

融合知识图谱表示学习和矩阵分解的推荐算法

陈平华, 朱 禹

(广东工业大学 计算机学院, 广东 广州 510006)

摘 要: 针对基于矩阵分解推荐算法准确率不高的问题, 提出一种融合知识图谱表示学习和矩阵分解的推荐算法。利用知识图谱表示学习方法, 将知识图谱中实体和关系的语义信息嵌入低维的向量空间中, 通过计算实体之间的语义相似性, 将实体自身的知识信息融入矩阵分解模型。算法弥补了矩阵分解方法没有考虑项目自身知识信息的不足, 在知识层面上增强了矩阵分解方法的效果。MovieLens 数据集上的实验结果表明, 该算法有效提升了推荐结果的准确率。

关键词: 推荐算法; 协同过滤; 矩阵分解; 知识图谱; 表示学习

中图法分类号: TP391 **文献标识号:** A **文章编号:** 1000-7024 (2018) 10-3137-06

doi: 10.16208/j.issn1000-7024.2018.10.024

Recommendation algorithm incorporating representation learning of knowledge graph with matrix factorization

CHEN Ping-hua, ZHU Yu

(School of Computer, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: To solve the problem of low accuracy of recommendation algorithm based on matrix factorization, a recommendation algorithm incorporating representation learning of knowledge graph with matrix factorization was presented. The representation learning of knowledge graph was utilized to project both entities and relations into a low-dimensional semantic space, the knowledge information of entities and matrix factorization was combined by calculating the semantic similarity between entities. The shortcoming of matrix factorization that it fails to consider the knowledge information of items is overcome and the effect of matrix factorization result is improved on the knowledge level. Experimental results on MovieLens data set demonstrate that the proposed algorithm has better recommendation accuracy.

Key words: recommendation algorithm; collaborative filtering; matrix factorization; knowledge graph; representation learning

0 引 言

针对推荐系统面临的数据稀疏性问题, 基于矩阵分解的推荐算法应运而生。传统的基于矩阵分解的推荐算法只考虑了对用户项目评分矩阵进行分解, 没有充分利用数据信息, 没有考虑项目之间的相似性问题, 导致推荐效果不佳。针对这些问题, 学者们提出了一系列的解决方法。文献 [1] 通过用户之间的信任关系来提高推荐系统的准确性, 提出了一种基于信任传播的矩阵分解模型, 但信任关系数据通常较难获取。文献 [2] 提出了一种基于项目属性耦合性的矩阵分解模型, 这种方法利用项目属性之间的耦

合关系作为隐含信息, 结合矩阵分解模型对用户评分数据约束求解。文献 [3] 提出了一种基于非对称的用户相似性方法应用于矩阵分解模型, 利用用户的相似性信息提高了推荐系统的准确性。如果在矩阵分解技术的基础上融合项目本身潜在的知识信息, 将会进一步提高推荐算法的准确率。

1 相关工作

近年来, 随着知识图谱技术的不断取得进展, 包含丰富实体和关系信息的知识库数据大量涌现, 例如 Freebase^[4], YOGA^[5], Wikidata^[6], DBpedia^[7] 等, 这些知识

收稿日期: 2017-08-29; 修订日期: 2017-10-10

基金项目: 广东省省级科技计划基金项目 (2016B030308001); 广东省省级科技计划基金项目 (2016B030306002); 广州市科技计划基金项目 (201604010099)

作者简介: 陈平华 (1969-), 男, 湖南株洲人, 教授, 硕士生导师, 研究方向为推荐系统与数据挖掘; 朱禹 (1993-), 男, 安徽六安人, 硕士研究生, 研究方向为数据挖掘、推荐系统、知识图谱、知识问答。E-mail: whatever_the_end@163.com

库提供了丰富的知识信息,知识图谱也引起了学术界广泛的关注和研究。文献[8]使用异构信息网络表示知识图谱中的项目和项目属性关系,并采用基于贝叶斯的协同过滤解决实体推荐问题。文献[9]采用扩散激活技术,将知识图谱中的网络结构特性融入推荐系统中。文献[10]提出了协同知识库嵌入框架,将知识图谱中的语义信息作为隐性反馈融入协同过滤中,加强了协同过滤算法的性能。文献[11]利用知识图谱表示学习计算物品之间的语义相似性,将物品的语义信息融入协同过滤。现有的研究表明,知识图谱表示学习可以将知识图谱中的实体和关系嵌入在低维的向量空间中,从而在向量空间中进行计算实体和关系的语义联系,有效解决冷启动和数据稀疏问题。与上述工作相比,本文提出了一种融合知识图谱表示学习和矩阵分解的推荐方法,将矩阵分解模型与知识图谱表示学习后的项目潜在知识信息相结合。该模型首先通过知识图谱表示学习算法获取项目的潜在知识信息,再将项目的潜在知识信息融入到矩阵分解模型中,从而产生更为准确的推荐结果。

2 基本理论

2.1 矩阵分解

矩阵分解方法的基本思想是认为每个用户和每个项目都有自己的特征,用户的兴趣只受少数因素影响,利用矩阵分解的方法可以从用户项目交互矩阵中分解出用户特征矩阵和项目特征矩阵。可以将矩阵分解模型抽象为如下公式

$$R = UV^T \quad (1)$$

式中: $U \in R^{m \times d}$ 和 $V \in R^{n \times d}$ 分别代表用户项目交互矩阵分解后的 d 维用户特征矩阵和 d 维项目特征矩阵。通过分解后低维度的用户特征矩阵和项目特征矩阵的乘积 UV^T 来近似拟合已有的项目真实评分矩阵。模型的学习主要通过最小化如下目标函数进行训练

$$\mathcal{L} = \sum_{i=1}^m \sum_{j=1}^n I_{ij} (U_i V_j^T - r_{ij})^2 + \lambda_1 \|U_i\|_F^2 + \lambda_2 \|V_j\|_F^2 \quad (2)$$

式中: I_{ij} 表示用户 i 对项目 j 是否有评分。如果用户 i 对项目 j 有评分, I_{ij} 值为 1, 否则 I_{ij} 的值为 0。 r_{ij} 表示用户 i 对项目 j 的实际评分, U_i 和 V_j 分别是分解后的 d 维用户特征向量和 d 维项目特征向量, $U_i V_j^T$ 表示预测评分, $(U_i V_j^T - r_{ij})^2$ 代表预测值和实际值的平方误差。 $\|\cdot\|_F^2$ 是弗罗贝尼乌斯(Frobenius)范数, $\lambda_1 \|U\|_F^2 + \lambda_2 \|V\|_F^2$ 是用来避免模型在参数学习过程中过拟合的正则项。其中 λ_1 和 λ_2 分别控制用户和项目特征矩阵正则化在整个模型中的比例。为了更好控制模型在参数学习时的过拟合问题,文献[12]提出了一种针对 λ 参数加权正则化的方法,分别引入用户数量 n_{u_i} 和项目数量 n_{v_j} 来规避目标函数向活跃用户或者热门的项目偏移。改进后的损失函数如下

$$\mathcal{L} = \sum_{i=1}^m \sum_{j=1}^n I_{ij} (U_i V_j^T - r_{ij})^2 + \lambda_1 \sum_i n_{u_i} \|U_i\|_F^2 + \lambda_2 \sum_j n_{v_j} \|V_j\|_F^2 \quad (3)$$

2.2 知识图谱表示学习

Bordes 等提出的 TransE 模型^[13],它是基于知识图谱中实体和关系的分布式向量表示。如图 1 所示,对于知识图谱三元组 (h, r, t) 中的关系 r 看作是头实体 h 到尾实体 t 的翻译,通过不断调整 h , r 和 t 的向量,使得 $h+r$ 尽可能的与 t 相等,即 $\|h+r\| \approx t$ 。其中 $\|\cdot\|$ 可以选择 L1 或者 L2 范数。

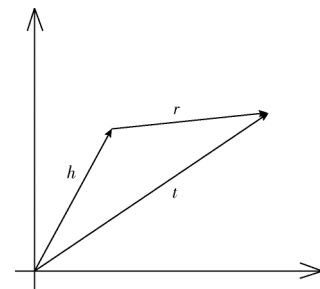


图 1 TransE 模型

TransE 在处理一对一的关系上得到很好的应用,但在处理复杂关系(一对多,多对一,多对多)时存在不足。为了解决 TransE 在处理复杂关系时的不足,文献[14]提出了 TransR 模型,TransR 的基本思想是每个实体都有多种属性,不同的关系所侧重的实体属性不同。两个实体在同一实体空间可能比较相似,但是在特定的关系空间可能会有差别。所以 TransR 模型定义了一个映射矩阵,将实体从实体空间投影到特定的关系空间后,再建立两个实体之间的翻译关系,如图 2 所示。

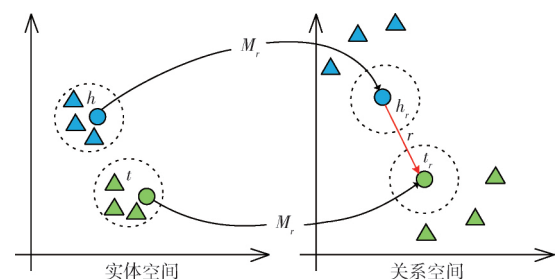


图 2 TransR 模型

其中 M_r 是关系 r 的映射矩阵, h_r 和 t_r 分别是头实体 h 和尾实体 t 在关系空间 r 中的向量表示。它们之间的关系可以用下式表示

$$h_r = h M_r \quad (4)$$

$$t_r = t M_r \quad (5)$$

TransR 模型的目标函数可以定义为下式

$$\mathcal{L} = \sum_{(h,r,t) \in S} \sum_{(h',r',t') \in S'} \max(0, f_r(h,t) + \gamma + f_r(h',t')) \quad (6)$$

式中: $\max(0, f(x))$ 是最大间隔函数, 用于加强表示学习后的区分能力。 $f_r(x)$ 是损失函数, 定义如下

$$f_r(h,t) = \|h_r + r - t_r\| \quad (7)$$

γ 是一个边际参数。 S 是正例三元组, S' 是负例三元组。负例三元组 S' 采用头实体或者尾实体随机替换为其它任意实体的方式来得到, 可以表示为

$$S'_{(h,r,t)} = \{(h',r,t | h' \in E)\} \cup \{(h,r,t' | t' \in E)\} \quad (8)$$

式中: h' 和 t' 分别表示被替换的头实体和尾实体。

3 融合知识图谱表示学习和矩阵分解的推荐算法

3.1 算法基本思想

矩阵分解求解过程中, 通过梯度下降算法最优化目标函数式 (3) 求出用户特征矩阵和项目特征矩阵, 进而求出用户对项目评分的近似值。在此求解过程中, 可能会丢失项目潜在的信息。其中, 项目的相似性就是很重要的潜在信息。考虑项目之间的潜在因子特征, 我们提出了融合知识图谱表示学习和矩阵分解的推荐算法(记为 TransR-MF)。

TransR-MF 的基本思想是高度相似的项目它们矩阵分解后的向量表示也是高度相似的。比如, 项目 v_1 和项目 v_2 相似, 则代表他们的向量也是相似的。知识图谱表示学习将项目实体嵌入低维的向量空间, 通过相似度函数计算低维向量空间中项目之间的相似性, 将相似性潜在因子融入矩阵分解模型中, 以此融入项目间的隐含信息。

3.2 算法步骤及论述

融合知识图谱表示学习和矩阵分解的推荐算法的流程步骤如下:

步骤 1 首先通过知识图谱表示学习 TransR 算法训练得到实体的向量表达;

步骤 2 将用户项目交互矩阵中的项目实体与知识图谱中的实体进行匹配;

步骤 3 选取知识图谱中与待预测项目最相似的 k 个近邻融入矩阵分解模型;

步骤 4 通过模型学习求解矩阵分解后的低维用户和项目矩阵, 并计算得出项目的预测评分。

TransR-MF 算法如图 3 所示。

融合后的算法目标函数如下

$$\begin{aligned} \mathcal{L} = & \sum_{i=1}^m \sum_{j=1}^n I_{ij} (U_i V_j^T - r_{ij})^2 + \\ & \lambda_1 \sum_i n_{u_i} \|U_i\|_F^2 + \lambda_2 \sum_j n_{v_j} \|V_j\|_F^2 + \\ & \lambda_3 \sum_{j=1}^n \left(V_j - \sum_{V_k \in N_{V_j}} \text{sim}(V_j, V_k) V_k \right)^2 \end{aligned} \quad (9)$$

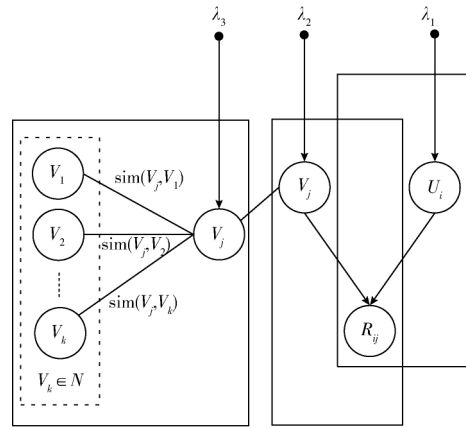


图 3 TransR-MF 算法

上述公式中第一项来自矩阵分解模型; 第二项和第三项分别是防止用户特征矩阵和项目特征矩阵过拟合的正则项; 第三项是通过知识图谱表示学习计算出的项目相似性潜在信息, 这里 N_{V_j} 表示项目 V_j 的最相似的 k 个近邻集合。其中 $\text{sim}(V_j, V_k)$ 是相似度函数, 本文使用余弦相似性函数。余弦相似性函数是一种计算两个向量之间相似度的方法, 其值的范围介于 $[-1, 1]$ 区间, 如式 (10) 所示, 其中 d 是 TransR 模型训练出来的实体向量的维度。这里为了取正数, 通过式 (11) 进行标准化处理

$$\text{sim}(V_j, V_k) = \frac{\sum_{i=1}^d (V_{jd} V_{kd})}{\sqrt{\sum_{i=1}^d (V_{jd})^2} \sqrt{\sum_{i=1}^d (V_{kd})^2}} \quad (10)$$

$$f(x) = (1+x)/2 \quad (11)$$

本文采用梯度下降算法最小化目标函数来求解用户特征矩阵 U 和物品特征矩阵 V , 利用式 (12) 和式 (13) 求解

$$\frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j=1}^n I_{ij} (U_i V_j^T - r_{ij}) V_j + \lambda_1 \sum_i n_{u_i} U_i \quad (12)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial V_j} = & \sum_{i=1}^n I_{ij} (U_i V_j^T - r_{ij}) U_i + \lambda_2 \sum_j n_{v_j} V_j + \\ & \lambda_3 \sum_{j=1}^n \left(V_j - \sum_{V_k \in N_{V_j}} \text{sim}(V_j, V_k) V_k \right) - \\ & \lambda_3 \sum_{V_k \in N_{V_j}} \text{sim}(V_j, V_k) \left(V_j - \sum_{V_l \in N_{V_j}} \text{sim}(V_j, V_l) V_l \right) \end{aligned} \quad (13)$$

4 实验结果及分析

4.1 数据集

本节主要对上述提出的 TransR-MF 算法进行验证与评估, 实验数据集采用 MovieLens-1M, 该数据集主要包括 6040 个 MovieLens 用户对 3900 部电影的 1 000 209 条评分纪录数据^[15]。

知识图谱数据集采用抓取 IMDB 电影资料库^[16]的方式建立。IMDB 是一个包含电影类型、电影演员、电影导演、

电视节目、和电影制作等信息的在线电影数据库。截止 2016 统计,IMDB 共收录了 4 007 049 部作品以及 7 593 030 个人物的数据^[17]。电影知识图谱的构建,本文参考了文献^[18]的构建流程。抓取 IMDB 电影资料库的电影数据,在对数据结构化之后进行知识抽取,抽取的数据以三元组的形式存储,电影知识图谱构建过程如图 4 所示。

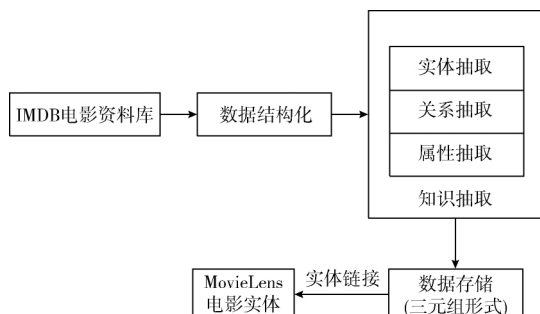


图 4 知识图谱构建流程

经过处理后最终得到 27 424 个实体、12 种关系属性和 218 115 条三元组数据。关系属性中保留了导演、主演、影片类别、影片发行国家、影片语言等 12 种关系。基本数据统计见表 1。

表 1 电影图谱数据统计

实体	关系	三元组
27 424	12	218 115

由于 IMDB 电影资料库构建的电影实体和 MovieLens-1M 数据集中的电影存在无法全部匹配的情况。例如,电影 Toy Story(1995) 和电影 Toy Story 是同一部电影,但因为添加电影发行年份的原因造成 IMDB 电影实体和 MovieLens 电影名无法完全匹配。此外,例如电影 Cup, The (Phörpa) 和电影 The Cup 实质上是同一部电影因为 MovieLens 添加了外文名导致无法匹配。

为了将 IMDB 电影资料库抽取的电影实体与 MovieLens-1M 电影匹配,本文采用实体链接方法将 MovieLens-1M 数据集中的每部电影映射到知识图谱中。经过映射后最终匹配了 3627 部电影实体数据,MovieLens-1M 中 93% 的电影能够准确映射到知识图谱中,这足以进行后续的工作。

4.2 交叉验证

本文采用 5 折交叉验证的方法进行训练和测试。其中,80% 的数据用于训练,20% 的数据用于验证。对于接下来的实验,最终的实验结果通过 5 次实验求平均值的方式得出。

4.3 评价指标

本文采用平均绝对值误差(MAE)和均方根误差

(RMSE) 两个指标对实验结果进行评估。平均绝对误差是所有项目实际评分和预测评分误差的绝对值之和的平均值,它衡量了真实值和预测评分之间的平均差异。均方根误差是所有项目真实评分和预测评分误差的平方之和的平均值的根值。

MAE 和 RMSE 是推荐系统评分预测评价指标中常用的两种度量方式。MAE 和 RMSE 分别定义如下公式所示

$$MAE = \frac{\sum_{(i,j) \in T} |R_{ij} - r_{ij}|}{|T|} \quad (14)$$

$$RMSE = \sqrt{\frac{\sum_{(i,j) \in T} (R_{ij} - r_{ij})^2}{|T|}} \quad (15)$$

平均绝对误差和均方根误差中定义的 T 表示的是需要预测的项目集合, R_{ij} 是用户 u_i 对项目 v_j 的真实评分值, r_{ij} 是模型的预测评分值。MAE 和 RMSE 的值越小,代表推荐的结果越准确。

4.4 实验结果及分析

实验硬件环境采用 ThinkCentre M8400t 台式机,处理器为 Intel Core i7 4770S,内存大小为 8 GB,硬盘大小为 1 TB,软件环境采用 Python2.7。

为了验证本文提出的 TransR-MF 算法的推荐准确度,我们对比了基于矩阵分解的其它算法:非负矩阵分解算法^[19](non-negative matrix factorization, NMF)、概率矩阵分解算法^[20](probabilistic matrix factorization, PMF)、融合因子学习的矩阵分解算法^[21](factor wise matrix factorization, FWMF)、结合项目偏置的矩阵分解算法^[22](biased matrix factorization, BMF)。

本文的推荐算法 TransR-MF 的参数设置如下:用户和项目的特征向量维度设为 100 维, λ_1 和 λ_2 设为 0.01, λ_3 设为 0.4,学习率设为 0.01。知识图谱表示学习的嵌入维度在选取不同值的时候,所取得的推荐效果也会有所不同,针对项目实体嵌入的维度,分别选取了 50-250 维进行实验,实验结果如图 5 和图 6 所示。

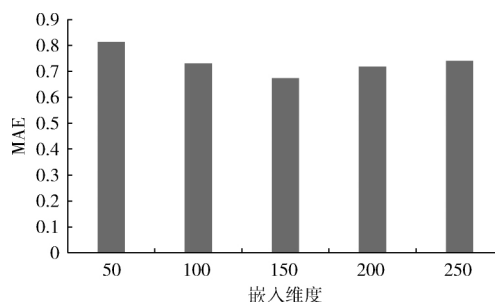


图 5 不同嵌入维度下的 MAE 值

从图 5 和图 6 可以看出,当知识图谱表示学习的嵌入维度为 150 维的情况下,在平均绝对值误差(MAE)和均方根误差(RMSE)上均有所改善,本文提出的算法能够获得

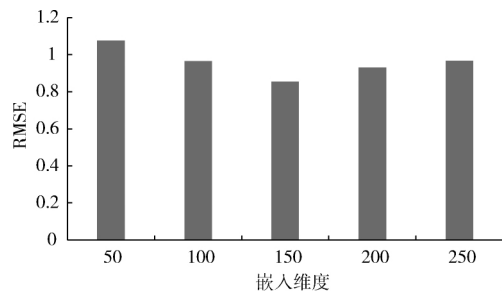


图 6 不同嵌入维度下的 RMSE 值

较好的推荐效果。

融合知识图谱表示学习过程中，待预测项目最相似的近邻个数在选取不同值的时候对推荐效果也会存在影响。采用控制变量法思想，在控制知识图谱嵌入维度为 150 维的情况下，通过选取项目不同的近邻个数来观察近邻个数对推荐效果的影响，这里分别设置近邻个数为 10-30 个的情况下进行实验，实验结果如图 7、图 8 所示。

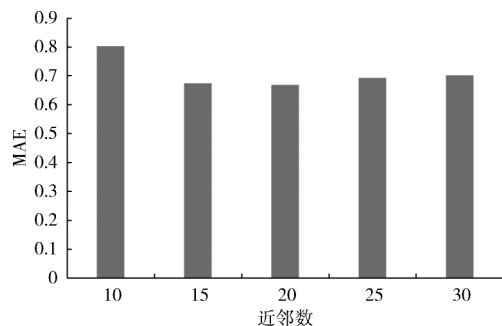


图 7 不同近邻数下的 MAE 值

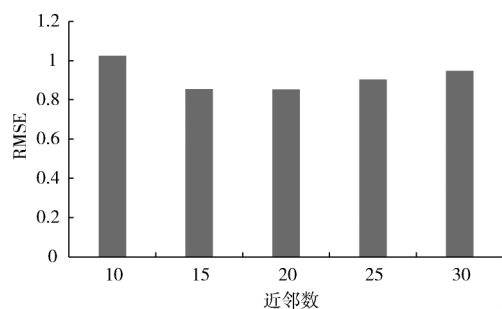


图 8 不同近邻数下的 RMSE 值

从图 7 和图 8 可以看出，近邻个数的选取对模型的推荐效果有所影响。当近邻个数为 20 时，MAE 和 RSME 的值均为最低。

为了验证该算法与同类算法的性能比较，这里选取知识图谱表示学习嵌入维度为 150 维，近邻个数为 20 个，用户特征向量和项目的特征向量维度设为 100 维。实验结果见表 2。

表 2 实验结果对比

方法	TransR-MF	NMF	PMF	FWMF	BMF
MAE	0.6682	0.7232	0.6916	0.6891	0.6845
Improve	—	5.5%	2.34%	2.09%	1.63%
RMSE	0.8524	0.9155	0.8836	0.8817	0.8726
Improve	—	6.31%	3.12%	2.93%	2.02%

通过表 2 可以看出在 MovieLens-1M 数据集上，融合知识图谱表示学习和矩阵分解的算法 TransR-MF 在 MAE 和 RMSE 上优于其它 4 种算法。因此融合项目实体潜在信息来预测用户评分是有效的，在一定程度上对矩阵分解模型有优化作用，提高推荐系统的准确性。

5 结束语

本文提出了一种融合知识图谱表示学习和矩阵分解的推荐算法，即 TransR-MF 算法。相对于只利用用户项目评分信息的传统推荐算法而言，TransR-MF 算法能够充分利用物品本身潜在的语义信息，弥补了矩阵分解算法没有考虑物品本身潜在信息的缺点。算法使用知识图谱表示学习把项目实体嵌入低维向量空间，计算项目之间潜在相似信息，并将其融入矩阵分解模型中，在一定程度上增强了矩阵分解模型的效果。此外，MovieLens 数据集上的实验结果验证了本文提出的 TransR-MF 算法有效提高了推荐结果的准确性。用户的潜在关系信息也是推荐系统中的一种重要潜在因素，因此在本文算法的基础上融合用户潜在信息将是下一步的研究重点。

参考文献：

[1] Forsati R, Mahdavi M, Shamsfard M, et al. Matrix factorization with explicit trust and distrust side information for improved social recommendation [J]. ACM Transactions on Information Systems, 2014, 32 (4): 17.

[2] Li F, Xu G, Cao L. Coupled item-based matrix factorization [C] //International Conference on Web Information Systems Engineering. Springer, 2014: 1-14.

[3] Pirasteh P, Hwang D, Jung JJ. Exploiting matrix factorization to asymmetric user similarities in recommendation systems [J]. Knowledge-Based Systems, 2015, 83 (1): 51-57.

[4] Berant J, Chou A, Frostig R, et al. Semantic parsing on free-base from question-answer pairs [C] //Proceedings of the Conference on Empirical Methods in Natural Language. ACL Anthology, 2013: 1533-1544.

[5] Rebele T, Suchanek F, Hoffart J, et al. YAGO: A multilingual knowledge base from Wikipedia, WordNet, and geonames [C] //International Semantic Web Conference. Springer International Publishing, 2016: 177-185.

- [6] Vrandečić D, Krötzsch M. Wikidata: A free collaborative knowledgebase [J]. *Communications of the ACM*, 2014, 57 (10): 78-85.
- [7] Lehmann J, Isele R, Jakob M, et al. DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia [J]. *Semantic Web*, 2015, 6 (2): 167-195.
- [8] Yu X, Ren X, Sun Y, et al. Personalized entity recommendation: A heterogeneous information network approach [C] // *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, 2014: 283-292.
- [9] Grad-Gyenge L, Filzmoser P, Werthner H. Recommendations on a knowledge graph [C] // *1st International Workshop on Machine Learning Methods for Recommender Systems*, 2015: 13-20.
- [10] Zhang F, Yuan NJ, Lian D, et al. Collaborative knowledge base embedding for recommender systems [C] // *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016: 353-362.
- [11] WU Xiyu, CHEN Qimai, LIU Hai, et al. Collaborative filtering recommendation algorithm based on representation learning of knowledge graph [J]. *Computer Engineering*, 2018, 44 (2): 226-232 (in Chinese). [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.]
- [12] Johnson CC. Logistic matrix factorization for implicit feedback data [J]. *Advances in Neural Information Processing Systems*, 2014: 27.
- [13] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data [C] // *Advances in Neural Information Processing Systems*, 2013: 2787-2795.
- [14] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion [C] // *Proc of AAAI*. AAAI Press, 2015: 2181-2187.
- [15] Harper FM, Konstan JA. The movielens datasets: History and context [J]. *ACM Transactions on Interactive Intelligent Systems*, 2016, 5 (4): 19.
- [16] Canet F, Valero MÁ, Codina L. Quantitative approaches for evaluating the influence of films using the IMDb database [J]. *Communication and Sociedad*, 2016, 29: 151-172.
- [17] Butler M, Robila S. Interface for querying and data mining for the IMDb dataset [C] // *Long Island Systems, Applications and Technology Conference*. IEEE, 2016: 1-6.
- [18] Jouili S, Vansteenberghe V. An empirical comparison of graph databases [C] // *International Conference on Social Computing*. IEEE, 2013: 708-715.
- [19] Hernando A, Bobadilla J, Ortega F. A non-negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model [J]. *Knowledge-Based Systems*, 2016, 97 (C): 188-202.
- [20] Hernández-Lobato JM, Houlsby N, Ghahramani Z. Probabilistic matrix factorization with non-random missing data [C] // *International Conference on Machine Learning*. IEEE, 2014: 1512-1520.
- [21] Huang K, Sidiropoulos ND, Swami A. Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition [J]. *IEEE Transactions on Signal Processing*, 2014, 62 (1): 211-224.
- [22] Sedhain S, Menon AK, Sanner S, et al. Autorec: Autoencoders meet collaborative filtering [C] // *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015: 111-112.