学校代码： 10385　　分类号：
研究生学号：1611414008　　密　级：

华侨大学
HUAQIAO UNIVERSITY

硕士专业学位论文

基于循环神经网络的推荐算法研究

Research on recommendation algorithm based on recurrent neural network

作者姓名：李玲
指导教师：王靖（教授）
实际单位导师：
专业学位类别：工程硕士
专业学位领域：计算机技术
研究方向：图像处理与模式识别
所在学院：计算机科学与技术

论文提交日期：　二零一九年五月二十二日

中国知网　https://www.cnki.net

中国知网 https://www.cnki.net

摘　要

随着云计算、大数据等技术的高速发展，互联网中越来越多的各类应用使得数据规模呈爆炸式增长。推荐系统作为解决信息过载问题的有效方法，在很多领域得到了广泛应用。但传统的推荐算法主要利用浅层模型或者人工提取特征的方式学习特征，导致算法无法获得用户和项目更深层次的特征表达。而且传统的推荐算法大多认为用户的属性是固定的，忽略了在用户行为的序列数据中用户兴趣爱好的动态变化对推荐结果造成的影响。

近年来，深度学习在计算机视觉、语音识别等领域取得了巨大成就，同时也给推荐系统带来了新的机遇。深度学习可以通过其深层次的非线性网络结构从海量用户行为数据中学习到用户和项目更本质的特征表达。其中循环神经网络可以通过对用户行为的序列数据进行建模，从而学习到用户的动态兴趣变化，提高下一时刻的推荐精度。

因此，本文主要围绕循环神经网络来进行推荐算法的研究。首先考虑到目前基于循环神经网络的推荐算法只关注用户行为序列的先后信息，忽略了用户对项目的评分偏好问题，提出了基于评分偏好的循环神经网络推荐算法。此外将循环神经网络和深度语义匹配模型相结合提出了基于深度语义匹配模型的循环神经网络推荐算法。具体研究内容如下：

1. 提出了基于评分偏好的循环神经网络推荐算法。算法通过循环神经网络对用户行为序列数据进行建模，然后将用户对项目的评分作为对序列中相应项目的偏好，进而将其结合到对下一时刻项目的预测中。通过在真实数据集上进行测试，结果表明该方法给用户推荐的下一时刻的项目不仅是用户将要点击的项目，同时也是用户感兴趣的项目。

2. 提出了基于深度语义匹配模型的循环神经网络推荐算法。算法将循环神经网络和深度语义匹配模型相结合，一方面通过循环神经网络对用户行为数据进行建模学习到用户当前兴趣的隐表示，另一方面通过深度语义匹配模型减少了模型参数，加快了模型的收敛。同时为了让算法学到更有效的用户特征表示，最后提出了带用户特征的基于深度语义匹配模型的循环神经网络推荐算法。通过多组实验表明该算法在收敛较快的同时提高了推荐的命中率。

中国知网　https://www.cnki.net

关键词：推荐系统　循环神经网络　评分偏好　深度语义匹配模型

中国知网　https://www.cnki.net

Abstract

With the rapid development of technologies such as cloud computing and big data, more and more applications in the Internet have caused the data scale to explode. As an effective method to solve the problem of information overload, the recommendation system has been widely used in many fields. However, the traditional recommendation algorithm mainly uses shallow models or artificially extracts features to learn features, resulting in the algorithm not being able to obtain deeper features of users and projects. Moreover, the traditional recommendation algorithm mostly considers that the user's attribute is fixed, ignoring the influence of the dynamic change of the user's interests in the sequence data of the user behavior on the recommendation result.

In recent years, deep learning has made great achievements in the fields of computer vision and speech recognition, and it has also brought new opportunities to the recommendation system. Deep learning can learn more essential features of users and items from massive user behavior data through its deep nonlinear network structure. The recurrent neural network can model the user's dynamic changes of interest by modeling the sequence data of the user behavior, and improve the recommendation accuracy of the next moment.

Therefore, this paper focuses on the research of recommendation algorithms around the recurrent neural network. Firstly, it is considered that the current recommendation algorithm based on the recurrent neural network only pays attention to the sequence information of the user behavior sequence, ignoring the user's preference for the item, and proposes a recurrent neural network recommendation algorithm based on the rating preference. In addition, a recurrent neural network recommendation algorithm based on deep semantic structured model is proposed by combining the recurrent neural network and the deep semantic structured model. The specific research contents are as

中国知网 https://www.cnki.net

follows:

1. A recurrent neural network recommendation algorithm based on rating preference is proposed. The algorithm models the user behavior sequence data through the recurrent neural network, and then takes the user's rating of the item as a preference for the corresponding item in the sequence, and then combines it into the prediction of the next moment item. By testing on the real data set, the results show that the method recommended to the user at the next moment is not only the item that the user will click, but also the item that the user is interested in.

2. A recurrent neural network recommendation algorithm based on deep semantic structured model is proposed. The algorithm combines the recurrent neural network and the deep semantic structured model. On the one hand, the user behavior data is modeled by the recurrent neural network to learn the implicit representation of the user's current interest. On the other hand, the model parameters are reduced by the deep semantic structured model, which speeds up. The convergence of the model. At the same time, in order to let the algorithm learn more effective user feature representation, a recurrent neural network recommendation algorithm based on deep semantic structured model with user features is proposed. Through multiple sets of experiments, the algorithm improves the recommended hit rate while converging faster.

Keywords: Recommendation system Recurrent neural network Rating preference Deep semantic structured model

中国知网 https://www.cnki.net

目 录

中国知网 https://www.cnki.net

中国知网 https://www.cnki.net

第 1 章　绪论

1.1　研究背景和意义

随着互联网的迅猛发展，人们的日常生活与互联网的联系变得越来越密切。丰富的互联网资源可以保证他们能够找到自己需要的内容，但由于信息的爆炸式增长，用户很难快速找到需要的内容，这也就是所谓的“信息过载”。以谷歌和百度为代表的搜索引擎在一定程度上缓解了“信息过载”的问题。搜索引擎主要是利用用户提供的关键字来对信息进行匹配，进而反馈给用户。但当用户不知道自己的需求或者需求不明确时，用户则很难从海量的信息中获取对自己有帮助的内容。为了帮助用户从海量的信息中选择其可能感兴趣的信息从而获得更好的用户体验，推荐系统应运而生[1]。

推荐系统是一个利用用户的偏好、项目特征、用户和项目之间的历史交互、人口统计信息等其它额外的信息，并通过一系列算法来挖掘用户潜在特征，进而为用户提供一些选择或者给用户推荐他们可能喜欢或者感兴趣的项目[2]。个性化推荐系统被广泛的应用在互联网的方方面面包括电子商务网站、电影和视频、音乐以及社交网络等。豆瓣电台根据用户的正负反馈推荐他们感兴趣的歌曲；今日头条根据用户的历史浏览记录来推荐新闻；亚马逊给用户推荐和他们之前喜欢项目类似的项目，此外亚马逊还按照用户在 Facebook 的好友关系，给用户推荐他们好友喜欢的项目；亚马逊前首席科学技术 Andreas Weigend 表示亚马逊有 20%-30%的销售来自于推荐系统。Netflix 的电影观看记录中有 80%来自于电影推荐[3]。YouTube 上主页的推荐算法贡献了 60%的视频点击率[4]。因此一个好的推荐系统不仅可以辅助用户做出决策提高用户的满意度，而且还可以使网站从中获益。

传统的推荐方法主要可以分为基于内容的推荐算法和协同过滤算法。基于内容的推荐系统是通过用户和项目的辅助信息文字或图片等来构建特征，并将和用户过去感兴趣的项目之间特征相匹配的项目推荐给用户[5]。基于内容的推荐方法有很多优点，第一，用户独立性。在基于内容的推荐中我们只需要根据当前用户的感兴趣的项目特征就可以完成推荐，而协同

中国知网　https://www.cnki.net

过滤的方法还需要寻找近邻用户。第二，可解释性强。我们只需要列出当前项目出现的原因是用户之前喜欢过类似的项目，就可以解释推荐系统的工作原理。第三，当新物品出现时，即使没有用户对其评分，也可以根据新物品的特征进行推荐。但基于内容的推荐系统的性能严重依赖于对内容特征的提取质量。协同过滤是指基于用户对项目的浏览、点击、评分或购买等历史行为对用户进行个性化推荐，它主要是寻找拥有相同喜好的用户，并推荐他们喜欢的项目[6]。这种方法只需要使用用户和项目间的显示或隐式的交互数据，而不需要了解用户或者商品的大量辅助信息，因此该算法成为最常用的推荐方法[7-9]。但由于用户往往只跟少量的项目有交互[17-20]，导致协同过滤算法常常遭受数据稀疏性问题。另外当新用户加入时由于系统没有其历史行为记录，所以会导致冷启动问题[21]。

近年来深度学习取得了飞跃性的进步和发展，并成为机器学习中的热门领域。如今深度学习在计算机视觉[64]、语音识别[65]等领域已经取得了巨大成就，如通过语音识别把语音转换成文字，对一张图片上出现的事物进行标注等。因为深度学习在处理复杂任务上能得到最好的效果，学术界和工业界都争相研究把深度学习的方法应用到更加广泛的领域中。最近深度学习也正在影响着推荐系统的变革，首先深度学习可以克服传统模型的缺点从而获得非线性的更高阶复杂的用户和项目特征表示方式，其次它还可以把多领域异构数据映射到同一隐空间中并获得数据间的统一表示[10]。从2016 年起，在推荐系统的顶级国际会议 RecSys 中开始定期组织基于深度学习的推荐系统会议。可见设计一个科学有效的算法，把深度学习和推荐系统结合起来是一个十分有意义和价值的课题。

1.2 研究现状

从各种文献[11-14]中可以得到，协同过滤算法主要包括基于近邻的协同过滤方法和基于模型的协同过滤方法。以基于用户的推荐为例，基于近邻的方法主要利用用户对项目的评分来寻找和当前用户具有相似评价习惯的其他用户即近邻用户，然后把近邻用户喜欢的并且当前用户没有评价过的项目推荐给当前用户。基于近邻的推荐算法通常只有一个近邻个数需要调节，因此该方法比较简单而且容易实现，并且可解释性强[15-16]。在大型

中国知网 https://www.cnki.net

商业应用中，近邻方法可以在离线阶段预先计算好，在线上时这种方法可以提供实时的高效率的推荐结果。但基于近邻的推荐方法有两个重要的缺陷，第一，近邻计算方法的假设具有局限。当用户间有相似爱好但他们之间很少有共同评分的项目时，基于近邻的方法不能准确的描述他们之间的相似性。第二，由于数据的稀疏性，用户之间很难会对相同的项目进行打分。这就造成使用有限的近邻进行推荐，导致推荐偏差。

基于模型的方法主要是通过已有的用户对项目的评价来学习用户和项目潜在特征之间的关联，然后为用户推荐新的项目[6]。基于模型的协同过滤方法有很多，其中包括基于聚类的协同过滤方法[22-26]、基于支持向量机的推荐方法[27]、基于贝叶斯网络的推荐方法[28-30]和矩阵分解方法[31-35]等。基于矩阵分解的方法是基于模型的协同过滤方法中最流行的方法[17,35-38]。基于矩阵分解的方法主要是将评分矩阵映射到低维隐空间中，通过低维表示重构评分矩阵。早期的研究[39-41]如奇异值分解模型(Singular Value Decomposition, SVD)[41]主要是对评分缺失值的填充，但填充的方法不仅增大了数据量而且会导致因填充数据不准确造成的推荐精度不高的问题。为解决 SVD 面临的问题，Simon Funk 在 2006 年提出了 FunkSVD[42]，它将评分矩阵分解为两个矩阵的乘积，通过最小化均方误差来优化模型。为了避免过拟合问题，同时提出了带 L2 正则项的优化函数，最后通过随机梯度下降来寻求最优解。在此之后出现了很多对该方法的改进，其中改进最成功的是 BiasSVD[43]，它额外考虑到系统中存在固有属性的影响。后来 Koren 提出了 SVD++[44]，主要在 BiasSVD 的基础上增加了对用户隐式反馈信息的建模。

传统的基于矩阵分解的协同过滤往往都是通过线性回归的方法去拟合评分矩阵，通常面临可扩展性不足的问题[46]。而基于深度学习的推荐系统主要有两点优势：第一，深度学习可以从海量的用户和项目数据中学习到更深层次非线性的用户和项目特征表示。第二，通过深度学习的方法可以把多源异构数据映射到同一隐空间中，从而获得对它们的统一表征[45]。在 2007 年 Salakhutdinov 等人[46]提出了基于受限玻尔兹曼机(Restricted Boltzmann Machine, RBM)的协同过滤算法，首次将深度学习和推荐系统结合起来。他们在传统的 RBM 模型中做了两点改进，第一，是在输入的时候使用评分的独热表示(one-hot encoding)，第二，由于评分矩阵的稀疏性，

中国知网 https://www.cnki.net

不考虑未评分的项目对模型的影响。后来也有许多基于受限玻尔兹曼机模型的改进[47-49]，但该模型由于可见层和隐藏层之间参数规模过大，而且RBM 的训练过程需要依靠变分推理和蒙特卡罗采样等方法使得模型训练时间过长。Sedhain 等人[50]提出基于自编码的协同过滤方法(AutoRec)，该方法通过一个编码器和一个解码器来对输入的评分进行重构，通过最小化均方误差来优化模型参数。Strub 等人[51]提出基于栈式降噪自编码模型(Stacked Denoising Auto Encoder, SDAE)的协同过滤，它在自编码模型的基础上对输入的评分数据添加了噪声，提高了模型的鲁棒性。另外为了获取更加丰富的用户特征表示，Elkahky 等人[52]将深度语义匹配(Deep Semantic Structured Models, DSSM)应用到推荐系统中，提出了结合多视角的深度神经网络模型(Multi-View DSSM，MV-DSSM)[53]。该方法把用户和项目两个实体映射到同一隐空间中，然后在该隐空间中通过余弦相似度来衡量两个实体间的关系，最后通过相似度进行推荐。

上述的推荐方法都忽略了用户行为序列对推荐的结果造成的影响。最早 Sarwar 等人[54]采用项目间的相似性来预测下一个项目，这种方法只考虑了和最后一次点击项目的相似性而忽略了点击的序列信息。Shani 等人[55]使用马尔可夫决策过程(Markov Decision Process, MDP)来描述用户点击的项目序列信息，通过状态转移概率来计算下一个点击项目的概率。尽管基于马尔科夫链模型的效果比基于近邻模型的效果好，但由于可转移状态的数量巨大，会造成可转移的状态随着问题维度指数增加，从而导致模型的计算复杂度过高。最近几年，受浅层神经网络在自然语言领域广泛应用的影响[56-57]，Oren 等人[58]提出 item2vec 的方法，利用用户点击的项目序列信息，将不同的项目映射到同一隐空间中，通过计算项目间的相似性来实现推荐。Wang 等人[59]将这种方法应用在预测下一次购物篮推荐问题中。虽然基于 item2vec 的方法简单高效而且可以对用户行为序列进行建模，但这种方法只考虑了序列的局部信息，并没有完全抓住整个序列的上下文影响。在利用深度学习的方法解决序列的推荐问题中，Hidasi 等人[60]首次提出把循环神经网络(Recurrent Neural Network, RNN)应用在基于会话的推荐中。他把用户之前点击的项目序列作为输入，来预测用户下一时刻将要点击的项目。Tan 等人[61]通过数据增强和模型预训练对 Hidasi 等人[60]提出的方法进行了改进。Devooght 等人[62]提出可以把协同过滤问题看作是

中国知网 https://www.cnki.net

序列预测问题，并测试了长短时记忆网络(Long Short Term Memory, LSTM)对于预测用户近期喜好的有效性。

需要注意的是，虽然基于深度学习的推荐系统已经得到越来越多的关注，但是从上面的研究现状中我们可以看出，目前深度学习在推荐系统中的应用仍处于初期发展阶段[63]。如何将深度学习和现有的推荐方法结合从而获得更加深层次的用户和项目的隐层特征表示以及如何提升基于深度学习的推荐系统的可解释性等问题都需要学者们进行更多更广泛的研究与尝试。

1.3 本文主要工作

本文的主要工作是围绕如何利用循环神经网络更好的捕获用户动态的兴趣爱好，以解决用户的兴趣爱好随时间变化而动态变化的问题。同时对模型的收敛速度、可解释性等问题提出了合理的改进。基于上述目标，本文的主要工作内容如下：

（1）对基于协同过滤的推荐算法和基于深度学习的推荐算法进行了梳理。在基于协同过滤的推荐算法中，重点总结了基于近邻的推荐算法和基于矩阵分解的推荐算法。在基于深度学习的推荐算法中，首先概括了深度学习在推荐系统中应用的两种主要框架，然后重点介绍了在深度学习中基于序列的推荐算法。

（2）在基于循环神经网络的推荐算法中，针对该模型只考虑了用户行为序列的影响，忽略了用户对不同项目之间评分偏好的影响，提出了基于评分偏好的循环神经网络推荐算法。这种算法给用户推荐的项目不仅是用户将要点击的项目，同时也是用户感兴趣的项目。

（3）在基于深度语义匹配模型中，针对无法学习到用户动态的兴趣变化这一问题，提出了基于深度语义匹配模型的循环神经网络推荐算法。该模型一方面可以动态的捕获用户当前的兴趣爱好，从而对用户当前的需求进行更精准的推荐。另一方面结合深度语义模型，可以减少模型的参数量，使得模型更容易收敛。

中国知网 https://www.cnki.net

1.4 本文组织结构

本文一共分为五个章节，论文结构及各章节主要内容如下：

第一章绪论。首先介绍了本论文的研究背景和意义，然后分析了协同过滤、基于深度学习的推荐算法和基于序列式的推荐算法的研究现状以及目前主要面临的问题。最后对全文的主要工作内容和论文组织结构进行了阐述。

第二章相关工作。首先对传统的推荐系统模型方法进行了梳理，主要描述了基于矩阵分解模型的改进过程。然后阐述了深度学习的主要模型以及两种基于深度学习的推荐系统框架。最后着重介绍了在深度学习中基于序列式的推荐算法，并分析了各自的优劣。

第三章基于评分偏好的循环神经网络推荐算法。由于直接将用户点击过的项目序列作为循环神经网络的输入，虽然模型可以捕获到用户兴趣的动态变化，但却忽略了用户对于不同的项目存在不同的偏好。因此将评分偏好引入到该模型中，使得该算法给用户推荐的商品不仅仅是用户下一时刻将要点击或观看的而且还是用户感兴趣的项目。

第四章基于深度语义匹配模型的循环神经网络推荐算法。将深度语义匹配模型引入到基于循环神经网络的推荐算法中，通过深度语义匹配，我们可以获得在同一隐空间中可解释性更强的用户和项目的隐表示。同时结合用户的特征，给用户提供更精准的推荐。最后通过实验验证了在不同的推荐场景下该方法的有效性。

第五章总结和展望。对本文的主要工作进行总结，分析尚且存在的一些问题，并对基于深度学习的序列式推荐系统的下一步研究工作给出一些想法和建议。

中国知网 https://www.cnki.net

第 2 章 相关工作

在生活中人们发现他们总是喜欢依赖别人的建议来做决定。比如,要看一部电影时，经常依赖于朋友的影评。随着电子商务的发展，互联网上资源呈现爆炸式增长，各类应用服务的大量出现使得人们无所适从。近年来推荐系统被认为是解决信息过载问题最有效的工具[1]。推荐系统主要利用用户过去的偏好，购买历史，浏览记录，人口统计信息等，通过一系列算法来挖掘用户潜在特征，进而为用户提供一些选择或者给用户推荐他们可能喜欢的物品。当用户每次发出请求时，推荐系统会利用数据库中存储的用户历史行为信息，捕获用户的喜好，进而给用户反馈推荐列表。而用户对于推荐的项目可以产生显式反馈(评分、购买等)或隐式反馈(点击、浏览等)，这些用户行为都会继续存储在推荐系统数据库中，并为下次推荐做准备。

以矩阵分解为主的推荐算法认为用户兴趣爱好是固定不变的，但在实际情况中，用户的兴趣随着时间的变化而变化。近年来，通过深度学习的方法对用户点击的项目序列进行建模来捕获用户兴趣的动态变化，越来越成为基于深度学习的推荐系统研究趋势之一[63]。本章的主要内容就是对协同过滤、深度学习和基于序列式的推荐算法进行简单的梳理。

2.1 推荐系统

在推荐系统中，我们用ε表示所有用户的行为集合。对于用户的某个行为，我们用元组 e 表示。以在 Netflix 大奖赛中给的数据为例，其中$\mathrm{e}=(\mathrm{i},\mathrm{j},r_{ij})$表示用户 i 给项目 j 的评分为$\mathrm{r_{ij}}$。通常我们还会有上下文信息如时间、用户使用的设备等，这样就可以表示为$\mathrm{e}=(\mathrm{i},\mathrm{j},r_{ij},\mathrm{t},\mathrm{d})$即用户 i 在时刻 t 使用设备 d 对项目 j 的评分为r_{ij}。其中假定一共有n个用户，m个项目，即$\mathrm{i}\in\{1,2,\cdots,\mathrm{n}\}$, $\mathrm{j}\in\{1,2,\cdots,\mathrm{m}\}$。我们可以把推荐问题定义成评分预测问题，以 Netflix 大赛数据集为例，对于用户某个行为元组$\mathrm{e}=(\mathrm{i},\mathrm{j},r_{ij})$,给定$(\mathrm{i},\mathrm{j})$来预测评分$r_{ij}$，即我们可以把元组 e 分成特征$\mathrm{x}=(\mathrm{i},\mathrm{j})$和标签$\mathrm{y}=r_{ij}$。当推荐系统模型考虑用户行为时间时，我们可以把推荐问题定义为下一时刻项目

中国知网 https://www.cnki.net

预测问题，给定用户 i 和当前时刻 t 来预测在 t 时刻要点击的项目 j，即 $x=(i,t)$，$y=j$。下表列出了本文要用到的符号表示。

表2.1　符号表示

符号	意义及说明
m	推荐系统中项目个数
n	推荐系统中用户个数
U_i，V_j	用户 i 和项目 j 可训练的低维嵌入表示，$U_i \in R^d$ $V_j \in R^d$
r_{ij}	用户 i 给项目 j 的评分
e	对于一个用户行为的元组表示，包含 k 个值
e_ℓ	表示元组 e 中第ℓ项
ε	所有用户行为的集合
χ_i	用户 i 的所有行为集合
$\chi_{i,t}$	在 t 时刻之前用户 i 的所有行为的集合
e^τ	发生在时刻τ的行为元组表示
R	用户和项目的评分矩阵

2.2　推荐系统的传统方法

协同过滤(Collaborative filtering，CF)方法是根据用户的兴趣爱好找到与其具有相同品味的用户，然后将相似用户喜欢的项目推荐给该用户。这种方法是简单、高效、可解释性强，因此这种方法被认为是推荐系统中应用最广的方法。协同过滤又可以分为基于邻域的推荐算法和基于模型的推荐算法。而基于模型的推荐算法中应用最广泛的是基于矩阵分解的推荐算法。由于协同过滤算法中基本使用用户对项目的评分矩阵来表示用户和项目之间的相互关系。

中国知网　https://www.cnki.net

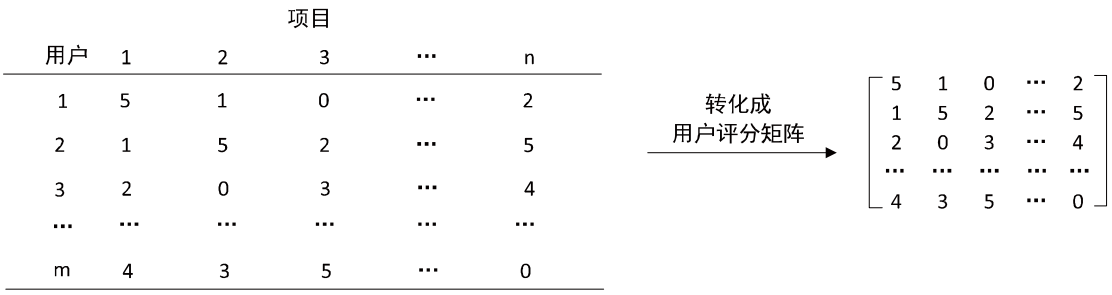

图2.1　用户项目评分矩阵转化图

其中评分矩阵 R 大小为$m \times n$，表示m个用户对n个项目的评分。r_{ij}表示用户 i 对项目 j 的评分，0 表示用户对该项目没有进行评分。

2.2.1　基于近邻的推荐算法

基于近邻的推荐算法能够发现数据间的一些隐藏关系，比如使用这种方法的电影推荐系统会帮助用户发现一些新的题材或者新的导演的电影。以基于近邻用户的协同过滤方法为例，该方法利用和用户 i 兴趣爱好相近且对项目 j 评分的近邻用户来预测用户 i 对项目 j 的评分 r_{ij}。用户 i 给项目 j 进行评分预测 $\hat{r}_{ij}$ 的流程如下：

（1）对评分进行标准化。由于不同用户存在不同的评价准则，比如有些用户可能更倾向于给所有项目较低的评分，为消除因为用户自己评分习惯问题造成预测结果的偏差，需要先对所有用户评分进行均值中心化。

$$\overline{r_i} = \frac{1}{|I_i|}\sum_{k \in I_i} r_{ik} \tag{2.1}$$

其中I_i表示用户 i 评分的项目集合，$|I_i|$表示该集合的数量。将该用户原始评分减去平均分$\overline{r_i}$就是均值中心化后的评分。

（2）计算两两用户之间的相似度w_{ij}。通常为了防止不同用户评分均值及方差间差异的影响，我们使用皮尔逊相似度(Pearson Correlation,PC)来表示w_{ij}。

$$w_{ij} = PC(i,j) = \frac{\sum_{k \in I_{ij}}(r_{ik}-\overline{r_i})(r_{jk}-\overline{r_j})}{\sqrt{\sum_{k \in I_i}(r_{ik}-\overline{r_i})^2 \sum_{k \in I_j}(r_{jk}-\overline{r_j})^2}} \tag{2.2}$$

其中，I_{ij}表示用户 i 和用户 j 共同评分的项目集合。

中国知网　https://www.cnki.net

（3）选择近邻并预测评分。使用和用户 i 具有兴趣相似度w_{ij}最高的近邻且对项目 j 已经评分的近邻集合N(i)来计算最后的评分预测结果。

$$\widehat{r_{ij}} = \overline{r_i} + \frac{\sum_{k\in N(i)} w_{ik}(r_{kj}-\overline{r_k})}{\sum_{k\in N(i)} |w_{ik}|} \quad (2.3)$$

基于近邻的方法有许多优点。该方法只有近邻数目需要调整，因此它很简单直接而且容易实现。其次基于近邻的方法实时性高，对存储的要求较低并且可解释性好。但这种方法也存在局限性。只有当用户对相同的项目进行评分时才能计算相似度，尤其是当数据稀疏时，用户的近邻数量很少，就会造成在预测评分时只使用了很少量的近邻，使得推荐结果造成偏差。

2.2.2 基于矩阵分解的推荐算法

为解决基于近邻推荐的方法对稀疏数据敏感的问题，我们可以通过将原来用户和物品的评分空间降维到隐变量空间来获取它们之间相关性。即使用户评分的物品都不相同，这种降维的方法也可以发现它们之间的关联。在这种方法中，通常的方法是把用户和物品的评分矩阵 R 近似表示为用户特征矩阵 P 和项目特征矩阵 Q 内积的形式：

$$R \approx PQ^T \quad (2.4)$$

其中$P \in R^{m\times k}, Q \in R^{n\times k}$，k 为降维到特征空间的维度。矩阵 P 的第 i 行($p_i \in R^k$)表示用户u_i映射到特征空间的坐标。我们可以通过最小化平方 F 范数下的评分重建误差来学习 P,Q:

$$L(P,Q) = \|R - PQ^T\|_F^2 = \sum_{i,j}\left(r_{ij} - p_i q_j^T\right)^2 \quad (2.5)$$

可以用对评分矩阵 R 进行奇异值分解的方法来达到最小化上式误差的目的。但使用奇异值分解的方法存在一个严重的问题，就是奇异值分解需要在稠密的矩阵中进行，而真实的评分矩阵 R 绝大部分r_{ij}是没有意义的，即用户 i 没有对项目 j 进行打分。因此 Simon Funk 提出解决这类问题同一的方法 FunkSVD[42]，该方法是在已知评分下，去学习 P、Q:

$$\underset{P,Q}{arg\,min}\, L(P,Q) = \sum_{r_{ij}\in R}\left(r_{ij} - p_i q_j^T\right)^2 + \lambda(\|p_i\|^2 + \left\|q_j\right\|^2) \quad (2.6)$$

中国知网 https://www.cnki.net

其中λ是正则化参数，用来衡量模型复杂度和准确度之间关系，通过对参数的复杂性加大惩罚来避免模型的过拟合。后来 Koren 等人[43]在上式的基础上又考虑了在预测评分r_{ij}时，添加了用户的偏置项b_i，物品的偏置项b_j及与系统偏置项μ对评分预测造成的影响，提出 BiasSVD 算法。BiasSVD 的优化函数如下：

$$\underset{P,Q}{arg\,min}\, L(P,Q) = \sum_{r_{ij}\in R}\left(r_{ij} - \mu - b_i - b_j - p_i q_j^T\right)^2 + \lambda\left(\|p_i\|^2 + \left\|q_j\right\|^2 + \|b_i\|^2 + \left\|b_j\right\|^2\right) \quad (2.7)$$

基于矩阵分解的推荐算法具有更好的可扩展性，同时通过用户和项目的低维特征表示重构评分矩阵，这种方法有效的压缩了模型，降低了计算的复杂度同时使得模型训练起来相对容易。但是这种基于矩阵分解的推荐算法采用的是浅层模型，通过线性分解去拟合评分矩阵，无法学习到用户和项目更加深层次的特征。

2.3 推荐系统的深度学习方法

2.3.1 深度学习技术

随着训练数据量的增加，更好的分布式计算设施的出现以及与日俱增的对精度的要求都使得深度学习被越来越广泛的应用到解决实际问题中[66]。深度学习崛起的序幕是在 ImageNet 图像分类比赛中，将 top5 的错误率从 26.1%降到了 15.3%[67]。在语音识别领域[65,68]，深度学习的方法使得有些错误率甚至降低了一半。在交通标志分类上，深度学习甚至取得了超越人类的表现[69]。由于本论文主要是对基于循环神经网络推荐算法进行改进，因此这里主要介绍循环神经网络的模型及其优化方法。

循环神经网络是一类用于处理序列数据的网络模型。不同于在卷积神经网络(Convolutional Neural Networks, CNN)和深度神经网络(Deep Neural Networks, DNN)中，样本的输入的维度固定。当样本输入序列的长短不同时，比如一段连续的视频帧、一句话等，我们很难直接把它们拆分成长度一致的独立样本进行训练。循环神经网络通过对时间序列进行建模，并广泛的应用于机器翻译、视频行为识别及命名实体识别等领域中。循环神经

中国知网 https://www.cnki.net

网络的主要模型如下：

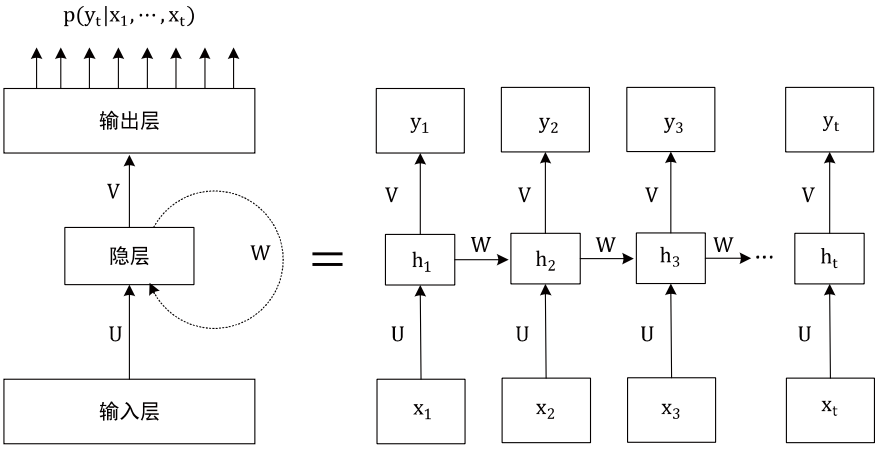

图2.2　循环神经网络模型

上图 2.2 中左边是循环神经网络整体的模型图，右边是按照时间序列展开的图。在每一个时间步中，x_t表示 RNN 在 t 时刻的输入，h_t表示 t 时刻隐藏状态，o_t表示 t 时刻未归一化的概率输出，预测的标签用$\hat{y}_t = \mathrm{softmax}(o_t)$表示，$y_t$为 t 时刻的真实标签。$h_t$由$x_t$和$h_{t-1}$共同决定。循环神经网络假定在时刻 t 的变量之后，时刻 t+1 的变量的条件概率分布是平稳的，也就是上一时刻和下一时刻之间的关系并不依赖于 t，因此 RNN 中使用了权值共享。上图中 U，V 和 W 即为 RNN 网络中共享的权重矩阵。

$$a_t = b + Wh_{t-1} + Ux_t \quad (2.8)$$

$$h_t = \phi(a_t) \quad (2.9)$$

$$o_t = c + Vh_t \quad (2.10)$$

$$\hat{y} = p(y_t|\{x_1, x_2, \dots, x_t\}) = \sigma(o_t) \quad (2.11)$$

其中 b 和 c 是偏置。$\phi(\cdot)$表示激活函数，在隐藏层时通常使用 tanh，在输出层激活函数通常使用 softmax 函数，用σ表示。对于每一时刻 t，我们通过对数似然函数L_t来表示$\hat{y}$和y之间的误差，所以最后目标是最大化所有时刻的对数似然函数之和，也就是最小化负对数似然，如下式：

$$L(\{x_1, x_2, \dots, x_\tau\}, \{y_1, y_2, \dots, y_\tau\}) = \sum_{t=1}^{\tau} L_t$$
$$= -\sum_{t=1}^{\tau} log\, p(y_t|\{x_1, x_2, \dots, x_t\}) \quad (2.12)$$

同 DNN 和 CNN 一样，在优化模型参数的时候使用梯度下降法来迭代

中国知网　https://www.cnki.net

更新模型参数。由于循环神经网络在前向传播时是基于时间序列进行的，所以在计算梯度时也需要穿越时间反向传播(Back Propagation Through Time, BPTT)。由于模型参数 U，V，W，b 和 c 在各个时刻都是共享的，所以我们每个时刻更新的参数都相同。每一次迭代要更新的梯度如下：

$$\frac{\partial L}{\partial c}=\sum_{t=1}^{\tau}\frac{\partial L_t}{\partial c}=\sum_{t=1}^{\tau}\frac{\partial L_t}{\partial o_t}\frac{\partial o_t}{\partial c} \tag{2.13}$$

$$\frac{\partial L}{\partial V}=\sum_{t=1}^{\tau}\frac{\partial L_t}{\partial V}=\sum_{t=1}^{\tau}\frac{\partial L_t}{\partial o_t}\frac{\partial o_t}{\partial V} \tag{2.14}$$

在计算参数 c 和 V 的梯度时比较简单，因为在前向传播时，o_t只于h_t有关。在计算参数 U，W 和 b 的梯度时就比较困难，不仅要考虑误差对当前时刻的影响，还要计算其对之前序列的影响，其梯度计算公式如下

$$\frac{\partial L}{\partial b}=\sum_{t=1}^{\tau}\frac{\partial L_t}{\partial b}=\sum_{t=1}^{\tau}\frac{\partial L_t}{\partial o_t}\frac{\partial o_t}{\partial h_t}\cdot\sum_{k=1}^{t}\left[\left(\prod_{i=k+1}^{t}\frac{\partial h_i}{\partial h_{i-1}}\right)\cdot\frac{\partial h_k}{\partial a_k}\right] \tag{2.15}$$

$$\frac{\partial L}{\partial W}=\sum_{t=1}^{\tau}\frac{\partial L_t}{\partial W}=\sum_{t=1}^{\tau}\frac{\partial L_t}{\partial o_t}\frac{\partial o_t}{\partial h_t}\cdot\sum_{k=1}^{t}\left[\left(\prod_{i=k+1}^{t}\frac{\partial h_i}{\partial h_{i-1}}\right)\frac{\partial h_k}{\partial W}\right] \tag{2.16}$$

$$\frac{\partial L}{\partial U}=\sum_{t=1}^{\tau}\frac{\partial L_t}{\partial U}=\sum_{t=1}^{\tau}\frac{\partial L_t}{\partial o_t}\frac{\partial o_t}{\partial h_t}\cdot\sum_{k=1}^{t}\left[\left(\prod_{i=k+1}^{t}\frac{\partial h_i}{\partial h_{i-1}}\right)\frac{\partial h_k}{\partial U}\right] \tag{2.17}$$

从上述公式可以看出在计算 U，W 和 b 的梯度时，当模型序列很长时，由于经过时间的反向传播，会存在许多$\frac{\partial h_i}{\partial h_{i-1}}$的累乘部分。而$\frac{\partial h_i}{\partial h_{i-1}}=\frac{\partial h_i}{\partial a_i}W$，$\frac{\partial h_i}{\partial a_i}\in(0,1]$，所以当 W 的取值过大或过小时，误差经过许多阶段传播后梯度倾向于消失或爆炸。为了缓解这一问题，后来人们提出了长短时记忆网络和门控循环单元(Gated Recurrent Unit, GRU)[70]。

2.3.2 基于深度学习的推荐系统架构

基于深度学习的推荐系统主要有两种系统架构[74]。如图 2.3 所示，一种是把用户和项目的相关信息分别作为深度学习模型的输入，然后通过深度学习得到用户和项目的低维隐表示，最后再由用户和项目的隐表示做内积或者余弦相似度得到用户和项目的得分。在显示反馈中，计算的得分就是去拟合用户对项目的评分。在隐式反馈中，计算的得分就表示用户和项目相互作用的概率，让该概率尽量接近于 1。这种架构的推荐算法有深度矩阵分解推荐算法[71]，基于自编码的推荐算法 AutoRec[50]，基于降噪自编

中国知网 https://www.cnki.net

码的推荐算法[51]等。另一种系统架构是把用户和项目的相关信息共同作为深度学习模型的输入，由深度学习模型直接学习到它们两者的融合特征，最后进行得分匹配。这种架构的推荐算法主要有：基于神经网络的协同过滤算法(Neural Collaborative Filtering, NCF)[72]，基于因子分解机模型的推荐算法(Factorization Machine, FM)[73]等。

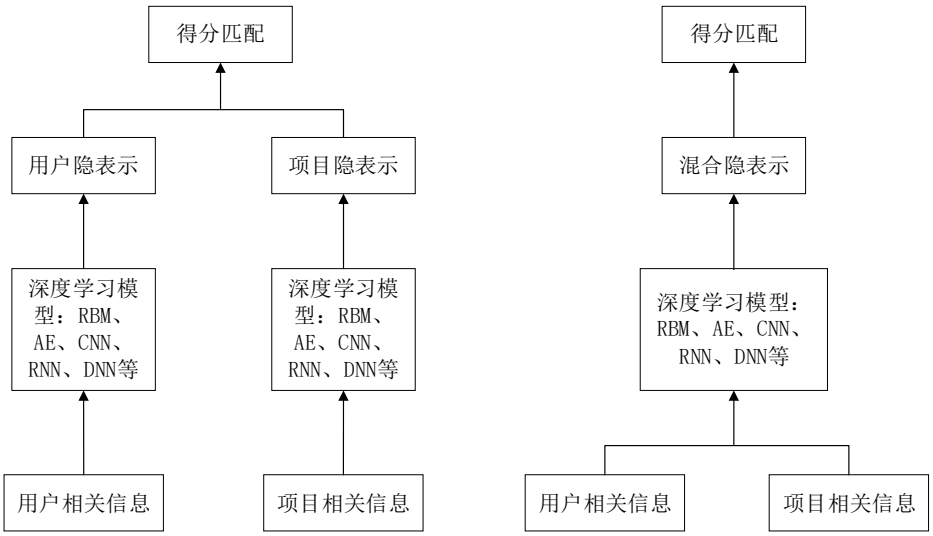

图2.3　推荐系统的两种系统架构

由于在很多推荐场景中，对用户点击的序列进行建模往往会对结果产生重要的影响。因此下面主要介绍下通过深度学习的方法对用户点击的项目序列进行建模的算法。

2.3.3　基于 item2vec 的推荐算法

item2vec[58]这一方法是 Oren 等人在 2016 年提出的。该方法主要参考了谷歌的 word2vec 方法，并把该方法推广到推荐系统中。其核心思想是把用户点击的项目序列看作一个集合，在这一集合中利用项目之间的共现性学习到项目的低维嵌入表示，最后通过最近邻等方法来产生推荐。遵循在表格 2.1 定义的符号表示，在用户 i 按照时间顺序点击的项目集合χ_i中，在τ时刻的用户行为元组表示为$e^{\tau} = (i, j)$，e_2^{τ}表示在时刻τ用户 i 点击的项目为 j。在 item2vec 中主要使用的是 skip-gram 模型，其模型架构图如下图 2.4 所示：

中国知网　https://www.cnki.net

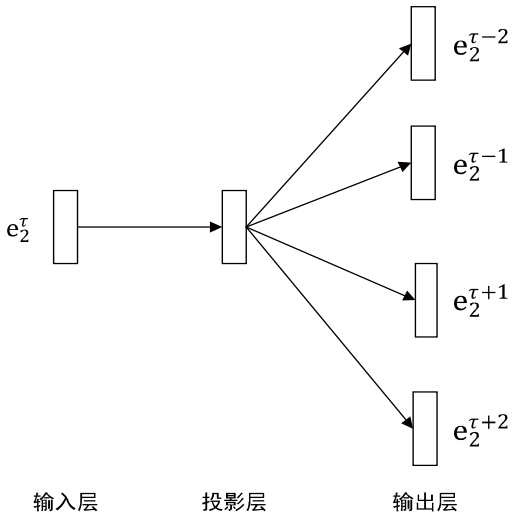

图2.4　skip-gram模型架构图

skip-gram 模型包括输入层、投影层和输出层，主要是根据当前的输入来预测其上下文。在 skip-gram 模型中，最大化其目标函数为：

$$L = \frac{1}{|\chi_i|}\sum_{\chi_i}\sum_{-c\le j\le c, j\neq 0} log\, p(e_2^{\tau+j}|e_2^{\tau}) \qquad (2.18)$$

其中 c 是上下文滑动窗口的大小，$p(e_2^{\tau+j}|e_2^{\tau})$是一个经 softmax 归一化后的概率，表达式：

$$p\left(e_2^{\tau+j}\middle|e_2^{\tau}\right) = \frac{\exp(v_{e_2^{\tau}} \cdot w_{e_2^{\tau+j}})}{\sum_{w_k\in w} exp(v_{e_2^{\tau}} \cdot w_k)} \qquad (2.19)$$

其中$v_{e_2^{\tau}} \in v(\subset R^d)$表示项目$e_2^{\tau}$的嵌入向量，$w_{e_2^{\tau}} \in w(\subset R^d)$表示项目$e_2^{\tau}$对应的权重向量。d 为嵌入向量的维度。上式中在计算$\sum_{w_k\in w}\exp(v_{e_2^{\tau}} \cdot w_k)$需要消耗很多的资源和时间，因此可以通过负采样把原来的问题缩小规模。

通过上述方法对用户点击的行为序列进行建模，它有一个明显的优点就是通过不断优化目标函数，我们可以得到项目对应的低维嵌入矩阵 V。采用项目的 one-hot 表示来区别不同的项目，用一个长度大小为项目数量的向量，只有该项目对应索引下标的位置为 1，其它位置为 0。这种方式很容易造成维度灾难并且不能很好的刻画词之间的相似性。而通过基于负采样的 skip-gram 模型产生的嵌入向量是有一定意义的，我们可以通过余

中国知网　https://www.cnki.net

弦相似性来衡量每个项目之间的相似性。但是这种方法虽然考虑了用户点击项目的序列信息，但是主要利用的是序列中局部项目之间的共现性，并不能完全有效的对用户序列行为进行建模，忽略了用户行为序列前后的之间的依赖关系。

2.3.4 基于会话的循环神经网络推荐算法

为了更加有效的利用用户的行为序列信息，Hidasi 等人[60]提出采用循环神经网络对用户的行为序列进行建模。该方法主要利用当前会话中用户点击过的历史项目序列信息，来计算下一时刻点击每个项目的概率，最后通过真实的下一时刻要点击的项目作为标签，来优化整个模型的参数。其模型结构如下所示：

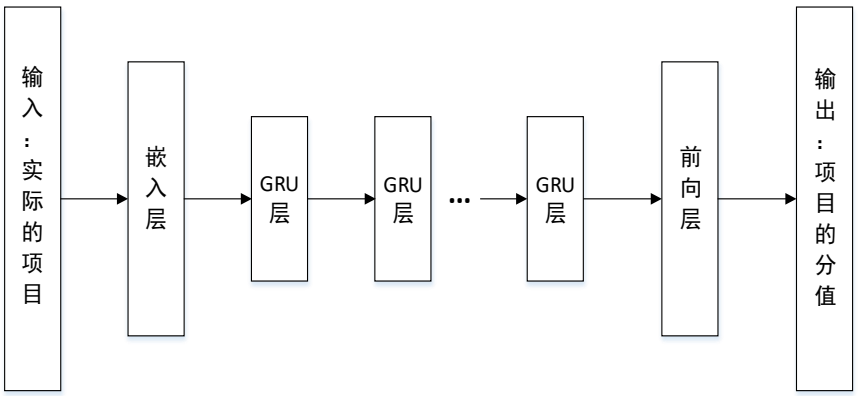

图2.5 Hidasi等人提出的模型结构图

在该模型中，使用用户行为序列的 one-hot 编码作为输入，然后通过嵌入层得到每个项目的低维表示，中间再经过多个堆叠的 GRU 层，最后通过前向层把最后的输出映射到 m 维，并通过 softmax 得到下一时刻点击每个项目的概率。当该系统中项目总个数 m 太多时，可以通过对项目的热门程度进行采样的方法减少最后 softmax 的计算量。为了加快模型训练，他们采取了两种优化策略进行优化。第一，是采用 mini-batch 来进行训练。考虑到用户行为会话有长有短差异很大，将不同的会话拼接起来，当遇到新的会话时，只需将 GRU 模型中参数重新初始化就可以了，这样提升了数据的利用率。第二，巧妙的把同一个 mini-batch 中其它会话的下一时刻要点击的项目作为负样本，这样做直接减少了负采样这一过程。这种方法

中国知网 https://www.cnki.net

虽然能够有效的对用户行为序列进行建模，但这种方法忽略了用户对不同项目存在不同偏好的影响。

2.4 本章小结

本章主要梳理了传统的推荐系统方法和基于深度学习的推荐算法。在传统的推荐系统方法中主要总结了协同过滤的原理，然后详细整理了基于近邻的协同过滤推荐方法和基于矩阵分解的推荐方法，并分析了它们的优缺点。在基于深度学习的推荐算法中，主要阐述了两种现阶段最流行的基于深度学习的推荐架构及它们的算法应用。考虑到用户的行为序列往往会对推荐造成重要的影响，介绍了两种对用户行为序列进行建模的方法。

中国知网 https://www.cnki.net

中国知网 https://www.cnki.net

第 3 章 基于评分偏好的循环神经网络推荐算法

在推荐系统中，时间因素对用户兴趣爱好的变化有着重要的影响。随着用户所处的上下文不同，用户的兴趣是动态变化的。比如随着季节的变化，工作的改变等因素，用户感兴趣的东西也会发生变化。在推荐系统中，应用最广泛的基于矩阵分解的推荐算法把用户兴趣爱好认为是固定不变的，忽略了用户行为序列的先后信息对推荐系统的影响。随着基于深度学习的推荐系统研究课题受到越来越多的关注，基于循环神经网络的推荐系统在捕获用户动态兴趣变化方面也已成为一个重要的研究方向。为了给用户提供更加精准的推荐服务，我们希望推荐的不仅仅是下一时刻用户将会点击的物品，而且还是用户感兴趣的物品。在这一章我们提出一种基于评分偏好的循环神经网络推荐算法来解决这一问题。

3.1 长短时记忆网络

在对序列信息进行建模中，最常用的就是循环神经网络。但是 RNN 模型存在梯度消失或梯度爆炸的问题。在实际应用中，最有效的解决循环神经网络梯度消失问题的模型是长短时记忆网络，其结构框图如下：

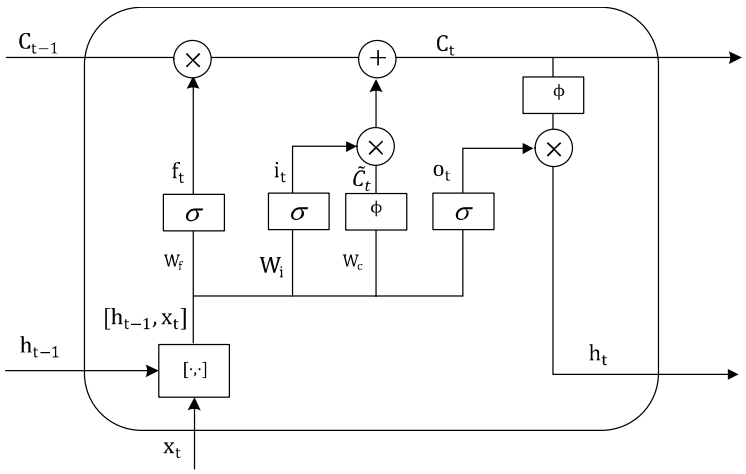

图3.1 LSTM内部结构框图

LSTM 模型的外部连接和 RNN 类似，但它每个内部结构采用了控制

中国知网 https://www.cnki.net

信息流动的门控单元，让神经网络可以动态的决定信息的遗忘程度。在LSTM中除了有和RNN一样的隐层状态h_t之外，还有一个隐藏状态我们称为细胞状态(Cell State)用C_t表示。除了隐藏状态，在LSTM中还有许多门控单元(Gate)，分别为遗忘门f_t，输入门i_t和输出门o_t。遗忘门(Forget Gate)是通过一定的概率来决定是否遗忘上一层的隐藏细胞状态。

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3.1}$$

其中W_f和b_f为输入的权重和偏置，$\sigma(\cdot)$为sigmoid激活函数。输入门(Input Gate)用来处理当前输入。我们通过输入门和新的细胞状态$\widetilde{C_t}$以及遗忘门来更新该时刻的最终细胞状态C_t。其公式为：

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.2}$$

$$\widetilde{C_t} = \phi(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3.3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t} \tag{3.4}$$

其中W_i，b_i，W_c和b_c都是输入的权重和偏置，ϕ为tanh激活函数。$\odot$为矩阵的哈达马积(Hadamard product)。通过新的细胞状态C_t和输出门，我们就可以更新隐藏状态h_t。用W_o和b_o表示输出门的权重和偏置。

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3.5}$$

$$h_t = o_t \odot \phi(C_t) \tag{3.6}$$

LSTM可以通过门控单元来解决梯度消失问题，但不能解决梯度爆炸问题。对于梯度爆炸问题，我们采用梯度截断。在LSTM的变体和替代的研究中，最突出的是GRU模型。它与LSTM主要区别就是单个门控单元同时控制遗忘因子并决定更新状态。在实际应用中，由于LSTM模型和GRU模型效果类似，所以在本文中主要使用LSTM模型。

3.2 基于长短时记忆网络的推荐算法

基于长短时记忆网络的协同过滤推荐算法[62]是把协同过滤问题看作是序列预测问题。其主要思想是利用循环神经网络对用户的行为序列进行建模，捕获用户的兴趣随点击行为的变化而变化，从而对用户下一时刻的

中国知网 https://www.cnki.net

将要点击的项目进行预测。用符号表示为给定用户 i 在 t 时刻之前点击行为序列集合$\chi_{i,t}$，来预测用户 i 在时刻 t 时将要点击的项目e_2^t，其中符号说明可详见表 2.1。其模型图如下所示：

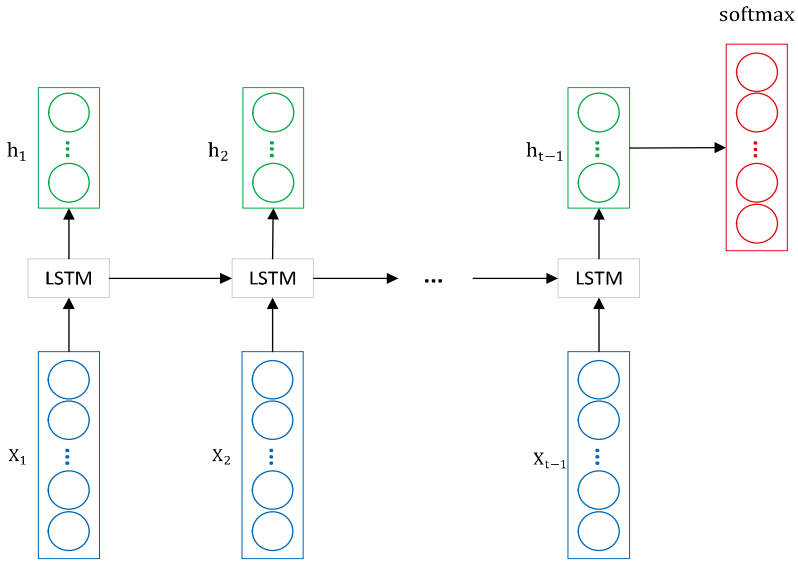

图3.2　基于循环神经网络的推荐算法模型图

在图 3.2 中，模型的输入为用户在每一个时间点上点击项目的 one-hot 表示。通过循环神经网络捕获到用户动态的兴趣变化，最后经过 softmax 层得到在下一时刻用户点击每一个项目的概率。用公式表示如下：

$$h_l = f_{LSTM}(h_{l-1}, x_l; \theta) \tag{3.7}$$

$$y = softmax(W_h h_{t-1} + b_h) \tag{3.8}$$

其中$l \in \{1,2,\cdots,t-1\}$，x_l为在l时刻用户点击项目e_2^l的 one-hot 表示，$f_{LSTM}(\cdot)$表示长短时记忆网络模型的算法。Θ为 LSTM 网络中涉及到的模型参数。$h_l \in R^d$,其中d为 LSTM 隐层神经元的个数。$W_h \in R^{m \times d}$是最后全连接层的权值矩阵，$b_h \in R^m$是偏置。

我们把该问题看作是项目分类问题，只有在时刻 t 用户点击项目e_2^t的概率为 1，其它项目被点击的概率为 0。最后我们使用多分类交叉熵损失函数作为最后的代价函数。用$y_{correct}$代表经过模型预测的在时刻 t 点击项目e_2^t的概率，所以我们可以把最后的损失函数写成：

$$L = -log(y_{correct}) \tag{3.9}$$

中国知网　https://www.cnki.net

最后我们通过随机梯度下降的方法最小化上述代价函数，从而更新网络所需要的参数θ，W_h和b_h。在给用户进行推荐时，只需将最后预测的概率最大的项目推荐给用户。

3.3 基于评分偏好的循环神经网络推荐算法

在基于序列的推荐中，我们希望模型不仅仅能预测到用户接下来看什么，而且预测的项目也是用户感兴趣的。在基于循环神经网络的推荐算法中，只考虑了输入项目之间的前后顺序问题，忽略了用户对于不同的项目有着不同的喜好这一问题。例如用户看了电影 A 评分为 5 分，然后又看了电影 B 并评分 2 分。在 LSTM 中由于它忽略了用户对不同电影之间的不同喜好，因此电影 A 和电影 B 除了顺序之外，对网络造成的影响是一样的。针对这一问题，我们引入了评分偏好来捕获用户对项目的喜好程度。

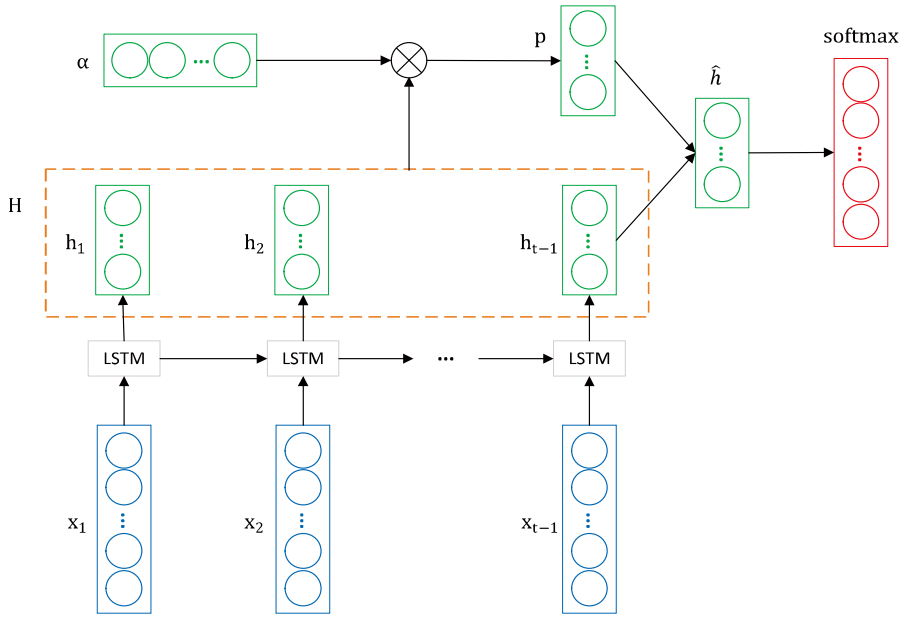

图3.3 基于评分偏好的循环神经网络推荐算法模型图

基于评分偏好的循环神经网络模型(Rating-based LSTM, R-LSTM)如上图 3.3 所示。我们让α_{ij}表示用户 i 给项目 j 的评分偏好值，α_{ij}定义为:

$$\alpha_{ij} = \frac{r_{ij}}{\sum_{k=1}^{t-1} r_{ie_2^k}} \quad (3.10)$$

中国知网 https://www.cnki.net

我们用$\alpha = [\alpha_{ie_2^1}, \alpha_{ie_2^2}, \cdots, \alpha_{ie_2^{t-1}}]$表示时刻 t 之前的评分偏好向量，其中$\alpha \in R^{1\times(t-1)}$。我们用$H = [h_1, h_2, \cdots, h_{t-1}]$表示 LSTM 模型的隐层，并且$H \in R^{d\times(t-1)}$。

$$p = H\alpha^T \tag{3.11}$$

上式$p \in R^d$表示经过评分偏好加权过后的隐层表达。受到 Rocktäschel 工作[75]的启发，最后结合经过评分偏好之后的隐层表达 p 和序列的隐层表达h_{t-1}之后的隐层表达为$\hat{h}$:

$$\hat{h} = tanh(W_p p + W_x h_{t-1}) \tag{3.12}$$

其中$\hat{h} \in R^d$，$W_p \in R^{d\times d}$，$W_x \in R^{d\times d}$分别是向量 p 和向量h_{t-1}所对应的权值矩阵。R-LSTM 模型的最后一步和基于 LSTM 模型的推荐方法一样，就是把隐层的输出经过全连接层映射到 m 维，最后经过 softmax 进行归一化，这样就得到了在下一时刻用户点击每一个项目的概率。最后采用交叉熵损失函数作为最后的代价函数，公式同 3.9 所示。

训练基于评分偏好的推荐算法，目的是预测用户下一时刻要看的并且是其喜欢看的项目。我们平常训练的样本中，用户 i 下一时刻要看的项目是e_2^t，但看后他对项目e_2^t的评分$r_{ie_2^t}$有时并不会很高。我们希望提出的模型不但可以给用户推荐下一时刻要看的而且是其喜欢看的项目。因此对于预测项目的评分不同我们给出了不同的训练方式。受到 TopicRNN[76]模型的启发，我们把高评分集合用 HS(high score)表示。如果要预测的项目的评分$r_{ie_2^t} \notin HS$，我们让$l_t = 1$，如果$r_{ie_2^t} \in HS$，$l_t = 0$。其中l_t表示在 t 时刻预测项目评分的指示器。那么之前的公式可以重新写为：

$$\hat{h} = tanh((1 - l_t)W_p p + W_x h_{t-1}) \tag{3.13}$$

上式可以看出l_t决定着模型中的评分偏好会不会对预测结果造成影响。只有要预测的下一时刻项目的评分属于高评分集合，我们才让这个样本去学习更新评分偏好的参数。假设我们的模型经过充分的训练之后，在测试过程中我们让$l_t = 0$。

中国知网 https://www.cnki.net

3.4 实验与分析

3.4.1 数据集介绍

在推荐系统中，由于用户只会对系统中少数的项目进行点击评价等操作，因此数据存在严重的稀疏性问题。为了衡量数据的稀疏程度，首先引入了稀疏度的概念。稀疏度的公式如下：

$$\text{稀疏度} = \frac{\sum_{i=1}^{n} |\chi_i|}{m \times n} \tag{3.14}$$

在该实验中主要用到了两个 MovieLens 数据集①。MovieLens 数据集[77]是由明尼苏达大学的 GroupLens 科研小组从 MovieLens 网站获取的电影评分的数据集。根据数据集的大小和收集的时间段的不同，这些数据集主要可以分为 100k 的数据集、1M 的数据集、10M 的数据集和 20M 的数据集。以大小为 1M 和 10M 为例，在 MovieLens 1M 中包含了 6040 个用户对 3883 个电影的一百万条评分数据，评分为从 1 到 5 的整数。其中每个用户至少给 20 个电影进行了打分，最多给 2314 部电影进行了打分。该数据集的稀疏度为 4.25%。在 MovieLens 10M 中包括了 71567 个用户对 10681 个电影的一千万条评分记录，评分是从 1 到 5 之间以 0.5 递增的小数。在 MovieLens 10M 数据集中评分的稀疏度为 1.31%。

3.4.2 实验指标

为了更好的衡量在用户行为序列预测中下一时刻项目预测的准确率以及在下一时刻高评分项目预测的准确率，我们提出两个评价指标，一个是下一时刻项目预测的准确率(Next Prediction Success，NPS)，另一个是下一时刻高评分项目预测的准确率(High Rating Prediction Success，HRPS)。这两个指标的结果都是越大表示推荐效果越好。我们定义给用户 u 推荐 k 个项目，表示为$\mathrm{I}_{\mathrm{k}}^{\mathrm{u}} = [i_1^u, i_2^u, \cdots, i_k^u]$，并且定义对于用户 u 来说取的 o 个项目为目标项目，则目标项目列表为$\mathrm{T}_{\mathrm{o}}^{\mathrm{u}} = [t_1^u, t_2^u, \cdots, t_o^u]$，N 是测试集中用户的总数量。下面给出了这两个评价指标的计算公式：

① MovieLens 数据集链接：https://grouplens.org/datasets/movielens/

中国知网 https://www.cnki.net

$$NPS(k,o)=\frac{\sum_{u=1}^{N}1\{|I_k^u\cap T_o^u|\neq 0\}}{N} \quad (3.15)$$

其中$1\{\cdot\}$是一个指示器，当满足条件时其值为 1，不满足条件时，其值为 0。在高评分项目预测的准确率中，我们把推荐列表I_k^u和用户目标列表T_o^u中重合的项目对应的评分求均值，如果均值评分属于高评分HS集合，则认为推荐正确。因此高评分项目预测的准确率计算公式如下：

$$HRPS(k,o)=\frac{\sum_{u=1}^{N}1\{|I_k^u\cap T_o^u|\neq 0\wedge\frac{\sum_{j\in I_k^u\cap T_o^u}r_j^u}{|I_k^u\cap T_o^u|}\in HS\}}{N} \quad (3.16)$$

3.4.3 实验对比方法及参数设置

为了验证 R-LSTM 算法的优越性，我们选择了五种经典的方法进行对比实验。分别是早期基于项目流行度的算法 POP、基于用户近邻的协同过滤算法 User-KNN、基于贝叶斯个性化排序的协同过滤算法 BPR-MF、基于马尔可夫链的推荐算法 MC 和基于循环神经网络的推荐算法 LSTM，下面对这些算法进行了简要介绍。

POP：基于流行度的推荐主要是在训练集中把项目按照热度排序，并把热度高的项目推荐给用户。这是最早期的推荐算法，但也是最简单有效的算法。

User-KNN：该算法主要分为三个步骤，首先计算目标用户和其它用户之间的相似度，然后找到与目标用户最近邻的用户，最后通过近邻用户产生对目标用户未评价项目的评分预测。

BPR-MF：该算法[78]是最常用的基于矩阵分解的推荐方法之一。它通过随机梯度下降的方法优化 PairWise 排序损失。

MC：该算法中把用户行为序列中不同的项目认为成马尔可夫链的每个状态，通过状态转移概率就可以得到用户在下一时刻点击每个项目的概率。其中状态转移概率是通过训练集中物品的转移频率计算的。由于该算法在预测下一时刻用户行为时只跟当前用户行为状态有关，即在给用户进行推荐时只跟用户最后一个时刻的行为有关。因此该算法是处理用户序列行为数据的基准模型。

LSTM：该算法通过循环神经网络来学习用户的行为序列特征，从而

中国知网 https://www.cnki.net

预测用户在下一时刻点击每一个项目的概率。在推荐时只需将预测概率高的项目推荐给用户即可。具体的模型结构和算法步骤可见 3.1 节。

在这些方法中，我们设置 User-KNN 中的最近邻个数取 80。BPR-MF 模型中设置隐空间维度为 200。LSTM 模型中参数设置同 R-LSTM 中相应的参数设置。

3.4.4 实验结果与分析

实验一：我们在两个 MovieLens1M 数据集和 MovieLens10M 数据集上测试了我们模型的结果。首先我们需要对 MovieLens 数据集进行预处理。首先需要对样本去重。然后按照用户以及用户对电影评分的时间进行排序，得到了每个用户按照时间顺序评分的电影序列。其次将用户评分电影个数少于 2 个的样本删除并且将电影流行度小于 5 的电影删除。最后对得到的用户和电影进行重新编码。最终我们得到了每个用户评分的电影序列。

对于每个数据集，我们都按照用户数量分成三部分。首先选出 N 个用户及其对应的所有评分集合作为测试集，然后再选出 N 个用户和其对应的所有评分集合作为校验集，剩下的用户作为训练集。参考 Devooght 论文中的做法[62]，我们设置 N 为 500，LSTM 网络隐层神经元的个数为 50，LSTM 模型的长度为 30。我们采用 Adagrad 作为优化器，其中学习率设置为 0.1。在测试中，我们将每条测试序列均分成两部分，使用序列的前半部分作为输入，使用剩下部分来验证最后预测结果的精度。我们设置目标项目个数 o=10，推荐项目个数 k=1 并且 HS={4, 5}。需要说明的是这里目标项目个数取 10 个，是取测试序列后半部分的前 10 个作为推荐的目标集合。

表格 3.1 展示了在 MovieLens 1M 和 MovieLens 10M 两个数据集上，对比方法 POP，User-KNN，MC 和 LSTM 以及本文方法 R-LSTM 在 NPS 和 HRPS 上的结果。经过实验得到的结果如下表 3.1 所示。从中可以发现本文的方法比其它对比方法取得了更好的效果。在 MovieLens 1M 数据集上，R-LSTM 在指标 NPS 上的结果为 25.45%。在 MovieLens 10M 数据集上 R-LSTM 在 NPS 上取得了 27.15%的结果。其它没有考虑到序列信息的方法最后的结果都低于 11%。

中国知网 https://www.cnki.net

表3.1　各算法在MovieLens 1M和10M数据集上的结果

数据集	算法	NPS（%）	HRPS（%）
MovieLens 1M	POP	6.80	4.80
	BPR-MF	8.20±0.35	6.43±0.30
	User-KNN	10.00	7.80
	MC	18.80	14.80
	LSTM	25.10±1.52	16.25±0.84
	R-LSTM	**25.45±0.81**	**17.90±0.81**
MovieLens 10M	POP	6.40	4.60
	BPR-MF	8.57±0.70	6.40±0.81
	User-KNN	10.60	6.80
	MC	19.80	9.80
	LSTM	**27.65±0.66**	15.65±0.96
	R-LSTM	27.15±0.76	**17.20±1.08**

由于目标的项目并不都是用户高评分的项目，所以本文的方法 R-LSTM 在 NPS 上结果有时会没有 LSTM 模型好。如在 MovieLens 10M 数据集中，LSTM 在 NPS 上取得了 27.65%的结果，比 R-LSTM 取得的 27.15%的结果要高。然而 LSTM 模型只考虑了输入的序列信息，忽略了用户对不同的项目有不同的评分，它认为所有输入的项目除了序列前后差别外没有其它不同，因此在 HRPS 的指标下其结果表现的并不好。

实验二：在 MovieLens 1M 数据集上为了更好的进行实验对比，我们分别对推荐项目的个数 k 和目标项目的个数 o 进行了不同数值的测试。

（1）当固定 k 取 1 时，o 的取值分别为{5，10，15，20}。实验结果如下图 3.4 所示。从图中可以看出，随着目标项目个数的逐渐增多，LSTM 和 R-LSTM 在两个指标上的结果都逐渐增长，但增速都逐渐放缓。说明在给用户进行下一时刻电影推荐时，循环神经网络可以很好的捕获用户的行为序列信息，进而给用户推荐跟其当前行为最相关的电影。另外 LSTM 和 R-LSTM 在 NPS 上结果类似，但在 HRPS 上 LSTM 的结果明显低于我们的方法。尤其是当推荐目标数目在 10 个时，R-LSTM 在 HRPS 上取得了

中国知网　https://www.cnki.net

17.9%的准确率，比 LSTM 的 16.25%高了 1.65%。这说明本文的方法在高评分推荐中效果更具优势。

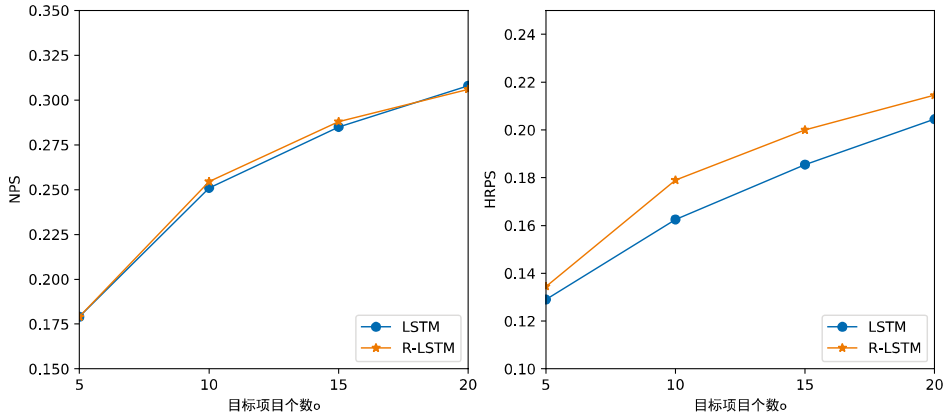

图3.4　不同目标项目个数上的准确率

（2）当固定目标项目个数 o 取 20 时，推荐项目个数 k 为{1, 5, 10, 15}。实验结果如下图 3.5 所示。通过对实验结果进行比较，我们发现在 NPS 指标下两个模型的结果几乎近似。当推荐项目 o 为 15 时，LSTM 在 NPS 中的结果时 86.75%，R-LSTM 的结果为 86.85%。但在 HRPS 的结果中，R-LSTM 效果要明显高于 LSTM。比如同样在 o 取 15 时，R-LSTM 结果为 47.9%比 LSTM 的结果高了 2.2%。

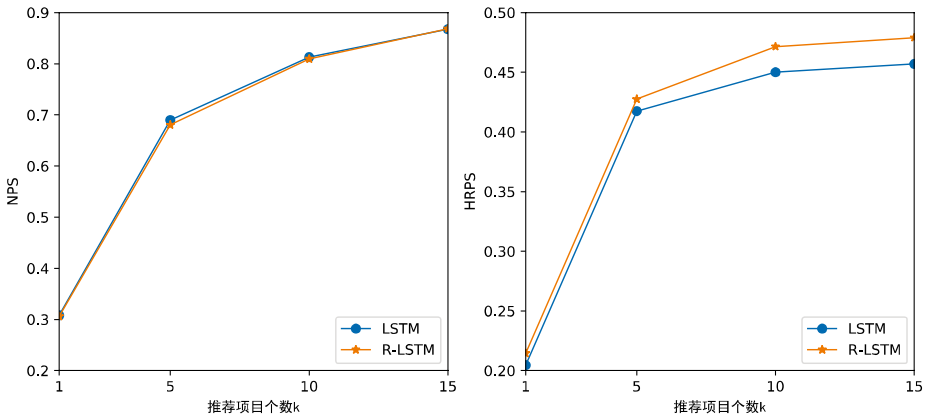

图3.5　不同推荐项目个数上的准确率

实验三：在 MovieLens 1M 数据集中，我们通过使用不同对测试集的

中国知网　https://www.cnki.net

划分方式来检测本文模型的有效性。在上述实验中，我们把测试集中每个用户的评分序列中前半部分作为输入，后半部分序列作为目标序列。在这里我们采用了对测试集序列的不同划分方式，下面表格 3.2 展示了最后的结果。

表3.2　各算法在MovieLens 1M测试集不同划分方式下的结果

不同划分比例	算法	NPS（%）	HRPS（%）
1：4	POP	10.60	7.80
	User-KNN	14.60	9.60
	MC	22.60	10.20
	LSTM	**32.00±1.41**	17.50±1.04
	R-LSTM	30.65±0.84	**19.20±1.42**
1：3	POP	10.40	7.40
	User-KNN	13.80	8.80
	MC	19.20	11.05
	LSTM	27.80±1.82	15.70±1.31
	R-LSTM	**30.10±0.87**	**19.30±0.74**
1：2	POP	9.60	7.80
	User-KNN	13.40	9.80
	MC	17.80	10.60
	LSTM	27.40±2.08	16.15±1.04
	R-LSTM	**28.50±0.48**	**19.25±0.30**

从上述表 3.2 中可以发现，LSTM 和 R-LSTM 在 NPS 和 HRPS 结果上远远高于其它对比方法。在基于循环神经网络模型的方法中，第一种 1：4 的划分方式即把序列的前五分之一作为输入，然后进行预测的结果要明显高于其它对用户行为序列的划分方式，并且随着对用户行为序列划分比例的增加，用户和项目交互矩阵的稀疏度也逐渐增加，而对用户下一时刻行为预测的准确率却逐渐减少。可以看出基于循环神经网络的协同过滤算法对数据集的稀疏性并没有特别敏感，因为通过序列信息可以弥补用户和

中国知网　https://www.cnki.net

项目之间行为交互的稀疏性。

3.5 本章小结

在本章中，针对基于循环神经网络的推荐算法模型，只根据用户对电影评分的序列来预测用户在下一时刻将要评分的电影，忽略了用户对电影有不同的评分喜好。为了让推荐算法模型既能学习到用户行为的序列信息，又能学习到用户对点击项目的不同喜好信息，本文提出了基于评分偏好的循环神经网络模型。这种方法可以给用户推荐下一时刻既是用户将要点击的项目，同时也是用户喜欢的项目。我们在两个电影评分数据集上进行了实验，证明了模型的有效性。

中国知网 https://www.cnki.net

第 4 章 基于深度语义匹配模型的循环神经网络推荐算法

由于深度学习摒弃了人工设计提取特征的局限，它可以通过神经网络模型学习到数据更本质的特征。目前深度学习在基于内容的推荐方法中应用最广的就是深度语义匹配模型。该模型克服了传统基于内容的推荐系统中，很难获取到用户特征的问题。DSSM 模型通过从项目的内容中提取到项目的隐表示，以及从用户的历史行为数据中获取到用户的隐表示，然后通过计算用户隐表示和项目隐表示的匹配度来进行推荐。但是在该模型中认为用户特征是固定的，忽略了时间因素对用户兴趣爱好的影响。在本章中，我们提出了一种基于深度语义匹配模型的循环神经网络推荐算法(DSSM-based LSTM, DSSM-LSTM)，该算法首先通过循环神经网络提取到用户当前兴趣的隐表示，然后通过计算用户当前兴趣隐表示和项目隐表示之间的匹配度进行推荐。

4.1 基于深度语义匹配模型推荐算法

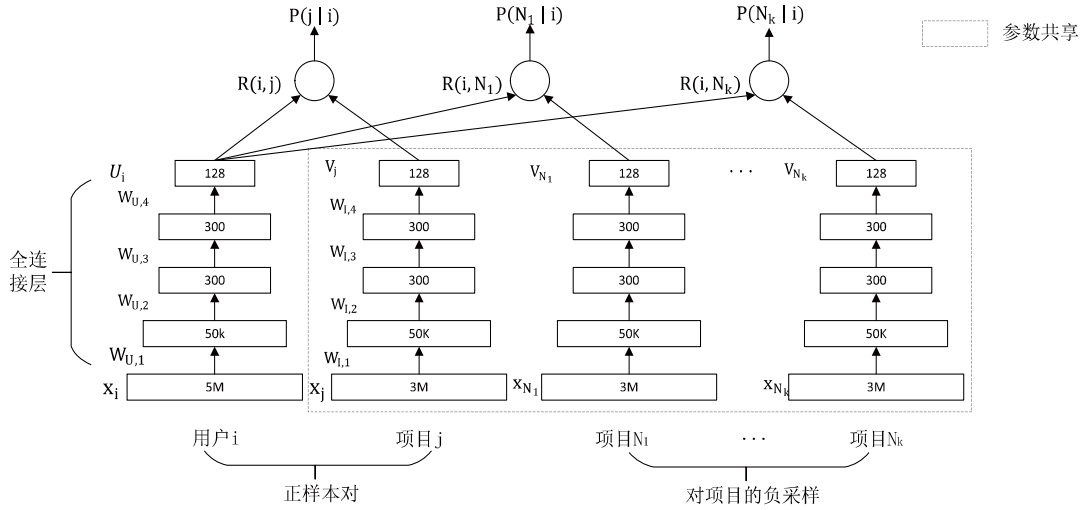

图4.1　基于深度语义匹配模型推荐算法的模型图

深度语义匹配模型[52]最早是微软在 2013 年提出的。当时 DSSM 模型主要用于搜索场景中，该模型通过计算搜索词和文章之间的语义相似度来优化模型。模型的目的是通过搜索词来给出与其语义信息最相似的文章的

中国知网 https://www.cnki.net

排序。后来在 2015 年微软提出了基于多视角的深度语义匹配模型[53]，首次将该模型应用在跨领域的推荐系统中。下面主要介绍 DSSM 模型在单一领域的推荐算法。

在上图 4.1 模型中，左边的网络表示通过全连接神经网络提取用户特征，并把原始高维的用户特征映射到低维空间中。右边网络表示同样也是用全连接神经网络的方法提取项目特征，并把原始高维的项目特征映射到和用户相同的低维语义空间中。我们的目标是让具有交互行为的用户和项目在相同的低维语义空间中的特征尽可能相似。以用户和项目之间的隐式反馈为例，假设用户 i 点击了项目 j，经过全连接神经网络之后学习到的用户低维特征表示为U_i，以及项目低维特征表示为V_j。我们用余弦相似度来表示他们之间的关系$R(i,j)$。

$$R(i,j) = cosine(U_i, V_j) = \frac{U_i^T V_j}{\|U_i\|\|V_j\|} \tag{4.1}$$

在具体的训练过程中，我们会对每一个用户项目样本对进行负采样。图 4.1 中表示对(i，j)样本对进行了 k 次负采样，其中负采样得到的项目列表为$[N_1, N_2, \cdots, N_k]$。我们用 D 表示所有与该用户相关的项目集合，其中包括与用户 i 相关的正例D^+和与用户 i 相关的采样出来的负例D^-。在计算完这些项目和用户之间的余弦相似度之后，我们用 softmax 函数对最后的结果进行归一化。

$$P(D^+|i) = \frac{exp(\gamma R(i,D^+))}{\sum_{D' \in D} exp(\gamma R(i,D'))} \tag{4.2}$$

其中γ是 softmax 函数的平滑系数。模型的目标函数就是对于给定用户 i 时其正样本的最大似然。我们取其负对数作为最后的损失函数：

$$L = -log \prod_{(i,D^+)} p(D^+|i) = -\sum_{i,D^+} log\, p(D^+|i) \tag{4.3}$$

我们可以使用随机梯度下降的方法最小化上述的损失函数从而得到模型的最优参数。在给用户进行推荐的时候，只需要得到用户的低维嵌入表示，然后计算用户的低维特征和采样项目对应的低维特征之间的余弦相似度，最后将相似度高的项目推荐给用户。

中国知网 https://www.cnki.net

4.2 基于深度语义匹配模型的循环神经网络推荐算法

为了解决用户兴趣爱好会随时间变化而动态变化的问题，我们采用循环神经网络模型来获取用户兴趣爱好的变化。在循环神经网络模型中，为了解决因为序列输入过长导致的梯度消失问题，最成功的改进是长短时记忆网络模型。因此我们主要基于 LSTM 构建深度语义匹配模型。DSSM-LSTM 模型主要通过把用户按照时间顺序点击的项目序列输入到 LSTM 模型中，LSTM 模型会捕获到用户对下一时刻项目的兴趣特征隐表示，然后通过计算它和下一时刻用户将要点击项目的隐表示之间的匹配度来优化模型。DSSM-LSTM 模型图如下所示：

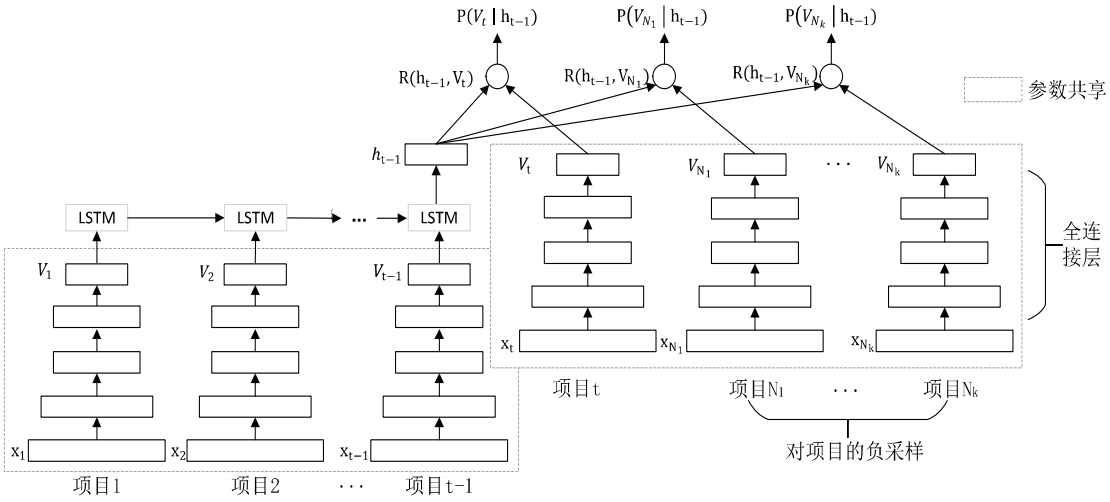

图4.2　基于深度语义匹配模型的循环神经网络推荐算法的模型图

DSSM-LSTM 模型主要通过使用 LSTM 来捕获用户兴趣的变化，以用户 i 在时刻 t 之前点击的行为序列$\chi_{i,t}$为例，目的是通过用户点击的项目序列$\chi_{i,t}$来预测用户在时刻 t 时要点击的项目。我们可以把该问题转化成排序问题，即我们要预测在时刻 t 点击的正样本D^{+}在 k 次负采样得到的负样本集合D^{-}中的排名。首先我们通过全连接层把项目原始特征映射到隐空间 V 中，其中包括对负采样的项目也进行同样的映射，即上图中用虚线框框起来的所有部分，它们之间参数共享。用公式表示为：

$$[V_1,\cdots,V_t,V_{N_1},\cdots,V_{N_k}]=f_{DNN}([x_1,\cdots,x_t,x_{N_1},\cdots,x_{N_k}],\theta_{DNN}) \quad (4.4)$$

其中θ_{DNN}表示模型映射函数f_{DNN}的参数。然后把用户按时间顺序点击的项

中国知网 https://www.cnki.net

目隐特征$[V_1, V_2, \cdots, V_{t-1}]$作为 LSTM 网络的输入，通过 LSTM 模型学习到用户 i 在 t 时刻时将要点击项目的特征h_{t-1}。用公式表示为：

$$\hbar_{t-1} = f_{LSTM}(h_{t-2}, V_{t-1}; \theta_{LSTM}) \quad (4.5)$$

其中θ_{LSTM}表示模型映射函数f_{LSTM}的参数，并且h_{t-1}为 V 空间中的一个向量表示，即h_{t-1}和$[V_t, V_{N_1}, \cdots, V_{N_D}]$都为同一隐空间的特征表示。为了避免把经过 LSTM 学习到的特征h_{t-1}经过全连接网络映射到 n 维时，导致模型参数量过多的问题。我们分别h_{t-1}和V_t以及 $[V_{N_1}, V_{N_2}, \cdots, V_{N_k}]$之间的匹配度，这里我们用余弦相似度来衡量它们之间的关系。用公式表示为：

$$R(h_{t-1}, V_j) = cosine(h_{t-1}, V_j) = \frac{h_{t-1}^T V_j}{\|h_{t-1}\| \|V_j\|} \quad (4.6)$$

通过计算完特征的匹配度之后，对计算得到的余弦值进行带γ平滑的 softmax 归一化如下式所示：

$$P(D^+|h_{t-1}) = \frac{exp(\gamma R(h_{t-1}, D^+))}{\sum_{D' \in D} exp(\gamma R(h_{t-1}, D'))} \quad (4.7)$$

最后通过随机梯度下降的方法优化交叉熵损失即正类的负对数似然来优化整个模型。最后的代价函数如下所示：

$$L = -log \prod_{(h_{t-1}, D^+)} p(D^+|h_{t-1}) = -\sum_{h_{t-1}, D^+} log\, p(D^+|h_{t-1}) \quad (4.8)$$

对比于第 3.1 节中介绍的基于循环神经网络的协同过滤推荐算法中，以用户的点击行为序列作为输入，预测用户在下一时刻要点击的项目。该算法把这一问题转化成对于项目的多分类问题，即把每一个项目看成一类，判断在下一时刻时，用户要点击每个项目的概率，把预测概率最高的项目作为最后的预测类别。这种做法有一个严重的问题，就是当项目数量特别多的时候，也就是我们所要分的类别数量特别多的时候，我们需要对每一类计算属于该类的概率，这样模型的参数量会很大，而且计算量也比较大，模型收敛缓慢。

4.3 结合用户特征的基于深度语义匹配的循环神经网络推荐算法

虽然上一节中提到将循环神经网络模型加入到深度语义模型中，可以

中国知网 https://www.cnki.net

动态的捕获用户兴趣的变化，来进行序列推荐。该模型主要通过学习用户点击项目序列的变化，进而得到用户当前的喜好。虽然基于循环神经网络模型可以动态捕获用户兴趣的变化，但当用户输入的点击项目序列过长时，会导致 LSTM 模型梯度消失，相应的模型也就无法学习。为了让模型可以学习到更有效的用户特征表达，我们把用户的特征和项目的特征进行拼接，一起作为 LSTM 模型的输入。这样就可以避免当输入用户点击序列过短时造成 LSTM 学习用户特征不充足，进而造成推荐精度不高的缺点。

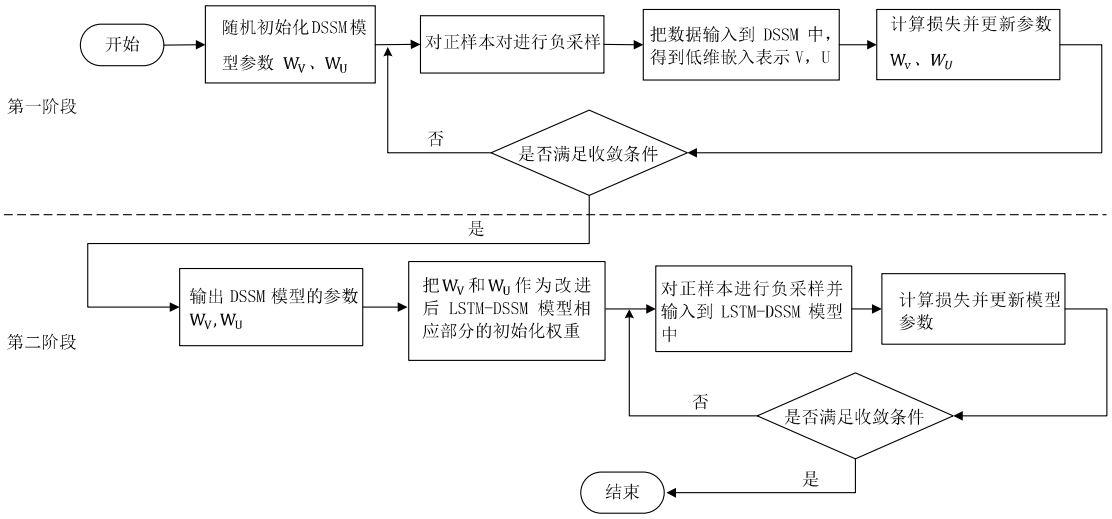

图4.3　加快DSSM-LSTM模型收敛的流程图

为了加快结合用户特征的 DSSM-LSTM 模型(Users' features DSSM-LSTM, DSSM-LSTM(U))收敛速度，我们可以先用 DSSM 模型学习到用户和项目的隐表示，然后用 DSSM 模型的参数作为改进后 DSSM-LSTM 模型相应部分的初始化参数，然后再训练 DSSM-LSTM 模型。其流程图如图 4.3 所示。

4.4　实验与分析

4.4.1 数据集介绍

在实验中，我们主要用到两个数据集来对比实验结果。第一个数据集是 MovieLens1M 数据集，这个数据集的详细介绍可见第 3.3 节。第二个数

中国知网　https://www.cnki.net

据集是 Globo 网站提供的新闻点击数据集②。Globo 是巴西最受欢迎的新闻门户网站，每个月将有超过 8000 万的独立用户和 10 万的新文章内容。该数据样本包扩了在 2017 年 10 月 1 日到 16 日期间的用户交互行为信息，每一个小时内的用户行为作为一个文件。其中包括了超过 300 万次的点击行为，而这些行为分布在大约 33 万用户的 120 万会话行为中。他们在此期间共阅读了 5 万多篇不同的新闻文章。在每个小时内用户行为数据中，主要包括字段为用户 id、会话 id、会话开始的时间、会话长度、点击的文章 id、点击的时间。该表内容数据主要如下表 4.1 所示。对于用户点击的新闻内容信息由于敏感性等因素，只提供了每篇文章 250 维的隐向量表示。

表4.1　Globo新闻数据集中点击行为表的前五行

用户 id	会话 id	开始的时间	长度	文章 id	点击的时间
0	1	1506825423000	2	157541	1506826828020
0	1	1506825423000	2	68866	1506826858020
1	2	1506825426000	2	235840	1506827017951
1	2	1506825426000	2	96663	1506827047951
2	3	1506825435000	2	119592	1506827090575

Globo 数据集在很多方面不同于 MovieLens 评分数据集。首先在 Globo 数据集中用户的行为间顺序性更加明显。而且基于用户行为会话的数据中，序列的长度一般不会太长，这就致使用户项目交互行为矩阵中数据的稀疏度会降低，这就更加不利于基于矩阵分解类似的模型方法。

4.4.2 实验指标

实验指标为在 Top-N 推荐中的命中率(Hit Rate, HR@N)[79]和平均倒数排名(Mean Reciprocal Rank, MRR@N)[79,60]。在基于用户点击项目序列的预测中，我们定义测试集中序列样本为 K 个，对于在序列 j 中要预测的下一时刻目标项目为t^j，给序列 j 推荐的 N 个预测项目序列表示为$I_N^j = [i_1^j, i_2^j, \cdots, i_N^j]$。命中率 HR@N 是指真实点击的项目是否出现在前 N 的推荐

② Globo 新闻数据集链接：https://www.kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom/

中国知网 https://www.cnki.net

排序列表中。公式表示为：

$$\mathrm{HR@N} = \frac{\sum_{j=1}^{K} 1\{t^j \in I_N^j\}}{K} \quad (4.9)$$

平均倒数排序排名 MRR@N 是指一个排序指标，主要计算方法是把真实点击的项目在前 N 的推荐排序列表中对应排序的倒数作为它的准确度，最后再对所有的点击预测结果取平均。该指标对目标项目在列表中的排名特别敏感。定义当满足$t^j \in I_N^j$时，目标项目t^j在预测项目列表I_N^j中的排名是p_j。公式表示为：

$$\mathrm{MRR@N} = \frac{\sum_{j=1}^{K} \frac{1}{p_j} \times 1\{t^j \in I_N^j\}}{K} \quad (4.10)$$

这两个指标都是当指标的值越大时，推荐结果的质量就越高。

4.4.3 Globo 新闻数据集上的实验结果与分析

在 Globo 新闻数据集上进行了实验。该数据集是按照用户点击会话开始的时间进行逐小时划分的，并且用户在一个会话中两次点击行为之间的时间间隔不超过 30 分钟。在实验开始时，我们需要对数据进行预处理。首先，对数据集进行去重，防止因采集问题造成的数据重复。其次把数据集按照会话开始的时间以及每次点击的时间进行排序，得到在每个会话中按照时间顺序的点击序列。由于点击序列长度均值为 3 左右，所以参考 Gabriel[80]的做法，最后把点击序列中序列长度小于 2 并且大于 20 的序列去掉。由于序列长度为 1 时，没有办法用来训练模型，而点击序列太长时则认为是离群点。

为了对本文提出方法的有效性进行验证。我们主要对比了基于深度语义匹配模型 DSSM、基于循环神经网络的推荐模型 LSTM 以及本文的提出的基于深度语义匹配模型的循环神经网络 DSSM-LSTM 和添加用户特征的 DSSM-LSTM(U)模型。

在实验中，我们随机从 Globo 新闻数据集上抽取一个小时内的所有用户行为会话信息，并将该会话信息按照会话 id 取 80%作为训练集，10%作为测试集，10%作为校验集。在所有需要进行结果对比的模型中，我们都采用了基于项目流行度的负采样方法[60-62]。无论在训练集还是在测试集中，负采样个数均为 50。由于 Globo 新闻数据集中提供了所有经过对新闻内容

中国知网 https://www.cnki.net

语义分析得到的项目隐向量，维度为 250 维，因此在这些方法中就不重新学习项目的特征表示，直接使用该项目特征表示。在 DSSM 模型中用户特征学习的部分，我们只设置了一层全连接神经网络把用户的 one-hot 特征映射到 250 维，即和项目特征在同一隐空间中。在计算带γ平滑的 softmax 时，γ取 20。在 LSTM 模型中，我们设置了 LSTM 的最大输入的序列长度为 20，在 LSTM 结构中的隐层维度为 250 维，这里只用了一层 LSTM 网络。在 DSSM-LSTM 模型中我们同样采用和 LSTM 模型中一样的参数设置。在添加用户特征的 DSSM-LSTM 模型中，同样把用户的特征映射到 250 维。在 LSTM、DSSM-LSTM 以及 DSSM-LSTM(U)这三个模型中，我们采用批次为 64,学习率为 0.001 的 Adam 模型优化器进行优化，Adam 优化器里面的参数使用默认值。在 DSSM 模型中我们采用学习率为 0.01 的 Adam 优化器对模型进行优化，其中 Adam 优化器中参数取默认值。我们分别测试了四种模型在 HR@1、HR@3、MRR@3、HR@5 和 MRR@5 指标上的结果，如下表所示：

表4.2　各算法在Globo新闻数据集中的结果

算法	HR@1	HR@3	MRR@3	HR@5	MRR@5
DSSM	0.600	0.664	0.628	0.717	0.641
LSTM	0.477	0.555	0.508	0.603	0.519
DSSM-LSTM	0.481	0.558	0.512	0.605	0.523
DSSM-LSTM(U)	0.844	0.844	0.844	0.845	0.844

从上表 4.2 的结果我们可以看出 LSTM 和 DSSM-LSTM 的方法结果比较接近，但 DSSM-LSTM 模型的结果比 LSTM 模型的结果都高一点，证明了 DSSM-LSTM 模型的有效性。而 DSSM 模型的结果远远高于 LSTM 和 DSSM-LSTM 模型的结果。通过分析我们得到造成这个结果的两点原因：第一，经统计数据集中用户行为序列长度均值为 3，由于序列平均长度太短导致循环神经网络模型的效果并没有凸显出来。第二，DSSM 模型相比于 LSTM 和 DSSM-LSTM 模型会额外用到了用户信息。但通过 DSSM 模型结果相对较高说明用户点击的项目之间相似度比较高，即这是一个用户兴趣爱好比较集中的场景。在考虑用户特征的 DSSM-LSTM 模型中，我们可以看到其结果在 HR 和 MRR 指标上都达到了最优，在 HR@1 上的结果

中国知网 https://www.cnki.net

为 84.4%，在 HR@3 和 HR@5 上的结果都和 HR@1 上的结果近似，而且 MRR@3 的结果和 HR@3 的结果相近，MRR@5 和 HR@5 的结果相近。这说明考虑用户特征的 DSSM-LSTM 模型在序列预测方面能够很好的学习到用户下一时刻点击项目的特征。

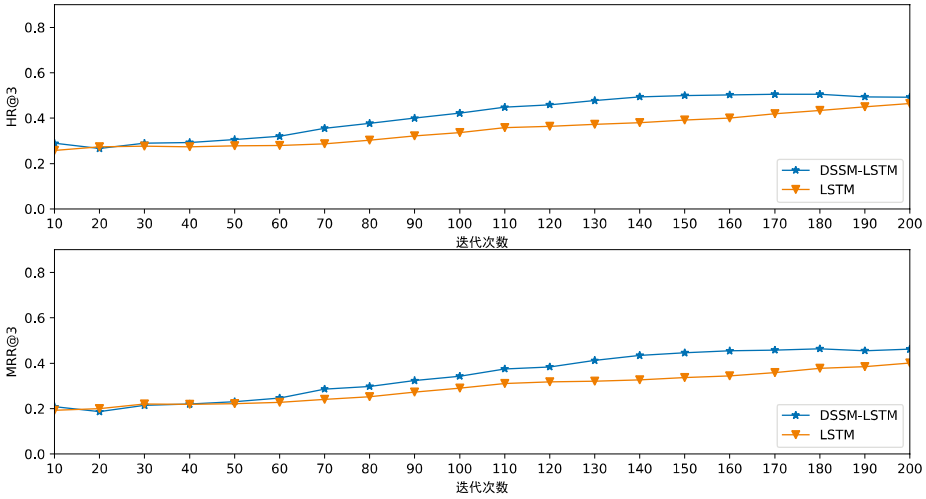

图4.4　随着训练批次的增加各方法在校验集上的结果变化图

上图 4.4 中展示了随着在训练集中迭代次数的增加，DSSM-LSTM 和 LSTM 这两种方法在校验集上的 HR@3 以及 MRR@3 的结果的变化过程。从中我们可以看出，DSSM-LSTM 模型收敛的速度和效果要比 LSTM 模型好。这是因为 LSTM 模型需要学习从循环神经网络的输出到所有项目之间全连接层的参数，而 DSSM-LSTM 模型是根据余弦相似度来得出项目之间的匹配程度，不需要额外学习这些参数。

4.4.4 MovieLens 1M 数据集上的实验结果与分析

在 MovieLens 1M 数据集上进行了该实验。在这个数据集中，我们只考虑用户给电影的评分序列，不考虑用户给电影的评分对模型造成的影响。首先对数据集进行预处理。预处理方法同第 3.3.4 节中实验的预处理方式。经过样本去重，按照用户以及用户对电影评分的时间进行排序，经统计用户评分的电影序列长度均值为 165。将用户评分电影个数少于 2 个的样本删除并且将电影流行度小于 5 的删除后，最终得到 6040 个用户观看的电

中国知网　https://www.cnki.net

影序列，共观看了 3706 部电影。对于每条用户观看的电影序列，我们划分前 80%的序列作为训练，后面 20%的序列作为测试。对于所有的 6040 个用户的电影序列，我们从中取 10%的序列用其测试部分作为校验，来检测模型的效果。从中另取 10%的序列用其测试部分作为测试，最终用来衡量模型的效果。

对比方法我们采用了 LSTM、DSSM-LSTM 以及 DSSM-LSTM(U)模型。由于在实验过程中对项目进行负采样时，可能是因为用户行为序列太长或者是项目数量过多，导致循环神经网络很难收敛。因此在下面的实验中没有对项目进行负采样，也就没有将 DSSM 模型列入对比算法中。

在 LSTM 模型中，我们首先把每个电影的 one-hot 表示映射到 50 维，然后再作为循环神经网络的输入。其中循环神经网络隐层神经元个数为 50 维，序列的最大长度设置为 30。在 DSSM-LSTM 模型中同样先把电影的 one-hot 表示映射到 50 维，其循环神经网络中参数和 LSTM 模型设置的一样。只不过 LSTM 模型需要将循环神经网络的输出映射到 3706 维，从而计算下一时刻点击每个电影的概率。而 DSSM-LSTM 模型只需要将循环神经网络的输出和所有 3706 个电影的 50 维嵌入向量之间计算余弦相似度，然后再计算带λ的 softmax 即可，其中λ设置为 20。在带用户特征的 DSSM-LSTM 模型中，我们额外的把用户的 one-hot 向量映射到 50 维，和电影映射之后的 50 维进行拼接，作为循环神经网络的输入部分。在这些模型训练过程中我们采用批次为 128，学习率为 0.001 的 Adam 模型优化器进行优化，Adam 优化器里面的参数使用默认值。我们分别测试了在 HR@5、MRR@5 和 HR@10 以及 MRR@10 上的结果。

从下表 4.3 中的结果，我们可以得出由于没有对电影进行负采样，所以模型结果整体偏低。LSTM 模型和 DSSM-LSTM 模型在结果上比较接近，但是 DSSM-LSTM 模型结果会略微高于 LSTM 模型的结果，这说明借助于相似度度量函数可以让模型学习到更有效的特征。在 DSSM-LSTM 模型中，HR@10 的结果比 HR@5 的结果高了 9%。在 LSTM 模型中，HR@10 的结果比 HR@5 的结果高了 10%。由于 HR@10 的结果并没有远远高于 HR@5 说明这两个模型学习到了用户观看电影的序列信息。带用户特征的 DSSM-LSTM 模型在 MR@5 上的结果为 42%，MR@10 上的结果为 51%，都高于不带用户特征的 DSSM-LSTM 模型，这说明添加用户特征之后可以额外帮

中国知网 https://www.cnki.net

助网络学习到更加丰富的用户特征，从而在给用户进行序列推荐时结果也更准确。

表4.3　各算法在MovieLens 1M 电影数据集中的结果

算法	HR@5	MRR@5	HR@10	MRR@10
LSTM	0.313	0.191	0.418	0.206
DSSM-LSTM	0.332	0.206	0.422	0.218
DSSM-LSTM(U)	0.421	0.273	0.519	0.287

为了对 LSTM、DSSM-LSTM 模型进行更详细的描述。下图 4.5 是在模型在训练过程中，随着在训练集上迭代次数的增加，LSTM 和 DSSM-LSTM 模型在校验集上的结果变化图。横坐标表示模型在训练集中迭代的次数，横坐标取了两个模型的前 19000 个批次，纵坐标分别为 HR@5 和 MRR@5 的值。

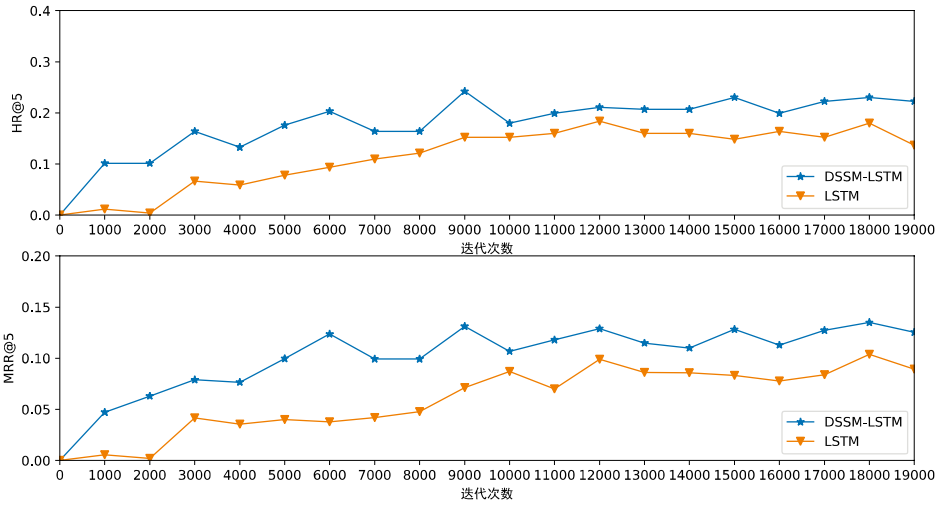

图4.5　随着训练批次的增加各方法在校验集上的结果变化图

从图 4.5 中我们可以看出 DSSM-LSTM 模型在迭代时，无论时 HR@5 的结果还是 MRR@5 的结果，都明显高于 LSTM 的方法，并且该模型很快就达到了收敛。证明了 DSSM-LSTM 模型结合余弦相似度的代价函数可以帮助模型学习到更有效的特征，同时减少了模型参数加快了模型的收敛。

中国知网　https://www.cnki.net

4.5 本章小结

在本章中，针对基于深度语义匹配模型的推荐方法中，无法捕获到用户动态的兴趣变化这一问题，提出了基于深度语义匹配模型的循环神经网络。一方面该算法可以动态的捕获用户当前的兴趣爱好，从而对用户当前的需求进行更精准的推荐。另一方面结合深度语义模型，可以减少模型的参数量，使得模型更容易收敛。同时为了让 DSSM-LSTM 模型能够捕获到更有效的用户特征，我们提出了带用户特征的 DSSM-LSTM 模型。该方法无论在推荐命中率还是在平均倒数排名中都取得了更好的结果。为了加快带用户特征的 DSSM-LSTM 模型的收敛，我们提出了先训练 DSSM 模型，并用该模型的参数作为 DSSM-LSTM 模型参数初始化的方法，来加快提高模型训练效率。

中国知网 https://www.cnki.net

第 5 章 总结和展望

5.1 本文工作总结

在个性化推荐系统中，我们介绍了一些常用的推荐算法。其中包括：基于近邻的协同过滤、基于矩阵分解的协同过滤、以及基于深度学习的推荐算法等。由于随着各大门户网站、电商等平台上用户量和项目量越来越多，导致用户和项目之间交互行为的数据量也越来越大。基于传统的推荐方法很难有效的利用大量的用户行为数据进而提取更加复杂有效的特征。而且尤其是在类似新闻网站等时效性较强的网站中，用户和项目交互行为稀疏、项目数量的快速增长、项目价值随时间不断衰减以及用户兴趣的不断变化等因素都使得推荐变得困难。

随着深度学习的方法在推荐系统中得到越来越多的重视，为了更好的学习到用户兴趣的动态变化，越来越多国内外研究学者投入到研究将循环神经网络和推荐系统进行更好的结合，达到提高推荐精度的目的。本文研究了现有的基于循环神经网络的推荐算法，针对该方法忽略了用户对项目的不同评分喜好以及模型收敛慢的不足，提出了两种提高模型精度的方法。第三章中由于在基于循环神经网络的推荐方法中，只考虑了用户点击项目先后序列对模型的影响，忽略了用户在点击不同项目时存在着不同的评分喜好问题，提出了基于评分偏好的循环神经网络推荐算法。第四章中由于基于深度语义匹配模型无法学习到用户动态兴趣变化的问题，提出了基于深度语义匹配模型的循环神经网络。该方法无论在模型结果上，还是模型收敛速度上都取得了比较好的效果。为了让该模型学习到更有效的用户特征，因此我们提出了结合用户特征的循环神经网络语义匹配模型。在模型优化过程中，提出了使用基于深度语义匹配模型的参数作为该模型相应参数的初始化的方法，加快模型收敛。本文还合理的设计了相关实验，并且针对不同的数据集和不同的验证指标比较了本文算法和相关推荐算法的结果，并对实验结果进行了具体的分析。

中国知网 https://www.cnki.net

5.2 本文的主要特色

本文针对在推荐系统中用户兴趣的动态变化问题以及如何获取更有效的用户信息进行了研究和探索。在基于循环神经网络的基础上进行了改进，提高了推荐系统的运行效率以及推荐准确性。总而言之，本文的主要特色及改进如下：

（1）针对在基于循环神经网络的推荐算法中，只考虑了将用户点击的项目序列作为循环神经网络的输入来预测下一时刻用户将要点击的项目，忽略了用户对不同项目有不同的喜好这一因素的对模型的影响。因此本文在该模型的基础上提出了基于评分偏好的循环神经网络模型。在该模型中，本文将用户对不同项目的评分作为对项目的偏好，让模型不仅可以学习到由于用户点击行为序列对结果造成的影响，而且还可以学习到对不同项目的用户喜好。这样该模型在给用户做推荐预测的时候，推荐的项目不仅仅是用户在下一时刻将要点击的项目而且还是用户喜欢的项目。为了更好的衡量该算法的效果，本文提出了两个实验指标，第一个是在下一时刻项目预测的准确率，第二个是高评分项目预测的准确率。在实验中发现在对用户行为序列预测场景中，基于循环神经网络的方法比其它对比方法效果要好很多。而且在用户和项目交互稀疏的情况下，该方法也能取得良好的效果。

（2）针对基于深度语义匹配模型的推荐方法，忽略了用户兴趣会随时间而动态变化。本文提出了基于深度语义匹配模型的循环神经网络的推荐算法。该算法将循环神经网络和深度语义匹配模型相结合，一方面可以通过循环神经网络捕获用户的动态兴趣变化，另一方面通过结合深度语义匹配模型，减少了模型参数，提高了模型收敛的速度。为了让模型更有效的捕获用户特征，在 DSSM-LSTM 模型的基础上，提出了结合用户特征的 DSSM-LSTM。为了让该模型可以更快的收敛，本文提出了用训练好的 DSSM 模型的参数作为 DSSM-LSTM 模型相应部分的参数初始化。最后我们分别在新闻数据集和电影数据集上测试了结果。由于 DSSM-LSTM 模型在预测时不需要额外再训练全连接层，该模型参数量明显少于 LSTM 模型。并且结合余弦相似度的代价函数可以帮助模型学习到更有效的特征，提高了模型的收敛速度以及推荐精度。

中国知网 https://www.cnki.net

5.3 未来工作展望

本文考虑到用户兴趣爱好会随着时间而动态变化，提出了两种基于循环神经网络的推荐算法。通过在不同数据集上进行的实验，都表明本文提出算法的有效性和优越性。但是，在研究的过程中仍然发现存在一些问题和工作需要进一步完善和改进。主要有一下几个方面：

（1）本文在基于评分偏好的循环神经网络推荐算法中，考虑到除了用户点击项目的序列信息外，用户对点击过的不同项目存在着不同的喜好，通过利用用户对不同项目的评分来作为学习用户的偏好。由于在现实推荐场景中，像用户评分这样的显示反馈比较难获取。如何设计更为有效的获取用户偏好模型，能够让推荐系统自动抓住项目序列中最应关注的特征，从而达到提高推荐精度的目的。

（2）本文在基于深度语义匹配模型的循环神经网络推荐算法中，考虑到添加用户特征到 DSSM-LSTM 模型中，这里添加的用户特征只是将用户的 one-hot 向量映射到和项目隐表示相同的维度上。为了模型学习到更有效的用户特征，可以考虑将更多的辅助信息加入到模型中，如用户使用的通讯设备、新闻文章的长短、新闻发表时间等。

（3）本文提出的基于循环神经网络的相关推荐方法，都是只考虑到用户兴趣爱好会随时间变化而变化，没有考虑到项目的特征也会随时间变化而变化。如在时效性强的推荐场景中，随着新闻文章发表的时间越久，其价值也就越低。可以探索更准确的捕获项目特征的方法。

（4）本文提出的基于循环神经网络的相关推荐方法中，都使用了 LSTM 网络结构来捕获用户兴趣的动态变化。其中 LSTM 网络的参数的设置，包括 LSTM 网络序列的长短、梯度截断以及学习率。参数的选择问题也是一个难点。太长的序列输入会导致梯度爆炸，而太短的序列输入可能会让模型学习不到序列对推荐结果的影响。如何根据不同的数据集制定合理的参数选择机制，需要进一步探索。

中国知网 https://www.cnki.net

中国知网 https://www.cnki.net

参考文献

[1] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions[J]. IEEE Transactions on Knowledge & Data Engineering, 2005, 17(6): 734~749.

[2] Lam S K, Riedl J. Shilling recommender systems for fun and profit[C]. In: Proceedings of the 13th international conference on World Wide Web. New York: ACM, 2004. 393~402.

[3] Gomez-Uribe C A, Hunt N. The netflix recommender system: Algorithms, business value, and innovation[J]. ACM Transactions on Management Information Systems (TMIS), 2016, 6(4): 13.

[4] Davidson J, Liebald B, Liu J, et al. The YouTube video recommendation system[C]. In: Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010. 293~296.

[5] Melville P, Mooney R J, Nagarajan R. Content-boosted collaborative filtering for improved recommendations[J]. American Association for Artificial Intelligence, 2002, 23: 187~192.

[6] Ricci F, Rokach L, Shapira B, et al. Recommender Systems Handbook[M]. In: Proceedings of the Recommender systems handbook, 2011. 74~99.

[7] Herlocker J L, Konstan J A, Borchers A, et al. An algorithmic framework for performing collaborative filtering[C]. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999. 230~237.

[8] Herlocker J, Konstan J A, Riedl J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms[J]. Information retrieval, 2002, 5(4): 287~310.

[9] Leung C W, Chan S C, Chung F. A collaborative filtering framework based on fuzzy association rules and multiple-level similarity[J]. Knowledge and Information Systems, 2006, 10(3): 357~381.

[10] Peng Y, Zhu W, Zhao Y, et al. Cross-media analysis and reasoning: advances and directions[J]. Frontiers of Information Technology & Electronic Engineering, 2017, 18(1): 44~57.

[11] ADOMAVICIUS G. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions[J]. IEEE Trans. Knowl. Data Eng., 2005, 17(6): 734~749.

[12] Bell R M, Koren Y. Scalable collaborative filtering with jointly derived neighborhood interpolation weights[C] In: Proceedings of the ICDM. IEEE, 2007. 43~52.

[13] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C] In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., 1998. 43~52.

中国知网 https://www.cnki.net

[14] Deshpande M, Karypis G. Item-based top-n recommendation algorithms[J]. ACM Transactions on Information Systems (TOIS), 2004, 22(1): 143~177.

[15] Linden G, Smith B, York J. Amazon. com recommendations: Item-to-item collaborative filtering[J]. IEEE Internet computing, 2003 (1): 76~80.

[16] Hofmann T. Latent semantic models for collaborative filtering[J]. ACM Transactions on Information Systems (TOIS), 2004, 22(1): 89~115.

[17] Billsus D, Pazzani M J. Learning Collaborative Information Filters[C]. In: Proceedings of the ICML. 1998, 98. 46~54.

[18] Good N, Schafer J B, Konstan J A, et al. Combining collaborative filtering with personal agents for better recommendations[J]. AAAI/IAAI, 1999, 439.

[19] Sarwar B M, Konstan J A, Borchers A, et al. Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system[C]. In: Proceedings of the ACM conference on Computer supported cooperative work. ACM, 1998. 345~354.

[20] Pagare R, Shinde A. A Study of Recommender System Techniques[J]. International Journal of Computer Applications, 2013, 47(16): 1~4.

[21] Schein A I, Popescul A, Ungar L H, et al. Methods and metrics for cold-start recommendations[C]. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2002. 253~260.

[22] Xue G R, Lin C, Yang Q, et al. Scalable collaborative filtering using cluster-based smoothing[C]. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2005. 114~121.

[23] Sarwar B M, Karypis G, Konstan J, et al. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering[C]. In: Proceedings of the fifth international conference on computer and information technology. 2002, 1. 291~324.

[24] O’Connor M, Herlocker J. Clustering items for collaborative filtering[C]. In: Proceedings of the ACM SIGIR workshop on recommender systems. UC Berkeley, 1999. 128.

[25] Li Q, Kim B M. Clustering approach for hybrid recommender system[C]. In: Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003). IEEE, 2003. 33~38.

[26] Ungar L H, Foster D P. Clustering methods for collaborative filtering[C]. In: Proceedings of the AAAI workshop on recommendation systems. 1998, 1. 114~129.

[27] Grčar M, Fortuna B, Mladenič D, et al. kNN versus SVM in the collaborative filtering framework[M]. In: Proceedings of the Data Science and Classification. Berlin: Springer, 2006. 251~260.

[28] Miyahara K, Pazzani M J. Collaborative filtering with the simple Bayesian classifier[C]. In: Proceedings of the Pacific Rim International conference on artificial intelligence. Berlin: Springer, 2000. 679~689.

中国知网 https://www.cnki.net

[29] Su X, Khoshgoftaar T M. Collaborative filtering for multi-class data using belief nets algorithms[C]. In: Proceedings of the 18th IEEE international conference on Tools with Artificial Intelligence (ICTAI'06). IEEE, 2006. 497~504.

[30] Miyahara K, Pazzani M J. Improvement of collaborative filtering with the simple Bayesian classifier[J]. Information Processing Society of Japan, 2002, 43(11).

[31] Bell R, Koren Y, Volinsky C. Modeling relationships at multiple scales to improve accuracy of large recommender systems[C]. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2007. 95~104.

[32] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008. 426~434.

[33] Paterek A. Improving regularized singular value decomposition for collaborative filtering[C]. In: Proceedings of KDD cup and workshop, 2007. 5~8.

[34] Takács G, Pilászy I, Németh B, et al. Investigation of various matrix factorization methods for large recommender systems[C]. In: Proceedings of the 2008 IEEE International Conference on Data Mining Workshops. IEEE, 2008. 553~562.

[35] Takács G, Pilászy I, Németh B, et al. Scalable Collaborative Filtering Approaches for Large Recommender Systems[J]. The Journal of Machine Learning Research, 2009, 10: 623~656.

[36] Srebro N, Rennie J, Jaakkola T S. Maximum-margin matrix factorization[C]. In: Proceedings of the Advances in neural information processing systems, 2005. 1329~1336.

[37] Rennie J D M, Srebro N. Fast maximum margin matrix factorization for collaborative prediction[C]. In: Proceedings of the 22nd international conference on Machine learning. ACM, 2005. 713~719.

[38] Takács G, Pilászy I, Németh B, et al. Investigation of various matrix factorization methods for large recommender systems[C]. In: Proceedings of the IEEE International Conference on Data Mining Workshops. IEEE, 2008. 553~562.

[39] Kim D, Yum B J. Collaborative filtering based on iterative principal component analysis[J]. Expert Systems with Applications, 2005, 28(4): 823~830.

[40] Sarwar B, Karypis G, Konstan J, et al. Application of dimensionality reduction in recommender system-a case study[R]. Minnesota Univ Minneapolis Dept of Computer Science, 2000.

[41] Billsus D, Pazzani M J. Learning Collaborative Information Filters[C]. In: Proceedings of the ICML. 1998, 98. 46~54.

[42] Funk S. Netflix update: Try this at home[EB/OL].

https://sifter.org/~simon/journal/20061211.html, 2006-12-11.

中国知网 https://www.cnki.net

[43] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009 (8): 30~37.

[44] Koren Y . Factor in the Neighbors: Scalable and Accurate Collaborative Filtering[J]. ACM Transactions on Knowledge Discovery from Data, 2010, 4(1): 1~24.

[45] 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，No.427 (7): 191~219.

[46] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering[C]. In: Proceedings of the 24th international conference on Machine learning. ACM, 2007. 791~798.

[47] Phung D Q, Venkatesh S. Ordinal Boltzmann machines for collaborative filtering[C]. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, 2009. 548~556.

[48] Georgiev K, Nakov P. A non-iid framework for collaborative filtering with restricted boltzmann machines[C]. In: Proceedings of the International conference on machine learning. 2013. 1148~1156.

[49] 何洁月, 马贝. 利用社交关系的实值条件受限玻尔兹曼机协同过滤推荐算法[J]. 计算机学报, 2016, 39(1): 183~195.

[50] Sedhain S, Menon A K, Sanner S, et al. Autorec: Autoencoders meet collaborative filtering[C]. In: Proceedings of the 24th International Conference on World Wide Web. ACM, 2015. 111~112.

[51] Strub F, Mary J. Collaborative filtering with stacked denoising autoencoders and sparse inputs[C]. In: Proceedings of the NIPS workshop on machine learning for eCommerce. 2015.

[52] Huang P S, He X, Gao J, et al. Learning deep structured semantic models for web search using clickthrough data[C]. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. ACM, 2013. 2333~2338.

[53] Elkahky A M, Song Y, He X. A multi-view deep learning approach for cross domain user modeling in recommendation systems[C]. In: Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2015. 278~288.

[54] Pivk A. Item-based collaborative filtering algorithms[C]. In: Proceedings of the Proc. of the Fourth International Conference on Information Society. 2001. 65~68.

[55] Shani G , Heckerman D , Brafman R I , et al. An MDP-Based Recommender System[J]. Journal of Machine Learning Research, 2005, 6(1):1265~1295.

[56] Mikolov T , Chen K , Corrado G , et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.

中国知网 https://www.cnki.net

[57] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]. In: Proceedings of the Advances in neural information processing systems. 2013. 3111~3119.

[58] Barkan O, Koenigstein N. Item2vec: neural item embedding for collaborative filtering[C]. In: Proceedings of the 26th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2016. 1~6.

[59] Wang P, Guo J, Lan Y, et al. Learning hierarchical representation model for nextbasket recommendation[C]. In: Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, 2015. 403~412.

[60] Hidasi, Balázs, Karatzoglou A , Baltrunas L , et al. Session-based Recommendations with Recurrent Neural Networks[J]. Computer Science, 2015.

[61] Tan Y K, Xu X, Liu Y. Improved recurrent neural networks for session-based recommendations[C]. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, 2016. 17~22.

[62] Devooght R, Bersini H. Collaborative filtering with recurrent neural networks[J]. arXiv preprint arXiv:1608.07400, 2016.

[63] Zhang S, Yao L, Sun A, et al. Deep learning based recommender system: A survey and new perspectives[J]. ACM Computing Surveys (CSUR), 2019, 52(1):5.

[64] Russakovsky O, Deng J, Su H, et al. Imagenet large scale visual recognition challenge[J]. International journal of computer vision, 2015, 115(3): 211~252.

[65] Hinton G, Deng L, Yu D, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups[J]. IEEE Signal Processing Magazine, 2012, 29(6): 82~97.

[66] Goodfellow I, Bengio Y, Courville A. Deep learning[M]. In: Proceedings of the MIT press, 2016. 227~253.

[67] Krueger K A, Dayan P. Flexible shaping: How learning in small steps helps[J]. Cognition, 2009, 110(3): 380~394.

[68] Dahl G, Mohamed A, Hinton G E. Phone recognition with the mean-covariance restricted Boltzmann machine[C]. In: Proceedings of the Advances in neural information processing systems. 2010. 469~477.

[69] CireşAn D, Meier U, Masci J, et al. Multi-column deep neural network for traffic sign classification[J]. Neural networks, 2012, 32: 333~338.

[70] Cho K, Van Merriënboer B, Bahdanau D, et al. On the properties of neural machine translation: Encoder-decoder approaches[J]. arXiv preprint arXiv:1409.1259, 2014.

[71] Xue H J, Dai X, Zhang J, et al. Deep Matrix Factorization Models for Recommender Systems[C]. In: Proceedings of the IJCAI. 2017. 3203~3209.

中国知网 https://www.cnki.net

[72] Bai T, Wen J R, Zhang J, et al. A neural collaborative filtering model with interaction-based neighborhood[C]. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, 2017.1979~1982.

[73] Rendle S. Factorization machines[C]. In: Proceedings of the 2010 IEEE International Conference on Data Mining. IEEE, 2010. 995~1000.

[74] Xu J, He X, Li H. Deep learning for matching in search and recommendation[C]. In: Proceedings of the The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. ACM, 2018. 1365~1368.

[75] Rocktäschel T, Grefenstette E, Hermann K M, et al. Reasoning about entailment with neural attention[J]. In: Proceedings of the arXiv preprint arXiv:1509.06664, 2015.

[76] Dieng A B, Wang C, Gao J, et al. Topicrnn: A recurrent neural network with long-range semantic dependency[J]. In: Proceedings of the arXiv preprint arXiv:1611.01702, 2016.

[77] Miller B N, Albert I, Lam S K, et al. MovieLens unplugged: experiences with an occasionally connected recommender system[C]. In: Proceedings of the 8th international conference on Intelligent user interfaces. ACM, 2003. 263~266.

[78] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback[C]. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, 2009. 452~461.

[79] Ludewig M, Jannach D. Evaluation of session-based recommendation algorithms[J]. User Modeling and User-Adapted Interaction, 2018, 28(4-5): 331~390.

[80] Moreira G S P, Ferreira F, da Cunha A M. News Session-Based Recommendations using Deep Neural Networks[J]. In: Proceedings of the arXiv preprint arXiv:1808.00076, 2018.

中国知网 https://www.cnki.net

致谢

三年的研究生生活转眼就要结束了。这三年里无论是在学习还是生活中，我都成长了很多，也收获了很多。身边的老师和同学也都给了我很多关心和照顾，在这里我想向他们表达我的感谢。

首先要感谢的是我的导师王靖老师。王靖老师学术态度严谨、专业知识渊博、工作认真负责，对我们更是耐心指导。本课题从选题到实验，再到论文撰写修改，导师都给我提出很多建设性意见。同时导师严谨的科研精神、敏锐的洞察力和活跃的思维方式都给了我很多启迪，让我终生受益。在这里，我要再一次向王靖老师表达内心的感激与敬意。

同时我要感谢师兄师姐们的指导和帮助，感谢师弟师妹们的合作与交流，正是你们帮助我解决了许多学术和生活上的困扰，让我在科研的道路上不再孤单。同时我要感谢我的舍友和同学，与你们朝夕相伴、互相鼓励、互相帮助，是我这三年最美好的回忆。

最后我要感谢我的父母和家人，是你们在背后默默支持才让我顺利完成学业。感谢我的男朋友一直以来的陪伴和鼓励，让我无论遇到什么困难都勇敢向前。

中国知网 https://www.cnki.net

中国知网 https://www.cnki.net

中国知网 https://www.cnki.net