Fuzzy Collaborative Reasoning

Huanhuan Yuan[1,2], Pengpeng Zhao[1*], Jiaqing Fan[1], Junhua Fang[1], Guanfeng Liu[2], Victor S. Sheng[3]

[1]Soochow University,
[2]Macquarie University,
[3]Texas Tech University
hhyuan@stu.suda.edu.cn, {ppzhao, jqfan, jhfang}@suda.edu.cn, guanfeng.liu@mq.edu.au, victor.sheng@ttu.edu

Abstract

Collaborative reasoning enhances recommendation performance by combining the strengths of symbolic learning and deep neural learning. However, current collaborative reasoning models rely on parameterized networks to simulate logical operations within the reasoning process, which (1) do not comply with all axiomatic principles of classical logic and (2) limit the model's generalizability. To address these limitations, a Fuzzy logic approach tailored for Collaborative Reasoning (FuzzCR) is proposed in this work, aiming to augment the recommendation system with cognitive abilities. Specifically, this method redefines the sequential recommendation task as a logical query answering process to facilitate a more structured and logical progression of reasoning. Moreover, learning-free fuzzy logical operations are implemented for the designed reasoning process. Taking advantage of the inherent properties of fuzzy logic, these logical operations satisfy fundamental logical rules and ensure complete reasoning. After training, these operations can be applied to flexible reasoning processes, rather than being confined to fixed computation graphs, thereby exhibiting good generalizability. Extensive experiments conducted on publicly available datasets demonstrate the superiority of this method in solving the sequential recommendation task.

Introduction

In recent years, deep learning approaches to Sequential Recommendations (SRs) have predominantly focused on learning user, item, or sequence embeddings, and determine recommendations through matching functions (Hidasi et al. 2016; Kang and McAuley 2018; Tang and Wang 2018; Qiu et al. 2022). The key assumption of these models is that a user's future behavior can be accurately predicted based on the similarity to their previous behaviors. By utilizing various deep learning techniques (e.g., self-attention), they effectively model intricate patterns and behaviors of users over time, and consequently achieve impressive performance.

However, these deep learning methods often rely on extensive samples and data training. They are effective for pattern recognition, particularly for common and widely occurring patterns in the training set. That makes these methods usually challenging to capture rare patterns in the dataset (Shi et al. 2020). Moreover, recommendation systems are designed to serve subjective human needs. The process of generating recommendations needs to have the ability to emulate human reasoning and cognitive processes, as few people choose recommended items by merely calculating their similarities in reality (Shi et al. 2020; Chen et al. 2021, 2022; Tang et al. 2023). For these reasons, some recent Collaborative Reasoning (CR) works have resorted to symbolic learning for recommendation tasks from the perspective of the cognitive process (Shi et al. 2020, 2022; Zhang et al. 2022; Chen et al. 2022; Liang et al. 2023; Yuan et al. 2023). These methods enhance recommendation performance by combining the strengths of symbolic learning and deep neural learning, enabling end-to-end training of the symbolic learning and reasoning process. In addition, they benefit from the artificial incorporation of effective logic priors into the neural network, allowing the model to achieve better results (Chen et al. 2021).

Nevertheless, current CR approaches are constrained by two limitations: (1) Existing CR methods (Shi et al. 2020; Chen et al. 2021, 2022; Liang et al. 2023) rely on parameterized networks to simulate logical operations within the reasoning process, requiring a large amount of training data containing such logical operations to learn the parameters. That restricts the generalizability of these approaches, i.e., it's hard for these models to be applied to situations that rarely appear in the training set. (2) The logical operators in these CR models do not fully satisfy basic logic rules (such as, the associative law $(\psi_1 \wedge \psi_2) \wedge \psi_3 \equiv \psi_1 \wedge (\psi_2 \wedge \psi_3)$ for logical formulae ψ_1, ψ_2, ψ_3), and are often defined ad-hoc. For example, some models (Shi et al. 2020; Chen et al. 2021, 2022) require a set of regularizers to constrain them to approximate the expected logical operations, rather than naturally satisfying logical rules as prior conditions. Box embedding based method Cbox4CR (Liang et al. 2023) can not handle logical operations such as $\vee$ and $\neg$, which limits its applicability for logic inference.

To address these limitations, we propose a Fuzzy logic model tailored for Collaborative Reasoning (FuzzCR) to endow recommendation models with cognitive abilities. Specifically, we first transform the sequential recommendation task into a First-Order Logic (FOL) query answering problem (Ren, Hu, and Leskovec 2020; Ren and Leskovec 2020; Chen, Hu, and Sun 2022), to facilitate the reasoning

[*]Corresponding author.

process. Then, we apply the conjunction, disjunction, and negation operations from the fuzzy logic system to the computation graph corresponding to the FOL query, determining the items the user is likely to interact with next. The advantages of this approach are threefold: (1) The entire reasoning process of the fuzzy logic system can be trained end-to-end, thus offering the simplicity of training similar to deep learning networks. (2) The operators in the fuzzy logic system are closed and proven to satisfy classical logical rules, ensuring complete reasoning (Chen, Hu, and Sun 2022). (3) The logical operations defined in our model are non-parametric neural networks. That means the only parameters that require training are the user, item, and relation embeddings. Once a dataset has learned good embeddings in the fuzzy logic space, these fuzzy embeddings can be applied to other logical reasoning formulas instead of a fixed computation graph, thereby providing excellent generalization performance.

In summary, our contributions are threefold:

- We successfully implement differentiable fuzzy logical operators in the recommendation system, which fully satisfy fundamental logical rules and maintain the completeness of the reasoning process.
- We leverage the properties of learning-free fuzzy logical operators to enable their application to more flexible reasoning processes, rather than with fixed computation graphs, thus enhancing the generalizability of the reasoning process.
- We demonstrate significant performance improvements over baselines on three public datasets, showcasing the effectiveness of our approach in real-world scenarios.

Related Works

Sequential Recommendation

In recent studies, various techniques such as Markov chains, Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs) are extensively used in sequential recommendation (SR) tasks. For example, FPMC (Rendle, Freudenthaler, and Schmidt-Thieme 2010) models item-to-item transitions, predicting the next item based on the user's most recent interactions. GRU4Rec (Hidasi et al. 2016) employs a multi-layered Gated Recurrent Unit (GRU) architecture to effectively capture sequential patterns. Similarly, Caser (Tang and Wang 2018) utilizes CNNs in sequential recommendation, applying different types of convolutional filters to extract hidden information from user sequences. SASRec (Kang and McAuley 2018) introduces self-attention mechanisms, significantly enhancing recommendation systems. Furthermore, recent models such as S3Rec (Zhou et al. 2020) and CL4SRec (Xie et al. 2022) integrate self-supervised learning signals to achieve superior performance. Different from them, we utilize CR model to endow SR model with symbolic reasoning capacity.

Collaborative Reasoning

Recent collaborative reasoning models advance recommender systems by integrating deep learning and symbolic learning. For example, ENRL (Shi et al. 2022) and NS-ICF (Zhang et al. 2022) generate explainable recommendations by deriving interpretable rules from user and item attributes. Other methods such as LINN (Shi et al. 2020), NCR-E (Chen et al. 2021), and GCR (Chen et al. 2022) represent logical operators using multilayer perceptrons and apply logical regularizers to reason and predict in a continuous space. SR-PLR (Yuan et al. 2023) combines deep learning with symbolic learning through a Beta embedding method. Cbox4CR (Liang et al. 2023) enhances collaborative reasoning by using box embeddings and introduces contrastive learning in CR. However, these models do not satisfy basic logic laws and restrict generalizability, which are the problems we need to solve in this work.

Logical Query Reasoning

Logical query reasoning is increasingly focused on existential first-order logical queries. To handle complex queries over Knowledge Graphs (KGs), methods have been developed to represent entities and queries as points (Arakelyan et al. 2021; Hamilton et al. 2018), regions (Choudhary et al. 2021b; Ren, Hu, and Leskovec 2020), or distributions (Ren and Leskovec 2020) in high-dimensional spaces. For example, GQE (Hamilton et al. 2018) embeds queries and entities in vector space, enabling efficient computation and flexible handling of logical operations. Query2Box (Ren, Hu, and Leskovec 2020) uses box representations to naturally handle $\exists$, $\wedge$, and $\vee$ operators, offering enhanced expressiveness and interpretability by defining geometric boundaries for queries. HypE (Choudhary et al. 2021b) models entities as hyperboloids in Poincaré ball space, supporting most logical queries except negation. BetaE (Ren and Leskovec 2020) uses Beta distributions for this purpose, and other methods employ different estimators (Choudhary et al. 2021a; Zhang et al. 2021). However, many logical reasoning rules cannot be fully satisfied in these methods.

Recent methods also investigate the application of fuzzy logic for query answering, in addition to the geometric representations as points, regions, or distributions. CQD (Arakelyan et al. 2021) uses t-norms and t-conorms from fuzzy logic for better performance in zero-shot settings. FuzzQE (Chen, Hu, and Sun 2022) and GNN-QE (Zhu et al. 2022) interpret embeddings as fuzzy sets to support logical query reasoning. However, researchers rarely consider the application of the fuzzy logic method to recommendation tasks and seldom explore its property of generalizability in recommendation scenarios.

Preliminary

In this section, we introduce some fundamental concepts of knowledge graphs and fuzzy logic reasoning to help readers better understand the methodology discussed later.

Knowledge Graphs for First-Order Logic Queries

One of the core tasks in working with KGs is to resolve complex queries that require logical reasoning, i.e., answering first-order logic queries that involve existential quantification ($\exists$), conjunction ($\wedge$), disjunction ($\vee$), and negation

($\neg$) (Chen, Hu, and Sun 2022; Liang et al. 2023). Usually, a KG is made up of a set of triples $\langle e_h, r, e_t \rangle$, where each triple indicates a relationship $r \in \mathcal{R}$ from the head entity $e_h \in \mathcal{E}$ to the tail entity $e_t \in \mathcal{E}$. And each triple can be represented as an atomic formula $r(e_h, e_t)$ in FOL, where $r \in \mathcal{R}$ denotes a binary predicate, and $e_h, e_t \in \mathcal{E}$ are its arguments. $r(e_h, e_t) =$ True when $r(e_h, e_t)$ exists.

To answer FOL queries, previous works define the Disjunctive Normal Form (DNF) of a FOL query q as:

$$q[V_x] = V_x : \exists V_1, \cdots, V_k : c_1 \vee c_2 \vee \cdots \vee c_n \quad (1)$$

where V_x represents the target variable of the query, $V_1, \cdots, V_k$ denote bound variables, and each c denotes a conjunctive query with one or more literals expressed as $c_i = v_{i1} \wedge v_{i2} \wedge \cdots \wedge v_{im}$. Each literal v denotes either an atomic formula or its negation, $v_{ij} = r(e, V)$ or $\neg r(e, V)$ or $r(V', V)$ or $\neg r(V', V)$, where $e \in \mathcal{E}$, $V \in \{V_x, V_1, \cdots, V_k\}$, $V' \in \{V_1, \cdots, V_k\}$, $V \neq V'$. Therefore, the aim of query answering becomes to find a set of answers $S_q = \{a | a \in \mathcal{E}, q[a] \text{ holds true}\}$.

Building on existing query answering methodologies (Chen, Hu, and Sun 2022; Liang et al. 2023), we incorporate the concept of *computation graph* into the recommendation task to outline the process of query reasoning. The computation graph is a directed acyclic graph that captures the structure of complex queries (illustrated in Figure 1). Beginning with anchor sets, the answer set is derived by iteratively applying operations to non-answer sets according to the directed edges in the computation graph. The types of edges in the computation graph are defined as:

1. **Relational Projection** $\mathcal{P}$**.** Given an entity set $S \subseteq \mathcal{E}$, and a relation $r \in \mathcal{R}$, projection operation maps S to another set $S' = \bigcup_{e \in S} \mathcal{P}_r(e)$, where $\mathcal{P}_r(e) \equiv \{e' \in \mathcal{E} : r(e, e') \text{ holds true}\}$.
2. **Conjunction/Intersection** $\mathcal{C}$. Given entity sets $\{S_1, \cdots, S_n\}$, the intersection operation computes logical intersection of entity sets as $\bigcap_{i=1}^{n} S_i$.
3. **Disjunction/Union** $\mathcal{D}$. Given entity sets $\{S_1, \cdots, S_n\}$, the union operation computes logical union of entity sets as $\bigcup_{i=1}^{n} S_i$.
4. **Negation/Complement** $\mathcal{N}$. Given an entity set $S \subseteq \mathcal{E}$, the complement set $\bar{S}$ is $\mathcal{E} - S$.

Fuzzy Logic Reasoning over Knowledge Graphs

Fuzzy logic (Zadeh 1965) extends Boolean logic by allowing truth values to range from 0 to 1, rather than just 0 (false) or 1 (true). In this way, it can support degrees of truth/false and provide a flexible framework for reasoning in complex and uncertain environments. Moreover, fuzzy logic ensures consistency in logical operations when truth values are 0 or 1, thereby integrating seamlessly with traditional logical systems. This extension makes fuzzy logic suitable for applications where binary true or false distinctions are inadequate, enabling it to handle uncertainty and partial truths effectively.

Additionally, fuzzy logic based models using t-norm logic systems maintain logical properties within the vector space, ensuring adherence to logical laws during reasoning (Chen, Hu, and Sun 2022). In these models, the embedding model scoring function $\phi(q, e)$ estimates the probability $I(q[e])$ that an entity e answers a query q. Here, $q[e]$ represents a logical formula where e is used to fill q, and $I(\cdot)$ denotes the truth value of the formula. Fuzzy logic based models inherently meet logical rules, such as those listed in Table 1 of FuzzQE (Chen, Hu, and Sun 2022), enhancing the effectiveness of reasoning. For example, they inherently satisfy conditions such as $\phi(q_1 \wedge q_2, e) \leq \phi(q_1, e)$ (or $I(\psi_1 \wedge \psi_2) \leq I(\psi_1)$), which reflect logical laws like the axiom $\psi_1 \wedge \psi_2 \rightarrow \psi_1$. This implies that an entity e is less likely to satisfy $\phi(q_1 \wedge q_2)$ than q_1.

The three commonly used fuzzy logic systems that possess properties satisfying logical laws are product logic, Gödel logic, and Łukasiewicz logic (Klement, Mesiar, and Pap 2013). In the context of product logic, the embeddings for $q_1 \wedge q_2$, $q_1 \vee q_2$, and $\neg q$ can be calculated as follows:

$$q_1 \wedge q_2 : \mathcal{C}(\mathbf{S}_{q_1}, \mathbf{S}_{q_2}) = \mathbf{S}_{q_1} \circ \mathbf{S}_{q_2} \quad (2)$$

$$q_1 \vee q_2 : \mathcal{D}(\mathbf{S}_{q_1}, \mathbf{S}_{q_2}) = \mathbf{S}_{q_1} + \mathbf{S}_{q_2} - \mathbf{S}_{q_1} \circ \mathbf{S}_{q_2} \quad (3)$$

$$\neg q : \mathcal{N}(\mathbf{S}_q) = \mathbf{1} - \mathbf{S}_q \quad (4)$$

where $\circ$ denotes element-wise multiplication, $\mathbf{1}$ is the all-one vector. Boldfaced notations $\mathbf{S}_q, \mathbf{S}_{q_1}, \mathbf{S}_{q_2}$ mean embeddings for entity sets S_q, S_{q_1}, S_{q_2}.

Alternatively, the conjunction and disjunction operators can be designed based on Gödel logic as follows:

$$q_1 \wedge q_2 : \mathcal{C}(\mathbf{S}_{q_1}, \mathbf{S}_{q_2}) = \min(\mathbf{S}_{q_1}, \mathbf{S}_{q_2}) \quad (5)$$

$$q_1 \vee q_2 : \mathcal{D}(\mathbf{S}_{q_1}, \mathbf{S}_{q_2}) = \max(\mathbf{S}_{q_1}, \mathbf{S}_{q_2}) \quad (6)$$

where $\min, \max$ denotes element-wise minimum and maximum operation respectively. And the operators based on Łukasiewicz logic are

$$q_1 \wedge q_2 : \mathcal{C}(\mathbf{S}_{q_1}, \mathbf{S}_{q_2}) = \max(\mathbf{S}_{q_1} + \mathbf{S}_{q_2} - 1, 0) \quad (7)$$

$$q_1 \vee q_2 : \mathcal{D}(\mathbf{S}_{q_1}, \mathbf{S}_{q_2}) = \min(\mathbf{S}_{q_1} + \mathbf{S}_{q_2}, 1) \quad (8)$$

Methodology

In this section, we detail the process of applying fuzzy logic reasoning to the sequential recommendation task and explain each component of our model.

Problem Formulation

Formally, given the user set $\mathcal{U}$, item set $\mathcal{I}$, the item $i_t^{(u)}, \in \mathcal{I}$ that $u \in \mathcal{U}$ interacts with at time step t, and the user u's interactions $S_u = [i_1^{(u)}, i_2^{(u)}, ..., i_T^{(u)}]$. Considering the problem formulation stated in (Liang et al. 2023), the task of predicting the next item that user u will interact with is formalized as a logical query: "What will the user who has interacted with items in S_u interact next?" The logical query can be expressed as a first-order logic query

$$q[I_{T+1}] = I_{T+1} : \exists U : Interacted(i_1^{(u)}, U) \wedge \cdots \wedge Interacted(i_T^{(u)}, U) \wedge Interact(U, I_{T+1}) \quad (9)$$

where I_{T+1} and U represent the target and bound variable, respectively. $Interacted(i_t^{(u)}, U)$ represents the interacted

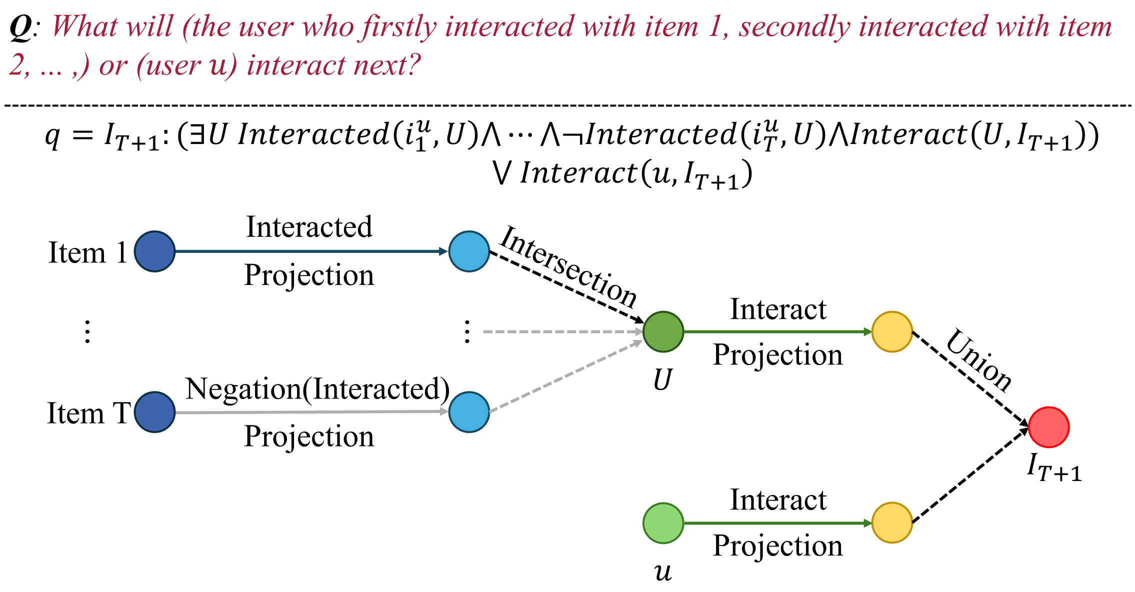
Q: What will (the user who firstly interacted with item 1, secondly interacted with item 2, ... ,) or (user u) interact next?
$q = I_{T+1}: (\exists U\ Interacted(i_1^u, U) \wedge \cdots \wedge \neg Interacted(i_T^u, U) \wedge Interact(U, I_{T+1})) \vee Interact(u, I_{T+1})$
Item 1
Interacted
Projection
Intersection
⋮
⋮
Item T
Negation(Interacted)
Projection
U
Interact
Projection
Union
I_{T+1}
Interact
Projection
u

Figure 1: The FOL query and computation graph for FuzzCR.

relation that identifies users interacted with item $i_t^{(u)}$, and $Interact(U, I_{T+1})$ represents the interact relation that finds items given users interact with. By doing this, the recommendation task is successfully formalized as a query reasoning problem.

However, making recommendations is a typical **subjective AI task**, where there is no rigid ground truth. This is different from many other purely objective AI tasks, where the ground truths are unaffected by subjective factors. For example, in the task of image classification, the models are designed to accurately predict labels such as a cat or a dog. These labels are objective facts and do not change with human feelings. In contrast, one user may like both item A and item B, while another user who liked item A may dislike item B. Hence, we consider specific user preferences in this work and represent the recommendation task to answer "What will (the user who firstly interacted with item 1, secondly interacted with item 2, $\cdots$,) or (user u) interact next?", i.e.,

$$\begin{aligned} q[I_{T+1}] = I_{T+1} : (\exists U : Interacted(i_1^{(u)}, U) \wedge \cdots \wedge \\ Interacted(i_T^{(u)}, U) \wedge Interact(U, I_{T+1})) \\ \vee Interact(u, I_{T+1}) \end{aligned} \tag{10}$$

Moreover, considering the different feedbacks, the negative interacted relations can be denoted as

$$Interacted\_N(\cdot,\cdot) = \mathcal{N}(Interacted(\cdot,\cdot)) \tag{11}$$

The corresponding FOL query and computation graph are shown in Figure 1. By applying relational projections using the three relations: the $Interacted$, $Interacted\_N$, and $Interact$, adjacent user and item entities can be identified through anchoring entities. This process enables the discovery of the target items through reasoning. Hence, the goal of logical reasoning in this work is to find the target item set S_q that satisfies the query $q[I_{T+1}]$ in Eq. (10), i.e., $i \in S_q \Leftrightarrow q[i] = \text{True}$.

Entity and Relation Embedding in Fuzzy Space

In this work, all entities (users/items) and relations are embedded into the *fuzzy space* $[0,1]^d$ (Katsaras and Liu 1977), where d is the embedding dimension. Taking item i as an example, let Ω denote the universe of all the elements, and $\{E_k\}_{k=1}^d$ denote a partition over Ω, i.e., $\Omega = \bigcup_{k=1}^d E_k$ and $E_k \cap E_j = \varnothing$ for $k \neq j$. We randomly initialize item i's fuzzy embedding matrix $\mathbf{i} \in [0,1]^d$ according to its ID number, and clamp all elements into [0,1]. Here each dimension of $\mathbf{i}$ denotes the probability whether entity i falls into the corresponding subset E_k, i.e., $\mathbf{i}(k) = \Pr(i \subseteq E_k)$. In the same way, $r \in \mathcal{R} = \{interacted, interact\}$ and $u \in \mathcal{U}$ are defined as $\mathbf{r} \in [0,1]^d$, $\mathbf{u} \in [0,1]^d$. This representation method offers two advantages: (1) It provides a probabilistic interpretation by defining the degree of membership of each element in an item's embedding as the probability of belonging to a fuzzy subset, and (2) each dimension of the embedding vector is constrained within $[0,1]$, satisfying the domain and range requirements of fuzzy logic. This allows the model to perform element-wise fuzzy conjunction, disjunction, and negation operation (Chen, Hu, and Sun 2022).

Fuzzy Logic Reasoning for Recommendation

Projection There are various projection operators as mentioned in (Ren et al. 2023). To maintain the non-parameterized nature of our model, we deviate from FuzzQE (Chen, Hu, and Sun 2022), which employs a neural network with a weight matrix as the projection operator. Instead, we opt for a simple element-wise dot projection operation. This choice not only preserves the non-parameterized property but also enhances the generalization capability of our model. Therefore, given the fuzzy embedding $\mathbf{r}$ of $r \in \mathcal{R}$ and the entity embedding $\mathbf{i}$ (or $\mathbf{u}$), the projection operator is defined as $\mathcal{P}_r(\mathbf{i}) = \mathbf{i} \circ \mathbf{r}$.

Reasoning After all the former definitions are prepared, we can describe the complete process of reasoning next item. Given the user u's interactions $s_u = [i_1^{(u)}, i_2^{(u)}, ..., i_T^{(u)}]$, corresponding fuzzy embedding $\mathbf{i}_1^{(u)}, \mathbf{i}_2^{(u)}, ..., \mathbf{i}_T^{(u)}$ and $\mathbf{u}$, we firstly formalize the next item prediction task as a FOL query answering task as mentioned in Eq. (10). And interactions with a rating score greater than a certain threshold (3 in this work) are considered positive feedback, while those below are considered negative feedback.

Secondly, the projections of interacted relations including positive and negative are applied to these item embeddings to get T user representations. For positive feedback, $\mathbf{u}_t = \mathbf{i}_t^{(u)} \circ \mathbf{r}_1$. Otherwise, $\mathbf{u}_t = \mathbf{i}_t^{(u)} \circ \mathcal{N}(\mathbf{r}_1)$, where $\mathbf{r}_1$ is the fuzzy embedding of *interacted* relation. To capture the feature of interaction order, the learnable positional embedding $\mathbf{P}_t \in \mathbb{R}^d$ is included and added with $\mathbf{u}_t$:

$$\mathbf{u}'_t = \mathbf{u}_t + \mathbf{P}_t \tag{12}$$

Then, after clamping $\mathbf{u}'_t$ into [0,1], the intersection operator can be applied on these T user representations to output the sequence's representation:

$$\mathbf{u}_{seq} = \mathcal{C}(\mathbf{u}'_1, \ldots, \mathbf{u}'_T) \tag{13}$$

Finally, we conduct the projection of *interact* relation over $\mathbf{u}_{seq}$ and disconjunct it with user fuzzy embedding $\mathbf{u}$:

$$\mathbf{S}_q = \mathcal{D}(\mathbf{u}_{seq} \circ \mathbf{r}_2, \mathbf{u} \circ \mathbf{r}_2) \tag{14}$$

where $\mathbf{S}_q$ is embedding fuzzy answer set S_q. Each dimension of $\mathbf{S}_q$ denotes the probability whether k the corresponding subset E_k is part of the answer set S_q, i.e., $\mathbf{S}_q(k) = \Pr(E_k \subseteq S_q)$. $\mathbf{r}_2$ is the fuzzy embedding of *interact* relation. More content about the choice of logic systems can be found in experiments, and we choose the product logic operators unless otherwise indicated.

Optimization

Accordingly, to determine whether item i belongs to the answer set S_q, we define the score function $\phi(q, i)$ as follows:

$$\begin{aligned} \phi(q,i) &= \mathbb{E}_{i \sim \mathbf{i}}[i \in S_q] = \\ &= \sum_{k=1}^{d} \Pr(i \in E_k) \Pr(E_k \subseteq S_q) = \mathbf{S}_q \mathbf{i}^\top \end{aligned} \tag{15}$$

We then utilize the widely used Bayesian Personalized Ranking (BPR) (Rendle et al. 2009) loss function to optimize the entire model. For a given query q:

$$\mathcal{L}_{bpr} = -\log \sigma \left(\gamma_{\text{coff}} \left(\phi(q,i) - \phi(q,i')\right)\right) \tag{16}$$

where $i \in S_q$ represents an answer to the query, $i' \notin S_q$ denotes a random negative sample, and γ_{coff} is a coefficient used to scale the difference between the positive and negative scores. This loss function inherently works to maximize $\phi(q, i)$ for $i \in S_q$ and minimize $\phi(q, i')$ for $i' \notin S_q$, thereby ensuring that relevant items are ranked higher than irrelevant ones. We also apply L_2 normalization to enhance embedding learning. Finally, the final loss function can be expressed as follows:

$$\mathcal{L} = \mathcal{L}_{bpr} + \lambda \|\mathbf{E}\|^2 \tag{17}$$

where $\mathbf{E}$ represents all the embedding parameters in our model, and λ is a coefficient that controls the influence of the L_2-regularization term.

Datasets	Users	Items	Ratings	Sparsity
ML100K	943	1,682	100,000	93.70%
Beauty	22,363	12,101	198,502	99.93%
Sports	35,598	18,357	296,337	99.95%

Table 1: Statistics of the datasets.

Experiments

In this section, we provide the details about experimental settings and results.

Experimental Setup

Datasets Our experiments are conducted on three publicly available datasets.

- **ML100k** (Harper and Konstan 2016): MovieLens is one of the most widely used datasets for recommendation. It includes 100,000 ratings provided by 943 users.
- **Amazon** (He and McAuley 2016): The Amazon dataset collection includes various e-commerce datasets crawled from the Amazon website. In our experiment, we select two relatively sparse subsets, **Beauty** and **Sports**, from the Amazon dataset for a comparative analysis.

To ensure a fair comparison, we adhere closely to the preprocessing procedures outlined in NCR (Chen et al. 2021). Specifically, for each dataset, items with ratings above 3 are treated as positive interactions, while those with ratings of 3 or below are considered negative. The last two positive interactions for each user are assigned to the validation set and test set, respectively, with the remaining historical interactions used for training. The detailed statistics of each dataset are presented in Table 1.

Baselines We compare our model with 9 baselines including two classical models with matrix factorization, one classical deep learning model, three session based models, and three CR models. The details are illustrated as follows:

- **BPRMF** (Rendle et al. 2009) optimizes the BPR ranking loss within a matrix factorization framework.
- **SVD++** (Koren 2008) extends Singular Value Decomposition (SVD) by incorporating user history interactions in user modeling.
- **NeuMF** (He et al. 2017) combines traditional matrix factorization with a multi-layer perceptron.
- **GRU4Rec** (Hidasi et al. 2016) models user sequential behaviors as a strict order by using GRU.

Methods	ML100k				Beauty				Sports			
	N@5	N@10	HR@5	HR@10	N@5	N@10	HR@5	HR@10	N@5	N@10	HR@5	HR@10
BPRMF	0.3024	0.3659	0.4501	0.6486	0.2271	0.2634	0.3198	0.4316	0.1822	0.2160	0.2591	0.3638
SVD++	0.3087	0.3685	0.4586	0.6433	0.2313	0.2727	0.3305	0.4080	0.1893	0.2307	0.2704	0.4010
NeuMF	0.3002	0.3592	0.4490	0.6316	0.2224	0.2538	0.3112	0.4077	0.1783	0.2103	0.2539	0.3531
GRU4Rec	0.3564	0.4122	0.5134	0.6856	0.2669	0.2980	0.3610	0.4573	0.2264	0.2675	0.3254	0.4530
NARM	0.3333	0.3928	0.4898	0.6742	0.2714	0.3016	0.3684	0.4623	0.2213	0.2616	0.3196	0.4450
STAMP	0.3560	0.4070	0.5159	0.6730	0.2471	0.2782	0.3333	0.4293	0.1994	0.2413	0.2889	0.4191
NLR	0.3602	0.4151	0.5102	0.6795	0.2710	0.3057	0.3781	0.4854	0.2313	0.2724	0.3288	0.4562
NCR	0.3760	0.4240	0.5456	0.6943	0.2611	0.2856	0.3666	0.4728	0.2123	0.2562	0.3111	0.4386
CBox4CR	0.4103	0.4624	0.5741	0.7353	0.2691	0.2913	0.3693	0.4799	0.2549	0.2989	0.3621	0.4988
FuzzCR	**0.4253**	**0.4719**	**0.5915**	**0.7385**	**0.2779**	**0.3114**	**0.3885**	**0.4900**	**0.2602**	**0.3044**	**0.3700**	**0.5069**
Improvement	3.65%	2.05%	3.03%	0.43%	2.39%	1.86%	2.75%	0.95%	2.08%	1.84%	2.18%	1.62%

Table 2: Performance comparison of FuzzCR and baseline models. The bold numbers highlight the best-performing results.

- **NARM** (Li et al. 2017) integrates GRU with an attention mechanism for sequence based recommendation.
- **STAMP** (Liu et al. 2018) captures users' interests by prioritizing short-term attention and memory.
- **NLR** (Shi et al. 2020) combines deep learning and logical reasoning in a logical integrated neural network.
- **NCR** (Chen et al. 2021) blends representation learning with logical reasoning by learning logical operations.
- **Cbox4CR** (Liang et al. 2023) integrates box embeddings in a contrastive learning based CR model.

Evaluation Metrics Following the approach used in NLR (Shi et al. 2020), we sample 100 negative interactions for each positive interaction and evaluate the ranking among these 101 candidates. We use Normalized discounted cumulative gain at rank K (N@K) and Hit Ratio at rank K (HR@K) as our evaluation metrics, reporting results for $K = 5, 10$. To ensure the reliability of our metrics, we repeat the experiment with different random seeds. We compute the test metrics ten times and report the mean value, as negative samples are generated randomly.

Implementation Details We conduct our implementation of all methods by using PyTorch (Paszke et al. 2017) with the Adam optimizer (Kingma and Ba 2015) on a 32GB Tesla V100-PCIE GPU. All models are trained for 100 epochs with a learning rate of 0.001, applying early stopping after 20 epochs. The L_2 regularization weight λ is set to 0.0001. Unless specified otherwise, the sequence length is configured to 5. We use a batch size of 256 and set the embedding dimension to 64. The coefficient γ_{coff} is adjusted to 15 for the Sports dataset and 25 for the other two datasets. Moreover, we choose a score of 3 out of 5 as a threshold for positive feedback in the CR model.

Overall Performance

The experimental results on three public datasets are shown in Table 2. From these results, FuzzCR achieves the best results on all three datasets by combining fuzzy logic with collaborative reasoning. The experimental results show that we can improve the performance of recommendation by benefiting from the complementary strengths of deep learning and cognitive reasoning. Additionally, we can draw the following conclusions:

Firstly, traditional methods like BPRMF, SVD++ and NeuMF perform poorly because they lack the ability to model sequential signals, which are crucial for understanding dynamic user behavior. In contrast, sequential models such as GRU4Rec, NARM and STAMP perform better by effectively incorporating temporal dependencies into their recommendations.

Secondly, among these baseline models, reasoning based approaches like NLR and CBox4CR show better performance, highlighting the importance of logical reasoning in recommendation tasks. CBox4CR outperforms the baselines by implementing logical reasoning through the use of box embeddings to model queries and entities, rather than relying on regularizations to constrain logical operations. However, they are still surpassed by FuzzCR. This may be due to the limitations in CBox4CR's ability to handle the disjunction and negation operator in a closed form, which are crucial elements in logical reasoning. For instance, the disjunction of two boxes remains separate rather than merging into one, and the negation of a box does not result in a box. Additionally, Cbox4CR includes contrastive learning, whereas FuzzCR is simpler in design.

Ablation Study

In this section, we first conduct three ablation studies, each corresponding to a different variant of our model. These variants include: removing the disjunction with user embedding as described in Eq. (10) (w/o$_{per}$), removing the negation operation in Eq. (11) (w/o$_{neg}$), and removing the position encoding in Eq. (12) (w/o$_{pos}$). Figure 2(a) compares the N@10 performance of FuzzCR with these three variants. The results show that FuzzCR outperforms the other variants, indicating that encoding user interests and sequence dynamics contributes positively to recommendation performance. Moreover, the introduction of logical negation helps to better represent user behavior by distinguishing between

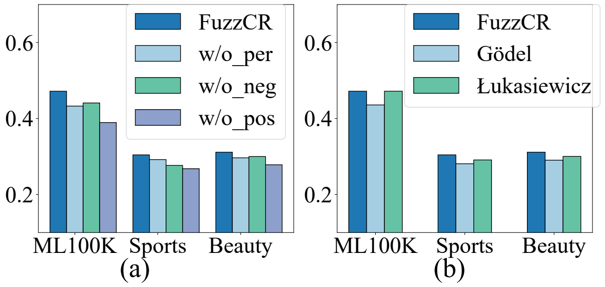

Figure 2: Ablation study. (a) FuzzCR vs Three Variants. (b) FuzzCR with Three Logic Systems.

positive and negative interactions.

Additionally, in Figure 2(b), we compare FuzzCR using different logical systems. The results indicate that product logic performs better than Gödel logic and Łukasiewicz logic. The reason is that product logic is well-suited for representing intersections, as it captures the probabilistic nature of each dimension in the fuzzy logic embedding. This alignment with the probabilistic multiplication rule ensures a more accurate physical interpretation of simultaneous events.

Generalizability of FuzzCR

To validate the generalizability of our fuzzy logic system based CR model, we conduct experiments by pre-training item, user, and relation fuzzy logic embeddings on 1p queries. Specifically, we train every triplet $Interacted(i_t^{(u)}, u)$ and $Interact(u, i_t^{(u)})$ in the train set by using fuzzy logic, focusing on answering queries like "which user will interact with $i_t^{(u)}$" and "which item will user u interact with," without considering sequence relationships. This process allows us to obtain all the necessary entity and relation embeddings. Since our logic operations are non-parametric, they theoretically generalize to all possible logical equations. In our experiments, we use the pre-trained embeddings to answer queries as defined in Eq. (10). As the results are shown in Table 3, the model pre-trained on 1p queries FuzzCR$_p$ outperforms traditional recommendation models to some extent. It is important to note that since position encoding is not trained during pre-training, the position encoding in Eq. (12) is also removed in FuzzCRp. This demonstrates the superior generalizability of the fuzzy logic embedding approach, which is also a key advantage of symbolic learning. Additionally, we test the predictive performance of FuzzCR$_p$ with different sequence lengths and compare it with four representative baselines, as illustrated in Figure 3(a). The results show that the model performs best when the sequence length is 3, suggesting that not all items in a sequence positively contribute to the recommendation performance.

Hyper-parameter Sensitivity

In this section, we examine the impact of different values of the most critical and sensitive parameter, γ_{coff}, on performance. In this experiment, we adjust γ_{coff} in Eq. (16) while keeping all other parameters constant to analyze its specific influence on the results. The γ_{coff} parameter in the BPR loss function acts as a scaling factor that adjusts the impact of the difference between the positive and negative samples on the overall loss. A larger value of γ_{coff} amplifies the difference, making the model more sensitive to even small variations, thereby accelerating the learning process. Conversely, a smaller value reduces the sensitivity, resulting in less reactive to minor differences. As shown in Figure 3(b), we find that setting γ_{coff} in the range of 20 to 30 provides the best performance, offering a balanced trade-off between sensitivity and stability in model optimization.

Datasets	Metrix	BPRMF	NeuMF	GRU4Rec	NLR	FuzzCR$_p$
ML100K	N@5	0.3024	0.3002	0.3564	0.3602	0.3644
	N@10	0.3659	0.3592	0.4122	0.4151	0.4187
	HR@5	0.4501	0.4490	0.5134	0.5102	0.5059
	HR@10	0.6486	0.6316	0.6856	0.6795	0.6737
Beauty	N@5	0.2271	0.2224	0.2669	0.2710	0.2409
	N@10	0.2634	0.2538	0.2980	0.3057	0.2801
	HR@5	0.3198	0.3112	0.3610	0.3781	0.3506
	HR@10	0.4316	0.4077	0.4573	0.4854	0.4718
Sports	N@5	0.1822	0.1783	0.2264	0.2313	0.2435
	N@10	0.2160	0.2103	0.2675	0.2724	0.2859
	HR@5	0.2591	0.2539	0.3254	0.3288	0.3475
	HR@10	0.3638	0.3531	0.4530	0.4562	0.4788

Table 3: Evaluation of FuzzCR$_p$ pre-trained on 1p queries compared to baseline models.

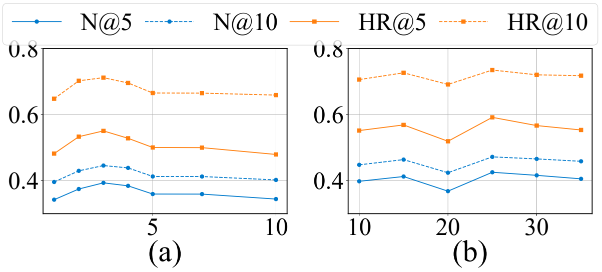

Figure 3: Impact of (a) Inference sequence length and (b) γ_{coff} in ML100K.

Conclusion

This work presents FuzzCR, a fuzzy logic model developed for collaborative reasoning in sequential recommendation tasks. It transforms the recommendation process into a first-order logic query answering problem and utilizes fuzzy logic operations to enhance reasoning capabilities. Benefiting from the learning-free fuzzy logical operations, the model adheres to fundamental logical rules and offers flexible application beyond fixed computation graphs. Extensive experimentation on multiple datasets demonstrates the model's superior performance and generalizability compared to existing methods. In future work, we could explore the integration of large language models to incorporate more diverse side-information into the framework, enabling the construction of more comprehensive queries tailored to the diverse tastes and preferences of users.

Acknowledgments

This research was partially supported by the NSFC (62376180, 62176175), the National Key Research and Development Program of China (2023YFF0725002), the major project of natural science research in Universities of Jiangsu Province (21KJA520004), the Suzhou Science and Technology Development Program (SYG202328), Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX24_3328) and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

Arakelyan, E.; Daza, D.; Minervini, P.; and Cochez, M. 2021. Complex Query Answering with Neural Link Predictors. In *ICLR*.

Chen, H.; Li, Y.; Shi, S.; Liu, S.; Zhu, H.; and Zhang, Y. 2022. Graph Collaborative Reasoning. In *WSDM*, 75–84.

Chen, H.; Shi, S.; Li, Y.; and Zhang, Y. 2021. Neural collaborative reasoning. In *WWW*, 1516–1527.

Chen, X.; Hu, Z.; and Sun, Y. 2022. Fuzzy Logic Based Logical Query Answering on Knowledge Graphs. In *AAAI*, 3939–3948.

Choudhary, N.; Rao, N.; Katariya, S.; Subbian, K.; and Reddy, C. K. 2021a. Probabilistic Entity Representation Model for Reasoning over Knowledge Graphs. In *NeurIPS*, 23440–23451.

Choudhary, N.; Rao, N.; Katariya, S.; Subbian, K.; and Reddy, C. K. 2021b. Self-Supervised Hyperboloid Representations from Logical Queries over Knowledge Graphs. In *WWW*, 1373–1384.

Hamilton, W. L.; Bajaj, P.; Zitnik, M.; Jurafsky, D.; and Leskovec, J. 2018. Embedding Logical Queries on Knowledge Graphs. In *NeurIPS*, 2030–2041.

Harper, F. M.; and Konstan, J. A. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.*, 5(4): 19:1–19:19.

He, R.; and McAuley, J. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-class Collaborative Filtering. In *WWW*, 507–517.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural Collaborative Filtering. In *WWW*, 173–182.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR*.

Kang, W.; and McAuley, J. J. 2018. Self-Attentive Sequential Recommendation. In *ICDM*, 197–206.

Katsaras, A.; and Liu, D. B. 1977. Fuzzy Vector Spaces and Fuzzy Topological Vector Spaces. *Journal of Mathematical Analysis and Applications*, 58(1): 135–146.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Klement, E. P.; Mesiar, R.; and Pap, E. 2013. *Triangular Norms*. Springer Science & Business Media.

Koren, Y. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *KDD*, 426–434.

Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural Attentive Session-based Recommendation. In *CIKM*, 1419–1428.

Liang, T.; Zhang, Y.; Di, Q.; Xia, C.; Li, Y.; and Yin, Y. 2023. Contrastive Box Embedding for Collaborative Reasoning. In *SIGIR*, 38–47.

Liu, Q.; Zeng, Y.; Mokhosi, R.; and Zhang, H. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-Based Recommendation. In *KDD*, 1831–1839.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch.

Qiu, R.; Huang, Z.; Yin, H.; and Wang, Z. 2022. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. In *WSDM*, 813–823.

Ren, H.; Galkin, M.; Cochez, M.; Zhu, Z.; and Leskovec, J. 2023. Neural Graph Reasoning: Complex Logical Query Answering Meets Graph Databases. *arXiv:2303.14617*.

Ren, H.; Hu, W.; and Leskovec, J. 2020. Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. In *ICLR*.

Ren, H.; and Leskovec, J. 2020. Beta Embeddings for Multihop Logical Reasoning in Knowledge Graphs. In *NeurIPS*, 19716–19726.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*, 452–461.

Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation. In *WWW*, 811–820.

Shi, S.; Chen, H.; Ma, W.; Mao, J.; Zhang, M.; and Zhang, Y. 2020. Neural Logic Reasoning. In *CIKM*, 1365–1374.

Shi, S.; Xie, Y.; Wang, Z.; Ding, B.; Li, Y.; and Zhang, M. 2022. Explainable Neural Rule Learning. In *WWW*, 3031–3041.

Tang, J.; and Wang, K. 2018. Personalized Top-n Sequential Recommendation via Cconvolutional Sequence Embedding. In *WSDM*, 565–573.

Tang, Z.; Floto, G.; Toroghi, A.; Pei, S.; Zhang, X.; and Sanner, S. 2023. LogicRec: Recommendation with Users' Logical Requirements. In *SIGIR*, 2129–2133.

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; and Cui, B. 2022. Contrastive Learning for Sequential Recommendation. In *ICDE*, 1259–1273.

Yuan, H.; Zhao, P.; Xian, X.; Liu, G.; Liu, Y.; Sheng, V. S.; and Zhao, L. 2023. Sequential Recommendation with Probabilistic Logical Reasoning. In *IJCAI*, 2432–2440.

Zadeh, L. A. 1965. Fuzzy sets. *Information and Control*, 8(3): 338–353.

Zhang, W.; Yan, J.; Wang, Z.; and Wang, J. 2022. Neuro-Symbolic Interpretable Collaborative Filtering for Attribute-based Recommendation. In *WWW*, 3229–3238.

Zhang, Z.; Wang, J.; Chen, J.; Ji, S.; and Wu, F. 2021. ConE: Cone Embeddings for Multi-Hop Reasoning over Knowledge Graphs. In *NeurIPS*, 19172–19183.

Zhou, K.; Wang, H.; Zhao, W. X.; Zhu, Y.; Wang, S.; Zhang, F.; Wang, Z.; and Wen, J.-R. 2020. S3-rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM*, 1893–1902.

Zhu, Z.; Galkin, M.; Zhang, Z.; and Tang, J. 2022. Neural-Symbolic Models for Logical Queries on Knowledge Graphs. In *ICML*, 27454–27478.