Abstract:

Traditional recommendation systems rely on user interaction history, leading to difficulties in grasping fine-grained and dynamic user preferences. Conversational Recommendation Systems (CRS) can tackle these issues by asking users for their preferred item attributes. However, current CRS methods only indirectly use these attributes, e.g., by updating latent user representations. This paper introduces Conversational Path Reasoning (CPR), a framework that treats CRS as an interactive path reasoning problem on a graph. 1.It uses explicit user preferences to traverse attribute vertices（nodes） and 2.utilizes the graph structure to exclude irrelevant attributes. A simpler version of CPR, called Simple CPR (SCPR), is proposed. In tests on Yelp and LastFM datasets, SCPR significantly outperformed advanced CRS methods such as EAR and CRM. This advantage increases as the number of attributes grows.

Introduction:

personalized recommendation systems typically make recommendations based on a user's historical actions,but it has limitations because it passively collects user feedback(例如被动的去等待user rating，而CRS通常是主动让user输入自己的喜好)，However, the current implementation of CRS generally maps attributes into a latent space而不是直接利用这些attribute. This process involves updating an obscure user embedding with user feedback or using preferred attributes to score items in the latent space. Some even feed user attribute preferences into a policy network to decide the next action. The paper suggests that these methods do not fully harness the potential of attribute feedback.我们来看他如何fully utilize

To achieve this, the authors propose a new framework called Conversational Path Reasoning (CPR).

一張含有 文字, 螢幕擷取畫面, 圖表, 字型 的圖片

自動產生的描述

An example is given where a user, TOM, wants music artist recommendations. The session begins at the "TOM" vertex, and he specifies an attribute "dance". The system then moves from "TOM" to "dance". The system then identifies a nearby attribute vertex to consult the user or recommends a list of items. If the user confirms a preference for the suggested attribute, the system moves(walk) to that attribute vertex. If the user rejects it or rejects a recommendation, the system stays at the same vertex and asks the user for another attribute. This process repeats until the user accepts the recommended items.

CPR advantages:

Improved explainability: CPR as an interactive path reasoning problem on a graph. Each step in the path is confirmed by the user, making the path a clear justification for the recommendation. This offers a more effective use of fine-grained attribute preferences than methods that model these preferences in a latent space.如果用latent factor modeling甚至都不知道哪些what exactly the latent factor is and how they affect the rating.

Efficient use of information: By introducing a graph structure, CPR takes advantage of a wealth of information. when the system is deciding which attribute to ask the user about next, it only considers attributes that are directly connected (adjacent) to the current point of focus in the graph., which greatly reduces the candidate space. 同时，通过user的feedback，可以修剪这个user的preference，例如修改path或attribute。

Efficacy of SCRP outperform the other’s CRS without using graph

This implementation targets multi-round conversational recommendation scenarios.The research shows that the larger the attribute space is, the more substantial the improvements that the model can achieve.

This section discusses how recommendation systems have evolved over time：

Initially, collaborative filtering hypothesis to infer user profiles，such as matrix factorization， issues : could not capturing dynamic user preferences and providing explainability

Later: using Markov models and multi-arm bandit methods, but they still lacked satisfactory explainability. Recent attention has shifted towards graph-based recommendation methods, either to better represent user/item relationships or to model recommendations as path reasoning problems on a graph.

Still have some issues: they're static and can't capture dynamic preferences, and they have high modeling complexity that necessitates(need) pruning.

Solution:graph + CRS 可解决, dynamically ask attribute questions and

make recommendations upon attribute answer is the key at current

stage of conversational recommendation. As such, we consider the

system asking user preference on attributes and making recommend dation based on those attributes in a multi-turn basis

Notation:

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

This section describes a multi-round conversational recommendation (MCR) scenario, a more complex and realistic setting for a conversational recommendation system (CRS).

1. A conversation session is initiated by a user who specifies an attribute they like (p0), for example, "I like some dance music".starting point
2. The CRS then has the freedom to ask the user about their preference for an attribute selected from a set of candidate attributes ({"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><semantics><mstyle mathsize=\"16px\"><mi mathvariant=\"normal\">&#x2119;</mi></mstyle><annotation encoding=\"application/json\">{\"x\":[[208,207,207,205,203,202,200,199,197,194,191,187,184,181,176,171,166,161,156,152,150,147,145,144,144,144,144,144,147,155,160,167,174,183,190,198,207,215,224,234,242,251,259,270,279,290,299,309,318,343,351,359,368,376,386,395,402,409,416,423,430,435,441,446,450,454,455,458,459,460,460,460,460,460,458,455,451,441,435,429,422,416,411,404,398,393,387,382,376,371,365,360,355,350,344,339,331,327,322,317,314,305,303,299,296,294,291,288,285,282,280,278,277,276,275,274,273,273,272,272,272,272,272,272,272,275,275,276,277,279,280,282,283,284,287,288],[287,287,285,285,285,284,284,283,283,283,283,282,282,281,279,278,275,275,272,271,270,269,269,269,268,268,268,267,267,266,264,263,262,261,260,259,257,255,253,251,249,248,247]],\"y\":[[175,175,175,175,175,175,175,175,175,175,173,172,171,168,165,163,160,156,152,148,145,141,137,134,130,127,123,118,115,104,100,95,90,85,81,78,76,74,72,70,68,68,67,67,67,65,65,65,65,65,65,66,68,70,73,76,80,84,88,94,99,105,112,117,124,128,134,139,144,150,156,162,168,173,180,188,193,204,208,212,216,217,220,222,224,226,227,229,230,231,232,232,232,232,232,232,229,227,223,220,217,210,208,204,200,193,187,178,170,161,153,144,138,131,124,119,113,108,102,99,95,92,89,86,83,77,76,74,72,70,69,68,67,66,65,64],[64,64,64,65,66,67,68,69,72,74,78,82,88,98,108,132,143,153,163,172,180,188,198,208,216,228,236,246,256,264,271,277,283,290,297,305,312,320,328,336,340,342,343]],\"t\":[[0,7,48,55,79,87,96,104,113,121,129,138,146,154,163,171,179,188,196,204,213,221,229,238,246,254,263,271,279,288,296,304,313,321,329,338,346,354,363,371,379,388,396,404,413,421,429,438,446,455,463,471,479,487,496,504,513,521,529,538,546,554,563,571,579,588,596,604,613,621,629,638,646,654,663,671,679,688,696,704,713,721,729,738,746,754,763,771,779,788,796,804,813,821,829,838,846,854,863,871,879,887,896,904,913,921,929,938,946,954,963,971,979,987,996,1004,1013,1021,1029,1038,1046,1054,1063,1071,1079,1087,1096,1104,1113,1121,1129,1138,1146,1159,1175,1196],[1536,1563,1571,1583,1591,1599,1607,1615,1623,1631,1639,1647,1655,1671,1679,1687,1696,1704,1713,1721,1729,1737,1746,1754,1763,1771,1779,1788,1796,1804,1813,1821,1829,1838,1846,1854,1863,1871,1879,1887,1896,1904,1929]],\"version\":\"2.0.0\"}</annotation></semantics></math>"}\_cand), or recommend items from a set of candidate items (V\_cand).
3. The user then provides feedback, either accepting or rejecting the attribute or item.
4. If the user accepts an attribute, it's added to the set of attributes preferred by the user ({"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><semantics><mstyle mathsize=\"16px\"><mi mathvariant=\"normal\">&#x2119;</mi></mstyle><annotation encoding=\"application/json\">{\"x\":[[208,207,207,205,203,202,200,199,197,194,191,187,184,181,176,171,166,161,156,152,150,147,145,144,144,144,144,144,147,155,160,167,174,183,190,198,207,215,224,234,242,251,259,270,279,290,299,309,318,343,351,359,368,376,386,395,402,409,416,423,430,435,441,446,450,454,455,458,459,460,460,460,460,460,458,455,451,441,435,429,422,416,411,404,398,393,387,382,376,371,365,360,355,350,344,339,331,327,322,317,314,305,303,299,296,294,291,288,285,282,280,278,277,276,275,274,273,273,272,272,272,272,272,272,272,275,275,276,277,279,280,282,283,284,287,288],[287,287,285,285,285,284,284,283,283,283,283,282,282,281,279,278,275,275,272,271,270,269,269,269,268,268,268,267,267,266,264,263,262,261,260,259,257,255,253,251,249,248,247]],\"y\":[[175,175,175,175,175,175,175,175,175,175,173,172,171,168,165,163,160,156,152,148,145,141,137,134,130,127,123,118,115,104,100,95,90,85,81,78,76,74,72,70,68,68,67,67,67,65,65,65,65,65,65,66,68,70,73,76,80,84,88,94,99,105,112,117,124,128,134,139,144,150,156,162,168,173,180,188,193,204,208,212,216,217,220,222,224,226,227,229,230,231,232,232,232,232,232,232,229,227,223,220,217,210,208,204,200,193,187,178,170,161,153,144,138,131,124,119,113,108,102,99,95,92,89,86,83,77,76,74,72,70,69,68,67,66,65,64],[64,64,64,65,66,67,68,69,72,74,78,82,88,98,108,132,143,153,163,172,180,188,198,208,216,228,236,246,256,264,271,277,283,290,297,305,312,320,328,336,340,342,343]],\"t\":[[0,7,48,55,79,87,96,104,113,121,129,138,146,154,163,171,179,188,196,204,213,221,229,238,246,254,263,271,279,288,296,304,313,321,329,338,346,354,363,371,379,388,396,404,413,421,429,438,446,455,463,471,479,487,496,504,513,521,529,538,546,554,563,571,579,588,596,604,613,621,629,638,646,654,663,671,679,688,696,704,713,721,729,738,746,754,763,771,779,788,796,804,813,821,829,838,846,854,863,871,879,887,896,904,913,921,929,938,946,954,963,971,979,987,996,1004,1013,1021,1029,1038,1046,1054,1063,1071,1079,1087,1096,1104,1113,1121,1129,1138,1146,1159,1175,1196],[1536,1563,1571,1583,1591,1599,1607,1615,1623,1631,1639,1647,1655,1671,1679,1687,1696,1704,1713,1721,1729,1737,1746,1754,1763,1771,1779,1788,1796,1804,1813,1821,1829,1838,1846,1854,1863,1871,1879,1887,1896,1904,1929]],\"version\":\"2.0.0\"}</annotation></semantics></math>"}\_u), and removed from the set of candidate attributes ({"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><semantics><mstyle mathsize=\"16px\"><mi mathvariant=\"normal\">&#x2119;</mi></mstyle><annotation encoding=\"application/json\">{\"x\":[[208,207,207,205,203,202,200,199,197,194,191,187,184,181,176,171,166,161,156,152,150,147,145,144,144,144,144,144,147,155,160,167,174,183,190,198,207,215,224,234,242,251,259,270,279,290,299,309,318,343,351,359,368,376,386,395,402,409,416,423,430,435,441,446,450,454,455,458,459,460,460,460,460,460,458,455,451,441,435,429,422,416,411,404,398,393,387,382,376,371,365,360,355,350,344,339,331,327,322,317,314,305,303,299,296,294,291,288,285,282,280,278,277,276,275,274,273,273,272,272,272,272,272,272,272,275,275,276,277,279,280,282,283,284,287,288],[287,287,285,285,285,284,284,283,283,283,283,282,282,281,279,278,275,275,272,271,270,269,269,269,268,268,268,267,267,266,264,263,262,261,260,259,257,255,253,251,249,248,247]],\"y\":[[175,175,175,175,175,175,175,175,175,175,173,172,171,168,165,163,160,156,152,148,145,141,137,134,130,127,123,118,115,104,100,95,90,85,81,78,76,74,72,70,68,68,67,67,67,65,65,65,65,65,65,66,68,70,73,76,80,84,88,94,99,105,112,117,124,128,134,139,144,150,156,162,168,173,180,188,193,204,208,212,216,217,220,222,224,226,227,229,230,231,232,232,232,232,232,232,229,227,223,220,217,210,208,204,200,193,187,178,170,161,153,144,138,131,124,119,113,108,102,99,95,92,89,86,83,77,76,74,72,70,69,68,67,66,65,64],[64,64,64,65,66,67,68,69,72,74,78,82,88,98,108,132,143,153,163,172,180,188,198,208,216,228,236,246,256,264,271,277,283,290,297,305,312,320,328,336,340,342,343]],\"t\":[[0,7,48,55,79,87,96,104,113,121,129,138,146,154,163,171,179,188,196,204,213,221,229,238,246,254,263,271,279,288,296,304,313,321,329,338,346,354,363,371,379,388,396,404,413,421,429,438,446,455,463,471,479,487,496,504,513,521,529,538,546,554,563,571,579,588,596,604,613,621,629,638,646,654,663,671,679,688,696,704,713,721,729,738,746,754,763,771,779,788,796,804,813,821,829,838,846,854,863,871,879,887,896,904,913,921,929,938,946,954,963,971,979,987,996,1004,1013,1021,1029,1038,1046,1054,1063,1071,1079,1087,1096,1104,1113,1121,1129,1138,1146,1159,1175,1196],[1536,1563,1571,1583,1591,1599,1607,1615,1623,1631,1639,1647,1655,1671,1679,1687,1696,1704,1713,1721,1729,1737,1746,1754,1763,1771,1779,1788,1796,1804,1813,1821,1829,1838,1846,1854,1863,1871,1879,1887,1896,1904,1929]],\"version\":\"2.0.0\"}</annotation></semantics></math>"}\_cand). The set of candidate items (V\_cand) is then updated to only include items that contain all the attributes confirmed by the user in the session.( V*cand* to V*cand* ∩ V*p)*
5. If the user rejects an attribute, it's removed from the set of candidate attributes ({"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><semantics><mstyle mathsize=\"16px\"><mi mathvariant=\"normal\">&#x2119;</mi></mstyle><annotation encoding=\"application/json\">{\"x\":[[208,207,207,205,203,202,200,199,197,194,191,187,184,181,176,171,166,161,156,152,150,147,145,144,144,144,144,144,147,155,160,167,174,183,190,198,207,215,224,234,242,251,259,270,279,290,299,309,318,343,351,359,368,376,386,395,402,409,416,423,430,435,441,446,450,454,455,458,459,460,460,460,460,460,458,455,451,441,435,429,422,416,411,404,398,393,387,382,376,371,365,360,355,350,344,339,331,327,322,317,314,305,303,299,296,294,291,288,285,282,280,278,277,276,275,274,273,273,272,272,272,272,272,272,272,275,275,276,277,279,280,282,283,284,287,288],[287,287,285,285,285,284,284,283,283,283,283,282,282,281,279,278,275,275,272,271,270,269,269,269,268,268,268,267,267,266,264,263,262,261,260,259,257,255,253,251,249,248,247]],\"y\":[[175,175,175,175,175,175,175,175,175,175,173,172,171,168,165,163,160,156,152,148,145,141,137,134,130,127,123,118,115,104,100,95,90,85,81,78,76,74,72,70,68,68,67,67,67,65,65,65,65,65,65,66,68,70,73,76,80,84,88,94,99,105,112,117,124,128,134,139,144,150,156,162,168,173,180,188,193,204,208,212,216,217,220,222,224,226,227,229,230,231,232,232,232,232,232,232,229,227,223,220,217,210,208,204,200,193,187,178,170,161,153,144,138,131,124,119,113,108,102,99,95,92,89,86,83,77,76,74,72,70,69,68,67,66,65,64],[64,64,64,65,66,67,68,69,72,74,78,82,88,98,108,132,143,153,163,172,180,188,198,208,216,228,236,246,256,264,271,277,283,290,297,305,312,320,328,336,340,342,343]],\"t\":[[0,7,48,55,79,87,96,104,113,121,129,138,146,154,163,171,179,188,196,204,213,221,229,238,246,254,263,271,279,288,296,304,313,321,329,338,346,354,363,371,379,388,396,404,413,421,429,438,446,455,463,471,479,487,496,504,513,521,529,538,546,554,563,571,579,588,596,604,613,621,629,638,646,654,663,671,679,688,696,704,713,721,729,738,746,754,763,771,779,788,796,804,813,821,829,838,846,854,863,871,879,887,896,904,913,921,929,938,946,954,963,971,979,987,996,1004,1013,1021,1029,1038,1046,1054,1063,1071,1079,1087,1096,1104,1113,1121,1129,1138,1146,1159,1175,1196],[1536,1563,1571,1583,1591,1599,1607,1615,1623,1631,1639,1647,1655,1671,1679,1687,1696,1704,1713,1721,1729,1737,1746,1754,1763,1771,1779,1788,1796,1804,1813,1821,1829,1838,1846,1854,1863,1871,1879,1887,1896,1904,1929]],\"version\":\"2.0.0\"}</annotation></semantics></math>"}\_cand).
6. If the user rejects a recommended item, it's removed from the set of candidate items (V\_cand).
7. The CRS then selects the next action (either asking about an attribute or recommending an item) based on the updated sets, and repeats the process.
8. The conversation session ends when the CRS recommends items that the user prefers, or when the maximum number of turns (T) has been reached.

Pseudocode:  
一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

two assumptions the MCR scenario makes:

It assumes the user expresses their preferences clearly and unequivocally, and there are enough items in the dataset containing the preferred attributes. Given this, the CRS considers attributes accepted by the user as a strong indication of preference. The CRS therefore prioritizes items that contain all the user-accepted attributes, which is a practical simplification due to the availability of such items.

The MCR scenario doesn't handle strong negative feedbackr. That is, if a user rejects an attribute, the CRS doesn't differentiate whether the user is indifferent to it or actively dislikes it. Due to the difficulty in obtaining such negative feedback in real-world data, for simplicity, the CRS treats all rejected attributes as 'does not care' and simply removes them from the candidate set.

Within this framework, several key research questions arise:

Which items should be recommended?

Which attribute should be asked about?

When should attributes be asked and when should recommendations be made?

The next part of the paper will explain how the proposed method offers benefits in addressing these questions.

**4 PROPOSED METHODS**

Conversational Path Reasoning (CPR) frame work：

a graph G is defined as a set of triplets {(h,r,t)} , indicating a certain relation r exists between the head entity h and the tail entity t. In this paper, we consider the graph containing three types of entities, namely, user u, item v, and attribute p

In the CPR framework, an active path refers to a sequence of attribute vertices that the system traverses, representing the attributes confirmed by the user during the conversation. The system "walks" this path by moving from one attribute vertex to another, based on the user's preferences.

CPR treats these attributes as the user's preference feedback, and uses this feedback to guide its path walking. The path is updated in chronological order of the user's confirmed preferences.

It's important to note that the CPR in this paper doesn't revisit attributes that have already been visited, and it doesn't consider the directions of edges. Additionally, it only walks over attribute vertices, not all types of vertices。

Assume the current active path is P = p0, p1, p2, ..., pt . The system stays at pt and is going to find the next attribute vertex to walk. This process can be decomposed into three steps: reasoning, consultation and transition.

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述

In reasoning step（主要是为了计算出candidate attributes和item的score）: solving the problem of which items to recommend and which attribute to ask when an attribute is initialized or confirmed by the user. The scoring process is designed as a message propagation on the graph, and attributes and items两个都要score are scored in an interdependent way. This is done through an alternating optimization strategy in an asynchronous(异步) manner。

for example, when a user expresses preference for an attribute, messages from this attribute propagate to directly connected items to score these items V\_cand. The scoring function for each item is :

一張含有 字型, 書法, 印刷術, 筆跡 的圖片

自動產生的描述

Second, the candidate items in turn propagate messages to the candidate attributes. The scores in the first step provide additional information to identify suitable attributes to ask the user about.

natural constraint of graph structure:considering only the transition to the adjacent attributes aa\_t

Finally, for a candidate attribute p ∈ Pcand , its score is calculated by propagating messages from the candidate items Vcand :

一張含有 字型, 筆跡, 印刷術, 書法 的圖片

自動產生的描述

Consultation（主要为了determine whether to ask about an attribute or to recommend items）：

with the aim of arriving at successful recommendations in the fewest number of turns.The consultation step is treated as a reinforcement learning (RL) problem.

A policy function, π(s), is designed to make this decision based on the global dialogue state s，

The output action space of the policy function contains two choices: a\_ask or a\_rec ,

If the RL decision is aask, the attribute with the highest score s\_p from the candidate attributes P\_cand is directly chosen. If the decision is arec, the top-k items from the candidate items V\_cand are recommended based on their scores s\_v.

跟别的CRS不一样的是，他们用RL来determine which attribute to ask,所以他们的action比我们多，而in this paper is determined whether to ask attributes,所以action space只有ask跟recommend。

Transition（根据user的response来update P\_cand 和 V\_cand）

The transition step will be triggered after the user confirms an asked attribute pt， CPR first performs walking from the last confirmed attribute pt−1 to pt, forming an extended path P = p0, p1, ...pt−1, pt。

Then, we add pt to the preferred attribute set Pu . Accordingly, the candidate attribute set is updated by Pcand = AAt \ (Pu ∪ Prej), and the candidate item set Vcand is updated by keeping the items that directly link to all attributes in the updated Pu。if an attribute is rejected by the user, we just remove it from the candidate attribute set (line 9 of Algorithm 1)

applied CPR framwork to SCPR

Reasoning - Item Scoring:

一張含有 字型, 白色, 文字, 筆跡 的圖片

自動產生的描述 where u, v and p denote the embedding of the user u and item v and attribute p, The first term models the message propagated from the user to the item, and the second term models the messages propagated from user-preferred attributes Pu to the item.

Reasoning - Attribute Scoring:

一張含有 文字, 字型, 行, 筆跡 的圖片

自動產生的描述 where σ is the sigmoid function to normalize the item score sv to (0, 1), Vcand denotes the candidate items, and Vp denotes the items that include the attribute p. Different from the standard entropy which treats each item equally, our weighted entropy employed here assign higher weights to the important items (i.e., the items in Vp and scored higher) in attribute scoring. If there is no message propagated to an attribute, we define its entropy to be 0. Note that, in this implementation, we do not consider user u for calculating д for simplicity. It does not not mean we don’t value the importance of u in deciding attribute. We leave the exploration of incorporating u for future works。

weight entropy 會assign更高的權重到那些score 更高的item上面，這樣只要這個attribute的對應的item有很高的score，計算出來的attribute score就會比較偏向該item

weight entropy 會assign更高的權重到那些score 更高的item上面，如果一个attribute出现在许多得分高的item中，那么这个attribute被认为是有价值的

w3

The policy network takes the state vector s as input and outputs the values Q(s, a) for the two actions,

the system will always choose the action with higher estimated reward of taking a\_rec or a\_ask

The state vector s is a concatenation of two vectors:

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述 where s\_his encodes the conversation history, which is expected to guide the system to act smarter, e.g., if the asked attributes are accepted for multiple turns, it might be a suitable timing to recommend. The s\_len encodes the size of candidate item set V\_cand. As discussed by [13], it is easier to make successful recommendations when there are fewer candidate items

The reward containing five kinds of rewards, namely, (1) r\_rec\_suc , a strongly positive reward when the recommendation succeeds, (2) r\_rec\_f ail , a strongly negative reward when the recommendation fails, (3) r\_ask\_suc , a slightly positive reward when the user accepts an asked attribute, (4) r\_ask\_fail , a negative reward when the user rejects an asked attribute, and (5) r\_quit , a strongly negative reward if the session reaches the maximum number of turns. The accumulated reward is the weighted sum of these five. The detailed value for each reward can be found in Sec 5.2.

two significant differences between EAR and SCPR. First, SCPR leverages on the adjacent attribute constraint on the graph, largely reducing the search space of attributes. Second, SCPR scores attributes through message propagation on the graph, instead of by the policy network as what has been done in EAR. This enables our policy network to have a much smaller decision space — only two actions, alleviating the pressure for policy making.

5. EXPERIMENTS setting

An offline training for scoring function of item in reasoning step. We use the historical clicking record in the training set to optimize our factorization machine offline (Eq. (3)) by strictly follow [13]. The goal is to assign higher score to the clicked item for each users.

An online training for reinforcement learning used in consultation step. We use a user simulator (c.f. Sec 5.2.2) to interact with the user to train the policy network using the validation set

The detailed rewards to train the policy network are: rr ec\_suc=1, rr ec\_f ail =-0.1, rask\_suc=0.01, rask\_f ail =-0.1, rquit =- 0.3.

The parameters of the DQN are empirically set as following: the experience replay memory size is 50,000, the sample batch size is 128, discount factor γ is set to be 0.999. We optimize policy network with RMSprop optimizer and update the target network every 20 epsiodes.

a user simulator simulates a conversation session for a user-item interaction record found in the validation or test dataset.

****Limited Flexibility****: The simulator operates under the assumption that the user's preference is fixed to a specific item. It only accepts a recommendation list if it contains this predetermined item and only confirms attribute preferences if they match this item. This might not reflect real user behavior, where preferences can be more fluid and change over the course of a conversation.

****No Discovery Factor****: The simulator doesn't account for the possibility that a user might enjoy discovering new items they didn't initially prefer. In a real-world scenario, users might not know what they want until they see it, which the simulator does not capture.

****False Negatives****: The simulation might "falsely" reject an item that the user would actually like. This can happen if the user hasn't clicked on the item in the past (and thus the system doesn't recognize it as something the user likes), even though the user might enjoy the item if they discovered it.

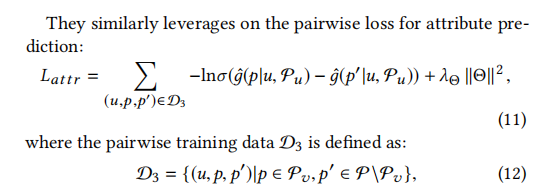
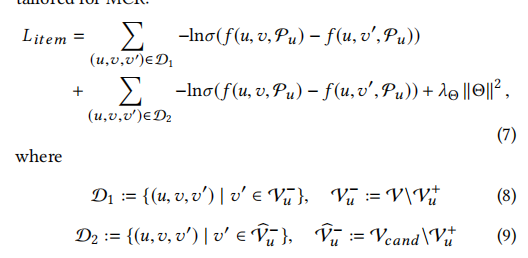
Why use negative sample:  
The ordinary negative samples (Vu-) help the model understand the user's general preference by contrasting interacted items with non-interacted ones.

The dynamic negative samples (Vbar u-) help the model capture the nuances of user preference as it evolves through an interaction sequence. These samples are crucial for understanding what the user chooses not to interact with from a set of recommended items, which can be different from simply not having interacted with an item.()

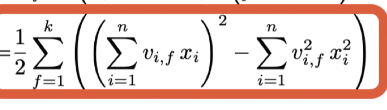
将 **D1** 和 **D2** 的负样本分开处理，是因为这两种负样本代表了用户的不同负偏好类型：**D1** 中的负样本代表用户对未互动过的物品的整体负偏好，而 **D2** 中的负样本代表用户在给定的推荐上下文中对未选择的候选物品的负偏好。

Loss function :  
在损失函数 Litem 中，两个主要的项（对应 D1 和 D2 的负样本）都试图最大化正样本评分与负样本评分之间的差异。具体来说，-lnσ(f(u,v, Pu) - f(u,v', Pu)) 这部分的目标是让用户u对于物品v的评分（对应正样本）要比对于物品v'的评分（对应负样本）高。这里的 σ 是sigmoid函数，它将输入值映射到0和1之间，所以这部分损失的范围在0（表示模型预测正确，正样本评分比负样本评分高）到正无穷（表示模型预测错误，负样本评分比正样本评分高）。

只要差距非常巨大，logsig就会接近0，差距很小，logsig就会接近1



FM：



****First-order term:****

In the provided code, the first order term is not explicitly calculated. However, you can note that in the embedding layers, weights are learned for each feature. This corresponds to the linear effect (**wi \* xi**) of the features.