计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202330017
61(12): 3134−3153, 2024

一种跨区域跨评分协同过滤推荐算法

于 旭[1,2,3] 彭庆龙[1] 詹定佳[1] 杜军威[1] 刘金环[1] 林俊宇[4] 巩敦卫[1,5] 张子迎[6] 于 婕[1]

[1](青岛科技大学信息科学技术学院 山东青岛 266061)

[2](中国石油大学(华东)计算机科学与技术学院 山东青岛 266580)

[3](符号计算与知识工程教育部重点实验室(吉林大学) 长春 130012)

[4](中国科学院信息工程研究所 北京 100093)

[5](中国矿业大学信息与控制工程学院 江苏徐州 221116)

[6](嘉应学院计算机学院 广东梅州 514011)

(yuxu0532@upc.edu.cn)

A Cross-Region and Cross-Rating Collaborative Filtering Recommendation Algorithm

Yu Xu[1,2,3], Peng Qinglong[1], Zhan Dingjia[1], Du Junwei[1], Liu Jinhuan[1], Lin Junyu[4], Gong Dunwei[1,5], Zhang Ziying[6], and Yu Jie[1]

[1](*College of Information Science and Technology*, *Qingdao University of Science and Technology*, *Qingdao*, *Shandong* 266061)

[2](*College of Computer Science and Technology*, *China University of Petroleum* (*East China*), *Qingdao*, *Shandong* 266580)

[3](*Key Laboratory of Symbol Computation and Knowledge Engineering* (*Jilin University*), *Ministry of Education*, *Changchun* 130012)

[4](*Institute of Information Engineering*, *Chinese Academy of Sciences*, *Beijing* 100093)

[5](*School of Information and Control Engineering*, *China University of Mining and Technology*, *Xuzhou*, *Jiangsu* 221116)

[6](*College of Computer Science*, *Jiaying University*, *Meizhou*, *Guangdong* 514011)

Abstract Traditional cross-rating collaborative filtering paradigm ignores the influence of rating density in the target domain on the accuracy of user and item latent vectors, resulting in less accurate rating prediction in regions with sparse ratings. To overcome the influence of regional rating density on rating prediction, based on the thought of transfer learning, a cross-region and cross-rating collaborative filtering recommendation algorithm (CRCRCF) is proposed. Compared with the traditional cross-rating collaborative filtering paradigm, CRCRCF algorithm can effectively exploit not only the important knowledge from the auxiliary domain, but also the important knowledge from the rating-dense regions in the target domain, which can further improve the rating prediction accuracy of the whole target domain, especially the rating-sparse regions. Firstly, for users and items, active users and inactive users, popular items and unpopular items are divided respectively. Graph convolution matrix complementation algorithm is used to extract the latent vectors of active users and popular items in the target domain and all users and items in the auxiliary domain. Secondly, for users and items in rating-dense regions, deep regression models based on self-taught learning are constructed to learn the mapping relationships between latent vectors in the target domain and in the auxiliary domain, respectively. Then the mapping relationships are generalized to the whole target domain, and the relatively accurate latent vectors of inactive users and unpopular items in the auxiliary domain are used to derive their

收稿日期: 2023-04-10; **修回日期**: 2023-12-22

基金项目: 国家自然科学基金项目(62472441, 62172249, 61773384, 62202253); 山东省自然科学基金项目(ZR2021MF092, ZR2019MF014, ZR2021QF074); 中央高校基本科研业务费专项资金(93K172022K01)

This work was supported by the National Natural Science Foundation of China (62472441, 62172249, 61773384, 62202253), the Natural Science Foundation of Shandong Province (ZR2021MF092, ZR2019MF014, ZR2021QF074), and the Fundamental Research Funds for the Central Universities (93K172022K01).

通信作者: 林俊宇(linjunyu@fudan.edu.cn)

latent vectors in the target domain, which achieves the cross-region mapping relationships transfer and cross-rating latent vector information transfer successively. Finally, the restricted graph convolutional matrix completion model is proposed with the obtained latent vectors of inactive users and non-popular items in the target domain as constraints, and the corresponding recommendation results are given. The simulation experiments on MovieLens and Netflix datasets show that the CRCRCF algorithm has obvious advantages over other state-of-the-art algorithms.

Key words collaborative filtering; cross-region and cross-rating recommendation; graph convolution matrix complementation; self-taught learning; deep regression network; restricted graph convolutional matrix completion

摘 要 传统跨评分协同过滤范式忽视了目标域中评分密度对用户和项目隐向量精度的影响，导致评分稀疏区域评分预测不够准确. 为克服区域评分密度对评分预测的影响，基于迁移学习思想提出一种跨区域跨评分协同过滤推荐算法（cross-rating collaborative filtering recommendation algorithm, CRCRCF），相对于传统跨评分协同过滤范式，该算法不仅能有效挖掘辅助域重要知识，而且可以挖掘目标域中评分密集区域的重要知识，进一步提升目标域整体，尤其是评分稀疏区域的评分预测精度. 首先，针对用户和项目，分别进行活跃用户和非活跃用户、热门项目和非热门项目的划分. 利用图卷积矩阵补全算法提取目标域活跃用户和热门项目、辅助域中全体用户和项目的隐向量. 其次，对活跃用户和热门项目分别构建基于自教学习的深度回归网络学习目标域和辅助域中隐向量的映射关系. 然后，将映射关系泛化到全局，利用非活跃用户和非热门项目在辅助域上相对较准确的隐向量推导其目标域上的隐向量，依次实现了跨区域映射关系迁移和跨评分的隐向量信息迁移. 最后，以求得的非活跃用户和非热门项目在目标域上的隐向量为约束，提出受限图卷积矩阵补全模型，并给出相应推荐结果. 在 MovieLens 和 Netflix 数据集上的仿真实验显示 CRCRCF 算法较其他最先进算法具有明显优势.

关键词 协同过滤；跨区域跨评分推荐；图卷积矩阵补全；自教学习；深度回归网络；受限图卷积矩阵补全

中图法分类号 TP391

传统协同过滤推荐算法[1-2]作为大数据时代解决信息过载问题的重要手段，在餐厅推荐[3]、景点推荐[4]、广告推荐[5]等领域得到了广泛的应用. 协同过滤算法的主要思想是基于用户反馈学习用户偏好，可以为用户提供个性化服务，提升用户满意度与平台商业收入. 然而当用户反馈数据非常稀疏时，协同过滤算法往往不能有效捕捉用户的偏好，数据稀疏性将导致推荐算法产生严重过拟合，影响推荐算法的性能.

针对存在 2 种评分格式的推荐场景，相对于 5 分制数值评分，用户更倾向于进行简单的 1, 0 二元（喜欢和不喜欢）评分，因此除目标域稀疏的 5 分制数值评分外，该推荐场景往往含有辅助域上相对较丰富的二元评分. 此处，目标域指的是运行推荐算法的数据集，往往可用于分析的评分数据比较匮乏. 辅助域指的是与目标域有一定关联的其他数据集，相对于目标域通常具有更为丰富的评分数据. 在迁移学习框架下，辅助域也被称为源域（source domain）. 针对该类推荐场景，近年来部分研究者提出了一种跨评分的推荐方法[6-10]，通过迁移辅助域上挖掘到的有价值的知识到目标域，提升目标域上的推荐性能. 为与其他推荐模式相区分，本文称之为跨评分协同过滤范式，定义如下.

定义 1. 跨评分协同过滤范式. 假设辅助域包含较丰富的 0, 1 评分，目标域包含稀疏的 1~5 评分，跨评分协同过滤范式是指将辅助域信息迁移到目标域来提升目标域推荐性能的推荐模式.

针对跨评分协同过滤范式，Pan 等人[6]提出基于联合分解的迁移模型（TCF），该模型假定目标域和辅助域共享同一用户隐特征空间和项目隐特征空间，但具有不同的簇级评分模式，通过联合分解的方式实现重要知识由辅助域向目标域的迁移. 在 TCF 基础上，针对数据异构造成用户隐向量的变化，Pan 等人[7]提出基于联合分解的交互丰富迁移模型（iTCF），通过共享预测性能来实现辅助域和目标域用户隐特征的交互. iTCF 是基于联合方式的迁移学习算法，该算法包含 2 个松耦合的矩阵分解任务. 为更充分地进行知识迁移，Pan 等人[8]进一步提出基于混合分解的迁移模型（TMF），该模型结合辅助域中数据提取用户的喜欢偏好和不喜欢偏好，并合并 2 种偏好辅助目标域矩阵分解，这种方式将原本 2 个松耦合的矩阵分解之间的联系变得更加密切，提高了模型的推

荐性能. Zhang 等人[9] 和 Jiang 等人[10] 均对三因子矩阵分解中的簇级评分模式进一步分解, 捕获不同域的共享评分模式, 同时分离领域特有的评分模式. 特别地, Zhang 等人[9] 提出用于协同过滤的基于多源异构反馈的增强知识转移模型(EKT), 通过提取目标域和辅助域中用户和项目的几何结构图信息, 来改善2个域的隐向量的学习效果. Jiang 等人[10] 则提出深度低秩稀疏联合分解模型(DLSCF), 该模型考虑项目类型的树状层次结构, 以多层三因子矩阵分解的方式建模项目的隐式层次结构.

然而, 以往跨评分协同过滤范式往往假设推荐系统中的所有区域数值评分均较为稀疏, 对不同区域采取一致的评分预测策略. 实际上, 尽管目标域整体数值评分较为稀疏, 但是仍然有部分活跃用户在部分热门项目上评分比较密集. 例如, 在 Movielens 数据集上, 我们可以容易地找出一个 100 活跃用户和 200 热门影片组成的评分子集, 该子集具有相对较高的评分密度. 以往跨评分协同过滤范式在知识迁移的过程中忽视了评分密度对用户和项目隐向量求解精度的影响, 导致评分稀疏区域评分预测不够准确. 考虑到活跃用户和热门项目所在区域上的 2 种评分均较为密集, 从该区域相关评分数据中挖掘有价值的知识, 并迁移到评分非密集区域可以有效提升模型对目标域整体的评分预测性能, 尤其是针对评分非密集区域的评分预测性能. 因此, 基于迁移学习思想, 本文提出一种跨区域跨评分协同过滤推荐算法(a cross-region and cross-rating collaborative filtering recommendation algorithm, CRCRCF). 首先, 基于目标域评分个数将全体用户划分为活跃用户和非活跃用户, 将全部项目划分为热门项目和非热门项目. 为了更好地进行用户和项目表征, 利用图卷积矩阵补全(graph convolutional matrix completion, GC-MC)算法[11] 提取目标域活跃用户和热门项目, 以及辅助域中全体用户和项目的隐向量. 由于目标域活跃用户和热门项目对应的评分密集区域以及辅助域中评分均较为丰富, 所提取的隐向量相对较为准确. 其次, 针对活跃用户和热门项目, 构建基于自教学习(self-taught learning)[12] 的深度回归网络分别学习目标域和辅助域上 2 种评分对应的用户隐向量和项目隐向量的映射关系. 然后, 将活跃用户和热门项目的隐向量的映射关系泛化到目标域非活跃用户和非热门项目上, 利用非活跃用户和非热门项目在辅助域上相对准确的隐向量推导其在目标域上的隐向量, 依次实现了跨区域映射关系迁移和跨评分的隐向量信息迁移. 最后, 以求得的非活跃用户和非热门项目在目标域上的隐向量为约束, 提出受限图卷积矩阵补全模型, 并给出相应推荐结果.

为了避免本文提出的跨区域推荐与传统跨域推荐混淆, 将二者定义为:

定义 2. 跨区域推荐. 跨区域推荐是指将活跃用户和热门项目对应的评分密集区域的信息迁移到非密集区域, 来提升评分非密集区域推荐性能的推荐模式.

定义 3. 跨域推荐. 假设辅助域与目标域物品种类不一致, 跨域推荐是指将评分数据较丰富的辅助域信息迁移到评分数据较匮乏的目标域来提升目标域推荐性能的推荐模式.

本文所提出的“跨区域”是指在不同评分密度的区域之间实现知识迁移, 是不同于传统跨域推荐的新范式. 在 MovieLens10M 和 Netflix 数据集上的广泛对比实验验证了本文算法在 4 种不同测试指标上较其他多种最先进的对比方法具有明显优势.

本文的主要贡献有 3 点:

1)提出跨区域推荐范式, 可以进行细粒度的精准推荐;

2)提出基于自教学习的深度回归网络学习活跃用户和热门项目在目标域和辅助域上对应的隐向量的映射关系, 可充分利用非活跃用户和非热门项目相关的大量无监督数据提高映射关系建模的准确性;

3)提出受限图卷积矩阵补全模型, 以有效融合目标域稀疏数值评分和辅助域二元评分, 有效避免迁移学习中的负迁移现象.

1 相关工作

1.1 传统跨域推荐

跨域推荐算法基于迁移学习或者多任务学习技术, 有效利用辅助域上重要信息缓解目标域数据稀疏性难题, 通常根据所讨论场景中用户的重叠情况被分为 3 类: 完全不重叠、部分重叠和完全重叠.

针对完全不重叠情况, Li 等人[13] 提出码本迁移模型(CBT), 该模型利用辅助域的丰富评分信息提取簇级评分模式, 即密码本, 并将密码本迁移到目标域上来提升目标域矩阵分解的准确性. 在此基础上, Li 等人[14] 又提出评分矩阵生成模型(RMGM), 通过多任务学习方法提高推荐性能. Zhang 等人[15] 提出结合标签系统语义相关性的跨域推荐模型(SCT), 利用目标域和辅助域标签语义信息的相似关系构建连接 2

个平台用户或项目的桥梁，实现辅助域知识向目标域的迁移. Li 等人[16]提出基于对抗学习深度稀疏自编码器的跨域推荐模型(DSAP-AL)，利用对抗生成网络对齐目标域和辅助域的用户和项目的潜在因子空间，并结合深度稀疏自编码器实现知识迁移.

针对部分重叠情况，Jiang 等人[17]提出半监督迁移学习模型(XPTRANS). Zhang 等人[18]提出基于核诱导知识迁移的跨域推荐模型 (KerKT). Zhu 等人[19]提出结合图和注意力机制的双向迁移跨域推荐模型(GA-DTCDR)，利用异构图模型捕获用户和项目的特征，并提出注意力机制来结合共享用户的特征，实现用户特征的双向迁移. Li 等人[20]进一步提出基于对偶度量学习的跨域推荐模型(DML)，通过将度量学习引入对偶学习减轻了对重叠用户数目的要求.

针对用户完全重叠情况的研究最为广泛，目前该方面的研究通常可以分为信息集成和知识集成 2 种. 在信息集成方面，Berkovsky 等人[21]提出基于邻域的跨域推荐模型(N-CDCF)，该模型是基于邻域的协同过滤模型(N-CF)[22]的跨域版本. 假定用户在不同域上共享潜在因子，Singh 等人[23]提出联合矩阵分解模型(CMF). 这些模型通过拼接目标域和辅助域评分矩阵来缓解目标域数据稀疏问题. 针对知识集成，Hu 等人[24]提出跨域三元分解模型(CDTF)来捕捉用户、项目和域之间的三元关系，并基于张量分解实现跨域推荐. Loni 等人[25]提出跨域因子分解机模型(CDFM)，该模型从辅助域中的用户评分信息中提取用户特征并将其转移到目标域，以提升目标域上的推荐性能. Yuan 等人[26]提出深度域适应跨域推荐模型(DARec)来提取和迁移潜在的评分模式. 通过进一步假设存在一些辅助域与目标域共享项目，Yu 等人[27-28]提出双侧跨域协同过滤算法，分别提取域无关特征和域依赖特征并转移到目标域，以提高目标域推荐性能. Pan 等人[29]提出坐标系统迁移模型(CST)，分别从用户辅助域和项目辅助域中提取用户和项目隐向量，并将它们迁移到目标域以辅助目标域进行矩阵分解. Yu 等人[30]提出具有隐私保护功能的跨域推荐模型(PPCDHWRec)，仅迁移用户在辅助域的域依赖和域无关特征，保留项目的隐特征，提升了目标域推荐系统的性能，且实现了辅助域原始评分隐私保护的效果.

1.2 传统跨评分推荐

Pan 等人[6]提出 TCF 算法，该算法针对目标域辅助域的评分矩阵进行如下联合分解，以缓解目标域数据的稀疏性难题.

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{B},\tilde{\boldsymbol{B}}} \mathcal{F}\left(\boldsymbol{R}\sim\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\right)+\lambda\mathcal{F}\left(\tilde{\boldsymbol{R}}\sim\boldsymbol{U}\tilde{\boldsymbol{B}}\boldsymbol{V}^{\mathrm{T}}\right), \tag{1}$$

其中 $\boldsymbol{U}$ 表示共享的用户隐向量矩阵，$\boldsymbol{V}$ 表示共享的项目隐向量矩阵，$\boldsymbol{B}$ 和 $\tilde{\boldsymbol{B}}$ 分别表示目标域和辅助域的数据依赖的簇级评分模式. 根据用户和项目隐向量约束条件的不同，TCF 可以分为 2 种变体，分别是联合矩阵三分解模型(CMTF)和联合奇异值分解模型(CSVD)，其中 CSVD 要求 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 是正交矩阵，即，具有更好的推荐性能.

考虑到 TCF 算法效率较低以及目标域和辅助域数据异构会造成用户隐向量发生变化，Pan 等人[7]提出 iTCF 算法，与 TCF 算法不同，iTCF 算法在评分矩阵分解过程中不再考虑簇级评分模式，直接将评分矩阵分解为用户隐向量矩阵和项目隐向量矩阵. 该算法通过求解如下优化问题实现辅助域和目标域用户隐特征的交互：

$$\min_{\boldsymbol{U},\boldsymbol{W},\boldsymbol{V}} \mathcal{F}\left(\boldsymbol{R}\sim\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}\right)+\lambda\mathcal{F}\left(\tilde{\boldsymbol{R}}\sim\boldsymbol{W}\boldsymbol{V}^{\mathrm{T}}\right),\ \text{s.t.}\boldsymbol{E}=\tilde{\boldsymbol{E}}, \tag{2}$$

其中 $\boldsymbol{U}$ 和 $\boldsymbol{W}$ 分别表示目标域和辅助域的用户隐向量矩阵，$\boldsymbol{V}$ 表示共享的项目隐向量矩阵，$\boldsymbol{E}$ 和 $\tilde{\boldsymbol{E}}$ 分别表示预测模型在 2 种评分上的误差.

尽管 iTCF 考虑到数据异构造成的用户隐向量的变化，但作为一种基于联合方式的迁移学习算法，该算法包含 2 个松耦合的矩阵分解任务，仍不能充分利用辅助域评分数据. Pan 等人[8]进一步提出 TMF 算法，该算法是一种混合迁移学习策略，包含联合的(collective)基于特征的迁移和综合的(integrative)基于实例的迁移 2 种迁移学习方式. 通过在目标域的分解过程中考虑辅助域提取的用户喜欢和不喜欢偏好，加强了 2 个松耦合矩阵分解之间的联系，可以更充分地进行知识迁移.

考虑到不同域的评分模式既包含不同域之间共享的评分模式，又包含各领域特有的评分模式，Zhang 等人[9]和 Jiang 等人[10]对不同域的评分模式进一步分解，以迁移不同域中共享的评分模式. Zhang 等人[9]的方法中目标域和辅助域的簇级评分模式 $\boldsymbol{B}_{\mathrm{t}}$ 和 $\boldsymbol{B}_{\mathrm{s}}$ 分别被分解为

$$\boldsymbol{B}_{\mathrm{t}}=[\boldsymbol{S},\boldsymbol{S}_{\mathrm{t}}],\ \boldsymbol{B}_{\mathrm{s}}=[\boldsymbol{S},\boldsymbol{S}_{\mathrm{a}}], \tag{3}$$

其中 $\boldsymbol{S}$ 表示共享的评分模式矩阵，$\boldsymbol{S}_{\mathrm{t}}$ 和 $\boldsymbol{S}_{\mathrm{a}}$ 分别表示目标域和辅助域特有的评分模型矩阵. Jiang 等人[10]的方法中目标域和辅助域的簇级评分模式 $\boldsymbol{B}$ 和 $\tilde{\boldsymbol{B}}$ 分别被分解为

$$\boldsymbol{B}=\boldsymbol{D}+\boldsymbol{E},\ \tilde{\boldsymbol{B}}=\boldsymbol{D}+\tilde{\boldsymbol{E}}, \tag{4}$$

其中 $\boldsymbol{D}$ 表示共享的低秩评分模式部分，$\boldsymbol{E}$ 和 $\tilde{\boldsymbol{E}}$ 分别表

示目标域和辅助域特有的评分模式部分. 在此基础上, Zhang 等人[9]提出了 EKT 算法, 通过用户和项目评分相关的几何结构信息构建正则项, 以此约束矩阵分解过程中用户和项目隐向量的学习, 避免了迁移过程中的负迁移和迁移不足问题. Jiang 等人[10]提出了 DLSCF 算法, 该算法以多层的方式对评分矩阵进行分解来捕获项目的潜在类别和潜在子类别之间的层次关系, 多层分解的方式为:

$$\boldsymbol{R}=\boldsymbol{U}_1\boldsymbol{B}_1\boldsymbol{V}_1,\ \boldsymbol{B}_1=\boldsymbol{U}_2\boldsymbol{B}_2\boldsymbol{V}_2,\ \cdots,\ \boldsymbol{B}_{l-1}=\boldsymbol{U}_l\boldsymbol{B}_l\boldsymbol{V}_l, \tag{5}$$

其中 l 表示多层矩阵分解的层数.

上述跨评分协同过滤算法将辅助域重要知识迁移到目标域, 以有效提升目标域上推荐结果的准确性. 但是上述方法默认目标域不同区域具有相同的评分密度, 对不同区域采取一致的评分预测策略, 忽视了评分密度对用户和项目隐向量求解精度的影响, 导致评分稀疏区域评分预测不够准确. 为克服这一不足, 本文将在第 2 节中提出 CRCRCF 模型, 相对于传统跨评分协同过滤范式, 该算法不仅能有效挖掘辅助域的重要知识, 而且可以挖掘目标域中评分密集区域的重要知识, 可以进一步提升评分稀疏区域的评分预测精度.

2 CRCRCF 模型

传统跨评分协同过滤范式往往假设推荐系统中的所有用户和所有项目都具有稀疏的数值评分, 对全体用户和全体项目同等看待. 该范式将辅助域 1, 0 评分信息中挖掘到的重要知识无差别地迁移到 1~5 数值评分组成的目标域, 以提升目标域上推荐算法的性能. 然而, 实际上, 尽管目标域整体数值评分较为稀疏, 但是仍然有部分活跃用户在部分热门项目上具有较高的评分密度. 如图 1 所示, 如果基于目标域评分个数将用户进一步划分为活跃用户和非活跃用户, 将项目进一步划分为热门项目和非热门项目, 则 a, b, c, d 这 4 个区域的 1~5 数值评分密度满足 $density(a)>density(b)>density(d)$, $density(a)>density(c)>density(d)$, 且 $density(b)$ 与 $density(c)$ 没有明确的大小关系.

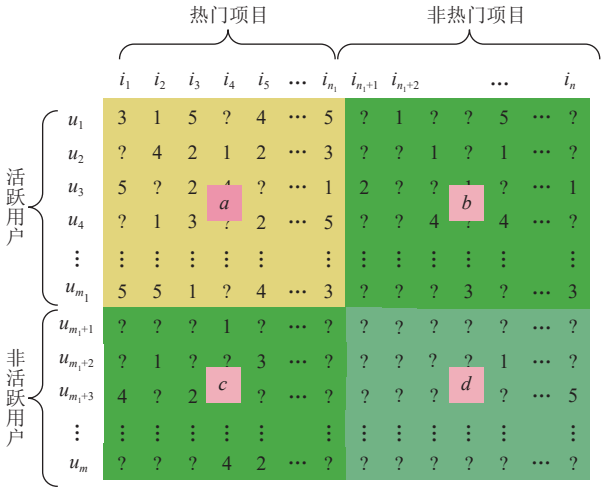


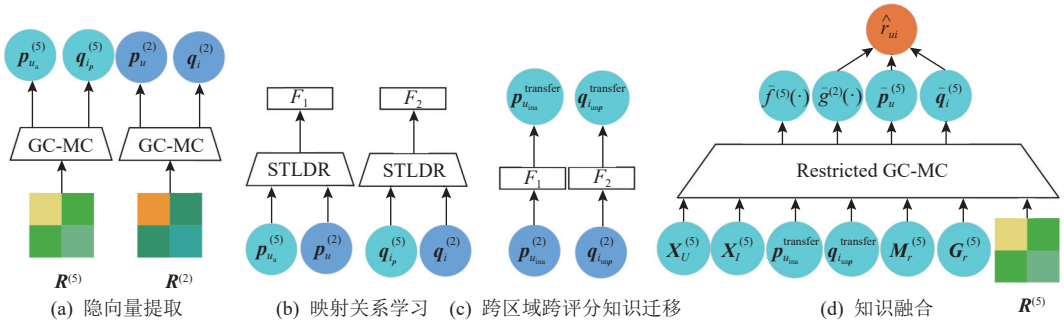
Fig. 1 Division of users and items in the target domain

图 1 目标域用户和项目的划分

矩阵评分密度的定义有:

定义 4. 矩阵评分密度. 设矩阵 $\boldsymbol{A}$ 的大小为 $m\times n$, 矩阵中评分个数为 s, 则评分密度 $density(\boldsymbol{A})=s/m\times n$.

很明显, 目标域中不同的区域具有不同的数值评分密度, 高密度区域反馈信息较丰富, 对辅助域信息的依赖较小, 低密度区域反馈信息较匮乏, 对辅助域信息的依赖较大. 因此, 应该有针对性地设计辅助域到目标域的知识迁移策略, 以有效提升各区域上评分预测的准确性. 为此, 本文提出了 CRCRCF 算法, 该算法相对于传统跨评分协同过滤范式将有望拥有更好的推荐性能, 其模型结构如图 2 所示. 首先, 图 2(a)利用图卷积矩阵补全算法(GC-MC)提取目标域活跃用户和热门项目, 以及辅助域中全体用户和项目的隐向量. 其次, 图 2(b)针对活跃用户和热门项



(a) 隐向量提取 (b) 映射关系学习 (c) 跨区域跨评分知识迁移 (d) 知识融合

Fig. 2 Structure diagram of CRCRCF model

图 2 CRCRCF 模型结构图

目，构建基于自教学习的深度回归网络（STLDR），分别学习目标域和辅助域上 2 种评分对应的用户隐向量和项目隐向量的映射关系. 然后，图 2(c)将活跃用户和热门项目的隐向量的映射关系泛化到目标域非活跃用户和非热门项目上，利用非活跃用户和非热门项目在辅助域上相对准确的隐向量推导其在目标域上的隐向量，依次实现了跨区域映射关系迁移和跨评分隐向量信息迁移. 最后，图 2(d)以求得的非活跃用户和非热门项目在目标域上的隐向量为约束，提出受限图卷积矩阵补全模型（restricted-GC-MC），求解用户和商品最终的隐向量，并对评分进行预测，实现了信息融合. 本文所提出的 CRCRCF 是一种迁移学习算法. 迁移学习算法通常包括“迁移什么”（what to transfer）与“如何迁移”（how to transfer）2 个核心问题. 其中，图 2(b)(c)可以看作是“迁移什么”模块，对应 2.4 节内容，图 2(b)求得了映射关系这一可迁移量，图 2(c)求得了利用映射关系得到的目标域隐向量这一可迁移量；图 2(d)可以看作是“如何迁移”模块，对应 2.5 节内容，通过自适应的方式实现了迁移的知识与目标域信息的融合.

2.1 用户和项目划分

本节基于用户和项目的评分个数将用户和项目划分为活跃用户和非活跃用户、热门项目和非热门项目，以更有针对性地推荐. 相关定义有：

定义 5. 活跃用户和非活跃用户. 对于任意一个用户 $u \in U = \{u_1, u_2, \cdots, u_m\}$，令 d_u 表示目标域用户 u 的评分个数（即用户 u 评价的所有项目的个数）. 将用户按照评分个数由大到小排序，取前 μ_1 的用户作为活跃用户，剩下的用户作为非活跃用户. 其中 μ_1 是一个预先设定的百分比参数，称为用户活跃度阈值，μ_1 的最优值通过实验来确定.

定义 6. 热门项目和非热门项目. 对于任意一个项目 $i \in I = \{i_1, i_2, \cdots, i_n\}$，令 d_i 表示目标域项目的评分个数（即评价过项目 i 的所有用户的个数）. 将项目按照评分个数由大到小排序，取前 μ_2 的项目作为热门项目，剩下项目作为非热门项目. 其中 μ_2 称为项目热门度阈值.

由定义 5 和定义 6 可以看出，活跃用户和热门项目的概念是相对概念，不同的阈值决定不同的活跃用户和热门项目集合.

2.2 问题形式化

本文的推荐场景如图 3 所示，$\boldsymbol{R}^{(5)}$ 为目标域数据，是 5 分制（1~5 分）评分矩阵，$\boldsymbol{R}^{(2)}$ 为辅助域数据，是二元（1/0，即喜欢/不喜欢）评分矩阵，$\boldsymbol{R}^{(5)}$ 和 $\boldsymbol{R}^{(2)}$ 共享相同的用户集合 U 和项目集合 I. 在图 3 中，为了便于观察 $\boldsymbol{R}^{(5)}$ 和 $\boldsymbol{R}^{(2)}$，我们用前后 2 个切片对其进行分别表示. 在 $\boldsymbol{R}^{(5)}$ 中 $U_{\mathrm{a}} = \{u_1, u_2, \cdots, u_{m_1}\}$ 和 $U_{\mathrm{ina}} = \{u_{m_1+1}, u_{m_1+2}, \cdots, u_m\}$ 分别表示活跃用户和非活跃用户集合，$I_{\mathrm{p}} = \{i_1, i_2, \cdots, i_{n_1}\}$ 和 $I_{\mathrm{unp}} = \{i_{n_1+1}, i_{n_1+2}, \cdots, i_n\}$ 分别表示热门项目和非热门项目集合. 所以 $a^{(i)}, b^{(i)}, c^{(i)}, d^{(i)}$ (i=5, 2) 分别表示目标域和辅助域上由活跃用户和热门项目、活跃用户和非热门项目、非活跃用户和热门项目、非活跃用户和非热门项目构成的评分区域.

通常活跃用户相对于非活跃用户会提供更多的评分，热门项目相对于非热门项目会获得更多的评分，因此，$density(a^{(i)})$ 比较高，且 $density(d^{(i)}) < density(b^{(i)}$ 或 $c^{(i)}) < density(a^{(i)})$，注意 $density(b^{(i)})$ 和 $density(c^{(i)})$ 通常不存在明显的大小关系，其中 i=5, 2. 此外，相对于较为复杂的数值评分，全体用户往往更倾向进行 1, 0 二元评分，因此，$density(\boldsymbol{R}^{(5)}) < density(\boldsymbol{R}^{(2)})$.

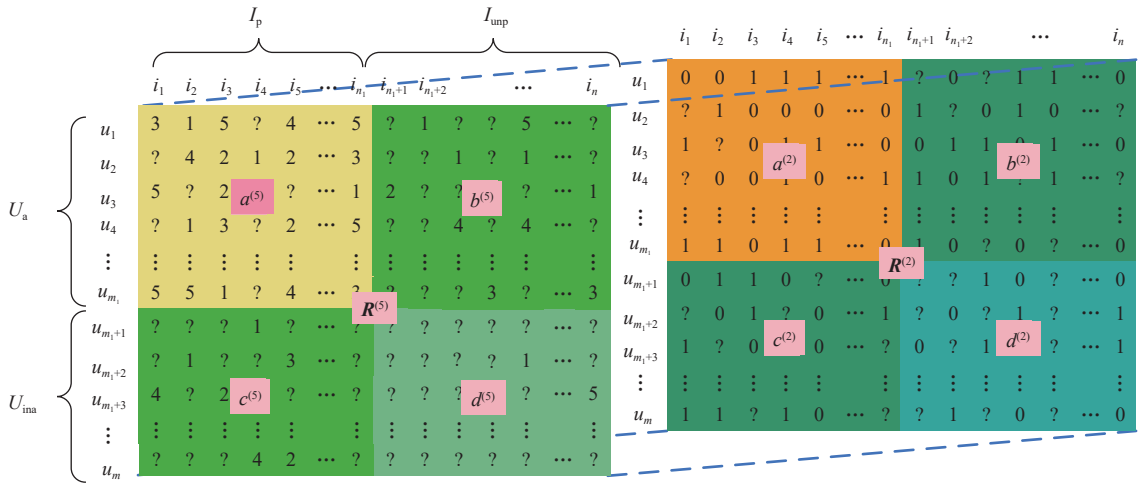


Fig. 3 The cross-region cross-rating recommendation scenario

图 3 跨区域跨评分推荐场景

本文所提 CRCRCF 模型的目标是充分挖掘辅助域较丰富的二元评分和目标域评分密集区域较为丰富的数值评分, 针对目标域不同区域制定个性化的知识迁移策略以生成更为准确的推荐.

2.3 隐向量提取

2.3.1 目标域隐向量提取

图卷积矩阵补全模型 GC-MC 被用来提取目标域用户和项目隐向量特征. 相较于传统协同过滤算法, GC-MC 可以更为精准地进行用户和项目表征[11]. 值得注意的是, 我们利用整体的评分矩阵 $\boldsymbol{R}^{(5)}$ 对 GC-MC 进行训练, 而不是仅利用活跃用户和热门项目关联的区域 $a^{(5)}$ 所对应的评分子矩阵 $\boldsymbol{R}(a^{(5)})$ 进行训练. 由于 $\boldsymbol{R}^{(5)}$ 比 $\boldsymbol{R}(a^{(5)})$ 具有更多的评分信息, 因此使用整体的评分矩阵 $\boldsymbol{R}^{(5)}$ 训练 GC-MC 可以获得更为精确的隐向量特征.

设用户-商品二部图为 $G^{(5)}=(\mathcal{W}^{(5)},\mathcal{E}^{(5)})$, 其中, $\mathcal{W}^{(5)}=U\cup I$ 为用户和项目节点集合, $\mathcal{E}^{(5)}=\left\{(u,r_{ui}^{(5)},i)|u,i\in\mathcal{W}^{(5)},\ r_{ui}^{(5)}\in\{1,2,3,4,5\}\right\}$ 为边集合. 图中节点的初始特征使用 one-hot 编码表示, 全体用户和全体项目节点的初始特征矩阵分别为 $\boldsymbol{X}_U^{(5)}$ 和 $\boldsymbol{X}_I^{(5)}$.

首先, 使用基于图卷积神经网络的编码器 $f^{(5)}(\cdot)$ 对节点初始特征矩阵 $\boldsymbol{X}_U^{(5)}$ 和 $\boldsymbol{X}_I^{(5)}$ 编码得到节点的嵌入表示 $\boldsymbol{p}_u^{(5)}$ 和 $\boldsymbol{q}_i^{(5)}$, 编码过程为:

$$[\boldsymbol{p}_u^{(5)},\boldsymbol{q}_i^{(5)}]=f^{(5)}(\boldsymbol{X}_U^{(5)},\boldsymbol{X}_I^{(5)},\boldsymbol{M}_1,\boldsymbol{M}_2,\boldsymbol{M}_3,\boldsymbol{M}_4,\boldsymbol{M}_5),\tag{6}$$

其中 $\boldsymbol{M}_r\in\{0,1\}^{m\times n},r=1,2,3,4,5$ 为评分 r 对应的邻接矩阵. 编码器 $f^{(5)}(\cdot)$ 包含一个图卷积层和一个线性层. 在图卷积层中, 用户 u 的向量表示计算过程为:

$$\begin{cases}\boldsymbol{h}_u^{(5)}=\sigma\left[accum\left(\sum\limits_{i\in\mathcal{N}_1(u)}\boldsymbol{\mu}_{i\to u,1}^{(5)},\cdots,\sum\limits_{i\in\mathcal{N}_5(u)}\boldsymbol{\mu}_{i\to u,5}^{(5)}\right)\right],\\ \boldsymbol{\mu}_{i\to u,r}^{(5)}=\dfrac{1}{\sqrt{|\mathcal{N}_r(u)||\mathcal{N}_r(i)|}}\boldsymbol{W}_r^{(5)}\boldsymbol{x}_i^{(5)},\end{cases}\tag{7}$$

其中 σ 为 ReLu 激活函数, $accum(\cdot)$ 为向量的拼接或者加和. $\mathcal{N}_r(u)$ 和 $\mathcal{N}_r(i)$ 分别表示用户 u 和项目 i 在邻接矩阵 $\boldsymbol{M}_r$ 中的邻居节点集合. $\boldsymbol{\mu}_{i\to u,r}^{(5)}$ 为图卷积过程中项目 i 传递给用户 u 的信息. $\boldsymbol{W}_r^{(5)}$ 为权重矩阵, $\boldsymbol{x}_i^{(5)}$ 为项目 i 的(初始)特征向量. 在图卷积操作后, 将用户特征 $\boldsymbol{h}_u^{(5)}$ 输入到线性层, 线性层输出最终编码后的用户特征:

$$\boldsymbol{p}_u^{(5)}=\sigma\left(\boldsymbol{W}\boldsymbol{h}_u^{(5)}\right),\tag{8}$$

其中 σ 为 ReLu 激活函数, $\boldsymbol{W}$ 为权重矩阵. 对所有用户节点和项目节点进行编码操作后, 可以得到用户和项目的嵌入表示 $\boldsymbol{p}_u^{(5)}\in\mathbb{R}^d$ 和 $\boldsymbol{q}_i^{(5)}\in\mathbb{R}^d$, d 为嵌入表示的维度.

然后, 使用双线性解码器 $g^{(5)}(\cdot)$ 通过预测用户和项目之间的评分类别来完成图的重构. 用户 u 对项目 i 评分的计算公式为:

$$\begin{aligned}&\hat{r}_{ui}^{(5)}=g^{(5)}(u,i)=E_{p\left(\hat{r}_{ui}^{(5)}=r\right)}[r]=\sum_{r\in\mathbb{R}}rp\left(\hat{r}_{ui}^{(5)}=r\right),\\&p\left(\hat{r}_{ui}^{(5)}=r\right)=\frac{\mathrm{e}^{(\boldsymbol{p}_u)^{\mathrm{T}}\boldsymbol{Z}_r\boldsymbol{q}_i}}{\sum\limits_{s=1}^{R}\mathrm{e}^{(\boldsymbol{p}_u)^{\mathrm{T}}\boldsymbol{Z}_s\boldsymbol{q}_i}},\end{aligned}\tag{9}$$

其中 $p\left(\hat{r}_{ui}^{(5)}=r\right)$ 表示评分 $\hat{r}_{ui}^{(5)}$ 为 r 的概率, $\boldsymbol{Z}_r\in\mathbb{R}^{d\times d}$ 为权重矩阵.

最后, 使用 Adam[31] 优化器最小化损失函数完成模型的训练:

$$L^{(5)}=-\sum_{(u,i)\in D^{(5)}}\sum_{r=1}^{5}I\left[r_{ui}^{(5)}=r\right]\lg p\left(\hat{r}_{ui}^{(5)}=r\right),\tag{10}$$

其中 $I[\cdot]$ 为指示函数, 当 $r_{ui}^{(5)}=r$ 时, 其值为 1, 否则为 0. $D^{(5)}$ 表示有评分 $r_{ui}^{(5)}$ 的 (u,i) 对集合.

如上所述, 我们利用 GC-MC 算法实现了对目标域活跃用户隐向量和热门项目隐向量的提取.

2.3.2 辅助域隐向量提取

类似地, 针对 $r_{ui}^{(2)}\in\{0,1\}$ 的情况, 利用 GC-MC 算法可以提取辅助域用户隐向量和项目隐向量.

2.4 知识迁移

2.4.1 跨区域映射关系迁移

由于活跃用户和热门项目相关的评分较为丰富, 有助于求解相对准确的隐向量特征, 本文首先针对活跃用户和热门项目计算隐向量特征, 进而建模活跃用户和热门项目在 2 种评分上对应的隐向量映射关系. 令 $\boldsymbol{p}_{u_{\mathrm{a}}}^{(5)}$ 和 $\boldsymbol{q}_{i_{\mathrm{p}}}^{(5)}$ 分别表示 5 分制评分矩阵 $\boldsymbol{R}^{(5)}$ 对应的活跃用户 u_{a}(user_active)和热门项目 i_{p}(item_popular)隐向量, $\boldsymbol{p}_{u_{\mathrm{a}}}^{(2)}$ 和 $\boldsymbol{q}_{i_{\mathrm{p}}}^{(2)}$ 分别表示二元评分矩阵 $\boldsymbol{R}^{(2)}$ 对应的活跃用户和热门项目的隐向量.

基于获取的活跃用户隐向量特征 $\boldsymbol{p}_{u_{\mathrm{a}}}^{(2)}$ 和 $\boldsymbol{p}_{u_{\mathrm{a}}}^{(5)}$, 我们以 $\boldsymbol{p}_{u_{\mathrm{a}}}^{(2)}$ 作为输入, 以 $\boldsymbol{p}_{u_{\mathrm{a}}}^{(5)}$ 作为输出, 可以构建深度回归网络学习它们之间的映射关系 F_1. 同样, 我们可以学习热门项目对应的 2 种隐向量映射关系 F_2. 然而, 由于活跃用户和热门项目数量往往偏少, 直接构建深度回归网络效果不够理想. 以活跃用户隐向量映射关系建模为例, 考虑到推荐平台还存在大量的非活跃用户, 他们在辅助域中的评分与目标域上的相比较为丰富, 所提取的隐向量更为准确, 而且他们的隐向量特征与活跃用户的隐向量特征共享同一特征空间, 为进一步提升映射关系建模的准确性, 本文映射关系建模时, 首先利用大量非活跃用户在辅助域的隐向量特征 $\boldsymbol{p}_{u_{\mathrm{ina}}}^{(2)}$ 作为无监督训练数据训练栈式降噪自编码器(stacked denoising autoencoders, SDAE)[32], 获

取隐向量特征的低维高层表示. 然后，在编码器的基础上外接一层线性回归单元，构建深度回归网络，并利用少量对应活跃用户的有监督训练数据$\{\boldsymbol{p}_{u_a}^{(2)},\boldsymbol{p}_{u_a}^{(5)}\}$对深度回归网络进行训练，建模映射关系. 针对栈式降噪自编码器的训练过程如图 4 所示，其中：图 4(a)进行逐层降噪自编码器(denoising autoencoder, DAE)学习；图 4(b)将多层降噪自编码器进行拼接；图 4(c)利用 BP[33] 算法对权重微调.

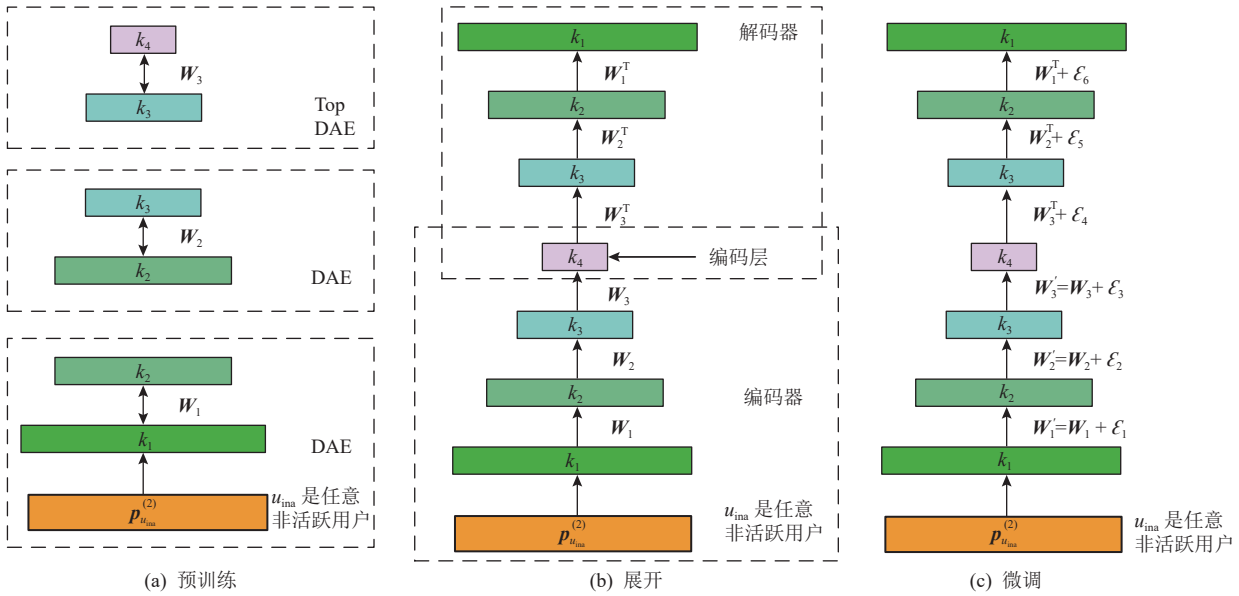


Fig. 4 Training the SDAE model with the latent vectors of inactive users in the auxiliary domain

图 4 利用非活跃用户的辅助域隐向量训练 SDAE 模型

通常，针对单个降噪自编码器训练时，首先让$\boldsymbol{x}=\boldsymbol{p}_{u_{ina}}^{(2)}$表示原始的训练数据，将$\boldsymbol{x}$添加高斯噪声转化为$\tilde{\boldsymbol{x}}$，其中，$\tilde{\boldsymbol{x}}\sim N(\boldsymbol{x},\sigma^2 I)$. 然后，对$\tilde{\boldsymbol{x}}$进行编码得到低维特征表示：

$$\boldsymbol{y}=f(\tilde{\boldsymbol{x}})=S(\boldsymbol{W}\tilde{\boldsymbol{x}}+\boldsymbol{b}), \tag{11}$$

其中$\boldsymbol{W}$和$\boldsymbol{b}$分别表示编码器权值矩阵和偏置向量，S表示 ReLu 激活函数. 最后，对$\boldsymbol{y}$进行解码得到输入数据的重构数据：

$$\boldsymbol{z}=g(\boldsymbol{y})=S(\boldsymbol{W}'\boldsymbol{y}+\boldsymbol{b}'), \tag{12}$$

其中$\boldsymbol{W}'$和$\boldsymbol{b}'$表示解码器权值和偏置. 损失函数为

$$J(\boldsymbol{X},\boldsymbol{Z})=\frac{1}{2}\sum_{i=1}^{M}\|\boldsymbol{x}^{(i)}-\boldsymbol{z}^{(i)}\|^2, \tag{13}$$

其中 M 为样本数.

在深度回归网络进行训练时，定义损失函数为：

$$L=\frac{1}{2|U_a|}\sum_{u_a\in U_a}\left\|\hat{\boldsymbol{p}}_{u_a}^{(5)}-\boldsymbol{p}_{u_a}^{(5)}\right\|^2, \tag{14}$$

其中$\boldsymbol{p}_{u_a}^{(5)}$是活跃用户$u_a$基于$\boldsymbol{R}^{(5)}$矩阵学习得到的隐向量，$\hat{\boldsymbol{p}}_{u_a}^{(5)}=F_1(\boldsymbol{p}_{u_a}^{(2)})$是基于深度回归网络预测的隐向量，其中$\boldsymbol{p}_{u_a}^{(2)}$为活跃用户$u_a$基于$\boldsymbol{R}^{(2)}$矩阵学习得到的隐向量. 建模过程如图 5 所示，其中线性回归单元不含有任何激活函数，仅仅计算各个输入单元的加权和. 使用图 4 中已训练好的 SDAE 中编码器的最终权重$\boldsymbol{W}_1',\boldsymbol{W}_2',\boldsymbol{W}_3'$初始化深度回归网络中编码器的权重，随机初始化最外层的线性回归单元权重$\boldsymbol{W}_4'$. 然后用 BP 算法对深度回归网络所有权重进行学习，得到最终的深度回归网络，即映射关系F_1.

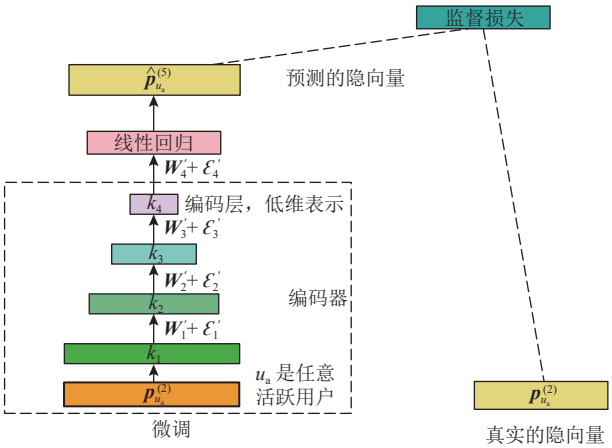


Fig. 5 Training the deep regression network

图 5 训练深度回归网络

本文的映射关系求解模型符合自教学习[12] 范式. 自教学习范式用于在监督分类任务中使用无标记数据提升分类任务的性能，并且该范式不假设未标记的数据与已标记的数据遵循相同的类标签或生成分

布. 自教学习范式包含 2 个阶段: 1)使用稀疏编码方法从大量的无标签特征数据中学习得到原始特征的高层表示; 2)将有标签训练数据从原始特征空间映射到高层特征表示的新特征空间, 得到新的训练样本数据, 并进行监督学习. 我们将本文的映射关系建模算法称为基于自教学习深度回归网络的映射关系建模算法. 同样的方法可以用于建模热门项目对应的 2 种隐向量的映射关系 .

本文基于自教学习深度回归网络的映射关系建模(self-taught learning based deep regression network for mapping relationship modeling, STLDR)算法如算法 1 所示. 为描述方便, 我们仅以活跃用户映射关系建模为例, 该算法返回训练后深度回归网络的所有权重参数统一表示为 F_1.

算法 1. STLDR 算法.

输入: 非活跃用户辅助域隐向量特征 $\boldsymbol{p}_{u_{\text{ina}}}^{(2)}$, 活跃用户目标域和辅助域的隐向量特征 $\boldsymbol{p}_{u_{\text{a}}}^{(2)}$ 和 $\boldsymbol{p}_{u_{\text{a}}}^{(5)}$;

输出: 活跃用户从辅助域到目标域的映射关系 F_1.

① $[\boldsymbol{W},\boldsymbol{B}]=SDAE(\boldsymbol{p}_{u_{\text{ina}}}^{(2)})$;

/* 基于非活跃用户辅助域隐向量训练 SDAE 模型*/

② $F_1=DeepRegression(\boldsymbol{p}_{u_{\text{a}}}^{(2)},\boldsymbol{p}_{u_{\text{a}}}^{(5)},\boldsymbol{W},\boldsymbol{B})$;

/*利用少量对应活跃用户的有监督训练数据 $\{\boldsymbol{p}_{u_{\text{a}}}^{(2)},\boldsymbol{p}_{u_{\text{a}}}^{(5)}\}$ 对深度回归网络进行训练, 并建模映射关系 F_1*/

③ return F_1.

进一步, 我们将活跃用户和热门项目对应的隐向量映射关系 F_1 和 F_2 扩展到目标域全体区域, 实现了跨区域映射关系迁移.

2.4.2 跨评分隐向量信息迁移

本节研究如何实现辅助域隐向量信息向目标域的迁移. 令 $\boldsymbol{p}_{u_{\text{a}}}^{(5)}$ 为活跃用户 u_{a} 基于评分矩阵 $\boldsymbol{R}^{(5)}$ 经 GC-MC 算法学习得到的隐向量, $\boldsymbol{q}_{i_{\text{p}}}^{(5)}$ 为热门项目 i_{p} 基于 $\boldsymbol{R}^{(5)}$ 经 GC-MC 算法学习得到的隐向量, 跨评分隐向量信息迁移过程为:

$$\boldsymbol{p}_u^{\text{transfer}}=\begin{cases}\boldsymbol{p}_u^{(5)}, & u=u_{\text{a}},\\ \boldsymbol{p}_u^{(5)F_1}, & u=u_{\text{ina}},\end{cases} \tag{15}$$

$$\boldsymbol{q}_i^{\text{transfer}}=\begin{cases}\boldsymbol{q}_i^{(5)}, & i=i_{\text{p}},\\ \boldsymbol{q}_i^{(5)F_2}, & i=i_{\text{unp}},\end{cases} \tag{16}$$

其中 $\boldsymbol{p}_{u_{\text{ina}}}^{(5)F_1}=F_1(\boldsymbol{p}_{u_{\text{ina}}}^{(2)})$ 为利用映射关系得到的非活跃用户 u_{ina}(user_inactive) 在目标域上的隐向量, $\boldsymbol{q}_{i_{\text{unp}}}^{(5)F_2}=F_2(\boldsymbol{q}_{i_{\text{unp}}}^{(2)})$ 为利用映射关系得到的非热门项目 i_{unp}(item_unpopular) 在目标域上的隐向量.

2.5 知识融合

本节研究如何将 2.4 节中所迁移的知识与目标域原始数值评分进行合理融合. 令 $r_{ui}^{(5)}$ 为数值矩阵 $\boldsymbol{R}^{(5)}$ 中用户 u 对项目 i 的评分, $\bar{\boldsymbol{p}}_u^{(5)}$ 为本文跨区域跨评分协同过滤模型最终求解的任意用户 u 的隐向量, $\bar{\boldsymbol{q}}_i^{(5)}$ 为最终求解的任意项目 i 的隐向量. $\bar{z}_r$ 为最终学习的双线性解码器对应评分 r 的权重矩阵, $\bar{f}^{(5)}(\cdot)$ 和 $\bar{g}^{(5)}(\cdot)$ 为最终学习的基于图卷积神经网络的编码器和双线性解码器.

针对目标域具有不同评分密度的各个区域, 本文通过对 GC-MC 算法添加约束实现了可迁移量的迁移. 我们通过求解式(17)中优化问题来获取目标域最终的用户和项目隐向量, 实现知识从辅助域和目标域评分密集区域向目标域评分非密集区域的转移.

$$\begin{aligned}\min -\sum_{(u,i)}\sum_{r=1}^{5} I\left[r_{ui}^{(5)}=r\right]\lg p\left(\hat{r}_{ui}^{(5)}=r\right)+\lambda\Bigg(&\left\|\bar{\boldsymbol{q}}_i^{(5)}-\boldsymbol{q}_i^{\text{transfer}}\right\|^2+\\ &\left\|\bar{\boldsymbol{p}}_u^{(5)}-\boldsymbol{p}_u^{\text{transfer}}\right\|^2\Bigg),\end{aligned} \tag{17}$$

其中 λ 为正则化系数, 且

$$\begin{cases}\hat{r}_{ui}^{(5)}=\bar{g}^{(5)}(u,i)=E_{p\left(\hat{r}_{ui}^{(5)}=r\right)}[r]=\displaystyle\sum_{r\in\mathbb{R}} r p\left(\hat{r}_{ui}^{(5)}=r\right),\\ p\left(\hat{r}_{ui}^{(5)}=r\right)=\dfrac{\mathrm{e}^{\left(\bar{\boldsymbol{p}}_u^{(5)}\right)^{\mathrm{T}}\bar{\boldsymbol{z}}_r\bar{\boldsymbol{q}}_i^{(5)}}}{\displaystyle\sum_{s=1}^{R}\mathrm{e}^{\left(\bar{\boldsymbol{p}}_u^{(5)}\right)^{\mathrm{T}}\bar{\boldsymbol{z}}_s\bar{\boldsymbol{q}}_i^{(5)}}},\\ \left[\bar{\boldsymbol{p}}_u^{(5)},\bar{\boldsymbol{q}}_i^{(5)}\right]=\bar{f}^{(5)}\left(\boldsymbol{X}_U^{(5)},\boldsymbol{X}_I^{(5)},\boldsymbol{M}_1,\boldsymbol{M}_2,\boldsymbol{M}_3,\boldsymbol{M}_4,\boldsymbol{M}_5\right).\end{cases} \tag{18}$$

在式(17)中, 采用 $\boldsymbol{p}_u^{\text{transfer}}$ 对目标域活跃用户和非活跃用户的隐向量进行约束, 如果 u 为活跃用户, 则 $\boldsymbol{p}_u^{\text{transfer}}=\boldsymbol{p}_u^{(5)}$, 即以活跃用户 u 基于评分矩阵 $\boldsymbol{R}^{(5)}$ 学习得到的隐向量作为约束. 如果 u 为非活跃用户, 则 $\boldsymbol{p}_u^{\text{transfer}}=\boldsymbol{p}_u^{(5)F_1}$, 即以非活跃用户 u 基于映射关系得到的隐向量作为约束. 对于项目, 采用 $\boldsymbol{q}_i^{\text{transfer}}$ 对目标域热门项目和非热门项目的隐向量进行约束, 如果 i 为热门项目, 则 $\boldsymbol{q}_i^{\text{transfer}}=\boldsymbol{q}_i^{(5)}$, 即以热门项目 i 基于 $\boldsymbol{R}^{(5)}$ 学习得到的隐向量作为约束. 如果 i 为非热门项目, 则 $\boldsymbol{q}_i^{\text{transfer}}=\boldsymbol{q}_i^{(5)F_2}$, 即以非热门项目 i 基于映射关系得到的隐向量作为约束. 因此, 本文通过式(17)的求解实现了知识融合.

值得注意的是, 式(17)中待求解的目标域用户和项目的隐向量维度与 $\boldsymbol{p}_u^{\text{transfer}}$ 和 $\boldsymbol{q}_i^{\text{transfer}}$ 相同, 也就是与目标域活跃用户和热门项目的隐向量维度相同. 因此, 式(17)中目标域用户和项目的隐向量维度不是可调参数, 而是固定的. 本文使用 Adam 优化器求解式(17), 上述添加约束的图卷积矩阵补全方法被称

为受限图卷积矩阵补全方法，其算法描述将在 2.6 节给出.

2.6 算法描述

本节我们给出受限图卷积矩阵补全(restricted graph convolutional matrix completion, Restricted-GC-MC)算法和 CRCRCF 的完整描述，分别见算法 2 和算法 3.

算法 2. Restricted-GC-MC 算法.

输入：目标域评分矩阵 $\boldsymbol{R}^{(5)}$，目标域用户-商品二部图 $G^{(5)}$，目标域全体用户和全体项目节点 one-hot 表示的初始特征矩阵 $\boldsymbol{X}_U^{(5)}$ 和 $\boldsymbol{X}_I^{(5)}$，目标域各评分等级对应的邻接矩阵 $\boldsymbol{M}_r \in \{0,1\}^{m\times n}, r=1,2,3,4,5$，正则化参数 λ，可迁移量 $\boldsymbol{p}_u^{\text{transfer}}(u=u_1,u_2,\cdots,u_m)$ 和 $\boldsymbol{q}_i^{\text{transfer}}(i=i_1,i_2,\cdots,i_n)$；

输出：训练后的编码器 $\bar{f}^{(5)}(\cdot)$ 和解码器 $\bar{g}^{(5)}(\cdot)$，用户隐特征 $\bar{\boldsymbol{p}}_u^{(5)}$，项目隐特征 $\bar{\boldsymbol{q}}_i^{(5)}$.

① 随机初始化编码器 $\bar{f}^{(5)}(\cdot)$ 和解码器 $\bar{g}^{(5)}(\cdot)$；

② $[\bar{\boldsymbol{p}}_u^{(5)},\bar{\boldsymbol{q}}_i^{(5)}]=f^{(5)}(\boldsymbol{X}_U^{(5)},\boldsymbol{X}_I^{(5)},\boldsymbol{M}_1,\boldsymbol{M}_2,\boldsymbol{M}_3,\boldsymbol{M}_4,\boldsymbol{M}_5)$；

③ $J_1=-\sum_{(u,i)}\sum_{r=1}^{5} I\left[r_{ui}^{(5)}=r\right]\lg p\left(\hat{r}_{ui}^{(5)}=r\right)+\lambda\left(\left\|\bar{\boldsymbol{q}}_i^{(5)}-\boldsymbol{q}_i^{\text{transfer}}\right\|^2+\left\|\bar{\boldsymbol{p}}_u^{(5)}-\boldsymbol{p}_u^{\text{transfer}}\right\|^2\right)$；

④ Repeat;

⑤ $[\bar{\boldsymbol{p}}_u^{(5)},\bar{\boldsymbol{q}}_i^{(5)}]=\bar{f}^{(5)}(\boldsymbol{X}_U^{(5)},\boldsymbol{X}_I^{(5)},\boldsymbol{M}_1,\cdots,\boldsymbol{M}_5)$；

⑥ for each $r_{ui}^{(5)}$ in $\boldsymbol{R}^{(5)}$

⑦ $\hat{r}_{ui}^{(5)}=\bar{g}^{(5)}\left(\bar{\boldsymbol{p}}_u^{(5)},\bar{\boldsymbol{q}}_i^{(5)}\right)$；

⑧ end for

⑨ $J_2=-\sum_{(u,i)}\sum_{r=1}^{5} I\left[r_{ui}^{(5)}=r\right]\lg p\left(\hat{r}_{ui}^{(5)}=r\right)+\lambda\left(\left\|\bar{\boldsymbol{q}}_i^{(5)}-\boldsymbol{q}_i^{\text{transfer}}\right\|^2+\left\|\bar{\boldsymbol{p}}_u^{(5)}-\boldsymbol{p}_u^{\text{transfer}}\right\|^2\right)$；

⑩ $\Delta J=|J_2-J_1|$；

⑪ $J_1=J_2$；

⑫ 更新 $\bar{f}^{(5)}(\cdot)$ 和 $\bar{g}^{(5)}(\cdot)$ 的参数；

⑬ until ($\Delta J<\delta=10^{-3}$ 或 $\tau>T_{\max}=100$)；

⑭ $[\bar{\boldsymbol{p}}_u^{(5)},\bar{\boldsymbol{q}}_i^{(5)}]=\bar{f}^{(5)}(\boldsymbol{X}_U^{(5)},\boldsymbol{X}_I^{(5)},\boldsymbol{M}_1,\cdots,\boldsymbol{M}_5)$；

⑮ return 训练后的编码器 $\bar{f}^{(5)}(\cdot)$ 和解码器 $\bar{g}^{(5)}(\cdot)$，用户隐特征 $\bar{\boldsymbol{p}}_u^{(5)}$，项目隐特征 $\bar{\boldsymbol{q}}_i^{(5)}$.

算法 3. CRCRCF 算法.

输入：目标域评分矩阵 $\boldsymbol{R}^{(5)}$，目标域用户-商品二部图 $G^{(5)}$，目标域全体用户和全体项目节点的初始特征矩阵分别为 $\boldsymbol{X}_U^{(5)}$ 和 $\boldsymbol{X}_I^{(5)}$，目标域各评分等级对应的邻接矩阵 $\boldsymbol{M}_r^{(5)}\in\{0,1\}^{m\times n}, r=1,2,3,4,5$，辅助域评分矩阵 $\boldsymbol{R}^{(2)}$，辅助域用户-商品二部图 $G^{(2)}$，辅助域全体用户和全体项目节点的初始特征矩阵分别为 $\boldsymbol{X}_U^{(2)}$ 和 $\boldsymbol{X}_I^{(2)}$，辅助域各评分等级对应的邻接矩阵 $\boldsymbol{M}_r^{(2)}\in\{0,1\}^{m\times n}$，$r=0,1$，用户活跃度阈值 μ_1，项目热门度阈值 μ_2，正则化参数 λ；

输出：预测的评分值 $\hat{r}_{ui}$.

① $[U_{\text{a}},U_{\text{ina}},I_{\text{p}},I_{\text{unp}}]=\textit{User-Item-Division}(\boldsymbol{R}^{(5)},\mu_1,\mu_2)$；

② $[\boldsymbol{p}_{u_{\text{a}}}^{(5)},\boldsymbol{q}_{i_{\text{p}}}^{(5)}]=\textit{GC-MC}(\boldsymbol{R}^{(5)},G^{(5)},\boldsymbol{M}_r^{(5)},\boldsymbol{X}_U^{(5)},\boldsymbol{X}_I^{(5)})$；

③ $[\boldsymbol{p}_{u_{\text{a}}}^{(2)},\boldsymbol{p}_{u_{\text{ina}}}^{(2)},\boldsymbol{q}_{i_{\text{p}}}^{(2)},\boldsymbol{q}_{i_{\text{unp}}}^{(2)}]=\textit{GC-MC}(\boldsymbol{R}^{(2)},G^{(2)},\boldsymbol{M}_r^{(2)},\boldsymbol{X}_U^{(2)},\boldsymbol{X}_I^{(2)})$；

④ $TS1=(\boldsymbol{p}_{u_{\text{a}}}^{(2)},\boldsymbol{p}_{u_{\text{a}}}^{(5)})$，$TS2=(\boldsymbol{q}_{i_{\text{p}}}^{(2)},\boldsymbol{q}_{i_{\text{p}}}^{(5)})$；/*构造监督学习的训练集*/

⑤ $F_1=\textit{STLDR}(\boldsymbol{p}_{u_{\text{ina}}}^{(2)},TS1)$，$F_2=\textit{STLDR}(\boldsymbol{q}_{i_{\text{unp}}}^{(2)},TS1)$；

⑥ $\boldsymbol{p}_{u_{\text{ina}}}^{(5)F_1}=F_1(\boldsymbol{p}_{u_{\text{ina}}}^{(2)})$，$\boldsymbol{q}_{i_{\text{unp}}}^{(5)F_2}=F_2(\boldsymbol{q}_{i_{\text{unp}}}^{(2)})$；

⑦ $\boldsymbol{p}_u^{\text{transfer}}=\begin{cases}\boldsymbol{p}_u^{(5)}, & u=u_{\text{a}}\\ \boldsymbol{p}_u^{(5)F_1}, & u=u_{\text{ina}}\end{cases}$，$\boldsymbol{q}_i^{\text{transfer}}=\begin{cases}\boldsymbol{q}_i^{(5)}, & i=i_{\text{p}}\\ \boldsymbol{q}_i^{(5)F_2}, & i=i_{\text{unp}}\end{cases}$；

⑧ $[\bar{f}^{(5)}(\cdot),\bar{g}^{(5)}(\cdot),\bar{\boldsymbol{p}}_u^{(5)},\bar{\boldsymbol{q}}_i^{(5)}]=\textit{Restricted-GC-MC}(\boldsymbol{R}^{(5)},G^{(5)},\boldsymbol{M}_r^{(5)},\boldsymbol{X}_U^{(5)},\boldsymbol{X}_I^{(5)},\lambda,\boldsymbol{p}_u^{\text{transfer}},\boldsymbol{q}_i^{\text{transfer}})$；

⑨ return $\hat{r}_{ui}=\bar{g}^{(5)}(u,i)$.

2.7 时间和空间复杂度分析

1)算法的时间复杂度分析

假设迭代次数为 K，目标域评分数为 q，辅助域评分数为 $\tilde{q}$，用户数和项目数分别为 M 和 N，图卷积层输出的隐向量维度 k，线性层输出的隐向量维度为 d. 本文 CRCRCF 算法可以分为 3 个模块：① 使用 GC-MC 获取目标域和辅助域中用户和项目隐向量；②基于深度回归网络的映射关系建模；③目标域评分矩阵的受限图卷积矩阵补全.

模块①中获取目标域和辅助域矩中用户隐向量和项目隐向量的时间复杂度分别为 $O(K(5k(M+N)(\bar{N}_{\text{T}}\times(M+N)+d)+5qd^3))$ 和 $O(K(2k(M+N)(\bar{N}_S(M+N)+d)+2\tilde{q}d^3))$，且目标域和辅助域的求解用户和项目隐向量过程是完全独立的，可以并行执行，故模块①的时间复杂度为 $O(K(k(M+N)(\max(5\bar{N}_{\text{T}},2\bar{N}_{\text{S}})(M+N)+d)+\max(5q,2\tilde{q})d^3))$，其中，$\bar{N}_{\text{T}}$ 和 $\bar{N}_{\text{S}}$ 为目标域和辅助域用户-商品图中节点的平均邻居数目. 模块②包含层数为 F 的用户侧和项目侧深度回归网络构建，其时间复杂度为 $O((M+N)\sum_{f=1}^{F-1}k_fk_{f+1})$，其中 k_f 表深度回归网络第 f 层的维度. 模块③受限于图卷积矩阵补全的时间复杂度与 GC-MC 算法相同，故本文 CRCRCF 算法的时间复杂度为 $O(K(k(M+N)(\max(5\bar{N}_{\text{T}},2\bar{N}_S)(M+N)+d)+\max(5q,2\tilde{q})d^3+(M+N)\sum_{f=1}^{F-1}k_fk_{f+1})+5k(M+N)(\bar{N}_{\text{T}}(M+N)+d)+5qd^3)$.

2)算法的空间复杂度分析

算法所需存储空间包括输入数据所需空间和算法参数所需空间. 算法输入数据包括用户和项目的

初始特征表示和评分数据, 这 2 部分所需的存储空间分别为 $O(M+N)^2$ 和 $O3(q+\tilde{q})$. CRCRCF 算法需要存储的参数变量包含 3 部分: ① 用户和项目的隐向量矩阵和 GC-MC 模型参数; ②基于自教学习的深度回归网络映射关系建模时的参数和映射后的用户和项目特征矩阵; ③ 限制图卷积矩阵补全模型参数和隐向量矩阵. 这 3 部分所占用的存储空间为: $O(2(M+N)d+7(MN+(M+N)k+Kd+d^2))$, $O(2\sum_{f=1}^{F-1}(k_f k_{f+1}+b_{f+1})+(M+N)d)$ 和 $O(5(MN+(M+N)k+Kd+d)+(M+N)d)$. 因此, CRCRCF 算法最终的空间复杂度为 $O(4(M+N)d+12(MN+(M+N)k+Kd+d^2)+2\sum_{f=1}^{F-1}(k_f k_{f+1}+b_{f+1}))$.

由以上分析可知, 基于 GC-MC 的隐向量提取导致 CRCRCF 算法时空复杂度较高. 本文的研究重点在于提高推荐性能, 如何在保证性能的同时降低时空复杂度将是后续工作的重点.

3 实 验

为充分评估 CRCRCF 的性能, 我们将该模型与当前多种经典的推荐算法进行对比. 我们选择 MovieLens10M 数据集和 Netflix 数据集进行广泛地实验, 实验运行的环境为 4.7 GHz, i7-10710U CPU, 16 GB RAM, 64 位 Windows 10. 实验中所涉及算法均使用 Python3.6 基于 2 个开源机器学习库 Scikit-learn 和 PyTorch 实现. 本实验主要讨论 3 个问题:

1)目标域中活跃用户和热门项目阈值对 CRCRCF 的性能是否有影响;

2)相比于其他仅结合辅助域信息的跨评分推荐模型, CRCRCF 还充分挖掘目标域活跃用户和热门项目相关的评分密集区域的评分信息, 是否可以有效提升目标域整体尤其是评分非密集区域的评分预测性能;

3)相比于 CRCRCF 不使用受限图卷积矩阵补全的算法变体 $\text{CRCRCF}_{\text{direct}}$ 以及仅利用少量活跃用户或热门项目在目标域和辅助域中的隐向量进行映射关系学习的算法变体 $\text{CRCRCF}_{\text{sv}}$, CRCRCF 是否更为有效.

3.1 对比方法

1)GC-MC[11]. 该模型是一种基于图神经网络的评分矩阵补全模型. 实验中设置图卷积层输出的用户和项目的隐向量维度 $k\in\{100, 300, 500, 700, 900\}$, 线性层输出的用户和项目的隐向量维度 $d\in\{30, 45, 60, 75, 90\}$, dropout 比例 $\rho\in\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$, 参考 GC-MC 原文, 设置图编码器层数 l=1. 模型参数的最优值通过交叉验证确定.

2)CSVD[6]. 该模型是 TCF 模型[6] 的一种变体. 实验中设置隐向量维度 $k\in\{5, 10, 15, 20, 25, 30, 35, 40\}$, 用户隐向量、项目隐向量和评分模式的正则项系数 $\alpha_u=\alpha_v=\beta\in\{0.001, 0.01, 0.1, 1, 10, 100, 1\,000\}$, 辅助域权重 $\lambda\in\{0.01, 0.1, 1\}$, 并通过交叉验证确定最优参数值.

3)TMF[8]. 该模型是一种基于混合分解的迁移模型. 实验中设置隐向量维度 $k\in\{5, 10, 15, 20, 25, 30, 35, 40\}$, 其余参数参考文献 [8] 进行设置, 即将正则项系数设为 0.01, 用户画像权重 $w_n=1$, $w_p\in\{1, 2, 3, 4, 5\}$, 辅助域权重 $\lambda=1$, 目标域和辅助域的交互系数 $\rho\in\{0.3, 0.4, 0.5, 0.6\}$.

4)DLSCF-S[10]. 该模型是一种深度低秩稀疏联合分解模型. 其参数取值参考文献 [10] 进行设置, 即分解层数 $l=2$, 第 1 层隐向量维度 $k_1=30$, 第 2 层隐向量维度 $k_2\in\{2,5,7,10,15,20\}$, 设置辅助域权重 $\lambda=1$, 评分模式系数 $\beta_1=\beta_2\in\{0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1\}$, 用户隐向量、项目隐向量的正则项系数 $\alpha_u=\alpha_v=1$.

5)EKT[9]. 该模型是一种考虑用户和项目几何结构图信息的矩阵分解模型. 实验中用户、项目的隐向量维度设置为 $k_1=k_2\in\{5, 10, 15, 20, 25, 30, 35, 40\}$. 其余参数参考文献 [9] 进行设置, 即辅助域权重 $\lambda=1$, 评分模式矩阵的正则项系数 $\alpha_s=\beta_s=\gamma_s=1$, 用户目标域和辅助域中隐向量的差值对应的正则项系数以及项目目标域和辅助域中隐向量的差值对应的正则项系数 $\theta_u=\theta_v\in\{0.1, 0.5, 1, 5, 10\}$, 目标域用户、项目的图正则项系数 $\alpha_u=\alpha_v\in\{0.01, 0.05, 0.1, 0.5, 1\}$, 辅助域用户、项目的图正则项系数 $\beta_u=\beta_v\in\{0.01, 0.05, 0.1, 0.5, 1\}$.

6)CRCRCF. 该模型是本文所提模型. 该模型提取隐向量的 GC-MC 子模块与 GC-MC 算法设置相同; 栈式降噪自编码器模型中的隐含层层数与节点个数依据实验中求得的隐向量维度值进行设置, 具体设置方法请见 3.4.3 节; 受限图卷积矩阵补全模型的正则项系数 $\lambda\in\{0.001, 0.01, 0.1, 1, 10, 100, 1\,000\}$, 其余参数与 GC-MC 设置相同.

7)$\text{CRCRCF}_{\text{direct}}$. 该模型是本文所提模型的变体, 其不使用受限图卷积矩阵补全, 而是将利用映射关系得到的非活跃用户和非热门项目的隐向量直接输入目标域 GC-MC 训练得到的双线性解码器中预测未知的评分. 提取隐向量的 GC-MC 子模块与 GC-MC 算法设置相同; 栈式降噪自编码器模型中的隐含

层层数与节点个数与 CRCRCF 算法相同.

8) $CRCRCF_{sv}$. 该模型是本文所提模型的变体，该模型仅对少量活跃用户或热门项目在目标域和辅助域中的隐向量进行有监督学习，获取活跃用户以及热门项目在目标域与辅助域上隐向量的映射关系. 提取隐向量的 GC-MC 子模块与 GC-MC 算法设置相同. 受限图卷积矩阵补全模型的正则项系数 $\lambda \in \{0.001, 0.01, 0.1, 1, 10, 100, 1\,000\}$.

3.2 评价指标

本实验采用平均绝对值误差(mean absolute error, *MAE*)和均方根误差(root mean squared error, *RMSE*)来衡量评分预测的精度，计算公式为：

$$MAE = \frac{1}{|T|}\sum_{(u,i)\in T}|r_{ui}-\hat{r}_{ui}|, \tag{19}$$

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{(u,i)\in T}(r_{ui}-\hat{r}_{ui})^2}, \tag{20}$$

其中 T 表示测试集样本集合，r_{ui} 和 $\hat{r}_{ui}$ 分别表示用户 u 对项目 i 的真实评分和预测评分. 通常，测试集上 *MAE* 和 *RMSE* 的值越小，推荐算法的评分预测精度越好.

3.3 数据准备

MovieLens10M(以下简称 ML10M)数据集包含 7.1×10^4 多个用户对 10^4 多个项目的超过 10^7 个评分数据，评分的取值范围为 {0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5}. Netflix 数据集包含 4.8×10^5 多个用户对 1.7×10^4 多个项目的超过 10^8 个评分数据，评分的取值范围为 {1, 2, 3, 4, 5}. 针对本文所讨论的推荐场景，要求实验数据满足：1)目标域和辅助域共享相同的用户集合和项目集合；2)辅助域整体 1, 0 评分比较密集，目标域整体{1,2,3,4,5}评分比较稀疏；3)部分用户对部分项目有 2 种格式的评分. 因此，参考文献 [6] 中数据处理方法，本实验中使用的数据集将按照 3 种方法进行数据处理：1)将原始数据集中的用户和项目按照评分个数分别排序，然后取评分数量最多的前 m 个用户和前 n 个项目的评分数据组成一个密集的评分矩阵 $\boldsymbol{R}$；2)将 $\boldsymbol{R}$ 中的评分数据分为 3 份，即 $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$，它们含有评分的比例 $N(\boldsymbol{A}_1):N(\boldsymbol{A}_2):N(\boldsymbol{A}_3)=1:1:18$；3)将 $\boldsymbol{A}_1$ 与 $\boldsymbol{A}_2$ 合并作为目标域评分矩阵 $\boldsymbol{R}^{(5)}$(针对 ML10M 时，目标域评分矩阵记为 $\boldsymbol{R}^{(10)}$)，将 $\boldsymbol{A}_2$ 与 $\boldsymbol{A}_3$ 合并，并将其中小于 4 分的评分置为 0(不喜欢)，大于等于 4 分的评分置为 1(喜欢)，以此作为辅助域评分矩阵 $\boldsymbol{R}^{(2)}$.

需要说明的是，对于目标域评分矩阵 $\boldsymbol{R}^{(10)}$，GC-MC 和 Restricted-GC-MC 设置 $r\in\{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$即可. 此外，划分 $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$ 时，对评分数据不采用随机抽取方式，以避免最终的目标域和辅助域评分矩阵出现空行与空列.

对照如上数据处理步骤和推荐场景要求，可以发现经过我们的数据处理后，目标域和辅助域的评分密度比为 2∶19，目标域评分较为稀疏，辅助域评分较为密集，即满足要求 2 中跨评分推荐场景实验数据要求；将 $\boldsymbol{A}_2$ 分别与 $\boldsymbol{A}_1$ 和 $\boldsymbol{A}_3$ 合并，构造辅助域和目标域，其目的是满足要求 3 中推荐场景中目标域和辅助域中部分用户对部分项目有 2 种格式评分的实验数据要求；此外，目标域和辅助域共享相同的用户集合和项目集合. 因此，经上述操作，实验数据满足要求. 最终实验数据集的统计信息如表 1 所示.

Table 1 Statistics of the Datasets

表 1 数据集统计信息

数据集	域名	用户数	项目数	评分格式	评分个数	评分密度/%
ML10M	目标域	5 000	5 000	[0.5, 5] 间隔为 0.5	253 673	1.01
	辅助域	5 000	5 000	{0, 1}	2 536 729	10.15
Netflix	目标域	3 000	3 000	[1, 5] 间隔为 1	55 024	0.61
	辅助域	3 000	3 000	{0, 1}	574 880	6.39

从上述 ML10M 和 Netflix 数据集中目标域评分矩阵的每行中分别选取 90%, 80%, 70%, 60% 的评分数据作为训练数据集，记为 TR_{90}, TR_{80}, TR_{70}, TR_{60}. 将剩下的 10%, 20%, 30%, 40% 作为测试数据集，记为 TE_{10}, TE_{20}, TE_{30}, TE_{40}. 从而可以构造 4 组不同的训练集和测试集，以更为充分地评估各种不同推荐算法的性能. 值得注意的是，为保证 GC-MC 算法顺利运行，选取训练集评分的过程中也要避免出现空行和空列.

3.4 实验过程说明

3.4.1 用户和项目划分

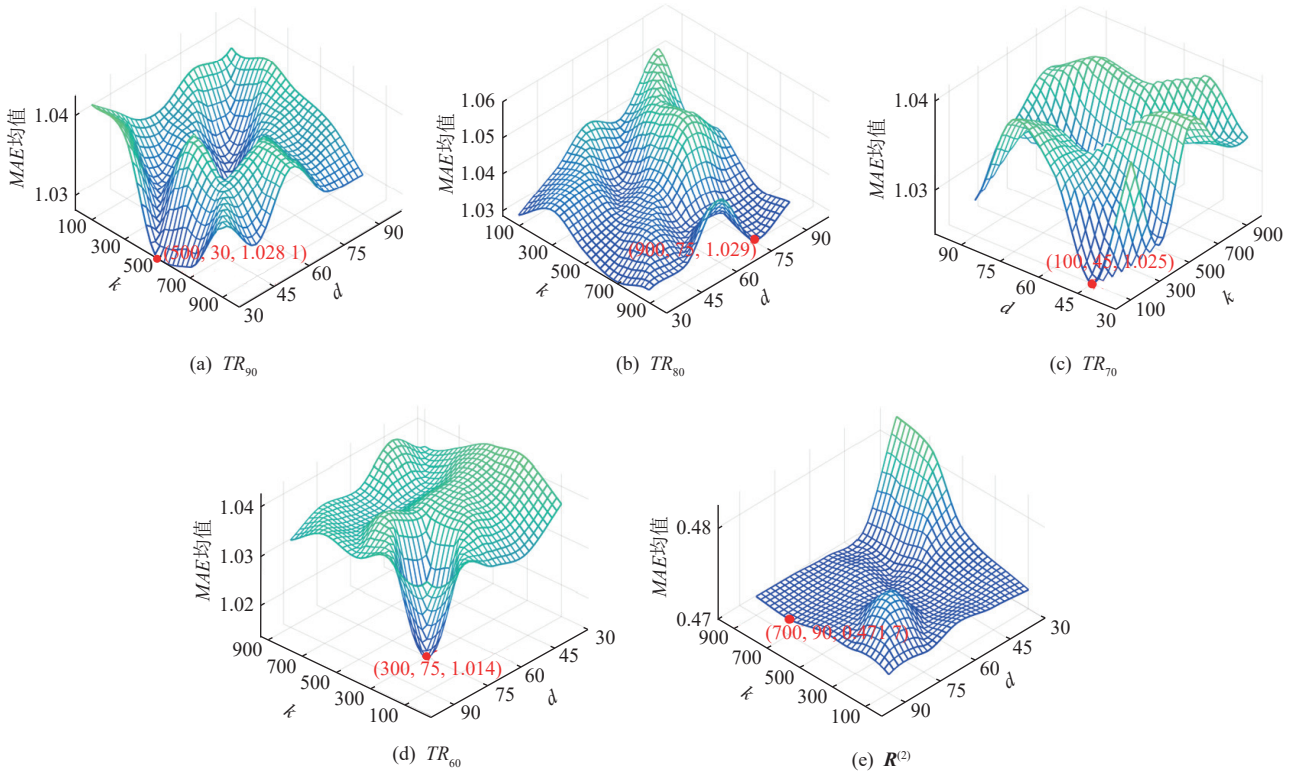
通过用户活跃度阈值来划分活跃用户和非活跃用户，阈值越小，则划分的活跃用户数量越少，即映射关系建模时监督训练样本数目越少，不利于映射关系的准确建模. 阈值越大，则活跃用户的评分密度越小，不利于隐向量的准确计算，也将影响映射关系建模的准确性. 因此，用户活跃度阈值过大或者过小均不利于模型学习. 同样，项目热门度阈值过大或者过小也不利于模型学习.

为了确定最优阈值，我们设定用户的活跃度阈值为 $\mu_1\in\{5\%, 10\%, 15\%, 20\%, 25\%, 30\%\}$，项目的热门度阈值为 $\mu_2\in\{5\%, 10\%, 15\%, 20\%, 25\%, 30\%\}$. 对于 ML10M 数据集，训练集是 $5\,000\times5\,000$ 的评分矩阵，不同阈值对应的活跃用户数和热门项目数分别是

250, 500, 750, 1 000, 1 250, 1500. 对于 Netflix 数据集，训练集是3 000×3 000的评分矩阵，不同阈值对应的活跃用户数和热门项目数分别是 150, 300, 450, 600, 750, 900.

3.4.2 提取用户和项目的隐向量

如 2.3 节所述，本文模型通过 GC-MC 算法分别获取活跃用户和热门项目在目标域和辅助域上的隐向量. GC-MC 模型中图卷积层输出的隐向量维度 k、线性层输出的隐向量维度 d、dropout 比例ρ的最优取值通过交叉验证来确定. 对于 ML10M 数据集，经交叉验证确认，目标域不同训练集 TR_{90}, TR_{80}, TR_{70}, TR_{60}和辅助域 $\boldsymbol{R}^{(2)}$上最优的 dropout 比例依次为 0.6, 0.6, 0.5, 0.7, 0.5, 不同参数k与d对应的 *MAE* 均值情况如图 6 所示，图 6 中标注点的前 2 维数值分别代表参数k和d的最优取值，第 3 维数值代表最优参数取值对应的模型在交叉验证过程中的 *MAE* 均值. 同样的，在 Netflix 数据集上也进行同样的操作，此处不再一一列出，仅在表 2 中给出最优参数组合.



(a) TR_{90} (b) TR_{80} (c) TR_{70}

(d) TR_{60} (e) $\boldsymbol{R}^{(2)}$

Fig. 6 Average values of *MAE* corresponding to different combinations of k and d for GC-MC algorithm

图 6 GC-MC 算法不同k和d组合对应的 *MAE* 均值

Table 2 Optimal Parameters Values of GC-MC for Netflix Dataset

表 2 在 Netflix 数据集上的 GC-MC 最优参数取值

参数	目标域				辅助域
	TR_{90}	TR_{80}	TR_{70}	TR_{60}	
ρ	0.6	0.7	0.5	0.6	0.6
d	45	45	45	45	80
k	100	500	500	300	700

3.4.3 构建映射关系模型

对于 ML10M 数据集，构建映射关系模型时，在自教学习框架下，首先利用数目众多的非活跃用户和非热门项目的隐向量分别计算用户隐向量和项目隐向量的低维高层特征表示. 根据 3.4.2 节，辅助域上隐因子向量维度(即线性层的输出维度)为 90, 故针对用户和项目的映射关系模型的输入向量维度均为 90, 即k_1=90. 采用栈式降噪自编码器获取输入隐向量的低维高层特征表示，将编码层的层数设置为 2. 设定 2 层编码层中的节点数目为：$k_2 \in \{35, 40, 45, 50, 55\}$，$k_3 \in \{10, 15, 20, 25, 30\}$，通过交叉验证来确定$k_2$, k_3的最优值. 以活跃用户隐向量映射关系建模为例，对应 5%, 10%, 15%, 20%, 25%, 30% 等不同活跃度阈值的栈式降噪自编码器算法参数k_2, k_3不同组合对应的交叉验证过程的 *MAE* 均值如图 7 所示，其中参数含义与图 6 类似.

然后在编码器基础上外接一层线性回归单元，利用少量对应活跃用户的有监督训练数据$\{\boldsymbol{p}_{u_a}^{(2)}, \boldsymbol{p}_{u_a}^{(5)}\}$

建模映射关系.

同样针对热门项目的隐向量映射关系建模情况的相关实验结果如图 8 所示，其最优参数情况和迭代收敛情况的分析与图 7 类似.

对于 Netflix 数据集，构建映射关系模型的方式与 ML10M 数据集一样，不同的是对于 Netflix 数据集，辅助域上用户和项目的隐向量维度为 80，故用户侧和项目侧映射关系建模的输入向量维度均为 80，即 k_1=80. 同样设定栈式降噪自编码器的编码层层数为 2，根据 k_1=80，设定 2 层编码层的节点个数为：$k_2 \in \{30, 35, 40, 45, 50\}$，$k_3 \in \{5, 10, 15, 20, 25\}$. 通过交叉验证来确定 k_2，k_3 的最优值. Netflix 数据集上参数调优情

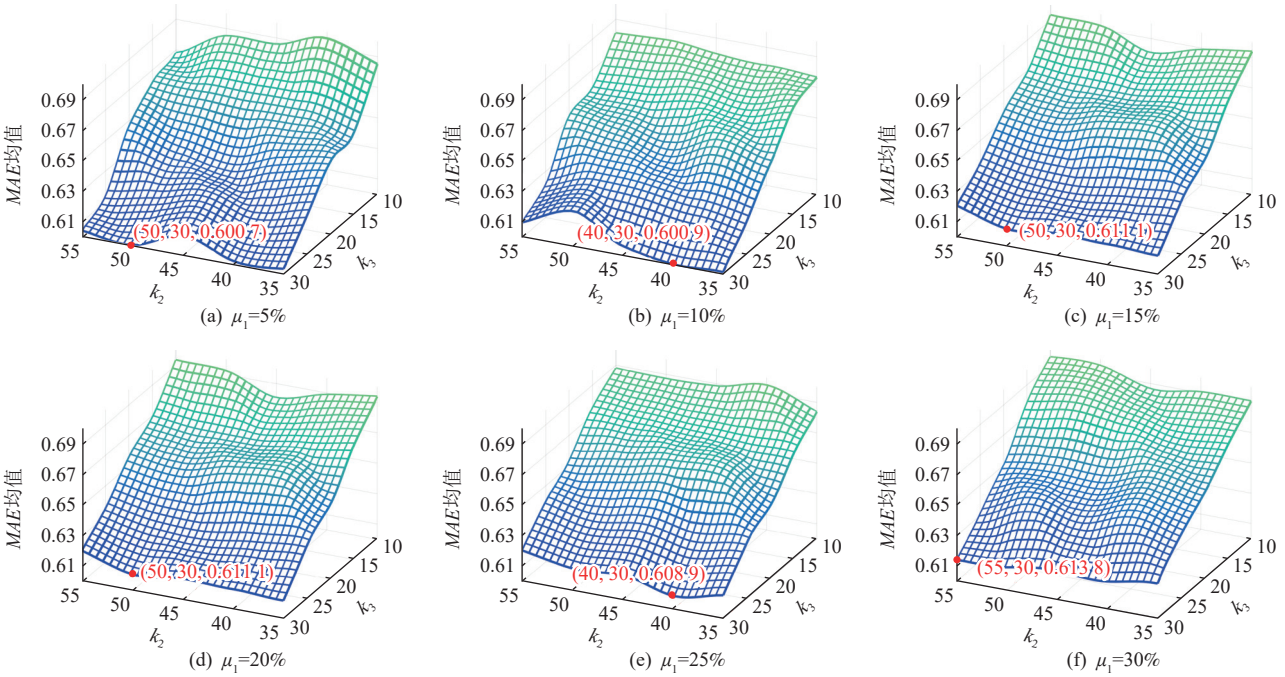


Fig. 7 Average values of *MAE* corresponding to different combinations of k_2 and k_3 for SDAE parameters on user-side

图 7 用户侧栈式降噪自编码器参数 k_2，k_3 不同组合对应的 *MAE* 均值

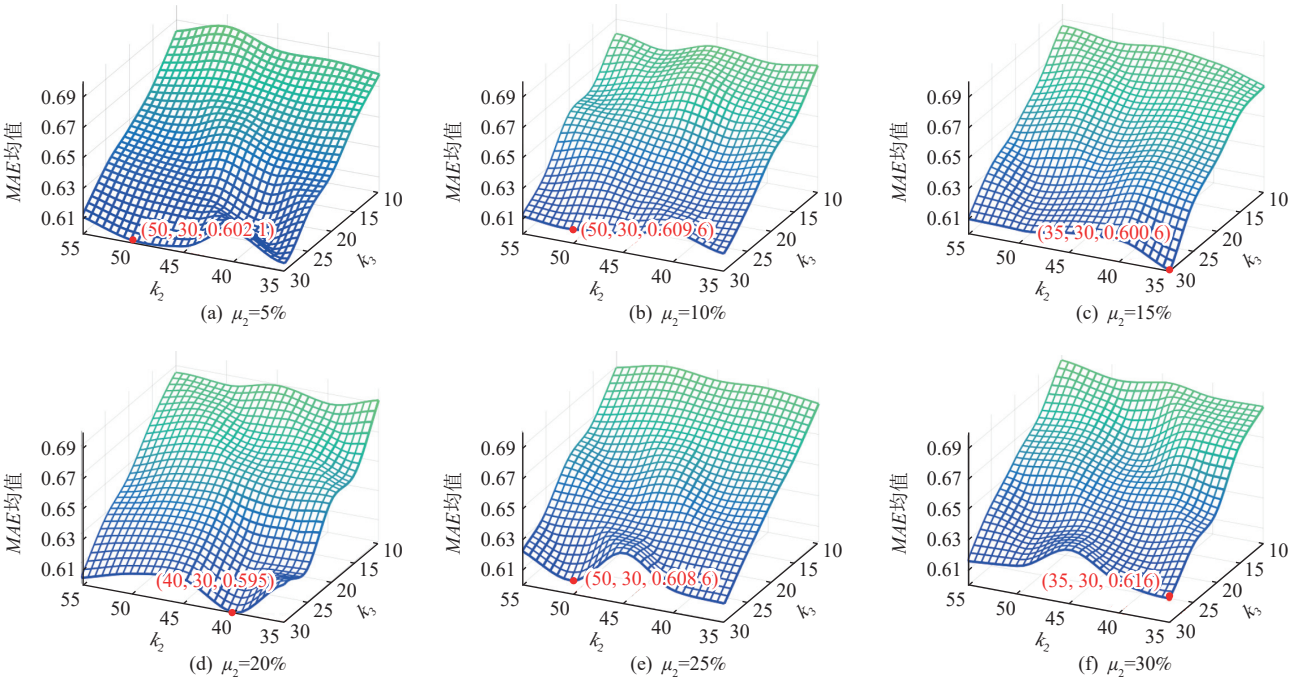


Fig. 8 Average values of *MAE* corresponding to different combinations of k_2 and k_3 for SDAE parameters on item-side

图 8 项目侧栈式降噪自编码器参数 k_2，k_3 不同组合对应的 *MAE* 均值

况与 ML10M 数据集情况相似, 为了节约篇幅, 此处仅给出最优参数取值, 如表 3 所示.

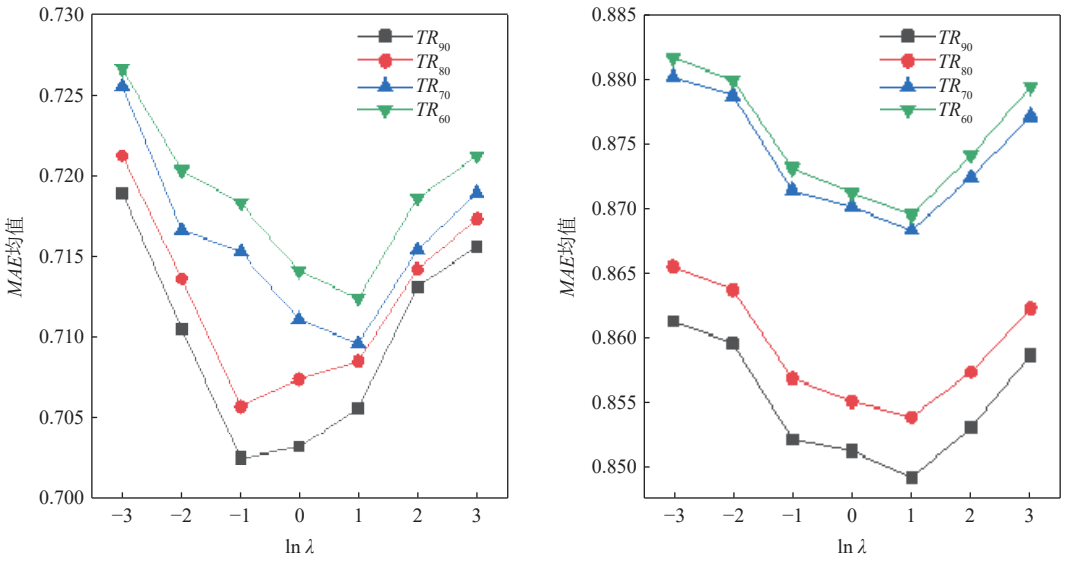
3.4.4 Restricted-GC-MC 训练

通过映射关系可以基于非活跃用户和非热门项目在辅助域的隐向量预测其在目标域中的隐向量, 从而以此为约束求解 Restricted-GC-MC 模型. 针对每一μ_1, μ_2组合, 利用TR_{90}, TR_{80}, TR_{70}, TR_{60}训练集训练 Restricted-GC-MC, 其中图卷积层输出的隐向量维度 k 和线性层输出的隐向量维度 d 以及 dropout 比例ρ的取值与 GC-MC 提取目标域隐特征时最优取值一致, 参数λ的最优值通过交叉验证确定. 为节约篇幅, 图 9 仅给出 ML10M 和 Netflix 数据集上, 当μ_1=μ_2=0.05 时, 4 种训练集上 Restricted-GC-MC 不同正则化系数$\lambda\in\{0.001, 0.01, 0.1, 1, 10, 100, 1\,000\}$对应的 *MAE* 均值.

Table 3 The Optimal Parameters of SDAE on User-Side and Item-Side with Different Thresholds for Netflix Dataset

表 3 Netflix 数据集不同阈值下用户侧和项目侧栈式降噪自编码器最优参数

维度	μ_1						μ_2					
	5%	10%	15%	20%	25%	30%	5%	10%	15%	20%	25%	30%
k_2	30	35	40	45	35	45	30	35	30	45	35	50
k_3	15	20	10	25	20	25	20	20	25	20	25	15



(a) ML10M (b) Netflix

Fig. 9 Average values of *MAE* corresponding to different λ for Restricted-GC-MC when μ_1 and μ_2 equal to 5%

图 9 μ_1=μ_2= 5% 时 Restricted-GC-MC 不同λ对应的 *MAE* 均值

3.4.5 不同阈值组合下 CRCRCF 算法在测试集上的效果

针对不同用户活跃度阈值和项目热门度阈值组合, 采用交叉验证方法确定的最优参数λ进行 Restricted-GC-MC 模型训练. 最终, 不同阈值组合下 CRCRCF 算法在 ML10M 和 Netflix 数据集上不同测试集上的实验结果如表 4 和表 5 所示.

因此, 针对本实验要讨论的第 1 个问题, 表 4 和表 5 的实验结果说明, 目标域中活跃用户和热门项目阈值对本文推荐算法的性能的确有影响. 从表 4 可以看出, 在 ML10M 数据集上, 当μ_1=5%, μ_2=20% 时, CRCRCF 算法在 4 种不同测试集上的 *MAE* 值具有最小均值 0.7027. 从表 5 中可以看出, 在 Netflix 数据集上, 当μ_1=25%, μ_2=5% 时, CRCRCF 算法具有最小均值 0.8416. μ_1=5%, μ_2=20% 和 μ_1=25%, μ_2=5% 是 2 种情况下 CRCRCF 算法的最优阈值组合.

3.4.6 对比其他推荐算法

当 CRCRCF 算法取最优活跃用户和热门项目阈值时, 我们将测试结果与单域协同过滤算法(如 GC-MC)和跨评分协同过滤推荐算法(包括 CSVD, TMF, DLSCF-S, EKT)进行对比, 表 6 和表 7 分别列出了 ML10M 数据集和 Netflix 数据集上, CRCRCF 和对比方法在不同测试集上的 *MAE* 值和 *RMSE* 值, 以及基于 *MAE* 值计算的 CRCRCF 与对比方法在显著性水平α=0.05 的双尾 t 检验下的 p 值. 表 8 和表 9 则分别列出了 ML10M 数据集和 Netflix 数据集评分非密集区域上不同算法的对比结果.

针对本实验要讨论的第 2 个问题: 1)相比于其他

Table 4 Values of *MAE* Corresponding to Different Activity Thresholds and Popularity Thresholds for ML10M Dataset

表 4 ML10M 数据集上不同活跃度阈值和热门度阈值对应的 *MAE* 值

μ_1	μ_2	*MAE*				*MAE* 均值	μ_1	μ_2	*MAE*				*MAE* 均值
		TE_{10}	TE_{20}	TE_{30}	TE_{40}				TE_{10}	TE_{20}	TE_{30}	TE_{40}	
5%	5%	0.695 3	0.701 1	0.702 8	0.701 7	0.700 2	20%	5%	0.693 9	0.701 2	0.704 5	0.705 6	0.701 3
5%	10%	0.696 9	0.718 8	0.729 9	0.744 3	0.722 5	20%	10%	0.697 1	0.720 2	0.729 8	0.726 8	0.718 5
5%	15%	0.696 5	0.700 1	0.703 6	0.735 9	0.709 0	20%	15%	0.696 6	0.700 6	0.703 8	0.708 6	0.702 4
5%	20%	0.698 1	0.699 8	0.702 9	0.733 3	0.708 5	20%	20%	0.699 3	0.692 6	0.701 3	0.700 3	0.698 4
5%	25%	0.700 1	0.699 5	0.702 8	0.745 6	0.712 0	20%	25%	0.697 1	0.699 2	0.700 1	0.706 8	0.700 8
5%	30%	0.696 2	0.699 2	0.701 3	0.706 1	0.700 7	20%	30%	0.691 2	0.696 8	0.698 3	0.705 5	0.698 0
10%	5%	0.695 2	0.700 6	0.703 8	0.704 1	0.700 9	25%	5%	0.687 1	0.706 6	0.702 6	0.705 5	0.700 5
10%	10%	0.695 3	0.718 9	0.730 1	0.751 2	0.723 9	25%	10%	0.689 8	0.720 2	0.731 1	0.728 9	0.717 5
10%	15%	0.698 6	0.702 2	0.703 8	0.708 6	0.703 3	25%	15%	0.699 5	0.702 1	0.703 9	0.728 6	0.708 5
10%	20%	0.701 1	0.699 5	0.703 2	0.699 8	0.700 9	25%	20%	0.702 8	0.703 5	0.715 1	0.726 9	0.712 1
10%	**25%**	**0.669 8**	**0.673 5**	**0.675 1**	**0.682 6**	**0.675 3**	25%	25%	0.700 8	0.701 9	0.717 7	0.728 2	0.712 2
10%	30%	0.680 1	0.688 7	0.686 6	0.697 8	0.688 3	25%	30%	0.698 7	0.702 6	0.708 9	0.719 8	0.707 5
15%	5%	0.698 6	0.703 3	0.703 8	0.704 6	0.702 6	30%	5%	0.692 5	0.699 5	0.724 9	0.727 2	0.711 0
15%	10%	0.695 6	0.718 9	0.730 2	0.766 9	0.727 9	30%	10%	0.699 1	0.719 3	0.729 7	0.728 9	0.719 3
15%	15%	0.698 5	0.701 2	0.699 2	0.706 3	0.701 3	30%	15%	0.700 1	0.698 6	0.703 1	0.708 2	0.702 5
15%	20%	0.697 8	0.697	0.699 3	0.705 1	0.699 8	30%	20%	0.701 1	0.695 6	0.711 5	0.725 3	0.708 4
15%	25%	0.700 3	0.699	0.702 2	0.710 1	0.702 9	30%	25%	0.702 6	0.700 1	0.705 8	0.708 8	0.704 3
15%	30%	0.701 7	0.700 8	0.702 9	0.708 9	0.703 6	30%	30%	0.708 8	0.700 1	0.705 6	0.708 6	0.705 8

注：黑体数值表示在 ML10M 数据集上最优阈值组合结果.

Table 5 Values of *MAE* Corresponding to Different Activity Thresholds and Popularity Thresholds for Netflix Dataset

表 5 Netflix 数据集上不同活跃度阈值和热门度阈值对应的 *MAE* 值

μ_1	μ_2	*MAE*				*MAE* 均值	μ_1	μ_2	*MAE*				*MAE* 均值
		TE_{10}	TE_{20}	TE_{30}	TE_{40}				TE_{10}	TE_{20}	TE_{30}	TE_{40}	
5%	5%	0.834 3	0.839 6	0.862 1	0.881 1	0.854 3	20%	5%	0.808 5	0.810 1	0.828 6	0.830 5	0.819 4
5%	10%	0.832 8	0.836 9	0.869 9	0.865 5	0.851 3	**20%**	**10%**	**0.802 6**	**0.803 5**	**0.820 2**	**0.824 5**	**0.812 7**
5%	15%	0.828 8	0.840 3	0.869 8	0.868 3	0.851 8	20%	15%	0.834 5	0.831 5	0.839 8	0.844 6	0.837 6
5%	20%	0.828 1	0.838 8	0.867 9	0.885 6	0.855 1	20%	20%	0.835 6	0.833 7	0.840 6	0.841 9	0.838 0
5%	25%	0.837 2	0.836 8	0.840 3	0.891 9	0.851 6	20%	25%	0.832 9	0.835 3	0.840 5	0.847 8	0.839 1
5%	30%	0.835 1	0.832 6	0.878 8	0.896 8	0.860 8	20%	30%	0.821 9	0.834 7	0.830 2	0.841 8	0.832 2
10%	5%	0.830 1	0.832 2	0.835 9	0.882 6	0.845 2	25%	5%	0.830 3	0.832 9	0.837 7	0.832 6	0.833 4
10%	10%	0.828 1	0.830 6	0.839 9	0.887	0.846 4	25%	10%	0.832 6	0.835 6	0.838 6	0.857 8	0.841 2
10%	15%	0.830 6	0.832 9	0.878 2	0.869 9	0.852 9	25%	15%	0.831 3	0.836 9	0.840 3	0.860 9	0.842 4
10%	20%	0.834 9	0.831 1	0.840 9	0.861 8	0.842 2	25%	20%	0.830 1	0.829 8	0.837 7	0.849 9	0.836 9
10%	25%	0.832 6	0.834 5	0.859 1	0.868 7	0.848 7	25%	25%	0.833 6	0.836 8	0.840 2	0.858 7	0.842 3
10%	30%	0.833 5	0.833 8	0.836 9	0.865 1	0.842 3	25%	30%	0.832 4	0.835 9	0.837 1	0.852 4	0.839 5
15%	5%	0.827 6	0.835 2	0.834 9	0.840 1	0.834 5	30%	5%	0.834 9	0.833 6	0.838 2	0.880 9	0.846 9
15%	10%	0.830 3	0.834 6	0.838 6	0.834 3	0.834 5	30%	10%	0.832 1	0.835 9	0.841 1	0.886 3	0.848 9
15%	15%	0.830 6	0.832 9	0.839 9	0.846 8	0.837 6	30%	15%	0.831 2	0.835 5	0.840 1	0.886 9	0.848 4
15%	20%	0.829 3	0.829 7	0.842 1	0.829 5	0.832 7	30%	20%	0.831 1	0.832 6	0.840 3	0.871 7	0.843 9
15%	25%	0.833 5	0.832 9	0.838 6	0.839 8	0.836 2	30%	25%	0.834 6	0.829 8	0.838 7	0.893 3	0.849 1
15%	30%	0.827 8	0.835 3	0.838 8	0.860 1	0.840 5	30%	30%	0.836 7	0.847 7	0.860 9	0.897 1	0.860 6

注：黑体数值表示在 Netflix 数据集上最优阈值组合结果.

仅结合辅助域信息的跨评分推荐模型, CRCRCF 是否充分挖掘目标域活跃用户和热门项目相关的评分密集区域的评分信息; 2)是否可以有效提升目标域整体尤其是评分非密集区域的评分预测性能. 从表 6 和表 7 可以看出, CRCRCF 在 2 个数据集的 *MAE* 和 *RMSE* 指标上比 CSVD, TMF, DLSCF-S, EKT 等算法表现更好. 在显著性水平 α=0.05 下, 根据计算的 p 值, CRCRCF 的性能在 2 个数据集上均显著优于其他对比方法.

更进一步, 从表 8 和表 9 可以看出, 针对目标域

Table 6 *MAE* and *RMSE* Values of Different Algorithms on ML10M Dataset

表 6 ML10M 数据集上不同算法的 *MAE* 和 *RMSE* 值

算法	*MAE*				*RMSE*				p 值
	TE_{10}	TE_{20}	TE_{30}	TE_{40}	TE_{10}	TE_{20}	TE_{30}	TE_{40}	
GC-MC	0.780 7	0.781 9	0.791 3	0.796 5	0.997 1	0.999 4	1.009 8	1.014 6	0.004 8
CSVD	0.710 1	0.718 0	0.728 5	0.729 4	0.911 0	0.918 9	0.931 8	0.935 9	0.003 9
TMF	0.720 7	0.724 0	0.733 7	0.738 7	0.927 2	0.929 0	0.941 4	0.947 5	0.003 1
DLSCF-S	0.706 9	0.710 5	0.718 1	0.718 6	0.906 3	0.908 4	0.917 8	0.919 5	0.004 9
EKT	0.714 7	0.717 4	0.723 8	0.726 0	0.914 7	0.918 2	0.921 9	0.931 3	0.004 0
$CRCRCF_{sv}$	0.726 6	0.729 3	0.738 2	0.740 2	0.930 1	0.932 8	0.945 9	0.951 7	0.002 7
$CRCRCF_{direct}$	0.713 3	0.719 2	0.730 1	0.729 9	0.915 1	0.922 3	0.933 9	0.937 7	0.003 7
CRCRCF (本文)	**0.669 8**	**0.673 5**	**0.675 1**	**0.682 6**	**0.860 7**	**0.875 6**	**0.879 2**	**0.886 3**	

注: 黑体数值表示在 ML10M 数据集上最优的性能指标数据, 下划线数字表示次优的性能指标数据.

Table 7 *MAE* and *RMSE* Values of Different Algorithms on Netflix Dataset

表 7 Netflix 数据集上不同算法的 *MAE* 和 *RMSE* 值

算法	*MAE*				*RMSE*				p 值
	TE_{10}	TE_{20}	TE_{30}	TE_{40}	TE_{10}	TE_{20}	TE_{30}	TE_{40}	
GC-MC	0.903 7	0.910 8	0.911 3	0.934 7	1.121 8	1.136 2	1.138 3	1.166 9	0.006 9
CSVD	0.846 2	0.852 2	0.857 4	0.865 6	1.065 1	1.072 8	1.075 6	1.088 5	0.003 8
TMF	0.874 0	0.877 6	0.882 5	0.900 5	1.098 2	1.110 6	1.117 7	1.140 2	0.001 6
DLSCF-S	0.841 3	0.845 1	0.849 1	0.861 7	1.053 3	1.062 6	1.065 7	1.080 2	0.004 5
EKT	0.843 8	0.851 1	0.852 6	0.863 4	1.058 7	1.069 9	1.069 7	1.085 7	0.004 1
$CRCRCF_{sv}$	0.878 2	0.881 5	0.887 6	0.902 8	1.103 6	1.121 9	1.126 5	1.143 1	0.001 4
$CRCRCF_{direct}$	0.849 8	0.857 1	0.859 9	0.868 2	1.069 4	1.080 6	1.081 2	1.092 1	0.003 4
CRCRCF (本文)	**0.802 6**	**0.803 5**	**0.820 2**	**0.824 5**	**1.006 2**	**1.007 8**	**1.023 9**	**1.027 6**	

注: 黑体数值表示在 Netflix 数据集上最优的性能指标数据, 下划线数字表示次优的性能指标数据.

Table 8 *MAE* and *RMSE* Values of Different Algorithms on Non-Rating-Dense Region of ML10M Dataset

表 8 ML10M 数据集评分非密集区域上不同算法的 *MAE* 和 *RMSE* 值

算法	*MAE*				*RMSE*				p 值
	TE_{10}	TE_{20}	TE_{30}	TE_{40}	TE_{10}	TE_{20}	TE_{30}	TE_{40}	
GC-MC	0.819 8	0.820 9	0.829 2	0.841 5	1.039 1	1.044 2	1.049 7	1.065 9	0.002 1
CSVD	0.743 4	0.745 3	0.757 2	0.767 0	0.952 8	0.954 3	0.968 2	0.984 5	0.002 3
TMF	0.750 0	0.761 2	0.777 1	0.778 1	0.954 7	0.977 4	0.998 1	1.025 5	0.001 5
DLSCF-S	0.735 5	0.740 1	0.749 9	0.754 5	0.937 9	0.946 0	0.958 5	0.974 2	0.003 0
EKT	0.739 9	0.741 7	0.755 1	0.762 3	0.949 3	0.95	0.965 2	0.980 3	0.002 6
$CRCRCF_{sv}$	0.758 8	0.769 5	0.780 9	0.785 1	0.962 8	0.985 6	1.004 9	1.034 5	0.001 2
$CRCRCF_{direct}$	0.746 6	0.743 7	0.760 1	0.769 9	0.957 2	0.951 3	0.972 1	0.988 7	0.002 2
CRCRCF (本文)	**0.679 5**	**0.684 6**	**0.685 2**	**0.693 1**	**0.878 9**	**0.880 1**	**0.902 5**	**0.913 3**	

注: 黑体数值表示在 ML10M 数据集评分非密集区域上最优的性能指标数据, 下划线数字表示次优的性能指标数据.

Table 9 *MAE* and *RMSE* Values of Different Algorithms on Non-Rating-Dense Region of Netflix Dataset

表 9 Netflix 数据集评分非密集区域上不同算法的 *MAE* 和 *RMSE* 值

算法	*MAE*				*RMSE*				*p* 值
	TE_{10}	TE_{20}	TE_{30}	TE_{40}	TE_{10}	TE_{20}	TE_{30}	TE_{40}	
GC-MC	0.935 1	0.945 7	0.976 8	1.031 1	1.194 8	1.197 6	1.339 4	1.356 1	0.003 3
CSVD	0.895 6	0.897 7	0.909 0	0.915 3	1.113 7	1.118 1	1.129 3	1.136 4	0.002 7
TMF	0.897 7	0.910 0	0.924 5	0.932 1	1.129 0	1.147 0	1.147 7	1.150 8	0.001 9
DLSCF-S	0.888 4	0.894 6	0.904 7	0.911 7	1.110 2	1.116 5	1.125 4	1.132 1	0.003 0
EKT	0.889 4	0.896 3	0.908 1	0.913 1	1.112 0	1.116 7	1.128 4	1.134 1	0.002 9
$CRCRCF_{sv}$	0.902 9	0.920 1	0.933 3	0.941 2	1.136 8	1.160 8	1.156 7	1.160 1	0.001 5
$CRCRCF_{direct}$	0.902 2	0.908 9	0.920 5	0.923 8	1.122 1	1.131 2	1.142 1	1.135 5	0.002 1
CRCRCF（本文）	**0.822 1**	**0.829 3**	**0.838 0**	**0.842 9**	**1.051 1**	**1.057 2**	**1.062 9**	**1.068 3**	

注：黑体数值表示在 Netflix 数据集评分非密集区域上最优的性能指标数据，下划线数字表示次优的性能指标数据.

评分非密集区域$d^{(5)}$, CRCRCF 在 *MAE* 和 *RMSE* 指标上比 CSVD, TMF, DLSCF-S, EKT 等算法具有更为明显的优势. 根据计算的 *p* 值, CRCRCF 的性能在 2 个数据集的评分非密集区域上均显著优于其他对比方法.

此外, 从表 6~9 中可以看出, 其他 4 种跨评分协同过滤推荐算法, 如 CSVD, TMF, DLSCF-S, EKT, 推荐性能均好于单域推荐算法 GC-MC. 这主要是因为跨评分协同过滤推荐算法能够迁移辅助域上挖掘到的有价值知识到目标域, 有效缓解目标域上数值评分的稀疏性难题.

上述实验结果说明, CRCRCF 能够充分挖掘辅助域较丰富的二元评分和目标域评分密集区域较为丰富的数值评分, 针对目标域不同区域制定个性化的知识迁移策略, 进而生成更为准确的推荐.

针对本实验要讨论的第 3 个问题: 相比于 CRCRCF 不使用受限图卷积矩阵补全的算法变体 $CRCRCF_{direct}$ 以及仅利用少量活跃用户或热门项目在目标域和辅助域中的隐向量进行映射关系学习的算法变体 $CRCRCF_{sv}$, CRCRCF 是否更为有效. 从表 6~9 中可以看出, 相比于 $CRCRCF_{direct}$ 与 $CRCRCF_{sv}$ 算法, CRCRCF 同样在 *MAE*, *RMSE* 指标上具有明显的优势. 根据计算的 *p* 值, CRCRCF 的性能在 2 个数据集整体尤其是评分非密集区域上均显著优于 $CRCRCF_{direct}$ 和 $CRCRCF_{sv}$, 其主要原因在于, 相对于 $CRCRCF_{direct}$, CRCRCF 将利用映射关系得到的非活跃用户和非热门项目的隐向量作为约束条件, 可以更为合理地融合目标域数据和辅助域数据, 而 $CRCRCF_{direct}$ 忽视了目标域数据的价值; 相对于 $CRCRCF_{sv}$, CRCRCF 基于自教学习范式有效利用了大量非活跃用户或非热门项目在辅助域中的隐向量, 可以更精准地建模映射关系, 而 $CRCRCF_{sv}$ 仅利用少量活跃用户或热门项目在目标域和辅助域中的隐向量进行映射关系学习, 无法精准建模映射关系.

4 结 论

本文提出了一种基于迁移学习思想的跨区域跨评分的协同过滤推荐算法(CRCRCF). CRCRCF 首先针对活跃用户和热门项目分别建模了辅助域和目标域的隐向量间的映射关系, 然后将活跃用户和热门项目的映射关系泛化到全局, 依次实现了跨区域映射关系迁移和跨评分的隐向量信息迁移. 最后设计了一种限制图卷积矩阵补全模型实现了目标域信息和辅助域信息的有效融合. CRCRCF 可以同时实现辅助域向目标域不同区域的个性化知识迁移, 以及目标域密集区域向非密集区域的有效知识迁移. 据我们所知, CRCRCF 是首个基于跨区域和跨评分 2 种思想的推荐算法, 通过有效利用辅助域和目标域评分密集区域的信息, CRCRCF 可以针对目标域整体, 尤其是评分非密集区域进行更为准确的评分预测. 在 MovieLens 和 Netflix 数据集上的对比实验说明 CRCRCF 较其他单域和跨评分推荐算法具有明显的优势. 关于后续工作, CRCRCF 仅针对评分信息建模, 如何充分利用用户或项目的多模态信息将是我们后续研究的重点.

作者贡献声明: 于旭提出论文的创新思路, 负责论文撰写与修改; 彭庆龙和詹定佳负责论文撰写; 杜军威和刘金环负责实验设计; 林俊宇负责算法实现; 巩敦卫负责论文修改; 张子迎负责数据准备与处理; 于婕负责论文修订.

参 考 文 献

[1] Anelli V W, Bellogín A, Noia T D, et al. Reenvisioning the comparison between neural collaborative filtering and matrix factorization[C]//Proc of the 15th ACM Conf on Recommender Systems. New York: ACM, 2021: 521−529

[2] Chen Biyi, Huang Ling, Wang Changdong, et al. Explicit and implicit feedback based collaborative filtering algorithm[J]. Journal of Software, 2020, 31(3): 794−805(in Chinese)
(陈碧毅, 黄玲, 王昌栋, 等. 融合显式反馈与隐式反馈的协同过滤推荐算法[J]. 软件学报, 2020, 31(3): 794−805)

[3] Du Min, Christensen R, Zhang Wei, et al. Pcard: Personalized restaurants recommendation from card payment transaction records[C]//Proc of the 28th World Wide Web Conf. New York: ACM, 2019: 2687−2693

[4] Gao Yuanning, Gao Xiaofeng, Li Xianyue, et al. An embedded GRASP-VNS based two-layer framework for tour recommendation[J]. IEEE Transactions on Services Computing, 2022, 15(2): 847−859

[5] Zhang Yujie, Dong Zheng, Meng Xiangwu. Research on personalized advertising recommendation systems and their applications[J]. Chinese Journal of Computers, 2021, 44(3): 531−563 (in Chinese)
(张玉洁, 董政, 孟祥武. 个性化广告推荐系统及其应用研究[J]. 计算机学报, 2021, 44(3): 531−563)

[6] Pan Weike, Liu N N, Xiang W E, et al. Transfer learning to predict missing ratings via heterogeneous user feedbacks[C]//Proc of the 22nd Int Joint Conf on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 2011: 2318−2323

[7] Pan Weike, Ming Zhong. Interaction-rich transfer learning for collaborative filtering with heterogeneous user feedback[J]. IEEE Intelligent Systems, 2014, 29(6): 48−54

[8] Pan Weike, Xia Shanchuan, Liu Zhuode, et al. Mixed factorization for collaborative recommendation with heterogeneous explicit feedbacks[J]. Information Sciences, 2016, 332(C): 84−93

[9] Zhang Hongwei, Kong Xiangwei, Zhang Yujia. Enhanced knowledge transfer for collaborative filtering with multi-source heterogeneous feedbacks[J]. Multimedia Tools and Applications, 2021, 80(16): 24245−24270

[10] Jiang Shuhui, Ding Zhengming, Fu Yun. Heterogeneous recommendation via deep low-rank sparse collective factorization[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 42(5): 1097−1111

[11] Berg R, Kipf T N, Welling M. Graph convolutional matrix completion[C]//Proc of the 24th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2018: 974−983

[12] Raina R, Battle A J, Lee H, et al. Self-taught learning: Transfer learning from unlabeled data[C]//Proc of the 24th Int Conf on Machine Learning. New York: ACM, 2007: 759−766

[13] Li Bin, Yang Qiang, Xue Xiangyang. Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction[C]//Proc of the 21st Int Joint Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2009: 2052−2057

[14] Li Bin, Yang Qiang, Xue Xiangyang. Transfer learning for collaborative filtering via a rating-matrix generative model[C]//Proc of the 26th Int Conf on Machine Learning. New York: ACM, 2009: 617−624

[15] Zhang Qian, Hao Peng, Lu Jie, et al. Cross-domain recommendation with semantic correlation in tagging systems[C/OL]//Proc of the 26th Int Joint Conf on Neural Networks. Piscataway, NJ: IEEE, 2019 [2023-10-18].https://doi.org/10.1109/IJCNN.2019.8852049

[16] Li Yakun, Ren Jiadong, Liu Jiaomin, et al. Deep sparse autoencoder prediction model based on adversarial learning for cross-domain recommendations[J]. Knowledge-Based Systems, 2021, 220: 106948

[17] Jiang Meng, Cui Peng, Yuan J N, et al. Little is much: Bridging cross-platform behaviors through overlapped crowds[C]//Proc of the 30th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2016: 13−19

[18] Zhang Qian, Lu Jie, Wu Dianshuang, et al. A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 30(7): 1998−2012

[19] Zhu Feng, Wang Yan, Chen Chaochao, et al. A graphical and attentional framework for dual-target cross-domain recommendation[C]//Proc of the 29th Int Joint Conf on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 2020: 3001−3008

[20] Li Pan, Tuzhilin A. Dual metric learning for effective and efficient cross-domain recommendations[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 35(1): 321−334

[21] Berkovsky S, Kuflik T, Ricci F. Cross-domain mediation in collaborative filtering[C]//Proc of the 11th Int Conf on User Modeling Conf. Berlin: Springer, 2007: 355−359

[22] Resnick P, Iacovou N, Suchak M, et al. Grouplens: An open architecture for collaborative filtering of netnews[C]//Proc of the ACM Conf on Computer Supported Cooperative Work. New York: ACM, 1994: 175−186

[23] Singh A P, Gordon G J. Relational learning via collective matrix factorization[C]//Proc of the 14th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2008: 650−658

[24] Hu Liang, Cao Jian, Xu Guandong, et al. Personalized recommendation via cross-domain triadic factorization[C]//Proc of the 22nd Int Conf on World Wide Web. New York: ACM, 2013: 595−606

[25] Loni B, Shi Yue, Larson M A, et al. Cross-domain collaborative filtering with factorization machines[C]//Proc of the 36th European Conf on Information Retrieval. Berlin: Springer, 2014: 656−661

[26] Yuan Feng, Yao Lina, Benatallah B. DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns[C]//Proc of the 28th Int Joint Conf on Artificial Intelligence.

San Francisco, CA: Morgan Kaufmann, 2019: 4227−4233

[27] Yu Xu, Chu Yan, Jiang Feng, et al. SVMs classification based two-side cross domain collaborative filtering by inferring intrinsic user and item features[J]. Knowledge-Based Systems, 2018, 141: 80−91

[28] Yu Xu, Jiang Feng, Du Junwei, et al. A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains[J]. Pattern Recognition, 2019, 94: 96−109

[29] Pan Weike, Xiang E A, Liu N N, et al. Transfer learning in collaborative filtering for sparsity reduction[C]//Proc of the 24th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2010: 230−235

[30] Yu Xu, Zhan Dingjia, Liu Lei, et al. A privacy-preserving cross-domain healthcare wearables recommendation algorithm based on domain-dependent and domain-independent feature fusion[J]. IEEE Journal of Biomedical and Health Informatics, 2022, 26(5): 1928−1936

[31] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint, arXiv: 1412.6980, 2015

[32] Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion[J]. Journal of Machine Learning Research, 2010, 11(12): 3371−3408

[33] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088): 533−536

Yu Xu, born in 1982. PhD, professor. Senior member of CCF. His main research interests include recommendation algorithms, transfer learning, and intelligent software engineering.

于　旭, 1982 年生. 博士, 教授. CCF 高级会员. 主要研究方向为推荐算法、迁移学习、智能软件工程.

Peng Qinglong, born in 1997. Master candidate. His main research interests include recommendation algorithms, transfer learning, and intelligent software engineering.

彭庆龙, 1997 年生. 硕士研究生. 主要研究方向为推荐算法、迁移学习、智能软件工程.

Zhan Dingjia, born in 1996. Master candidate. His main research interests include recommendation algorithms, transfer learning, and intelligent software engineering.

詹定佳, 1996 年生. 硕士研究生. 主要研究方向为推荐算法、迁移学习、智能软件工程.

Du Junwei, born in 1974. PhD, professor. Senior member of CCF. His main research interests include recommendation algorithms, transfer learning, and intelligent software engineering.

杜军威, 1974 年生. 博士, 教授. CCF 高级会员. 主要研究方向为推荐算法、迁移学习、智能软件工程.

Liu Jinhuan, born in 1990. PhD, master supervisor. Her main research interests include recommendation algorithms and information retrieval.

刘金环, 1990 年生. 博士, 硕士生导师. 主要研究方向为推荐算法、信息检索.

Lin Junyu, born in 1981. PhD, senior engineer. Distinguished member of CCF. His main research interests include artificial intelligence, content security, and ethics of science and technology.

林俊宇, 1981 年生. 博士, 高级工程师. CCF 杰出会员. 主要研究方向为人工智能、内容安全、科技伦理.

Gong Dunwei, born in 1970. PhD, professor. Distinguished member of CCF. His main research interests include intelligent optimization and control, and search-based software engineering.

巩敦卫, 1970 年生. 博士, 教授. CCF 杰出会员. 主要研究方向为智能优化与控制、基于搜索的软件工程.

Zhang Ziying, born in 1973. PhD, associate professor. His main research interests include pattern recognition, and artificial intelligence and its application in network security.

张子迎, 1973 年生. 博士, 副教授. 主要研究方向为模式识别、人工智能以及人工智能在网络安全中的应用.

Yu Jie, born in 1999. Master candidate. Her main research interests include recommendation algorithms, transfer learning, and intelligent software engineering.

于　婕, 1999 年生. 硕士研究生. 主要研究方向为推荐算法、迁移学习、智能软件工程.