation between the item-level and attribute-level feedback, FPAN designs a gating module to revise the embeddings of the rejected items based on the confirmed positive attributes, deriving item representations with the user's current unpreferred features. Similarly, another gating module is designed to revise the user embedding based on the confirmed negative attributes, deriving user representation with his current preferred features. These adapted user/item representations and the embeddings of attributes mentioned in conversation are further aggregated to estimate the user's preference on attributes and items.

To evaluate the effectiveness of FPAN, we conducted experiments on two benchmark datasets: Yelp and LastFM. The experimental results showed that FPAN significantly outperforms the state-of-the-art recommendation model adopted in the current CRS. We analyzed the results and found that FPAN improved the results through 1) adapting user preference to the online feedback and 2) leveraging the relation between feedback during the adaptation. Experimental results also showed that the multi-round CRS framework EAR using FPAN as its RC outperformed the baselines, including original EAR with FM, indicating that FPAN has the ability to enhance the user experience of the whole CRS.

2 RELATED WORK

Conversational recommendation has been studied under different settings. Christakopoulou et al. [6] present a recommender system that can query user preference on items. Yu et al. [33] and Zhang et al. [34] extend the item-level feedback in natural language form. Christakopoulou et al. [5] propose a single-round CRS that allows users to select preferred topics. Zhang et al. [37] propose to recommend one item or query one attribute at each turn with bandit learning. Zhang et al. [38] and Zou et al. [40] ask user's feedback towards attributes and predict items by matching attribute query and item description without consideration of item-level feedback. Bi et al. [2], Luo et al. [18] clarify user's intent by breaking down negative item-level feedback into attribute-level. Chen et al. [3], Li et al. [16] and Zhou et al. [39] focus on natural language generation, and estimate user's preference based on entities mentioned in the dialogue. Chen et al. [4] design an incremental multitask learning framework for explainable conversational recommendation. Zhang and Balog [35] propose a simulation framework that enables large

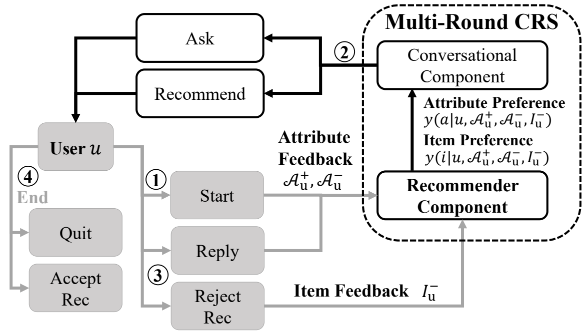


Figure 1: Workflow of multi-round CRS.

scale automatic evaluation of CRS. For a comprehensive overview of CRS, we refer interested readers to [13].

This paper focuses on user preference estimation under the multi-round conversational recommendation setting. Closely related to our work, CRM [23] considers multi-turn dialogue where the system queries attribute-level preference multiple times in one conversation and makes the recommendation at the final turn. EAR [14] extends the framework into multi-round conversational recommendation setting where the system is allowed to recommend multiple times. Li et al. [17] propose contextual Thompson Sampling to explore cold-start problem in multi-round conversational recommendation. Lei et al. [15] model conversational recommendation as an interactive path reasoning problem on a graph, but overlooking the adaptation of the model to the user's item-level feedback. In these studies, factorization machine(FM) [21] is adopted to learn user preference from the attribute-level (and item-level) feedback. Unlike previous work, we propose a novel model to capture the relation between user's feedback information.

In this paper, we apply Graph Neural Networks (GNN)[8, 12, 25] to embed the users, items, and attributes in multi-round conversational recommendation. GNN has achieved great success and are widely used in recommender systems [29]. For example, Wang et al. [30] leverages high-order connectivity to learn more informative user and item representation. Fan et al. [7], Wu et al. [31] consider social influence on user preference in social recommendation scenario. Qiu et al. [20], Wu et al. [32] construct session graph based on user's historical behavior to model current user preference in session-based recommendation. Wang et al. [26, 28] consider external item information and use GNN to encode knowledge graph.

3 BACKGROUND: MUTI-ROUND CRS

As shown in Figure 1, a multi-round CRS session starts with a preferred attribute specified by user(step ①). At each turn, the system chooses an action from $\{ask, recommend\}$(step ②): *ask* means the system asks the user whether he likes a given attribute, and the user replies with binary feedback (step ③); *recommend* means the system recommends an item list to the user, and the user examines whether his target item is contained in the list (step ③). The session ends when the user accepts the recommendation or the whole process takes too long(step ④). CRS mainly consist of the recommender component (RC) responsible for preference estimation and the conversational component (CC) responsible for user interaction.

At each turn, RC first estimates the user's preference on items and attributes considering the user's online feedback, to support the action decision of CC. Then, CC chooses to ask a selected attribute or make recommendation based on current conversation state(e.g. conversation history [23], attribute prediction from RC [14, 15]).

Formally, $\mathcal{U}$, $\mathcal{I}$ and $\mathcal{A}$ denote the set of users, items and attributes, respectively. Each item $i \in \mathcal{I}$ is associated with a set of attributes $\mathcal{A}_i \subseteq \mathcal{A}$. The historical interactions between users and items are stored in the system log, including each user $u \in \mathcal{U}$ and his interacted items $\mathcal{I}_u \subseteq \mathcal{I}$.

Suppose that a user $u \in \mathcal{U}$ starts a conversation session and his target item is $i^+ \in \mathcal{I}$. CRS collect the user's online feedback in current session, which consists of three sets: positive attribute set $\mathcal{A}_u^+ \subseteq \mathcal{A}_{i^+}$ containing the attributes that the user has given positive feedback to; negative attribute set $\mathcal{A}_u^- \subseteq \mathcal{A} \setminus \mathcal{A}_{i^+}$ containing the attributes that the user has given negative feedback to; and the negative item set $\mathcal{I}_u^- \subseteq \mathcal{I} \setminus \{i^+\}$ containing the items that the user has rejected in previous rounds. At each turn, the RC takes u's online feedback (i.e., $\mathcal{A}_u^+$, $\mathcal{A}_u^-$, and $\mathcal{I}_u^-$) as input. The output is the estimation of user's preference which indicates how likely u will prefer the given item i and attribute a, denoted as $y(i|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-)$ and $y(a|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-)$ respectively.

The estimated item preference $y(i|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-)$ can be directly used to generate the item list when CC chooses the action *recommend*. The attribute preference $y(a|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-)$ can help CC choose action. For example, in EAR [14], estimated attribute preference is encoded as part of the conversation state vector of CC. This paper focuses on RC that estimates the user's attribute preference and item preference based on his online feedback.

4 OUR APPROACH: FPAN

4.1 Model Overview

Figure 2 illustrates the architecture of the proposed Feedback-guided Preference Adaptation Network(FPAN) in which the user's preference is continuously adapted according to the online feedback information. FPAN consists of an offline representation learning module in which the initial embeddings of all the users, items, and attributes in the CRS are generated, and an online user preference adaptation module in which two gating components are used to revise item-level feedback and long-term preference, respectively. Then user's rich feedback information is aggregated to generate an adapted user preference representation. Finally, user preference on items and attributes are respectively estimated by modeling the affinity between the adapted user preference representation with the item embedding and attribute embedding.

4.2 Offline Representation Learning

FPAN learns the initial representations of users, items, and attributes based on the historical user activities collected in the log data and the relations between items and attributes. Specifically, an undirected heterogeneous tripartite graph can be constructed, consisting of three sets of nodes: users, items, and attributes, and two types of edges: user-item interactions and item-attribute relations. Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the constructed tripartite graph, where the set of nodes is denoted as $\mathcal{V} = \mathcal{U} \cup \mathcal{I} \cup \mathcal{A}$, and the set of edges $\mathcal{E}$ consists of two types of edges: the user-item edge (u, i) means the user u interacted with item i (i.e., u accepted the recommended item i at least once in the log), and the item-attribute edge (i, a) means that the item i contains the attribute a (i.e., $a \in \mathcal{A}_i$). Note that there is no edge between users and attributes, but the affinity between them can be estimated by message passing through the shared item neighbors.

GraphSAGE [8] is adopted to learn the node representations: first, each user, item or attribute is assigned with an unique node index, which is converted into d-dimensional vector representation by a node embedding matrix $\boldsymbol{H}^0 \in \mathbb{R}^{d \times |\mathcal{V}|}$ and each node's representation is denoted as $\boldsymbol{h}_v^0, \forall v \in \mathcal{V}$. Then, for each neighborhood depth k until L, GraphSAGE generates a neighborhood embedding with the aggregator function for each node and combines it with the existing embedding of the node:

$$\boldsymbol{h}_v^{k+1} = \sigma\left(\boldsymbol{W}_1^k \cdot \boldsymbol{h}_v^k + \boldsymbol{W}_2^k \cdot \frac{1}{|N(v)|} \cdot \sum_{v' \in N(v)} \boldsymbol{h}_{v'}^k\right),$$

where $\boldsymbol{h}_v^k$ denotes the k-th layer representation of node v, $N(v)$ denotes the set of v's neighbors, σ means the activate function LeakyReLU, $\boldsymbol{W}_1^k \in \mathbb{R}^{d \times d}$ and $\boldsymbol{W}_2^k \in \mathbb{R}^{d \times d}$ are trainable parameters.

To avoid the over-smoothed embedding at the last layer and capture different semantics at different layers [9], the final representation of the nodes, denoted as $\boldsymbol{e}_v$, are obtained by aggregating the representations generated at different layers:

$$\boldsymbol{e}_v = \frac{1}{L+1} \sum_{j=0}^{L} \boldsymbol{h}_v^j, \tag{1}$$

for all $v \in \mathcal{V}$. Since $\mathcal{V} = \mathcal{U} \cup \mathcal{I} \cup \mathcal{A}$, in the rest of the paper we will use $\boldsymbol{e}_u$, $\boldsymbol{e}_i$ and $\boldsymbol{e}_a$ to denote the embeddings of the user $u \in \mathcal{U}$, item $i \in \mathcal{I}$, and attribute $a \in \mathcal{A}$, respectively.

4.3 Online User Preference Adaptation

As for online interaction, suppose that a user u starts a conversation session. The user's feedback information in current session includes the set of positive attributes $\mathcal{A}_u^+$, set of negative attributes $\mathcal{A}_u^-$, and set of rejected (negative) items $\mathcal{I}_u^-$. Their corresponding embeddings are denoted as $\boldsymbol{e}_u$, $\{\boldsymbol{e}_{a^+}|a^+ \in \mathcal{A}_u^+\}$, $\{\boldsymbol{e}_{a^-}|a^- \in \mathcal{A}_u^-\}$, and $\{\boldsymbol{e}_{i^-}|i^- \in \mathcal{I}_u^-\}$. To make accurate preference estimation, the embeddings of the rejected items $\{\boldsymbol{e}_{i^-}|i^- \in \mathcal{I}_u^-\}$ and the user $\boldsymbol{e}_u$ are respectively adapted to the positive attributes $\mathcal{A}_u^+$ and negative attributes $\mathcal{A}_u^-$, both based on the gating mechanism.

4.3.1 Adapting item embedding to positive attribute feedback. As discussed in Section 1, the user usually rejects a recommended item due to only part of the attributes associated with the item. Directly utilizing the negative items in $\mathcal{I}_u^-$ (e.g., as the training instances) may influence all the associated attributes, which inevitably hurts the performance of recommendation because the rejected item also shares some attributes with the target item. Fortunately, it has been observed that the relation between item-level feedback $\mathcal{I}_u^-$ and attribute-level feedback $\mathcal{A}_u^+$ can be utilized to alleviate the aforementioned problem. In the previous "red iPhone" example, the attribute "Apple brand" may trigger the rejection if the user had explicitly expressed his preference on "red color" in previous turns.

Inspired by the observations, we propose to use the gating mechanism [19, 27] to model the relation between the item-level feedback

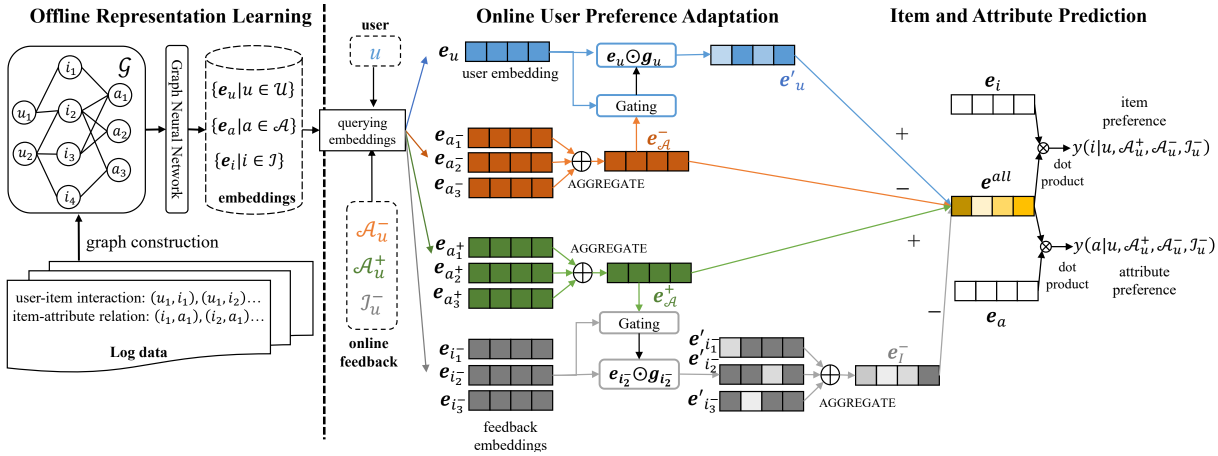


Figure 2: Workflow of FPAN. GNN and gating are used in offline representation and online user preference adaption, respectively.

and attribute-level feedback, deriving an adapted representation of the rejected item. Specifically, given the user u and the positive attribute feedback $\mathcal{A}_u^+$ provided in the conversation history, the embeddings of these attributes are first aggregated into one vector:

$$\boldsymbol{e}_{\mathcal{A}}^+ = AGGREGATE\left(\{\boldsymbol{e}_{a^+} | a^+ \in \mathcal{A}_u^+\}\right) = \frac{1}{|\mathcal{A}_u^+|} \sum_{a^+ \in \mathcal{A}_u^+} \boldsymbol{e}_{a^+}, \quad (2)$$

where $\boldsymbol{e}_{\mathcal{A}}^+$ represents positive signals at attribute level, and the operator $AGGREGATE$ refers to an aggregation function. In this paper, the average aggregation 'MEAN' is chosen.

Then, for each rejected item $i^- \in \mathcal{I}_u^-$, a gating module is applied to adapt its initial embedding $\boldsymbol{e}_{i^-}$ to the positive attribute feedback signals, achieving adapted item embedding $\boldsymbol{e'}_{i^-}$:

$$\boldsymbol{e'}_{i^-} = \boldsymbol{e}_{i^-} \odot \boldsymbol{g}_{i^-}, \quad (3)$$

where '$\odot$' denotes the element-wise product and the gating vector $\boldsymbol{g}_{i^-}$ is defined as:

$$\boldsymbol{g}_{i^-} = \sigma\left(\boldsymbol{W}_3 \cdot Concat\left(\boldsymbol{e}_{\mathcal{A}}^+, \boldsymbol{e}_{i^-}, \boldsymbol{e}_{\mathcal{A}}^+ \odot \boldsymbol{e}_{i^-}\right) + \boldsymbol{b}_3\right)$$

where '*Concat*' concatenates all of the input vectors, σ is the non-linear sigmoid function applied to each dimension, $\boldsymbol{W}_3 \in \mathbb{R}^{d \times 3d}$ is the weight matrix, and $\boldsymbol{b}_3 \in \mathbb{R}^d$ is the bias vector. Intuitively, the gating module controls the information propagated from the rejected items embeddings according to positive attribute signals.

Finally, the adapted item embeddings are further aggregated into a vector $\boldsymbol{e}_{\mathcal{I}}^-$ to represent negative signals at item level. Still, the average aggregation is used:

$$\boldsymbol{e}_{\mathcal{I}}^- = \frac{1}{|\mathcal{I}_u^-|} \sum_{i^- \in \mathcal{I}_u^-} \boldsymbol{e'}_{i^-}. \quad (4)$$

4.3.2 Adapting user embedding to negative attribute feedback. In multi-round CRS, the user's feedback in the current conversation session only reflects the user's current intent (i.e., short-term preference). The user's general interest(i.e., long-term preference), on the other hand, is usually derived from the history log data (i.e., the embedding $\boldsymbol{e}_u$ learned from the log). Balancing the user's short-term and long-term preference is important for CRS [10, 23].

In multi-round CRS, the user's negative feedback on the attributes can be utilized to balance the user's long- and short-term preference. For example, from the user's historical activity, the system derives "red iPhone" as his general interest. However, if the user has stated that he dislikes the attribute "Apple brand " in the current conversation, the general interest should be adjusted to "red phone". Based on the observation, we also propose to adapt the general interest of u (represented as $\boldsymbol{e}_u$) to the negative attribute-level feedback $\mathcal{A}_u^-$ (represented as $\{\boldsymbol{e}_{a^-} | a^- \in \mathcal{A}_u^-\}$), still based on the gating mechanism.

Similarly, given a user u and the negative attribute feedback $\mathcal{A}_u^-$ provided in the previous conversation turns, the embeddings of these attributes are aggregated with the 'MEAN' function:

$$\boldsymbol{e}_{\mathcal{A}}^- = \frac{1}{|\mathcal{A}_u^-|} \sum_{a^- \in \mathcal{A}_u^-} \boldsymbol{e}_{a^-}. \quad (5)$$

Then, a gating module is applied to adapt the initial user embedding $\boldsymbol{e}_u$ to the negative attribute feedback signals, achieving the adapted user embedding $\boldsymbol{e'}_u$:

$$\boldsymbol{e'}_u = \boldsymbol{e}_u \odot \boldsymbol{g}_u, \quad (6)$$

where the gating vector $\boldsymbol{g}_u$ is defined as

$$\boldsymbol{g}_u = \sigma\left(\boldsymbol{W}_4 \cdot Concat\left(\boldsymbol{e}_{\mathcal{A}}^-, \boldsymbol{e}_u, \boldsymbol{e}_{\mathcal{A}}^- \odot \boldsymbol{e}_u\right) + \boldsymbol{b}_4\right),$$

where $\boldsymbol{W}_4 \in \mathbb{R}^{d \times 3d}$ and $\boldsymbol{b}_4 \in \mathbb{R}^d$ refers to the weight matrix and bias vector, respectively.

4.4 Item and Attribute Prediction

We derive the user's preference representation by aggregating different kinds of feedback signals:

$$\boldsymbol{e}^{all} = \boldsymbol{e'}_u - \boldsymbol{e}_{\mathcal{I}}^- + \boldsymbol{e}_{\mathcal{A}}^+ - \boldsymbol{e}_{\mathcal{A}}^-, \quad (7)$$

where $\boldsymbol{e}_{\mathcal{A}}^+$ and $\boldsymbol{e}_{\mathcal{A}}^-$ are defined in Equation 2 and Equation 5 respectively. Please note that the signs before $\boldsymbol{e}_{\mathcal{I}}^-$ and $\boldsymbol{e}_{\mathcal{A}}^-$ are minus signs, representing the negative feedback information. Besides the derived representations with the gating mechanism, we also directly involved the attribute-level feedback $\boldsymbol{e}_{\mathcal{A}}^-$ and $\boldsymbol{e}_{\mathcal{A}}^+$, emphasizing the precise attribute preference explicitly expressed by the user.

Given an arbitrary item $i \in \mathcal{I}$, the affinity score between the user u and item i can be estimated as the dot product between item embedding $\boldsymbol{e}_i$ and the aggregated user preference representation $\boldsymbol{e}^{all}$:

$$y(i|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-) = \langle \boldsymbol{e}_i, \boldsymbol{e}^{all} \rangle. \quad (8)$$

Similarly, given an arbitrary attribute $a \in \mathcal{A}$, the affinity score between the user u and attribute a can be estimated as the dot product between the attribute embedding $\boldsymbol{e}_a$ and $\boldsymbol{e}^{all}$:

$$y(a|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-) = \langle \boldsymbol{e}_a, \boldsymbol{e}^{all} \rangle. \quad (9)$$

4.5 Model Training

The set of trainable parameters of FPAN, denoted as $\boldsymbol{\Theta}$, includes the initial embeddings of users, items, and attributes $\boldsymbol{h}_v^0, \forall v \in \mathcal{V}$, the GraphSAGE parameters for representation $\boldsymbol{W}_1^k, \boldsymbol{W}_2^k$, the gating parameters $\boldsymbol{W}_3, \boldsymbol{b}_3, \boldsymbol{W}_4$, and $\boldsymbol{b}_4$. These parameters are trained with the conversation history data [2] $\mathcal{D} = \{\mathcal{S}_k\}_{k=1}^N$, which consists of N sessions which record the past user-system interactions. The k-th session $\mathcal{S}_k = \{u_k, i_k^+, \mathcal{A}_{u_k}^+, \mathcal{A}_{u_k}^-, \mathcal{I}_{u_k}^-\}$ contains the user of the k-th session u_k, the target item i_k^+, the set of positive attributes $\mathcal{A}_{u_k}^+$, the set of negative attributes $\mathcal{A}_{u_k}^-$, and the set of rejected items $\mathcal{I}_{u_k}^-$. Following the practices in EAR [14], we adopt pairwise Bayesian Personalized Ranking objective [22]. To learn user preference on both items and attributes, the training objective consists of two loss functions: the loss w.r.t. item prediction $\mathcal{L}_{item}$ and the loss w.r.t. attribute prediction $\mathcal{L}_{att}$.

4.5.1 Loss w.r.t. Item Prediction. Given a session $\mathcal{S} = \{u, i^+, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-\} \in \mathcal{D}$, the target item i^+ is considered as a ground-truth positive sample. Like traditional BPR, the negative samples are generated from user's non-interacted items, and the loss function is defined as:

$$\mathcal{L}_{item1} = \sum_{(u,i^+,i^-) \in \mathcal{D}_1} -\ln \sigma \left(y(i^+|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-) \; - y(i^-|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-) \right),$$

where $\mathcal{D}_1 := \{(u, i^+, i^-)|i^- \in \mathcal{I} \setminus \mathcal{I}_u\}$ denotes the set of item pairs for training and σ is sigmoid function. The preferred item i^+ is the target item and the unpreferred item i^- is sampled from the set of non-interacted items of user u, which is denoted as $\mathcal{I} \setminus \mathcal{I}_u$ where $\mathcal{I}_u$ is the set of items historically interacted by user u.

Besides directly sampling the non-interacted items as the unpreferred items, we can also derive more informative unpreferred items for training considering the user feedback to attributes[14]. That is, the non-interacted items satisfying the user's current attribute requirements need to be discriminated from ground-truth item i^+:

$$\mathcal{L}_{item2} = \sum_{(u,i^+,i^-) \in \mathcal{D}_2} -\ln \sigma (y(i^+|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-) - y(i^-|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-)),$$

where $\mathcal{D}_2 := \{(u, i^+, i^-)|i^- \in \mathcal{I}_{cand} \setminus (\mathcal{I}_u \cup \mathcal{I}_u^-)\}$ contains non-interacted items in candidate item set $\mathcal{I}_{cand}$ excluding the items with explicit negative feedback, and $\mathcal{I}_{cand}$ is candidate item set containing items that satisfy user's attribute requirements in current conversation session. It has been observed that the model trained with this kind of negative samples achieves better performance of item prediction[14].

The final loss function of item prediction is:

$$\mathcal{L}_{item} = \mathcal{L}_{item1} + \mathcal{L}_{item2}$$

4.5.2 Loss w.r.t. Attribute Prediction. For attribute prediction, the model needs to rank the attributes of the target item $a \in \mathcal{A}_{i^+}$ higher than others. Therefore, given a conversation session $\mathcal{S} = \{u, i^+, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-\}$, the loss function w.r.t. attribute prediction is defined as:

$$\mathcal{L}_{att} = \sum_{(u,a^+,a^-) \in \mathcal{D}_3} -\ln \sigma (y(a^+|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-) - y(a^-|u, \mathcal{A}_u^+, \mathcal{A}_u^-, \mathcal{I}_u^-)),$$

where $\mathcal{D}_3 := \{(u, a^+, a^-)|a^+ \in \mathcal{A}_{i^+} \setminus \mathcal{A}_u^+, a^- \in \mathcal{A} \setminus (\mathcal{A}_{i^+} \cup \mathcal{A}_u^-)\}$ denotes the set of attribute pairs for training. All the attributes related to the target item i^+ excluding known positive attributes are considered as preferred attributes. As for unpreferred attributes, they are sampled from the attributes unrelated to target item i^+ excluding known negative attributes.

4.5.3 Multi-task training. To optimize the loss function of both item prediction and attribute prediction, we follow the practice in [14] and optimize the parameters Θ by performing multi-task training and the training objective is:

$$\mathcal{L} = \mathcal{L}_{item} + \mathcal{L}_{att} + \lambda \|\Theta\|^2,$$

where $\|\Theta\|^2$ is the regularizer term to avoid overfitting, and $\lambda > 0$ is the regularization parameter. Specifically, the model is iteratively optimized with $\mathcal{L}_{item}$ and $\mathcal{L}_{att}$. To accelerate the training process, dynamic negative sampling(DNS) [36] is applied to pick the negative samples that ranked highest adaptively. DNS has been known as one of the most effective samplers for BPR loss.

4.6 Discussion

FPAN is a simple yet powerful method for user preference estimation tailored for multi-round CRS. It offers several advantages compared with the existing methods such as FM.

First, FPAN utilizes gating mechanism to model the relation between item-level and attribute-level feedback and balance the user's general interest with his current intent. Therefore, the static embeddings of the rejected items and users are dynamically adapted to the user's current attribute-level feedback: (1) Existing methods directly treat the rejected items as the training instances, failing to consider that the rejection is triggered by only part of the associated attributes. The gating module of FPAN, however, utilizes the positive attribute-level feedback to discriminate the unpreferred attributes from the preferred ones and therefore derives more precise representations of negative items; (2) Similarly, the negative attribute-level feedback is used to discriminate the attributes that associated with historically interacted items but not preferred in the current conversation, still based on the gating mechanism. In this way, FPAN generates adapted user embedding that well balances the long-term and short-term preference.

Second, FPAN conducts effective learning under an end-to-end framework where the parameters of offline representation and online preference adaptation can be learned simultaneously. We extend the training framework in [14] and derive the training instances considering the user's negative feedback on attributes (i.e.,

[2] Please refer to Section 5.1.1 for details of conversation history generation.

Table 1: Statistics of Yelp and LastFM.

Dataset	#User	#Item	#Attributes	#Interactions
Yelp	27,675	70,311	590	1,368,606
LastFM	1,801	7,432	33	76,693

$\mathcal{A}_u^-$) and rejected items (i.e., $\mathcal{I}_u^-$), which enriches the training data and makes the training process closer to the online scenario.

Third, though FPAN focuses on item prediction and attribute prediction as the recommender component of CRS, the experiments also demonstrated that accurately estimating user preference can benefit the conversational component, and therefore improve the performance of the whole multi-round CRS.

5 EXPERIMENTS

We conducted extensive experiments to answer the following two research questions:

RQ1: How does the proposed FPAN perform in terms of estimating user preference on items and attributes as compared with state-of-the-art recommendation models adopted in current CRS?

RQ2: How does the multi-round CRS with FPAN perform as compared with state-of-the-art CRS?

We answered these two questions in Section 5.1 and Section 5.2, respectively. All of the experiments were conducted based on the Yelp and LastFM datasets [3]. Following the practices in [14], the users interacted with less than 10 items are removed from the datasets. Each of the datasets is split into training data (70%), validation data (20%) and test data (10%). Table 1 reports the statistics of the two datasets. The data, source code, and experimental results can be found at https://github.com/xxkkrr/FPAN.

5.1 Evaluating FPAN model (RQ1)

5.1.1 Experimental settings. **Conversation history data generation:** In this experiment, we trained and evaluated the recommendation models in multi-round CRS based on the data generated by the conversation simulation methods [14, 23, 38], to mitigate the lack of real conversation history data. Specifically, a user simulator and a rule-based agent were built and further used to simulate the user-system interaction process. The user simulator generated responses to CRS based on the target item i^+ and its related attributes $\mathcal{A}_{i^+}$. It randomly chose an attribute $a \in \mathcal{A}_{i^+}$ to start the conversation. In the following turns, it provided binary feedback to *ask* action by checking whether the target item contains the asked attribute and accepted recommendation only when the target item is included in the recommended item list. The rule-based agent was built to generate actions of CRS. At each turn, the agent chose *recommend* action with probability $\min\left(1, \frac{\text{len}}{|\mathcal{I}_{cand}|}\right)$ or chose the attribute with maximum entropy to ask, where 'len' is the length of recommended item list. The intuition behind is that the probability of *recommend* action should increase when the length of the candidate item list shrinks.

Given a user-item pair (u, i), a conversation session was simulated based on the interactions between user simulator and rule-based agent. The attributes with positive and negative feedback are recorded as $\mathcal{A}_u^+$ and $\mathcal{A}_u^-$, respectively. To simulate negative item feedback, we randomly sampled $\mathcal{I}_u^-$ from candidate item list, i.e. $\mathcal{I}_u^- \subseteq \mathcal{I}_{cand} \setminus \{i\}$. We conducted offline training and evaluation based on the conversation history data collected above.

Training details: In FPAN, the embedding size d is set to 64. Adam optimizer [11] was adopted to perform the multi-task training. The learning rate was set to 0.001 and 0.0003 for item prediction task and attribute prediction task, respectively. The regulation parameter is $\lambda = 1e - 5$ and we augment our model with 0.1 dropout. The number of layers in GraphSAGE is $L = 2$.

Baselines: FPAN was compared with the following state-of-the-arts baselines: (1)EAR-FM [14]: It's Factorization machine (FM) model [21] used in EAR, a state-of-the-art multi-round CRS framework, to predict item and attribute. To incorporate negative item feedback, EAR-FM treats rejected items as negative samples and retrains their model. (2)ConUCB [37]:It's a generalization of LinUCB [1], which incorporates conversational mechanism into contextual bandit. In this experiment, the reward of the user's positive feedback (i.e., $\mathcal{A}_u^+$) was set to 1 and that of negative feedback (i.e., $\mathcal{A}_u^-, \mathcal{I}_u^-$) was set to 0.

We also evaluated two FPAN variations to show the effectiveness of different FPAN components: (1)FPAN w/o graph: To test the effects of the user-item-attribute graph, we evaluated the performance of FPAN in which the embeddings of users, items, and attributes are learnt without graph structure(by directly looking up embedding in $\boldsymbol{H}^0$, i.e. $\boldsymbol{e}_v = \boldsymbol{h}_v^0, v \in \mathcal{V}$). (2)FPAN w/o gating: To test the effects of the gating mechanism, we evaluated the FPAN model in which the gating modules were removed, and the feedback signals and long-term preference were directly summed together.

5.1.2 Performance Comparison. Table 2 reports the performance of the attribute prediction and item prediction w.r.t. AUC score for all of the methods on Yelp and LastFM. These methods were tested under two settings: (1) given only attribute feedback $\mathcal{A}_u^+$ and $\mathcal{A}_u^-$, denoted as "$\mathcal{A}_u^+ + \mathcal{A}_u^-$" in the table; and (2) given attribute feedback together with negative items $\mathcal{A}_u^+$, $\mathcal{A}_u^-$ and $\mathcal{I}_u^-$, denoted as "$\mathcal{A}_u^+ + \mathcal{A}_u^- + \mathcal{I}_u^-$" in the table.

From the reported results, we can see that FPAN achieved better performance of preference estimation on both attributes and items, and outperformed all baseline methods on both datasets under two settings. We conducted t-tests on the improvements over the best baseline EAR-FM and all of the improvements are significant (p-value < 0.05). The results indicated FPAN's effectiveness of adapting user preference to the online feedback, capturing the user's dynamic preference in a fine-grained way. Note that ConUCB is a bandit based algorithm that focuses on sequentially recommending items; therefore, it got relatively lower AUC scores for attribute prediction.

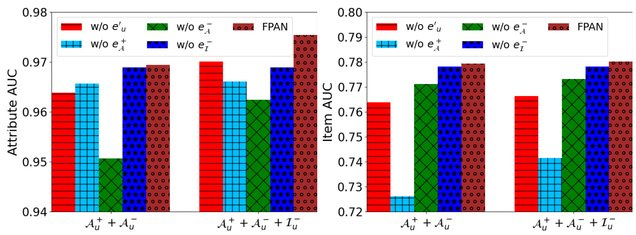
The results also showed that FPAN outperformed "FPAN w/o graph" and "FPAN w/o gating" in which the graph structure and gating modules are respectively removed. The results indicated that the importance of these two components in FPAN. Please noted that "FPAN w/o graph" and "FPAN w/o gating" outperformed EAR-FM and ConUCB, indicating the effectiveness of graph structure (for user/attribute/item embeddings) and gating mechanism (for adapting to online feedback) in user preference estimation.

5.1.3 Empirical Analysis. By comparing AUC scores in columns under $\mathcal{A}_u^+ + \mathcal{A}_u^-$(only attribute feedback) and in columns under

[3] https://www.yelp.com/dataset/ and https://grouplens.org/datasets/hetrec-2011/

Table 2: Performance comparison in terms of AUC. '$*$' indicates the improvement over EAR-FM is significant.

Dataset	Yelp				LastFM			
Setting	$\mathcal{A}_u^+ + \mathcal{A}_u^-$		$\mathcal{A}_u^+ + \mathcal{A}_u^- + \mathcal{I}_u^-$		$\mathcal{A}_u^+ + \mathcal{A}_u^-$		$\mathcal{A}_u^+ + \mathcal{A}_u^- + \mathcal{I}_u^-$	
Preference on	attributes	items	attributes	items	attributes	items	attributes	items
ConUCB	0.2701	0.6603	0.2326	0.6601	0.4808	0.3657	0.5762	0.4912
EAR-FM	0.8911	0.7382	0.8972	0.7389	0.7184	0.3729	0.7185	0.3760
FPAN w/o graph	0.9252*	0.7411*	0.9289*	0.7434*	0.7538*	0.5131*	0.7575*	0.5166*
FPAN w/o gating	0.9567*	0.7737*	0.9681*	0.7712*	0.7781*	0.5412*	0.7512*	0.5634*
FPAN (our approach)	**0.9694***	**0.7794***	**0.9754***	**0.7802***	**0.7848***	**0.6159***	**0.7852***	**0.6258***



(a) AUC of attribute prediction on Yelp (b) AUC of item prediction on Yelp

Figure 3: Ablation study on Yelp.

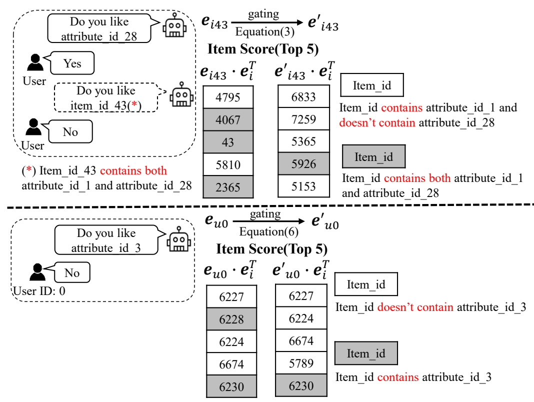


Figure 4: Two simulated conversations on LastFM showing how the gating modules work. (upper: Eq. (3), lower: Eq. (6))

$\mathcal{A}_u^+ + \mathcal{A}_u^- + \mathcal{I}_u^-$(attribute and negative item feedback), we found that introducing the negative item feedback $\mathcal{I}_u^-$ can improve the prediction accuracy not only for item prediction but also for attribute prediction. For example, on Yelp, the AUC of FPAN's item prediction was increased from 0.7794 to 0.7802 after introducing $\mathcal{I}_u^-$ to the model. At the same time, the AUC of FPAN's attribute prediction also improved from 0.9694 to 0.9754. The phenomenon can be observed for most methods on both datasets, clearly indicating the importance of utilizing $\mathcal{I}_u^-$ in multi-round CRS.

We further conducted an ablation study to show the contributions of different types of online feedback based on Yelp dataset, by removing the corresponding terms in Equation (7) (e.g., "w/o $\boldsymbol{e}'_u$" means $\boldsymbol{e}^{all}$ is calculated by setting $\boldsymbol{e}'_u = \boldsymbol{0}$ in Equation (7)). Figure 3(a) and 3(b) respectively show the AUC scores on attribute prediction and item prediction. We can see that removing any type of feedback information results in performance drop. The results also indicated that all types of online feedback are important for multi-round CRS. Among them, the positive attribute-level feedback is particularly important for item prediction since it directly reflects the characteristics of the target item.

We also conducted experiments to analyze how the gating modules in FPAN improved the preference estimation, using two simulated user-system interaction on the LastFM dataset, as shown in Figure 4. In the first example, given a user's online positive feedback on "attribute_id_28" and rejection on "item_id_43" (contains "attribute_id_1" and "attribute_id_28"), the corresponding item embedding $\boldsymbol{e}_{i_{43}}$ was adapted into $\boldsymbol{e}'_{i_{43}}$ by the gating module (Eq. 3). Comparing the top-ranked similar items by $\boldsymbol{e}_{i_{43}}$ and $\boldsymbol{e}'_{i_{43}}$ respectively, we can see that the items containing positive attribute "attribute_id_28" (shaded blocks) were ranked lower. The example showed that the gating module could identify the possible reason (attribute_id_1) that triggered the user rejection. In the second example, given a user's online negative feedback on "attribute_id_3", the corresponding user embedding $\boldsymbol{e}_{u_0}$ was adapted into $\boldsymbol{e}'_{u_0}$ by the gating module (Eq. 6). Comparing the top-ranked similar items by $\boldsymbol{e}_{u_0}$ and $\boldsymbol{e}'_{u_0}$ respectively, we can see that the items containing the negative attribute "attribute_id_3" (shaded blocks) were ranked lower. The example showed how the gating module balanced the user's long-term and short-term preference.

5.2 Evaluating multi-round CRS with FPAN (RQ2)

We tested the performance of the EAR framework in which FPAN was adopted as its RC, denoted as "EAR (FPAN)".

5.2.1 Experimental settings. **Multi-round CRS setup:** To conduct the experiments, the RC of EAR [14] was replaced by FPAN and CC was kept unchanged: the user-system interaction process was modeled as a multi-round decision making problem and solved with reinforcement learning. The conversation state $\boldsymbol{s}$ was defined as:

$$\boldsymbol{s} = \boldsymbol{s}_{ent} \oplus \boldsymbol{s}_{pre} \oplus \boldsymbol{s}_{his} \oplus \boldsymbol{s}_{len}$$

where $\oplus$ denotes concatenation, $\boldsymbol{s}_{ent}$ encodes entropy information of each attributes, $\boldsymbol{s}_{pre}$ encodes attribute preference from RC, $\boldsymbol{s}_{his}$ encodes conversation history, and $\boldsymbol{s}_{len}$ is the one-hot encoding of candidate item list length. The actions space includes (1) *ask* one

Table 3: Performance comparison of different multi-round CRS on Yelp and LastFM. '$*$' indicates the improvement over "EAR (FM)" is significant (t-tests and p-value ≤ 0.05).

Dataset	Yelp		LastFM	
Metrics	SR@15	AT	SR@15	AT
Abs Greedy	0.271	12.26	0.209	13.63
Max Entropy	0.919	5.77	0.290	13.61
CRM	0.923	5.33	0.325	13.43
EAR (FM)	0.971	4.71	0.429	12.45
Our method: EAR (FPAN)	**0.988***	**4.18***	**0.667***	**10.14***

attribute from the set $\mathcal{A}$, and (2) *recommend* an item list. A two-layer perceptron was used as the policy network to map the state into a probability distribution over action space.

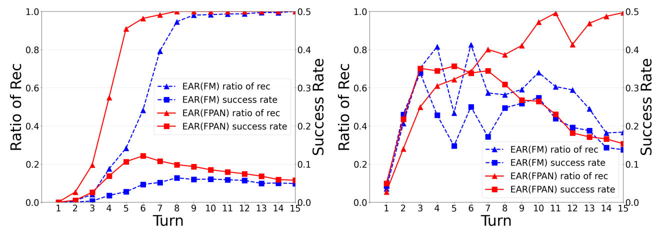
Training details: The policy network was first trained based on the rule-based agent's action strategy as initialization[14, 23]. Then, it was further optimized with Policy Gradient [24] through interactions with the user simulator. The learning rate of SGD is set to 0.001 during online training, the length of the recommended item list *len* is set to 10, and the maximum conversation turn is set as 15. For a fair comparison, the hidden layer size in the policy network and the reward setting were identical to original EAR [14].

Baselines: EAR (FPAN) was compared with the following baselines: (1) Abs Greedy [6]: It only recommends items in each turn and updates the model by incorporating item-level feedback. (2) Max Entropy: It is a rule-based system based on the max entropy criteria introduced in Section 5.1.1. (3) CRM [23]: It is a CRS originally designed under single-round setting, which estimates preference and chooses action based on conversation state encoded by a belief tracker. (4) EAR (FM) [14]: It is the state-of-the-art multi-round CRS that consists of three stages to converse with users, namely Estimation, Action and Reflection. FM is adopted as its RC.

All the systems were evaluated with the user simulator in both binary question setting and enumerated question setting. In binary question setting, CRS ask the user to give feedback to a certain attribute; in enumerated question setting, CRS provide a list of attributes and the user can reply with multiple preferred attributes. Following the practices in [14], Yelp with a manually built two-level taxonomy on the attributes was used for enumerated question setting. LastFM was used for the binary question setting. Please note that evaluation with the simulated environment is a practical and realistic approach at current stage. Though it fails to generate real human user's response and may cause problems like false rejection [15], similar settings are adopted in many previous works [14, 15, 23]. The design of human-centric evaluation is beyond the scope of this paper.

As for the evaluation metrics, we used the success rate before 15 conversation turns (SR@15) and average conversation length when the interaction process ends (average turns, AT). Note smaller AT indicates better performance.

5.2.2 Performance Comparison. Table 3 reports the performance of different CRS. From the results, we can see that our approach significantly outperformed all the baselines on both datasets in terms of SR@15 and AT. The results demonstrated that FPAN could improve the performance of CRS by using it as the RC. We noted that all methods achieved better performance on Yelp than LastFM, because in enumerated question setting multiple attributes are specified in one turn, which sharply shrink the candidate items [14].



(a) Results on LastFm dataset (b) Results on Yelp dataset

Figure 5: Ratio of CC to select action "recommend" and success rate of recommendation at different conversation turns.

We further conducted experiments to find how FPAN improved the performance of CRS. Specifically, for EAR(FPAN) and EAR(FM), we calculated the ratio of selecting action "recommend" and the success rate of item recommendation at each turn. Figure 5(a) illustrates the results of EAR (FPAN) and EAR (FM) on LastFM. We can observe that (1) at almost every turn, EAR (FPAN) has higher success rate of item recommendation, indicating that FPAN can rank the target item higher than FM; (2) at almost every turn, EAR(FPAN) has higher ratio of selecting "recommend". In other words, EAR (FPAN) usually made recommendations earlier than EAR (FM), indicating that FPAN improved the confidence of CC when making recommendations, and therefore shortened the conversation length. Similar results were also observed in the experimental results on the Yelp dataset (Figure 5(b)).

6 CONCLUSION

In this paper, we present a novel model for adapting user preference to his online feedback in multi-round conversational recommendation. The proposed model, called FPAN, makes use of GNN to learn the offline representations and two gating modules to aggregate the online feedback information considering relation between feedback signals. An end-to-end approach was designed to train the model parameters. Experiments on two benchmarks demonstrated that FPAN outperformed the state-of-the-art baselines in terms of user preference estimation. Experimental results also showed that FPAN could improve the conversation component of CRS and enhance the user experience of the CRS.

ACKNOWLEDGMENTS

This work was funded by the National Key R&D Program of China (2019YFE0198200), National Natural Science Foundation of China (61872338, 61832017), Beijing Academy of Artificial Intelligence (BAAI2019ZD0305), Beijing Outstanding Young Scientist Program (BJJWZYJH012019100020098), and the Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No. 2019BDKFJJ002).

REFERENCES

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved Algorithms for Linear Stochastic Bandits. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.* 2312–2320.

[2] Keping Bi, Qingyao Ai, Yongfeng Zhang, and W. Bruce Croft. 2019. Conversational Product Search Based on Negative Feedback. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019.* 359–368.

[3] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. 2019. Towards Knowledge-Based Recommender Dialog System. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019.* 1803–1813.

[4] Zhongxia Chen, Xiting Wang, Xing Xie, Mehul Parsana, Akshay Soni, Xiang Ao, and Enhong Chen. 2020. Towards Explainable Conversational Recommendation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020.* 2994–3000.

[5] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H. Chi. 2018. Q&R: A Two-Stage Approach toward Interactive Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018.* 139–148.

[6] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016.* 815–824.

[7] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Yihong Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019.* 417–426.

[8] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA.* 1024–1034.

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020.* 639–648.

[10] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2020. A Survey on Conversational Recommender Systems. *CoRR* abs/2004.00646 (2020).

[11] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015.*

[12] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017.*

[13] Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. 2020. Conversational Recommendation: Formulation, Methods, and Evaluation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020.* 2425–2428.

[14] Wenqianig Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-Action-Reflection: Towards Deep Interaction Between Conversational and Recommender Systems. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020.* 304–312.

[15] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive Path Reasoning on Graph for Conversational Recommendation. *CoRR* abs/2007.00194 (2020).

[16] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.* 9748–9758.

[17] Shijun Li, Wenqiang Lei, Qingyun Wu, Xiangnan He, Peng Jiang, and Tat-Seng Chua. 2020. Seamlessly Unifying Attributes and Items: Conversational Recommendation for Cold-Start Users. *CoRR* abs/2005.12979 (2020).

[18] Kai Luo, Scott Sanner, Ga Wu, Hanze Li, and Hojin Yang. 2020. Latent Linear Critiquing for Conversational Recommender Systems. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020.* 2535–2541.

[19] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical Gating Networks for Sequential Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.* 825–833.

[20] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the Item Order in Session-based Recommendation with Graph Neural Networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019.* 579–588.

[21] Steffen Rendle. 2010. Factorization Machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010.* 995–1000.

[22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009.* 452–461.

[23] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018.* 235–244.

[24] Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. 1999. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems, NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999.* 1057–1063.

[25] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018.*

[26] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.* 968–977.

[27] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2020. Beyond Clicks: Modeling Multi-Relational Item Graph for Session-Based Target Behavior Prediction. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020.* 3056–3062.

[28] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.* 950–958.

[29] Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2019. Learning and Reasoning on Graph for Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019.* 2971–2972.

[30] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019.* 165–174.

[31] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A Neural Influence Diffusion Model for Social Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019.* 235–244.

[32] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.* 346–353.

[33] Tong Yu, Yilin Shen, and Hongxia Jin. 2019. A Visual Dialog Augmented Interactive Recommender System. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.* 157–165.

[34] Ruiyi Zhang, Tong Yu, Yilin Shen, Hongxia Jin, and Changyou Chen. 2019. Text-Based Interactive Recommendation via Constraint-Augmented Reinforcement Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada.* 15188–15198.

[35] Shuo Zhang and Krisztian Balog. 2020. Evaluating Conversational Recommender Systems via User Simulation. *CoRR* abs/2006.08732 (2020).

[36] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013.* 785–788.

[37] Xiaoying Zhang, Hong Xie, Hang Li, and John C. S. Lui. 2020. Conversational Contextual Bandit: Algorithm and Application. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020.* 662–672.

[38] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards Conversational Search and Recommendation: System Ask, User Respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018.* 177–186.

[39] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion. *CoRR* abs/2007.04032 (2020).

[40] Jie Zou, Yifan Chen, and Evangelos Kanoulas. 2020. Towards Question-based Recommender Systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020.* 881–890.