冯祥,杨庆红. 结合知识图谱进行信息强化的协同过滤算法[J]. 江西师范大学学报(自然科学版),2022,46(4):386-393.
FENG Xiang,YANG Qinghong. The collaborative filtering algorithm for information enhancement combined with knowledge graph[J]. Journal of Jiangxi Normal University(Natural Science),2022,46(4):386-393.

文章编号:1000-5862(2022)04-0386-08

结合知识图谱进行信息强化的协同过滤算法

冯 祥,杨庆红*
(江西师范大学计算机信息工程学院,江西 南昌 330022)

摘要: 针对传统协同过滤算法存在使用信息单一、基础评分数据过于稀疏导致推荐效果不佳等问题,该文提出一种结合知识图谱进行信息强化的协同过滤(KGRI-CF)算法. 该算法利用电影的特征数据构建1张关于电影的知识图谱,对用户-评分矩阵进行有条件的填充,有效改善了传统协同过滤算法的数据稀疏性问题. 通过对评分数据进行统计与挖掘获取用户的偏好信息,构建了关于用户偏好的知识图谱. 利用实体向量化算法将知识图谱中的实体以及关系向量化后计算出用户信息相似度,将其与基于用户的传统协同过滤算法得到的用户评分相似度以一定比例进行融合,从而得到最终的用户相似度,并以此为基础进行评分预测并得到推荐列表. 实验结果表明:与传统协同过滤算法相比,该算法能有效地改善数据稀疏性问题,预测结果的精准率和召回率均有显著提升,同时具有较好的可解释性.

关键词: 协同过滤;知识图谱;信息强化;相似度融合
中图分类号: TP 311 **文献标志码:** A **DOI:** 10.16357/j.cnki.issn1000-5862.2022.04.09

0 引言

随着社会信息化进程的不断加快,大数据以及云计算技术不断运用于社会的各个领域,随之而来的是不断增大的数据量,在庞大的数据资源中如何筛选出用户所需要的信息已经成为一个刻不容缓的现实问题,个性化推荐算法成为目前研究的热点问题之一. 目前,主流推荐算法可分为3类:基于内容的推荐算法、基于协同过滤的推荐算法和混合推荐算法[1]. 在互联网行业中应用最广泛的推荐算法是协同过滤推荐算法.

传统的基于协同过滤的推荐算法[2]的主体思路是通过用户对物品的评分来构建评分矩阵,并进行近邻计算从而得到推荐列表. 但存在使用的用户数据特征过于单一和数据稀疏问题,这将产生推荐结果的可解释性不足和推荐效果差等问题.

许多学者考虑从不同的角度对协同过滤算法进行优化. 对于协同过滤算法存在的数据稀疏性问题,有研究使用项目的内容信息[3]、标签数据[4]、用户对项目的隐式反馈数据[5]等进行评分矩阵的填充或使用SVD[6]、PMF[7]等方法对评分矩阵进行矩阵分解. 对于推荐效果解释性问题,有些研究考虑引入推荐物品的辅助信息对物品进行多方面的挖掘[8],并将挖掘的结果融入推荐算法,以此增强推荐结果的可解释性.

本文提出了一种结合知识图谱进行信息强化的协同过滤算法:KGRI-CF. 该算法包括2张知识图谱. 一张是关于项目的知识图谱,它主要用于结合原有信息对用户-项目评分矩阵进行有条件地填充从而得到新的评分矩阵,改善了数据稀疏性问题. 另一张是关于用户的知识图谱,它主要用于计算用户的信息相似度,并与传统协同过滤计算出的用户相似度以一定比例进行融合得到一个新的用户相似度矩阵,最后利用该矩阵得到推荐结果. 该模型在MovieLens数据集上进行测试并通过调整参数的方式以达到最优的推荐效果. 实验结果表明:本文提出的结合知识图谱进行信息强化的协同过滤(KGRI-CF)算

收稿日期:2022-01-13
基金项目:国家自然科学基金(61877031)资助项目.
通信作者:杨庆红(1968—),女,江西南昌人,教授,主要从事软件形式化和智能教育软件的研究. E-mail:yangqh120@163.com

法在一定程度上提高了推荐的精准率与召回率,具有较好的可解释性及推荐效果.

1　相关研究

1.1　基于知识图谱的推荐算法

在基于知识图谱进行推荐的领域中,文献[9]提出了一种 TransE-CF 算法,它使用知识图谱表示学习方法,将业界已有的语义数据嵌入一个低维的语义空间中,通过计算物品之间的语义相似性,将物品自身的语义信息融入推荐方法中. 文献[10]提出了一种基于知识图谱和注意力机制的 KGAT 模型,该模型通过用户和项目之间的属性将用户-项目实例链接在一起,摒弃用户-项目之间相互独立的假设,同时将用户-项目和知识图谱融合在一起形成一种新的网络结构,并从该网络结构中抽取高阶链接路径来表示网络中的节点. 文献[11]提出了一种将知识图谱实体嵌入表示与神经网络融合进行新闻推荐的模型 DKN 中,它将新闻的语义表示与知识表示融合形成新的嵌入表示,再建立从用户的新闻点击历史到候选新闻的注意力机制,选出得分较高的新闻推荐给用户.

目前基于知识图谱的推荐算法大多是将知识图谱与神经网络结合进行推荐,此类算法虽然在推荐精度上有所提升,但由于神经网络的不可见性和复杂性,所以该类算法的可解释性较差并且难以应用于工业领域中. 本文提出的 KGRI-CF 算法将知识图谱与传统协同过滤算法相结合,不仅有效地提高了传统算法推荐精度,而且具有较好的可解释性,资源消耗较少,适合应用于工业领域中.

1.2　知识图谱的向量化算法

目前在推荐领域中学者都尝试使用将知识图谱融入推荐算法中,以加强对推荐物品更深层信息的利用,因此知识图谱的向量化成为该类算法的一个重要步骤. 文献[12]提出了一种 transE 算法,它将知识图谱的实体和关系初始化为一个向量集合,随机抽取某部分向量通过 L_1 或 L_2 运算进行更新,最终达到使用向量表示知识图谱中 3 元组的效果. 文献[13]提出了一种 transH 算法,它在 transE 算法的基础上,考虑了实体之间一对多、多对一、多对多的关系,并将实体在超平面上进行映射,让不同向量在不同关系下拥有不同的表示. 文献[14]提出了一种 transR 算法,它额外考虑到了一个实体在多层面上存在的信息差异,在 2 个不同的空间中建模实体和关系,并在对应的关系空间中进行转换. 文献[15]提出了一种 transD 算法,它不仅考虑到了关系的多样性,也考虑了实体的多样性,而且具有更少的参数,可以应用于大规模的知识图谱中.

2　结合知识图谱进行信息强化的协同过滤算法

本文提出了一种结合知识图谱进行信息强化的协同过滤 KGRI-CF 算法. 该算法以基于用户的传统协同过滤算法为基础,结合关于电影(本文研究的物品以电影为例)的知识图谱和关于用户的知识图谱进行信息增强,达到填充评分矩阵和辅助计算用户相似度的效果. 该算法的基本流程如图 1 所示.

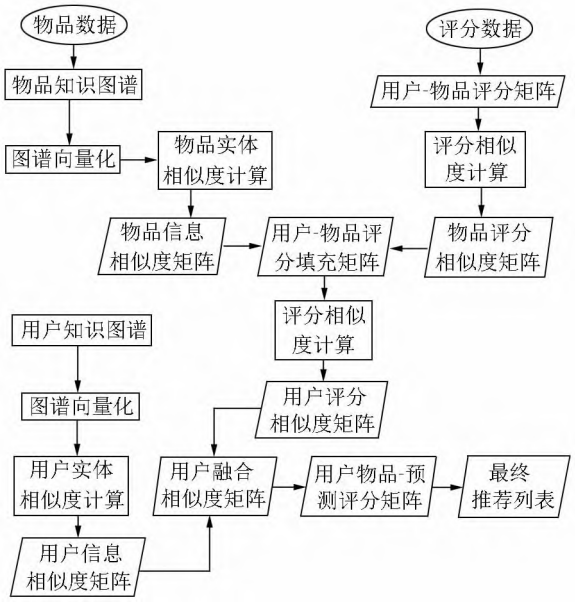

图 1　KGRI-CF 算法流程图

2.1　用户评分相似度矩阵构建

本节将具体介绍用户评分相似度矩阵的构建过程,全文使用的基础数据为网络公开数据集 MovieLens 的用户评分数据和电影信息数据,因此本文在具体研究过程中的物品即为电影. 鉴于电影数据内容过于单一,因此使用爬虫在百度百科、维基百科、豆瓣网等渠道中攫取电影的具体信息以补充电影信息数据,用户对电影的评分即代表对物品的权重.

首先使用基于物品的协同过滤算法生成电影评分相似度矩阵,然后使用电影数据构建关于电影的知识图谱,再利用向量化算法将知识图谱进行向量化以构建一个电影信息相似度矩阵. 将这 2 个相似度矩阵进行融合得到 1 个更新后的电影相似度矩阵,再以该矩阵为基础对用户-评分矩阵进行有条件地填充,以解决评分稀疏性问题,进而构建一个用户

评分相似度矩阵. 相比于传统的协同过滤算法,由于本文方法充分利用了物品自身特征之间的关联对评分矩阵进行填充,使得最终得到的用户评分相似度结果更精确,且有效地解决了传统算法的冷启动问题.

2.1.1 物品评分相似度矩阵构建 在一个推荐系统中用户的评分是推荐算法运行的重要数据来源,因此首先对基础数据中的用户评分数据进行处理得到一个用户–评分矩阵. 矩阵的每一行代表某个用户对所有电影的评分,每一列代表所有用户对某部电影的评分. 在利用评分矩阵计算物品的相似度时,使用余弦相似度,计算公式为

$$f_{\text{similarity}}(x,y) = \sum_{i=1}^{n} x_i y_i \Big/ \left(\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2} \right), \quad (1)$$

其中 x、y 代表 2 个任意的用户评分列. 利用式(1)可以通过用户–评分矩阵计算出每个物品和其他物品的相似度,从而得到一个物品评分相似度矩阵 S_1,该矩阵为一个 $n \times n$ 对称矩阵,n 为物品的数量,在矩阵中的元素表示其对应行所代表物品与对应列所代表物品的评分相似度. 该矩阵将在后续进行用户评分矩阵填充时被使用.

2.1.2 物品信息相似度矩阵构建 在推荐系统中的推荐物一般都具有多种特征,这些特征可以当作多个包含信息的特征实体,将推荐物所构成的实体与特征实体以某种关系通过有向边进行连接便构成了一个 3 元组. 这些 3 元组通过公共结点联系在一起便形成了一个关于推荐物的知识图谱. 因为在图谱中的特征实体包含了推荐物的信息特征,所以利用知识图谱可以充分表示推荐物品的特征关联.

以本文使用的电影数据集为例,一个电影实体主要包含导演、编剧、演员、类别、语言、上映年代、时长等重要特征. 通过这些重要特征可以建立电影实体与特征实体间的 3 元组集合. 3 元组集合通过公共节点产生的联系便构成了一个关于电影的知识图谱(见图 2).

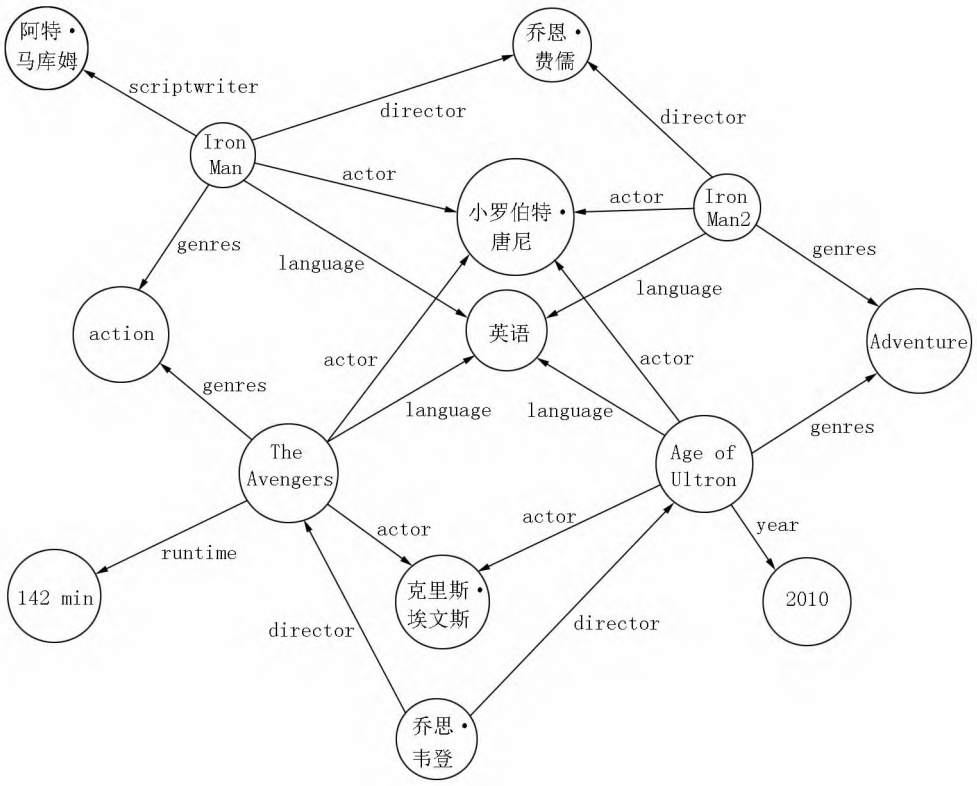

图 2 电影知识图谱

在图 2 中,不同的电影实体因为存在公共的特征节点而联系起来. 通过观察可以发现: 对于比较相似的 2 部电影,它们对应的实体结点往往在图谱中距离较近,反之则距离相对较远. 以此为依据,便可以利用知识图谱表现出的节点距离来计算电影之间的信息相似度,并以此构建一个物品信息相似度矩阵. 而这需要将知识图谱中的节点向量化,并通过向量体现出图谱中节点的距离关系. 由于在图谱中的实体不存在歧义,都表现为一对一的关系,因此本文选用 transE 算法对知识图谱中的实体进行向量化. 该算法使用的损失函数为

$$L = \sum_{(h,l,t) \in S} \sum_{(h',l,t') \in S(h,l,t)} (\lambda + d(h+l,t) - d(h'+l,t')).$$

该算法可以将知识图谱中距离较近的节点进行向量化,从而达到头节点向量加上关系向量接近尾

节点向量的效果,同时也能使 2 个不相关的节点在向量化后距离更远,算法效果通过损失函数进行评估修正,最终可通过向量体现在图谱中节点距离. 通过此算法可以得到一个在知识图谱中电影实体的向量矩阵,该矩阵的行数为在知识图谱中电影实体的数量,列数则为算法设置的向量维度.

在得到电影实体的向量矩阵后,再通过向量矩阵进行计算,从而得到一个物品信息相似度矩阵. 由于在使用 transE 算法时对距离的计算使用 L_1 范式,因此在计算相似度时也使用 L_1 范式计算距离,并将距离转化为相似度,其计算公式为

$$F_{\text{similarity}}(\boldsymbol{x},\boldsymbol{y}) = 1/\left(1+\sum_{i=1}^{d}|x_i-y_i|\right), \quad (2)$$

其中 $\boldsymbol{x}$、$\boldsymbol{y}$ 代表任意 2 个电影实体向量,d 为向量的维度. 从式(2) 可以看出: 若 $\boldsymbol{x}$、$\boldsymbol{y}$ 在图谱中距离越远,则相似度越小; 若距离越近,则相似度越大. 由式(2) 便能得到一个 $n\times n$ 的物品信息相似度矩阵 $\boldsymbol{S}_2$,该相似度矩阵将在后续进行用户评分矩阵填充时被使用.

2.1.3　**用户评分相似度矩阵构建**　由于传统的基于用户的协同过滤算法高度依赖用户的评分信息,在用户评分信息较少时,用户 -评分矩阵过于稀疏,所以这会产生推荐效果不理想、冷启动等问题. 对评分矩阵进行填充是解决上述问题的有效方法. 在上文已构建的物品评分相似度矩阵与物品信息相似度矩阵的基础上,可以将这 2 个矩阵以一定比例进行融合得到一个物品相似度矩阵. 将物品相似度矩阵的值作为权重可对用户 -评分矩阵进行有条件的填充,最后对填充后的评分矩阵进行相似度计算,相似度的计算公式为

$$g_{\text{similarity}}(u_1,u_2) = \sum_{i=1}^{m} u_{1i}u_{2i}\Big/\left(\sqrt{\sum_{i=1}^{m}u_{1i}^2}\sqrt{\sum_{i=1}^{m}u_{2i}^2}\right), \quad (3)$$

其中 m 为用户的数量,u_1、u_2 代表 2 个不同的用户对某电影的评分行,通过式(3) 可得到一个 $m\times m$ 的用户评分相似度矩阵 $\boldsymbol{S}_3$. 具体算法流程如下.

算法 1

输入: 物品评分相似度矩阵 $\boldsymbol{S}_1$,物品信息相似度矩阵 $\boldsymbol{S}_2$,用户 -物品评分矩阵 $\boldsymbol{R}$,融合比例 p,填充阈值 N;

输出: 用户评分相似度矩阵 $\boldsymbol{S}_3$;

(i) $\boldsymbol{S}\leftarrow p*\ \boldsymbol{S}_1+(1-p)*\ \boldsymbol{S}_2$;

(ii) $R_c\leftarrow \text{copy}(R)$;

(iii) for $R_{ij}\in R(1\leqslant i\leqslant m, 1\leqslant j\leqslant n)$ do:

(iv) $\text{sim} = S_j$;

(v) if $\text{sum}(R_i>0) > N$:

(vi) $R_{ij}\leftarrow(\text{sim}*\ R_{ij})/\text{sum}(\text{sim})$;

(vii) $R_c\leftarrow R_c.\text{append}(R_{ij})$;

(viii) end for;

(ix) $\boldsymbol{S}_3 = \text{cosine\_similarity}(R_c)$.

2.2　用户信息相似度矩阵构建

针对基于用户的协同过滤算法存在的使用信息单一的问题,通过对用户评分数据和电影详细信息的统计与挖掘构建一张关于用户的知识图谱. 对用户知识图谱使用算法进行向量化得到一个向量集合,使用该向量集合便可通过相似度计算来构建一个用户信息相似度矩阵.

2.2.1　**用户知识图谱构建**　将用户评分数据构建成一个用户 -评分矩阵,对该矩阵分析可以发现: 用户的评分为 0 ~ 5,其中 0 代表无评分,而 1 ~ 5 代表用户对电影的正式评分. 因此需要根据评分筛选出用户感兴趣的电影,并得到一个喜好电影列表. 简单地设置一个评分阈值来评判用户对该电影的感兴趣与否是一种普遍使用的方法,但不同用户的评分可能会存在这样一种情况: 不同的用户都会有自己的评价尺度,有的用户的评分意愿总体偏低,有的用户评分意愿总体偏高. 而单纯的阈值筛选显然没有考虑到用户的个人评价尺度,因此可以先将评分进行 z-score 标准化,其计算公式为

$$h(r_{ij}) = (r_{ij}-\bar{r}_i)/\sigma_i, \quad (4)$$

其中 r_{ij} 代表当前用户的评分,$\bar{r}_i$ 代表当前用户对已评分电影的平均分,σ_i 代表用户评分标准差. 若标准化后的分数大于 0 则用户对该电影感兴趣,小于 0 则用户对该电影不感兴趣. 标准化后的评分不仅考虑了用户评价尺度带来的偏差,还考虑了评分范围不同所带来的差异性,因此使用该计算公式可以得到一个更加准确的喜好电影列表.

根据上述方法得到的喜好电影列表,再结合通过多种渠道获取的电影信息数据,便可以通过数学统计得到一个用户的爱好列表. 在这个爱好列表中包括 6 个类型: 电影类别、导演、演员、编剧、语言以及年代,在每个类型中包括一定数量的类型元素. 将用户构造为用户实体,将爱好列表内部的类型元素构造为类型实体,2 个实体之间根据喜欢关系进行有向连接,便组成了关于用户的知识图谱(见图 3).

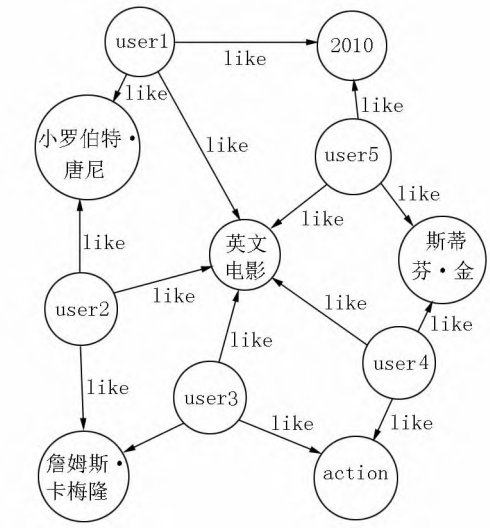

图 3 用户知识图谱

2.2.2 相似度矩阵构建 在用户知识图谱中的实体语义为一对一的类型，因此仍选用上文中的 transE 算法进行实体向量化，得到向量矩阵后再使用式(2)进行相似度计算得到一个 $m \times m$ 的用户信息相似度矩阵 S_4. 该矩阵在构建时充分利用了物品评分数据进行统计挖掘，在得到用户的详细偏好后构建知识图谱进行用户偏好分析，这使得用户的相似性度量更加精准与全面.

2.3 推荐结果生成

用户评分相似度矩阵 S_3 与用户信息相似度矩阵 S_4 已经完成构建，将这 2 个相似度矩阵以一个比例系数 r 进行融合得到一个最终的用户相似度矩阵. 通过用户相似度矩阵对用户评分矩阵进行加权计算得到一个用户预测评分矩阵，对每一个用户的预测评分进行排序，将其前 K 个评分对应的物品加入推荐列表得到该用户的物品推荐列表中. 用户推荐结果的生成算法如下.

算法 2

输入: 用户评分相似度矩阵 S_3、用户信息相似度矩阵 S_4、用户 -物品评分矩阵 R、融合比例 r、相似用户数 K;

输出: Top K 推荐列表 R_L;

(i) $S \leftarrow r * S_3 + (1 - r) * S_4$;

(ii) $R_P \leftarrow S * R$;

(iii) for $R_{P_i} \in R_P (1 \leqslant i \leqslant m)$:

(iv) $R_{\text{sort}} = \text{sort}(R_{P_i})$;

(v) list $\leftarrow$ top $K(R_{\text{sort}}, K)$;

(vi) $R_L \leftarrow R_L$. append(list);

(vii) end for;

(viii) 输出物品推荐列表 R_L.

2.4 算法总体描述

结合算法 1、算法 2 与流程图将算法整体描述如下.

输入: 用户 -评分矩阵 R 物品信息列表 L;

输出: 物品推荐列表 R_L,

(i) 根据式(1)通过用户 -评分矩阵 R 构建一个物品评分相似度矩阵 S_1;

(ii) 利用物品信息列表 L 构建一个关于物品的知识图谱 itemKG;

(iii) 将 itemKG 内的实体通过 transE 算法向量化得到一个物品的向量集合;

(iv) 根据式(2)对第(iii)步所得结果进行信息相似度计算，由此得到一个物品信息相似度矩阵 S_2;

(v) 根据算法 1 以及矩阵 S_1 与矩阵 S_2 将用户评分矩阵 R 进行有条件地填充，并通过式(3)对填充后的评分矩阵进行相似度计算，得到一个用户评分相似度矩阵 S_3;

(vi) 根据式(4)对用户 -评分矩阵 R 进行分析得到一个用户喜好物品列表，并结合物品信息列表 L 构建一个关于用户的知识图谱 userKG;

(vii) 将在 userKG 内的实体通过 transE 算法向量化得到一个用户的向量集合;

(viii) 根据式(2)对第(vii)步所得结果进行信息相似度计算，由此得到一个用户信息相似度矩阵 S_4;

(ix) 根据算法 2 在矩阵 S_3 与矩阵 S_4 融合后对评分矩阵进行加权计算出预测评分矩阵 R_P 并生成最终的物品推荐列表 R_L，将列表 R_L 推送给用户.

3 实验及结果分析

3.1 数据集

本文实验所使用的数据集为电影数据集 MovieLens-latest-small，由于在数据集中电影的信息种类极少，因此使用 Python 爬虫在百度百科、维基百科、豆瓣网等渠道中攫取电影具体信息补充电影信息数据并以此构建知识图谱. 攫取的内容包括导演、编剧、演员、时长、语言以及年代，最终所使用的电影信息数据包括原有的电影流派一共 7 个种类、用户 610 个、电影 9 742 部和评分 100 836 条(见表 1).

在数据集中由于用户对电影的评价只有评分，因此使用式(4)对每个用户的所有评分进行标准化，若原有评分大于此标准化评分，则用户喜欢该电影，此评分样本为正样本，反之则此评分样本为负样

本. 本文对用户评分数据以用户为单位进行划分,其中 70% 的用户评分数据为训练集,30% 的用户评分数据为测试集,训练集用于相似度计算,测试集用于算法性能测试.

表 1　数据集信息

数据类型	数量
用户	610
电影	9 742
评分数据	100 836
导演	3 949
编剧	8 835
演员	9 673
流派	20
语言	196
时长	203
年代	108

实验使用 CPU 为 Intel Core I7 9700,内存为 16 GB,代码使用 Python 进行编写,运行环境为 Python3. 7.

3.2　评价指标

对于推荐算法所产生的推荐结果,本文使用精准率与召回率进行评价,对于算法给出的当前用户的推荐列表,精准率描述的是推荐结果准确的数量与当前总的推荐数量的比值,召回率描述的是推荐结果准确的数量与当前用户感兴趣的物品数量的比值,计算公式分别为

$$P_{recision} = \sum_{i \in I} | R(i) \cap L(i) | / \sum_{i \in I} | R(i) |,$$

$$R_{ecall} = \sum_{i \in I} | R(i) \cap L(i) | / \sum_{i \in I} | L(i) |,$$

其中 $R(i)$ 为推荐算法给出的当前用户的推荐物品列表,$L(i)$ 为当前用户感兴趣的物品列表. 由于本文提出的推荐算法最终采用 top K 方法确定每个用户的推荐列表,因此使用 Precision@K,Recall@K (K =5,10,15,20) 指标进行最后的评测,即选取不同的推荐列表长度 K,计算其精准率与召回率并与其他算法进行比较.

3.3　实验参数调整

本文所提出的算法主要可调节参数包含下列 2 个: 物品评分相似度矩阵 S_1 与物品信息相似度矩阵 S_2 的融合比例 p、用户评分相似度矩阵 S_3 与用户信息相似度 S_4 的融合比例 r,对于不同的融合比例,推荐算法的推荐效果也会不同.

在调整算法 1 参数时,依据用户总体的电影评分情况,给定填充阈值 N 为 500,将融合比例 p 从0. 1 开始且以 0. 1 为步长逐步上调得到不同的融合相似度矩阵,使用均方根误差对不同的融合比例下矩阵的填充效果进行评估,图 4 为在不同比例下的均方根误差.

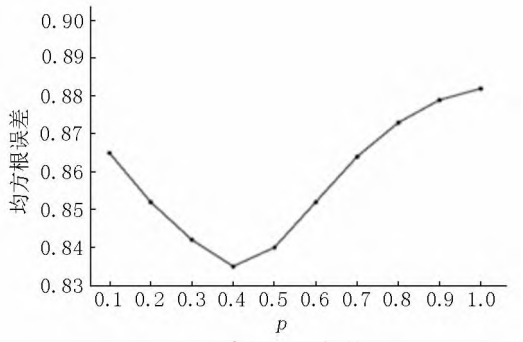

图 4　均方根误差趋势图

从图 4 可以看出: 在其他条件不变的情况下,当物品评分相似度所占比例较低时的误差比物品评分相似度所占比例较高时的均方根误差总体上更低,当 p = 0. 4 时,误差最低. 因此,在填充评分矩阵时融合矩阵以物品信息相似度矩阵为主可以有效降低误差,且在融合比例 p = 0. 4 时填充效果最好.

在调整算法 2 参数时,将推荐数量 K 设置为 20,并使用算法 1 中当 p = 0. 4、N = 500 时所得到的用户评分相似度矩阵. 用户评分相似度矩阵的所占比 r 从 0. 1 开始以 0. 1 为步长逐步上调. 图 5 为在不同比例系数下的精准率曲线图,图 6 为在不同比例下的召回率曲线图,每个值都是将算法循环 10 次所取的平均值.

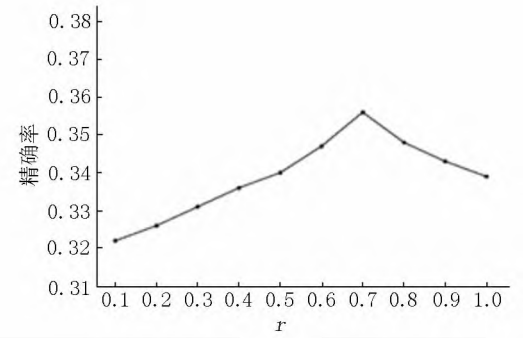

图 5　精准率趋势图

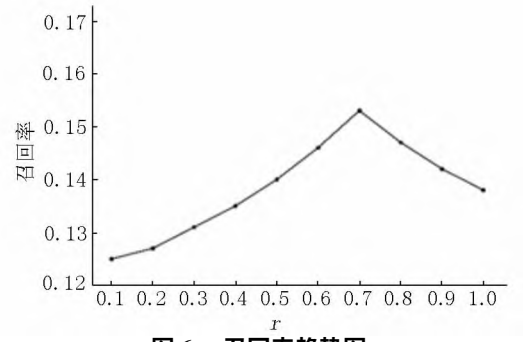

图 6　召回率趋势图

从图 5 及图 6 可以看出: 在其他条件不变的情况下,精准率及召回率随着比例 r 的增大而逐渐上

升,当 $r = 0.7$ 时,精准率与召回率达到最高. 因此,当 $p = 0.4$,$n = 500$,$r = 0.7$ 时,算法整体效果最佳.

3.4 实验结果分析

为了验证实验结果的有效性,选取以下几种模型算法在 Movielens 数据集上进行实验对比,对比指标为 Precision@K 与 Recall@K. 实验结果如表 2 所示,其中 User-CF 代表基于用户的协同过滤算法,Item-CF 代表基于项目的协同过滤算法,traneE-CF 代表基于知识图谱表示学习的协同过滤算法,HeteRs 代表一种基于图的推荐算法. 具体实验数据如表 2 与表 3 所示.

从表 2 和表 3 中可以看出: 由于感兴趣物品列表的数量始终不变,因此推荐列表的长度 K 值越大,则命中感兴趣物品的概率越大,计算得到的精准率会逐步下降,同时,召回率会逐步上升. 同时本文提出的结合知识图谱进行信息强化的协同过滤算法 KGRI-CF 在精准率和召回率上优于传统协同过滤算法及其他推荐算法,并且在不同 K 值的情况下都能保持优势. 这充分说明 KGRI-CF 算法充分地考虑了推荐物品的特征信息和用户的偏好信息,具有更好的推荐效果.

表 2 不同算法精准率对比

模型	Precision@5	Precision@10	Precision@15	Precision@20
Hete-Rs	0.363	0.322	0.281	0.266
Item-CF	0.379	0.335	0.296	0.275
User-CF	0.412	0.367	0.318	0.283
transE-CF	0.398	0.356	0.323	0.292
KGRI-CF	0.465	0.423	0.382	0.356

表 3 不同算法召回率对比

模型	Recall@5	Recall@10	Recall@15	Recall@20
Hete-Rs	0.031	0.065	0.096	0.113
Item-CF	0.036	0.075	0.113	0.132
User-CF	0.048	0.083	0.115	0.138
transE-CF	0.037	0.081	0.119	0.142
KGRI-CF	0.055	0.096	0.130	0.153

4 总结与展望

本文提出了一种结合知识图谱进行信息强化的协同过滤推荐算法,该算法在基于用户的传统协同过滤算法的基础上,利用物品的评分数据进行物品特征挖掘和用户偏好挖掘,使用物品的特征信息构建一个物品知识图谱用以填充评分矩阵并计算用户评分相似度; 使用用户偏好信息构建一个用户知识图谱,并计算用户信息相似度. 最终融合这 2 种相似度对评分矩阵进行加权计算得到推荐结果. 结果表明: 该算法将推荐物品的特征信息用于填充评分矩阵并以此挖掘用户的偏好信息,在精准率和召回率上比传统的协同过滤算法及其他推荐算法取得了更好的效果,并具有较好的算法可解释性.

下一步可以考虑将用户偏好之外的更多用户信息加入用户知识图谱,并尝试使用不同方式融合相似度以优化算法性能.

5 参考文献

[1] 翁小兰,王志坚. 协同过滤推荐算法研究进展 [J]. 计算机工程与应用,2018,54(1):7-8.

[2] GOLDBERG D,NICHOLS D,Oki B M,et al. Using collaborative filtering to weave an information tapestry [J]. Communications of the ACM,1992,35(12):61-70.

[3] DI Jiaqi,WANG Nihong. Incremental collaborative filtering algorithm based on gridgis [EB/OL]. [2021-06-16]. http://en.cnki.com.cn/Article_en/CJFDTOTAL-JSJA201312048.htm.

[4] XU Yueshen,YIN Jianwei. Collaborative recommendation with user generated content [J]. Engineering Applications of Artificial Intelligence,2015,45:281-294.

[5] CUI Hua,ZHU Ming. Collaboration filtering recommendation optimization with user implicit feedback [J]. Journal of Computational Information Systems,2014,10(14):5855-5862.

[6] ZHOU Xun,HE Jing,HUANG Guangyan,et al. SVD-

based incremental approaches for recommender systems [J]. Journal of Computer and System,2015,81(4):717-733.

[7] SHAN Hanhuan,BANERJEE A. Generalized probabilistic matrix factorizations for collaborative filtering [EB/OL]. [2021-06-17]. https://doi.org/10.1109/ICDM.2010.116.

[8] 李浩,张亚钏,康雁,等.融合循环知识图谱和协同过滤电影推荐算法[J].计算机工程与应用,2020,56(2):106-114.

[9] 吴玺煜,陈启买,刘海,等.基于知识图谱表示学习的协同过滤推荐算法[J].计算机工程,2018,44(2):226-232,263

[10] WANG Xiang,HE Xiangnan,CAO Yixin,et al. Kgat: knowledge graph attention network for recommendation [EB/OL]. [2021-06-11]. https://arxiv.org/abs/1905.07854v2.

[11] WANG Hongwei,ZHANG Fuzheng,XIE Xing,et al. DKN: deep knowledge-aware network for news recommendation [EB/OL]. [2021-06-21]. https://arxiv.org/pdf/1801.08284.pdf.

[12] BORDES A,USUNIER N,GARCIA-DURÁN A,et al. Translating embeddings for modeling multi-relational data [C]//BURGES C J,BOTTOU L,WELLING M,et al. Proceedings of the 26th International Conference on Neural Information Processing Systems. Cancouver: MIT Press,2013:2787-2795.

[13] WANG Zhen,ZHANG Jianwen,FENG Jianlin,et al. Knowledge graph embedding by translatingonhyperplanes [J]. Proceedings of the AAAI Conference on Artificial Intelligence,2014,28(1):1112-1119.

[14] DAI Shaozhi,LIANG Yanchun,LIU Shuyan,et al. Learning entity and relation embeddings with entity description for knowledge graph completion [EB/OL]. [2021-06-16]. http://download.atlantis-press.com/article/25894200.pdf.

[15] JI Guoliang,HE Shizhu,XU Liheng,et al. Knowledge graph embedding via dynamic mapping matrix [EB/OL]. [2021-06-16]. https://aclanthology.org/P15-1067/.

The Collaborative Filtering Algorithm for Information Enhancement Combined with Knowledge Graph

FENG Xiang,YANG Qinghong*

(School of Computer Information Engineering,Jiangxi Normal University,Nanchang Jiangxi 330022,China)

Abstract: The collaborative filtering algorithm that combines knowledge graphs for information enhancement(KGRI-CF) is proposed,aiming at the problems of single use information and too sparse basic scoring data leading to poor recommendation effect in traditional collaborative filtering algorithms. The algorithm uses the feature data of the movie to construct a knowledge map about the movie,and conditionally fills the user-rating matrix,which effectively improves the data sparsity problem of the traditional collaborative filtering algorithm. Preference information is used to build a knowledge graph about user preferences. The entity vectorization algorithm is used to vectorize the entities and relationships in the knowledge graph to calculate the similarity of user information,which is fused with the similarity of user ratings obtained by the traditional user-based collaborative filtering algorithm in a certain proportion to obtain the final user similarity. On this basis the score prediction is performed and the recommendation list is obtained. The experimental results show that,compared with the traditional collaborative filtering algorithm,the algorithm can effectively improve the data sparsity problem,the accuracy and recall rate of the prediction results are significantly improved,and it has better interpretability.

Key words: collaborative filtering; knowledge graph; information enhancement; similarity fusion

(责任编辑:冉小晓)