

学校代码: 10022



北京林业大学

Beijing Forestry University

本科毕业论文(设计)

基于知识图谱的电影推荐系统

Movie Recommender System Based on Knowledge Graph

刘长鑫

学 院	信息学院
--------	------

专 业	计算机科学与技术
--------	----------

指导教师	李冬梅 教授
------	--------

2021 年 6 月 1 日

独创性声明

本人声明所呈交的论文（设计）是本人在导师指导下独立进行的设计、研究工作及取得的设计、研究成果。尽我所知，除了论文（设计）中特别加以标注和致谢的地方外，论文（设计）中不包含其他人已经发表或撰写过的研究成果，本论文（设计）中没有抄袭他人研究成果和伪造数据等行为。与我共同工作的人员对本研究所做的任何贡献均已在论文（设计）中作了明确的说明并表示了谢意。

作者签名：_____ 日期：_____年____月____日

关于毕业论文（设计）使用授权的说明

本人完全了解北京林业大学有关保留、使用毕业论文（设计）的规定，即：本科生在校期间毕业论文（设计）工作的知识产权单位属北京林业大学；学校有权保留并向国家有关部门或机构送交论文（设计）的纸质版和电子版，允许毕业论文（设计）被查阅、借阅和复印；学校可以将毕业论文（设计）的全部或部分内容公开或编入有关数据库进行检索，可以允许采用影印、缩印或其它复制手段保存、汇编毕业论文（设计）。

（保密的论文在解密后应适用本授权书）

作者签名：_____ 指导老师签名：_____

日 期：_____年____月____日

北京林业大学 2021 届本科毕业论文（设计）任务书

论文(设计)题目	基于知识图谱的电影推荐系统		
	Movie Recommender System Based on Knowledge Graph		
指导教师	李冬梅	职 称	教授
题目来源	科研	题目类别	论文+设计
选题所需学生专业	计算机科学与技术		
研究的目的和意义	基于知识图谱的推荐系统能够利用有价值的外部知识作为辅助信息,解决数据的稀疏性和冷启动问题。目前,大多数推荐系统没有充分利用知识图谱中信息来辅助推荐。为了更好地利用知识图谱中的信息,基于知识图谱的电影推荐系统具有一定的研究意义。		
主要工作内容和要求	<p>内容如下:</p> <p>(1) 学习知识图谱嵌入表示,学习 TensorFlow 相关技术,将知识图谱引入推荐系统。</p> <p>(2) 将该技术在真实数据集 MovieLens 上进行验证,与其他基于知识图谱的推荐系统进行对比分析。</p> <p>要求:</p> <p>(1) 英文基础好,具有较强的英文文献阅读能力。</p> <p>(2) 自学能力强,能够自学知识图谱嵌入表示和推荐系统的相关算法。</p> <p>(3) 利用 Python 实现相关算法。</p>		

指导教师签字: 李冬梅

教研室主任签字: _____

日期: 2020 年 12 月 21 日

日期: 年 月 日

摘要

随着在线电影数量不断增加,用户选择电影的时间成本不断上升,准确的推荐算法成为了必然要求。为解决协同过滤推荐算法中的稀缺性问题与冷启动问题,研究人员用商品属性或社交网络等信息来辅助推荐算法。现有的将知识图谱作为辅助信息的推荐算法包括基于嵌入的方法和基于路径的方法,但这两种方法均存在一些缺陷,没有充分有效地利用知识图谱中的相关信息,推荐的准确度较低。

本文实现了基于知识图谱的“涟漪网络”推荐算法。“涟漪网络”算法的核心是利用现实生活中雨滴产生的涟漪在水面上不断扩散的思路,来模拟用户偏好的扩散。对于每一个用户,涟漪网络将其过往偏好作为知识图谱中的一个种子集,然后沿知识图谱中的关系路径不断地拓展用户偏好,进而发现该用户对某个候选物品以等级划分的潜在兴趣,其中多个“涟漪”重叠形成知识图谱中的用户偏好分布。该算法的实验结果和以往的 CKE、DKN、PER 等模型结果相比,表现出更优的性能。利用该算法,本文设计并实现了一个基于知识图谱的电影推荐系统,该系统包括管理员用户和普通用户,管理员能新增、编辑和删除电影与用户,普通用户能浏览、收藏与购买电影。该系统可以高效准确地为用户推荐电影,方便用户选择满足自己偏好的电影。

关键词: 知识图谱, 推荐系统, 涟漪网络, 用户偏好, 电影商店

Abstract

As the number of online movies continues to increase and the time cost for users to choose movies continues to rise, accurate recommendation algorithms have become a necessary requirement. In order to address the scarcity and cold start problem of collaborative filtering, researchers usually make use of side information, such as product attributes or social networks as side information to assist the recommendation. The existing recommendation algorithms that use knowledge graph as side information include embedding-based methods and path-based methods, but both methods have some shortcomings. They do not make full and effective use of the relevant information in the knowledge graph, and the accuracy of recommendation is relatively low.

This paper implements a recommendation algorithm, “Ripple Network”, based on knowledge graph. The core of the Ripple Network algorithm is to use the idea that the ripples produced by raindrops in real life continue to spread on the water surface to stimulate the spread of user preferences. For each user, Ripple Network uses its past preference as a seed set in the knowledge graph, and then continuously expands the user’s preferences along the relationship path in the knowledge graph, and then discovers his hierarchical potential interests concerning a certain candidate item. Multiple “ripples” overlap to form the user preference distribution in the knowledge graph. Compared with previous model results of CKE, DKN, PER, etc., the experimental results of this algorithm show better performance. Using this algorithm, this paper designs and implements a recommendation system based on the movie knowledge graph. The system includes administrator users and general users. The administrator can add, edit and delete movies and users, and general users can browse, collect and purchase films. The system can provide users with an efficient movie recommendation function, which is convenient for users to choose movies that match their preferences.

Keywords: Knowledge graph, recommender system, Ripple Network, user preferences, movie store

目录

1 绪论	1
1.1 研究背景与意义	1
1.2 国内外研究现状	2
1.2.1 推荐系统研究现状	2
1.2.2 知识图谱的研究现状	2
1.2.3 基于知识图谱的推荐系统研究现状	4
1.3 本文的主要研究内容	6
1.4 论文结构	6
2 相关理论和技术	8
2.1 知识图谱	8
2.1.1 知识图谱的概念及发展进程	8
2.1.2 知识图谱的构建方法	8
2.1.3 知识图谱的应用	8
2.2 推荐系统	9
2.2.1 传统的推荐系统	9
2.2.2 基于知识图谱的推荐系统	9
2.3 系统开发技术	10
2.3.1 数据采集技术	10
2.3.2 数据存储技术	10
2.3.3 后端技术	11
2.3.4 前端技术	12
3 基于涟漪网络知识图谱的推荐算法	13
3.1 涟漪网络	13
3.1.1 架构	13
3.1.2 涟漪集	13
3.1.3 偏好扩散	15
3.2 学习算法	16
3.3 分析	18
3.3.1 可解释性	18
3.3.2 涟漪重叠	18

3.4 测试	18
3.4.1 数据集	18
3.4.2 基线	18
3.4.3 测试步骤	19
3.4.4 结果	19
4 基于知识图谱的电影推荐系统	20
4.1 系统整体设计	20
4.2 数据库设计	20
4.3 系统功能说明	22
4.3.1 系统导航	23
4.3.2 未登录用户	23
4.3.3 注册与登录	25
4.3.4 普通用户	25
4.3.5 管理员用户	28
4.4 电影推荐流程	28
4.5 系统安全性	29
5 研究结论和展望	33
5.1 工作总结	33
5.2 工作展望	33
参考文献	35
致谢	38

1 绪论

1.1 研究背景与意义

一直以来,电影推荐都是在线流媒体播放平台发展中的一个重要问题,做好电影推荐可以使用户能在海量电影中选择满足其偏好的电影,提高用户满意度,从而提高在线流媒体播放平台的流量转化率及购买率,并最终提高在线流媒体播放平台的经济收益。近年来,随着电影行业及互联网行业的不断发展,在线电影数量不断增加,用户在海量电影中选择满足其偏好的电影的难度不断上升,性能优异的推荐算法成为了必然要求。自从在线流媒体播放平台出现以来,人们便开始尝试利用推荐算法来提高平台流量转化率,出现了诸如协同过滤的推荐算法^[1]。但这些算法未能解决数据稀缺性及冷启动问题,并不能为在线流媒体播放平台提供良好的推荐性能。为此,人们尝试将辅助信息融入推荐算法中以解决数据稀缺性及冷启动问题^[2],并提高推荐性能。

知识图谱是一种结构化的语义知识库,被用于迅速提供对物理世界中的概念和相互关系的描述,为解决推荐问题提供了新的方法^[3],近年来受到国内外研究人员的广泛关注,成为了当前的研究热点。知识图谱通过对复杂的原始数据进行加工、处理及整合,转化成简单可靠、清晰明了的“实体,关系,实体”三元组,汇聚了大量的知识信息,从而能实现基于知识信息的响应和推理。

知识图谱常被用于作为辅助信息添加至推荐算法中以提高推荐的准确性,这已是当前的研究焦点。

近年来,随着在线流媒体播放平台的发展,电影推荐系统应运而生,它作为现代信息技术与传统电影产业相结合的产物,对流媒体播放平台的用户体验起着至关重要的作用,其可大幅度减少用户寻找满足其偏好电影的时间,并提高用户点击率与购买率,从而提高流媒体播放平台的经济收益。电影推荐系统能为用户推荐其感兴趣的电影,而减少对用户不感兴趣内容的显示,从而满足用户需求。但是,目前网络上绝大多数的电影推荐系统都使用传统的基于协同过滤的推荐算法,未能有效利用知识图谱等辅助信息,返回给用户的推荐结果中包含大量用户不感兴趣的电影,此外,还将一些本应推荐给用户的、用户感兴趣的电影丢弃了。造成这一问题的主要原因是推荐算法缺少有关新加入的无用户评分的电影以及新注册的用户的信息。

对于海量的电影数据,为了实现准确地推荐给用户其感兴趣的电影,基于协同过滤的传统推荐算法是满足了不用户需求的,特别是对新注册用户,推荐的准确度无法得到保证。所以,本文旨在以知识图谱作为辅助信息,构建一个合适的电影推荐系统,

并利用知识图谱中包含的丰富的辅助信息，最终实现一个电影推荐系统，为用户提供有效的、准确的电影推荐，从而提高用户满意度，提高平台收益。

1.2 国内外研究现状

1.2.1 推荐系统研究现状

推荐系统由 Jussi Karlgren 于哥伦比亚大学在一份技术报告中以“数字书架”的名称被首次提及，而后自 1994 年起被在 SICS 的 Jussi Karlgren、由 Pattie Maes 于 MIT 领导的研究团队、位于 Bellcore 的 Will Hill 以及同样位于 MIT 的 Paul Resnick 大规模实现并在技术性报告及出版物大量出现，以上人员与 GroupLens 的工作被授予了 2010 年 ACM 软件系统奖。

自从在 90 年代中期首批有关协同过滤的论文出现后推荐系统便成为了重要的研究领域。工业界与学术界出现了众多有关建设新的推荐系统的工作。由于该领域包含众多的研究问题及其能帮助用户解决在过多信息中提供个性化推荐的实际应用，因此研究人员对该领域的兴趣依旧很高。

当前，推荐列表在推荐系统将其产生的过程中的方式主要有两种：基于内容推荐与协同过滤推荐^[4]。基于内容推荐的算法使用一些有关物品的离散特征来推荐出拥有相似性的物品。协同过滤方法根据用户的历史行为（诸如其评价过的、点击过的、购买过的商品等）并与其他用户的相似决策结合起来建立模型，这种算法可以被用于预测用户对物品的偏好程度。这两种方法常常被同时使用。

此外，近来有越来越多的推荐算法被提出：多准则推荐系统。多准则推荐系统可以被定义为包含多标准偏好信息的推荐系统，而不是构建基于单一凭据的推荐系统。风险感知推荐系统，现存推荐系统的主要方法聚焦于基于上下文信息推荐最相关内容给用户，而没有将不相关内容打扰用户的推荐风险考虑在内。考虑推送推荐内容给用户而造成打扰用户的风险在某些情况下是需要考虑的，比如，在一个工作会议上或是在凌晨或是在晚上。而此类推荐算法为该问题提供了解决方案。

现实中，大量使用的往往是混合推荐系统。其同时结合了协同过滤、基于内容过滤以及其他方法，这种混合式的推荐方法现在被大量使用。混合推荐可以在多种方式下被实现：将基于内容的和基于协同过滤的预测模型被分开实现再将他们结合起来，亦或是将两种方法统一到同一模型中并一同实现。

1.2.2 知识图谱的研究现状

在知识表示与知识推理中，知识图谱是一个使用图结构的数据模型或拓扑结构的知识库。知识图谱常被用来存储基于内连接的对实体的描述——对象、事件、情形或抽象概念，并包括自由形式的语义。

该词汇最早在 1972 年于一个有关构建模块化指令系统的课程中被提出。在 80 年代末, 命名为知识图谱的项目主要聚焦于语义网络的设计。在 2007 年, DBpedia 和 Freebase 作为基于图的常识性知识库被创立。DBpedia 致力于从 Wikipedia 中抽取数据, 而 Freebase 也包含一系列公开的数据集。它们两者都未将自己称为“知识图谱”, 但实则构建并描述了有关知识图谱的相关概念。

在 2012 年, 谷歌正式提出了知识图谱的概念^[5], 并构建了基于 DBpedia 和 Freebase 以及其他数据源的谷歌知识图谱。他们后来将包括 CIA World Factbook, Wikidata 和 Wikipedia 的网站的 RDFa, Microdata, JSON-LD 内容从它们的索引网站上抽取并纳入知识图谱。与知识图谱相关联的实体与关系类型使用来自 scheme.org 的术语被更广泛地组织了起来。

谷歌知识图谱中非常重要的一项是“自由库”, 它是一个大型的协同工作知识库。“自由库”使用图结构作为其数据结构, 可以将其视作一个有向图。谷歌知识图谱模型便是谷歌团队利用其在“自由库”之上的技术积累开发出来的^[6]。

自此, 数个大型跨国公司开始利用它们的知识图谱, 很快知识图谱流行起来, 有免费的知识图谱, 如 DBpedia 和 NELL, 以及商业性质的知识图谱, 如谷歌知识图谱和 Microsoft Satori。这些知识图谱被成功应用于问答系统、词向量嵌入与文字分类等。在 2019 年, IEEE 将其关于“大数据”和“数据挖掘与智能计算”的年度国际会议融入有关知识图谱的国际会议。

以上是知识图谱发展的主要进程。

知识图谱一般表示为三元组的形式, 即 $G = (h, r, t)$, 其中 G 为知识图谱, h 和 t 是知识库中的实体集合, 分别为三元组的头结点与尾节点^[7]。知识图谱也可以看作是一种结点对应实体、边对应关系的有向异构图。

目前知识图谱的构建过程中主要使用了定义层次学习及事实学习。

定义层次学习是通过使用机器学习等技术, 从文字、图片等信息中提取知识表达中的定义, 并确定其上下文信息。而知识图谱本质上是属于一种精炼的异构信息网络。利用定义层次学习可以在一定程度上完成对于知识图谱的构建。

事实学习分为有监督的事实学习、半监督的事实学习以及无监督的事实学习。有监督的事实学习通过人为标注的语料信息输入以及深度学习方法来完成知识图谱的构建, 而半监督的事实学习方法使用启发式地自动标注文本, 但缺陷是训练数据集中可能含有大量的噪声数据。而无监督的学习方法主要使用基于深度学习模型的自然语言处理 (NLP) 的方法, 无须人为干预, 由训练模型自动完成信息抽取、信息整合。随着深度学习算法的发展, 目前基于无监督的事实学习逐渐成为主流^[8]。

目前出现了诸多使用以上理论方法设计的知识图谱嵌入算法: 基于翻译的

TransE^[9]、TransH^[10]、TransR^[11]和基于语义分析的 DistMult^[12]等。

目前，知识图谱在业界的应用已经取得了巨大成功^[13]：

(1) 搜索引擎技术。如前所述，知识图谱最早便是由谷歌提出以改善其搜索引擎的。在将知识图谱应用于搜索引擎之后，搜索引擎能对一些常见的搜索内容基于其使用的知识图谱信息快速给出搜索结果。同时，对于一些名词性的搜索内容，搜索引擎能使用知识图谱以知识卡片的形式给出其相关信息。研究表明，在谷歌将知识图谱技术融入其搜索引擎之后，用户点击进入维基百科的次数明显减少，说明利用知识图谱技术的搜索引擎可以更好的直接给出用户需要的信息而不需要用户再浏览其他网站。

(2) 智能问答。信息检索系统中非常常见的一种便是问答系统。问答系统在用户输入问题后能迅速给出用户需要的答案。而在将知识图谱技术应用于问答系统后，其回答的准确率有非常大的提升。目前苹果的智能助理 Siri 以及谷歌的智能助理谷歌助理（Google Assistant）都使用了知识图谱技术。

此外，知识图谱还广泛应用于金融、医疗等多个领域。

1.2.3 基于知识图谱的推荐系统研究现状

由于传统的推荐系统无法解决稀缺性问题和冷启动问题，因此研究人员企图将辅助信息加入到推荐算法中以改善推荐性能。而这类辅助信息有社交网络、用户/物品属性、图像与上下文等。

在数种类型的辅助信息中，知识图谱通常包含有更丰富的信息以及物品间的联系。图1.1中说明了知识图谱提供丰富的信息与物品间的连接，有利于提高推荐结果的准确性、多样性和可解释性。知识图谱可以从以下三个方面提高推荐性能：

(1) 知识图谱引入了实体语义相关性，这可以帮助找到实体间潜在的联系并提高推荐物品的准确性。

(2) 知识图谱由各种类型的关系构成，这对合理地扩展用户兴趣并增加推荐商品的多样性是有益的。

(3) 知识图谱连接用户的历史记录及推荐的商品，因此提高了推荐系统的可解释性。

将知识图谱作为辅助信息加入至推荐系统中是当前的热点问题，目前基于知识图谱构建推荐系统的方法主要有以下几类：

(1) 基于嵌入的方法。基于嵌入的方法通过知识图谱嵌入计算算法将知识图谱中的实体信息与关系信息映射为低维向量空间中的向量表示（嵌入表示）。而后推荐算法利用该嵌入表示来完成相应的计算，并给出推荐结果。目前将知识图谱中嵌入表示信息作为辅助信息的推荐模型有 CKE^[14]、DKN^[15]以及 SHINE^[16]等。CKE 是微软在 KDD2016 年发表的，其模型结构在原有系统过滤得到 U, V 向量的基础上，将物品的

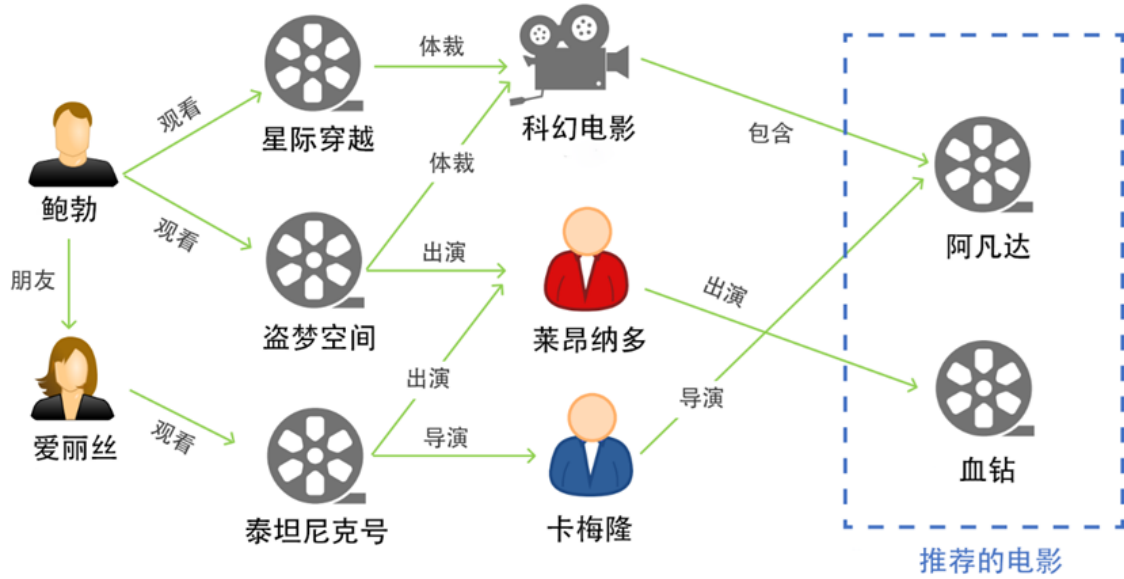


图 1.1 基于知识图谱的电影推荐系统

Fig. 1.1 Knowledge graph enhanced movie recommendation system

嵌入与其他描述信息相结合，这些信息主要有：采用 TransR 算法嵌入得到的知识图谱，图谱内每个实体嵌入表示被提取为物品的结构化向量信息。采用 SDAE 模型得到物品描述性文本的文本嵌入表示。采用 SCAE 模型得到物品相关图像的视觉嵌入表示。DKN 是之前同样由微软团队在 WWW2018 会议上发表的。它是一个主要针对新闻推荐任务提出的框架，知识图谱用于辅助计算新闻标题的嵌入表示。DKN 提出对新闻标题内每一个关键实体，在知识图谱内找到其实体嵌入和上下文嵌入。SHINE 设计深度自编码器来嵌入语义网络，社交网络并进行推荐。其为用户-物品的交互使用自动编码器并计算用户感兴趣的概率。此外，嵌入表示可以包含实体的独热编码^[17]、词汇集^[15]、上下文信息^[2]或属性^[16]，而选择哪一种嵌入向量的计算方式取决于其应用场合。

（2）基于路径的方法。基于路径的方法将知识图谱作为异构信息网络，主要是通过知识图谱中的路径关系信息来辅助推荐算法完成推荐相关的计算工作。目前基于知识图谱中路径信息的推荐模型有 HIN^[18]、RKGE^[19]等。HIN 是 2017 年在 KDD 上提出的，与 PER^[20]类似，都将知识图谱作为异构信息网络。它针对 Yelp 和 Amazon 数据集分别设计了元路径，并得到了 L 个评估函数。并在建模时使用 FM 模型，使用 FM 损失函数组进行梯度下降，并最终得到推荐结果。RKGE 模型使用了基于循环神经网络的方法。在人为定义元路径并抽取出所有路径之后，对每种元路径都使用循环神经网络模型来抽取并推理其路径中所包含的信息。

（3）混合式方法。基于嵌入的方法未能充分利用实体间的关系模式信息，而基于

路径的方法仅考虑了实体间的语义连通信息，未能有效使用用户/物品本身的语义信息。因此，目前出现了混合式方法，即同时结合了基于嵌入的方法与基于路径的方法的优点。混合式方法是当前的研究热点与焦点，使用该方法的典型模型有 AKUPM^[21]和 RCoLM^[22]等。AKUPM 是通过注意力机制增强的推荐模型，RCoLM 则是基于联合学习并以任务为导向的推荐算法。本文实现的电影推荐系统基于文献 [23]，其所使用的算法模型是也属于混合式方法。

1.3 本文的主要研究内容

本文在推荐算法、知识图谱和基于知识图谱的推荐算法的相关研究基础上，做了以下三方面的研究：

(1) 对推荐系统所需的数据进行采集和处理，使用“MovieLens 1M Dataset”作为数据集，此外还从 IMDb 及豆瓣网爬取了相关电影数据并进行处理，作为本文的研究对象。

(2) 根据文献 [23] 提出的算法，实现了基于知识图谱的涟漪网络推荐算法，此算法能根据用户的历史行为为用户进行电影推荐。不同于文献 [23] 中仅使用用户评分计算用户偏好，本文在用户偏好的计算过程中还结合了用户收藏，这在一定程度上缓解了冷启动问题并改进了推荐性能。此外，将涟漪网络算法与其他基于知识图谱的推荐算法（DKN^[15]、CKE^[14]、PER^[20]、SHINE^[16]、LibFM^[24]和 Wide&Deep^[25]等）进行了性能比较。

(3) 实现了一个基于知识图谱的电影推荐系统，该系统能够根据用户的历史行为（评分、收藏等）来为用户进行电影推荐。该系统分为管理员、未登录用户、普通登录用户。管理员能增加、修改和删除电影和普通用户；未登录用户能根据电影分类查看电影列表以及查看电影详情；普通登录用户除了能进行未登录用户的所有操作外，还能购买、收藏及为电影评分。

1.4 论文结构

本论文一共分为五个章节。

第 1 章是绪论。简明介绍知识图谱作为辅助信息在推荐算法中的作用以及推荐算法对在线流媒体播放平台的作用，同时介绍了推荐系统、知识图谱和基于知识图谱的推荐系统的发展和应用，并说明了本文的主要工作。

第 2 章是相关理论和技术。介绍了目前知识图谱和推荐系统的研究现状，并对传统的推荐算法与加入辅助信息的推荐算法作了较为细致的分类与阐述。同时还介绍了本文实现的电影推荐系统所使用的技术栈，诸如数据库、前端架构、后端架构等。

第3章是涟漪网络。介绍了涟漪网络算法，以及计算相关实体嵌入的过程中使用的深度学习算法，并对以上算法进行了讨论。最后，还对涟漪网络推荐算法的准确度进行了测试。

第4章是电影推荐系统。介绍了以涟漪网络算法为基础构建的电影推荐系统：介绍了系统架构、系统功能、电影推荐流程以及如何保证该系统的安全性。

第5章是研究结论和展望。总结了涟漪网络算法的设计与实现、电影推荐系统的功能，并指出了其中可以改进的地方。

2 相关理论和技术

2.1 知识图谱

2.1.1 知识图谱的概念及发展进程

知识图谱，是一种知识可视化或知识领域映射地图，是显示知识发展进程与结构关系的一系列三元组（实体，关系，实体），用可视化技术描述知识信息及其载体，挖掘、分析、建立、绘制和可视化知识以及实体之间的相互关系。

知识图谱是谷歌于 2012 年提出的，目的是为了增强其搜索引擎。而后微软于 2013 年 7 月发布了 Microsoft Satori 知识图谱。除传统搜索服务提供商之外，包括 Apple, Facebook, IBM 等互联网企业也在探索该领域。此外，业界也出现了社区构建的开源知识图谱，如 DBpedia, NELL 和 Wikidata 等。以 Freebase、WordNet 和 GeneOntology 等为例的知识图谱已成为支持人工智能相关应用的非常重要的资源。

2.1.2 知识图谱的构建方法

知识图谱是由多实体结点和不同关系边构成的多关系图，一个边是一个事实三元组（头结点，关系，尾结点），记为 (h, r, t) 。而知识图谱的构建便是寻找这样的三元组。

在过去十年间，出现了诸多构建大型知识图谱的方法，然而，还没有清晰准确的构建知识图谱的范式。主要有两个困难：（1）知识图谱是符号化的逻辑系统而知识图谱的应用常常包含在连续空间中的计算；（2）在一张知识图谱上进行知识聚合是困难的。

目前，知识图谱主要通过以下 3 个步骤来完成构建：

（1）信息抽取。即从各类型的源数据中抽取出实体（概念）、实体间的关系，并在此基础上形成确切的知识表达。

（2）知识融合。对新获得的知识进行整合，以完成消歧义。

（3）知识加工。经过质量核查后将融合后的新知识的部分加入至知识图谱中。

2.1.3 知识图谱的应用

知识图谱以更接近人类认知的方式为互联网的信息表达提供了一种新的方式，而且提供了一种更益于组织与利用海量数据的方式。当前知识图谱主要用于推荐系统、搜索引擎、个人助理以及问答系统。

2.2 推荐系统

推荐系统是内容过滤系统的一个子集，被广泛使用于多个领域，它向用户推荐满足其偏好的物品，并通常具有为视频或音乐服务生成推荐清单的表现形式。

2.2.1 传统的推荐系统

传统的推荐系统主要分为基于内容的推荐系统、基于协同过滤的推荐系统和混合推荐系统^[26]。基于内容的推荐系统主要用于文本相关的项目，因为内容常常是使用关键字来描述的。该算法通过计算相关物品的离散特征，然后推荐具有与用户历史偏好物品类似特征的有关物品。基于协同过滤的推荐系统主要通过将具有与目标用户相似特征的用户所交互过的物品推荐给目标用户。

基于内容的推荐与基于协同过滤的推荐的有关区别可以通过比较两个流行音乐推荐系统的实现方式看出。(1) 潘多拉音乐使用一个基于内容推荐的推荐算法。它根据艺人或歌曲的属性来生成一个包含相似风格歌曲的电台，并且该电台的内容会根据收听用户的反馈进行调整。当用户对一首歌曲“不感兴趣”时该算法将弱化该歌曲的一些属性；而当用户喜欢一首歌曲时，将强化一些属性。并且该算法会根据该属性调整歌曲的顺序，若属性的值越过一个阈值时推荐算法便将某一歌曲从列表中删除。(2) “终级 fm”使用协同过滤的推荐算法。它记录用户经常收听的乐队和歌手，然后与其它用户的有关行为进行比较，建立一个电台，并以此推荐歌曲。“终级 fm”会给用户推荐其他具有相似特征用户的播放列表（并保证目标用户未收听过）。

基于内容的推荐与基于协同过滤的推荐各有优缺点。潘多拉音乐所使用的基于内容推荐的推荐算法是根据物品本身的性质来进行推荐的，因此不需要用户信息就可以有较好的准确度。但该算法严重依赖物品本身的特性，因此局限性较大，推荐的内容都是与种子集相关的，推荐结果的多样性较低。而“终级 fm”使用的协同过滤算法需要根据用户与物品的交互来生成推荐结果，因此需要大量的用户数据，存在数据稀缺性问题与冷启动问题。

2.2.2 基于知识图谱的推荐系统

基于内容推荐与协同过滤推荐两者均存在一些局限性，为了提高推荐的准确性，解决传统推荐算法的数据稀缺性与冷启动问题，研究者将一些辅助信息加入至推荐算法中，通常这些辅助信息包括上下文信息、用户或物品的属性、图片和社交网络^[27-28]。

而随着知识图谱的发展，将知识图谱作为辅助信息来提高推荐系统的性能已经成为了热门的研究方向。将知识图谱作为辅助信息加入推荐算法的优点有：

(1) 提高推荐系统的准确性。知识图谱可以表示不同实体间的关系，可以将物品及其相关属性对应于知识图谱之中，从而推荐算法可以理解物品之间的关系。此外，

用户与用户间的关系信息也可以映射至知识图谱之中，从而推荐算法可以更准确地分析物品与物品间的关系、用户与物品之间的关系。

(2) 提高推荐系统的可解释性。由于推荐结果依照知识图谱中的关系信息，因此可以由知识图谱中的关系路径得到推荐系统给出推荐结果的原因。

目前主要的基于知识图谱的推荐算法有基于嵌入的方法、基于路径的方法以及混合式方法：

(1) 基于嵌入的方法。基于嵌入的方法使用知识图谱的信息来完善实体的嵌入表示。为了将知识图谱中的信息添加至推荐算法中来辅助推荐，需要使用知识图谱嵌入表示算法 (Knowledge Graph Embedding, KGE) 表计算实体嵌入 (实体嵌入指由知识图谱中的信息得到的在低维向量空间中的向量表示)。KGE 算法有 TransE^[9]、TransH^[10]、TransR^[11]和 DistMult^[12]等。而推荐算法利用该嵌入表示来进行相关计算，从而对用户进行物品推荐。

(2) 基于路径的方法^[29-31]。基于路径的方法将知识图谱视为异构信息网络。而推荐系统利用该异构信息网络寻找实体间的关系，从而完成推荐。

(3) 混合式方法。基于嵌入的方法未能充分利用实体间的关系模式信息，而基于路径的方法仅考虑了实体间的语义连通信息，未能有效利用用户/物品本身的语义信息。混合式方法结合了基于嵌入的方法与基于路径的方法的优点，该方法是当前的研究热点与焦点，使用该方法的典型模型有 AKUPM^[21]和 RCoLM^[22]等。本文实现的算法属于混合方法。

2.3 系统开发技术

2.3.1 数据采集技术

电影推荐系统使用的数据来自“MovieLens 1M Dataset”¹数据集，此外还从 IMDb²及豆瓣网³爬取了相关电影数据并进行处理作为本文的研究对象。

在数据采集的过程中，使用 Scrapy 框架进行了数据爬取，爬取的数据包括电影名称、电影封面、预告片图片、电影情节介绍、导演、演员以及剧作家。

最终，总计爬取了 3684 部电影的数据。其中，从 IMDb 爬取了 3494 条电影数据，从豆瓣爬取了 190 条电影数据（由于豆瓣网限制了每 IP 访问量故爬取的数据较少）。

2.3.2 数据存储技术

本系统使用 MongoDB 与 Neo4j 存储数据。

(1) MongoDB

¹<https://grouplens.org/datasets/movielens/1m/>

²<https://www.imdb.com/>

³<https://www.douban.com/>

MongoDB 是一个新兴的非关系型存储的分布式存储数据的数据库,属于 NoSQL,与 MySQL 等传统的关系型数据库相比,它提供更好的高并发支持,具有高可靠性的优点。与此同时,得益于其基于文档储存的特点,相比较传统的关系型数据库,它的存储结构更灵活。

在本系统中, MongoDB 存储普通用户数据、管理员数据、电影数据等。

(2) Neo4j

Neo4j 是一个具有高性能的图数据库,它将结构化的数据信息储存在网络上而不是存储在表中。它具有健壮和成熟的数据库的所有特点。虽然 Neo4j 是一个新兴的数据库,但它已在具有超过 1 亿节点、关系和属性的产品中得到了应用,充分体现了其高性能、高可靠性的特点。

在本系统中,最终需存储的图结点有 182011 个,需存储的边有 1241995 条。如果将它们存储在传统的关系型数据库中,会因大量的连接查询导致极大的性能开销,表现为查询耗时久。Neo4j 对图数据处理做了优化,因此查询等操作可以在较短的时间内完成,故本系统将知识图谱数据存储在 Neo4j 中而不是关系型数据库中。

2.3.3 后端技术

本系统使用 Flask 框架作为网站后端框架。Flask 是一个 Python 编写的轻量级微框架。它具有轻量、便捷、可扩展等特点。系统使用 Flask 框架充分利用了其便捷、可扩展以及开发便捷的特点,与本系统要求相符。

系统使用表现层状态转换 (REST) 风格的应用程序接口 (API) 作为前后端的交互接口。表现层状态转换 (Representational State Transfer, REST) 是一种基于 HTTP 协议的软件交互体系结构样式,该协议是由 Roy Thomas Fielding 博士在 2000 年的博士论文中引用的^[32],目的是促进不同的软件/程序的开发。在网络上互相传输数据。表示层状态转换基于超文本传输协议 (HTTP) 的一组约束,该协议是一种旨在提供万维网服务的软件构造样式。符合或兼容此体系结构样式 (简称为 REST 或 RESTful) 的网络服务允许客户端发出使用统一资源标识符访问和操作网络资源的请求,这些资源与一组预定义的操作一致。因此,表示层状态转换提供了 Internet 上计算机系统之间相互使用资源的互操作性。与其他类型的网络服务 (例如 SOAP 服务) 相比,它们使用自己定义的一组操作来访问网络资源。

得益于本系统使用了 REST 风格的 API,本系统能更有效地使用缓存来提升响应速度,同时前后端间的通讯的无状态性可以让不同的服务器处理一连串请求中不同的请求,提高了本系统的可扩展性。与此同时,得益于 REST 风格语义化的接口,使前后端交互接口更清晰明了,使接口的使用者能更高效便捷地进行相关的开发工作。

2.3.4 前端技术

本系统前端使用 Angular 框架。Angular（通常指“Angular 2+”或“Angular v2 及以上”）是一个基于 TypeScript 的开源网络应用框架，它的开发由 Google 的 Angular 团队领导同时也有个人及其他公司维护。Angular 完全从相同团队开发的 AngularJS 重新编写而来。Angular 常常作为 MEAN 技术栈的一部分。MEAN 技术栈指 MongoDB 数据库、Express.js 网络服务器框架、Angular/AngularJS 以及 Node.js 服务器运行时。不同于 MEAN 技术栈，本系统将其中的“E”（指 Express.js 网络服务器框架）替换为了 Flask 网络服务器框架。

通过使用 Angular 框架，本系统具有一个现代化、模块化、响应式的网页架构。同时借助于 Angular Material Components 官方组件库，本系统前端 UI 符合材料设计（Material Design）。

3 基于涟漪网络知识图谱的推荐算法

3.1 涟漪网络

本算法基于文献 [23] 实现，并在其基础上进行了一定的改进：文献 [23] 中计算用户偏好时并未考虑用户收藏，本文在计算用户偏好时，将用户收藏考虑在内，这在一定程度上缓解了数据稀缺性问题和冷启动问题并改进了推荐性能。

3.1.1 架构

涟漪网络的总体架构如图3.1所示，图上方的知识图谱中展示了由用户交互产生的涟漪。涟漪网络以一个用户 u 和一个电影 v 作为输入，并输出用户 u 与电影 v 之间产生交互的概率。对输入用户 u 而言，其历史交互记录 V_u 是知识图谱中的种子集，而后沿着知识图谱中的关系边形成多个涟漪集 S_u^k ($k = 1, 2, \dots, H$)。第 k 个涟漪集是种子集 V_u 经过 k 跳得到的知识三元组。然后迭代地利用这些涟漪集与电影 v 的嵌入表示（黄色的块）计算出用户 u 对电影 v 的响应（绿色的块），最后结合得到用户的最终嵌入表示（灰色的块）。最终，利用用户 u 与电影 v 的嵌入表示计算出用户 u 对电影 v 感兴趣的预测概率 y_{uv} 。

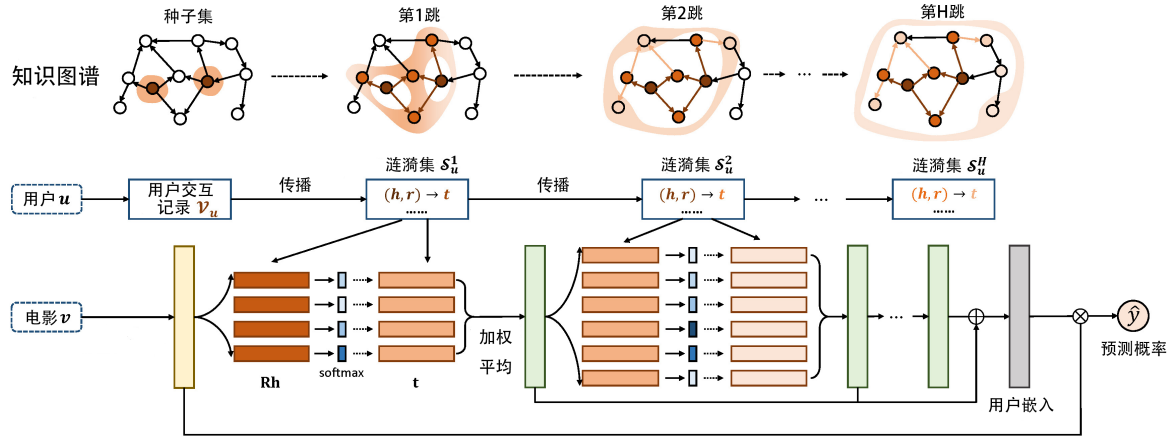


图 3.1 涟漪网络的总体架构

Fig. 3.1 The overall framework of the Ripple Network

3.1.2 涟漪集

知识图谱常常含有丰富的信息事实与实体间的联系。比如，图3.2（图中不同颜色的圆圈表示不同跳数的涟漪集，越浅的蓝色代表种子集与该区域内实体的关联程度越低）中电影“阿甘正传”与“罗伯特·泽米吉斯”相连，它们之间的联系为“罗伯特·泽米吉斯”是电影“阿甘正传”的导演。而“回到未来”也与“罗伯特·泽米吉

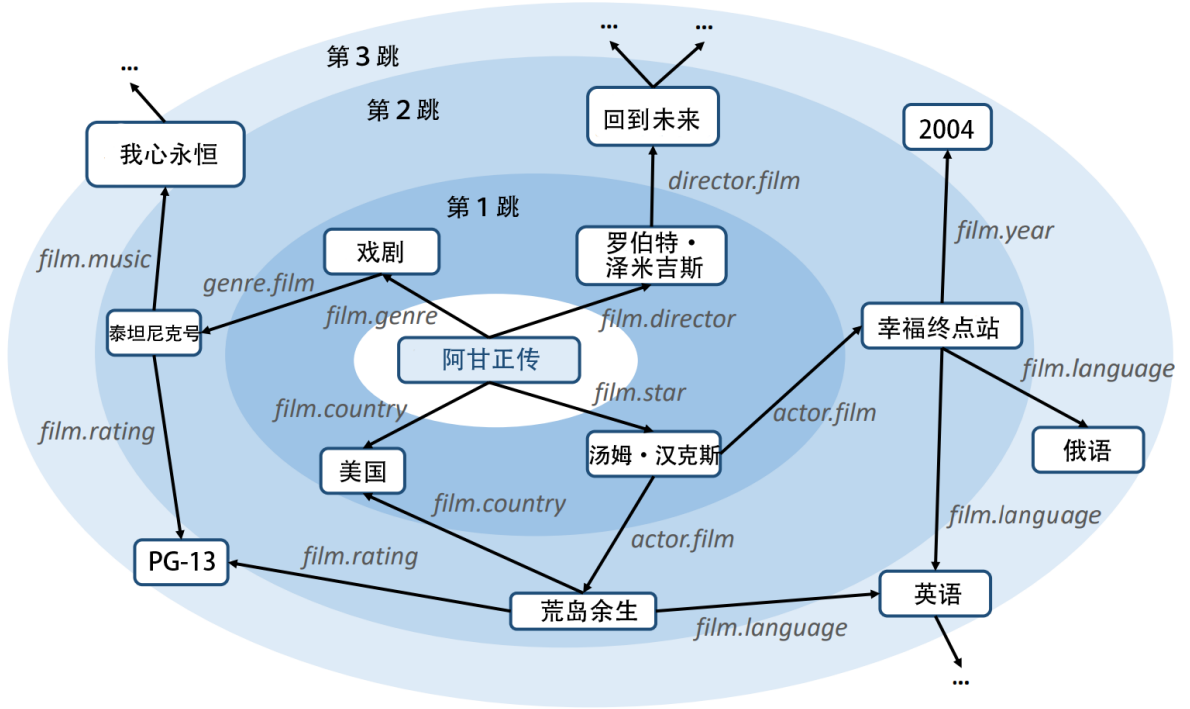


图 3.2 电影知识图谱中由“阿甘正传”激发的涟漪集

Fig. 3.2 Sets of ripples of “Forest Gump” in Knowledge Graph of movies

斯”相连。因此，如果一个用户与电影“阿甘正传”交互过，则他很有可能也对“回到未来”感兴趣。为了描述用户在知识图谱中分层次的潜在偏好集，递归定义用户 u 的 k 跳相关实体如下：

定义 1（相关实体集） 给定交互矩阵 Υ 与知识图谱 G ，则用户 u 的 k 跳相关实体集的定义为式(3.1)。

$$E_u^k = \{t | (h, r, t) \in G \text{ 且 } h \in E_u^{k-1}\}, k = 1, 2, \dots, H \quad (3.1)$$

式(3.1)中， $E_u^0 = V_u = \{v | y_{uv} = 1\}$ 是用户的历史偏好集（历史偏好集是指用户在最高为 5 的评分中给出的评分 ≥ 4 的物品以及用户已加入收藏的物品），可以看作是用户 u 在知识图谱中的种子集。

相关实体集可以看作是用户的历史偏好集在知识图谱中的自然扩展。给定相关实体集的定义，以下定义用户 u 的 k 跳涟漪集：

定义 2（涟漪集） 用户 u 的 k 跳涟漪集可以看作是知识图谱中以 E_u^{k-1} 为起点的知识三元组，定义如式(3.2)所示。

$$S_u^k = \{(h, r, t) | (h, r, t) \in G \text{ 且 } h \in E_u^{k-1}\}, k = 1, 2, \dots, H \quad (3.2)$$

“涟漪”这个词有两重意思：（1）对由多个雨点产生的真实涟漪的模拟，用户对

电影的潜在兴趣集在知识图谱中由近及远地传递。这一过程如图3.2所示。(2) 用户的潜在兴趣随着知识图谱中传递的跳数 k 的增大逐渐递减。图3.2中蓝色的变浅显示了潜在兴趣递减的过程。

一个可能出现的问题是在跳数 k 增加的过程中涟漪集的大小可能过大。为了解决这个问题，注意到：

(1) 在真实使用的知识图谱中大量的实体是沉没实体，意思是它们只有传入链路而没有传出链路。

(2) 在电影推荐的具体情境下，关系可以限制在情境相关的分类中以减少涟漪集的数量并提高实体间的相关度。

(3) 最大跳数 H 通常在实际应用中不会过大，因为离用户历史偏好集较远的实体会带来更多的错误推荐而不是更丰富的推荐结果。

(4) 在涟漪网络中，我们可以对一个固定大小的邻集而不是对完整的涟漪集采样从而更进一步地减小计算工作量。设计这样的采样器是一个重要的工作，尤其是非统一的采样器能更好地捕获用户的分等级的潜在兴趣。

3.1.3 偏好扩散

传统的协同过滤算法是通过学习用户与物品间的潜在联系来完成推荐，而在涟漪网络算法中，这一过程是通过偏好扩散完成的：对每个用户，涟漪网络将他的过往兴趣视为知识图谱中的种子集，然后沿知识图谱中的路径不断地拓展用户的潜在兴趣集，进而得到按等级划分的关于候选物品的潜在兴趣集。我们利用现实生活中的由雨滴产生的涟漪在水面上扩散来模拟偏好扩散的过程，其中多个“涟漪”重叠形成基于知识图谱的用户偏好分布。

如图3.1，每部电影都有一个嵌入表示 v ， $v \in \mathbb{R}^d$ ，其中 \mathbb{R} 是实数集， d 是嵌入表示向量的维数。给定电影的嵌入表示 v 以及用户 u 跳涟漪集 S_u^1 ，可以利用电影 v 、 S_u^1 中的三元组中头节点 $head_i$ 以及该三元组中的关系 r_i 来计算出电影 v 和实体 $head_i$ 之间的相关度，如式(3.3)所示。

$$p_i = \text{softmax}(v^T R_i h_i) = \frac{\exp(v^T R_i h_i)}{\sum_{(h,r,t) \in S_u^1} \exp(v^T R h)} \quad (3.3)$$

式(3.3)中， $R_i \in \mathbb{R}^{d \times d}$ 与 $h_i \in \mathbb{R}^d$ 分别是关系 r_i 和头节点 h_i 的嵌入表示。相关度 p_i 为向量空间 R_i 中的电影 v 与实体 h_i 之间的相似度。同时还考虑了关系信息的嵌入表示矩阵在计算电影 v 与实体 h_i 之间的相似度，因为不同电影-实体对之间的相似度在不同关系的场景下可能有所不同。比如，“阿甘正传”与“回到未来”在导演方面有很高的相似度，但它们在体裁或剧作家方面可能没有联系。

在完成相似度计算之后，将 S_u^1 中三元组的尾节点根据计算得到的相似度进行加权求和，得到向量 o_u^1 ，如式(3.4)所示。

$$o_u^1 = \sum_{(h_i, r_i, t_i) \in S_u^1} p_i t_i \quad (3.4)$$

式(3.4)中， $t_i \in \mathbb{R}^d$ ， t_i 是事实三元组中的尾节点。向量 o_u^1 可以视为用户偏好集 V_u 关于电影 v 的 1 次响应。与协同过滤算法类似的是，它根据用户交互过的物品来表示用户嵌入而不是使用单独的参数来表示用户，从而减少了参数数量。通过式(3.3)与式(3.4)，用户的偏好从其交互过的电影（即种子集）成功在知识图谱中进行了一次扩散，这称为涟漪网络中的偏好扩散。

可以将式(3.3)中的 v 替换为 o_u^1 ，我们可以再次进行偏好扩散以得到用户的 2 次响应 o_u^2 。以此类推，最终我们可以将该用户的偏好集扩散至 H 跳，得到用户的涟漪集 $S_u^i, i = 1, \dots, H$ 。从中可以得到用户的多次响应 o_u^1, o_u^2, o_u^H 。用户 u 关于电影 v 的嵌入表示是通过结合所有的不同响应并计算得到的，如式(3.5)所示。

$$u = o_u^1 + o_u^2 + \dots + o_u^H \quad (3.5)$$

尽管最后一跳的用户响应中理论上包含了所有之前得到的信息，但是对 o_u^k 进行求和仍然是有必要的，因为之前得到的用户响应可能在偏好传递的过程中稀释了，因此仅使用最后一跳的响应不能很好地表示用户嵌入。最终，利用用户嵌入与电影嵌入，计算出用户对电影感兴趣的概率，如式(3.6)所示。

$$y_{uv} = \zeta(u^T v) \quad (3.6)$$

其中， $\zeta(x) = \frac{1}{1+\exp(-x)}$ 是 sigmoid 函数。

3.2 学习算法

在涟漪网络算法中，我们的目标是通过式(3.6)求出用户对电影感兴趣的概率。但是，为了将其计算出来我们需要先求出所有物品、用户以及关系的嵌入表示，如第3.1.3节中所述，用户的嵌入表示可以通过其涟漪集来代替，以此消元。但物品及关系的嵌入表示却是需要求出来的。

为了求出物品及关系的嵌入表示，这里使用最大后验概率估计的方法。

我们的目标是在已经得到知识图谱 G 以及用户与电影的交互矩阵 Υ 的情况下最大化模型参数 Γ ，如式(3.7)所示。

$$\max p(\Gamma|G, \Upsilon) \quad (3.7)$$

式(3.7)中, 参数 Γ 包含了全部的实体、关系和电影的嵌入表示。可以将式(3.7)等价于最大化式(3.8)。

$$p(\Gamma|G, \Upsilon) = \frac{p(\Gamma, G, \Upsilon)}{p(G, \Upsilon)} \propto p(\Gamma) \cdot p(G|\Gamma) \cdot p(\Upsilon|\Gamma, G) \quad (3.8)$$

根据贝叶斯公式, 在式(3.8)中, 第一项 $p(\Gamma)$ 代表模型参数的先验概率。设 $p(\Gamma)$ 为均值为 0, 方差为对角协方差矩阵的正态分布, 如式(3.9)所示。

$$p(\Gamma) = N(0, \lambda_1^{-1} I) \quad (3.9)$$

式(3.8)中, $p(G|\Gamma)$ 是给定 Γ 的知识图谱的最大似然函数。最近, 有研究者提出了多种计算知识图谱嵌入的方法, 包括基于距离的方法和基于语义匹配的方法。在本算法中, 使用三路张量分解的方法来建立知识图谱嵌入的最大似然函数, 如式(3.10)。

$$p(G|\Gamma) = \prod_{(h,r,t) \in E \times R \times E} p((h,r,t)|\Gamma) = \prod_{(h,r,t) \in E \times R \times E} N(I_{h,r,t} - h^T R t, \lambda_2^{-1}) \quad (3.10)$$

式(3.10)中, 若 $I_{h,r,t} \in G$ 则标志 $I_{h,r,t}$ 等于 1, 否则为 0。式(3.8)中的第三项是给定 Γ 与知识图谱的似然函数, 可以看作伯努利分布的累乘, 如式(3.11)。

$$p(\Upsilon|\Gamma, G) = \prod_{(u,v) \in \Upsilon} \zeta(u^T v)^{y_{uv}} \cdot (1 - \zeta(u^T v))^{1-y_{uv}} \quad (3.11)$$

对式(3.8)取负对数, 有如式(3.12)所示的损失函数:

$$\begin{aligned} \min L &= -\log(p(\Upsilon|\Gamma, G) \cdot p(G|\Gamma) \cdot p(\Gamma)) \\ &= \sum_{(u,v) \in \Upsilon} -(y_{uv} \log \zeta(u^T v) + (1 - y_{uv} \log(1 - \zeta(u^T v)))) \\ &\quad + \frac{\lambda_2}{2} \sum_{r \in R} \|I_r - E^T R E\|_2^2 + \frac{\lambda_1}{2} (\|V\|_2^2 + \|E\|_2^2 + \sum_{r \in R} \|R\|_2^2) \end{aligned} \quad (3.12)$$

式(3.12)中, V 和 E 是所有电影与实体的嵌入表示, I_r 是在知识图谱中对关系 r 的标志向量 I 的分向量。在式(3.12)中, 第一项是交互矩阵 Υ 与涟漪网络的估测值之间的交叉熵, 第二项是真实的知识图谱分向量 I_r 以及人为构建的指示矩阵 $E^T R E$ 之间的均方误差, 最后一项是为防止过拟合加入的正则项。

直接求解上式来得到参数 Γ 是不可能的, 因此可以使用随机梯度下降算法递归地优化损失函数来求解模型参数, 而后再计算参数 Γ 的损失函数的梯度, 并根据采样得到的一小批数据反向传递, 然后更新参数并最终得到参数 Γ 。

3.3 分析

3.3.1 可解释性

可解释的推荐系统旨在阐释为什么用户会对一件物品感兴趣，这帮助提升用户对推荐结果的满意度以及对推荐系统的信任。对推荐结果的解释通常基于标签、语义分析等。因为涟漪网络探索用户基于知识图谱的兴趣，因此它提供了一种基于知识图谱中的关系路径来阐述推荐结果的全新方式。比如，在图3.2中，当用户对“幸福终点站”感兴趣，则该用户也可能对“荒岛余生”感兴趣。因为在知识图谱中，“汤姆·汉克斯”与“幸福终点站”相连，关系是演员，而“汤姆·汉克斯”与“荒岛余生”也相连，关系也是演员，换句话说，“荒岛余生”与“幸福终点站”有相同的演员。这便解释了用户对“幸福终点站”和“荒岛余生”同时感兴趣的原因。涟漪网络算法通过在知识图谱中寻找与用户交互过的电影（种子集）相连的物品，并不断扩散，最终确保推荐结果具有较高的准确性。

3.3.2 涟漪重叠

在涟漪网络中，一个可能的问题是涟漪集中的电影非常多，从而在偏好传递的过程中不可避免地导致用户的真实潜在偏好信息被稀释。然而，用户点击记录中不同的电影常常高度重叠（从种子集出发到达一部电影常常有不止一条路径），这在很大程度上避免了真实潜在偏好信息被稀释的问题。比如，在图3.2中，如果一个用户喜欢“阿甘正传”，则他也可能喜欢“荒岛余生”。在该知识图谱中，从“阿甘正传”到“荒岛余生”有两条路径：“阿甘正传-U.S.-荒岛余生”与“阿甘正传-汤姆·汉克斯-荒岛余生”，这正是涟漪重叠的表现。

3.4 测试

3.4.1 数据集

本测试使用“MovieLens 1M Dataset”数据集。该数据集由电影信息、用户信息以及用户对电影的评价三部分组成。其中，含有电影数据 3883 条、用户数据 6040 条以及 1000209 条用户对电影的评价数据。因该数据集数据量适中，数据准确可靠，因此在推荐系统的性能测试中被广泛使用。

本测试使用的知识图谱来自 Microsoft Satori，是依据“MovieLens 1M Dataset”中的电影名称从 Microsoft Satori 中提取相关的节点的与关系数据得到的。

3.4.2 基线

本测试中，会将本算法的测试结果与以下算法的相比较：

DKN^[15]是由微软团队在 WWW2018 会议上发表的。它是一个主要针对新闻任务提出的框架，知识图谱用于辅助计算新闻标题的嵌入表示。DKN 提出对新闻标题内每一个关键实体，在知识图谱内找到其实体嵌入和上下文嵌入。

CKE^[14]是微软在 KDD2016 年发表的，其模型结构在原有系统过滤得到 U, V 向量的基础上，将物品的嵌入与其他描述信息相结合，这些信息主要有：采用 TransR 算法计算知识图谱嵌入表示，知识图谱内每个实体嵌入表示被提取为物品的结构化向量信息。采用 SDAE 模型得到物品描述性文本的文本性嵌入表示。采用 SCAE 模型得到物品相关图像的视觉嵌入表示。

SHINE^[16]设计深度自编码器并嵌入语义网络以及社交网络来进行推荐。其为用户-物品的交互使用自动编码器并刻画用户感兴趣的概率。

PER^[20]是以基于路径的方法来将知识图谱作为辅助信息中的比较经典的算法。其提出的元路径可以为推荐系统提供可靠的方向，但是需要使用者了解领域内知识，进行人为路径设计。

LibFM^[24]是一个广泛使用的在 CTR 场景中的分解推荐模型。

Wide&Deep^[25]是一个结合线性路径的推荐模型。类似于 LibFM，我们将用户、物品及实体的嵌入表示作为其输入。

3.4.3 测试步骤

在涟漪网络中，设置跳数 $H = 2$ 。根据实验结果，较大的跳数几乎无法提高性能却会造成较大的计算开销。我们将数据划分为训练集、评估集与测试集，按照 6:2:2 的比例进行分配。实验进行 5 次，计算准确度以及 AUC 然后取平均值。

3.4.4 结果

测试结果如表3.1中所示，总体上涟漪网络算法的性能最佳，其次是 Wide&Deep 算法，说明他们可以充分利用知识图谱中的有效信息来辅助推荐算法。而表现最差的是 PER 算法，这可能是由于手工定义的元路径在电影推荐方面效果较差。

表 3.1 在兴趣预测计算中的 AUC 和准确度

Table 3.1 AUC and ACC in interest prediction

算法	AUC	准确度
涟漪网络	0.899	0.835
DKN	0.655	0.589
CKE	0.796	0.739
SHINE	0.778	0.732
LibFM	0.892	0.812
Wide&Deep	0.903	0.822

4 基于知识图谱的电影推荐系统

4.1 系统整体设计

如第2.3节中所述,本系统使用的数据来自“MovieLens 1M Dataset”、IMDb 和豆瓣网。其中 IMDb 和豆瓣网的数据是使用爬虫爬取得到的。最后再使用 Python 脚本对这些数据进行处理加工,并导入 MongoDB 数据库和 Neo4j 数据库。

本系统使用 MongoDB 和 Neo4j 存储数据。其中, MongoDB 作为非关系型的新兴 NoSQL 数据库,以灵活的非结构化的方式存储普通用户数据、管理员数据、电影数据等,并提供高并发的性能支持与可分布式存储的扩展性。而在以高性能著称的图数据库 Neo4j 中,存储用于推荐算法使用的电影知识图谱。

本系统后端使用 Python 编写的 Flask 框架, Flask 框架是轻量级的微框架,以高可扩展性著称,为本系统的后端网页应用服务器提供支持。在 Flask 中,基于 REST 风格构建 API 以供前端使用, REST 风格的语义化 API 使后端 API 的构建与前端 API 的使用简单方便。

本系统前端使用 Angular 作为前端网页框架。Angular 是基于 TypeScript 的网页框架,以模块化及可重用的组件设计著称。Angular 为本系统提供了模块化的网页设计,使本系统前端逻辑清晰,易于维护。此外,本系统使用了 Angular 材料组件库(Angular Material Components),得益于此,本系统遵循材料设计(Material Design)语言,使视觉传达风格简洁美观,具有响应式动画与过渡效过、光线与阴影等。

4.2 数据库设计

如前所述,本系统的 MongoDB 数据库用于存储普通用户数据、管理员数据、电影数据等,而 Neo4j 作为高性能的图数据库,用于存储知识图谱。以下逐一说明 MongoDB 数据库与 Neo4j 数据库的存储结构。

不同于 MySQL 等关系型数据库, MongoDB 数据库作为非结构化存储的 NoSQL 数据库,没有表、列与行的概念,而只有集合与文档的概念,即在数据库中存储集合(一定程度上对应于关系型数据库的表),在集合中存储文档(一定程度上对应于关系型数据库的行),而关系型数据库中的列则对应于 MongoDB 数据库中文档的属性。

本系统在 MongoDB 数据库中建立 4 个集合,分别用于存储用户、管理员、体裁与电影数据。因其中的数据为非结构化数据,因此无法用表格的形式给出,目前习惯上以 JSON 形式给出数据的逻辑结构,以下以 JSON 形式来表示存储结构:

以下为普通用户集合的数据结构，用户集合中的文档存储用户 ID、哈希处理后的密码、其进行过评分的电影（包括电影 ID、评分及时间）以及由推荐算法服务器写入的推荐列表、已购买电影及心愿单：

```
{
  _id: Integer,
  password: String,
  rating: Array([
    {
      movieId: Integer,
      rating: Integer,
      timestamp: Integer
    },
    ...
  ]),
  recommendation: Array([Integer, ...]),
  bought: Array([Integer, ...]),
  wishlist: Array([Integer, ...])
}
```

以下为管理员集合的数据结构，其存储结构较简单，管理员集合中的文档存储管理员 ID、哈希处理后的密码：

```
{
  _id: String,
  password: String
}
```

以下为体裁集合的数据结构，其存储结构较简单，体裁集合中的文档存储体裁名称：

```
{
  _id: String
}
```

以下为电影集合的数据结构，电影集合中的文档存储电影 ID、电影封面、体裁、价格、评分累加和、评分总数量、预告片图片、剧情介绍、导演、剧本作家及演员。需

要说明的是评分累加和以及评分总数量对应于用户集合中的评分，但只起缓存作用，定期计算。（经测试表明，每次用户访问时计算的开销较大，用户得到响应的时间过长）：

```
{
  _id: Integer,
  title: String,
  cover: String,
  genres: Array([String, ...]),
  price: Float,
  ratingSum: Integer,
  ratingCount: Integer,
  trailer_image_url: String,
  storyline: Array([String, ...]),
  directors: Array([String, ...]),
  writers: Array([String, ...]),
  actors: Array([String, ...]),
}
```

Neo4j 数据库用于存储推荐算法使用的知识图谱，含有 182011 个结点、1241995 条边，其数据结构可表示为：

```
node: actor | country | director | film | genre | language
      | person_or_entity_appearing_in_film | rating | star
      | writer
relationship: actor.film | director.film | film.country
              | film.director | film.genre | film.language | film.rating
              | film.star | film.writer | genre.film
              | person_or_entity_appearing_in_film.film | writer.film
edge = (node) - [relationship] -> (node)
```

4.3 系统功能说明

本系统用户角色分为未登录用户、普通用户与管理员用户，其用例说明如图4.1。

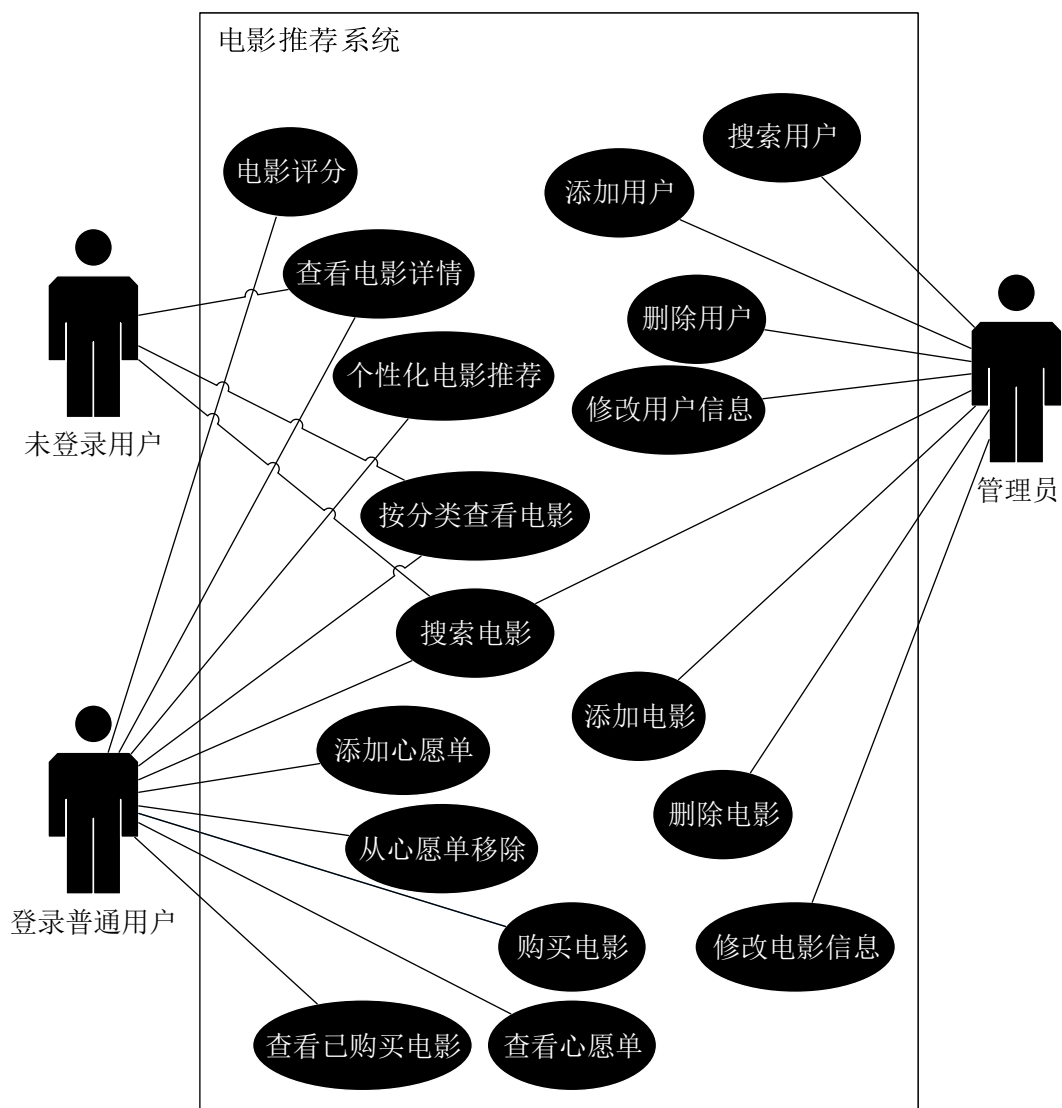


图 4.1 系统功能用例图

Fig. 4.1 Use case diagram for the system

4.3.1 系统导航

本系统使用浮动侧边栏作为导航方式，如图4.2。点击侧导航栏右下角的固定的按钮可以将浮动侧边栏设为固定，再次点击后将取消固定。

4.3.2 未登录用户

未登录用户能进行以下操作：

(1) 接收随机的电影推荐

未登录用户首页随机显示 50 部电影，如图4.3。

(2) 按分类查看电影

未登录用户可以根据电影的分类来查看电影，如图4.4。

(3) 查看电影详情



图 4.2 系统侧导航栏（管理员）

Fig. 4.2 Side navigation panel of the system (for administrators)

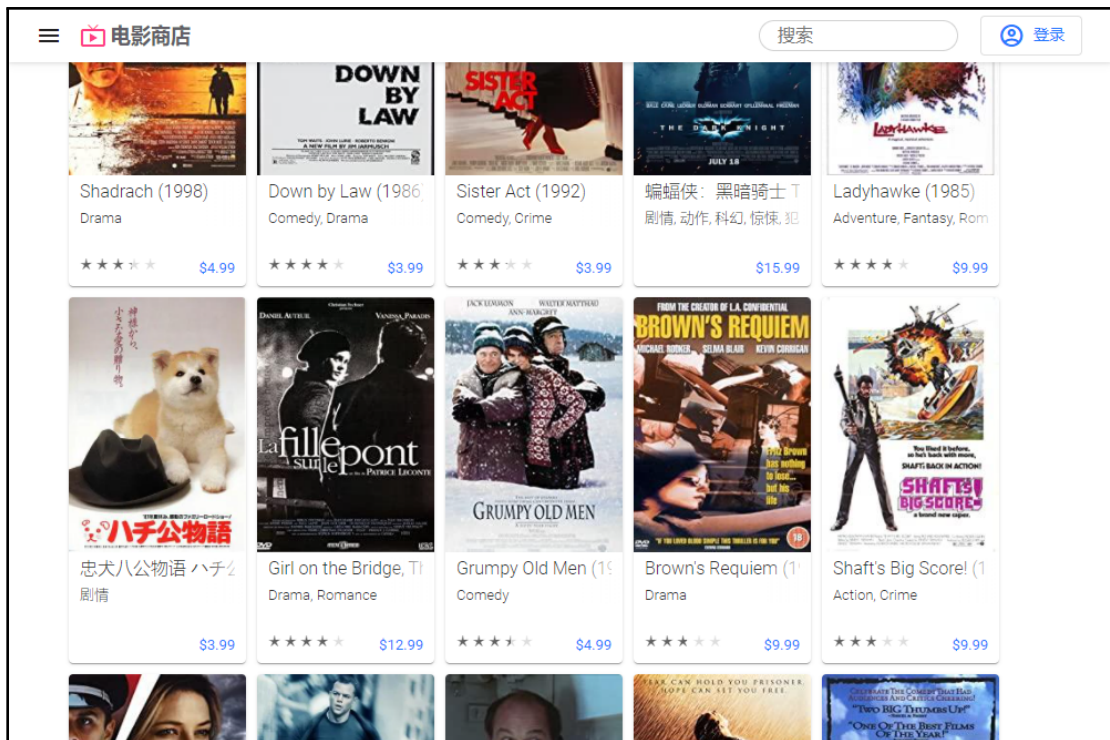


图 4.3 未登录用户首页

Fig. 4.3 Index page for anonymous users

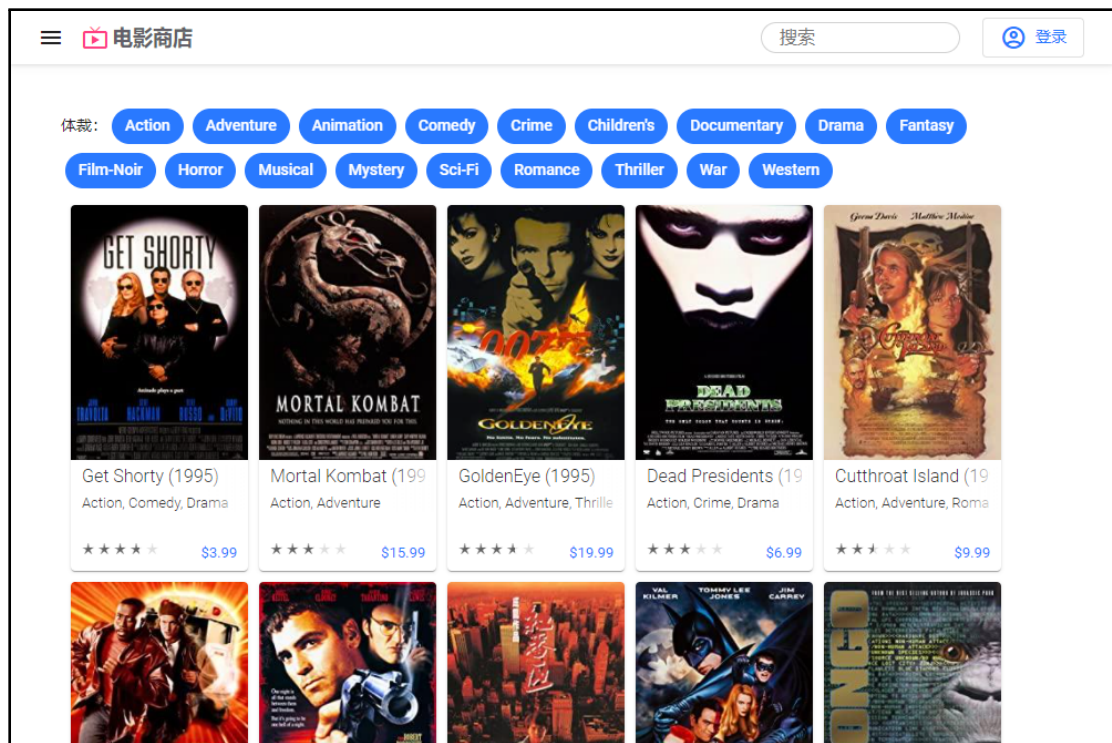


图 4.4 未登录用户分类页面

Fig. 4.4 Category page for anonymous users

未登录用户可以查看电影详情，如图4.5。点击页面上的“添加至心愿单”和“购买”会跳转至登录界面。

(4) 搜索电影

未登录用户可以使用应用栏右侧的搜索框搜索电影，如图4.6。本系统支持模糊搜索，与此同时，随着用户搜索内容的不断输入，系统会在搜索框下方显示候选的匹配词条，方便用户直接点击查看，此外，用户也能通过回车跳转至完整的搜索结果页面。

4.3.3 注册与登录

(1) 注册

本系统对普通用户开放注册功能，并能提供用户名是否含不合法字符、是否与已注册用户名冲突、密码是否符合复杂度要求等的检测。

(2) 登录

本系统提供登录功能，并能根据不同的登录错误类型相应地作出响应。

4.3.4 普通用户

普通用户具备未登录用户的所有功能。此外，还具备以下功能：

(1) 添加心愿单

已登录用户能在电影详情页面添加电影至心愿单。

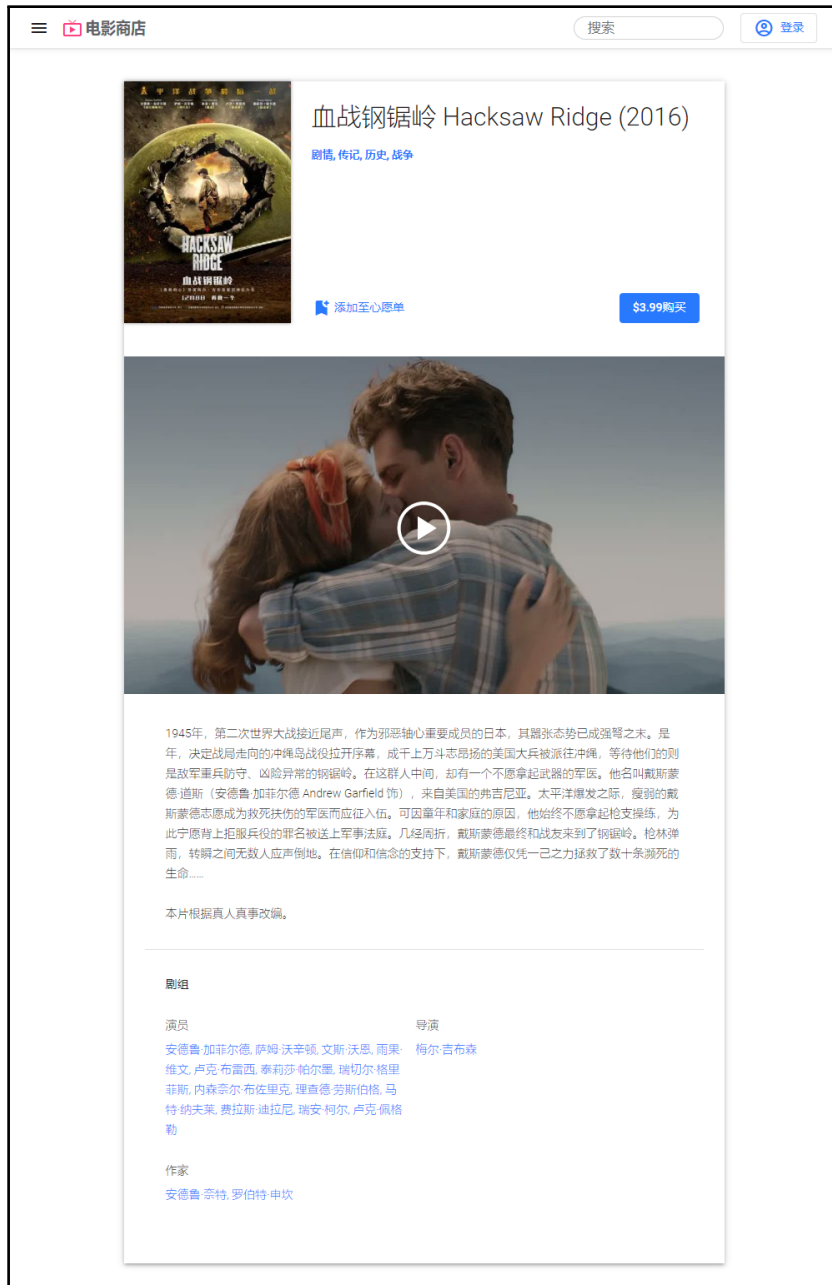


图 4.5 未登录用户电影详情页面

Fig. 4.5 Movie details page for anonymous users

(2) 购买

已登录用户能在电影详情页面购买电影。

(3) 评分

已登录用户能在电影详情页面给电影评分。电影评分功能位于电影详情页面，如图4.7所示。

(4) 查看已购买的电影

已登录用户能通过侧导航栏跳转至查看已购买的电影的页面。

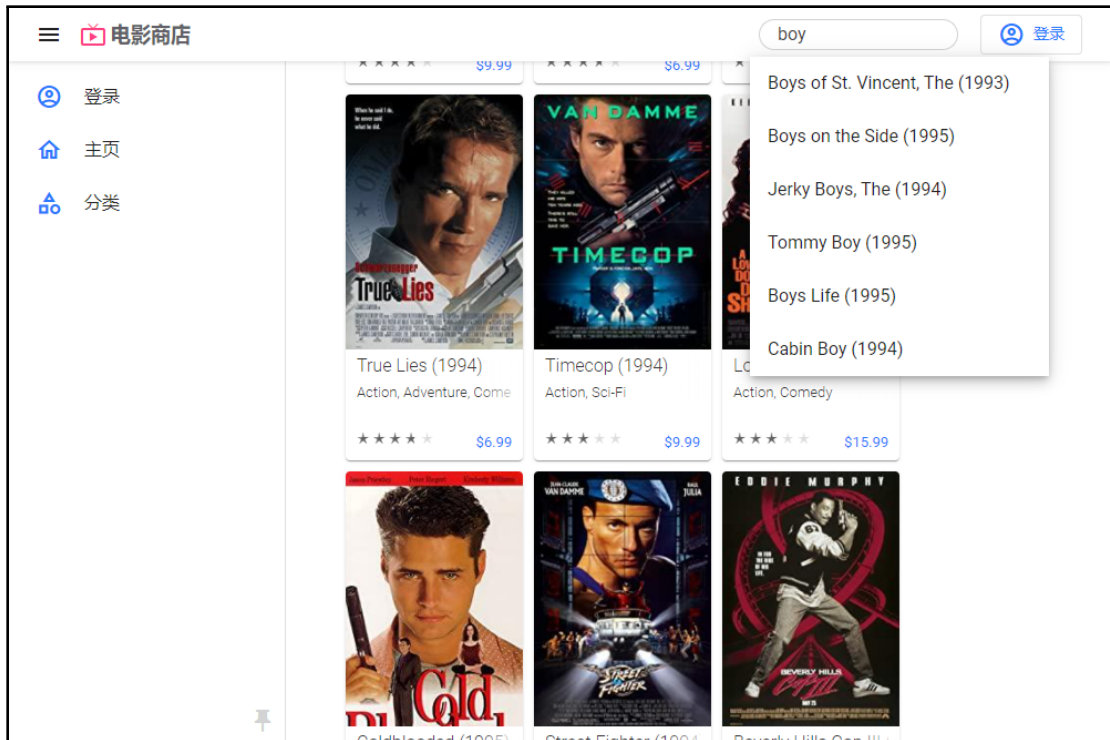


图 4.6 未登录用户电影搜索界面（侧导航栏已固定）

Fig. 4.6 Movie search page for anonymous users (side navigation panel pinned)

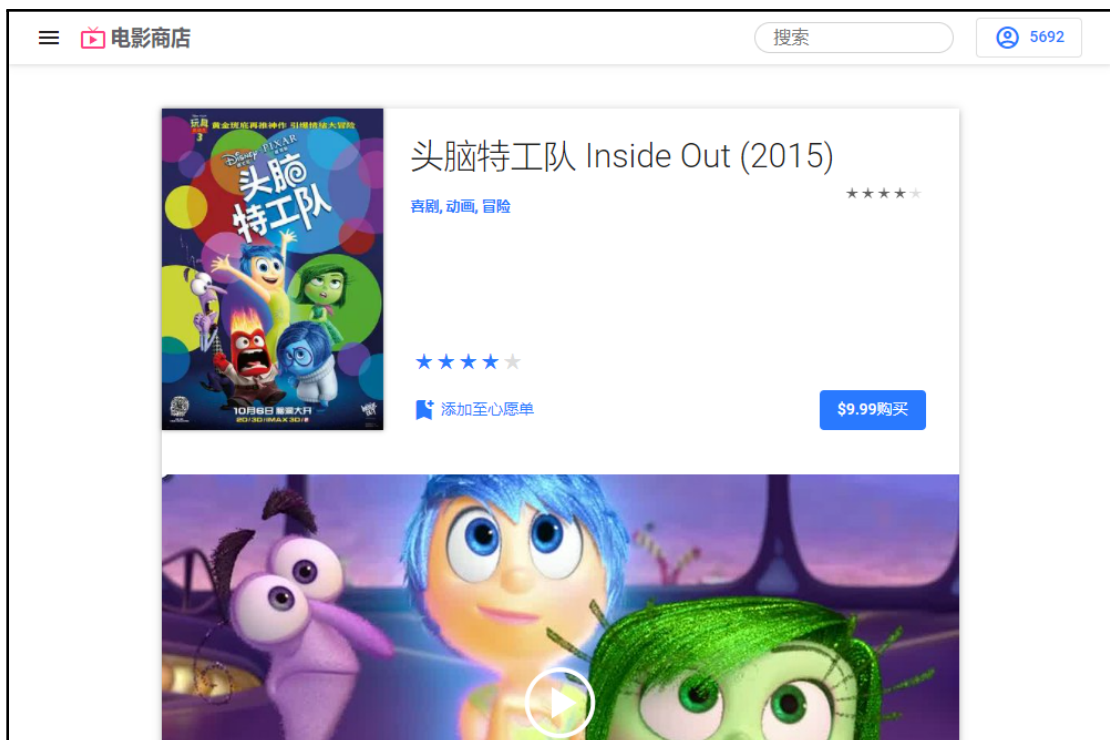


图 4.7 电影评分页面

Fig. 4.7 Rating page for movies

(5) 查看已添加心愿单的电影

已登录用户能通过侧导航栏跳转至查看已添加心愿单的电影的页面。

4.3.5 管理员用户

(1) 管理电影信息

管理员能增加电影、删除电影与修改电影信息，如图4.8所示。



名称	体裁	价格	评分	操作
Grumpier Old Men (1995)	Comedy, Romance	\$19.99	3.02	 
Nixon (1995)	Drama	\$15.99	3.54	 
Sense and Sensibility (1995)	Drama, Romance	\$6.99	4.03	 
Four Rooms (1995)	Thriller	\$3.99	3.34	 
Toy Story (1995)	Animation, Children's, Comedy	\$4.99	4.15	 
Tom and Huck (1995)	Adventure, Children's	\$6.99	3.01	 
American President, The (1995)	Comedy, Drama, Romance	\$19.99	3.79	 
Ace Ventura: When Nature Calls (1995)	Comedy	\$15.99	2.48	 
Jumanji (1995)	Adventure, Children's, Fantasy	\$12.99	3.20	 
Balto (1995)	Animation, Children's	\$15.99	3.26	  
Father of the Bride Part II (1995)	Comedy	\$12.99	3.01	 
Shanghai Triad (Yao a yao dao waipo qiao) (1995)	Drama	\$9.99	3.65	 

图 4.8 管理电影页面

Fig. 4.8 Movie administration page

(2) 管理用户信息

管理员能增加用户、删除用户与修改用户密码。

(3) 管理知识图谱

管理员能增加、删除、修改以及查找知识图谱中的结点与关系，如图4.9所示。该界面中的结点及关系可以以动态的方式呈现，同时支持以填写选项的方式以及使用 Cypher 语句的方式来增加、删除、修改以及查找知识图谱中的结点与关系。当鼠标悬浮于某一节点或关系之上时，将显示有关这一节点或关系的有关信息。

4.4 电影推荐流程

本系统的推荐流程分为离线推荐与实时推荐，如图4.10所示。

其中，离线推荐使用第3章所述的涟漪网络算法，此算法是基于文献 [23] 实现的。不同于文献 [23] 中仅使用用户评分计算用户偏好，本推荐系统在用户偏好的计算过

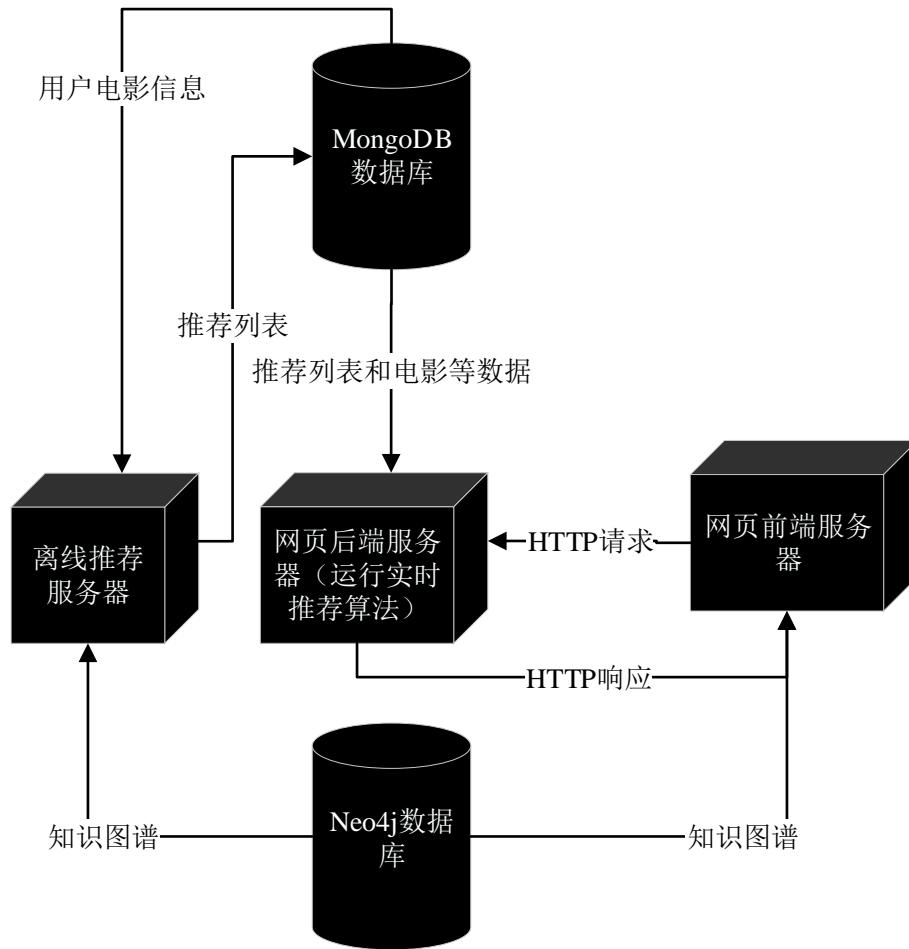


图 4.10 电影推荐流程

Fig. 4.10 Movie recommendation procedure

证系统的安全性，本系统的总体安全性设计如图4.11所示。

在用户登录时，Flask 后端网页服务器会生成一个长期 JSON 网络令牌和一个临时 JSON 网络令牌，令牌中存储有用户 id 以及过期时间，并且该令牌使用非对称加密算法加密，其中含有由 Flask 后端网页服务器颁发的签名信息，这确保了 JSON 网络令牌不会被伪造。

前端网页服务器在接收到令牌后会将令牌存储在浏览器的 localStorage 中，并在每一个接下来的请求头中附上该令牌，而后端服务器只有在该令牌有效（指确实为后端网页服务器颁发且该另令牌尚未过期）的情况下才会继续执行有关请求。与此同时，若该请求是用户内容相关的，则后端服务器还会检测令牌中的 id 是否与请求中使用的 id 相同。若不满足以上任意一项，则后端网页服务器返回未授权错误。

其中，若后端网页服务器发现令牌过期，则会将该信息发送给前端网页服务器，前端网页服务器将会将长期令牌发送给后端服务器以刷新（renew）临时令牌。

此外，本系统还使用了哈希算法对密码进行哈希处理，数据库中的密码全部为哈

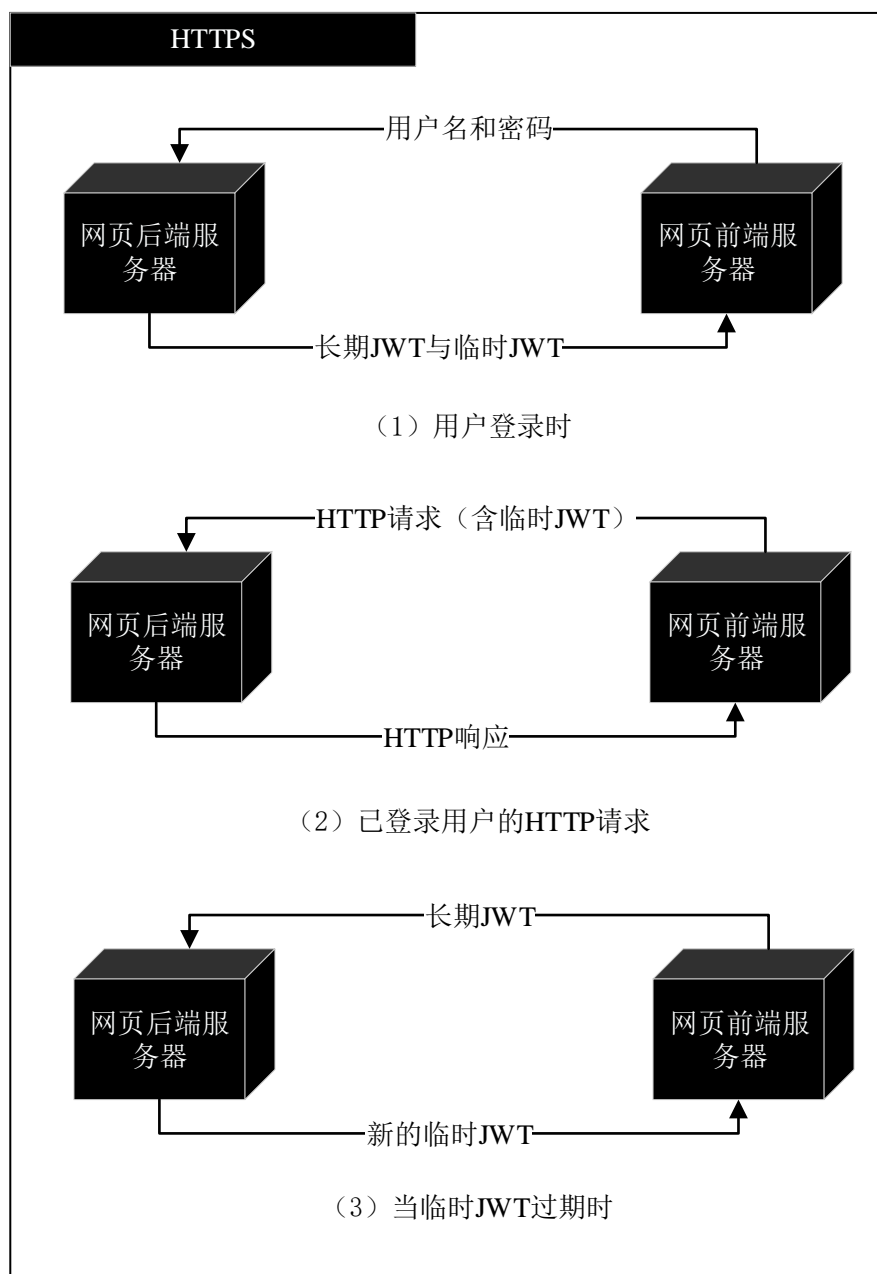


图 4.11 系统总体安全性设计

Fig. 4.11 Overall security design for the system

希处理后的密码。同时，用户登录过程是后端网页服务器使用原密码与哈希处理后的密码进行比较，而没有开放哈希处理后的密码与哈希处理后的密码进行比较的接口，这保障了本系统的安全性。这在同时令网站仅允许 HTTPS（HTTPS-only）网络连接的情况下可以充分保证本系统的安全性。

需要指出的是，虽然本系统在用户登录与注册时密码在前端没有进行哈希处理，但是本系统在部署时使用 HTTPS-only 的连接，在这种情况下，所有的数据传输都处于加密状态，避免了中间人攻击的可能，因此明文传输密码而不使用哈希处理是安全

的，在实际使用的过程中可行。而本系统的后端网页服务器对前端网页服务器传入的明文密码进行哈希处理的目的是避免数据库中存入明文的密码。如果数据库中存入明文的密码，则当数据库中用户密码泄露时攻击者可以轻易使用泄露的密码登录本系统。而如上所述，本系统后端只开放了接收明文密码的接口，即只将数据库中哈希处理后的密码与用户请求中的明文密码进行对比，即使攻击者获取了用户哈希后的密码，要计算其对应的明文密码也是困难的，甚至在明文密码足够复杂的情况下是不可能的。因此，以上措施保证了本系统的安全性。

5 研究结论和展望

5.1 工作总结

随着互联网的发展,用户可选择的电影数量不断增加,为了让用户快速找到感兴趣的电影,各种推荐算法应运而生。这些算法存在推荐准确性较低、数据稀缺性以及冷启动问题。本文针对此现状,以电影推荐为研究对象,使用知识图谱作为辅助信息,利用涟漪网络算法实现了电影推荐系统。具体地说,主要的研究工作如下:

(1) 使用基于 Scrapy 框架的爬虫从 IMDb 和豆瓣网上爬取了 3684 条电影数据。其中,从 IMDb 爬取了 3494 条电影数据,从豆瓣网爬取了 190 条电影数据(由于豆瓣网限制了每 IP 访问量故爬取的数据较少)。这些电影数据包括电影封面图片、电影情节介绍、电影预告片图片、电影演员列表、导演以及剧本作家等信息。

(2) 根据文献 [23] 实现了基于知识图谱的涟漪网络推荐算法,通过使用“MovieLens 1M Dataset”数据集以及从 IMDb 和豆瓣网上爬取的 movie 数据,实现了基于用户心愿单和用户评分并以知识图谱为辅助信息的推荐算法。不同于文献 [23] 中仅使用用户评分计算用户偏好,本文在用户偏好的计算过程中还结合了用户收藏,这在一定程度上缓解了冷启动问题并改进了推荐性能。并对实现的推荐算法进行了实验,计算了其 AUC 和准确度两个关键的性能指标,将该指标与 DKN^[15]、CKE^[14]、PER^[20]、SHINE^[16]、LibFM^[24]以及 Wide&Deep^[25]算法的进行对比。并以此发现,涟漪网络算法的性能最优。

(3) 基于涟漪网络算法实现了基于知识图谱的电影推荐系统。该系统为未登录用户提供按分类查看电影、查看电影详情的功能;为普通用户提供电影推荐、按分类查看电影、查看电影详情、电影评分、将电影加入心愿单以及购买电影功能;为管理员提供增加、删除、修改、查找电影及用户的功能。同时使用 JSON 网络令牌、HTTPS 以及哈希化密码等手段保障系统的安全性。

5.2 工作展望

为了给用户提供准确的推荐电影,需要不断优化并改进推荐过程。推荐系统的基础是用户历史偏好集、知识图谱信息以及推荐算法本身。因此有效地记录用户历史偏好集,完善知识图谱信息并改进推荐算法本身是提高推荐结果的重要因素。本文虽然对基于知识图谱的推荐算法有了一定的研究,并且将其应用于电影推荐系统中,但是由于时间和能力所限,该系统仍存在着不足的地方,主要体现在以下两个方面:

(1) 需要完善知识图谱的信息

知识图谱是本文使用的涟漪网络算法的基础，但本文所使用的知识图谱的信息还不够完善，因此在一定程度上影响了涟漪网络算法的性能。

(2) 涟漪网络算法有待进一步改进

尽管涟漪网络算法相比于传统的推荐算法在准确度上有所提升，但涟漪网络算法目前仅适用于离线推荐，而无法用于实时推荐，这使它的适用场景受到了限制。未来可考虑改造该算法，使其能满足实时推荐的需求。

对于以上提及的问题，未来还需要更深入地学习有关知识图谱、推荐算法、深度学习的有关知识，对系统进行改进，从而使其更完善。

参考文献

- [1] HE X, LIAO L, ZHANG H, et al. Neural collaborative filtering[A]//BARRETT R. Proceedings of the 26th international conference on world wide web[C]. New York City: Association for Computing Machinery, 2017: 173-182.
- [2] SUN Y, YUAN N J, XIE X, et al. Collaborative intent prediction with real-time contextual data[J]. ACM Transactions on Information Systems (TOIS), 2017, 35(4): 1-33.
- [3] ZOU X. A survey on application of knowledge graph[A]//JPCS. Journal of Physics: Conference Series[C]. Bristol: IOP Publishing, 2020: 12-16.
- [4] JAFARKARIMI H, SIM A T H, SAADATDOOST R. A naive recommendation model for large databases[J]. International Journal of Information and Education Technology, 2012, 2(3): 216-219.
- [5] SINGHAL A. Introducing the knowledge graph: things, not strings[J]. Official google blog, 2012, 5: 1-16.
- [6] 漆桂林, 高桓, 吴天星. 知识图谱研究进展[J]. 情报工程, 2017, 3(1): 004-025.
- [7] 徐增林, 盛泳, 潘贺, 荣王雅芳. 知识图谱技术综述[J]. 电子科技大学学报, 1(4): 589-606.
- [8] 李涓子, 侯磊, 等. 知识图谱研究综述[J]. 山西大学学报 (自然科学版), 2017(03): 454-459.
- [9] BORDES A, USUNIER N, GARCIA-DURAN A, et al. Translating embeddings for modeling multi-relational data[A]//BURGES C. Neural Information Processing Systems (NIPS)[C]. Lake Tahoe: Curran, 2013: 1-9.
- [10] WANG Z, ZHANG J, FENG J, et al. Knowledge graph embedding by translating on hyperplanes[A]//AAAI. Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence[C]. Menlo Park: AAAI Press, 2014: 1112-1119.
- [11] LIN H, LIU Y, WANG W, et al. Learning Entity and Relation Embeddings for Knowledge Resolution[J]. Procedia Computer Science, 2017, 108: 345-354.
- [12] YANG B, YIH W T, HE X, et al. Embedding Entities and Relations for Learning and Inference in Knowledge Bases[J]. ArXiv e-prints, 2014: 1412-1423.
- [13] 曹倩, 赵一鸣. 知识图谱的技术实现流程及相关应用[J]. 情报理论与实践, 2015, 38(12): 13-18.
- [14] ZHANG F, YUAN N J, LIAN D, et al. Collaborative knowledge base embedding for recommender systems[A]//KRISHNAPURAM B. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining[C]. New York: Association for Computing Machinery, 2016: 353-362.

- [15] WANG H, ZHANG F, XIE X, et al. DKN: Deep knowledge-aware network for news recommendation[A]//CHAMPIN P A. Proceedings of the 2018 world wide web conference[C]. Republic: International World Wide Web Conferences Steering Committee, 2018: 1835-1844.
- [16] WANG H, ZHANG F, HOU M, et al. Shine: Signed heterogeneous information network embedding for sentiment link prediction[A]//CHANG Y. Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining[C]. New York: Association for Computing Machinery, 2018: 592-600.
- [17] KOREN Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[A]//LI Y. Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining[C]. New York: Association for Computing Machinery, 2008: 426-434.
- [18] ZHAO H, YAO Q, LI J, et al. Meta-graph based recommendation fusion over heterogeneous information networks[A]//MATWIN S. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. New York: Association for Computing Machinery, 2017: 635-644.
- [19] SUN Z, YANG J, ZHANG J, et al. Recurrent knowledge graph embedding for effective recommendation[A]//PERA S. Proceedings of the 12th ACM Conference on Recommender Systems[C]. New York: Association for Computing Machinery, 2018: 297-305.
- [20] YU X, REN X, SUN Y, et al. Personalized entity recommendation: A heterogeneous information network approach[A]//CHANG Y. Proceedings of the 7th ACM international conference on Web search and data mining[C]. New York: Association for Computing Machinery, 2014: 283-292.
- [21] TANG X, WANG T, YANG H, et al. AKUPM: Attention-enhanced knowledge-aware user preference model for recommendation[A]//TEREDESAL A. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining[C]. New York: Association for Computing Machinery, 2019: 1891-1899.
- [22] CAO Y, WANG X, HE X, et al. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences[A]//LIU L. The world wide web conference[C]. New York: Association for Computing Machinery, 2019: 151-161.
- [23] WANG H, ZHANG F, WANG J, et al. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems[A]//CUZZOCREA A. Proceedings of the 27th ACM International Conference on Information and Knowledge Management[C]. New York: Association for Computing Machinery, 2018: 417-426.
- [24] RENDLE S. Factorization machines with libfm[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2012, 3(3): 1-22.

- [25] CHENG H T, KOC L, HARMSSEN J, et al. Wide & deep learning for recommender systems[A]//KARATZOGLOU A. Proceedings of the 1st workshop on deep learning for recommender systems[C]. New York: Association for Computing Machinery, 2016: 7-10.
- [26] 黄立威, 江碧涛, 吕守业, 等. 基于深度学习的推荐系统研究综述[J]. 计算机学报, 2018, 41(7): 1619-1647.
- [27] 常亮, 张伟涛, 古天龙, 等. 知识图谱的推荐系统综述[J]. 智能系统学报, 2019, 14(2): 207-216.
- [28] 秦川, 祝恒书, 庄福振, 等. 基于知识图谱的推荐系统研究综述[J]. 中国科学: 信息科学, 2020, 50(7): 937-956.
- [29] LIN Y, LIU Z, LUAN H, et al. Modeling Relation Paths for Representation Learning of Knowledge Bases[J]. ArXiv e-prints, 2015: 1506-1515.
- [30] GUU K, MILLER J, LIANG P. Traversing Knowledge Graphs in Vector Space[A]//LLUÍS MÀRQUEZ J S, Chris Callison-Burch. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing[C]. Lisbon: Association for Computational Linguistics, 2015: 318-327.
- [31] TOUTANOVA K, LIN X V, YIH W T, et al. Compositional learning of embeddings for relation paths in knowledge base and text[A]//ERK K. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)[C]. Berlin: Association for Computational Linguistics, 2016: 1434-1444.
- [32] FIELDING R T. Architectural styles and the design of network-based software architectures[M]. Berkeley: University of California, Irvine Irvine, 2000: 1-4.

致谢

论文的撰写工作已经接近尾声，在本文的撰写过程中我要特别感谢李冬梅老师的指导。在毕业论文完成期间，李冬梅老师多次对我的毕业论文提出意见与建议。同时，她多次询问论文的撰写进度，更是体现了她对学生无微不至的关怀。每每当我在论文编写中遇到艰深晦涩之处时，她都能不失时机而恰到好处地给予我最有用的提示与建议，使我的毕业论文得以顺利完成。

其次，我要感谢我本科期间曾经为我上课的所有老师。比如，由我的概率论老师为我打下的数学基础才使我得以用最大后验概率估计法完成嵌入表示的学习算法，蓝海洋老师任教的网站设计课程是我完成电影推荐系统的基石。总之，我要感谢大学四年来所有老师孜孜不倦的谆谆教诲与夜以继日的无私奉献。

接着，我还要感谢我的同学，是他们思维的火花激起了论文写作中的灵感。与他们的交流使我受益匪浅。

最后，我还要感谢我的父母，他们一直以来对我生活上与学习上的鼓励与引导是我得益完成论文的关键要素。

感谢所有在论文撰写工作中为我提供过帮助的人。