细节决定成败：推荐系统实验反思与讨论

施韶韵, 王晨阳, 马为之, 张　敏, 刘奕群, 马少平

清华大学计算机系, 北京信息科学与技术国家研究中心, 北京 100084

摘要　近些年来, 随着互联网的迅速发展, 用户在各种在线平台上接收到海量的信息, 信息爆炸成为一个关键性问题。在此背景下, 推荐系统逐步渗透到人们工作生活的各个场景, 已成为不可或缺的一环。它不仅可以帮助用户快速获得想要的信息和服务, 还可以提高资源利用效率, 从而给企业带来更多效益。因此, 个性化推荐算法不仅获得了工业界广泛的关注, 也是科研领域的研究热点之一。在个性化推荐的研究中, 受限于平台与效率等因素, 研究者大多无法将算法部署到在线系统上进行评价, 因此离线评价成为推荐领域研究的主要方式。然而个性化推荐涉及到的场景复杂, 可获得的数据信息多种多样, 用户行为多为隐式反馈且存在许多噪声, 这使得推荐系统离线评价的实验设定复杂多变, 存在大量易被忽视却十分重要的细节。比如在训练采样负例时, 既可以仅从用户没有交互过的商品中采样, 也可以将验证测试集的商品视作未知交互加入采样池。同样, 从训练到测试在很多其他环节也涉及这样的实现细节(如数据集处理、已知负样本的使用、Top-*N* 排序候选集范围等)。这些实验细节通常不会在学术论文中被显式提及, 却潜在影响了模型效果的对比, 还决定着实验的科学性, 甚至会导致相反或错误的分析结论。本文从数据集处理、模型训练、验证与测试、效果评价等多个角度, 系统地讨论与反思了推荐系统实验中的细节设定。对于每个环节, 我们枚举了若干常见设定, 并在真实数据集上验证了其中某些设定的实际影响。实验结果表明一些细节确实会导致关于模型优劣的不同结论。最终我们形成了关于推荐系统实验细节的指导性总结, 包括可选、建议、必须的三类设定, 希望帮助推荐算法研究者规避实现细节上的陷阱, 更科学合理地设计实验。

关键词　推荐系统; 实验设计; 算法评价; 反思与讨论

中图法分类号　TP181 自动推理、机器学习　DOI 号 10.19363/J.cnki.cn10-1380/tn.2021.09.04

Details Matter: Revisiting Experimental Settings in Recommender Systems

SHI Shaoyun, WANG Chenyang, MA Weizhi, ZHANG Min, LIU Yiqun, MA Shaoping

Department of Computer Science and Technology, Institute for Artificial Intelligence,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China

Abstract　With the development of Internet in recent years, information overload has become a critical issue for users on various online platforms. To address this issue, recommender system stands out and comes to constitute a vital part in people's daily life. It not only makes it easier for users to access the information and services in need, but also brings benefits for companies by improving resource utilization. Therefore, personalized recommendation algorithms have gained increasing attention in industry and have attracted a surge of research interests in the meantime. Restricted by practical factors such as platform and efficiency, many researchers in personalized recommendation have no access to online systems to evaluate their algorithms. Thus, offline evaluation becomes the most common practice in the research area. However, different recommendation scenarios involve heterogeneous types of data. Furthermore, most user behaviors are implicit feedback with plenty of noises. These factors lead to complicated and divergent experimental settings in offline recommendation experiments. In practice, many important details are easy to be neglected and different researchers may have different perceptions towards detailed settings. For example, it can be a question that whether the item pool of negative sampling during training should include the interacted items in the valid/test dataset. One may simply sample negative items from non-interacted items or also view valid/test items as possible negative items. Similarly, there also exist various detailed settings during other processes, from training to testing (e.g., data preprocessing, the usage of known negative samples, the choice of candidates in Top-*N* ranking). These experimental details are usually omitted in the writing of research papers but potentially affect the comparison between recommendation algorithms. Besides, these settings somewhat determine the scientificity of experiment designs and some of them may even lead to opposite or wrong conclusions. Given these observations, this work thoroughly revisits the details in different aspects of recommendation experiments, including data preprocessing, model training, validation, testing, and evaluation metrics. We enumerate the common

通讯作者：张敏, 博士, 特别研究员, Email: z-m@tsinghua.edu.cn

本课题得到国家重点研发计划(No. 2018YFC0831900)、国家自然科学基金(No. 61672311, No. 61532011)和清华大学国强研究院资助。
收稿日期: 2021-05-09; 修改日期: 2021-08-05; 定稿日期: 2021-08-10

choices in each aspect and some are coupled with empirical experiments to demonstrate the effects of different experimental settings. We show that some settings indeed lead to flipped positions when ranking different recommendation algorithms. Finally, a guiding summary of experimental details is concluded, involving principles that are optional, suggestive, or necessary to be adopted. With the help of this summary, researchers are more capable of avoiding possible implementation traps and designing recommendation experiments in a scientific way.

Key words recommender system; experimental settings; algorithm evaluation; revisiting and discussion

1 引言

在当今泛信息化时代, 每个用户不仅是信息的消费者, 同时也是信息的生产者, 信息资源的爆炸式增长不可避免地造成了信息过载问题。为了提升用户获取并利用信息的效率, 信息检索技术应运而生。信息检索技术关系到社会整体的信息处理水平与利用效率, 对信息化社会发展水平有着重要意义。以搜索引擎为代表的信息检索是人们主动获取信息的一种重要方式, 但它依赖于用户基于明确的信息需求, 检索系统只能被动地提供信息。推荐系统作为信息检索的另一个重要发展方向, 提供了一种系统主动提供信息的方式, 能够通过对用户进行侧写建立用户画像, 主动从海量信息中提供最契合用户偏好的内容。这使得用户可以更高效地获取信息, 同时也提高了资源利用效率, 从而给内容提供者(平台、公司等)带来更多效益, 很大程度上缓解了互联网信息过载的问题[1]。因此, 个性化推荐算法不仅获得了工业界各大公司的关注, 也是科研领域的研究热点之一。

在推荐领域的研究中, 受限于平台、效率等因素, 研究者往往无法通过在线评价的方式将算法部署到实际系统上进行对比。因此, 使用系统日志记录的离线评价方式成为研究者们的主要选择。在离线评价的设定下, 研究者往往首先对公开的推荐数据(如MovieLens、Amazon)进行预处理, 划分出训练、验证与测试集, 然后在划分的数据集上训练并测试不同的推荐算法, 根据测试集上的评价指标评估算法间的优劣。对于新提出的算法, 能否得到超过基线模型的结果被认为是衡量有效性的主要途径。然而个性化推荐涉及到多种不同的场景, 系统日志中可获得的信息多种多样, 用户行为多为隐式反馈且存在许多噪声, 这使得离线评价时的实验设定复杂多变, 存在大量易被忽视却十分重要的细节。

实验细节决定了实验的科学性, 并且可能涉及离线评价的各个阶段。比如在数据预处理时相同时间戳的交互怎么排序、数据中如果存在已知负例(呈现给用户但未交互)如何使用、训练时负例的采样范围如何设定、Top-N 排序评测时候选集如何选择等。这些问题由于学术论文的篇幅限制, 大多并不会在论文中显式地描述。研究者往往根据自己的理解进行实验, 这导致不同工作的实验设定千差万别, 不仅难以直接对比不同论文间的实验结果, 不少算法本身的结果也无法复现。近年来有工作[2]就对推荐领域的可复现性产生了质疑, 在所选的 18 篇近期顶级国际会议论文中, 只有 7 个方法可以被成功复现。推荐领域表面上的百花齐放可能隐藏着长时间的停滞不前。Rendle 等人[3]也提出对于基线模型的调优是非常困难的, 基础的矩阵分解模型经过精心调参后甚至能够超过大多数近年来新提出的算法。某些实验设定的细节也可能带来一定的不公平性, 导致基线模型难以取得较优的结果, 从而得到有误导性的甚至错误的分析与结论。

这些现象和问题不可避免地对推荐领域的研究带来了阻碍: (1) 对研究者来说, 选择合适的基线模型变得越来越困难, 仅通过论文中的描述很难确定作者所采取的实验设定是否科学, 不同论文之间的结果也基本无法对比, 往往只能通过参考代码复现的方式确定算法优劣, 需要耗费大量时间对他人的方法进行调优; (2) 对于研究社群来说, 也很难确定某一个具体任务上最先进(state-of-the-art)的模型到底是哪一个。在这种环境下, 有投机想法的研究者会倾向于选择最近发表但效果较差的算法作为主要对比对象, 而审稿人往往只根据对比模型发表的时间来判断是否合适。这导致真正有效的模型少有人去对比, 从而得不到相应的关注, 产生劣币驱逐良币的现象。

针对目前推荐系统实验的上述问题, 本文从数据集处理、模型训练、验证与测试、效果评价等角度, 详细梳理了推荐算法实验的一般过程(尤其针对近年来发展迅速的深度推荐模型), 系统地讨论与反思了实验过程中各个阶段需要注意的细节设定。对于一些在论文中不常提及的实验设定, 我们还在公开的真实数据集上进行了实证研究, 并分析了实验细节的变化会对模型对比带来怎样的影响。通过实验我们发现, 有些研究者容易忽略的细节, 可能对算法结果有着出乎意料显著的影响(比如训练时负采样是否包括验证测试集交互的商品)。这种影响在数

据集规模较小时, 甚至会导致算法间的优劣关系出现变化, 得到误导性的错误结论。基于这些实验与讨论, 我们最后总结形成了推荐系统实验细节的指导性总结, 包括可选、建议、必须 3 种类别的设定, 希望能推动研究者形成实验设定方面的共识, 更加科学合理地设计推荐实验, 促进推荐研究的健康发展。

本文剩余部分将按如下结构组织: 第 2 章介绍与推荐实验相关的工作; 第 3 到第 6 章分别从数据集、模型训练、验证与测试、评价指标 4 个方面回顾推荐系统实验涉及到的种种实验细节及影响; 最后第 7 章对全文进行总结。

2 相关工作

2.1 推荐实验反思

在推荐场景下, 输入的数据通常可以表示为一个“用户-商品”矩阵, 其中非空的单元表示观察到了对应用户对商品的交互, 这里交互可能是商品评分、听一首歌、点击一个链接、留下评论等等。基于以上输入, 推荐系统旨在对于给定的用户和商品, 给出一个表示用户对商品喜好程度的分数。在较早的推荐领域研究中, 评分预测(Rating Prediction)是主要的研究任务。评分预测任务下推荐实验的设定较为简单, 典型的评测方式是在随机划分出的测试集上计算模型预测评分与用户实际评分之间的均方根误差(RMSE)。为了契合推荐系统的实际应用场景, 即从海量商品中得到针对用户的个性化推荐列表, 近年来大多数工作转向采用 Top-*N* 排序任务对推荐算法进行评价。这种设定下需要将目标商品和一批候选商品共同排序, 通过 HR(Hit Ratio)、NDCG(Normalized Discounted Cumulative Gain)等评价指标衡量排序列表的质量[4]。Top-*N* 排序任务虽然更加符合推荐系统的定位, 但涉及的实验细节显著增多, 特别是随着深度学习的快速发展, 这些实验设定并未在研究者间形成共识, 不同工作的细节设定千差万别, 推荐领域的可复现性受到了严峻的挑战。

近年来, 陆续有工作对推荐领域国际顶级会议的论文可复现性提出质疑。Rendle 等人[3]发现在许多工作中并没有对基线模型进行深入的调优, 研究者会通过有选择的实验结果来彰显所提出模型的优越性。经过细致地调参后, 简单的矩阵分解模型就能够超过大多数近年来提出的新算法。Dacrema 等人[2]尝试复现一系列近期推荐算法的结果, 但能够成功复现的仅占非常少的一部分。有些论文要么没有公开实现代码, 要么结果因为某些没有公开的细节处理而无法重现。对于能够重现结果的论文, 也有很多存在基线模型选择的问题, 有些理应被比较的模型没有被纳入考虑。

基于以上对推荐领域可复现性的质疑, 有工作[5]针对推荐系统离线评测下的 Top-*N* 排序任务, 总结了评测过程中的不同选择, 包括数据集过滤与划分、缺失值处理、去重、显著性检验等 9 个分歧点, 通过实验说明了在不同评测设定的组合下甚至会得到相反的结论, 期望能启发研究者们形成统一的评测规范。此外, 有些工作关注 Top-*N* 排序时评价指标计算的设定。由于基于排序的评价对于每个用户都需要对整个候选商品集合进行排序, 如果商品规模较大的话会不可避免地带来效率问题, 因此目前的常见做法是随机采样一些用户没有交互过的商品(如 100 个), 将目标商品和这些“负样例”一同排序, 衡量这 101 个候选商品的排序结果[6-7]。Krichene 等人[8]对基于采样以及基于全量商品的排序指标计算进行了研究, 发现基于采样的评价方式在某些情况可能会导致和全量评测不一致的结果。并且采样集合越小算法间指标的差异越小, 当采样集合特别小时, 所有的评价指标都将退化为 AUC, 这对一直以来沿用的基于采样的排序评测方法提出了挑战。但同时也有工作提出不同的观点, Li 等人[9]从理论上分析了 HR 在采样和非采样设定下的对应关系, 证明存在从采样指标到非采样指标的一个映射。

我们的工作延续了对推荐系统实验进行反思和讨论的研究方向, 但不同于以上工作的是我们关注更细节的实验设定, 尤其是经常不在学术论文中提及的, 甚至是研究者不曾注意的细节设定, 比如训练时配对样本(pair-wise)训练负采样是否应当排除验证测试集交互过的商品。这些细节往往在研究者搭建系统时会想当然地根据自己的认知进行实现, 却可能带来潜在的性能影响, 在进行效果对齐实验时很难被研究者自行发现。因此我们通过对此类细节设定的总结与讨论, 希望能够为研究者提供经验性的指导。

2.2 推荐算法框架

近年来, 有越来越多的工作致力于构建一个统一的训练评测框架, 使得不同模型能够在同样的设定下进行对比, 希望在工程实践方面推动研究社区的建设。例如早期的 LibFM[10]、Librec[11]主要关注传统推荐算法, 任务上以评分预测为主。随着深度学习在推荐领域的应用越来越广泛, 出

现了以 OpenRec[①][12]、DeepRec[②][13]、TFRS[③]、NeuRec[④]、RecBole[⑤]、ReChorus[⑥]为代表的深度推荐框架。NeuRec 基于 TensorFlow 进行开发, 实现了多种基于神经网络的推荐算法。RecBole[14]基于 PyTorch 进行开发, 不仅实现了多种推荐模型, 还包含了常用的数据集以及相关预处理的代码, 形成了一个完整的推荐实验平台。ReChorus[15]则以轻量灵活为特点, 用尽可能简洁的代码规定了训练评测的常见范式, 方便研究者快速上手进行模型扩展。基于这些框架, 研究者可以沿用框架开发者预设的数据处理与训练评测流程, 而无需关注具体细节, 能够专注在模型的设计上。此外由于框架的代码开源, 并且近年发布的框架大多都支持参数自动搜索(如 RecBole、ReChorus), 一定程度上缓解了推荐领域的可复现性问题。

然而这些工作只是根据开发者的经验对实验细节进行了预设, 没有形成指导性的总结, 并且不同框架间的细节设定也存在差异。我们则基于实践经验以及对这些框架的使用情况, 系统地反思与讨论了其中涉及到的实验细节, 帮助研究者以及框架开发者更科学合理地设计实验。

3 数据集

3.1 常见数据集

个性化推荐作为一个在工业界广泛应用的场景, 有大量的真实环境下的用户数据, 许多公司公开了一些实验数据集供科研使用。例如, 比较著名的电影领域公开数据集 MovieLens[⑦][16]包含了用户对电影的五级评分, 按评分数量分为 10 万、100 万、1000 万、2000 万等多个版本, 每个都包含了电影的名字、年代、类别等信息。较小的版本还包含用户年龄、性别、职业等用户属性, 是评分预测的经典数据集。电商领域的著名数据集有圣地亚哥加利福尼亚大学的 McAuley 团队和亚马逊公司在 2014 年合作公开的 Amazon[⑧][17-18]数据集, 该数据集包含了用户在亚马逊购物网站上的大量评分和评论文本, 还提供了大量商品的标签、图片、品牌、价格等信息, 是基于评论的推荐模型首选的实验数据集之一。为了便于离线实验, 发布者按亚马逊商品分类提供了多个每个领域下的子数据集, 大小各不相同供实验者使用。之后发布者还更新了 2018 版本的数据集[⑨], 提供了更多的商品类别, 更大量的评分和评论数据。另外, 每年许多公司和组织都会举办推荐系统相关的竞赛, 一来为公司提升算法性能提供新思路, 二来促进领域科学发展, 同时也会发布公开的竞赛数据集。比较著名的竞赛有 RecSys Challenge, 每年和国际推荐系统顶级国际会议 RecSys 联合举办[⑩]会涉及工作推荐、音乐推荐等实际业务场景。除此之外, 还有许多其他例如 Yelp[⑪]、Book-Crossing[⑫]、LastFM[⑬], 在此不一一列举。每个数据集包含不同的信息, 有不同的数据分布, 研究者们需要根据具体的研究场景选择合适的数据集进行实验。

3.2 数据集划分

虽然这些公开数据集为研究者们提供了评测模型的基准, 但是由于推荐场景的复杂性, 在不同的细分任务和研究场景中, 往往会涉及不同的数据集划分方式, 并且每个研究工作处理数据集的过程中的细节也会有所不同。一般来说, 大家把一个交互, 即用户和商品对, 作为一个训练或测试样本。样本输入是用户和商品的信息, 有时例如序列推荐还包括用户交互历史。标签一般是多级评分, 或是否点击、购买, 与输入信息一起构成完整的一个样本。在机器学习任务中, 为了准确地训练和评估模型, 一般会将样本集合划分为训练集、验证集和测试集。训练集用来学习模型参数, 验证集用来调整模型超参数, 测试集用来测试模型效果。比较常见的划分方式包

① https://github.com/ylongqi/openrec
② https://github.com/cheungdaven/DeepRec
③ https://www.tensorflow.org/recommenders?hl=zh-cn
④ https://github.com/wubinzzu/NeuRec
⑤ https://github.com/RUCAIBox/RecBole
⑥ https://github.com/THUwangcy/ReChorus
⑦ https://grouplens.org/datasets/movielens/
⑧ http://jmcauley.ucsd.edu/data/amazon/
⑨ https://nijianmo.github.io/amazon/index.html
⑩ https://recsys.acm.org/
⑪ https://www.yelp.com/dataset
⑫ http://www2.informatik.uni-freiburg.de/~cziegler/BX/
⑬ https://grouplens.org/datasets/hetrec-2011/

括但不限于以下几种:

(1) 随机划分: 随机划分是指按一定比例(例如 8∶1∶1)随机将所有样本划分为训练集、验证集、测试集, 或者随机分为 K 折进行交叉验证。该划分方式比较适用于评分预测或点击预测任务, 且要求样本之间关联性不是很强, 模型不依赖于序列信息。例如一些利用协同过滤做评分预测的方法[19-21], 一般使用矩阵分解填充和估计用户商品矩阵中的未知元素, 而矩阵中元素没有时间依赖关系, 可以使用这样的划分方式。而现有许多推荐算法依赖于用户交互历史序列, 如此划分有可能造成测试集信息在训练集中泄露。例如某用户的两个交互, 时间靠前的被划分到了测试集, 时间靠后的被划分到了训练集(如图 1 所示), 训练集样本对应的历史交互序列可能包含了测试集交互样本, 使得信息泄露, 模型如果建模了交互序列会使得测试效果偏高。

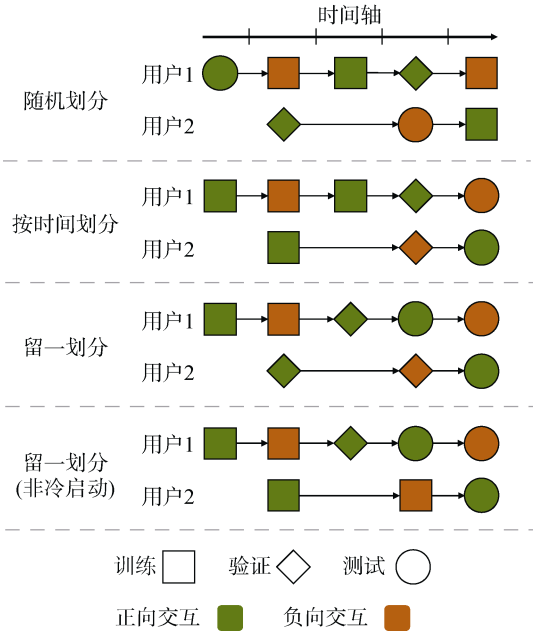

图 1 数据集划分方式示意

Figure 1 Illustration of dataset division

(2) 按时间划分: 对于建模了交互序列或是时间敏感的任务, 例如新闻推荐场景下许多新闻有时效性, 按时间节点划分数据集是一个比较常用的方式[22-23]。例如对于一个有时间跨度 10 天的数据集, 可以取最后一天的数据作为测试集, 倒数第二天的数据作为验证集, 其余作为训练集(如图 1 所示)。这样和真实线上环境比较接近, 在训练时模型完全无法获得未来时刻的信息。但该划分方式不能保证验证或测试集中的用户一定有正向交互, 会对 Top-N 排序任务的评测带来一定影响。并且容易带来冷启动问题, 因为每天系统中都会产生新的用户和商品, 那么会有一些用户和商品只在最后的一段时间出现过而不在训练集中, 构成冷启动用户或商品。根据具体研究课题是否考虑冷启动问题, 可以选择性去除这一部分用户和商品。

(3) 用户交互序列留一划分: 另一个比较常用的方式是按用户交互序列进行留一划分。对于每一个用户, 将其最后一个正交互(及其时间之后的负向交互)作为测试样本, 倒数第二个正向交互(及其时间之后且在最后一个正样本之前的负向交互)作为验证集样本, 其余为训练集[6-7,24](如图 1 所示)。根据是否研究冷启动问题, 可以考虑保证用户的前 N 个正向交互一定在训练集, 这样可以保证在验证和测试阶段出现的用户一定在训练集中出现过。对于特别大的数据集, 为提升验证和测试效率, 可以限制验证测试集中的用户数。该方法与按时间划分几点不同: 保证验证和测试集中的用户有正向交互, 比较适用于 Top-N 排序任务的评测; 每个用户在验证和测试集中最多只会出现一次, 削弱了高频用户对测试结果的影响。而按时间划分可能会使得测试集中高频用户占比很高, 模型更偏向高频用户; 训练集和验证、测试集的时间跨度可能交叉, 适用于不那么对时间敏感或是新商品不多的场景, 否则可能在训练阶段已知了最后时段新出现的商品或热点, 有泄露信息的风险。

其中, 对于后两种划分方式, 一般在预处理数据集时, 需要对交互记录按时间戳排序。值得注意的是, 尽量使用稳定排序保持数据集原有的交互顺序。因为由于日志记录问题, 对某个用户的多个交互历史, 可能出现时间戳相同的情况。时间戳相同时, 日志中记录的顺序可能由其他因素影响, 涉及展现位置、曝光顺序等信息, 保留日志原序更接近真实展现顺序。

综上所述, 以上几种常用数据集划分方式各有优劣, 应在具体任务和场景下选择合适的划分方式, 减少实验偏差。

3.3 本文数据集

本文主要使用的数据集有两个, 分别是 MovieLens 100k(下文缩写为 ML100k)和 RecSys Challenge 2017(下文缩写为 RecSys2017)数据集。其中 ML100k 是一个 MovieLens 系列中较早发布的经典数据集, 一共包含了一万个用户对电影的 1~5 级的评分, 还有少量用户和商品特征字段, 且较为稠密, 多年来被许多研究工作使用过。RecSys2017 是 2017 年在 RecSys Challenge 上发布的公开数据集, 是

招聘求职网站 Xing[①]整理发布的工作推荐数据集。该数据集数据量较大, 用户和商品都有大量特征字段, 且交互记录中包含展现但用户未交互的信息。两个数据集部分信息见表 1。

表 1 本文数据集信息

Table 1 Information of dataset

	ML100k	RecSys2017
用户数	944	419195
用户特征数	3	11
>3 评分数	55375	1586576
商品数	1683	153589
商品特征数	20	10
<4 评分数	44625	1793778

评分预测和点击预测的实验设定较为简单, 本文不作过多讨论, 而是关注于 Top-N 排序任务对设定。因此, 本文对每个数据集按用户序列使用留一法划分数据集。对于 ML100k, 保证训练集中有至少有五个正向交互, 对于 RecSys2017 保证训练集中至少有一个正向交互。对于 ML100k, 默认用户所有评分过的商品都属于正向交互, 没有已知的负向交互(显式负样本), 所有未交互记为负向(隐式负样本)。除此之外, 为了研究显式负样本的在训练和测试过程中的使用对模型效果的影响, 本文对 ML100k 数据集另作了转换, 将用户评分大于 3 的交互认为是用户显式表达喜欢, 属于正向交互, 用户评分小等于 3 的交互为认为是用户表达不喜欢, 为显式负样本, 其余为隐式负样本(记该转化后的数据集为 ML100k01)。划分后的数据集信息见表 2。

表 2 本文数据集划分

Table 2 Information of dataset division

	ML100k	ML100k01	RecSys2017
训练集正向	98114	53509	1566576
验证集正向	943	932	10000
测试集正向	943	934	10000
训练集负向	0	41238	1771198
验证集负向	0	1365	11107
测试集负向	0	2022	11473

本文实验任务为 Top-N 排序任务, 在验证和测试集中, 每个用户有一个正向交互, 与所有非正向交互(包括已知负向交互和未交互)够成候选商品集合。要求模型对候选集合中每个商品打分, 以测试正向交互在候选集合中的排序位置, 评价指标为 HR@K 和 NDCG@K。但对于有些数据集, 例如 RecSys2017, 商品数量过多, 全量商品评测代价太大, 因此对于 RecSys2017 默认从非正向中采样 1000 个构成候选集。已有工作验证采样 1000 个商品与在全量商品上评测, 对模型效果优劣的评价基本一致[8-9]。对于 ML100k 默认取全部非正向交互, 有不同处理会在对应实验中做具体说明。

4 模型训练

模型训练过程是指利用训练集的样本学习模型参数, 它决定了模型在使用过程中的实际预测效果。为探讨不同的实验设定对不同类型推荐模型的影响, 本文选择了 3 种不同的推荐模型:

(1) MostPopular: 推荐候选集合中最流行的商品, 非个性化推荐。最简单的推荐算法之一, 可以作为参考基准, 了解在某个设定下简单的推荐算法可以取得的效果。

(2) BPRMF[25]: 基于矩阵分解的经典协同过滤模型, 最简单的个性化推荐算法之一。模型为每个用户和商品都存储一个向量, 通过向量点乘表示用户对该商品的偏好得分。

(3) GRU4Rec[26]: 基于循环神经网络的序列推荐方法, 最简单的序列推荐深度模型之一。模型输入为一个用户的历史交互序列, 用循环神经网络建模后表示成用户偏好向量, 与商品向量点乘作为偏好得分。

这 3 种方法分别为非个性化推荐、协同过滤、序列推荐模型的典型代表, 且没有使用用户和商品属性特征。

4.1 训练方式

模型训练方式是指模型输入样本的方式以及计算损失函数的方法。不同的训练方式有不同的侧重, 适用于不同的任务。推荐场景中较常用的训练方式有两种: 单样本(point-wise)训练和配对样本(pair-wise)训练, 如图 2 所示。

4.1.1 单样本训练

这种训练方式与一般的回归、分类实验类似, 训练集由多个独立的样本组成, 每一轮训练包含的样本无变化, 训练目标是优化模型参数使得对每一个样本预测值都尽可能准确。该训练方法常用在评分预测或点击预测任务中。评分预测希望对每个单一样本的预测值都尽可能接近已知评分。点击预测希

① https://www.xing.com/

望模型对每个样本, 即用户商品对, 能够准确预测其是否点击。

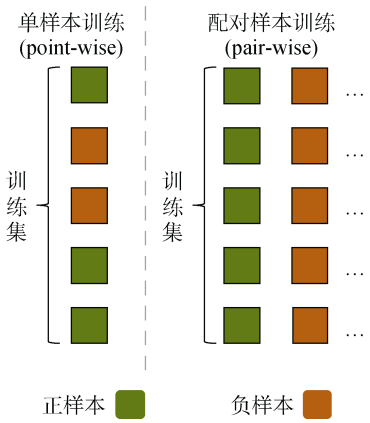

图 2 模型训练方式示意

Figure 2 Illustration of various ways of training

如果用 y_i 表示第 i 个样本的标签(比如 1~5 的评分, 或者表示点击与否的 0/1), 用 $\hat{y}_i$ 表示模型预测的分数, 评分预测中经常使用 y_i 和 $\hat{y}_i$ 间的均方根误差(Root Mean Square Error, RMSE)作为损失函数进行训练:

$$L_{RMSE}=\sqrt{\frac{1}{n}\sum_{i}^{n}(\hat{y}_i-y_i)^2}.$$

当标签 y_i 为 0/1 时(如是否点击), 除了 RMSE 外, 还经常采用二分类交叉熵(Binary Cross Entropy, BCE)损失函数进行训练:

$$L_{BCE}=-\frac{1}{n}\sum_{i}^{n}\left[y_i\log\hat{y}_i+(1-y_i)\log(1-\hat{y}_i)\right].$$

4.1.2 配对样本训练

这种训练方式对于每一个正样本, 会采样一定数目(通常采样一个, 也可以采样多个)的负样本与其匹配。训练中每一轮与正样本匹配的负样本可能不同。训练目标是优化模型参数使得在每一对(组)正负样本中, 正样本的预测值要尽可能大于负样本。由于该方法更加关心正样本与负样本之间预测值的大小关系, 因此常用在排序任务中。例如 Top-N 推荐任务中, 理想情况下需要正样本预测值比负样本大, 排在负样本前面即可, 而不在乎其预测值的绝对大小、是否接近 0 或 1, 正样本负样本的预测值都可以很大或很小。

配对样本训练最有代表性的是贝叶斯个性化排序(Bayesian Personalized Ranking, BPR)损失函数, 通过优化对于不同用户正样本得分高于负样本的似然概率来达到优化排序的目标:

$$L_{BPR}=-\frac{1}{n}\sum_{i}^{n}\log\sigma(\hat{y}_i-\hat{y}_i^-).$$

其中 $\hat{y}_i^-$ 表示模型对于负样本的预测分数。然而 BPR 损失函数只支持单个负样本的情况, 当每个正样本对应多个负样本时, 可以在 BPR 的基础上将每个正样本和多个负样本看做多组 pair, 计算其平均值(称为 BPR+):

$$L_{BPR+}=-\frac{1}{n}\sum_{i}^{n}\left(\frac{1}{k}\sum_{j}^{k}\log\sigma(\hat{y}_i-\hat{y}_j)\right).$$

其中 k 表示对每个正样本负采样的个数, j 索引了每个负样本, $\hat{y}_j$ 表示模型对负样本的预测分数。另一种方式是采用交叉熵(Cross Entropy, CE)损失函数, 使用 softmax 得到正样本和所有负样本的概率分布, 最大化正样本对应的概率:

$$L_{CE}=-\frac{1}{n}\sum_{i}^{n}\left(\frac{\exp(\hat{y}_i)}{\exp(\hat{y}_i)+\sum_{j}^{k}\exp(\hat{y}_j)}\right).$$

虽然 L_{BPR+} 和 L_{CE} 都能够处理多个负样本的情况, 但简单的负样本对梯度几乎没有影响, 反而因为对负样本个数做平均削弱了真正有价值负样本对梯度的贡献。因此有工作[27]提出 BPR-max 损失函数对不同负样本进行加权:

$$L_{BPR-\max}=-\frac{1}{n}\sum_{i}^{n}\sum_{j}^{k}s_j\log\sigma(\hat{y}_i-\hat{y}_j).$$

其中 $s_j=\exp(\hat{y}_j)/\sum_{l}^{k}\exp(\hat{y}_l)$ 是根据负样本预测得分进行 softmax 得到的权重, 得分越高(越难和正样本区分)的负样本权重越高。

实验证明随着负样本数量的增加, 引入权重的 $L_{BPR-\max}$ 显著优于 L_{BPR+} 和 L_{CE}。

综上所述, 在进行配对样本训练时, 一方面可以采用 L_{BPR}, 所有模型都只采样一个负样本进行对比; 另一方面如果采样多个负样本的话, 应当优先使用负样本加权的 $L_{BPR-\max}$。本文主要关注于 Top-N 推荐任务, 更关心排序, 因此使用配对样本训练方式。并且若非特殊说明, 训练时每个正样本配对采样一个负样本, 即采用 L_{BPR} 作为损失函数进行训练。

4.2 训练负样本

在推荐实验中, 负样本对于训练起着至关重要的作用。根据数据是否为纯隐式反馈, 负样本可以分为“显式负样本”和“隐式负样本”两种。对于纯隐式反馈的数据集, 只有用户的正向交互被记录下来, 未交互的样本可能是用户不感兴趣或者只是没有出现在用户的视野里, 这些样本统称为“隐式负样本”。有些数据集在此基础上还记录了商品是否呈现给用户的信息(impression), 明确知道哪些样本是用户接触到但没有交互的, 这些样本称为“显式负样本”。

一般来说, 推荐系统数据集中负样本的规模相比用户交互过的正样本来说要大很多, 因此通常的训练方式会对负样本进行采样。比如 pair-wise 的训练方式就会在每一轮训练中对每一个正样本随机采样一个负样本进行训练。当两种类型的负样本同时存在时, 如何对这些负样本进行采样就成为一个开放性的问题。此外, 即使只有隐式负样本, 也涉及在训练时负样本的采样范围是否应该包括验证测试集交互这样的细节问题。下面将通过实际数据集上的实验对显式负样本的使用以及隐式负样本的范围定义进行分析与讨论。

4.2.1 如何使用显式负样本

为探讨训练时采样显式负样本比例对训练结果的影响, 本小节在 ML100k01 和 RecSys2017 两个数据集上固定验证和测试集(两数据集验证测试样本正负比例均取 1:1000), 调整训练时每一轮采样显式负样本的比例(其余采样自训练集非正样本), 例如 0.1 表示以 0.1 的概率从用户的显式负样本中采样训练负样本。观察模型效果变化, 结果如图 3 所示。

首先, 在本文两个数据集上, 不同模型在训练时采样不同比例显式负样本时, 测试效果变化趋势基本一致, 总体上采样越多显式负样本效果越差。可能原因是测试和验证集负样本是从非正样本中采样得来, 大多来自隐式负样本, 训练时采样太多显式负样本会导致训练和测试的数据分布差异过大。

其次, 不同模型的变化曲线略有不同: 首先, MostPopular 只需要计算每个商品训练集中被正向交互的次数, 与负样本采样无关, 不受显式负样本使用的影响; 其次, BPRMF 在采样少量(例如 0.1 比例)的显式负样本, 会比完全不采样略优, 可能原因是对 BPRMF 来说, 显式负样本提供了更准确的矩阵元素信息, 有利于矩阵分解; 然而, 对于 GRU4Rec 序列推荐模型, 训练时采样越多的显式负样本, 模型测试效果会持续下降, 原因可能是 GRU4Rec 是基于序列推荐的模型, 其背后的学习方法和模型逻辑与 BPRMF 有较大差异, 受显式负样本的影响程度不同。值得注意的是, GRU4Rec 随着采样现实负样本比例的增大, 效果下降趋势较 BPRMF 更快。在采样比例为 0, 即完全从非正样本中采样时, GRU4Rec 明显优于 BPRMF, 但随着采样显式负样本比例增加, 模型优劣可能会发生反转, GRU4Rec 会差于 BPRMF, 甚至两个模型都会差于 MostPopular。这提醒大家在使用显式负样本时需要根据数据集和模型情况调整包括采样比例在内的使用方式。

综上所述, 本节的实验结果验证了训练时采样显式负样本对模型测试效果有很大影响, 需要在实验时引起注意。但该实验结果不一定适用所有数据集和所有模型, 具体需要根据实际情况考虑显式负样本的使用。

4.2.2 隐式负样本的范围定义

当数据中只有隐式反馈时, 许多论文会将训练负采样过程描述为从用户没有交互过的商品集合(隐式负样本)中采样。如果用 $\mathcal{I}$ 表示商品集合, S_u 表示用户交互的商品集合, 隐式负样本可以表示为商品集合和用户交互集合的差集 $\mathcal{I} \setminus S_u$。然而这里 S_u 的具体含义可能有歧义, 是只包含训练集中用户交互过的商品, 还是包含数据集中用户交互过的所有商品(包括验证测试集), 这直接影响了负采样的范围。形式化地, 我们用 S_u^a 表示用户在训练集交互过的商品集合, 用 S_u^b 表示用户在验证测试集交互过的商品集合, 那么有 $S_u = S_u^a \cup S_u^b$。此时两种负采样设定可以描述为:

- exclude: 表示隐式负样本不包括验证测试集中交互的商品, 即负样本采样范围是所有数据集中已知用户没有交互的样本 $\mathcal{I} \setminus S_u$;
- include: 表示采样范围包括验证测试集, 假设训练时用户对这些样本的交互未知, 也属于隐式负样本, 即 $\mathcal{I} \setminus S_u^a$。

从实验科学性的角度分析, 验证集和测试集在训练阶段对模型应当是透明的, 在 exclude 设定下其实一定程度上泄露了验证测试的信息。然而测试验证集交互的商品数量往往较少, 很难判断训练负采样阶段剔除这部分商品会对模型效果产生多大程度的影响。因此我们在两个真实数据集上进行了实验, 主要选取的模型选为矩阵分解模型 BPRMF 以及利用历史序列信息的 GRU4Rec(MostPopular 不存在训练过程, 没有负样本范围的影响, 但也汇报结果作为参考), 用 NDCG 和 HR 作为评价指标, 实验结果如表 3 所示。通过实验我们主要有以下观察:

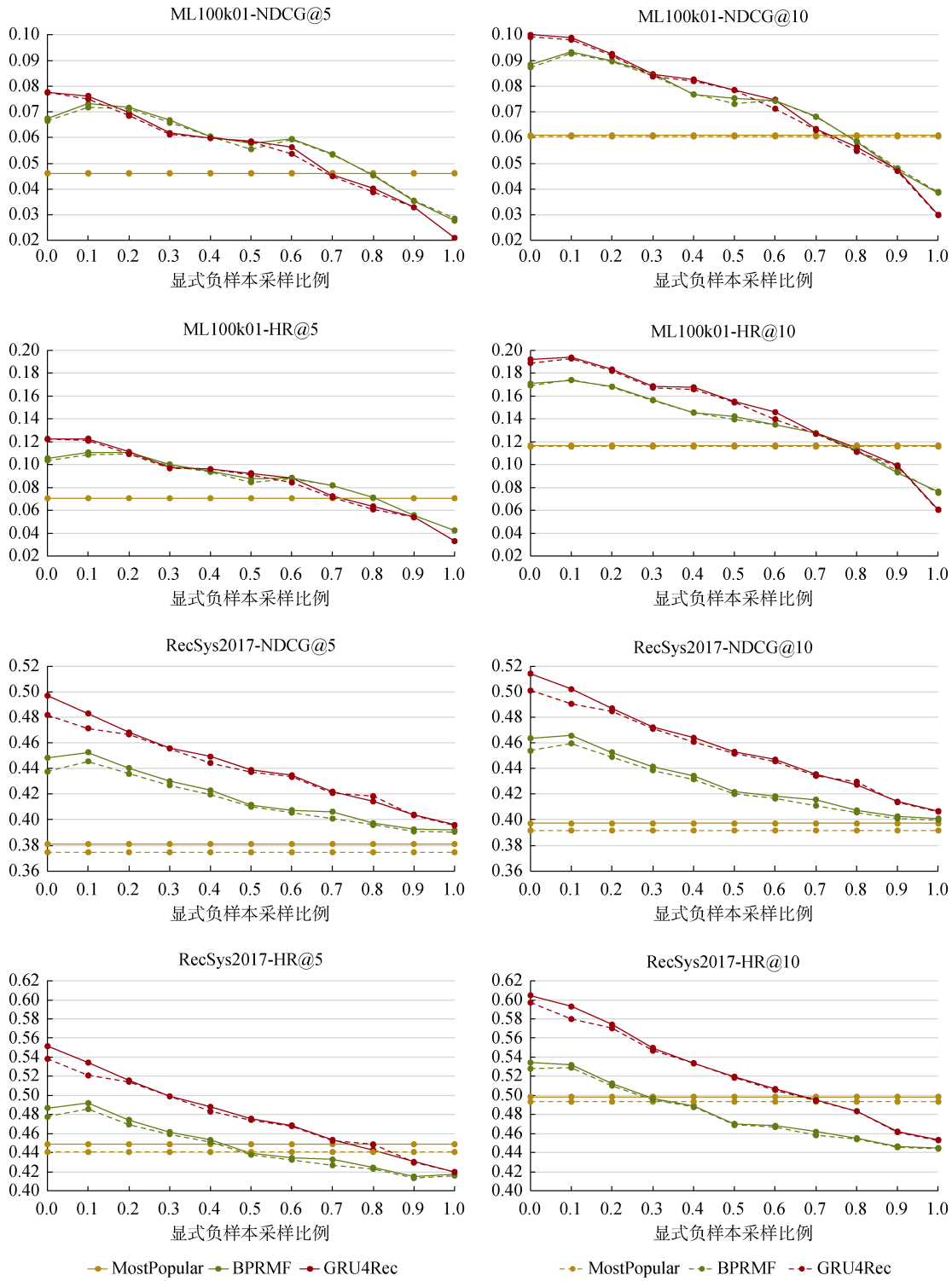

图 3　训练时采样显式负样本比例对测试结果影响(测试正负样本 1:1000)，实线为测试集负样本从全部负样本中随机采样时的结果，虚线为测试集负例优先来源自已知显式负样本

Figure 3　Effects of sampling different amounts of explicit negative training samples (test positive:negative samples is 1:1000). Solid lines are results when negative test samples are randomly sampled from all negative samples. Dash lines are results when negative test samples are firstly taken from known negative samples.

- 不同的隐式负样本范围确实会带来模型性能的变化，影响的程度与模型类别、数据集的规模和稠密程度都有关。
- 对于 ML100k 这种数据规模较小、用户交互相对稠密的数据集，负采样范围带来的影响极大，甚至改变了模型间的优劣顺序。

表 3 训练时隐式负样本范围定义实验

Table 3 Experiments of using different implicit negative training samples

		MostPopular		BPRMF		GRU4Rec	
		exclude	include	exclude	include	exclude	include
ML100k	NDCG@5	0.0364	0.0364	0.1008	0.0534	0.0547	0.0681
	HR@5	0.0583	0.0583	0.1389	0.0795	0.0891	0.1082
	NDCG@10	0.0450	0.0450	0.1212	0.0690	0.0748	0.0902
	HR@10	0.0859	0.0859	0.2015	0.1283	0.1516	0.1771
	NDCG@20	0.0553	0.0553	0.1436	0.0894	0.0945	0.1125
	HR@20	0.1273	0.1273	0.2906	0.2100	0.2312	0.2651
RecSys2017	NDCG@5	0.3811	0.3811	0.4576	0.4574	0.5003	0.5003
	HR@5	0.4488	0.4488	0.4978	0.4979	0.5543	0.5547
	NDCG@10	0.3972	0.3972	0.4708	0.4708	0.5168	0.5186
	HR@10	0.4987	0.4987	0.5387	0.5397	0.6055	0.6111
	NDCG@20	0.4117	0.4117	0.4834	0.4821	0.5311	0.5334
	HR@20	0.5564	0.5564	0.5886	0.5845	0.6620	0.6699

- 对于矩阵分解模型 BPRMF, exclude 设定下能取得更好的结果; 对于序列推荐模型 GRU4Rec, include 设定下效果更好。

ML100k 数据集本身商品数量较少, 且用户平均交互数量较多, 因此在 include 设定下有较大的可能性采样到验证测试集交互的样本; 而 RecSys2017 数据集由于其数据规模和稀疏性, 用户未交互的商品集合较大, 验证测试集交互的商品相对来说非常少, 采样到的可能性也较小。这解释了两种设定在两个数据集上影响的差异。

对于隐式负样本范围对两种模型的的不同影响, 我们认为和模型的特点有关。基于矩阵分解的 BPRMF 显式建模了用户向量, 因此如果测试集的商品作为负样本在训练时被采样到, 损失函数的梯度回传会使得用户向量朝远离测试集商品向量的方向更新, 具有更强的记忆能力, 在测试时遇到这个商品也会给出较低的分数。而 GRU4Rec 作为序列推荐模型, 参数集中在建模给定序列下的转移关系。当训练时采样到验证测试集中的商品时, 模型感知到的信息是某个给定序列下目标应当是用户实际交互的商品而不是验证测试集中的商品, 这对于建模序列转移关系有着正向的影响, 而这个信息在 exclude 设定下是无法被模型捕获的。总的来说, exclude 设定下一定程度上泄露了验证测试集的信息, BPRMF 对这种泄露更敏感, 记忆能力更强, 因此取得了更好的效果; 而 GRU4Rec 对这种泄露不敏感, 同时 exclude 设定限制了模型本能感知到的信息, 因此 GRU4Rec 在 include 设定下效果较好。

综上所述, 在进行模型训练时, 隐式负样本应当依照 include 设定, 采样负样本时包含验证测试集中交互的商品。这样不仅在实验科学性上更为合理, 保证了验证测试集在训练阶段对模型的透明, 在真实数据集的实验中这种设定下的模型优劣关系也更符合一般认知。

5 验证与测试

模型验证与测试是指在模型训练之后, 用训练集以外的数据测试其效果。其中验证集一般被用来调整模型超参数, 模型最终表现优劣以测试集结果为准。

5.1 验证测试负样本

在 Top-N 排序任务下, 验证测试时正样本需要和一系列负样本共同排序得到排序列表来计算评价指标, 这就涉及负样本如何选择的问题。总体来看, 一般分为“基于采样的评测”和“全量评测”两种方式。“基于采样的评测”会选取一定数量的负样本(如 100 个)和正样本一同排序; “全量评测”则把所有商品看作候选集合, 衡量正样本在所有商品中的排序情况。随着深度推荐系统的发展, 出于效率的考虑, 基于采样的评测成为研究者们的普遍选择。然而也有工作指出基于采样的评测可能会带来和全量评测不一致的结果, 应当使用哪种方式进行评测目前在推荐领域尚无定论。但不管是哪种设定, 都存在一些论文中不常提及的细节问题, 下面将分别对这两

种设定进行分析与讨论。

5.1.1 基于采样的评测

首先, 验证集和测试集中对每个正样本, 构成排序候选集合采样的负样本数目可以不同。早期的一些工作通常采样 100 个负样本[6], 然而后来一些学者通过实验和理论说明了基于采样的评价方式, 在采样数目较少时可能会出现与真实全量商品集合上评测结果产生较大偏差[8-9]。作为验证, 本节保持与 4.2.1 节相同的设定, 但是验证集和测试集正负样本比例修改为 1∶20, 实验结果见图 4。与图 3 对比, 可以观察到: 在评价指标上, 测试负样本数目减少后评价指标的绝对数值有了明显上升, 这是因为评价集合更简单了, 正例更容易被排到靠前的位置。并且, 在不考虑训练采样显式负样本时(显式负样本采样比例为 0), 测试正负样本比例 1∶20 时模型优劣顺序与 1∶1000 时有较大不同。在测试正负样本比例 1∶20 时, ML100k01 上 BPRMF 和 GRU4Rec 差别不明显, 甚至在 RecSys2017 上, MostPopular 结果要优于 BPRMF 甚至接近 GRU4Rec, 而 1∶1000 时 GRU4Rec、BPRMF 和 MostPopular 有明显的优劣顺序。综合下小节中全量评测的结果, 以及前人工作中的实验[8-9]可知, 测试正负样本比例 1∶1000 时的结果接近在全量商品集合上评测的结果, 模型优劣顺序更接近真实情况。因此验证集和测试集若采用基于采样的评测, 应至少采样 1000 个负样本。

其次, 如前文提到, 负样本有显式负样本和隐式负样本之分, 因此验证和测试中采样负样本时如何使用显式负样本对模型结果也有一定影响。图 3 和图 4 中实线为测试集负样本从全部负样本中随机采样时的结果, 虚线为测试集负例优先来源自已知显式负样本, 不够 1000 或 20 时再从隐式负样本中采样。对比对应的实线与虚线可知, 若验证和测试集合中优先来自已知显式负样本, 模型结果相比于全部随机采样自负样本一般更低, 其可能原因是显式负样本一般是数据平台已有推荐系统已推荐但用户不喜欢的商品, 已有推荐系统认为这些商品用户可能会喜欢, 说明这些样本与正样本更接近, 是更难的负样本。

最后, 对比测试正负样本比例 1∶1000 与 1∶20 时的结果可以发现, 在测试负样本采样总数较少时, 加入已知显式负样本对测试结果影响更大, 评价指标下降更显著。因为显式负样本在测试集合全部负样本中的占比更高, 因此影响更大。在测试正负样本比例 1∶1000 时加入显式负样本影响较小, 但最终评价指标更接近全量评测时的结果。

综上所述, 在采取基于采样的评测时, 对每个正样本应至少匹配采样 1000 个负样本以够成候选集。对于显式负样本可以根据需要考虑是否优先加入候选集进行评测。综合 4.2.1 节的结果, 在本节涉及的数据集中, 训练显式负样本采样比例在 0~0.1 之间调整比较合适, 但如果测试显式负样本占比更高时也可能需要训练时采样更多显式负样本。

5.1.2 全量评测

近年来, 有越来越多的工作诉诸全量评测来避免采样带来的偏置。然而看似简单直接的全量评测也存在容易忽略的细节设定, 具体来说, 我们归纳出 3 种全量评测的实现方法:

- all: 直接对所有商品进行排序, 用户交互过的商品也被当做负样本。
- all\inter: 排序时排除用户交互过的商品, 即只将用户没有交互过的商品作为负样本。
- all\inter+: 在 all\inter 的基础上更严谨地对验证集进行特除处理, 认为在验证集上进行评测时测试集未知, 将测试集中以及没有交互过的商品作为负样本, 对于测试集的评测和 all\inter 等同。

all 和 all\inter 两种方法的本质区别在于是否将已知用户交互过的商品从候选集合中“隐掉”, all\inter+则对“交互过的商品”有更细致的定义。现在采用全量评测的工作往往并不会详细说明具体采取的哪种设定, 即使都在同一个数据集上进行全量评测的两个工作有时也无法对比结果。为了说明不同方法之间的效果差异, 我们在真实数据集上进行了对比实验。由于 RecSys2017 商品数量太大全量评测较为困难, 我们在 ML100K 上进行实验, 模型除了 BPRMF 和 GRU4Rec 外还包括了不需要参数训练的 MostPopular, 来观察评测设定本身带来的影响。实验结果如表 4 所示。

通过实验我们主要有以下观察:

- all 设定下的排序任务更难, 各个模型得到的评价指标数值显著低于排除用户交互商品的其他两个方法。
- all\inter+只影响验证集上的模型选取, 相比 all\inter 基本变化不大。
- 模型间的相对提升甚至优劣关系在 all 和 all\inter 设定下会出现不一致(如 MostPopular 在 all 设定下优于 BPRMF, 而在 all\inter 下 BPRMF 效果更好; all 设定下 GRU4Rec 相比 BPRMF 的提升比 all\inter 设定下更大)。

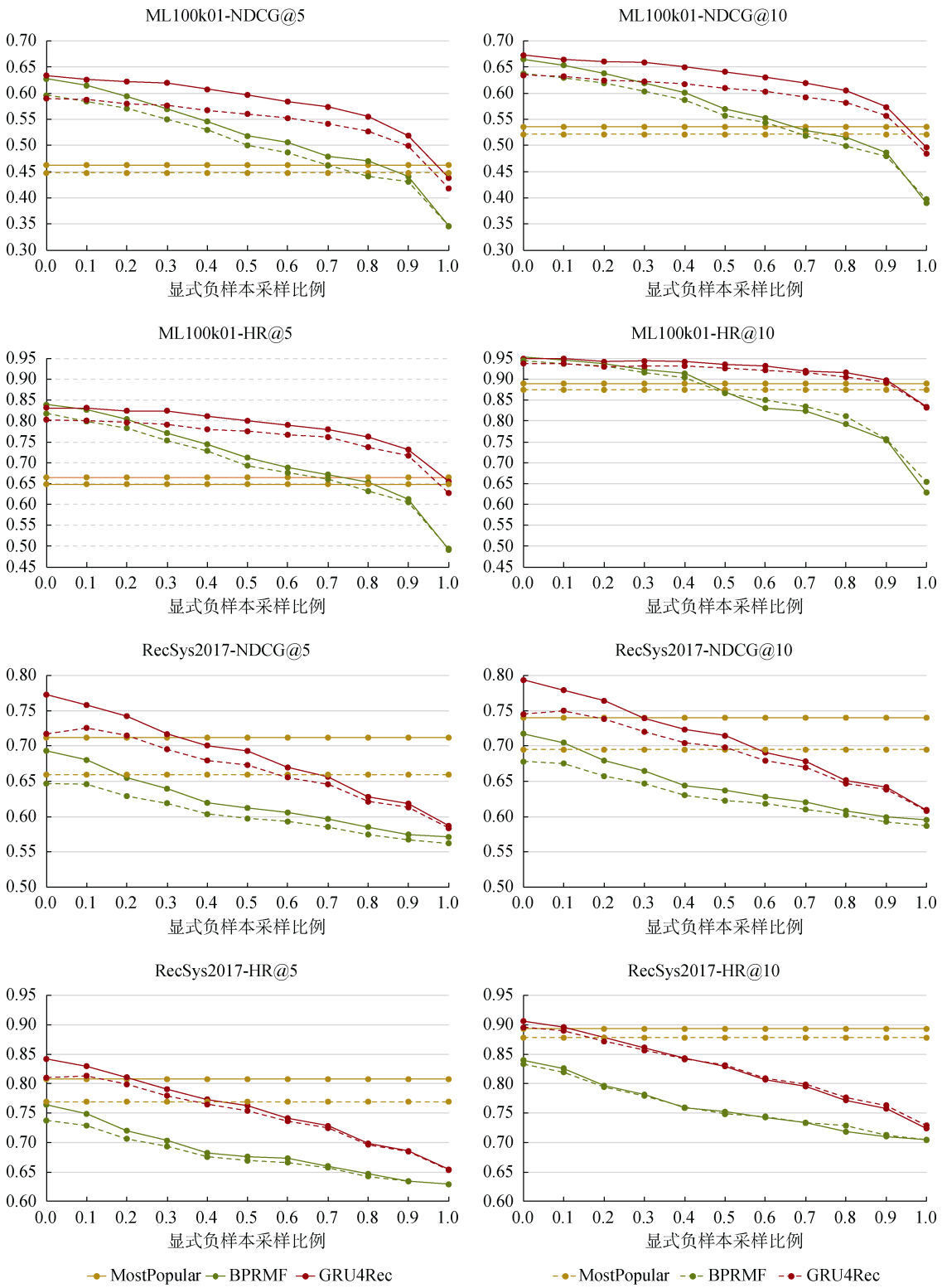

图 4　训练时采样显式负样本比例对测试结果影响(测试正负样本 1∶20)。实线为测试集负样本从全部负样本中随机采样时的结果, 虚线为测试集负例优先来源自已知显式负样本

Figure 4　Effects of sampling different amounts of explicit negative training samples (test positive:negative samples is 1∶20). Solid lines are results when negative test samples are randomly sampled from all negative samples. Dash lines are results when negative test samples are firstly taken from known negative samples.

表 4 全量评测不同方法对比

Table 4 Comparison of evaluation methods on all items

		ML100k			
		NDCG@5	HR@5	NDCG@10	HR@10
MostPopular	all	0.0150	0.0255	0.0230	0.0498
	all\inter	0.0364	0.0583	0.0450	0.0859
	all\inter+	0.0364	0.0583	0.0450	0.0859
BPRMF	all	0.0131	0.0223	0.0200	0.0435
	all\inter	0.0534	0.0795	0.0690	0.1283
	all\inter+	0.0494	0.0742	0.0679	0.1315
GRU4Rec	all	0.0334	0.0530	0.0467	0.0944
	all\inter	0.0681	0.1082	0.0902	0.1771
	all\inter+	0.0681	0.1082	0.0902	0.1771

虽然 all 设定下评测实现最为简单, 但将用户交互过的商品也作为负样本一同参与排序并不符合实验科学性, 从实验结果来看也会导致一些不符合一般认识的结果(比如 MostPopular 优于 BPRMF)。另一方面 all\inter+是最符合实验设计原则的, 严格界定了每个评测阶段模型所能感知到的信息, 因此建议默认采取 all\inter+的全量评测方法。对于 all\inter, 其主要区别在于训练过程中的模型选取, 从实验结果来看相比 all\inter+在多数情况下没有变化, 因此考虑到实现的便利性也是一个可行的选择。

5.2 超参数调整

一般的推荐模型都涉及若干可供调整的超参数, 比如学习率、正则化权重等。根据机器学习的基本实验准则, 超参数的调整应当在验证集上进行, 用验证集上表现最好的超参数设置在测试集进行测试。然而很多研究者为了突出所提出模型的提升, 会直接选取测试集上表现最好的超参数进行汇报, 许多推荐系统框架也会在训练时默认每轮输出测试集上的结果。此外, 有些研究者对于基线模型并没有进行认真的调参, 而是直接用原论文中的默认参数跑一个结果, 这很大程度上阻碍了推荐领域的健康发展, 论文中汇报的提升可能只是“cherry-picking”的结果。因此在实验中应当对每一个基线模型进行细致的调参, 给出超参数搜索的空间, 并且确保超参数的选择是根据验证集的结果确定的。另外需要注意的是, 论文中参数分析的部分建议也采用验证集上的结果进行汇报, 这样能展现实际的参数调整过程, 确认最终的超参数是在验证集上调整的结果。

6 评价指标

对于不同的任务场景, 需要使用不同的评价指标来衡量模型表现好坏。

对于评分预测任务, 常用的评价指标有 MAE、MSE、RMSE 等。其中 MAE 是平均绝对误差, MSE 是均方误差。相比于 MAE, MSE 通过平方放大了差异, 更关注预测值差异大的样本。RMSE 是均方根误差, 即 MSE 的平方根。

对于点击预测任务, 除了上述指标外, 还可以使用一些二分类的指标如精准度(Precision)、召回率(Recall), 以及它们的调和平均(F1)。还有一个更鲁棒更常用的点击预测评价指标是 AUC(Area Under ROC Curve, ROC 曲线下的面积)[28], 它主要衡量了模型将正样本(点击)排在负样本(非点击)前的能力, 在工业界点击预测模型中被广泛使用, 因为模型线下实验的 AUC 高低和线上表现优劣较为一致。但需要注意的是, AUC 这样的评价指标需要在完整测试集合上预测完后, 在全部样本上计算, 而不可以分批次计算 AUC 之后再取平均。这样会造成与实际全局的 AUC 有较大偏差, 有少数论文公开的代码中采用了这样的计算方式, 实际是不规范的。

而排序任务的评价指标计算更为复杂, 例如命中率 HR@K 计算的是模型对候选集排序后的前 K 个样本中是否有正样本。精准率 Precision@K 计算的是模型排序后前 K 个样本中正样本的比例。召回率 Recall@K 计算的是前 K 个样本中正样本数目占候选集中全部正样本的比例。更为常用和鲁棒的一个排序评价指标是归一化折损累计增益 NDCG@K (Normalized Discounted Cumulative Gain)[29]。它假设排序列表每个位置的增益依次是有折损的, 计算的是模型对候选集合的 Top-N 排序列表与理想情况下的 Top-N 排序列表的差异, 指标越高则模型排序越接近理想排序。

虽然对单个排序列表计算评价指标的过程比较明确, 但往往验证和测试集合中包含多个用户的多

个列表。模型对不同列表的排序效果可能不同, 通常大家会对每个列表计算出的指标求平均, 但求平均的方式也会对模型评价带来影响。例如图 5 中的测试集有3个用户共6个列表, 如果计算命中率HR@1, 对所有列表取平均的话指标 HR@1 为 0.5。但是可以发现用户 1 在测试集中有 3 个列表, 他贡献了一半的测试列表。他可能是个较高频的用户因此模型对他的偏好预测较准确, 而对其他更低频的用户偏好预测实际没有那么准确。如果按用户先对每个用户的列表计算出的 HR@1 取平均, 再对所有用户的 HR@1 取平均得到的值是 0.3889, 远低于 0.5。因此, 如果有的模型更偏重于高频用户但冷启动用户预测的非常差, 而有的模型冷启动用户预测非常准确, 在两种计算方式下可能会得出截然相反的模型效果优劣评价。

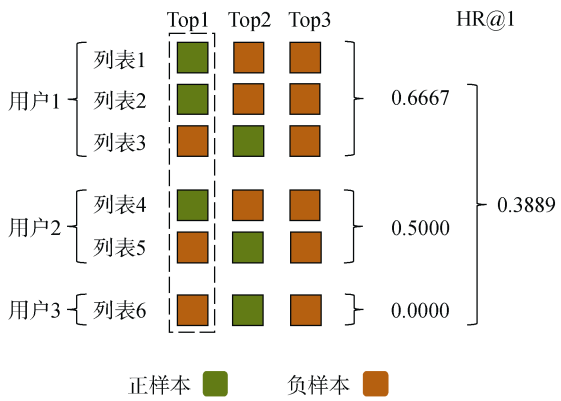

图 5　评价指标按用户平均示例

Figure 5　Illustration of average metrics by each user

综上所述, 在选取评价指标时, 需要根据具体任务、模型, 以及需要关注的问题(例如是否关注冷启动用户), 选择合适的评价指标。并且可以选择多个评价指标, 采取分用户、分列表多种计算方式, 更全面地评价模型效果。

7　总结

通过上述讨论和实验, 本文系统地讨论与反思了推荐系统实验中的细节设定, 包括数据集处理、模型训练、验证与测试、效果评价等多个角度。本节总结文中提到的可选、建议、必须的 3 类细节设定, 帮助个性化推荐算法研究者更科学合理地设计实验。

必要设定:

- 训练时隐式负样本的采样范围应当包括验证和测试集交互的商品, 否则有泄露信息风险, 会造成模型测试结果偏高;
- 尽量采用全量评测, 若规模不允许, 应尽可能多地采样负样本。结合前人工作[8-9], 经验上正负样本比例至少 1∶1000, 否则测试结果可能与全量评测实际结果有较大偏差;
- 全量评测时, 用户交互过的商品不应当参与排序(更严谨地, 测试时去除用户交互过的所有商品, 验证时只去除训练验证集中该用户交互的商品), 否则会造成模型测试结果偏低;
- 对基线模型在每个使用的数据集上进行超参数搜索, 汇报搜索空间, 否则会造成基线模型测试结果偏低;
- 依据验证集的结果进行确定超参数设置, 否则会造成过拟合验证集;
- 计算 AUC 等评价指标时应在全部样本上计算, 不可按批次计算后取平均, 否则和实际值差异较大;

建议设定:

- 数据预处理时对于时间戳相同的记录保持原数据中的顺序;
- 配对样本训练时如果采样多个负样本采用 $L_{BPR-\max}$ 对负样本进行加权;
- 参数敏感性分析中汇报验证集上的结果。

可选设定:

- 选择合适的数据集划分方式, 如随机划分、按时间划分、留一划分等;
- 选择合适的模型训练方式和损失函数, 如单样本或配对样本训练, BPR 或交叉熵;
- 训练测试时应考虑合适的显式和隐式负样本使用方式, 可适当调整显式负样本比例;
- 选择合适的评价指标和计算方式衡量模型效果, 例如排序任务中根据需要选择多个列表计算出指标的平均方式。

参考文献

[1] Resnick P, Varian H R. Recommender Systems[J]. *Communications of the ACM*, 1997, 40(3): 56-58.

[2] Dacrema M F, Cremonesi P, Jannach D. Are we Really Making much Progress? a Worrying Analysis of Recent Neural Recommendation Approaches[C]. *The 13th ACM Conference on Recommender Systems*, 2019: 101-109.

[3] Rendle S, Zhang L, Koren Y. On the Difficulty of Evaluating Baselines: A Study on Recommender Systems[EB/OL]. 2019.

[4] Steck H. Evaluation of Recommendations: Rating-Prediction and Ranking[C]. *The 7th ACM conference on Recommender systems*, 2013: 213-220.

[5] Cañamares R, Castells P, Moffat A. Offline Evaluation Options for

Recommender Systems[J]. *Information Retrieval Journal*, 2020, 23(4): 387-410.

[6] He X N, Liao L Z, Zhang H W, et al. Neural Collaborative Filtering[C]. *The 26th International Conference on World Wide Web*, 2017: 173-182.

[7] Chen J Y, Zhang H W, He X N, et al. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention[C]. *The 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017: 335-344.

[8] Krichene W, Rendle S. On Sampled Metrics for Item Recommendation[C]. *The 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020: 1748-1757.

[9] Li D, Jin R M, Gao J, et al. On Sampling Top-K Recommendation Evaluation[C]. *The 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020: 2114-2124.

[10] Rendle S. Factorization Machines with libFM[J]. *ACM Transactions on Intelligent Systems and Technology*, 2012, 3(3): 1-22.

[11] Guo G, Zhang J, Sun Z, et al. Librec: A Java Library for Recommender Systems[C]. *CEUR Workshop*, 2015: 1388.

[12] Yang L Q, Bagdasaryan E, Gruenstein J, et al. OpenRec: A Modular Framework for Extensible and Adaptable Recommendation Algorithms[C]. *The Eleventh ACM International Conference on Web Search and Data Mining*, 2018: 664-672.

[13] Zhang S, Tay Y, Yao L N, et al. DeepRec: An Open-Source Toolkit for Deep Learning Based Recommendation[C]. *The Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019: 6581-6583.

[14] Zhao W X, Mu S L, Hou Y P, et al. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms[EB/OL]. 2020: arXiv: 2011.01731[cs.IR]. https://arxiv.org/abs/2011.01731.

[15] Wang C Y, Zhang M, Ma W Z, et al. Make it a Chorus: Knowledge- and Time-Aware Item Modeling for Sequential Recommendation[C]. *The 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020: 109-118.

[16] Harper F M, Konstan J A. The MovieLens Datasets[J]. *ACM Transactions on Interactive Intelligent Systems*, 2016, 5(4): 1-19.

[17] He R N, McAuley J. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering[C]. *The 25th International Conference on World Wide Web*, 2016: 507-517.

[18] McAuley J, Targett C, Shi Q F, et al. Image-Based Recommendations on Styles and Substitutes[C]. *The 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015: 43-52.

[19] Zhang Y F, Lai G K, Zhang M, et al. Explicit Factor Models for Explainable Recommendation Based on Phrase-Level Sentiment Analysis[C]. *The 37th international ACM SIGIR conference on Research & development in information retrieval*, 2014: 83-92.

[20] McAuley J, Leskovec J. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text[C]. *The 7th ACM conference on Recommender systems*, 2013: 165-172.

[21] Shi S Y, Zhang M, Liu Y Q, et al. Attention-Based Adaptive Model to Unify Warm and Cold Starts Recommendation[C]. *The 27th ACM International Conference on Information and Knowledge Management*, 2018: 127-136.

[22] Hu L M, Li C, Shi C, et al. Graph Neural News Recommendation with Long-Term and Short-Term Interest Modeling[J]. *Information Processing & Management*, 2020, 57(2): 102142.

[23] Wu C H, Wu F Z, An M X, et al. NPA: Neural News Recommendation with Personalized Attention[C]. *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019: 2576-2584.

[24] Shi S Y, Chen H X, Ma W Z, et al. Neural Logic Reasoning[C]. *The 29th ACM International Conference on Information & Knowledge Management*, 2020: 1365-1374.

[25] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian Personalized Ranking from Implicit Feedback [C]. *The 25th Conference on Uncertainty in Artificial Intelligence*, 2009: 452-461.

[26] Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-Based Recommendations with Recurrent Neural Networks[EB/OL]. 2015

[27] Hidasi B, Karatzoglou A. Recurrent Neural Networks with Top-k Gains for Session-Based Recommendations[C]. *The 27th ACM International Conference on Information and Knowledge Management*, 2018: 843-852.

[28] Bradley A P. The Use of the Area under the ROC Curve In the Evaluation of Machine Learning Algorithms[J]. *Pattern Recognition*, 1997, 30(7): 1145-1159.

[29] Järvelin K, Kekäläinen J. Cumulated Gain-Based Evaluation of IR Techniques[J]. *ACM Transactions on Information Systems*, 2002, 20(4): 422-446.

施韶韵 于 2017 年在清华大学计算机科学与技术系获得学士学位。现在清华大学计算机科学与技术系攻读博学位。研究领域为人工智能、信息检索。研究兴趣包括: 推荐系统、深度学习。Email: shisy17@mails.tsinghua.edu.cn

王晨阳 于 2018 年在清华大学计算机科学与技术系获得学士学位。现在清华大学计算机科学与技术系攻读博学位。研究领域为人工智能、信息检索。研究兴趣包括: 序列推荐、对比学习。Email: wangcy18@mails.tsinghua.edu.cn

马为之 于 2019 年在清华大学计算机科学与技术系获得博士学位。现任清华大学计算机科学与技术系智能技术与系统国家重点实验室信息检索课题组博士后研究员。研究领域为人工智能、信息检索。研究兴趣包括: 用户建模、跨领域推荐。Email: mawz@mail.tsinghua.edu.cn

张敏 于 2003 年在清华大学计算机科学与技术系获得博士学位。现任清华大学计算机科学与技术系智能技术与系统国家重点实验室信息检索课题组特别研究员。研究领域为人工智能、信息检索。研究兴趣包括: 用户建模、个性化推荐。Email: z-m@tsinghua.edu.cn

刘奕群 于 2007 年在清华大学计算机科学与技术系获得博士学位。现任清华大学计算机科学与技术系智能技术与系统国家重点实验室信息检索课题组长聘教授。研究领域为人工智能、信息检索。研究兴趣包括: 用户行为分析、网络搜索。Email: yiqunliu@tsinghua.edu.cn

马少平 于 1997 年在清华大学计算机科学与技术系获得博士学位。现任清华大学计算机科学与技术系智能技术与系统国家重点实验室信息检索课题组长聘教授。研究领域为人工智能、信息检索。研究兴趣包括: 网络搜索、推荐系统、智能信息处理。Email: msp@tsinghua.edu.cn