# Social Attentive Deep Q-networks for Recommender Systems

社交关注型的DQN推荐系统

Abstract—Recommender systems aim to accurately and actively provide users with potentially interesting items (products, information or services). Deep reinforcement learning has been successfully applied to recommender systems, but still heavily suffer from data sparsity and cold-start in real-world tasks. In this work, we propose an effective way to address such issues by leveraging the pervasive social networks among users in the estimation of action-values (Q). Specifically, we develop a Social Attentive Deep Q-network (SADQN) to approximate the optimal action-value function based on the preferences of both individual users and social neighbors, by successfully utilizing a social attention layer to model the influence between them. Further, we propose an enhanced variant of SADQN, termed SADQN++, to model the complicated and diverse trade-offs between personal preferences and social influence for all involved users, making the agent more powerful and flexible in learning the optimal policies. The experimental results on real-world datasets demonstrate that the proposed SADQNs remarkably outperform the state-of-the-art deep reinforcement learning agents, with reasonable computation cost.

摘要—推荐系统旨在准确，主动地为用户提供潜在有趣的项目（产品，信息或服务）。深度强化学习已成功地应用于推荐系统，但在现实世界中的任务中仍然遭受数据稀疏和冷启动的困扰。在这项工作中，我们提出了一种有效的方法来解决此类问题，方法是利用用户之间普遍存在的社交网络来估算行为价值（Q）。具体来说，我们成功的利用社交注意层对二者之间的影响进行建模，从而开发了Social Attentive Deep Q-network (SADQN)，以根据个人用户和社交邻居的偏好来近似最佳行为价值函数。此外，我们提出了SADQN的增强型变体，称为SADQN ++，为所有相关用户建立复杂多样的个人偏好和社会影响权衡模型，使agent在学习最优策略时更加强大和灵活。在真实数据集上的实验结果表明，所提出的SADQN明显优于最新的深度强化学习代理，并且具有合理的计算成本。

**Index Terms**—DQN, reinforcement learning, recommender systems, social networks.

1 INTRODUCTION

RECOMMENDER systems aim to accurately and actively provide users with potentially interesting items (products, information or services), which have become a powerful tool to improving user experience and business profit in many online applications such as E-commerce, social networking and video sites. Online recommender systems are intrinsically interactive, in the sense that the recommender and target user usually interacts in a multi-step recommendation process. At each step, the recommender makes a decision to recommend some items to the user, and receives feedback from him/her. Normally, the decision made at current step affects not only the immediate reward, but also the future rewards at later steps. For example, an item recommended currently may not be liked by the user, but the received feedback provides valuable information about his/her interests, which can help the recommender make better recommendations in future .

推荐系统的目标是准确和积极地向用户提供潜在的有趣的项目(产品、信息或服务)，这已经成为许多在线应用(如电子商务、社交网络和视频网站)改善用户体验和商业利润的强大工具。在线推荐系统具有内在的交互性，在这个意义上，推荐者和目标用户通常在一个多时间步的推荐过程中进行交互。在每个时间步中，推荐者做出一个决定，向用户推荐一些项目，并从他/她那里得到反馈。通常情况下，当前时间步所做的决定不仅会影响当前的奖励，还会影响后续时间步中未来的奖励。例如，当前被推荐的一个项目可能没有被用户喜欢，但是收到的反馈提供了关于用户兴趣的有价值的信息，这可以帮助推荐者在未来做出更好的推荐。

Unfortunately, the majority of research on recommender systems is based on supervised learning, which focuses on learning accurate predictive models from historical feedback data for only single-step recommendations. Most of these recommendation approaches cannot provide a satisfactory solution to the multi-step interactive recommendation problem in real-world scenarios.

不幸的是，大多数关于推荐系统的研究都是基于监督学习的，它侧重于从历史反馈数据中学习准确的预测模型，仅用于单步推荐。这些推荐方法中的大多数都不能为现实场景中的多步交互式推荐问题提供令人满意的解决方案。

To address this issue, a potential way is to use reinforcement learning, which aims at learning an agent that can auto-control its behavior in an environment, in order to achieve a goal . By integrating both reinforcement learning and deep neural networks, deep reinforcement learning agents have shown human-level or even better performance in solving many complex decision making problems such as playing Atari and Go. Recently, a number of researchers incorporated the ideas and techniques of deep reinforcement learning into recommender systems, and proposed several novel recommendation algorithms which have shown great potential in a variety of recommendation domains. Compared to traditional recommenders, a notable merit of reinforcement learning agents is that they are able to actively discover users’ interests through user-agent interactions and recommend items that may bring maximal future rewards.

为了解决此问题，一种潜在的方法是使用强化学习，该方法可以在环境中自动控制其行为以实现目标的智能体。 通过整合强化学习和深度神经网络，深度强化学习智能体在解决许多复杂的决策问题（例如玩Atari和Go）方面表现出了人类水平甚至更高的性能。 最近，许多研究人员将深度强化学习的思想和技术纳入了推荐系统中，并提出了几种新颖的推荐算法，这些算法在各种推荐领域中都显示出巨大的潜力。 与传统推荐者相比，强化学习代理的一个显着优点是，他们能够通过用户-智能体交互来主动发现用户的兴趣，并推荐可能带来最大回报的项目。

Successful as they are, the existing reinforcement learning based approaches only exploit user-item feedback data, whose recommendation performance may be greatly reduced when data sparsity and cold-start1 occur. Fortunately, with the emergence of online social networks such as Twitter, additional social information of users is usually available to the recommender. According to the social influence theory, users are influenced by others in the social network, leading to the homophily effect that social neighbors may have similar preferences. Thus, it is a potential way to improve the quality of recommendations by leveraging available social networks, which has been widely studied and demonstrated in traditional social recommendation domains. However, most of the existing social recommendation models cannot be directly extended to reinforcement learning based systems, as they are based on the paradigm of supervised learning.

现有的基于强化学习的方法虽然成功，但仅利用用户-项的反馈数据，当出现数据稀疏和冷启动时，其推荐性能可能会大大降低。 幸运的是，随着诸如Twitter之类的在线社交网络的出现，推荐者通常可以使用用户的其他社交信息。 根据社会影响力理论，用户受到社交网络中其他人的影响，导致同质效应，即社交邻居可能具有相似的偏好。 因此，这是一种通过利用可用的社交网络来提高推荐质量的潜在方法，该方法已在传统的社交推荐领域中得到了广泛的研究和证明。 但是，大多数现有的社会推荐模型不能直接扩展到基于强化学习的系统，因为它们基于监督学习的范式。

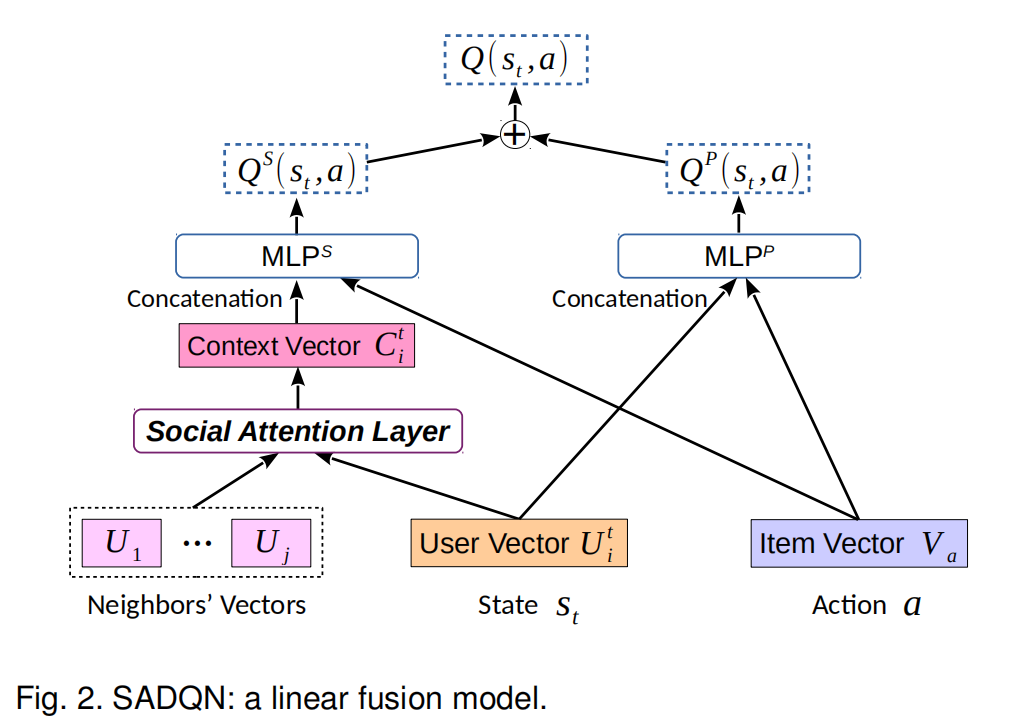
To the best of our knowledge, this paper makes the first attempt to improve the performance of deep reinforcement learning based recommenders, by effectively utilizing available social networks. In the context of deep reinforcement learning, the performance of many agents is determined by the estimation of the long-term rewards, e.g., action-values, which indicate, in the long run, how many benefits the agent will obtain if it recommends the items at current time . Similar to the prediction of short-term rewards (e.g., ratings) in traditional recommendation domains, it is biased and inefficient to estimate the action-values based on only users’ own preferences, due to the issues of cold-start and data sparsity. Thus, we propose to leverage social neighbors’ preferences to promote the estimation of action-values.

据我们所知，本文首次尝试通过有效利用现有的社交网络来提高基于深度强化学习的推荐器的性能。在深度强化学习的背景下，许多agent的表现是由对长期奖励的估计来决定的，例如action-value，它表示agent在多步的情况下，如果它在当前时间步推荐项目，它将获得多少利益。与传统推荐领域对短期奖励(如评分)的预测类似，由于冷启动和数据稀疏的问题，仅基于用户自己的偏好来估计行为值是有偏见和效率低下的。因此，我们建议利用社会邻居的偏好来促进对的估计。。

To implement this idea, we develop a novel deep reinforcement learning agent, termed Social Attentive Deep Q-network (SADQN). The key idea is that we estimate the action-values by a linear combination of two actionvalue functions, the personal action-value function QP and the social action-value function QS (see Figure 2). Intuitively, QP estimates action-values based on target user’s personal preferences, as most of the existing methods do. In contrast, QS is able to estimate action-values based on his/her social neighbors’ preferences, by utilizing an attention mechanism to model the influence from different social neighbors to target user. By integrating both functions, SADQN is able to learn recommendation policies that take advantage of both personal preferences and social influence.

为了实现这一思想，我们开发了一种新型的深度强化学习agent，称为社会关注深度Q-network (SADQN)。关键思想是，我们通过两个action-value函数的线性组合来估计行为价值，个人行为价值函数和社会行为价值函数(见图2)。就像大多数现有的方法一样，直观地根据目标用户的个人偏好来估计action-value。而能够根据其社会邻居的喜好来估算行为价值，它利用注意机制来模拟不同社会邻居对目标用户的影响。通过整合这两种功能，SADQN能够学习利用个人偏好和社会影响的推荐策略。

这里通常叫做Q-learning的目标，被称为学习率，并且表示智能体与环境交互所经历的转换。当迭代次数趋于无穷大时，估计的Q函数将收敛到，对采取greedy贪婪的策略将会成为最优策略



While SADQN is straightforward and easily understandable, the simple linear combination of function approximators QP and QS is not able to model the complicated and diverse trade-offs between personal interests and social in- fluence for all involved users, which limits the performance in approximating the optimal action-value function Q∗ . To handle this challenge, we propose an enhanced variant of SADQN, termed SADQN++ (see Figure 3), to fuse QP and QS more deeply by learning appropriate trade-offs from data autonomously. More specifically, we leverage QP and QS to learn some relevant hidden representations from personal and neighbors’ preferences. Then, we employ additional neural layers to summarize valuable features from these hidden representations, and predict the final actionvalue based on the summarized features. This way provides more flexibility to SADQN++ in modeling the trade-offs, leading to stronger capability in approximating the optimal action-value function. As a result, SADQN++ is able to learn the optimal policies more effectively

尽管SADQN简单明了且易于理解，但函数逼近器和的简单线性组合无法为所有相关用户模拟个人偏好和社会影响力之间的复杂多样的权衡，这限制了用户的性能逼近最优动作价值函数。为了应对这一挑战，我们提出了一种增强的SADQN变体，称为SADQN ++（请参见图3），可以通过自动从数据中学习适当的取舍来更深入地融合和。更具体地说，我们利用和从个人和邻居的偏好中学习一些相关的隐藏表示。然后，我们使用附加的神经层从这些隐藏的表示中总结出有价值的特征，并根据总结的特征预测最终的动作值。这种方式为SADQN ++在权衡建模中提供了更大的灵活性，从而在逼近最优动作价值函数方面具有更强的功能。结果表明，SADQN ++能够更有效地学习最佳策略

We empirically validate the performance of the proposed SADQNs by conducting solid experiments on three realworld datasets. The results show that they remarkably outperform four state-of-the-art deep reinforcement learning agents that fail to consider social influence, as well as several traditional recommendation methods. In particular, the relative improvements of SADQN++ against the best performing baseline are at least larger than 8.5% for coldstart recommendation, and 3.5% for warm-start recommendation. More importantly, the significant improvements of SADQNs over the state-of-the-art agents are accomplished with reasonable computation cost.

我们通过对三个真实世界的数据集进行可靠的实验，经验地验证了所提出的SADQN的性能。结果表明，它们明显优于四种不考虑社会影响力的最新深度强化学习代理以及几种传统的推荐方法。尤其是，相对于最佳性能的基线算法，SADQN ++的改进在冷启动情况下性能提高8.5％和在热启动情况下性能提高3.5％。 更重要的是，可以用合理的计算成本来实现SADQN与现有Agent的显着改进。

The remainder of this paper is organized as follows. In Section 2, we formalize the problem of T-step interactive recommendation, and introduce the background of reinforcement learning and Deep Q-network. Then, we present the major content of SADQNs in Section 3. We describe our experimental design, and show the results and analysis in Section 4. We review the related work in Section 5. Finally, we conclude this work in Section 6.

本文的其余部分安排如下。 在第二部分中，我们对T步互动推荐的问题进行了形式化，并介绍了强化学习和深度Q网络的背景。 然后，在第3节中介绍SADQN的主要内容。在第4节中描述我们的实验设计，并显示结果和分析。在第5节中回顾相关工作。最后，在第6节中结束这项工作。

3 SOCIAL ATTENTIVE DEEP Q-NETWORKS

As mentioned before, it is biased and inefficient to estimate the optimal Q∗ function (corresponding to an optimal recommendation policy) only based on individual users’ own feedbacks, due to the issues of cold-start and data sparsity. In this section, we propose a novel class of deep reinforcement learning (RL) agents, which are able to estimate Q∗ more effectively, by leveraging the available social network among users. Let S ∈ Rm×m be the adjacency matrix of the social network, where Sij = 1 if user i has a positive relation to user j (follows, trusts, etc.), and Sij = 0 otherwise. Let N (i) = {j : Sij = 1} denote the set of social neighbors whom user i trusts/follows.

如前所述，由于冷启动和数据稀疏性的问题，仅根据单个用户自己的反馈来最优动作价值函数（对应于最佳推荐策略）是有偏见的且效率低下的。 在本节中，我们提出了一种新型的深度强化学习（RL）代理，它可以通过利用用户之间的可用社交网络来更有效地估计。 令为社交网络的邻接矩阵，其中如果用户i与用户j具有正相关关系（跟随，信任等），则，否则，。 令表示用户*i*信任/关注的社交邻居的集合。

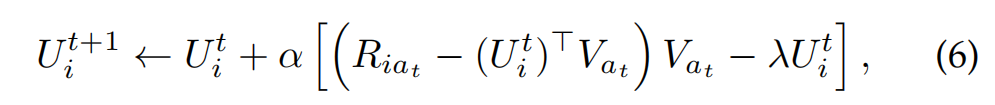
In what follows, we first describe a state/action representation method that utilizes matrix factorization (MF) to learn high-level vector representations of states and actions. Then, we present a basic Social Attentive Deep Q-network (SADQN), and an enhanced variant, named SADQN++. After that, we propose an enhanced representation method that utilizes a Social MF model, instead of the standard MF model, to learn the high-level state/action representations. Lastly, we describe the training algorithm of SADQNs.

接下来，我们首先描述一种状态/动作表示方法，该方法利用矩阵分解（MF）来学习状态和动作的高级矢量表示。 然后，我们介绍一个基本的社交注意力深层Q网络（SADQN），以及一个名为SADQN ++的增强型变体。 之后，我们提出了一种增强的表示方法，该方法利用社交MF模型而不是标准MF模型来学习高级状态/动作表示。 最后，我们描述了SADQN的训练算法。

**3.1 MF-based States and Actions**

We assume that the state st is a *f-*dimensional feature vector Uit ∈ Rf , denoting the real-time preferences of target user i at time step t. For each user j ∈ U, there is a f-dimensional feature vector Uj ∈ Rf , denoting the overall preferences of user j observed in advance. For each item (action) a ∈ I, there is also a f-dimensional feature vector Va ∈ Rf , denoting the overall features of item a. The feature matrices U ∈ Rf×m and V ∈ Rf×n are trained by standard matrix factorization (MF) [24] together with negative sampling [25] based on the historical feedback data R, which are held fixed during the user-agent interactive recommendation process. The target user’s vector Uit (i.e., state st) is initialized as the trained Ui at time step t = 0, and is updated by performing online MF on the real-time feedback data Riat for each time step t:

我们假设状态是*f-*dimensional特征向量，表示目标用户i在时间步t的实时偏好。 对于每个用户，都有一个f维特征向量，表示预先观察到的用户j的总体偏好。 对于每个项目（动作），还有一个*f-*dimensional特征向量，表示项目的整体特征。 特征矩阵和由standard matrix factorization（MF）[24]以及基于历史反馈数据的负采样[25]进行训练，这些历史反馈数据在用户代理期间保持固定互动推荐过程。 目标用户的向量(即状态）在时间步初始化为训练后的，并通过对每个时间步的实时反馈数据进行在线MF进行更新：



where α is the learning rate, and λ is the L2 regularization parameter.

其中α是学习率，而λ是L2正则化参数。

In other words, our proposed SADQNs are two-stage pipeline agents, which are different from the typical endto-end deep RL agents. A typical end-to-end agent directly estimate action-value based on the input of the raw state (the currently observed feedback information of user) and the raw action (the corresponding item id). However, in SADQNs, the action-value prediction is a two-stage procedure. At the first stage, we transform the raw state to a highlevel state representation, i.e., the user embedding, by using online MF. At the second stage, the Q-network predicts the action-value based on the MF-based state and the MF-based action (i.e., the item embedding that is trained by offline MF before the user-agent interactive process).

换句话说，我们提出的SADQN是两阶段流水线代理，与典型的端到端深层RL代理不同。 典型的端到端代理基于原始状态（用户当前观察到的反馈信息）和原始动作（对应项id）的输入直接估计动作值。 但是，在SADQN中，动作值预测是一个两阶段过程。 在第一阶段，我们使用在线MF将原始状态转换为高级状态表示，即用户嵌入。 在第二阶段，Q网络基于基于MF的状态和基于MF的动作（即，在用户代理交互过程之前由离线MF训练的项目嵌入）来预测动作值。

3.2 SADQN: A Linear Fusion Model

3.2 SADQN：线性融合模型

The basic idea behind SADQN is that the action-value is estimated by a linear combination of two actionvalue functions and ,which denote the personal action-value function and the social action-value function, respectively. Intuitively, QP estimates the action-values based on target user i’s personal real-time preferences, i.e., Uit . In contrast, QS estimates the action-values based on his/her social neighbors’ pre-observed preferences, i.e., Uj for j ∈ N (i), which are correlated with Uit according to the social influence theory.

SADQN的基本思想是，通过两个行为动作价值函数 and 的线性组合来估算动作价值函数，分别表示个人行为价值功能和社会行为价值功能。 直观地，根据目标用户的个人实时偏好（即）估算操作值。 相比之下，根据他/她的社交邻居的预先观察到的偏好（即的）来估计行动值，根据社会影响理论，与相关。

The architecture of SADQN is illustrated in Figure 2. The right part of SADQN is the personal action-value function approximator QP , which is a standard 4-layer MLP. It takes the concatenation of user vector Uit (i.e., the features of state st) and item vector Va (i.e., the features of action a) as input, followed by two fully connected (FC) layers, and outputs the personal action-value QP (st, a). In our experiments, each FC layer consists of 256 neurons with ReLU activation. Therefore, the specific architecture of MLPP is 2f → 256 → 256 → 1.

SADQN的体系结构如图2所示。SADQN的右边是个人行为价值函数近似器，它是一个标准的4层MLP(Multi-layerPerceptron)。用户向量 (也就是说状态)的特点和项目向量 (即动作的特征)进行连接作为输入,其次是两个全连接 (FC)层和输出的个人行为价值。在我们的实验中,每个FC层由256个神经元使用ReLU激活函数。因此，的具体架构为2f→256→256→1。

The left part of SADQN is the social action-value function approximator QS, in which the core is a social attention (SA) layer. The goal of the SA layer is to select influential social neighbors for target user i at time step t, and summarize the neighbors’ features to a context vector Cit . Then, the concatenation of context vector Cit and item vector Va is used to feed MLPS with the same architecture, which will output the social action-value QS(st, a).

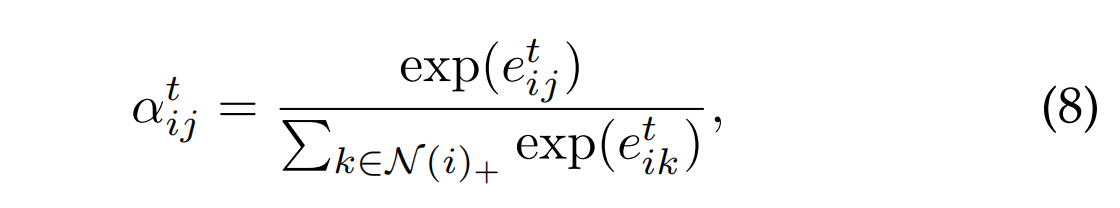
Specifically, we compute the context vector Cit by the following procedure. We employ the CONCAT attention mechanism to calculate the attention coefficient of target user i and his/her social neighbor j ∈ N (i):

具体来说，我们通过以下过程计算上下文向量。我们使用CONCAT注意力机制计算目标用户与其社会邻居的注意力系数:

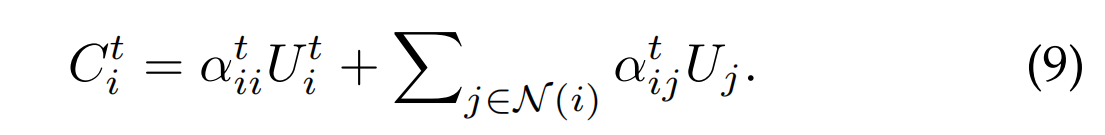


where w ∈ R2f is the weight vector of a single-layer feedforward network. The attention coefficient et ij indicates the social influence strength of user j to user i at time step t. Similar to [26], we also compute the attention coefficient of user i and himself/herself by: et ii = ReLU(wT·CONCAT(Uit , Uit )). Then, we compute the normalized attention coefficients αt ij by softmax function:

其中是单层前馈网络的权重向量。 注意力系数表示用户j在时间步t对用户i的社交影响力。我们还通过以下公式计算用户i和他/她自己的注意力系数。 然后，我们通过softmax函数计算归一化注意力系数：

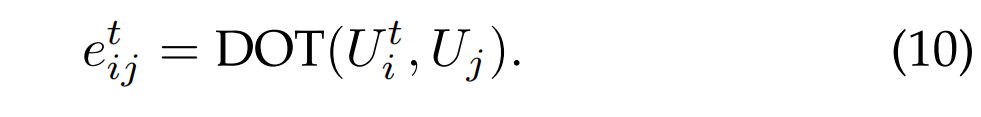


这里 . 最终, 我们获得了向量 by:



In our experiments, we also tried several different attention mechanisms to compute the context vector Cit .For example, the attention coefficient et ij can be computed by the simple DOT product:

在我们的实验中，我们还尝试了几种不同的注意力机制来计算上下文向量，例如，注意力系数可以通过简单的DOT乘积来计算：

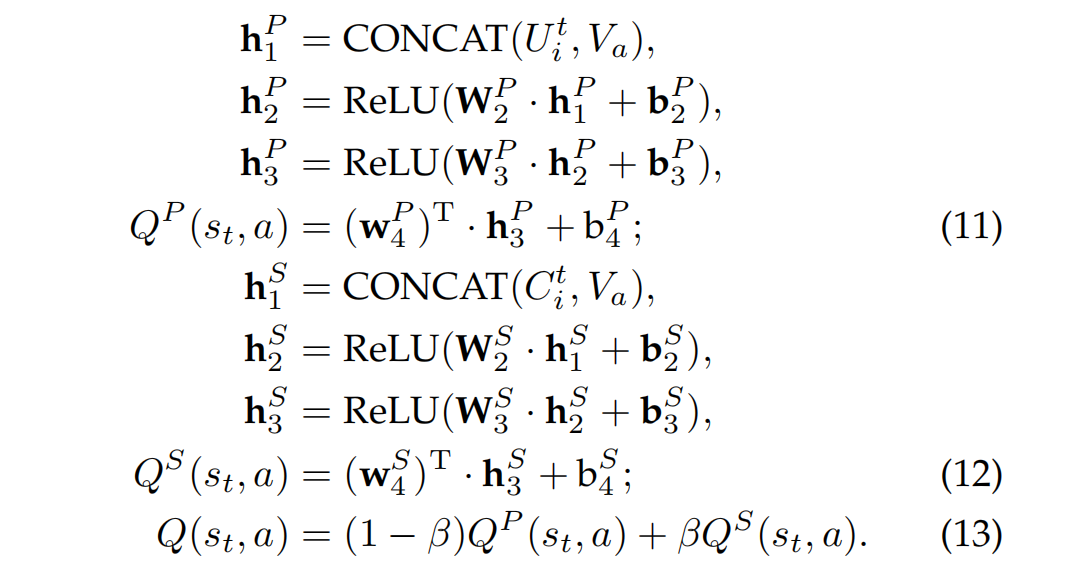


However, the performance of this approach showed no significant difference with the one in Equation 7. Moreover, we also implemented a single-layer graph attention network (GAT) [26] to compute Cit . Unfortunately, it did not show comparable performance against the above two approaches (see Table 4 for comparison).

但是，该方法的性能与公式7中的方法没有显着差异。此外，我们还实现了单层图注意力网络（GAT）[26]来计算。 不幸的是，它没有显示出与上述两种方法相当的性能（比较请参见表4）。

To summarize, the action-value Q(st, a) estimated by SADQN is formally defined as:

综上所述，由SADQN估算的动作值正式定义为：



Here, hPl , WPl and bPl denote the outputs, trainable weights and biases of l-th layer of MLPP , respectively. Similar notations are used for MLPS. β ∈ [0, 1] controls the trade-off between personal preferences and social influence

在此，hP1，WP1和bP1分别表示MLPP的第l层的输出，可训练权重和偏差。 MLPS也使用类似的符号。 β∈[0，1]控制着个人偏好和社会影响力之间的权衡

The network parameters of SADQN can be trained by performing the Q-learning updates in Equation 5. Note that if we only use QP or QS to estimate the action-values, SADQN will reduce to a pure personal model SADQNP (which is equivalent to the basic DQN model shown in Figure 1b) or a pure social model SADQNS.

可以通过在执行Q-learning更新来训练SADQN的网络参数。请注意，如果我们仅使用或来估算动作值，则SADQN将还原为纯个人模型（相当于 基本的DQN模型（如图1b所示）或纯社交模型。

3.3 SADQN++: A Deep Fusion Model

In the previous SADQN model, the personal approximator QP and the social approximator QS are simply fused by a linear combination at the output level. While this approach is straightforward and easily understandable, such a shallow fusion might limit the performance of Q-network in approximating the optimal action-value function Q∗. For example, the trade-offs between personal interests and social influence may vary considerably, for different users i at different time steps t, or even on different items a.

在以前的SADQN模型中，个人近似器和社会近似器只是在输出时通过线性组合进行融合。 尽管这种方法简单易懂，但这种浅层融合可能会限制Q网络在逼近最佳作用值函数时的性能。 例如，对于不同的用户i在不同的时间步长t或什至在不同的项目a上，个人利益和社会影响力之间的权衡可能会有很大的不同。

As such, using the same trade-off parameter β (in Equation 13) for all situations is obviously inappropriate. In fact, we tested different β ∈ {0.2, 0.5, 0.8} in our experiments. But the overall performance shows no significant difference, which implies that a good trade-off for one situation may be improper for another. On the other hand, it is infeasible to search or learn an optimal value of β for every situation, e.g., for each state-action pair (st, a).

因此，在所有情况下都使用相同的权衡参数β（在公式13中）显然是不合适的。 实际上，我们在实验中测试了不同的β∈{0.2，0.5，0.8}。 但是总体性能没有显着差异，这意味着对一种情况进行良好的权衡可能对另一种情况是不合适的。 另一方面，对于每种情况，例如对于每个状态-动作对，搜索或学习β的最佳值是不可行的。

To address this issue, we propose an enhanced variant of SADQN, termed SADQN++, to more deeply fuse the two approximators QP and QS by autonomously learning good trade-offs from data via additional neural layers.

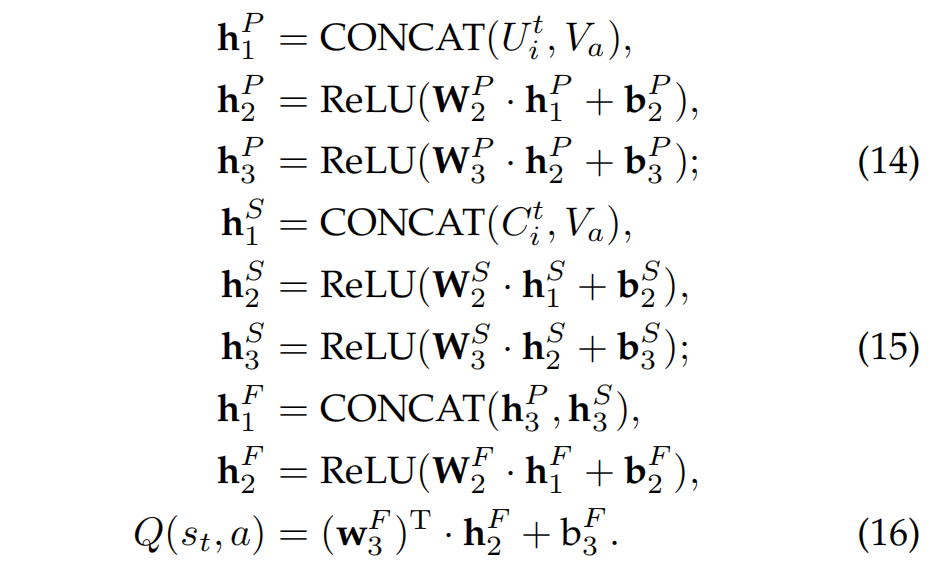
为了解决这个问题，我们提出了一种称为SADQN ++的增强型SADQN变体，它可以通过额外的神经层从数据中自主学习良好的取舍，从而更深入地融合两个近似器和。

The architecture of SADQN++ is illustrated in Figure 3, which is quite similar to SADQN, with only differences in the last few layers. Rather than using QP and QS to directly estimate action-values from personal and neighbors’ preferences, in SADQN++, we leverage them to learn some hidden vector representations that are relevant to the estimation of action-values. More specifically, MLPS (MLPP ) in SADQN++ is a 3-layer MLP with the architecture of 2f → 256 → 256, which will output the social hidden representation (personal hidden representation). Then, we employ MLPF with the architecture of 512 → 256 → 1, to autonomously summarize valuable features from both hidden representations, and to predict the final action-value Q(st, a) based on the summarized features.

SADQN ++的体系结构如图3所示，它与SADQN非常相似，仅在最后几层有所不同。 在SADQN ++中，我们没有使用和直接根据个人和邻居的偏好来估计行动值，而是利用它们来学习一些与行动 值的估计有关的隐藏矢量表示。 更具体地说，SADQN ++中的是具有2f→256→256体系结构的3层MLP，它将输出社交隐藏表示（个人隐藏表示）。 然后，我们采用512→256→1的架构，从两个隐藏的表示中自动总结出有价值的特征，并根据总结的特征预测最终的动作值。

This way of deep fusion provides more flexibility to the agent in approximating optimal action-values, making it possible to capture the complicated and diverse trade-offs between personal preferences and social influence for all involved users in real-world scenarios. Thus, SADQN++ is able to learn optimal policies in a more effective way. More formally, the action-value Q(st, a) estimated by SADQN++ is given by:

这种深度融合的方式为代理提供了更大的灵活性，使其可以逼近optimal action-values，从而可以捕获现实情况下所有相关用户的个人喜好和社会影响力之间的复杂多样的权衡。 因此，SADQN ++能够以更有效的方式学习最佳策略。 更正式地说，由SADQN ++估算的作用值由下式给出：



3.4 Enhancing SADQNs with Social MF-based States and Actions

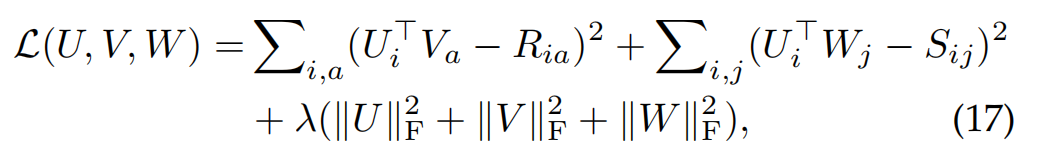
3.4通过基于社交MF的状态和动作来增强SADQN

To further model social information into the policy learning process, we propose an enhanced representation method that utilizes a Social MF model, instead of the standard MF model, to train the user/item embeddings (i.e., the high-level state/action representations), which can further improve the performance of SADQNs. We refer to the enhanced SADQNs as eSADQNs.

为了进一步将社会信息建模到策略学习过程中，我们提出了一种增强的表示方法，该方法利用社交MF模型而不是标准MF模型来训练用户/项目嵌入（即高级状态/动作表示） ，可以进一步提高SADQN的性能。 我们将增强的SADQN称为eSADQN。

More specifically, we employ the SoRec model [16] with uniform negative sampling [25] to train three feature matrices: the user feature matrix U ∈ Rf×m, the item feature matrix V ∈ Rf×n, and the social feature matrix W ∈ Rf×m, by jointly factorizing the rating matrix R and the social network S in a shared latent feature space. Formally, the joint loss function to be minimized is:

更具体地说，我们采用具有统一负采样[25]的SoRec模型[16]来训练三个特征矩阵：用户特征矩阵，项目特征矩阵和社交特征矩阵，通过在共享的潜在特征空间中共同分解评级矩阵R和社交网络S来实现。 正式地，要最小化的关节损失功能是：

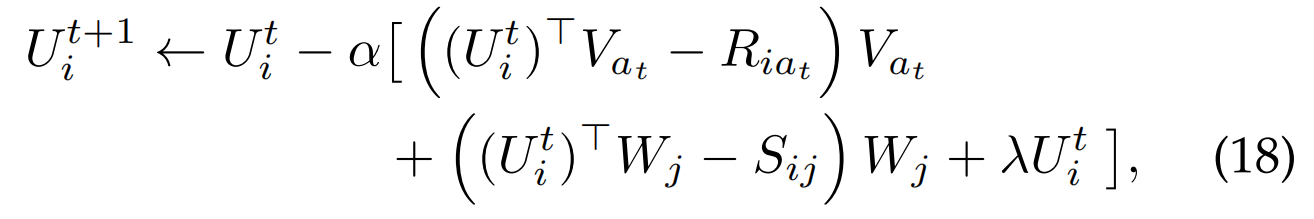


where Ria is either an observed positive feedback (Ria = 1) or a uniformly sampled negative one (Ria = 0), Sij is an observed social relation, k · kF denotes the Frobenius norm, and λ is the regularization parameter.

其中是观察到的正反馈或均匀采样的负反馈，是观察到的社会关系，而λ是正则化参数。

During the user-agent interactive recommendation process, the real-time user vector Uit is initialized as the trained Ui at time step t = 0, and is always updated by performing stochastic gradient descent (SGD) based on a uniformly sampled trust relation Sij and the real-time feedback Riat received after each time step t:

在用户代理交互推荐过程中，实时用户向量在时间步t = 0初始化为训练后的，并且始终通过基于均匀采样的信任关系执行随机梯度下降（SGD）来进行更新每个时间步t之后收到的实时反馈：



**3.5 Training Algorithm**

To train SADQNs (i.e., the proposed Q-networks), we employ the popular Q-leaning algorithm [23]. We do not adopt the training techniques of experience replay and target network used by the original DQN [7], as they are not able to improve the Q-learning performance for our task. To make the Qnetwork Q converge well, sufficient transitions (s, a, r, s0) of all possible states and actions are needed for Q-learning updates [6]. To this end, we propose a particular training scheme that enables the agent to collect transitions based on the feedback data of all training users.

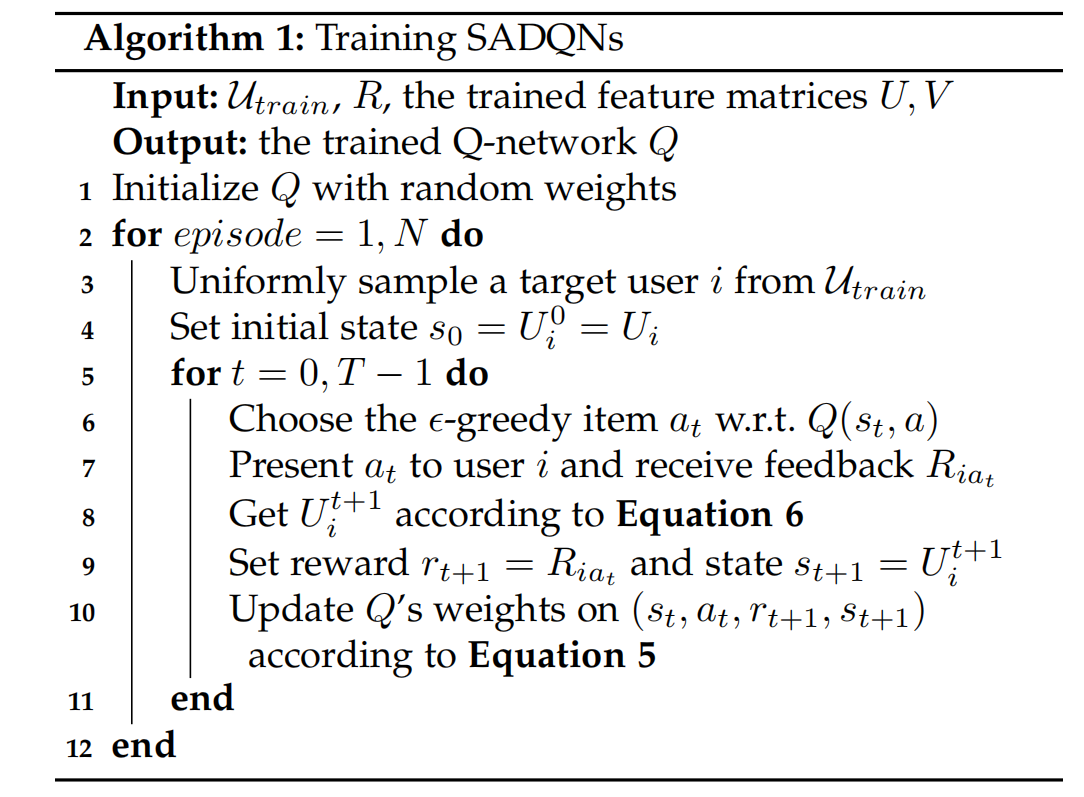
为了训练SADQN（即提出的Q网络），我们采用了流行的Q-leaning算法[23]。 我们不采用原始DQN [7]使用的经验重播和目标网络的训练技术，因为它们不能提高我们Q-learning性能。 为了使Qnetwork很好地收敛，需要对所有可能的状态和动作进行足够的转换以进行Q-learning的更新。 我们提出了一个特殊的训练方案，使代理能够根据所有训练用户的反馈数据收集转换。

Specifically, in each episode, we uniformly sample a user i from training set Utrain as the current target user, which will interact with the agent and generate corresponding states and rewards. To ensure exploration, in each state st, the agent uses a -greedy policy that selects a greedy action at = arg maxa Q(st, a) with probability 1 − and a random action with probability . The full algorithm for training SADQNs is presented in Algorithm 1. The training process could last for any number of episodes as long as the Q-network has not converged. At the testing stage, the trained agent can be used to make real-time interactive recommendations for any new user j. It only needs to interact with user j, observe state st (Ujt ), and recommend the greedy item at = arg maxa Q(st, a) at each time step t.

具体来说，在每一幕中，我们从训练集中均匀采样用户i作为当前目标用户，该用户将与代理进行交互并生成相应的状态和奖励。 为了确保探索，在每个状态中，代理程序均使用策略，该策略选择概率为-的处的贪婪动作和概率为的随机动作。 在算法1中介绍了用于训练SADQN的完整算法。只要Q网络尚未收敛，训练过程就可以持续任意数量的幕。 在测试阶段，可以使用受过训练的代理为任何新用户j进行实时交互式推荐。 它仅需要与用户j进行交互，观察状态，并在每个时间步长t 处推荐贪婪项。

3.5.1 Computational Complexity Analysis

3.5.1计算复杂度分析



In the inner for-loop in Algorithm 1, the computation time is mainly taken in computing Q-values for available items (line 6), updating user vector Uit+1 (line 8), and updating Q-network (line 10). The cost of computing Q-values is O(n|θ|), where |θ| is the number of Q-network weights and n is the number of all items. The cost of updating Q-network is O(|θ|). The cost of updating Uit+1 is O(f), where f is the dimensionality of latent feature space. Therefore, the time complexity of the training algorithm of SADQNs is O(NT(n|θ| + f)), where N is the number of episodes and T is the number of time steps. Similarly, we can derive that the cost of performing T-step interactive recommendations for a new user is O(T(n|θ| + f)).

在算法1的内部for循环中，计算时间主要用于计算可用项的Q值（第6行），更新用户向量（第8行）和更新Q网络（第10行）。 计算Q值的成本为，其中是Q网络权重的数量，n是所有项目的数量。 更新Q网络的成本为。 更新的成本为，其中f是潜在特征空间的维数。 因此，SADQNs训练算法的时间复杂度为，其中N为幕数，T为时间步数。 同样，我们可以得出结论，为新用户执行T步互动推荐的成本为。

**4 EXPERIMENTS**

To validate the performance of the proposed SADQNs, we conduct extensive experiments on real-world datasets. In this section, we first introduce our experimental setup, followed by presenting the experimental results and analysis.

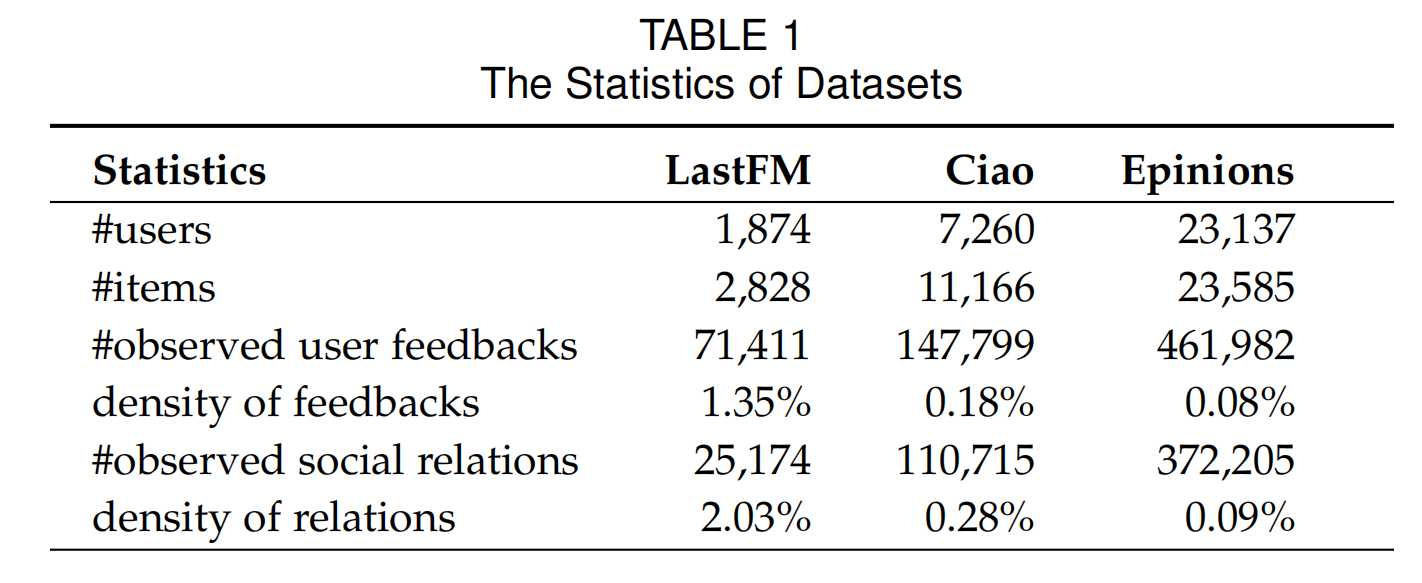
为了验证所提出的SADQN的性能，我们在现实世界的数据集上进行了广泛的实验。 在本节中，我们首先介绍我们的实验设置，然后介绍实验结果和分析。

**4.1 Experimental Setup**

*4.1.1 Datasets*

We employ three publicly available datasets: LastFM2 [27], Ciao3 [28], and Epinions4 [29] for our experiments. All the datasets contain a user-item feedback matrix and a useruser social network. As we consider the recommendation problem with implicit feedback, we convert the values of all observed feedbacks to 1. Besides, we remove the users or items that have fewer than 5 feedbacks, so as to ensure that there is enough data for training and testing. The basic statistics of the obtained datasets are shown in Table 1.

我们使用三个公开可用的数据集：LastFM2 ，Ciao3 和Epinions4 用于我们的实验。 所有数据集都包含一个用户项目反馈矩阵和一个user-user社交网络。 当我们考虑带有隐式反馈的推荐问题时，会将所有观察到的反馈的值转换为1。此外，我们删除反馈少于5个的用户或项目，以确保有足够的数据用于培训和测试。 表1显示了获得的数据集的基本统计信息。

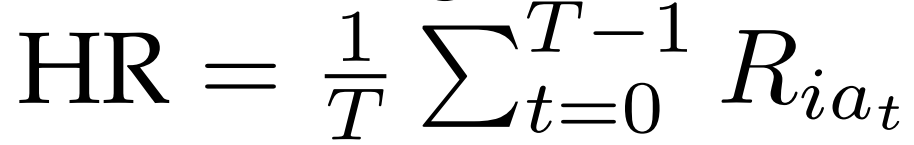


*4.1.2 Evaluation Methodology*

To conduct experiments on interactive recommendations, we assume that the observed feedbacks in the datasets are unbiased, as proposed in [30], [31]. Similar to [32], we randomly choose 1000 unobserved (i, a) pairs of user i as the negative feedbacks. During the T-step interactive recommendation process, the agent is forced to pick items from the available set that consists of the 1000 negative items and the observed positive items.

为了进行交互式建议的实验，我们假设数据集中观察到的反馈是无偏见的。 类似于[32]，我们随机选择1000对用户i的未观察对作为否定反馈。 在T步交互式推荐过程中，代理被迫从可用集合中选择由1000个否定项目和观察到的肯定项目组成的项目。

We adopt two popular evaluation metrics Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). The HR metric indicates the ratio of positive items among the T recommended items, which is defined as: HR = 1T PT −1 t=0 Riat , where at is the item recommended at time step t, and Riat = 1 (Riat = 0) if at is a positive (negative) item w.r.t. user i. The NDCG metric is computed by following procedure. At each time step t, a ranking list of the available items is produced according to the agent’s predictions (e.g., Q-values). The DCG(t) value is calculated by: DCG(t) = Pkj=1 2R(j)−1 log2 (1+j) , where k is the length of the ranking list, j denotes the rank position in the ranking list, and R(j) is the ground-truth feedback of the j-th item. The NDCG(t) is calculated by: NDCG(t) = DCG(t) Z , where Z is the DCG(t) value of the optimal ranking list sorted by ground-truth feedbacks. The final NDCG is obtained by averaging the NDCG(t) values for time steps t = 0, ..., T −1. We truncate the ranking list at 10 to compute the NDCG@10 values, and set T = 20 for evaluation.

我们采用两种流行的评估指标：命中率（HR）和归一化折合累积收益（NDCG）。 HR指标表示T个推荐项目中肯定项目的比率，定义为：， NDCG度量是通过以下过程计算的。在每个时间步骤t，根据代理商的预测（例如Q值）生成可用项目的排名列表。 DCG（t）值的计算公式为：

，其中k是排名列表的长度，j表示排名列表中的排名位置，而是第j个项目的真实反馈。 NDCG（t）的计算公式为：，其中Z是通过地面真实性反馈排序的最佳排名列表的DCG（t）值。通过将时间步t = 0，...，T -1的NDCG（t）值平均，可以得到最终的NDCG。我们将排名列表截断为10以计算NDCG @ 10值，并将T = 20进行评估。

We conduct experiments for two different recommendation scenarios: cold-start and warm-start. In the cold-start setting, we assume that the agent has no feedback data of target user at time step t = 0, i.e., at the beginning of the interactive recommendation process. We randomly choose 10% users who have at least 20 positive feedbacks as the testing set Utest, and others as the training set Utrain = U \Utest, which are used to test and train the agent, respectively. The data of Utrain is also used to train the feature matrices U and V . To fit the cold-start scenario, in each episode of training phase, the target user’s vector Ui0 (i.e., initial state s0) is set to a randomized vector rather than the trained Ui (see line 4 in Algorithm 1).

我们针对两种不同的推荐方案进行实验：冷启动和热启动。 在冷启动设置中，我们假设代理在时间步骤t = 0（即在交互式推荐过程开始时）没有目标用户的反馈数据。 我们随机选择至少有20个正面反馈的10％用户作为测试集，将其他用户作为训练集，分别用于测试和训练代理。 的数据还用于训练特征矩阵U和V。 为了适应冷启动情况，在训练阶段的每个阶段中，将目标用户的向量（即初始状态）设置为随机向量，而不是训练后的（请参见算法1中的第4行）。

In the warm-start setting, we assume that the agent already has 10 positive feedbacks of target user at time step t = 0. We select the users who have at least 30 positive feedbacks as the target set Utar, and others as the pretraining set Upre = U \ Utar. We randomly choose 10% users from Utar as the testing set Utest, and the remaining as the training set Utrain = Utar \ Utest. We use Rpre, Rtrain and Rtest to denote the data of Upre, Utrain and Utest, respectively. Then, for each user in Utrain (Utest), we randomly choose 10 positive feedbacks and move them from Rtrain (Rtest) to Rpre. The final data Rpre, Rtrain and Rtest is used to train the feature matrices, train and test the agent, respectively. In this warm-start scenario, the target user’s vector Ui0 (state s0) already captures some preference information of him/her.

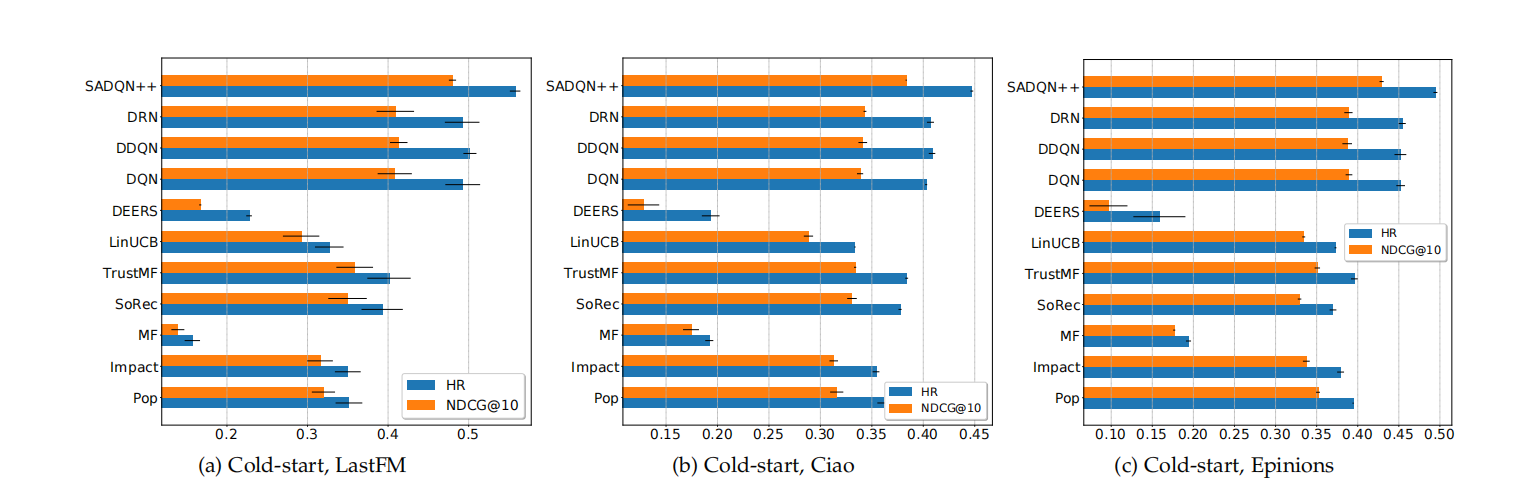
在热启动设置中，我们假设代理在时间步t = 0时已经有10个目标用户的正反馈。我们选择至少具有30个正反馈的用户作为目标集，将其他用户作为预训练集。 。 我们从中随机选择10％的用户作为测试集，其余的作为训练集。 我们使用，和分别表示，和的数据。 然后，对于（）中的每个用户，我们随机选择10个正反馈并将其从（）移至。 最终数据，和用于训练特征矩阵，分别训练和测试代理。 在这种热启动方案中，目标用户的向量（即初始状态）已经捕获了他/她的一些偏好信息。

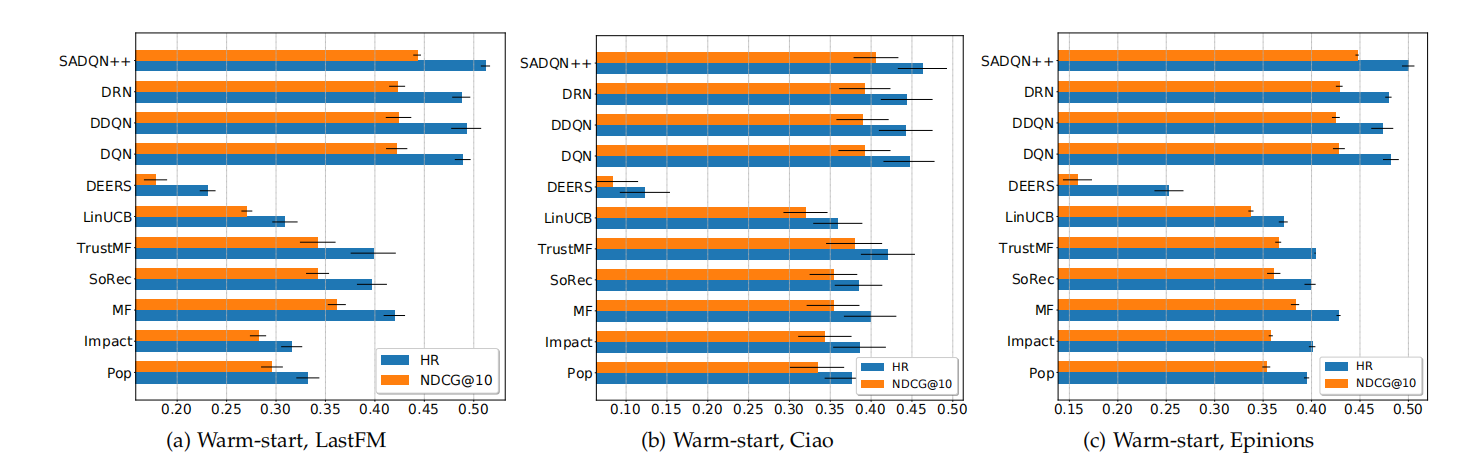
For both cold-start and warm-start settings, the testing HR or NDCG@10 value for an ideal agent will be 1. Besides, for each setting and each dataset, we conduct each experiment on 5 data splits obtained with different random seeds, and calculate the mean and standard deviation of the 5 groups of results for evaluation.

对于冷启动和热启动设置，理想代理的测试HR或NDCG@10值均为1。此外，对于每一组设置和每一个数据集，我们使用不同的随机种子进行5次数据分割进行每次实验，并计算5组结果的均值和标准差进行评价。

*4.1.3 Baselines*

We comparatively evaluate the proposed SADQNs against a variety of baselines, which are listed below.





1) DQN [7], a state-of-the-art deep reinforcement learning agent, which is originally designed for playing Atari. 2) DDQN [8], a state-of-the-art deep reinforcement learning agent, which extends DQN with double Q-learning [33]. 3) DRN [10], a state-of-the-art deep reinforcement learning agent for news recommendation, which is based on Dueling DQN [34] that estimates the action-values via both value function and advantage function. 4) DEERS [11], a state-of-the-art deep reinforcement learning agent for recommendation, which utilizes Gated Recurrent Units (GRU) to learn state features from both positive and negative item click sequences. 5) LinUCB [30], a representative contextual bandit algorithm for news recommendation. 6) SoRec [16], a representative social matrix factorization method, which factorizes both feedback matrix and social network simultaneously. 7) TrustMF [21], a representative social matrix factorization method, which models the mutual influence between trusters and trustees in the trust network. 8) MF [24], a conventional matrix factorization model, which only exploits the feedback matrix. 9) Impact [35], an active learning method, which picks the item that has highest impact on other items, where the impact is computed on the user-item bipartite graph. 10) Pop, a popularity-based method, which picks the item which has most positive feedbacks given by users.

1）DQN [7]，一种最先进的深度强化学习代理，最初是为玩Atari设计的。

2）DDQN [8]，一种最新的深度强化学习代理，它通过双重Q学习扩展了DQN [33]。

3）DRN [10]，一种用于新闻推荐的最新深度强化学习代理，它基于Dueling DQN [34]，它通过价值函数和优势函数两者来估算动作值。

4）DEERS [11]，一种最新的推荐深层强化学习代理，它利用门控循环单元（GRU）从正项和负项单击序列中学习状态特征。

5）LinUCB [30]，一种用于新闻推荐的代表性上下文赌博机算法。

6）SoRec [16]，一种代表性的社交矩阵分解方法，该方法同时对反馈矩阵和社交网络进行分解。

7）TrustMF [21]，一种代表性的社会矩阵分解方法，该模型模拟了信任网络中信任者和受托者之间的相互影响。

8）MF [24]，一种传统的矩阵分解模型，仅利用反馈矩阵。

9）**Impact** [35]，一种主动学习方法，它选择对其他项目影响最大的项目，其中影响是根据用户项目二部图计算出来的。

10）**Pop**，一种基于流行度的方法，它选择用户反馈最积极的项目。

To make the baselines applicable to our task, we adopt the same state/action features and training scheme of SADQNs for DQN, DDQN, DRN and LinUCB, and use a negative sampling technique of uniform distribution for MF, SoRec and TrustMF. We also adopt the same hidden layers of the personal action-value function of SADQN, i.e., two FC layers of 256 units with ReLU activation, for DQN, DDQN and both the value and advantage functions of DRN, which lead to better performance. For DEERS, we adopt the same architecture suggested in the original reference [11]. Moreover, to make a fair comparison, we set the feature dimensionality f = 64 for all methods (excluding Pop, Impact and DEERS). Other parameters are tuned based on cross-validation, which are set as follows: the regularization parameter λ = 0.01, the learning rate for updating feature vectors α = 0.01, the learning rate for updating Q-networks α = 0.0001, the discount factor γ = 0.5, and the -greedy parameter = 0.1.

为了使基线适用于我们的任务，我们对DQN，DDQN，DRN和LinUCB采用相同的状态/动作特征和SADQN训练方案，并对MF，SoRec和TrustMF使用均布的负采样技术。 对于DQN，DDQN以及DRN的价值和优势功能，我们还对SADQN的个人行动价值功能采用相同的隐藏层，即两个具有ReLU激活的256个FC层，以提高性能。 对于DEERS，我们采用原始参考文献[11]中建议的相同体系结构。 此外，为了进行公平的比较，我们为所有方法（不包括Pop，Impact和DEERS）设置了特征维数f = 64。 基于交叉验证来调整其他参数，这些参数的设置如下：正则化参数λ= 0.01，更新特征向量的学习率α= 0.01，更新Q网络的学习率α= 0.0001，折扣因子γ = 0.5，并且ε-greedy参数= 0.1。

It is important to notice that, our interactive recommendation task is distinctly different from session-based recommendation [36] or temporal social recommendation [37]. In their works, the recommender is developed to passively learn a predictive model from time series data such as a1, a2, ..., at−1, and to predict the next item at that may appear in the series. In contrast, our reinforcement learning agent is designed to actively learn a recommendation policy from user-agent interactions, and to provide items that may optimize the cumulative reward in a T-step recommendation process. As such, those Recurrent Neural Network based models [36], [37] are inapplicable to our task, and are not compared in our experiments.

重要的是要注意，我们的交互式推荐任务与基于会话的推荐[36]或时态社交推荐[37]明显不同。 在他们的工作中，推荐器的开发目的是从时间序列数据（例如a1，a2，...，at-1）中被动学习预测模型，并预测该序列中可能出现的下一项。 相反，我们的强化学习代理旨在从用户与代理的交互中主动学习推荐策略，并提供可以在T步推荐过程中优化累积奖励的项目。 因此，那些基于递归神经网络的模型[36]，[37]不适用于我们的任务，并且在我们的实验中未进行比较。

Moreover, although there are many social recommendation models proposed very recently, they cannot be applied to interactive recommendation unless making critical extensions to them. Unfortunately, most of them cannot be easily extended, such as the ones proposed in [38], [39], [40], [41], [42]. Thus, we only compare with two representative social recommendation models, SoRec and TrustMF, which are very flexible and can be extended to fit our task by online learning and negative sampling

而且，尽管最近提出了许多社会推荐模型，但是除非对其进行关键扩展，否则它们不能应用于交互式推荐。 不幸的是，它们中的大多数不能轻易扩展，例如[38]，[39]，[40]，[41]，[42]中提出的那些。 因此，我们仅与SoRec和TrustMF这两个具有代表性的社会推荐模型进行比较，它们非常灵活，可以通过在线学习和否定抽样进行扩展以适合我们的任务

**4.2 Performance Comparison against Baselines**

We now compare the performance of SADQN++ (the best performing variant of SADQN) against the baseline methods. Figure 4 and 5 show the comparison results in terms of the mean (bar) and standard deviation (line) of HR and NDCG@10 metrics for cold-start and warm-start recommendations, respectively. From these results, we have the following main findings. 现在，我们将SADQN ++（SADQN的最佳性能变体）与基准方法的性能进行比较。 图4和图5分别显示了冷启动和暖启动建议的HR和NDCG @ 10指标的均值（巴）和标准偏差（线）的比较结果。 从这些结果，我们有以下主要发现。

For cold-start recommendation (see Figure 4), the proposed SADQN++ model shows the best performance in terms of both metrics on all datasets. It remarkably outperforms the four deep reinforcement learning agents DQN, DDQN, DRN and DEERS that fail to consider social in- fluence, as well as other types of baselines. For example, its improvements in HR metric against the best performing baseline are 11.19%, 9.47% and 8.88% on LastFM, Ciao and Epinions datasets, respectively. These results not only verify the capability of SADQN++, but also demonstrate that social influence plays a fundamental role in improving the performance of deep reinforcement learning recommenders. 对于冷启动建议（请参见图4），建议的SADQN ++模型在所有数据集的两个指标上均显示出最佳性能。 它明显优于四个没有考虑社会影响力的深度强化学习代理DQN，DDQN，DRN和DEERS，以及其他类型的基准。 例如，在LastFM，Ciao和Epinions数据集上，相对于表现最佳的基线，其HR指标的改进分别为11.19％，9.47％和8.88％。 这些结果不仅验证了SADQN ++的功能，而且证明了社会影响力在改善深度强化学习推荐者的表现中起着根本性的作用。

The DQN, DDQN and DRN agents show the secondclass performance, while DEERS performs almost the worst which implies that it might be inappropriate to our task. The traditional matrix factorization model MF shows poor performance, as no feedback data is available at time step t = 0, while the social recommendation models SoRec and TrustMF demonstrate much better performance. Besides, the popularity method Pop and active learning method Impact are also competitive baselines in the cold-start setting.

DQN，DDQN和DRN代理表现出第二等的性能，而DEERS表现最差，这表明它可能不适合我们的任务。 传统的矩阵分解模型MF表现不佳，因为在时间步t = 0时没有反馈数据可用，而社交推荐模型SoRec和TrustMF则表现出更好的性能。 此外，流行方法Pop和主动学习方法Impact也是冷启动环境中的竞争基准。

For warm-start recommendation (see Figure 5), the proposed SADQN++ model also shows significantly better performance than the competitors. Specifically, its improvements in HR (NDCG@10) metric against the best performing baseline are 3.9%, 3.6% and 3.7% (4.4%, 3.5% and 4.3%) on LastFM, Ciao and Epinions datasets, respectively. The results of baselines show similar trends compared to coldstart setting, with one exception of the MF model, which shows competitive performance in the warm-start setting.

对于热启动建议（请参见图5），建议的SADQN ++模型还显示出比竞争对手更好的性能。 具体而言，在LastFM，Ciao和Epinions数据集上，相对于表现最佳的基准，其HR（NDCG @ 10）指标的改进分别为3.9％，3.6％和3.7％（4.4％，3.5％和4.3％）。 基线结果与冷启动设置相比显示出相似的趋势，但MF模型除外，它显示了在热启动设置中的竞争性能。

4.3 The Impact of Social Influence

To quantitatively analyze the impact of social influence, here, we make a more detailed comparison among the four variants of SADQN. They are the pure personal model SADQNP (which is equivalent to the DQN baseline), the pure social model SADQNS, the linear fusion model SADQN and the deep fusion model SADQN++, respectively (see Section 3 for more details). The comparison results of the mean and standard deviation of HR and NDCG@10 metrics are reported in Table 2. The relative improvements of SADQNS, SADQN and SADQN++ against SADQNP are shown in the brackets, which tell us clearly how social influence increasingly improves the performance when we model it from shallowly to deeply. Also, the best result in each case is highlighted. From this part of results, we observe the following main points.

为了定量分析社会影响力的影响，在这里，我们对SADQN的四个变体进行了更详细的比较。 它们分别是纯个人模型SADQNP（相当于DQN基线），纯社会模型SADQNS，线性融合模型SADQN和深度融合模型SADQN ++（更多信息，请参见第3节）。 表2中报告了HR和NDCG @ 10指标的均值和标准差的比较结果。括号中显示了SADQNS，SADQN和SADQN ++对SADQNP的相对改进，这清楚地告诉了我们社会影响力如何逐渐提高绩效 当我们从浅到深建模时。 此外，突出显示了每种情况下的最佳结果。 从这部分结果中，我们观察到以下要点。bserve the following main points. The deep fusion model SADQN++ performs much better than others. In particular, its relative improvements against the personal model SADQNP are at least larger than 9.0% for cold-start recommendation, and 3.5% for warm-start recommendation. Also, the improvements in cold-start setting show an interesting trend that they are totally consistent with the densities of social relations in the datasets. More specifically, the order of the improvements in HR (NDCG@10) metric on LastFM, Ciao and Epinions datasets is 13.2% > 11.0% > 9.3% (17.5% > 13.2% > 10.2%), same as the order of relation densities 2.03% > 0.28% > 0.09% (see Table 1). This demonstrates that, more social relations SADQN++ exploits, more benefits it will obtain.

符合以下要点。 深度融合模型SADQN ++的性能要比其他模型好得多。 特别是，相对于个人模型SADQNP，相对于冷启动推荐，其相对改进至少大于9.0％；对于热启动推荐，其相对改进至少大于3.5％。 同样，冷启动设置的改进显示出一个有趣的趋势，即它们与数据集中社会关系的密度完全一致。 更具体地说，LastFM，Ciao和Epinions数据集上的HR（NDCG @ 10）度量标准改进的顺序为13.2％> 11.0％> 9.3％（17.5％> 13.2％> 10.2％），与关系密度的顺序相同 2.03％> 0.28％> 0.09％（请参阅表1）。 这表明，利用SADQN ++开发更多的社会关系将获得更多的好处。

The linear fusion model SADQN performs second-best in all cases, and shows similar trends with SADQN++. The pure social model SADQNS also performs better than SADQNP in cold-start setting (except for the case of HR metric on Epinions dataset), but shows worse performance in warm-start setting (in most cases). This implies that, when the social network data is extremely sparse, or when the user-item feedback data is sufficient, the SADQNS model purely using social influence cannot achieve desirable recommendation performance.

线性融合模型SADQN在所有情况下均表现第二好，并且与SADQN ++表现出相似的趋势。 在冷启动设置中，纯社交模型SADQNS的性能也比SADQNP更好（Epinions数据集上的HR指标除外），但在热启动设置中，性能表现较差（在大多数情况下）。 这意味着，当社交网络数据非常稀疏时，或者当用户项反馈数据足够时，仅使用社交影响力的SADQNS模型就无法实现理想的推荐性能。

4.4 The Impact of Social MF-based States/Actions

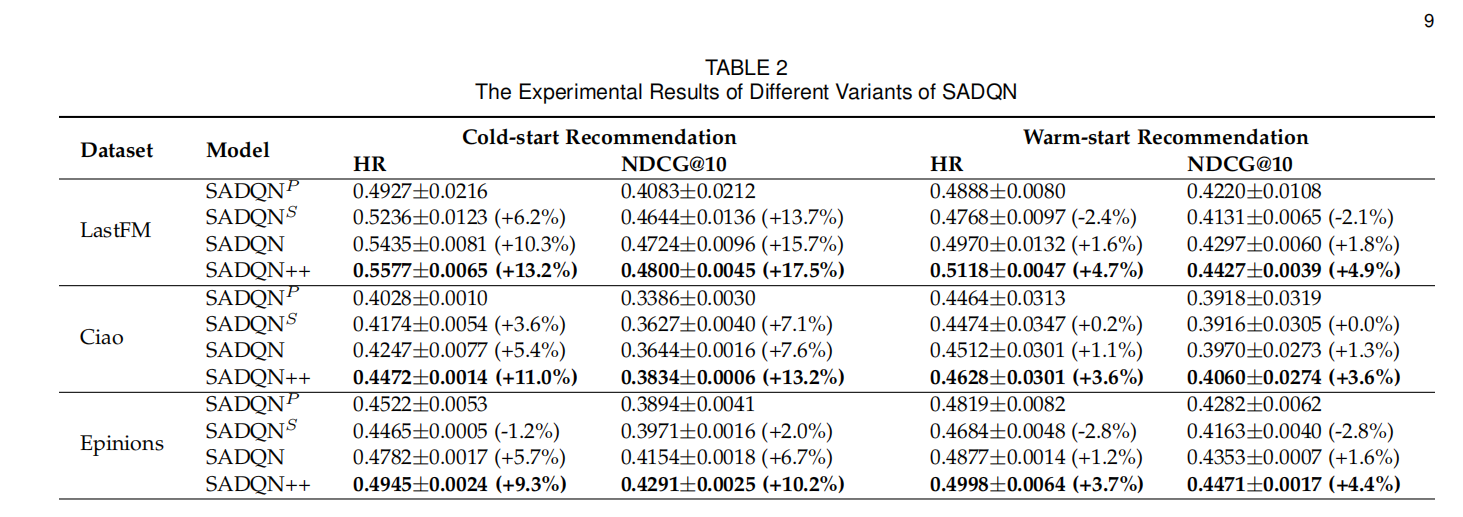
To investigate the impact of the proposed social MF-based states/actions, we compare the performance of SADQN++ with its enhanced version, eSADQN++. This experiment is only conducted in cold-start recommendation setting. The comparison results in terms of both HR and NDCG@10 metrics on all three datasets are shown in Table 3, where the bold font indicates the best result in each case. As we can see, eSADQN++ consistently performs better than SADQN++ with significant margins. This demonstrate that by additionally modeling social information into the latent state/action representations, our proposed SADQNs can be further improved.

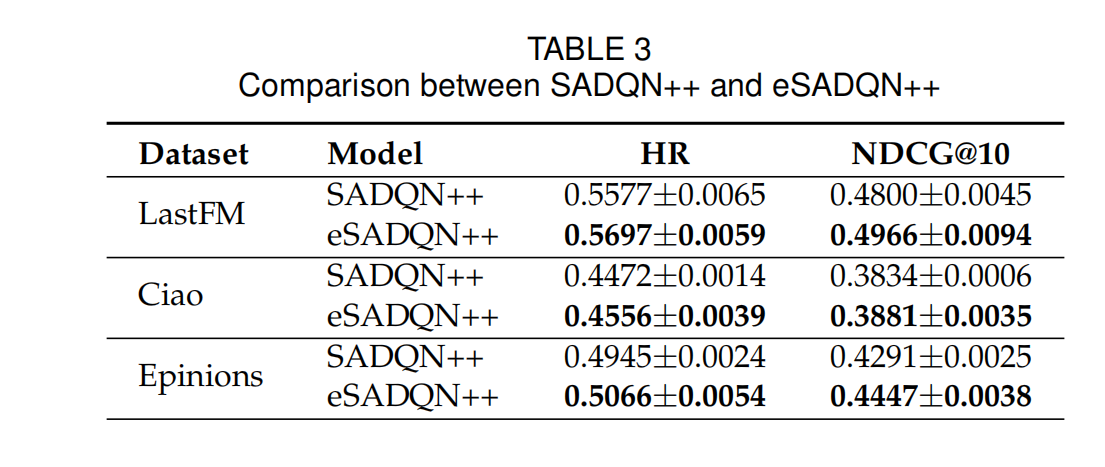
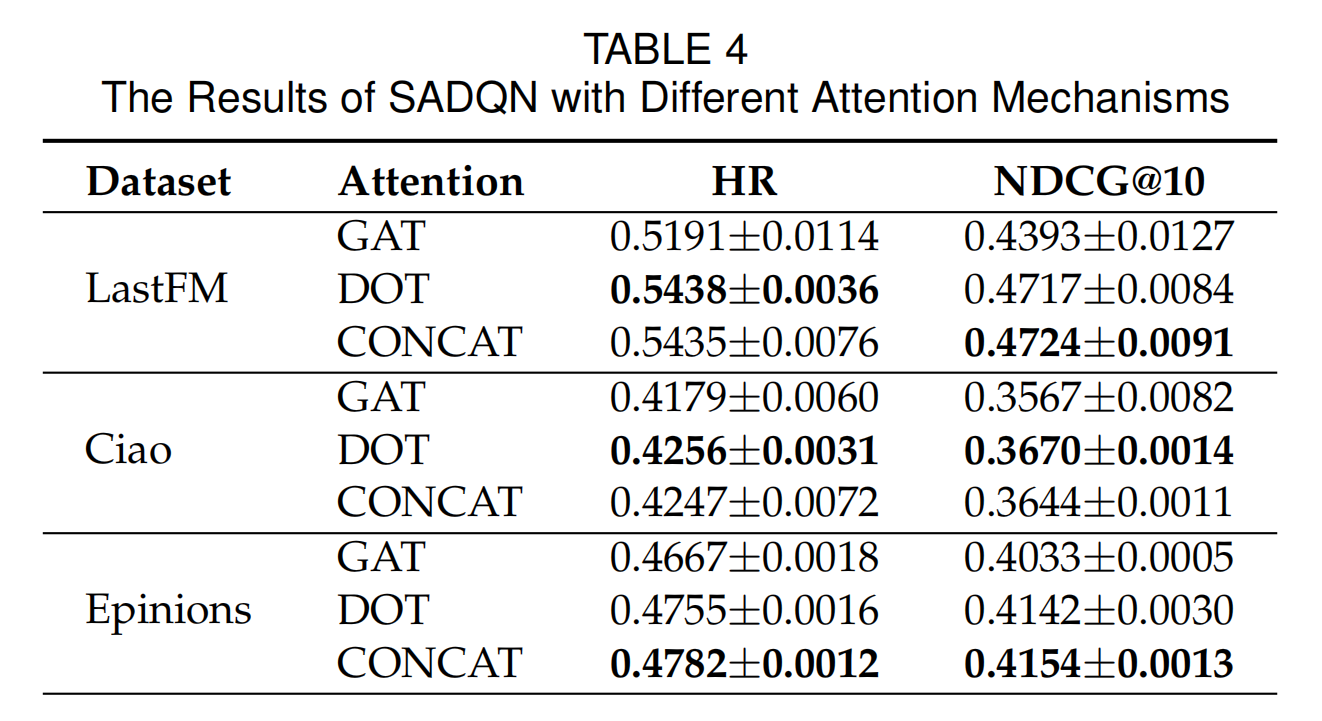
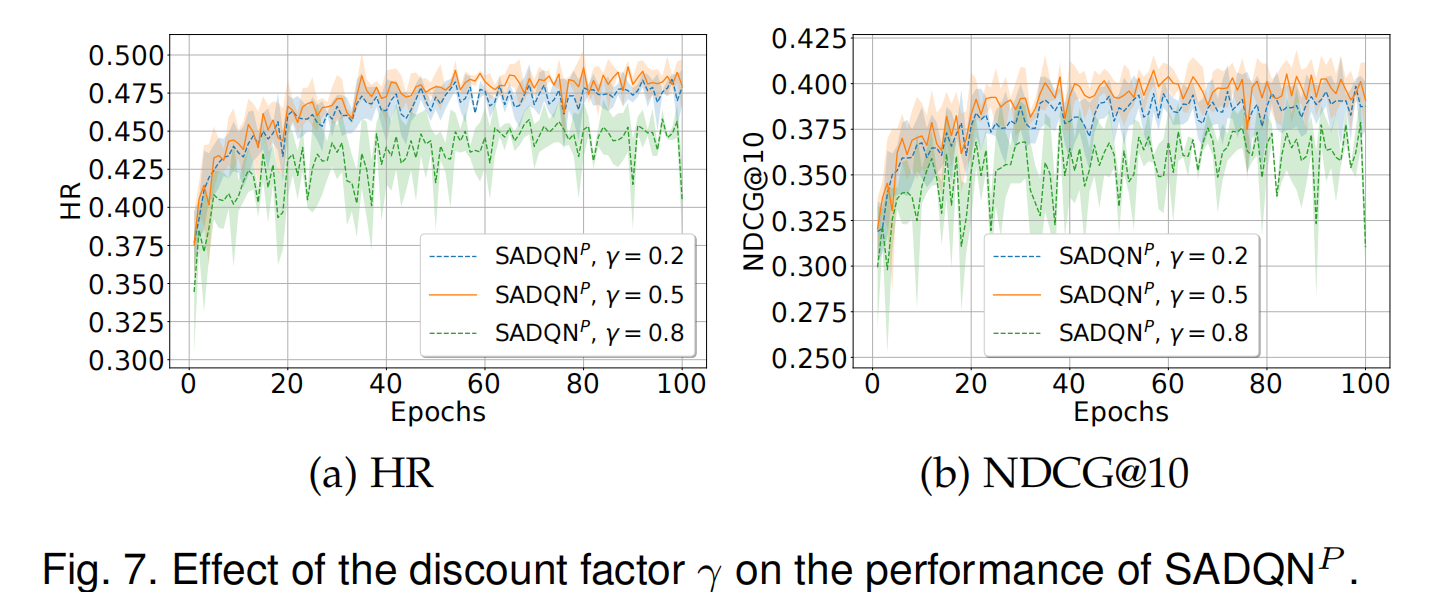
为了调查提议的基于社交MF的状态/动作的影响，我们比较了SADQN ++及其增强版本eSADQN ++的性能。 此实验仅在冷启动推荐设置中进行。 表3中显示了所有三个数据集在HR和NDCG @ 10指标方面的比较结果，其中粗体表示每种情况下的最佳结果。 如我们所见，eSADQN ++的性能始终优于SADQN ++，并具有可观的利润。 这表明，通过将社交信息附加建模为潜在状态/动作表示，可以进一步改进我们提出的SADQN。

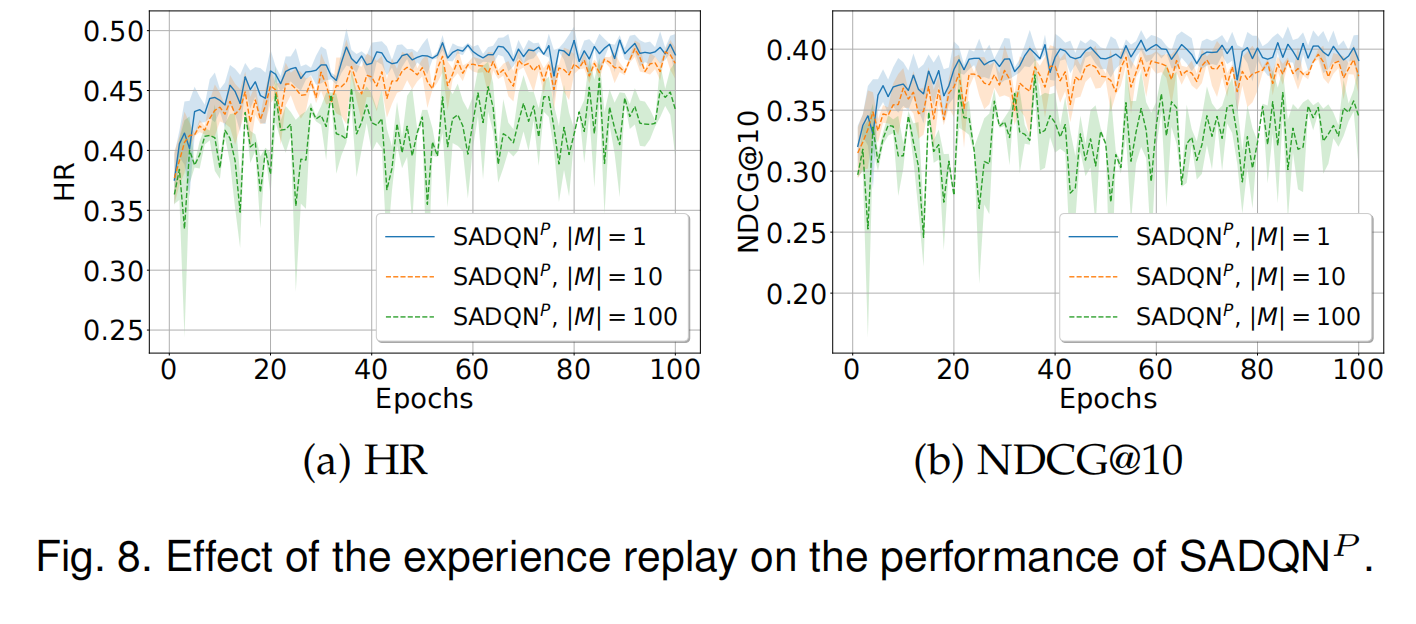
4.5 Comparison of Different Attention Mechanisms

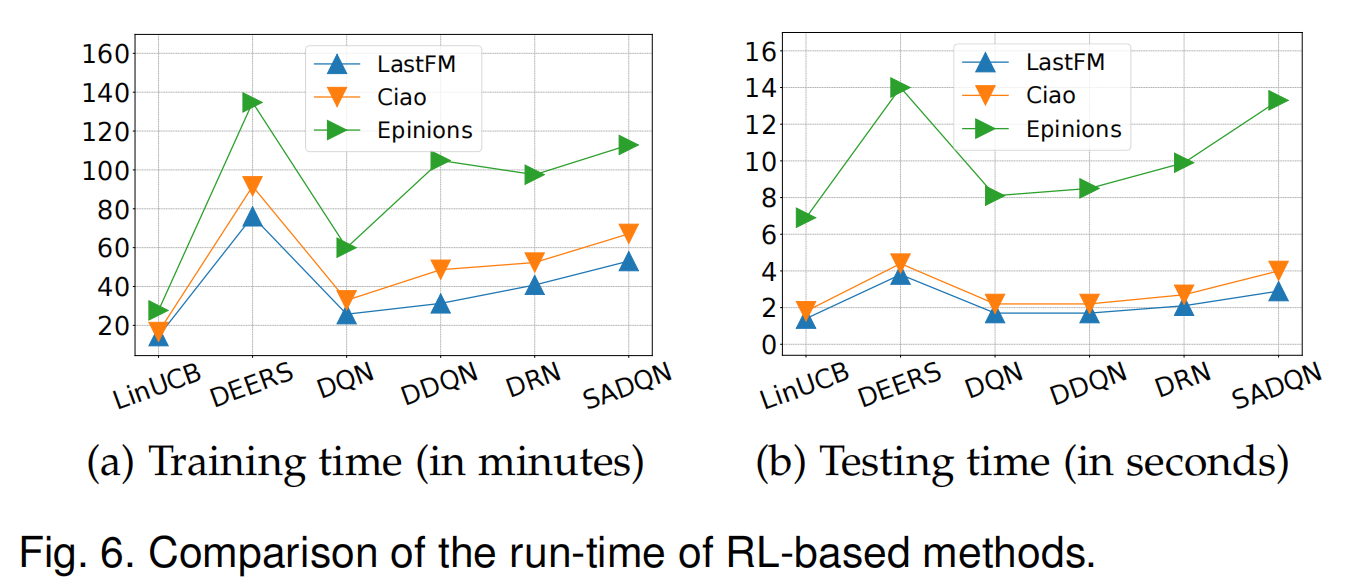
We now compare the performance of different attention mechanisms discussed in Section 3.2. We conduct an experiment to compare three SADQN agents which adopt the attention mechanisms GAT [26], DOT (Equation 10), and CONCAT (Equation 7, i.e., the default one used by SADQN), respectively. This experiment is only conducted in cold-start recommendation setting. The comparison results in terms of both HR and NDCG@10 metrics on all three datasets are shown in Table 4, where the bold font indicates the best result in each case. The two attention mechanisms CONCAT and DOT perform very closely, and both outperform GAT.

现在，我们比较第3.2节中讨论的不同注意力机制的性能。 我们进行了一项实验，比较了三种分别采用注意机制GAT [26]，DOT（等式10）和CONCAT（等式7，即SADQN使用的默认代理）的注意机制。 此实验仅在冷启动推荐设置中进行。 表4显示了所有三个数据集的HR和NDCG @ 10指标方面的比较结果，其中粗体表示每种情况下的最佳结果。 两种注意机制CONCAT和DOT的执行效果非常接近，并且都优于GAT。







5 RELATED WORK

In this section, we review the most relevant studies to our work, which can be categorized into two classes: (generalized) collaborative filtering and reinforcement learning.

在本节中，我们回顾与我们的工作最相关的研究，这些研究可以分为两类：（广义）协同过滤和强化学习。

5.1 Collaborative Filtering协同过滤

Collaborative filtering (CF) is probably the most prevalent and successful approach to building personalized recommender systems. The core idea behind CF is to infer users’ personal preferences from their nearest neighbors (i.e., collaborators), and thus make personalized and collaborative recommendations for all users.

协作过滤（CF）可能是构建个性化推荐系统的最流行和最成功的方法。 CF的核心思想是从最近的邻居（即合作者）中推断出用户的个人偏好，从而为所有用户提供个性化和协作性的推荐。

Early CF methods are essentially some memory-based (or heuristic-based) algorithms, which aim to predict the rating of a given user-item pair based on the aggregation of the ratings of other similar users (user-based CF), similar items (item-based CF), or both of them (Hybrid CF) [2], [44], [45], [46], [47], [48]. These memory-based CF methods have to compute the similarities of users or items over the entire user-item matrix, which usually have limited prediction accuracy and cannot be applied to large datasets. To improve the prediction accuracy and scalability, researchers incorporate the techniques in machine learning such as matrix factorization and neural networks, and design a large number of model-based CF methods that aim to learn an accurate prediction model from data. On the other hand, to alleviate the issues of cold-start and data sparsity, many researchers exploit additional data sources of the available social networks to help infer users’ preferences and develop a number of social CF methods. We will discuss these methods in the following. 早期的CF方法本质上是一些基于内存（或基于启发式）的算法，旨在基于其他相似用户（基于用户的CF），相似项目的评分汇总来预测给定用户-项目对的评分。 （基于项目的CF），或两者（混合CF）[2]，[44]，[45]，[46]，[47]，[48]。这些基于内存的CF方法必须在整个用户项矩阵上计算用户或项的相似度，这些相似度通常具有有限的预测准确性，因此无法应用于大型数据集。为了提高预测的准确性和可扩展性，研究人员将这些技术结合到了机器学习中，例如矩阵分解和神经网络，并设计了许多基于模型的CF方法，旨在从数据中学习准确的预测模型。另一方面，为了缓解冷启动和数据稀疏性的问题，许多研究人员利用可用社交网络的其他数据源来帮助推断用户的偏好并开发了多种社交CF方法。我们将在下面讨论这些方法。

5.1.1 Matrix Factorization

Matrix factorization (MF) maps users and items into a latent feature space and makes predictions via the inner product between the feature vectors of users and items. A lot of MF models have been proposed for the rating prediction task in explicit feedback datasets [24], [49], [50], [51]. A representative MF model is the probabilistic matrix factorization (PMF) proposed by Salakhutdinov and Mnih [50], which alleviates the over-fitting problem by defining Gaussian priors on the latent feature vectors. Koren [51] propose another famous MF model, SVD++, which achieves dominant performance in the rating prediction task and win the 2008 Netflix Prize. The core prediction model of SVD++ consists of: the inner product of the user and item feature vectors, the average rating in the observed data, the user bias parameter, the item bias parameter, and the implicit feedback term.

矩阵分解（MF）将用户和物品映射到潜在的特征空间，并通过用户和物品的特征向量之间的内积进行预测。 已经为显式反馈数据集中的评级预测任务提出了许多MF模型[24]，[49]，[50]，[51]。 代表性的MF模型是Salakhutdinov和Mnih [50]提出的概率矩阵分解（PMF），它通过在潜在特征向量上定义高斯先验来减轻过拟合问题。 Koren [51]提出了另一种著名的MF模型SVD ++，该模型在收视率预测任务中取得了显著成绩，并获得了2008年Netflix奖。 SVD ++的核心预测模型包括：用户和项目特征向量的内积，观测数据中的平均评分，用户偏差参数，项目偏差参数和隐式反馈项。

On the other hand, a number of MF models have been proposed for the top-k item recommendation task in implicit feedback datasets [52], [53], [54]. Hu et al. [52] investigate the special properties of implicit feedback datasets in a TV show recommender system, and propose a MF model by incorporating a confidence factor of users’ positive interests into the squared loss function. Pan et al. [53] improve the performance of MF for implicit feedback recommendation by proposing several negative example weighting/sampling strategies, including uniform weighting/sampling, user-oriented weighting/sampling, and item-oriented weighting/sampling.

另一方面，对于隐式反馈数据集[52]、[53]、[54]中的top-k条目推荐任务，已经提出了一些MF模型。Hu et al.[52]研究了电视节目推荐系统中隐含反馈数据集的特殊性质，将用户正兴趣的置信度引入平方损失函数，提出了MF模型。Pan等人[53]通过提出均匀加权/抽样、面向用户的加权/抽样、面向物品的加权/抽样等几种反例加权/抽样策略，提高了MF在隐式反馈推荐中的性能。

5.1.2 Neural Collaborative Filtering

Some researchers generalize the idea of CF into deep learning and design a number of neural CF methods. For instance, Wang et al. [55] propose a hierarchical Bayesian model, named collaborative deep learning (CDL), which jointly performs deep representation learning for the content information and CF for the feedback matrix. Wang et al. [56] develop a collaborative recurrent autoencoder (CRAE) which models the generation of content sequences in the CF setting. Zhang et al. [57] propose an integrated autoencoderbased framework, termed collaborative knowledge base embedding (CKE), which jointly learns the latent representations in CF and the items’ semantic representations from the knowledge base.

一些研究者将CF的概念推广到深度学习中，并设计了一些神经CF方法。例如，Wang等人[55]提出了一种名为协作深度学习(CDL)的分层贝叶斯模型，该模型对内容信息进行深度表示学习，对反馈矩阵进行CF学习。Wang等人[56]开发了一个协作循环自动编码器(CRAE)，它在CF设置中对内容序列的生成进行建模。Zhang等人[57]提出了一种基于自动编码器的集成框架，称为协作知识库嵌入(collaborative knowledge base embedding, CKE)，该框架联合学习CF中的潜在表示和知识库中项目的语义表示。

He et al. [58] propose a neural matrix factorization model, named NeuMF, which integrates multi-layer perceptron (MLP) with generalized matrix factorization to make predictions, and is trained by minimizing a binary crossentropy loss over both observed positive user-item pairs and sampled negative ones. Song et al. [59] propose a similar neural network architecture to make predictions, but the network is trained with a different pairwise ranking loss function. Ebesu et al. [60] propose a collaborative memory network (CMN) method, which integrates memory networks with MF models to perform top-k recommendations. Liang et al. [61] propose a neural CF method based on variational autoencoders for implicit feedback recommendations, which utilizes a generative model with multinomial likelihood and uses Bayesian inference for parameter estimation. Recently, a number of neural CF methods that leverage graph convolutional networks (GCNs) have been proposed for a variety of recommendation domains and have demonstrated appealing performance [62], [63], [64]. Besides, more works on neural CF methods can be found in a recent survey on deep learning based recommender systems [5].

他等人[58]提出了一个名为neuf的神经矩阵分解模型，该模型集成了多层感知器(MLP)和广义矩阵分解来进行预测，并通过在观察到的正用户-项目对和抽样的负用户-项目对上最小化二进制交叉熵损失来进行训练。Song等人[59]提出了一种类似的神经网络结构来进行预测，但该网络使用不同的两两排序损失函数进行训练。Ebesu等人[60]提出了一种协作记忆网络(collaborative memory network, CMN)方法，将记忆网络与MF模型相结合，进行top-k推荐。Liang等人[61]提出了一种基于变分自编码器的神经CF方法用于隐式反馈建议，该方法利用了具有多项似然性的生成模型，并使用贝叶斯推理进行参数估计。最近，一些利用图卷积网络(GCNs)的神经CF方法被提出用于各种推荐领域，并表现出了吸引人的性能[62]，[63]，[64]。此外，在最近关于基于深度学习的推荐系统[5]的调查中可以找到更多关于神经CF方法的工作。

5.1.3 Social Collaborative Filtering

A number of memory-based social CF methods [65], [66], [67] explore the trust propagation in social network, which generate rating predictions for a target user by directly aggregating the feedback data of his/her trusted friends. These memory-based social CF methods are able to improve the coverage of recommendations compared to traditional memory-based CF methods, but are not suitable to largescale datasets as they need to compute similarities over the entire rating matrix and the whole social network.

许多基于记忆的社交CF方法[65]、[66]、[67]探索了社交网络中的信任传播，这些方法通过直接聚合目标用户信任朋友的反馈数据来生成对目标用户的评级预测。与传统的基于内存的CF方法相比，这些基于内存的社会CF方法能够提高推荐的覆盖范围，但不适合大规模数据集，因为它们需要计算整个评级矩阵和整个社会网络的相似性。

Model-based social CF methods [16], [17], [18], [19], [21], [68], [69] employ the technique of MF to exploit the available social network data, which can be applied to large datasets. SoRec [16], TrustMF [21] and PSLF [19] factorize simultaneously the user-item rating matrix and the user-user social network simultaneously in a shared latent feature space. RSTE [17] fuses target user’s interests and his/her trusted friends’ tastes to model/predict a specific rating of a useritem pair. SocialMF [18] is similar to RSTE, but uses a different implementation to take into account the interests of trusted friends by incorporating a regularization term to control the distance between target user’s feature vector and the averaged feature vector of his/her trusted friends. In [68], the authors introduce several social regularization terms that are similar to the one in SocialMF. The main difference is that the trust value in social regularization term is replaced by the Pearson correlation coefficient (PCC) calculated on users’ rating data.

基于模型的社交CF方法[16]，[17]，[18]，[19]，[21]，[68]，[69]采用MF技术挖掘可用的社交网络数据，可应用于大数据集。SoRec[16]、TrustMF[21]和PSLF[19]同时对用户-物品评级矩阵和用户-用户社交网络进行因子分解，同时在一个共享的潜在特征空间中。RSTE[17]融合了目标用户的兴趣和他/她信任的朋友的兴趣，来建模/预测useritem对的特定评级。SocialMF[18]与RSTE类似，但使用了不同的实现方式，通过加入正则化术语来控制目标用户的特征向量与其可信好友的平均特征向量之间的距离，以考虑可信好友的利益。在[68]中，作者介绍了几个与SocialMF中相似的社会正规化术语。主要区别在于，将社会正则化项中的信任值替换为根据用户评分数据计算的皮尔逊相关系数(PCC)。

Recently, several complex social CF models beyond MF have been proposed for social recommender systems [38], [39], [40], [41], [42]. SREPS [38] simultaneously models the structural information in the social network, and the rating and consumption information in the user-item feedback data under an essential preference space, by using both network embedding and MF. GraphRec [41] utilizes graph neural networks to learn user and item latent feature vectors from both user-item feedback graph and user-user social graph. SamWalker [42] simultaneously learns personalized data confidence and draws informative training instances by leveraging the social network information. SAMN [39] utilizes an attention-based memory module to learn userfriend relation vectors, and builds a friend-level attention component to adaptively select informative friends for user preference modeling. DANSER [40] introduces dual graph attention networks to collaboratively learn representations for two-fold social effects, which are captured by a userspecific attention weight and a dynamic context-aware attention weight, respectively.

最近，针对社会推荐系统[38]，[39]，[40]，[41]，[42]，提出了几种MF以外的复杂社会CF模型。SREPS[38]同时利用网络嵌入和MF模型，对在必要偏好空间下的社交网络中的结构信息和用户-物品反馈数据中的评级和消费信息进行了建模。GraphRec[41]利用图神经网络从用户-项目反馈图和用户-用户社交图中学习用户和项目潜在特征向量。SamWalker[42]同时学习个性化数据信心，并利用社交网络信息绘制信息丰富的训练实例。SAMN[39]利用基于注意的记忆模块学习用户好友关系向量，并构建好友级注意组件自适应选择信息丰富的好友，用于用户偏好建模。DANSER[40]引入了双图注意网络来协作学习双重社会效应的表征，这些表征分别由用户特定的注意权重和动态上下文感知的注意权重捕获。

6 CONCLUSION

Deep reinforcement learning has been successfully applied to recommender systems, but still heavily suffer from data sparsity and cold-start in real-world tasks. In this work, we addressed these issues by strengthening the state-of-theart deep reinforcement learning recommenders with social influence among users. We developed a class of Social Attentive Deep Q-networks (SADQNs) to estimate actionvalues based on the preferences of both individual users and social neighbors, by successfully utilizing an attention mechanism to model the social influence between them. In particular, we proposed an enhanced variant of SADQN, termed SADQN++, which is able to model the complicated trade-offs between personal preferences and social influence for all users, making the agent more powerful and flexible in learning optimal policies.

深度强化学习已成功应用于推荐系统，但在现实任务中仍然严重受到数据稀疏和冷启动的影响。在这项工作中，我们通过加强最先进的深度强化学习推荐系统来解决这些问题。我们开发了一类社会关注深度q网络(SADQNs)，通过成功地利用注意机制来模拟个体用户和社会邻居之间的社会影响，来估计行为价值。特别地，我们提出了一种增强的SADQN变体，称为SADQN++，它能够对所有用户的个人偏好和社会影响之间的复杂权衡进行建模，使agent在学习最优策略方面更加强大和灵活。

We conducted extensive experiments on three real-world datasets to verify the effectiveness and efficiency of the proposed SADQNs. The results have demonstrated that social influence plays a fundamental role in improving the recommendation performance of deep reinforcement learning, especially in the cold-start recommendation scenarios. More importantly, the significant improvements of SADQNs over the state-of-the-art agents are accomplished with reasonable computation cost.

我们在三个真实世界的数据集上进行了大量的实验，以验证所提出的SADQNs的有效性和效率。研究结果表明，社会影响对提高深度强化学习的推荐性能具有重要作用，特别是在冷启动推荐场景下。更重要的是，与最先进的代理相比，SADQNs的显著改进是通过合理的计算成本实现的。

We notice that our SADQNs might have computation issue when deploying them in production systems that contain billions of items. This is because during prediction, all candidate tiems need to go through the neural network once, meaning that N feedforward passes are needed for N candidate items. This issue is also faced by some deep learning based methods such as NCF [58], but is not faced by other methods such as CDL [55] since it adopts the autoencoder architecture that will take the whole candidate items as input. To address the computation issue of our methods, a possible approach is to employ another type of DQN architecture that only takes state as input and outputs the action-values of all candidate items. This new architecture only needs one feedforward pass for N candidate items (additional computation cost only appears in the last layer of the Q-network). However, this new architecture might reduce the recommendation quality, as it ignores the information of item representations. We will explore more about the new architecture in the future work.

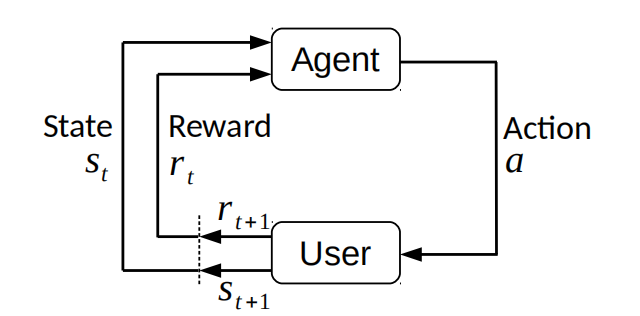
我们注意到，当我们的sadqn部署到包含数十亿项的生产系统中时，它们可能会出现计算问题。这是因为在预测过程中，所有候选时间都需要经过神经网络一次，这意味着N个候选项目需要N个前馈传递。这也是一些基于深度学习的方法如NCF[58]所面临的问题，但其他方法如CDL[55]所没有面临的问题，因为它采用了将整个候选项作为输入的自动编码器架构。为了解决我们方法的计算问题，一种可能的方法是采用另一种类型的DQN体系结构，它只接受状态作为输入，并输出所有候选项的动作值。这个新架构只需要N个候选项的一个前馈传递(额外的计算成本只出现在Q-network的最后一层)。然而，这种新体系结构可能会降低推荐质量，因为它忽略了项目表示的信息。我们将在未来的工作中探索更多关于新架构的内容。

We consider a recommender system with user set U = {1, ..., m} and item set I = {1, ..., n}. Let R ∈ Rm×n denote the user-item feedback matrix, where Ria = 1 if user i gives a positive feedback on item a (clicks, watches, etc.), and Ria = 0 otherwise.

我们考虑一个推荐系统，其用户集，项目集。 令表示用户-项目反馈矩阵，其中如果用户i对项目a（点击，观看等）给出正反馈，则，否则，0。

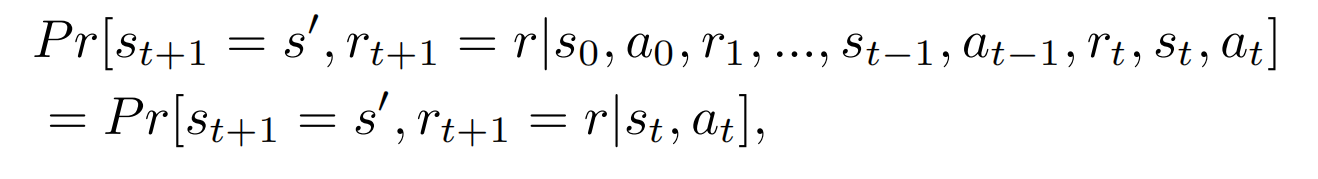
We focus on the task of recommending items in sequential user-agent interactions, which can be formulated as a standard reinforcement learning (RL) problem [6]. Specifi- cally, a recommendation agent and a user (an environment that describes the interactive recommendation process of the user) interact at discrete time steps (see Figure 1a). At each time step t, the agent observes the environment’s state st (representing the current preferences of user i), and accordingly takes an action (item) at based on its policy (probability distributions over actions given states). One time step later, as a consequence of its action, the agent receives a reward rt+1 (Riat ) and next state st+1 from the environment. The goal of the agent is to maximize the cumulative reward it receives in T interactions.

我们专注于在顺序的用户-智能体交互中推荐项目的任务，可以将其表述为标准强化学习（RL）问题。 具体来说，推荐智能体和用户（描述用户的交互式推荐过程的环境）在不连续的时间步长进行交互。 在每个时间步骤*t*，代理观察环境状态（代表用户*i*的当前偏好），并相应地根据其策略采取行动（项目）（给定状态下动作的概率分布）。 一步之后，作为其动作的结果，代理从环境接收到奖励和下一个状态。 智能体的目标是在T步互动中最大化将来的累积奖励。



More formally, the environment (i.e., the user’s interactive recommendation process) can be mathematically described by a Markov decision process (MDP), which satisfies the Markov property:

更正式地讲，环境（即用户的交互式推荐过程）可以通过马尔可夫决策过程（MDP）来数学描述，该过程满足Markov属性：



**2.2 Reinforcement Learning**