文章编号：1003-0077(2020)01-0097-09

模仿排序学习模型

曾 玮[1,2]，俞蔚捷[3]，徐 君[3]，兰艳艳[1]，程学旗[1]

(1. 中国科学院 计算技术研究所 网络数据科学与技术重点实验室，北京 100190；
2. 中国科学院大学，北京 100049；
3. 中国人民大学 高瓴人工智能学院 大数据管理与分析方法研究北京市重点实验室，北京 100872)

摘 要：文档排序一直是信息检索(IR)领域的关键任务之一。受益于马尔科夫决策过程强大的建模能力，以及强化学习方法强大的求解能力，近年来基于强化学习的排序模型被提出并取得了良好效果。然而，由于候选文档中会包含大量的不相关文档，导致基于“试错”的强化学习方法存在效率低下的问题。为解决上述问题，该文提出了一种基于模仿学习的排序学习算法 IR-DAGGER，其基于文档标注信息构建专家策略，在保证文档排序精度的同时提高了算法的学习效率。为了测试 IR-DAGGER 的性能，该文基于面向相关性排序任务的 OHSUMED 数据集和面向多样化排序的 TREC 数据集进行了实验，实验结果表明 IR-DAGGER 在上述两个数据集上均提升了文档排序的精度和效率。

关键词：排序；模仿学习；强化学习

中图分类号：TP391 **文献标识码**：A

Imitation Learning to Rank

ZENG Wei[1,2], YU Weijie[3], XU Jun[3], LAN Yanyan[1], CHENG Xueqi[1]

(1. CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;
2. University of Chinese Academy of Sciences, Beijing 100049, China;
3. Gaoling School of Artifical Intelligence, Beijing Key Laboratory of Big Data Management and Analysis Methods, Renmin University of China, Beijing 100872, China)

Abstract: Document ranking is one of the central tasks in a number of IR applications. In recent years, efforts have been made to apply reinforcement learning for learning document ranking models and a number of methods have been developed. Though preliminary success has been achieved, existing reinforcement methods still suffer from the sparseness of the relevant documents. In this paper, we propose to involve ground-truth ranking lists during the learning process, achieving a novel imitation learning-based learning to rank algorithm called IR-DAGGER. It utilizes the ranking lists sampled by the expert policy, which can enhance the learning efficiency while keeping the ranking accuracies. Experimental results based on OHSUMED and TREC showed that IR-DAGGER can outperform the state-of-the-art baselines for the tasks of relevant ranking and diverse ranking, indicating the effectiveness and efficiency of imitation learning in document ranking.

Keywords: learning to rank; imitation learning; reinforcement learning

收稿日期：2019-07-19 **定稿日期**：2019-10-11

基金项目：国家自然科学基金(61872338,61832017,61773362,61425016,61472401,61722211,61906180)；北京高校卓越青年科学家计划项目(BJJWZYJH012019100020098)；北京智源人工智能研究院(BAAI2019ZD0305)；中国人民大学科学研究基金(2018030246)；中国科学院青年创新促进会优秀会员项目(20144310,2016102)；国家重点研发项目(2016QY02D0405)

0 引言

作为文档检索、推荐、专家发现等众多检索任务的核心问题，排序模型的效率问题一直是研究的热点。排序学习模型[1]将排序问题转化成经典的分类、回归等机器学习问题，被广泛应用到各个领域。近年来，考虑到排序任务的序列性特点，研究者们开始尝试构建基于马尔科夫决策过程的排序模型，提出了一系列基于强化学习的排序学习算法，并成功应用于相关性排序、多样化排序等多种任务[2-3]，被称为强化排序学习。

强化排序学习将搜索引擎看作智能体，将用户看作环境，通过搜索引擎和用户间的多次交互实现学习排序模型的目的。具体到网页排序任务，通过将文档排序序列根据排序位置分解，将排序表达成序列决策问题，并通过马尔科夫决策过程[4]进行建模。受益于马尔科夫决策过程强大的建模能力，以及强化学习对于序列决策问题强大的学习能力，基于强化学习的排序模型在各个排序任务上崭露头角，取得了优异成绩，典型的方法包括相关性排序模型 MDPRank[5]、多样性排序模型 MDPDIV[6]、多页排序模型 MDPMPS[7]、以及会话排序模型 Win-Win Search[8]等。

强化学习基于试错[9]的机制进行学习，其依赖于智能体和环境不断进行交互，得到一系列交互轨迹数据。强化学习算法直接将这些交互轨迹数据作为训练数据，以奖励信息为导向进行学习。但是，具体到文档排序任务，一个查询所对应的候选文档中往往含有大量的不相关文档。因此，在学习初期，随机初始化的排序策略会选择不相关文档排在文档序列前端，这些不相关文档无法提供有效的奖励信号，无法构建高质量的文档序列，进而导致排序模型学习效率低下。

为了解决强化排序模型的学习效率问题，本文尝试通过文档的标注信息构建专家策略来辅助搜索引擎与用户进行更好的交互，从而生成高质量的文档排序，并提供有效的奖励信号。具体而言，本文提出了基于模仿学习的排序学习算法 IR-DAGGER，其结合专家策略和马尔科夫决策过程(Markov decision process，MDP)自身策略进行文档选择。其中，专家策略每次都会选择最佳文档以获得较高的奖励信息，保证数据的利用效率；MDP 自身策略使得搜索引擎根据当前排序模型的策略与用户进行交互，减少随机策略的累积误差，从而保证了排序学习的性能。

为了验证本文所提出模型的有效性，本文分别在面向相关性排序的标准数据集 OHSUMED 和面向多样化排序的数据集 TREC 上进行了实验。实验结果表明，IR-DAGGER 超越了所有的基准方法，取得了最好的排序精度和学习性能，实验结果表明模仿排序学习可以有效地提升传统强化排序学习算法的精度和效率。

1 研究背景：强化排序学习

1.1 任务描述

在排序任务中，对每一个查询 $q_k(1\leqslant k\leqslant n)$，有 m_k 个文档构成候选文档集合$\{d_1^k,d_2^k,\cdots,d_{mk}^k\}$，每一篇候选文档都标有其与该查询的相关性信息$\{y_1^k,y_2^k,\cdots,y_{mk}^k\}$。给定查询和候选文档后，可以通过查询—文档特征 $x=\Phi(q,d)$ 来表达查询和文档的关系，其中包含词频、逆向文档频率、BM25、LMIR 等信息。文档标注代表该文档与查询的相关程度，比如可以取值$\{0,1\}$，0 表示不相关、1 表示相关；也可以进行多级相关性标注取值$\{0,1,2\}$，0 表示不相关、1 表示部分相关、2 表示相关。排序任务旨在学习根据查询—文档特征计算文档与该查询的相关度。

1.2 面向排序的交互模型

传统的排序模型基于“评分—排列”[10]机制来构建文档序列。搜索引擎会通过评分函数对每一篇文档单独进行打分，然后直接根据评分进行排序。因此该机制无法建模文档序列中文档之间的交互对排序所带来的影响，与此同时，用户往往最关心序列中的前 K 个文档[11-12]，因而文档的排序位置信息十分重要。然而，基于“评分—排列”机制在评分过程中，搜索引擎无法获取文档的排序位置，无法有效地利用排序位置信息。于此同时，在该机制下，搜索引擎会根据文档评分从大到小对文档进行排序，也就是说，每篇文档仅出现一次，从而导致生成的文档排序训练样本单一。

信息检索领域的研究者们已经尝试采用序列决策过程来建模排序过程，其将文档序列按照排序位置进行分解，每一次决策对应着搜索引擎根据当前排序状态从候选文档中挑选一篇文档放入当前排序位置。在此框架下，可将排序过程看成搜索引擎与

用户不断进行交互的过程，在每一次交互中，搜索引擎根据用户的查询等各种信息，从候选文档集中选出一篇文档传递给用户，而用户浏览该文档并反馈对该文档的评价信息。

1.3　基于 MDP 的排序模型

MDP 是一种经典的建模交互系统的数学模型，如图 1 所示，文档排序过程可以很自然地建模成一个 MDP：每一个时刻对应一个排序位置；在每一个时刻，搜索引擎根据用户当前的信息需求，通过排序模型为当前排序位置选择文档。

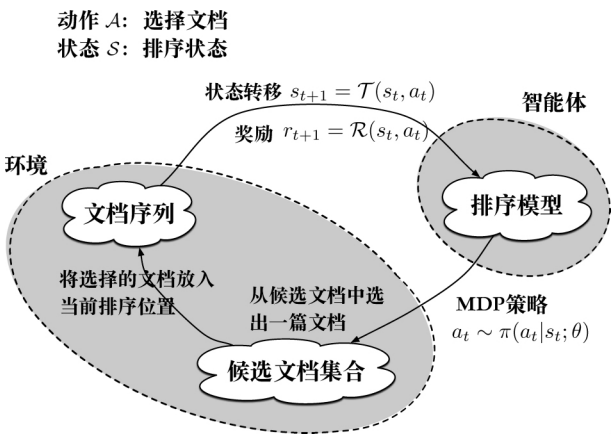

图 1　强化排序模型

用户浏览文档并反馈对该文档的评价信息。具体地，该 MDP 的状态、动作、策略、状态转移和奖励函数的描述如下：

状态：状态是对当前排序环境的描述。在排序过程中，排序策略通过考虑候选文档集合和已排序文档的变化来描述排序状态的演变，因此可以将排序任务中的环境状态描述为三元组，如式(1)所示。

$$s_t=<t,X_t^r,X_t^c> \tag{1}$$

其中，t 表示当前排序位置，X_t^r 表示当前已排序文档，X_t^c 表示当前候选文档集。

动作：动作是搜索引擎影响排序环境的媒介。我们通过当前的可选文档来构建动作集合。由于在排序过程可选文档是不断变化的，因此候选文档集合会依赖于当前状态，可表达为 $A(s_t)$。在 t 时刻，搜索引擎会根据当前策略，选择一个候选文档 $\boldsymbol{x}_{m(a_t)}\in X_t$，将其放入当前文档序列的末尾。其中，$m(a_t)$ 是选择文档的索引。

状态转移：状态转移用于描述排序环境的变迁，其描述了搜索引擎选择一篇文档传递给用户后，排序环境的变化。其中，对于排序位置来说，其会依次递增。对于候选文档集合来说，其会依次减少一篇文档。状态转移函数可以具体的表达为式(2)：

$$\begin{aligned}s_{t+1}&=T(s_t,a_t)\\&=<t+1,X_t^r\oplus x_{m(a_t)},X_t^c-x_{m(a_t)}>\end{aligned} \tag{2}$$

其中，$\boldsymbol{x}_{m(a_t)}$ 为动作 a_t 所选择的文档。

奖励函数：奖励函数用于评价搜索引擎所选文档的质量。对于排序来说，评价一篇文档所能带来的信息收益时，不仅要考虑该文档与查询的相关度，也要考虑当前的排序位置。我们可以基于信息检索评价指标来设计奖励函数。例如，对于相关性排序来说，我们可以在选择文档后，用文档序列的 DCG[13] 增量来计算奖励，如式(3)所示。

$$R(s_t,a_t)=\begin{cases}2^{y_{m(a_t)}}-1 & t=0\\(2^{y_{m(a_t)}}-1)/\log_2(t+1) & t>0\end{cases} \tag{3}$$

其中，$y_{m(a_t)}$ 是所选文档 $d_{m(a_t)}$ 的标注信息。亦或对于多样性排序的任务，我们基于多样性评价标准 α-DCG 来设计奖励函数，如式(14)所示。

$$R(s_t,a_t)=\alpha\text{-DCG}[t+1]-\alpha\text{-DCG}[t] \tag{4}$$

策略：策略用于描述搜索引擎如何为当前排序位置选择文档，一般表达为候选文档的概率分布。我们通过线性函数，根据查询—文档特征，为每一篇候选文档计算得分。在此基础上得到 softmax 策略，自动地平衡在训练过程中的探索和利用。

1.4　模型求解

策略梯度方法[14] 作为一个经典的强化学习方法，其将策略进行参数化表示，直接在参数空间中进行搜索最优策略。基于最优策略，智能体可以从环境中获得最高的奖励。策略梯度的目标函数可以表示为式(5)：

$$J(\boldsymbol{w})=\sum_{s,a\sim\pi}R(s,a) \tag{5}$$

其中，$\boldsymbol{w}$ 为策略的参数。

如图 2 所示，基于策略梯度方法智能体可直接利用交互数据更新策略参数。首先，智能体和环境不断进行交互，产生交互轨迹数据 $(S_0,A_0,R_1,\cdots,S_{M-1},A_{M-1},R_{M-1})$，然后智能体直接利用交互轨迹中的状态、动作和奖励值对策略参数进行更新。其中，REINFORCE[15] 算法中策略梯度的计算如式(6)所示。

$$\Delta\boldsymbol{w}=G_t\,\nabla\log\pi(A_t\mid S_t,\boldsymbol{w}) \tag{6}$$

其中，G_t 为选择动作 A_t 后，搜索引擎所能获取的未来累积收益值，其计算公式可以描述为式(7)。

$$G_t = \sum_{t}^{M-1} R_t \tag{7}$$

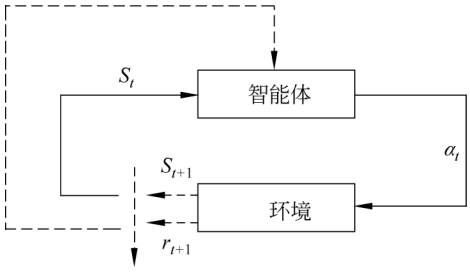

图2 策略梯度方法

综上所述，基于策略梯度的排序模型可以描述为：

（1）搜索引擎依据用户的搜索，获取用户的初始状态 $S_t(t=0)$，根据当前策略 $\pi(a_t, S_t)$ 从候选文档中选择文档，传递给用户；

（2）用户接收搜索引擎选取的文档，浏览该文档序列，反馈奖励值 R_t；

（3）搜索引擎接收到用户的反馈信号，更新用户的状态为 S_{t+1}，搜索引擎再一次为用户选择文档；

（4）搜索引擎考虑更新后的状态，再次选择文档传递给用户，直到用户开始新的搜索任务或者离开；

（5）重复上述过程，可以获得搜索引擎与用户的交互轨迹 $(S_0, A_0, R_1, \cdots, S_{M-1}, A_{M-1}, R_{M-1})$，并根据轨迹信息，依公式计算策略梯度来更新排序模型。

2 强化排序学习的训练效率问题

对于排序任务来说，不同查询的候选文档数目、相关文档所占的比例也不尽相同。为了方便分析，不失一般性，本文假设一个拥有10篇候选文档的查询，其中5篇候选文档与该查询相关，另外5篇为不相关候选文档。在学习初期，搜索引擎会采用随机初始化的MDP策略构建文档序列。如图3中的MDP策略所示，随机初始化的MDP策略，将不相关文档排在序列前端，使得文档序列质量较差。在传统的强化学习框架中，智能体基于MDP策略，不断地和环境进行交互，通过试错的机制进行学习。而不相关的文档无法提供有效的奖励信号。因此，这种情况下，基于强化学习的排序模型在学习训练时，搜索引擎无法获得有效的奖励信息对排序策略进行优化，从而导致学习效率低下。

对于排序任务来说，可以获取训练数据集中所有候选文档的标注信息。因此，本文考虑根据候选文档的标注信息，建立专家策略，来指导搜索引擎和用户之间的交互。如图3所示，通过专家策略，搜索引擎会直接选择信息量最大的候选文档传递给用户，基于最优文档排序序列来学习排序策略，从而可以提高搜索引擎的学习效率。然而，贪心的专家策略构建出的文档列表比较单一，无法提供丰富的文档列表训练数据。与此同时，贪心的策略会导致排序序列的误差累积，从而会影响排序模型的性能。

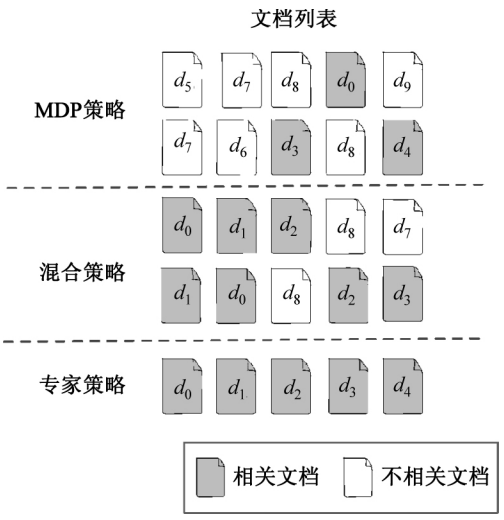

图3 文档列表示意图

已有研究表明，基于数据融合的模仿学习方法可提升传统强化学习的效果，例如，Ross等提出DAGGER算法[16-17]，可以通过专家策略来辅助智能体与环境进行交互，从而获得高质量的交互轨迹数据提高模型学习效率。本文在该模型的基础上依据文档排序任务的特点进行了改进：为了保持策略梯度方法对排序问题强大的建模和学习能力，考虑在获得的混合策略交互轨迹的基础上，直接依照策略梯度的方法来对排序模型进行学习优化，提出了适用于文档排序任务的模仿学习算法

如图3所示，本文结合专家策略和MDP策略构建混合策略来指导搜索引擎构建文档序列。从图3中可以看出，基于MDP策略构建的文档序列具有较强的随机性，包含了大量的不相关文档，使得搜索引擎无法得到有效的奖励信号。基于专家策略构建的文档序列，搜索引擎可以获得大量有效的奖励信号，仅考虑最优排列会导致训练样本的单一性。然而，基于混合策略构建的文档序列在保持文档序列多样性的同时使得构建的文档序列中拥有大量的相关文档，进而提升了搜索引擎的学习效率。总体而言，由于如下两个原因，混合策略比专家策略和

MDP 策略更优。

（1）专家策略每次都会选择最优文档以获得较高的奖励信息，可以保证数据的利用率。但是其会使得到的文档序列分布远离基于 MDP 策略的样本空间，从而导致学习到的策略可能陷于局部最优。

（2）MDP 策略使得搜索引擎根据当前排序模型的策略与用户进行交互，保证了数据分布的一致性。然而在学习初期，排序模型无法构建出高质量的文档序列，从而导致数据利用率低，模型学习效率低下。

3　我们的方法：模仿排序学习

3.1　专家策略

为了提高模型的学习效率，本文通过专家策略来辅助搜索引擎与环境进行交互。根据候选文档的标注信息来构建专家策略。将专家策略定义为选择出该状态下能获得最大奖励值的文档，其可以表示为如式(8)所示。

$$\pi^*(a_t \mid s_t)=\begin{cases}1 & a_t \in \arg\max\limits_{a_t \in A(s_t)} R(s_t,a_t)\\ 0 & \text{else}\end{cases} \tag{8}$$

其中，$R(s_t,a_t)$为奖励函数。

单纯使用专家策略会导致训练数据单一，使得排序序列的误差随着排序位置进行累计，并且造成训练数据分布远离排序模型策略交互的数据分布，影响排序模型的排序性能。

3.2　策略融合

本文通过一个加权参数 β_i 将专家策略与 MDP 策略相结合，构建出一个混合策略，其可以表达为式(9)：

$$\tilde{\pi}(a_t \mid s_t)=\beta_i \cdot \pi^*(a_t \mid s_t)+(1-\beta_i)\cdot \pi(a_t \mid s_t,\boldsymbol{w}) \tag{9}$$

其中，$0\leqslant\beta_i\leqslant 1$ 为当前步下的加权参数，表示专家策略在当前混合策略中所占的比例。为了保证算法的收敛性，加权参数 β_i 需要满足式(10)。

$$\lim_{n\to\infty}\frac{1}{N}\sum\nolimits_{i=0}^{N}\beta_i=0 \tag{10}$$

本文采用了指数衰减的方式来构建加权函数，如式(11)所示。

$$\beta_i=p^i \tag{11}$$

其中，$0<p<1$ 是衰减参数。

可以看出，随着交互的增加，平衡参数 β_i 将由 1 逐渐减少收敛到 0。也就是说，在学习开始阶段，$\beta_i=1$，搜索引擎会完全依赖专家策略与用户进行交互，保证排序模型的学习效率。而随着交互的增加，β_i 的衰减，搜索引擎会越来越信任当前排序模型，MDP 策略在交互中起到的作用越来越大，在保证训练样本多样性的同时，避免文档序列的累积误差所造成的影响，保证排序模型最终的排序性能。

3.3　IR-DAGGER 算法

为了保持强化学习方法强大的学习能力，本文在混合策略交互轨迹的基础上，直接依照策略梯度的方法来对排序模型进行学习优化，提出了一种新的适用于排序任务的基于模仿学习的算法 IR-DAGGER。图 4 展示了基于模仿学习的排序模型的基本框架，在搜索引擎和用户的交互中，搜索引擎会同时考虑专家策略和 MDP 策略来和用户进行交互，进而获得高质量的交互轨迹数据，提高模型学习效率。

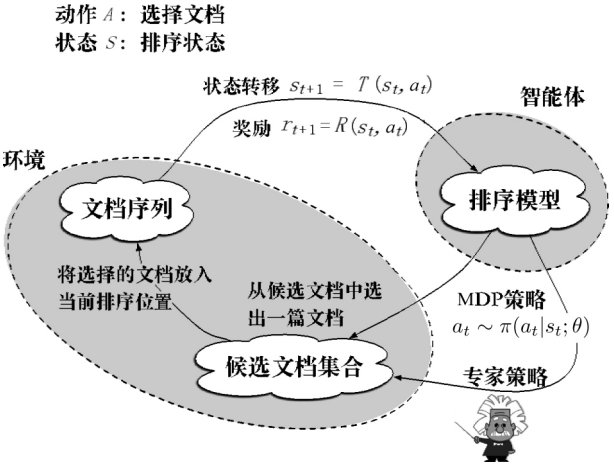

图 4　模仿排序模型

如算法 1 和算法 2 所示，模仿排序学习算法 IR-DAGGER 的主要步骤有 4 个，分别为构建混合策略、搜索引擎和用户进行交互、记忆池更新和策略梯度更新。其具体可以描述如下。

首先初始化一个容量为 C_N 的记忆池，对排序模型参数和平衡参数 β 进行初始化。然后在每一次迭代 $i=1,2,\cdots,N$ 中，依次进行构建混合策略、搜索引擎和用户交互生成训练数据，更新记忆池，进行策略梯度更新。其中：

（1）混合策略构建：我们通过当前的平衡参数，基于专家策略和 MDP 策略构建混合策略，指导搜索引擎和用户的交互行为。

（2）搜索引擎和用户进行交互：首先给定的查询以及查询对应的候选文档得到初始化的状态：$S_0=S(q,X)$。对于每一个时间步 $t=1,2,\cdots,M$，通过采样阈值的方式来实现基于混合策略的交互。首

先从[0,1]均匀分布中采样一个阈值，如果阈值小于当前状态下的平衡参数 β_t，则基于专家策略选择候选文档；反之，则基于 MDP 策略进行交互。然后，根据奖励函数得到当前状态 S_t 下选择动作 A_t 能得到的奖励值。至此可以得到包含状态、动作、奖励的三元组(S_t, A_t, R_{t+1})，将其放入交互轨迹的末端。于此通过根据状态转移函数 T 得到受动作 A_t 影响，变化后状态 S_{t+1}。重复该过程，直到构建出完整的交互轨迹数据 $\tau=(S_0, A_0, R_1, \cdots, S_{M-1}, A_{M-1}, R_M)$

算法 1：IR-DAGGER

输入：训练数据 $D=\{(q^{(n)}, X^{(n)}, Y^{(n)})\}$，参数化策略 $\pi(a_t|s_t, \boldsymbol{w})$，专家策略 $\pi^*(a_t|s_t)$，容量为 C_N 的记忆池 D_m，平衡参数衰减率 p，学习速率 η。
输出：排序策略 $\pi(a_t|s_t, \boldsymbol{w})$

For $i=1$ **to** N **do**
  For all $\{q, X, Y\}\in D$ **do**
    执行混合策略获得交互轨迹 τ
    For $t=0$ **to** $M-1$ **do**
      计算累积奖励 $G_t \leftarrow \sum_{k=1}^{M-t} R_{t+k}$
      存储交互轨迹(S_t, A_t, G_t)至记忆池 D_m
  For all $\{S_k, A_k, G_k\}\in D_m$ **do**
    根据交互轨迹更新策略参数
      $\boldsymbol{w}\leftarrow \boldsymbol{w}+\eta \cdot G_k \nabla \log(A_k|S_k, \boldsymbol{w})$
  更新平衡参数：$\beta\leftarrow\beta \cdot p$

算法 2：执行混合策略

输入：数据(q, X, Y)，参数化策略 $\pi(a_t|s_t, \boldsymbol{w})$，专家策略 $\pi^*(a_t|s_t)$，平衡参数 β，奖励函数R 。
输出：交互轨迹 τ

根据查询和候选文档集合初始化状态 S_0
For $t=1$ **to** M **do**
  随机采样参数 $\hat{\beta}\sim$Uniform[0,1]
  if $\beta>\hat{\beta}$ **then**
    采样动作 $A_t\sim\pi^*(a_t|s_t)$
  else
    采样动作 $A_t\sim\pi(a_t|s_t, \boldsymbol{w})$
  计算奖励值 $R_{t+1}=R(A_t, S_t)$
  存储交互信息 $\tau=\tau\oplus(S_t, A_t, R_{t+1})$
  状态转移 $S_{t+1}\leftarrow T(S_t, A_t)$

(3) 记忆池更新：在初始化阶段，我们可以得到一个容量为 C_N 的记忆池，用于存储搜索引擎和用户交互所产生的训练数据。在训练的过程中，得到新的交互轨迹 τ 后，如果记忆池还有容量，则直接将交互轨迹存储至记忆池；而如果记忆池已达到分配的容量，则需要随机地删除一条交互轨迹，为新的交互轨迹提供存储空间。

(4) 策略梯度更新：我们对于记忆池中的每一条交互数据，计算策略梯度，并根据策略梯度进行策略梯度更新。

4 实验分析

我们考虑相关性排序[5]和多样性排序[6]的场景，通过实验从性能和效率两个方面对本文提出的模仿排序学习模型进行分析。

4.1 相关性排序

4.1.1 实验设置

本文在经典的排序学习数据集 OHSUMED[18] 之上进行了相关性排序的实验。该数据集有 106 条查询，均来源于医生在给病人看病过程中所提交的查询字符串。OHSUMED 数据集一共标注了 16 140 个查询—文档对，包含了 2 557 个相关、2 932 个部分相关以及 12 948 个不相关的查询-文档对。

为了验证 IR-DAGGER 的有效性，我们考虑了相关性排序中多类的基准方法，包括排序学习的方法 Regression、RankSVM[19]、RankBoost[20]、ListNet[21]、SVMMAP[22]、AdaRank-MAP[23]、AdaRank-NDCG[23]、MLL，以及强化学习方法 MDPRank。我们采用传统的信息检索评价指标 MAP、Precision@5、Precision@10、NDCG@5 和 NDCG@10 对实验结果进行评价。

在实验中，本文测试了平衡参数的衰减指数取 $p\in\{0, 0.9, 1\}$ 的情况。当 $p=0$ 时，则 $\beta=0$，搜索引擎和用户仅通过 MDP 策略进行交互。当 $p=1$ 时，则 $\beta=1$，搜索引擎和用户仅通过专家策略进行交互。当 $p=0.9$ 时，初始时 $\beta=1$，搜索引擎通过专家策略进行交互，而随着迭代次数的增加，β 会以 0.9 的概率衰减，搜索引擎逐渐地增加通过 MDP 策略进行交互的概率，并最终主要以 MDP 策略与用户进行交互。

搜索引擎能获得高质量的训练数据，快速地提高排序模型的性能。随着排序模型性能的提高，MDP 策略所占的比例逐步提高，使得交互数据分布与排序模型策略交互数据差异性逐步缩小，进一步提高排序模型的性能。

4.1.2 实验结果

表 1 展示了我们的方法和其他所有基准方法，包括 Regression、RankSVM、RankBoost、ListNet、SVMMAP、AdaRank-MAP、AdaRank-NDCG、List-

MLE，以及强化学习方法 MDPRank，在 5 个评价准则上的表现，包括 NDCG @ 5，NDCG @ 10，Precesion@5，Precesion@10 和 MAP 上的实验结果。其中粗体表示所有实验中最好的结果。从结果可以看出，我们的方法 PPG 在 5 个相关性评价准则指标上均得到了最好的性能。

表 1　基于面向相关性排序的 OHSUMED 数据集上的排序精度

方法	NDCG@5	NDCG@10	Precesion@5	Precesion@10	MAP
Regression	0.427 8	0.411 0	0.533 7	0.466 6	0.422 0
RankSVM	0.416 4	0.414 0	0.531 9	0.486 4	0.433 4
RankBoost	0.449 4	0.430 2	0.544 7	0.496 6	0.441 1
ListNet	0.443 2	0.441 0	0.550 2	0.497 5	0.445 7
SVMMAP	0.451 6	0.431 9	0.552 3	0.491 0	0.445 3
AdaRank-MAP	0.461 3	0.442 9	0.567 4	0.497 6	0.448 7
AdaRank-NDCG	0.460 6	0.440 8	0.567 2	0.501 1	0.449 2
ListMLE	0.464 0	0.452 0	0.546 4	0.498 3	0.453 3
MDPRank	0.486 7	0.470 9	0.565 5	0.507 9	0.460 5
IR-DAGGER($p=0$)	0.464 9	0.450 9	0.555 7	0.497 4	0.454 8
IR-DAGGER($p=1$)	0.486 6	0.470 4	0.573 1	0.511 7	0.460 5
IR-DAGGER($p=0.99$)	**0.493 5**	**0.473 2**	**0.576 9**	**0.511 7**	**0.461 2**

图 5 展示了取不同衰减率 $p=\{0,0.9,1\}$ 时 IR-DAGGER 的学习曲线，以及 $p=0.9$ 时平衡参数 β 的衰减曲线。可以看出，在学习初期，受益于专家策略能够给出最优排序序列训练样本，IR-DAGGER ($p=1$)快速收敛，其收敛明显快于其他排序模型。然而，随着迭代次数的增加，仅依靠 MDP 策略的排序模型 IR-DAGGER($p=0$)性能逐步提升，在将近 300 次迭代时，其性能超过 IR-DAGGER($p=1$)，从而验证了 MDP 策略可以增加文档序列训练样本的多样性，减少文档累积误差所带来的影响。IR-DAGGER($p=0.9$)在学习初期可以通过专家策略，获得高质量的交互轨迹，随着学习增加搜索引擎通过 MDP 策略与用户进行交互保证了排序模型的性能，最终取得了最好的排序结果。

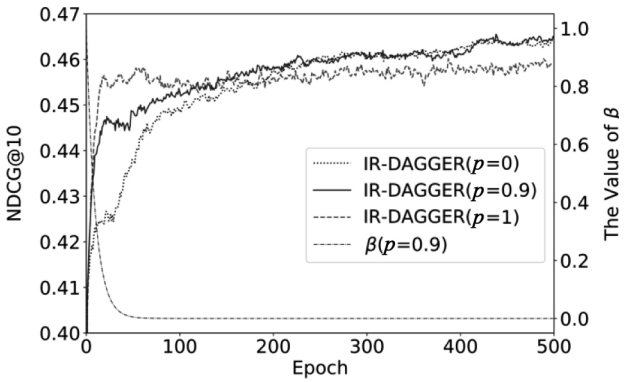

图 5　IR-DAGGER 在 OHSUMED 数据集上的收敛速度

4.2　多样化排序

4.2.1　实验设置

MDP 也可以用于多样化排序建模[6]，被称为 MDPDIV。具体地，MDPDIV 将搜索引擎视为智能体，将用户视为环境，通过循环递归网络(RNN)，根据用户查询以及已排序的文档来表达多样性排序过程中的排序状态，同时通过一个双线性函数，根据当前排序状态以及文档的表达来计算对文档的评分，构建 Soft-Max 策略作为 MDP 策略。本文所提出的模仿学习算法 IR-DAGGER 也可以用于提升 MDPDIV 的效果。

根据文献[6]中的设置，本文将四个 TREC 基准测试数据集(TREC 2009 Web Track，TREC 2010 Web Track，TREC 2011 Web Track and TREC 2012 Web Track)合并为一个数据集合，一共包含 200 个查询和 45 000 个带标注信息的查询-文档对，候选文档从 ClueWeb09Category B (包含约 5 000 万英文网络文档)语料库中获得。实验中，查询和文档的分布式向量表达基于 GENSIM 库获取。

在实验中，已标注的查询被随机等分为五份并进行了五折交叉验证。为了验证 IR-DAGGER 的有效性，实验所选择的基线方法包括启发式的排序模型 MMR、XQuAD 和 PM-2，基于学习的排序模型 SVM-

DIV,R-LTR,PAMM,NTN-DIV和ListMLE,以及基于强化学习的排序模型MDP-DIV。与此同时,平衡参数的衰减指数p取值为0,0.99或者1。

4.2.2 实验结果

表2展示了IR-DAGGER和所有基线方法的多样化排序精度,评价准则包括α-NDCG@5,α-NDCG@10,ERR-IA@5和ERR-IA@10,其中粗体表示精度最高的结果。可以看出,基于模仿排序学习的方法IR-DAGGER在所有评价准则指标上均得到了最好的性能。

表2 面向多样化排序的TREC数据集上的排序精度

方法	α-NDCG@5	α-NDCG@10	ERR-IA@5	ERR-IA@10
MMR	0.275 3	0.297 9	0.200 5	0.230 9
XQuAD	0.316 5	0.394 1	0.231 4	0.289 0
PM-2	0.304 7	0.373 0	0.229 8	0.281 4
SVM-DIV	0.303 0	0.373 0	0.229 8	0.281 4
R-LTR	0.349 8	0.413 2	0.252 1	0.301 1
PAMM	0.371 2	0.432 7	0.261 9	0.302 9
NTN-DIV	0.396 2	0.457 7	0.271 8	0.328 5
ListMLE	0.394 5	0.438 6	0.309 7	0.331 0
MDP-DIV	0.449 1	0.487 4	0.349 8	0.370 3
IR-DAGGER($p=0$)	0.355 8	0.412 1	0.271 8	0.297 8
IR-DAGGER($p=1$)	0.459 6	0.497 7	0.357 6	0.378 5
IR-DAGGER($p=0.99$)	**0.468 8**	**0.509 0**	**0.365 9**	**0.387 3**

图6展示了p取不同衰减率时IR-DAGGER的学习曲线;以及$p=0.99$时平衡参数的衰减曲线,可以看出,通过利用专家交互轨迹信息,可以极大地提高多样性排序学习算法的学习效率。

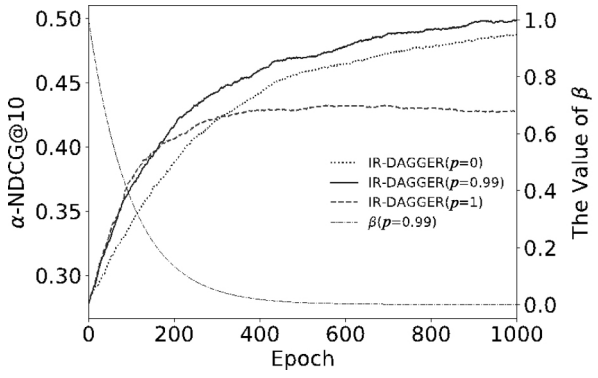

图6 IR-DAGGER在TREC数据集上的收敛速度

5 总结

本文研究了基于MDP的排序学习模型,通过引入专家策略,从数据融合的角度提出基于模仿学习的排序模型IR-DAGGER,提升了基于排序模型的排序精度和学习效率。具体而言,IR-DAGGER融合了MDP策略和专家策略,其在算法训练初期通过专家策略与用户进行交互,从而获得高质量的交互轨迹数据,提升学习效率;其在算法运行中后期逐步增加MDP策略在交互策略中所占比例,有效避免训练数据集中累积误差的出现,提高了模型的排序精度。基于相关性排序任务和多样性排序任务的实验结果表明,相对于基于强化学习的排序算法,本文提出的IR-DAGGER算法在排序精度和学习效率上都有较大提升。

参考文献

[1] Liu T Y. Learning to rank for information retrieval[J]. Foundations and Trends in Information Retrieval, 2009, 3(3): 225-331.

[2] Derhami V, Khodadadian E, Ghasemzadeh M, et al. Applying reinforcement learning for web pages ranking algorithms[J]. Applied Soft Computing, 2013, 13(4): 1686-1692.

[3] Hofmann K, Whiteson S, De Rijke M. Balancing exploration and exploitation in learning to rank online[C]//Proceedings of the European Conference on Information Retrieval. Springer, Berlin, Heidelberg, 2011: 251-263.

[4] Howard R A. Dynamic programming and Markov processes[J]. The Mathmatical Gazette, 1972, 46

(358)：120.
[5] Zeng W，Xu J，Lan Y，et al. Reinforcement learning to rank with Markov decision process[C]//Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM，2017：945-948.
[6] Xia L，Xu J，Lan Y，et al. Adapting Markov decision process for search result diversification[C]//Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM，2017：535-544.
[7] Zeng W，Xu J，Lan Y，et al. Multi Page Search with Reinforcement Learning to Rank[C]//Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval. ACM，2018：175-178.
[8] Luo J，Zhang S，Yang H. Win-win search：Dual-agent stochastic game in session search[C]//Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval. ACM，2014：587-596.
[9] Sutton R S，Barto A G. Reinforcement learning：An introduction[M]. MIT press，2018.
[10] Fan D P. Information processing analysis system for sorting and scoring text[P]. US：Patent 5，371，673. 1994-12-6.
[11] Fagin R，Kumar R，Sivakumar D. Comparing top k lists[J]. SIAM Journal on Discrete Mathematics，2003，17(1)：134-160.
[12] Ilyas I F，Beskales G，Soliman M A. A survey of top-k query processing techniques in relational database systems[J]. ACM Computing Surveys (CSUR)，2008，40(4)：11.
[13] Järvelin K，Kekäläinen J. Cumulated gain-based evaluation of IR techniques[J]. ACM Transactions on Information Systems (TOIS)，2002，20(4)：422-446.
[14] Sutton R S，McAllester D A，Singh S P，et al. Policy gradient methods for reinforcement learning with function approximation[C]//Proceedings of Advances in Neural Information Processing Systems，2000：1057-1063.
[15] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine Learning，1992，8(3-4)：229-256.
[16] Ross S，Gordon G，Bagnell D. A reduction of imitation learning and structured prediction to no-regret online learning[C]//Proceedings of the 14th International Conference on Artificial Intelligence and Statistics，2011：627-635.
[17] Ross S，Bagnell J A. Reinforcement and imitation learning via interactive no-regret learning[J]. arXiv preprint arXiv：1406.5979，2014.
[18] Hersh W，Buckley C，Leone T J，et al. OHSUMED：An interactive retrieval evaluation and new large test collection for research[C]//Proceedings of the SIGIR'94. Springer，London，1994：192-201.
[19] Burges C，Shaked T，Renshaw E，et al. Learning to rank using gradient descent[C]//Proceedings of the 22nd International Conference on Machine Learning (ICML-05)，2005：89-96.
[20] Freund Y，Iyer R，Schapire R E，et al. An efficient boosting algorithm for combining preferences[J]. Journal of Machine Learning Research，2003，4 (Nov)：933-969.
[21] Cao Z，Qin T，Liu T Y，et al. Learning to rank：From pairwise approach to listwise approach[C]//Proceedings of the 24th International Conference on Machine Learning. ACM，2007：129-136.
[22] Yue Y，Finley T，Radlinski F，et al. A support vector method for optimizing average precision[C]//Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM，2007：271-278.
[23] Xu J，Li H. AdaRank：A boosting algorithm for information retrieval[C]//Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM，2007：391-398.

曾玮(1991—)，博士研究生，主要研究领域为强化学习、信息检索。
E-mail：zengwei@ict.ac.cn

俞蔚捷(1992—)，博士研究生，主要研究领域为强化学习、信息检索。
E-mail：yuweijie@ruc.edu.cn

徐君(1979—)，通信作者，博士，教授，主要研究领域为排序学习、信息检索。
E-mail：junxu@ruc.edu.cn