

**硕 士 学 位 论 文**

MASTER’S DISSERTATION

论文题目 基于上下文的深度混合推荐系统的研究

作者姓名刘建成

学科专业计算机科学与技术

指导教师张大鹏 副教授

**2018年5月**

中图分类号：TP312 学校代码：10216

UDC：004.8 密级：公开

**工学硕士学位论文**

|  |  |  |
| --- | --- | --- |
| 硕士研究生 | ： | 刘建成 |
| 导师 | ： | 张大鹏 副教授 |
| 申请学位 | ： | 工学硕士 |
| 学科专业 | ： | 计算机技术 |
| 所 在 单 位 | ： | 信息科学与工程学院 |
| 答 辩 日 期 | ： | 2018年5月 |
| 授予学位单位 | ： | 燕山大学 |

A Dissertation in Computer Science and Technology

**RESEARCH OF PARALLELIZED COLLABORATIVE dEEPLEARNING**

by Liu Jiancheng

Supervisor: Associate Professor Zhang Dapeng

**Yanshan University**

May, 2018

燕山大学硕士学位论文原创性声明

本人郑重声明：此处所提交的硕士学位论文《协同深度推荐算法并行化研究》，是本人在导师指导下，在燕山大学攻读硕士学位期间独立进行研究工作所取得的成果。论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签字： 日期： 年 月 日

燕山大学硕士学位论文使用授权书

《协同深度推荐算法并行化研究》系本人在燕山大学攻读硕士学位期间在导师指导下完成的硕士学位论文。本论文的研究成果归燕山大学所有，本论文的研究内容不得以其它单位的名义发表。本人完全了解燕山大学关于保存、使用学位论文的规定，同意学校保留并向有关部门送交论文的复印件和电子版本，允许论文被查阅和借阅。本人授权燕山大学，可以采用影印、缩印或其它复制手段保存论文，可以公布论文的全部或部分内容。

保密□，在 年解密后适用本授权书。

本学位论文属于

不保密□。

(请在以上相应方框内打“√”)

作者签名： 日期： 年 月 日

导师签名： 日期： 年 月 日

摘 要

用户到项目评分数据的稀疏性是推荐系统质量恶化的主要因素之一。为了处理稀疏性问题，一些推荐技术考虑辅助信息来提高评分预测精度。当评分数据特别稀疏时，基于文档建模的方法通过利用文本数据(例如评论，摘要或概要)来提高准确性。然而，由于词袋模型的固有限制，它们在有效利用文档的上下文信息方面仍然存在困难，只能对文档浅层理解。本文采用一种新颖的基于上下文的深度混合推荐模型，将卷积神经网络(CNN)集成到概率矩阵分解(PMF)中的卷积矩阵分解(ConvMF)模型。ConvMF捕获文档的上下文信息，并进一步提高评分预测精度。但是，ConvMF中的卷积神经网络，在对文本进行分析时，忽略了特征的位置信息，丢失了同一特征的信息强度。

为了解决上述模型中出现的问题，本文对卷积网络进行了深入研究，提出了一种能有效分析文本数据并用于推荐系统的改进算法模型(ConvMF with Segmeng-Max Pooling，ConvMF-S)。本文内容主要分为以下几个部分：

首先，本文介绍了推荐系统目前的研究现状，对推荐系统中的各种算法进行了全面的介绍特别是个性化推荐系统中最经典的协同过滤技术。在充分了解了各种算法的原理后，深入分析了这些算法的优点和缺点。然后介绍了深度学习在推荐系统的应用。

其次，本文对卷积神经网络在自然语言处理中的应用进行了深入研究，将已有的ConvMF模型进行了改进，提出了一种新的模型ConvMF-S。

再次，本文对ConvMF-S模型的优化进行了深入研究。首先，通过预先训练词向量模型， 对CNN的嵌入层进行初始化，然后，通过。。。。。。。对模型进行进一步的优化。

最后，本文通过相关实验对改进的模型算法进行验证，并从多个方面与其他算法进行比较，对实验结果进行分析评估。

关键词：深度学习；推荐系统；自然语言处理；词向量

# Abstract

Keywords: Deep Learning; Recommendation; NLP; Word vector

目 录

[摘 要 I](#_Toc508828384)

[Abstract II](#_Toc508828385)

[第 1 章 绪 论 1](#_Toc508828386)

[1.1 课题背景及研究意义 1](#_Toc508828387)

[1.2 国内外研究现状 2](#_Toc508828388)

[1.3 现有算法存在的问题 3](#_Toc508828389)

[1.3.1 基于内容的算法 4](#_Toc508828390)

[1.3.2 协同过滤算法 4](#_Toc508828391)

[1.3.3 基于知识的算法 4](#_Toc508828392)

[1.3.4 基于人口统计学的算法 5](#_Toc508828393)

[1.3.5 混合推荐算法 5](#_Toc508828394)

[1.4 论文研究内容 5](#_Toc508828395)

[1.5 论文的组织结构 6](#_Toc508828396)

[第 2 章 卷积矩阵分解相关理论研究 7](#_Toc508828397)

[2.1 协同过滤推荐算法 7](#_Toc508828399)

[2.2 卷积神经网络 10](#_Toc508828400)

[2.3 词向量相关研究 12](#_Toc508828401)

[2.4 推荐系统评估方法 14](#_Toc508828402)

[2.5 本章小结 16](#_Toc508828403)

[第 3 章 卷积矩阵分解的模型改进 17](#_Toc508828404)

[3.1 引言 17](#_Toc508828406)

[3.2 ConvMF模型介绍 17](#_Toc508828407)

[3.2.1 ConvMF的概率模型 18](#_Toc508828408)

[3.2.2 卷积神经网络模型架构 19](#_Toc508828409)

[3.3 ConvMF模型改进 21](#_Toc508828410)

[3.4 本章小结 25](#_Toc508828411)

[第 4 章 预训练词向量及模型优化 26](#_Toc508828412)

[4.1 引言 26](#_Toc508828414)

[4.2 词向量模型 26](#_Toc508828415)

[4.3预训练词向量 28](#_Toc508828416)

[4.3.1数据预处理 28](#_Toc508828417)

[4.3.2实验分析 29](#_Toc508828418)

[4.4模型优化 30](#_Toc508828419)

[4.4.1模型复杂度分析 33](#_Toc508828420)

[4.5本章小结 33](#_Toc508828421)

[第 5 章 实验及结果分析 34](#_Toc508828422)

[5.1 实验环境及软件 34](#_Toc508828424)

[5.1.1实验环境 34](#_Toc508828425)

[5.2.2实验软件 34](#_Toc508828426)

[5.2 推荐质量评价方法 34](#_Toc508828427)

[5.3 实验数据 34](#_Toc508828428)

[5.4 实验步骤 35](#_Toc508828429)

[5.4.1数据预处理 35](#_Toc508828430)

[5.4.2实验细节 36](#_Toc508828431)

[5.5 实验分析 37](#_Toc508828432)

[5.6 本章小结 38](#_Toc508828433)

[结 论 39](#_Toc508828434)

[参考文献 41](#_Toc508828435)

[攻读硕士学位期间承担的科研任务与主要成果 46](#_Toc508828436)

[致 谢 47](#_Toc508828437)

[作者简介 48](#_Toc508828438)

1. 绪 论

## 课题背景及研究意义

近年来，互联网飞速发展，数据信息也爆发式的增长，大量的数据信息在满足人们需求的同时也造成了严重的“信息过载”(information Overload)问题。推荐系统根据用户的需求，帮助人们从信息的海洋中筛选出有用的信息，有效解决了“信息过载”的问题。目前，不同类型的推荐系统已经应用于各个领域，影响着人们生活的方方面面。比如，阿里巴巴和亚马逊将推荐系统应用于电子商务领域为用户推荐他们感兴趣的商品；Facebook、微博、腾讯等将推荐系统应用于社交网络；网易云音乐根据用户的口味给用户推荐其可能感兴趣的歌曲。

推荐算法作为推荐系统的核心，主要包括基于协调过滤的推荐、基于内容的推荐和混合推荐算法。其中，协同过滤是最经典也是应用最广泛的算法。过去几年，推荐系统在各行各业都取得了很大的成果，特别是在电子商务领域。Amazon使用推荐技术来为其用户进行商品推荐，并由此带来了35%的销售额增长。电子商务服务中的用户和项目数量的爆炸式增长增加了用户对项目评分数据的稀疏性。最终，这种稀疏性降低了传统协同过滤技术的评分预测精度。为了提高准确度，一些推荐技术提出不仅要考虑评分信息，还要考虑用户的人口统计信息、社会网络和项目描述文件等辅助信息。

2006年Hinton等人提出了深度学习的概念。很快深度学习就成为了机器学习领域中一个热门的研究方向。深度学习的概念来源于人工神经网络的研究。含多个隐层的多层感知器就是一种深度学习结构。深度学习通过组合低层特征形成更加抽象的高层来表示属性类别或特征，以发现数据的分布式特征表示。近年来，深度学习在图像处理、语音识别和自然语言处理等方面取得了革命性的进展。

将深度学习应用于推荐系统，一方面，深度学习可以学习用户和项目相关的海量数据，从样本中学习数据集的本质特征，获取用户和项目的深层次表示。另一方面，深度学习通过从多源异构数据中进行自特征学习，将不同数据映射到一个相同的隐空间，能够获得数据的同一表征，在此基础上融合传统推荐方法进行推荐，能够有效利用多源异构数据，缓解数据稀疏、冷启动和预测精度下降的问题。

以往的工作证明，深度学习在推荐系统中学习到的特征具有代表性，且对于解决推荐系统中数据稀疏问题具有有效性，可以提高学习性能。实际应用中，推荐系统的数据是复杂且稀疏的，基于矩阵分解的协同过滤方法很难学习到有效的特征信息。目前基于深度学习的推荐算法还比较少，但是初步的论文成果已经证明它们在实际数据应用中的有效性。因此，研究基于深度学习的推荐算法将是一个具有现实意义的问题。

## 国内外研究现状

目前，对推荐算法的研究越来越广泛。根据推荐算法所使用信息和知识来源的不同，推荐算法被分成很多类别。主流的推荐算法被分成下面五个类别：基于内容的推荐算法，基于协同过滤(Collaborative filtering, 简称CF)推荐算法，基于知识的推荐算法，基于人口统计学的推荐算法和混合推荐算法。其中协同过滤是当下最流行的推荐方法之一，该技术在不同的方向上都引起了研究者的广泛关注。

协同过滤推荐算法[1],[2]的思想是首先寻找与目标用户有相似兴趣的用户群体，然后再根据这些用户过去喜欢过的商品对目标用户进行推荐，也就是说用户获得的推荐是基于和他有着相似兴趣爱好的人群。协同过滤的原理是，根据用户对商品或者信息的喜好，发现商品或者内容本身的相关性，或者发现用户的相关性，然后基于这些关联性进行推荐。基于协同过滤的推荐算法又可以分为三个子类：基于用户的推荐算法(User-based CF Recommendation)，基于商品的推荐算法(Item-based CF Recommendation)和基于模型的推荐算法(Model-based CF Recommendation)。

基于用户的协同过滤推荐算法[3]的基本原理是，根据用户对物品或信息的偏好，发现与当前用户口味和兴趣相似的“邻居”用户群(一般采用k-近邻算法)，然后，基于这k个近邻的历史偏好信息，为当前用户进行推荐。基于商品的协同过滤推荐算法[4][5]的基本原理与此类似，它使用所有用户对商品或信息的偏好，发现商品和商品之间的相似度，然后根据用户的历史偏好信息，将类似的商品推荐给用户。基于模型的协同过滤推荐算法[6]是利用基于用户样本的用户喜好信息，通过模型训练(如贝叶斯模型，聚类模型或依赖网络模型)来学习用户的兴趣偏好，由此来进行相似商品的推荐。

在各种不同的协同过滤技术中，矩阵分解(Matrix factorization，简称MF)是基于模型的协同过滤算法中最热门的一种。矩阵分解方法通过将用户-项目评分矩阵分解成一个与用户相关的低秩矩阵(用户特征矩阵)和一个与商品相关的低秩矩阵(商品特征矩阵)，并使用分解后的两个低秩矩阵的乘积来拟合原评分矩阵，来对没有评分的用户-项目进行预测。一些算法改进工作已经被应用到该方法上，比如带权重的矩阵分解方法[7]、基于非负矩阵的矩阵分解方法[8]，和基于矩阵局部性的矩阵分解方法[9]等，这些方法都展示了其在推荐系统上的效果。

深度学习是机器学习领域中近年来关注度非常高的研究领域，是由多个层次或多个非线性信息处理模块组成的模型，是一种监督和非监督学习方法在深度模型的更高抽象层次里的特征表示。近些年来，深度学习技术已经被证明在自然语言处理、计算机视觉和语音识别中能够学习到很好的数据特征表示。深度学习算法中应用最为广泛的是卷积神经网络(CNN)和循环神经网络(RNN)。其中卷积神经网络(CNN)是一种深度判别模型。其基本结构是，每个模块都由一个卷积层和一个采样层构成。这些模块通常相互堆叠，一个模块在另一个模块的上方，这样形成了一个深度模型。卷积层的权值共享和采样层对卷积层输出进行子采样，都有效减少了下一层的数据率。卷积层共享的权值和采样层的池化过程，使得CNN有一些抗扭曲的能力，即有一定的不变性。目前，CNN已经在计算机视觉和图像识别[10]-[11][12]应用中取得非常好的成果。近年来，CNN在自然语言处理方向也取得了很好地发展[13]-[14][15][16]将深度学习技术应用于推荐系统中进行特征学习，能够学习到更具有代表性的用户商品特征。目前，已经有一些工作致力于将深度学习应用到推荐系统中。Salakhutdinov等人提出了基于玻尔兹曼机的协同过滤算法[17]，得到了比Netflix系统更好的推荐效果。最近，研究者们已经提出了基于文档建模(如LDA主题型和堆栈去燥自动编码器(SDAE))的方法来利用项目描述文档(如评论，摘要或概要)。具体来说，Wang等人提出了协作主题回归模型(CTR)，其结合了主题建模(LDA)和协同过滤的概率方法[18]。后来又有人提出了不同的CTR，将LDA整合到协同过滤中，以不同的集成方法分析项目描述文档[19],[20]。最近，Wang等人提出将SDAE整合到概率矩阵分解(PMF)中的协同深度学习(Collaborative deep learning，简称CDL)[21]，从而在评分预测精度方面产生更准确的潜在模型[22]，但是基于深度学习的推荐算法的研究仍然非常少，不够系统。

## 现有算法存在的问题

目前，许多科学家已经对推荐算法进行了广泛的研究，不同的推荐算法也被应用到了各个领域。其中，主流的推荐算法主要有下面五个类别：基于内容的推荐算法，协同过滤推荐算法，基于知识的推荐算法，基于人口统计学的推荐算法和混合推荐算法。

### 基于内容的推荐算法

基于内容的推荐算法[23]旨在将和用户过去喜欢过的商品相似的商品推荐给用户。根据用户画像，能够找到最适合用户的推荐[24]。这里建立用户画像的方法，可以使用基于记忆(启发式)的方法，例如信息检索中的词频-逆文档频率(TF-IDF)；也可以使用基于模型的方法，例如使用决策树、贝叶斯分类器和聚类等方法得到用户的兴趣表示。基于内容的推荐算法可以对用户的兴趣很好的建模，并通过对物品属性维度的增加，获得更好的推荐精度。但是如果用户没有对足够的商品进行评分，那他将无法得到商品的推荐，即无法适应冷启动问题。另外，当用户的兴趣模型被建立起来后，系统很难再适应的改变其兴趣走向。

### 协同过滤推荐算法

协同过滤推荐算法[1],[2]给用户推荐和他兴趣相似的用户过去喜欢的商品，也就是说用户获得的推荐是基于和他有着相似爱好的人群。基于协同过滤的推荐算法不需要对商品或者用户进行严格的建模，而且也不需求商品的描述是机器可理解的，因此这种方法是与领域无关的，该方法很好的支持用户发现潜在的兴趣爱好。然而，由于该方法的推荐效果依赖于用户历史偏好的数据的多少和准确性。大部分情况下，用户的历史偏好使用稀疏矩阵存储，稀疏矩阵计算上的问题会导致少部分人的错误偏好会对推荐结果的准确度有很大的影响。

### 基于知识的推荐算法

基于知识的推荐算法[25]需要对商品的属性特征和用户画像(用户的兴趣特征)有一个基本的理解，该方法需要对商品和用户进行全面建模。基于知识的推荐算法利用对用户和商品的知识理解，来推导出最符合用户需求的商品。为了学习到商品和用户的知识，它需要能够获取到足够的知识且能够持续积累。该方法不会存在冷启动的问题，因为任何一个新的用户或商品进入系统时，已经有了对该用户或商品的清晰认识。它不需要样本用户，因为推荐结果的计算不需要依靠大量的用户数据。然而该方法的不足之处在于，很难获取到足够的知识来构建用户和商品知识库。

### 基于人口统计学的推荐算法

基于人口统计学的推荐算法[26]是一种最易于实现的推荐算法，它只是简单的依据系统用户的基本信息发现用户的相关关系，然后将相似用户喜爱的其他物品推荐给当前用户。而这里人口统计学的基本信息包括年龄、性别、社会阶层、教育程度和地理位置等，根据这些信息可以得到用户画像。根据统计的用户画像，特定的匹配商品能够被推荐给用户。该方法通过学习用户的属性，将用户进行分类，从而给出相应的推荐。基于人口统计学的方法也不需要用到用户对商品的评分，它不存在冷启动问题。然而，收集用户画像信息在一定程度上侵犯了用户的隐私，而且用户画像信息的收集可能有很大的噪音。

### 混合推荐算法

混合推荐算法，单独使用上述推荐算法中的一种，都会存在一些优点和缺陷。于是往往将多个方法混合在一起，从而达到更好的推荐效果[27]。关于如何组合各种推荐算法，目前有几种比较流行的组合机制。比如，加权组合(Weighted Hybridization)：用线性公式将几种不同的推荐算法按照一定的权重组合起来，具体权重值的设置要在测试集上反复试验，以达到最好的推荐效果。切换混合(Switching Hybridization)：由于不同的情况(数据量，用户和商品等)，推荐算法的性能可能有很大的不同，该方式允许在不同的情况下，选择最为合适的推荐算法。分区混合(Mixed Hybridization)：采用多种推荐算法，并将不同的推荐结果分不同的区域显示给用户，该方法被很多电子商务网站采用，用户可以得到很全面的推荐，也更容易找到他们想要的东西。基于深度学习的推荐算法，也算是一种混合推荐。

## 论文研究内容

本文在对卷积矩阵分解算法进行深入研究后进行改进，提出了ConvMF-S模型，通过利用文本信息，可以有效缓解数据稀疏性问题。在改进的模型基础上进一步优化模型的运行效率，并通过嵌入Word2vec预训练的词向量提高推荐系统的准确性。

为了证明其有效性，本文对通过多个对比实验，验证了改进的ConvMF-S模型的优越性。

本文贡献可简要概括为：

1. 对卷积矩阵分解进行改进，基于ConvMF算法提出了ConvMF-S模型。
2. 对模型进行优化，有效提高了模型收敛速度。
3. 利用Word2vec算法以IMDB为数据集预训练词向量，并通过将其嵌入模型中，提高推荐质量。

## 论文的组织结构

本文对基于上下文的深度混合推荐系统进行了研究，主要内容有：

第1章 主要介绍了课题的研究背景及意义，国内外研究现状，以及传统推荐算法中存在的问题。

第2章 介绍了与研究课题相关的算法理论，包括协同过滤推荐算法、概率矩阵分解(PMF)、卷积神经网络(CNN)、词向量(Word vector)以及推荐系统的评估方法；

第3章 对卷积矩阵分解做了详细介绍，针对其中的不足，提出了ConvMF-S模型，通过在池化层中保留更多的特征信息，提高生成的项目特征矩阵的质量，从而提高推荐系统的准确性。

第4章 对改进的算法模型进行仔细研究后，改进了模型中的部分函数，使其能更快的收敛，提高了系统运行效率。并且利用Word2vec词向量模型以IMDB的电影评论为数据集训练词向量将其嵌入算法模型中。

第5章 在前文理论分析的基础上，构建模型代码进行算法实现，并与已有的多种推荐算法进行比较，然后对实验结果进行仔细分析，证明了改进算法的优越性。

最后，对本文的研究课题和研究成果做出总结，并指明了未来需要进一步研究的方向以及在研究中可能面临的困难。

1. 卷积矩阵分解相关理论研究
2. 1. 协同过滤推荐算法

推荐算法具有非常多的应用场景和商业价值，因此推荐算法值得好好研究。推荐算法种类很多，主要有基于协同过滤的推荐，基于人口统计学的推荐和基于内容的推荐等等。其中，协同过滤是当下最流行的推荐方法之一，该技术在不同的方向上都引起了研究者的广泛关注。协同过滤算法的本质就是利用其它用户对商品的历史行为来协助对目标用户的推荐。一般来说，协同过滤推荐分为三种类型。第一种是基于用户(user-based)的协同过滤，第二种是基于项目(item-based)的协同过滤，第三种是基于模型(model-based)的协同过滤。

基于用户(user-based)的协同过滤主要考虑的是用户和用户之间的相似度，根据所有用户对物品或者信息的偏好，发现与当前用户口味和兴趣相似的“邻居”用户群(一般采用k-近邻算法)。然后，基于这k个近邻用户的历史偏好信息，找到评分最高的若干个物品为当前用户进行推荐。而基于项目(item-based)的协同过滤和基于用户的协同过滤类似，只不过这时要寻找的是物品和物品之间的相似度，只要找到目标用户对某些物品的评分，那么我们就可以对相似度高的类似物品进行预测，将评分最高的若干个相似物品推荐给用户。比如你在网上买了一本机器学习的书，网站上马上就会推荐一堆与机器学习、大数据相关的书给你，这里就明显用到了基于项目的协同过滤的思想。基于模型的协同过滤算法[28]是基于用户样本的用户喜好信息，通过模型训练(如贝叶斯模型，聚类模型或依赖网络模型)来学习用户的兴趣爱好，由此来进行相似商品的推荐。

基于用户的协同过滤需要在线寻找用户和用户之间的相似度关系，计算复杂度要比基于项目的协同过滤高，但是可以帮助用户找到新类别的有惊喜的物品。而基于项目的协同过滤，考虑到物品的相似性一段时间不会改变，因此可以很容易的离线计算，准确度一般也可以接受，但是在推荐的多样方面，很难带给用户惊喜。一般对于小型的推荐系统来说，基于项目的协同过滤是主流。但是如果是大型的推荐系统来说，使用基于用户的协同过滤效果更好。

基于内容的协同过滤的推荐算法不需要对商品或者用户进行严格的建模，而且也不要求商品的描述是机器可理解的，因此这种方法是与领域无关的，该方法很好的支持用户发现潜在的兴趣爱好。然而，由于该方法的推荐效果依赖于用户历史偏好数据的多少和准确性。大部分情况下，用户的历史偏好使用稀疏矩阵存储，稀疏矩阵计算上的问题会导致少部分人的错误偏好会对推荐结果的准确度有很大的影响。

在不同的协同过滤技术中，基于模型(model based)的协同过滤是目前最主流的协同过滤类型了，其推荐质量也被实验证明是协同过滤技术中最好的。矩阵分解方法通过将用户-商品评分矩阵分解成一个与用户相关的低秩矩阵(用户特征矩阵)和一个与商品相关的低秩矩阵(商品特征矩阵)，并使用分解后的两个低秩矩阵的乘积来拟合原评分矩阵，同时对未观察到的用户-商品对进行评分预测。基于矩阵分解的方法，最重要的因素是所学习的用户特征因子矩阵和商品特征因子矩阵，好的用户特征和商品特征能带来更好的预测结果。

随着Netflix Prize推荐比赛的成功举办，近年来隐语义模型(Latent Factor Model, LFM)受到越来越多的关注。隐语义模型最早在文本挖掘领域被提出，用于寻找文本的隐含语义。矩阵分解技术是实现隐语义模型使用最为广泛的一种方法，其思想也正是来源于此，著名的推荐领域大神Yehuda Koren更是凭借矩阵分解模型勇夺Netflix Prize推荐比赛冠军，实验结果表明，在个性化推荐中使用矩阵分解模型要明显优于传统的基于邻域的协同过滤(又称基于记忆的协同过滤)方法，如UserCF、ItemCF等，这也使得矩阵分解成为了目前个性化推荐研究领域中的主流模型。

矩阵分解的核心思想是认为用户的兴趣只受少数几个因素的影响，因此将稀疏且高维的User-Item评分矩阵分解为两个低维矩阵，即通过User和Item的评分信息学习到用户特征矩阵P和物品特征矩阵Q，再利用重构的低维矩阵预测用户对产品的评分。

矩阵分解方法将高维的User-Item评分矩阵映射为两个低维的用户和物品矩阵，改善了数据稀疏性问题。使用矩阵分解具有以下优点：比较容易编程实现，随机梯度下降方法依次迭代即可训练出模型。比较低的时间和空间复杂度，高维矩阵映射为两个低维矩阵节省了存储空间，虽然训练过程比较费时，但是可以离线完成；评分预测一般在线计算，直接使用离线训练得到的参数，可以实时推荐。预测的精度比较高，预测准确率要高于基于领域的协同过滤以及基于内容的推荐等方法。非常好的扩展性，很方便在用户特征向量和物品特征向量中添加其它因素。

推荐系统中常用的矩阵分解有非负矩阵分解(NMF)、奇异值分解(SVD)、概率矩阵分解(PMF)等。

Lee等人最早提出了非负矩阵分解(Non-negative Matrix Factorization，NMF)算法[29]，NMF是在矩阵中所有元素均为非负数约束条件之下的矩阵分解方法。该算法是多变量分析和线性代数的算法。非负矩阵分解处理大规模数据更快更便捷，实现简便性、分解形式和分解结果上的可解释性，占用存储空间少。然而，NMF中只用了一层来表示隐变量，无法处理复杂学习问题。NMF只约束了输出的两个矩阵的非负性(这是唯一先验，只要求满足这个)，而没有考虑到对于该先验，矩阵内部元素间的相关性。

奇异值分解就是将用户对物品的评分矩阵进行SVD分解，并通过选择部分较大的一些奇异值来同时进行降维。这种方法简单直接，但有一个很大的问题就是SVD要求评分矩阵是稠密的，也就是说矩阵的所有位置不能有空白。有空白的时候评分矩阵没有办法直接去SVD分解。然而，现实生活中的User-Item矩阵极大(User数量极大，Item数量极大)，而用户的兴趣和消费能力有限，对单个用户来说消费的物品，产生评分记录的物品是极少的。这样造成了User-Item矩阵含有大量的空值，数据极为稀疏。传统的SVD采用的方法是对评分矩阵中的缺失值进行简单的补全，比如用全局平均值或者使用用户对物品评分的平局值进行补全，得到补全后的评分矩阵，然后再进行SVD分解并降维。虽然有了补全策略，但是传统的SVD在推荐算法上还是很难使用的。因为用户数和物品数量都成千上万非常大，这么大的矩阵SVD分解非常耗时。

如上所述，User-Item评分矩阵维度较高且极为稀疏，传统的奇异值分解方法只能对稠密矩阵进行分解，即不允许所分解矩阵有空值。因而，若采用奇异值分解，需要首先填充User-Item评分矩阵。这样就造成了两个问题，其一，填充数据大大增加了数据量，增加了算法复杂度；其二，简单粗暴的数据填充很容易造成数据失真。这些问题导致了传统的SVD矩阵分解并不理想。

许多现有的协同过滤方法既不能处理非常大的数据集，也不能轻松处理评分很少的用户数据。在本文中，我们采用概率矩阵分解(PMF)模型，该模型与观测值的数量呈线性关系，更重要的是，在大型，稀疏且不平衡的数据集上也有良好的表现。

概率矩阵分解(PMF)假设用户U和项目V的特征矩阵均服从高斯分布，通过评分矩阵已知值来求得U和V的特征矩阵，然后用特征矩阵去预测评分矩阵中的未知值。

用户U的特征矩阵满足均值为0，方差为的高斯分布，有如下公式：



同理：项目V的特征矩阵满足如下公式：



其中，表示变量x满足均值为u，方差为的高斯分布。

假设已经观测到的评分数据条件概率也服从高斯先验分布，即：



其中，是指示函数，当用户u对物品i有评分时，其值为1，否则为0。

利用贝叶斯推导，可得用户和物品的隐式特征的后验概率为：



将上式展开并取对数，求极大值就可以得到在已知参数，，和现有的评分矩阵的前提下得到最有可能的U和V的隐式特征矩阵。

最后，PMF的概率图表示如下所示：



1. PMF矩阵分解概率模型图
   1. 卷积神经网络

提到卷积神经网络(CNN)，首先想到的是它在计算机视觉(computer vision)上的应用。CNN在图像分类中取得了重大的突破，从Facebook的图像标注到自动驾驶，CNN成为计算机视觉系统的核心算法。

虽然CNN最初是为计算机视觉而开发设计的，但是CNN的关键思想已经被积极应用于信息检索和自然语言处理(NLP)，如搜索查询检索，句子建模与分类和其他传统的NLP任务。虽然针对NLP任务需要对CNN架构进行大量修改，但最终都有助于提高各种NLP任务的性能。

传统的词袋模型或者连续词袋模型都可以通过构建一个全连接的神经网络对句子进行情感标签的分类，但是这样存在一个问题，通过激活函数可以让某些结点激活，但是由于在这样网络构建里，句子中词语的顺序被忽略，我们无法捕获由两个连续词所构成的关键特征词的含义。在语言模型里n-gram模型将两个连续的词作为一个整体纳入到模型中，可以用来解决上面提出的问题。但是如果我们使用多元模型，实际训练时的参数是一个非常大的问题，因为假设有20000个词，加入bi-gram实际上就要有400000000个词，这样的参数训练显然是爆炸的。另外一点，相似的词语在这样的模型中不能共享参数权重等，这样就会导致相似词无法获得交互信息。

现有的集成模型并不能完全捕获文档信息，因为它们使用的词袋模型忽略诸如周围单词和单词顺序的文档的上下文信息。 例如，假设在文件中给出了以下两个句子：“人们相信这个人”，“人们最终背叛了他的信任”，由于LDA和SDAE认为该文件是一个不可忽略的单词的集合，所以他们无法区分每一个事件 的“信任”。 确切地说，虽然每个“信任”的发生似乎都具有几乎相同的含义，但是这些词之间存在微妙的语法差异——一个动词和一个名词。 文档中的这种微妙的差异对于更深入地理解文档也是一个非常重要的因素，并且进一步的加深这种理解有助于提高评分预测精度。

为了解决上述问题，我们采用卷积神经网络(CNN)来处理NLP任务，它是最先进的机器学习方法，在诸如计算机视觉，自然语言处理(NLP)和信息检索等各个领域表现出了高性能。

卷积神经网络(Convolutional Neural Network，简称 CNN)是具有以下组件的前馈神经网络的变体：1)用于产生局部特征的卷积层，2)通过仅选择几个典型的局部特征（即，通过激活函数获得具有最高分数的特征）来表示数据，池化层通过获取卷积层的输出可以获得更简洁的特征。

在图像中卷积核通常是对图像的一小块区域进行计算，而在文本中，将一句话所构成的词向量作为输入。每一行代表一个词的词向量，所以在处理文本时，卷积核通常覆盖上下几行的词，所以此时卷积核的宽度与输入的宽度相同，通过这种方式，我们就能够捕捉到多个连续词之间的特征，并且能够在同一类特征计算中共享权重。卷积神经网络的一个重要概念就是池化层，一般是在卷积层之后。池化层对输入做降采样。池化的过程实际上也是对卷积层分区域求最大值或者对每个卷积层求最大值。首先，池化可以输出一个固定大小的矩阵，这对于自然语言处理当中输入句子的长度不一有非常大的作用。例如，如果你用了200个卷积核，并对每个输出使用最大池化，那么无论卷积核的尺寸是多大，也无论输入数据的维度或者单词个数如何变化，你都将得到一个200维的输出。这让你可以应对不同长度的句子和不同大小的卷积核，总是能得到一个相同维度的输出结果，用作最后的分类。另外池化层在降低数据维度的同时还能够保留显著的特征。每一种卷积核都是用来检测一种特定的特征。在句子分类中，每一种卷积核可以用来检测某一种含义的词组，如果具有某种含义的词语出现了，该卷积核的输出值就会非常大，通过池化过程就能够尽可能地将该信息保留下来。

然而，CNN尚未被积极应用于推荐系统领域。据我们所知，van den Oord等人首先将CNN 应用于音乐推荐[30]，通过声学分析观点利用CNN分析了歌曲，并提出了一种基于声学CNN获得的项目潜在模型来预测评分。然而，他们的CNN模型，专为声音信号处理而设计，不适合处理文档。文档和声学信号对周围特征的质量有固有的差异。声学信号中一定时间的信号固有地类似于其周围的信号，即具有轻微时间差的信号，而文档中某个位置处的一个字与周围的字可能有很大的语义差异。周围特征之间的这种相似度的差异影响局部特征的质量，最终需要不同的CNN架构。此外，该模型还没有充分反映协同信息。具体来说，项目潜在模型主要由CNN进行音频信号分析的结果而不是协同信息决定的。因此，总体推荐的性能甚至没有达到加权矩阵分解(WMF)的效果，WMF是处理隐式反馈数据集的基于MF的协同过滤算法之一。

* 1. 词向量相关研究

深度学习掀开了机器学习的新篇章，目前深度学习在图像和自然语言处理等领域已经产生了突破性的研究进展。深度学习一直被人们推崇为一种类似于人脑结构的人工智能算法。就目前而言，Deep Learning在NLP领域中的研究已经将高深莫测的人类语言撕开了一层神秘面纱。其中最有趣也最基本的，就是“词向量”了。

自然语言处理的问题要转化为机器学习的问题，第一步就是要找到能够把这些字符转换为数字的方法。NLP中最直观，也是目前为止最常用的词表示方法是One-hot Representation，这种方法把每个词表示为一个很长的向量。这个向量的维度是词表的大小，其中绝大多数元素为0，只有一个维度的值为1，这个值为1的维度就代表了当前的词。

举个例子：

“话筒”表示为[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 …]

“麦克”表示为[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 …]

这种One-hot Representation如果用稀疏方法存储，会非常的简洁，也就是直接给每个词分配一个数字ID。比如在刚才的例子中，话筒标记为3，麦克标记为8(假设从0开始)。如果要编程实现的话，用Hash表给每个单词分配一个编号就可以了。如此简洁的表示方法再配合上最大熵、SVM等算法就可以完成NLP领域的各种主流任务。

然而，这种表示方法也存在一个重要的问题就是“词汇鸿沟”现象：任意两个词之间都是孤立的。单从两个词向量中看不出两个词是否有关系，就像话筒和麦克这样的同义词也不能幸免。此外，这种表示方法还容易发生维数灾难，尤其是Deep Learning相关的一些应用中。

针对上述问题Hinton最早提出了Distributed representation词向量表示方法，它是一种低维实数向量，既能表示词本身又可以考虑语义距离。它最大的贡献就是让语义上相关或者相似的词，在距离上更接近了。向量的距离可以用传统的欧式距离来衡量，也可以用余弦夹角来衡量。用这种方式表示的向量，“麦克”和“话筒”的距离会远远小于“麦克”和“天气”的距离。在理想情况下，“麦克”和“话筒”的表示是完全一样的。

将word映射到一个新的空间中，并以多维的连续实数向量进行表示叫做“Word Representation”或“Word Embedding”。自从21世纪以来，人们逐渐从原始的词向量表示法过渡到现在的低维空间中的密集表示。用稀疏表示法在解决实际问题时经常会遇到维数灾难，并且语义信息无法表示，无法揭示word之间的潜在联系。而采用低维空间表示法，不但解决了维数灾难问题，并且挖掘了word之间的关系属性，从而提高了语义向量上的准确度

矩阵分解模型(LSA)，采用线性代数中的奇异值分解方法，选取前几个比较大的奇异值所对应的特征向量将原矩阵映射到低维空间中，从而达到词矢量的目的。潜在语义分析概率模型(PLSA)，从概率学的角度重新审视了矩阵分解模型，并得到一个从统计，概率角度上推导出来的和LSA相当的词矢量模型。文档生成模型(LDA)，按照文档生成的过程，使用贝叶斯估计统计学方法，将文档用多个主题来表示。LDA不只解决了同义词的问题，还解决了一词多义的问题。目前训练LDA模型的方法有原始论文中的基于EM和差分贝叶斯方法以及后来出现的Gibbs Samplings采样算法。Word2Vector模型，是最近几年刚刚火起来的算法，通过神经网络机器学习算法来训练N-gram语言模型，并在训练过程中求出word所对应的vector的方法。与潜在语义分析(Latent Semantic Index, LSI)、潜在狄立克雷分配(Latent Dirichlet Allocation，LDA)的经典过程相比，Word2vec利用了词的上下文，语义信息更加地丰富。

Word2vec是将词表征为实数值向量的一种高效的算法模型，其利用深度学习思想，通过训练，把对文本内容的处理简化为k维向量空间中的向量运算，而向量空间上的相似度可以用来表示文本语义上的相似。

* 1. 推荐系统评估方法

面对众多的推荐算法，如何有效地评价推荐系统的优劣成为了颇具挑战的问题[31][32][33]。目前已有的推荐系统的评价指标数不胜数，本文只对最常用的几种进行介绍：

### 2.4.1预测评分准确度的评估方法

顾明思议，这类方法评价的是推荐算法所预测的用户评分和用户的真实评分的贴近程度。

1. 平均绝对值误差

平均绝对误差(mean absolute error，MAE)[34],[35]是预测评分准确度方法中最经典的一种，其定义为：

其中，表示用户对商品的真实评分，表示用户对商品的预测评分，代表的是测试集。

MAE求的是单个预测评分与真实评分的差的绝对值求和然后再求平均，来表示平均绝对误差。MAE所得的值越小，说明预测评分与用户的实际评分差值越小，推荐系统的效果也越好。

此外，平均平方误差(mean squared error，MSE)、均方根误差(root mean squared error, RMSE)以及标准平均绝对误差(normalized mean absolute error，NMAE)[36]都是和平均绝对误差类似的方法。它们的公式定义分别如下所示：

1. 平均平方误差



1. 均方根误差



1. 标准平均绝对误差



上式中的和分别代表用户评分区间的最大值和最小值。由于MSE和RMSE对预测值和真实值的绝对误差求了平方，所以它们对比较大的绝对误差有更重的处罚。NMAE对结果在评论区间上做了归一化，可以用来评价同一推荐算法在不同数据集上的表现。

### 分类准确度的评估方法

与预测评分准确度不同，分类准确度不会要求推荐系统准确的预测用户的评分，它关注的是推荐系统能否正确预测用户喜欢或不喜欢某个商品。因此，当用户只有二元选择时，即要么喜欢，要么不喜欢，用分类准确度进行评价较为合适。目前为止，最常用的分类准确度指标有准确率(precision)、召回率(recall)、F1指标。

1. 准确率

准确率定义为系统的推荐列表中用户喜欢的产品和所有被推荐产品的比率。准确率表示用户对一个被推荐产品感兴趣的可能性。假设R(u)表示向用户u推荐的N个物品的集合，T(u)表示数据集合中用户u真实评价过的物品的集合，则准确率的公式表示如下：



1. 召回率

召回率定义为推荐列表中用户喜欢的产品与系统中用户喜欢的所有产品的比率，召回率表示一个用户喜欢的产品被推荐的概率。



1. F1指标

利用准确率和召回率对推荐系统进行评价的最大问题在于它们必须要一起使用才能全面评价算法的好坏。

为了同时考察准确率和召回率，Pazzan iM[37],[38]等把二者综合考虑提出了F1指标。F1指标定义为：



其中，P为准确率，R为召回率。

* 1. 本章小结

本章对本文算法模型中的基本模块进行了介绍。具体包括了协同过滤推荐算法、概率矩阵分解(PMF)模型的原理及相关公式、卷积神经网络的原理以及其在自然语言处理中的应用、词向量的相关研究以及推荐系统的评价指标等。

1. 卷积矩阵分解的模型改进
2. 1. 引言

为了解决评分数据的稀疏性以及提高推荐系统的推荐质量，人们提出了文档建模的方法通过利用文本数据(例如评论，摘要或概要)来提高准确性。由于词袋模型的固有限制，不能充分利用文档的上下文信息，对文档的理解不深刻。Donghyun Kim[39]等人提出了一种新颖的文档上下文感知推荐系统，将卷积神经网络(CNN)集成到概率矩阵分解(PMF)中的卷积矩阵分解(ConvMF)。CNN通过建模组件有效地捕获图像或文档的局部特征，这些模型组件包括局部感知野，共享权重和下采样。CNN的使用可以有助于更深入的了解文档信息，产生比LDA和SDAE更好的潜在模型，特别是对于由于缺乏评分而诉诸于其描述文档的项目。但是该模型在CNN模块中的Max Pooling层取的是整个区域的最大值作为特征，提取的信息过于单一。本文在该模型的基础上，提出了一个新的Segment-Max Pooling层，对每个卷积后的特征层分k段选取特征，这样既提取到了多个特征，同时又保留了他们的相对位置信息。

* 1. ConvMF算法模型

现有的CNN不适合推荐任务，因为卷积网络的目标与推荐目标不同。具体来说，常规CNN主要解决分类任务，如预测单词，短语或文档的标签。相反，推荐的目标被认为是一个回归任务，旨在准确地近似目标对项目的评分。因此，现有的CNN不能直接适用于现有的推荐任务。

ConvMF正是将CNN无缝集成到概率矩阵分解(PMF)中，而PMF通常用于推荐任务。因此，综合模式遵循推荐目标，最终有效地利用协作信息和上下文信息。因此，即使评分数据非常稀疏，ConvMF也能准确地预测未知的评分。

### ConvMF的概率模型



1. ConvMF概率模型：左边为PMF部分；右边为CNN部分

图3-1显示了ConvMF的概率模型的概述，它将CNN集成到PMF中。假设我们有个用户和个项目，并且观察到的评分由的矩阵表示。然后，我们的目标是找到其乘积()能够重建评级矩阵的用户和项目潜在模型(和)。从概率观点上看，评分的条件分布如下式所示：



其中是具有平均值 和方差的高斯正态分布的概率密度函数，是在第二章中提到的指标函数。

作为用户潜在模型的生成模型，将传统的先验，方差为的零均值球面高斯置于用户潜在模型上。



然而，与常规PMF中项目潜在模型的概率模型不同，本文中假设项目潜在模型由三个变量生成：1) CNN中的内部权重W，2) 表示项目j的文档，以及3) 变量作为高斯噪声，使我们能够进一步优化评级的项目潜在模型。因此，最终项目潜在模型通过以下等式获得。





对于W中的每个权重，我们使用先前最常用的零均值球面高斯先验：



项目的隐模型的条件分布如下：



其中，是项目的描述文件集合。使用从CNN模型获得的文档潜在矢量作为高斯分布的平均值，并将该项目的高斯噪声用作高斯分布的方差，作为CNN和PMF之间的桥梁的重要作用，有助于完全分析描述文件和评分。

### 卷积神经网络模型架构

深度学习近年来在图像处理和NLP任务上都取得了令人瞩目的成绩。通常，图像处理的任务是借助CNN来完成的，其特有的卷积、池化结构能够提出图像中各种不同程度的纹理、结构，并最终结合全连接网络实现信息的汇总和输出。

在短文本分析任务中，由于句子句长长度有限、结构紧凑、能够独立表达意思，使得CNN在处理这一类问题上称为可能。Yoon Kim将卷积神经网络用于句子分类[40]该卷积神经网络的结构比较简单，使用长度不同的filter对文本矩阵进行卷积，filter的宽度等于词向量的长度，然后使用max-pooling对每一个filter提取的向量进行操作，最后每个filter对应一个数字，把这些filter拼接起来，就得到了一个表征该句子的向量。最后的预测都是基于该句子的。Kalchbrenner等人提出了用于句子建模的卷积神经网络[41]，Chen Y等人提出的动态多池化卷积神经网络进行事件提取[42]，将word2vec横向组合，以发现句子层级的特征。Barzilay等人提出了一种为文本建模的非线性，非连续卷积的CNN，对卷积层进行了非线性化改造[43]。Dong L等人对同一输入做多重CNN分类，将结果进行组合[44]。Kalchbrenner提出了扩张卷积神经网络(Dilated Convolution)[45]应用于机器翻译领域。Dilated Convolution主要解决了池化层的池化会损失很多信息(无论该信息是有用还是无用)的问题。在该模型中，句子建模时输入是以句子的字符级别开始的，之后随着卷积核所能覆盖的范围扩展，不断地区交互信息，同时还能保证原始的输入信息不被丢失。文献[46]

本文中CNN架构的目标是从项目文档中生成文档潜在向量，用于组合具有变量的项目潜在模型。图3-2显示了该模型的CNN架构，由四层组成：1)嵌入层，2)卷积层，3)池化层，4)输出层。

1. 卷积神经网络模型架构图



1. Embedding Layer

嵌入层将原始文档转换成文档的密集数字矩阵，以方便下一层的卷积层做卷积处理。详细地说，就是将该文件作为一个l个单词的序列，我们通过连接文档中单词的向量来将文档表示为矩阵。单词向量随机初始化，然后，通过优化过程进一步训练单词向量。最后，文档矩阵可以写成如下形式：



其中，l是文档的长度，p是单词的维度。

1. Convolution Layer

卷积层是用来提取上下文的特征的。正如我们在第二章中讨论的，文档本质上不同于信号处理或计算机视觉的上下文信息。因此，本文中使用[43],[44]中的卷积架构来正确分析文档。上下文特征由第个共享权重提取，其中窗口大小确定周围单词的数量：



其中，表示的是卷积运算符，是的偏置，是非线性激活函数。在诸如sigmoid，tanh和整流线性单元(ReLU)的非线性激活函数中，我们使用ReLU来避免梯度消失的问题，但是其导致优化收敛特别缓慢，并且很容易导致较差的局部最小值，不能达到全局最优。然后，具有的文档的语境特征向量由等式构成：



然而，因为卷积神经网络的特征，一个共享权重只能捕获到一种类型的上下文特征。 因此，本文中使用了多个共享权重来捕获多种类型的上下文特征，这使得我们能够生成与的数目一样多的上下文特征向量。

1. Pooling Layer

池化层用来从卷积层提取出具有代表性的特征，并且还可以通过构建固定长度的特征向量来处理可变长度的文档。 在卷积层之后，文档被表示为具有个上下文特征的向量，其中每个上下文特征向量具有可变长度(即，的向量长度)。CNN在NLP的应用中一般都是直接使用的Max pooling，通过从每个上下文特征向量中提取最大上下文特征，将每个文档表示为具有固定长度的向量，如下所示。



其中，是通过第个共享权重提取的长度为的上下文特征向量。

1. Output Layer

一般情况下，在输出层，从上一层获得的高级特征应该转换为特定的任务。因此，将用户和项目潜在模型的k维空间上的投影用于我们的推荐任务，最终通过使用传统的非线性投影产生文档潜向矢量：



其中，，是投影矩阵，，是，的偏置向量。

最终，CNN架构以原始文档作为函数的输入，经过卷积池化处理后在输出层返回每个文档的潜在向量，公式如下：



其中， 表示所有权重和偏置变量(为了防止公式杂乱)，表示项目的原始文档，表示项目的文档潜在向量。

* 1. ConvMF模型改进

CNN是目前自然语言处理中和RNN并驾齐驱的一种最常见的深度学习模型。图3-2展示了在NLP任务中使用CNN模型的典型网络结构。一般而言，输入的字或者词用Word Embedding的方式表达，这样本来一维的文本信息输入就转换成了二维的输入结构，假设输入文档包含m个字符，而每个字符的Word Embedding的长度为d，那么输入就是m\*d的二维向量。

这里可以看出，因为NLP中的句子长度是不同的，所以CNN的输入矩阵大小是不确定的，这取决于m的大小是多少。卷积层本质上是个特征抽取层，可以设定超参数F来指定设立多少个特征抽取器(Filter)，对于某个Filter来说，可以想象有一个k\*d大小的移动窗口从输入矩阵的第一个字开始不断往后移动，其中k是Filter指定的窗口大小，d是Word Embedding长度。对于某个时刻的窗口，通过神经网络的非线性变换，将这个窗口内的输入值转换为某个特征值，随着窗口不断往后移动，这个Filter对应的特征值不断产生，形成这个Filter的特征向量。这就是卷积层抽取特征的过程。每个Filter都如此操作，形成了不同的特征抽取器。Pooling 层则对Filter的特征向量进行降维操作，形成最终的特征。一般在Pooling层之后连接全联接层神经网络，形成最后的分类过程。

可见，卷积和Pooling是CNN中最重要的两个步骤。Max Pooling Over Time是CNN模型中最常见的一种下采样操作。意思是对于某个Filter抽取到若干特征值，只取其中得分最大的那个值作为Pooling层保留值，其它特征值全部抛弃，值最大代表只保留这些特征中最强的，而抛弃其它弱的此类特征。

Max Pooling能减少模型参数数量，有利于减少模型过拟合问题。但是，对于文本分析任务来说，CNN模型采取Max Pooling Over Time有很多缺点：首先在文本中，特征的出现位置是很重要的信息，比如主语出现位置一般在句子开头，宾语一般出现在句子结尾等等，这些位置信息其实有时候对于文本分析任务来说还是很重要的，但是Max Pooling基本把这些信息抛掉了。在卷积层其实是保留了特征的位置信息的，但是通过取唯一的最大值，现在Pooling层只知道这个最大值是多少，而其出现位置信息并没有保留；另一个明显的缺点是：有时候有些强特征会出现多次，比如我们常见的TF-IDF公式，TF就是指某个特征出现的次数，出现次数越多说明这个特征越强，但是Max Pooling只保留了一个最大值，所以即使某个特征出现多次，现在也只能看到一次，就是说同一特征的强度信息丢失了。Max Pooling层的网络结构如下图所示。



1. Max Pooling层的网络结构

本文针对Max Pooling在处理文本分析时的不足，提出了一种新的池化方法Segment-Max Pooling(S-Max Pooling)。它将卷积层得到的特征向量按照需求进行分段，然后在切割出来的各个段落中取该部分的最大值。如0中，将Filter的特征向量切割成k个segment，然后在每个segment中取其最大值作为该部分的特征。该方法很明显保留了同一特征向量中的多个最大值特征，同时保留了多个局部特征的相对位置信息，如果多次出现的强特征，该方法也可以捕获到其特征强度。

1. Segment-Max Pooling层的网络结构



利用卷积层的权重矩阵提取的上下文特征(表示第个单词的号特征)，每篇文档词序列长度为，就会有个单词会被提取上下文特征，每个单词由种不同的共享权重提取种特征，所以一篇文档，经卷积层提取出的上下文特征的shape是，相当于图像识别的CNN中特征图深度这一概念。













其中，是通过第个共享权重提取的长度为的特征。

池化层的作用就是从卷积层提取出具有代表性的特征，对于上文中提到的上下文特征向量为可变长度()，Max Pooling从每个特征向量中取出一个最大值，也就是作为该层卷积的代表性特征，一共有取出个最大值，作为该文档最终的上下文特征向量。

为了在池化层保留更多的特征信息，我们需要对该文档的每个特征向量进行切割，求得各个部分的最大特征值。如果将该特征向量切割为k个局部特征，那么每个局部特征向量的长度为：



其向量表示形式如下所示：



其中，代表的是对第个共享权重提取的长度为的特征向量的S-Max Pooling处理。

因为文档长度是可变的，卷积层的单词选取范围也是可变的，所以卷积层处理之后所得到的上下文特征向量的长度也是可变的。我们需要在池化层来处理其可变性，使其成为固定长度的向量。传统的Max Pooling方法在每个特征向量中取一个最大值，可以将每个文档表示为具有固定长度的向量，如公式。本文中提出的方法同样可以保证在Pooling层输出固定长度的特征向量。



此特征向量是一个二维矩阵。其中，代表的是卷积层第个Filter所得到的特征向量在池化层的处理结果，它是一个长度为k的向量。最终，代表的特征矩阵。

通过上述分析，我们可以得到ConvMF-S算法的主要步骤，如表3-1所示。

|  |
| --- |
| 1. ConvMF-S算法 |
| 输入：每个项目的描述文档X和评分矩阵R | |
| 1. 嵌入层将one-hot编码的词向量嵌入成p维，得到词序列 2. 进入卷积层之前，reshape成channels\_last的形式 3. 卷积层使用3中不同window size的filter，对上一层的输出做卷积处理 4. 池化层对得到的每个特征向量做S-Max Pooling，提取特征 5. 将池化结果flatten，把多维的输入一维化 6. 将上面得到的结果作为inputs输入后面的全连接层，projection层进行降维处理 7. CNN输出文档的隐向量，然后结合高斯噪声初始项目特征矩阵 8. 初始化用户特征矩阵，并用项目和用户的特征矩阵去拟合评分矩阵R 9. 由数据计算得到PMF的结果，然后根据公式对算法进行优化，更新参数 | |

* 1. 本章小结

本章首先介绍了ConvMF的概率模型，说明了卷积神经网络集成到概率矩阵分解的过程。然后对卷积神经网络模型架构做了详细介绍，描述了其处理数据的过程。最后，针对卷积神经网络处理文本问题时其模型中存在的不足之处进行了分析，并提出了新的方法对模型进行改造。

1. 预训练词向量及模型优化
2. 1. 引言

基于文档上下文的深度混合推荐系统，最为关键的问题就是如何能更高效的利用文本数据生成高质量的项目潜在特征向量。这就涉及到了自然语言处理中的文本分析任务，因此本文在改进算法模型的基础上加入预先训练好的词向量，以提高推荐系统的质量，为了达到此目的本章使用词向量模型中非常热门的Word2vec算法，在IMDB电影评分数据集的基础上预训练词向量。并对模型进行了优化，提高了训练效率和推荐的准确性。

* 1. 词向量模型

词向量又称为词嵌入，是一类将词的语义信息映射到空间中去的自然语言处理技术。简单来说就是，对词汇表中的任意一个词，指定一个固定长度的实数值向量，就称为的词向量，为词向量的长度。由这些向量形成的几何空间被称为一个嵌入空间。

目前最流行的词嵌入模型是Word2vec。它是用来生成广泛语义关系的词嵌入模型。语言建模力求在给定之前的词语的情况下，计算一个词语出现的概率，也就是。运用链式法则和马尔科夫假设，我们就可以近似地通过之前出现的个词语得到每个词的概率乘积，从而得到整个句子或整篇文章的乘积:



Word2vec是从大量文本预料中以无监督的方法学习语义知识的一种模型，它被大量地用在自然语言处理(NLP)中。Word2vec其实就是通过学习文本数据然后用词向量的方式表征词的语义信息， Word2vec输出的词向量可以被用来做很多NLP相关的工作，比如聚类、找同义词、词性分析等等。其使用的词向量不是我们在第二章中提到的One-hot Representation那种词向量，而是Distributed representation的词向量表示方法。其基本思想是，通过训练将每个词映射成k维实数向量(k一般为模型中的超参数)，然后通过词之间的距离(比如余弦相似度，欧式距离等)来判断他们之间的语义相似度。而Embedding其实就是一个映射，将单词从原先所属的空间映射到新的多维空间中，也就是把原先的每个词从其所在空间嵌入到一个新的空间中去。

Word2vec模型的训练过程实际上是分为两部分的，第一部分为建立模型，第二部分是通过模型获取嵌入词向量。Word2vec的整个建模过程实际上与自动编码器(auto-encoder)的思想很相似，即先基于训练数据构建一个神经网络，当这个模型训练好以后，我们并不会用这个训练好的模型处理新的任务，我们真正需要的是这个模型通过训练数据所学得的参数。例如，隐层的权重矩阵，这些权重在Word2vec模型中实际上就是我们试图去学习的word vectors。

Word2vec模型实际上有两种不同的算法：Continuous Bag of Words (CBOW) 和 Skip-gram。这两种方法都是利用人工神经网络来作为它们的分类算法。其采用一个三层的神经网络，输入层-隐藏层-输出层。其核心的技术是根据词频用Huffman编码，使得所有词频相似的词隐藏层激活的内容基本一致，出现频率越高的词语，他们激活隐藏层数据越少，这样有效降低了计算的复杂度。

CBOW的目标是根据上下文来预测当前词语的概率。Skip-gram则与其刚好相反，它是根据当前词语来预测上下文的概率。两种算法的模型如图4-1所示。起初，每个单词都是一个随机的 N 维向量。经过训练之后，该算法利用CBOW或者 Skip-gram 的方法获得了每个单词的最优向量，也就是我们最终需要的词向量。

1. CBOW和Skip-gram模型框架



词典中的每一个词用一个向量表示，对应矩阵中的一个列向量，而具体对应哪一个向量则取决于该词在词典中的具体位置。

具体地，给定一个用于训练的词序列，词向量模型的目标函数是最大化平均似然概率：



预测任务可以通过一个多分类器完成，如softmax分类器：



上式中的表示词归一化之前的输出值。其计算表达式为：



其中，和是softmax的参数，通过对词向量进行取平均或者相连接得到。

对于得到的词向量，具有相近含义的词(如“powerful”和“strong”)在向量空间中的距离较近。这里有一个经典的结论是。

|  |
| --- |
| 1. Word2vec词向量训练算法示意 |
| 输入：单词序列 | |
| 1. 读取预料，统计词频信息 2. 构建词典，并初始化霍夫曼树以及随机初始化每个词的对应向量(维度默认为200) 3. 以行为单位训练模型 4. 获取当前行中的一个输入样本(当前词向量以及相邻几个词的词向量) 5. 累加上下文词向量中每个维度的值并求平均得到投影层向量X(w) 6. 遍历当前词到根节点(输出层的霍夫曼树)经过的每个中间节点 7. 计算中间节点对应的梯度g\*学习速率 8. 刷新投影层到该中间节点的误差向量 9. 刷新中间节点向量 10. 刷新上下文词向量(其实就是将误差向量累加到初始向量中) | |

## 4.3预训练词向量

4.3.1数据分析及预处理

本节实验中使用IMDB电影评论数据集，其核心数据集包含50000条评论，平均分为25K训练集和25K测试集。其标签的整体分布也是平衡的，有25k乐观的电影评论和25K悲观电影评论。整个数据集中，任何给定的电影不允许超过30条评论，因为同一电影的评论倾向具有相关性。此外，训练集和测试集各包含一组不连续的电影集合。在训练集和测试集有标签的评论中，负面评论的得分低于5分，积极评论的得分高于10分。

因为数据集中每条评论都是以单个文件的形式存在的，即训练集和测试集中各包含25000个txt文件。本文中使用训练集中的数据进行词向量的训练，为了方便Python利用panda算法库对数据进行处理，首先需要合并数据并保存在tsv类型的文件中。将25000个文件的文件名作为id存入第一列。我们认为大于5分的评分表示喜欢，小于5分的评分表示不喜欢，然后用1代表喜欢的情绪，0代表不喜欢的情绪，作为第二列保存在文件中，再将每个id对应的txt文件中的文本内容作为第三列存入文件中。其格式如下：

训练集数据格式

|  |  |  |  |
| --- | --- | --- | --- |
|  | id | sentiment | review |
| 0 | 5814\_8 | 1 | “With all this…kay.<br/><br/>Visually…. |
| 1 | 2381\_9 | 1 | “\”The Classic War of the Worlds\” by… |
| 2 | 7759\_3 | 0 | “The film starts with a manager(Nichola… |
| 3 | 3630\_4 | 0 | “Superbly trashy and wondrously … |

其中，最左边的是行数，id代表的是文件名，sentiment代表用户的情绪，review是用户的评论。

观察训练集发现，在评论中有一些字符对我们处理文本生成词向量没有多大的帮助，所以我们需要进行数据清洗。首先需要利用BeautifulSoup工具包删除html标签。然后利用正则表达式删除标点和数字。为了更方便的处理文本，我们将单词全部转换为小写，并分割成独立的单词数据。最后，删除了一些经常出现但没啥用的词语，也就是停用词，比如“a”、“an”、“the”等。

4.3.2实验结果分析

在生成词向量的实验中，我们使用gensim算法库，因为该算法库中有很多可读的Word2vec实现。

我们还可以通过寻找相似的词，来观察他们的语义信息和词向量是否相近，比如：寻找与“man”最相似的词，得到如下结果：

('woman', 0.6256189346313477)，('lady', 0.5953349471092224)，

('lad', 0.576863169670105)，('person', 0.5407935380935669)，

('farmer', 0.5382746458053589)，('chap', 0.536788821220398)，

('soldier', 0.5292650461196899)，('men', 0.5261573791503906)，

('guy', 0.5213091373443604)

寻找与“awful”最相似的词，得到如下结果：

('terrible', 0.7551683187484741)，('atrocious', 0.7340768575668335)，

('horrible', 0.7315883040428162)，('dreadful', 0.7080680131912231)，

('abysmal', 0.7010548114776611)，('horrendous', 0.6951696872711182)，

('appalling', 0.691646933555603)，('horrid', 0.6708598136901855)，

('amateurish', 0.6481891870498657)，('embarrassing', 0.6306308507919312)

从上文实验结果可以看出，我们训练出来的词向量能够区别不同的单词，能够区分“man”,“awful”的不同。我们得到了一个不错的词向量结果，可以将其嵌入推荐系统。

4.4模型优化

ConvMF模型中的CNN模块中output layer使用传统的tanh产生文档潜在矢量如公式所示。在各种激活函数中tanh是sigmoid函数的一个变种，其表达式为：



该激活函数当神经元的激活在接近0或1处时会饱和导致梯度消失。为了优化该问题，本文中使用了Maxout激活函数，其拟合能力非常强，它可以集合任意的凸函数。Maxout函数具有ReLU的优点，如计算简单，又不容易饱和，同时又没有ReLU的一些缺点，如容易崩溃。

为了优化用户潜在模型，项目潜在模型。CNN权重和偏差等变量，本文中使用了最大后验(MAP)估计，如下：



由上式可以得到：



其中，代表的是，代表的是，代表的是。

对于优化潜在向量的过程，本文中使用了坐标下降的方法。坐标下降在每一次迭代的时候，并不是按照梯度的方向走，而是按照坐标的方向走(比如我们要估计的是p维的一个向量，则坐标的方向则有p个，分别是p维空间上的单位向量)。坐标下降的每一次迭代过程，会按照一定的顺序(比如按照1,2,…,p的顺序)对每一个参数进行更新(更新第j个参数的时候，其他p-1个参数保持不变)，直到收敛。

1. 坐标下降算法

|  |
| --- |
| 要求一个的问题，其中各个是自变量，使用坐标下降的方法 |
| 1. 首先给定一个初始的点，如 2. for dim in range(1,n):   固定(其中i是除dim以外的其它维度)  以为自变量求取使得取得最小值的 |
| 1. 循环执行步骤(2)，直到的值不在变化或变化很小。 |

其关键点就是每次只变换一个维度，而其它维度都用当前值进行固定，如此循环迭代，最后得到最优解。在本文中也就是说假设W和V(或U)是一个常量，那么上式就变成了U(或V)的一个二次函数。然后，通过简单地将优化函数L相对于(或)进行微分，可以以封闭形式分析计算U(或V)的最优解：





其中，是具有的对角矩阵，作为其对角元素，是用户的的向量。对于项目，和的定义和和很像。公式表示了CNN的文档潜在矢量在生成项目潜在模型中的影响，其中表示的是平衡参数。

然而，不能像和那样通过分析解决方案进行优化，因为与CNN架构的特征密切相关，如池化层和非线性激活函数。尽管如此，我们观察到当和临时恒定时，L可以解释为具有正则项的平方误差函数。



为了优化W，本文中采用了反向传播算法。总体优化过程(,和被迭代地更新)被重复直到收敛。通过优化,和，最后可以预测用户对项目的未知评分：





为了训练CNN的权重，本文中使用了基于批量的RMSProp。它是从AdaGrad上修改得来的，也是一个自适应算法。

|  |
| --- |
| 1. RMSProp算法 |
| 全局学习速率，衰减速率，初始参数  小常数通常设置为(用于被小数除时的数值稳定) | |
| 1. 初始化累积变量 2. 从训练集中采样包含个样本{}的小批量，对应目标 3. 计算梯度： 4. 累积平方梯度： 5. 计算参数的更新：(逐元素应用) 6. 更新参数： 7. 重复上面(2)-(6)的步骤，直到达到停止准则 | |

相比于AdaGrad的历史梯度：



RMSProp优化算法不是像AdaGrad算法那样暴力直接的累加平方梯度，而是加了一个衰减系数来控制历史信息的获取多少。



这里加的系数，作用就相当于加了一个衰减系数来控制历史信息的获取多少。

鉴于神经网络都是非凸条件下的，RMSProp在非凸条件下结果更好，改变梯度累积为指数衰减的移动平均以丢弃遥远的过去历史。设置全局学习率之后，每次通过全局学习率逐参数的除以经过衰减系数控制的历史梯度平方和的平方根，使得每个参数的学习率不同。在参数空间更为平缓的方向会取得更大的进步(因为平缓，所以历史梯度平方和较小，对应学习下降的幅度较小)，并且能够使得陡峭的方向变得平缓，从而加快训练速度。

4.4.1模型复杂度分析

在每个迭代时期，所有用户和项目潜在模型在中更新，其中是观测到的评分数。在更新W时计算文档潜在矢量。用于更新W的时间复杂度由卷积层的计算支配，因此CNN的所有权重和偏置变量在中更新。结果，每个时期的总时间复杂度为，并且该优化过程与给定数据的大小成线性关系。

4.5本章小结

本章首先介绍了词向量的基本概念，然后详细介绍了词向量模型中最经典的Wrod2vec算法。其次本章对IMDB数据集进行了数据预处理，然后实现了Word2vec算法，以IMDB电影评论数据集为基础生成了词向量，并验证了生成的词向量的效果。最后在对ConvMF-S算法进一步研究的基础上，对模型做出了优化。

1. 实验及结果分析
2. 1. 实验环境及软件

5.1.1实验环境

本文所用实验环境为：处理器为Intel(R) Core(TM) i5-4430，GPU为3.00GHz，内存8G，硬盘150G。系统为Windows7 64位，Python版本3.5.2，Tensorflow版本为1.2.1，Keras版本为2.1.1。

5.2.2实验软件

本文中所有实验均是基于Python语言的，Python是一种高效的脚本语言，拥有很多标准库和丰富的第三方算法库，如本文中使用到了Numpy、gensim、Matplotlib等等。本文在基本编程语言Python的基础上，使用了以Tensorflow为后端的Keras深度学习框架，使模型实现起来更加方便，代码更加简洁。

* 1. 推荐质量评价方法

在本章实验中，作为评估推荐质量好坏的指标，使用了均方根误差(Root Mean Squared Error，RMSE)和平均绝对误差(Mean Absolute Error，MAE)，这与常规评分预测模型的目标函数直接相关。

其中，均方根误差RMSE的计算如式所示。



平均绝对误差MAE的计算如式所示。



* 1. 实验数据

为了证明本文提出的方法对提高评分预测的有效性，我们使用了从MovieLens和Amazon获得的三个真实世界的数据集。

本文中使用了MovieLens数据集中的MovieLens 100k和MovieLens 1M两个数据集。MovieLens 100k数据集中记录了943个用户对1682部电影的共100,000个评分，每个用户至少对20部电影进行了评分。保存了item的信息，也就是电影的信息，共1682部电影，其id依次是1、2、……、1682。文件中每一行保存了一部电影的信息，格式如下：

movie id | movie title | release date | video release date | IMDb URL | unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |

其中，最后19个字段保存的是该电影的类型，一个字段对应一个类型，值为0代表不属于该类型，值为1代表属于该类型。

该数据集中主要保存了所有的评分记录，每一行是一个用户对一部电影的评分，共100000条记录。当然，如果用户没有对电影进行评分，则不会被保存在该数据集中。评分的值在1到5之间。每一行数据格式如下：

user id | item id | rating | timestamp

其中，user id和item id分别代表用户和电影的id，rating是用户对电影的评分，时间戳timestamp是评分时间。

MovieLens 1M数据集中保存的是6040个用户对3952部电影的1000209个评分记录。

1. 2. 实验步骤

5.4.1数据预处理

本文实验中为了评估推荐模型在真实世界数据集中的整体性能，将每个数据集随机分为训练集(80%)，验证集(10%)和测试集(10%)。

类似于文献[41],[42]本文对所有数据的预处理描述如下：1)将原始文档的最大长度设置为300，2)删除停用词，3)计算每个词的TF-IDF得分，4)删除文档频率高于0.5的语料库特定停用词，5)选择顶部8000个不同的单词作为词汇，6)从原始文档中删除所有非词汇单词，7)为了使PMF能处理所有用户和项目，所有数据集的训练集中都至少包含每个用户和项目的一个评分。

5.4.2实验细节

为了训练CNN的权重，本文中使用了基于批量的RMSprop，每个小批量由128个训练项目组成。对于CNN的详细结构，本文中使用了以下设置：1)将文档的最大长度设置为300。2)在卷积矩阵分解模型中我们初始化了尺寸为200的随机字隐向量，这些字隐向量通过优化过程被训练。3)在本文第4章我们预训练了一个尺寸为200的嵌入模型来初始化潜在向量，并通过优化过程来训练这些字隐向量。4)在卷积层中，我们使用窗口大小为3，4和5的共享权重来考虑周围词的不同长度，并且使用每个窗口大小为100个特征层。5)代替CNN权重相关的L2正则化，本文中使用dropout来防止CNN的过渡拟合，并设置dropout率为0.2。

### 实验参数

为了研究不同参数对实验结果的影响，找到使模型达到最好的评估值的参数，本章中使用了网格搜索算法来训练最优参数。

网格搜索是指定参数值的一种穷举搜索算法，通过将估计的参数通过交叉验证方法来得到最优的学习算法。即，将各个参数可能的取值进行组合排列，列出所有可能的组合结果生成“网格”。然后将各个组合用于训练模型，并使用交叉验证对表现结果进行评估。例如，模型有三个参数，，，需要调节，每个参数分别有，，个可能的取值，那么三个参数一共组合个情况，其中每一种组合都可能是最优参数。将每种组合带入模型，采用交叉验证的方法对模型进行评估，所以一共有个对应的评估，选取其中最好的评估，其参数就是要找的最优参数。

交叉验证包括k折交叉验证(k-folder cross-validation)和留一法(leave-one-out cross-validation)。其中，k折交叉验证需要先将数据集划分为k个大小相似的互斥子集，即

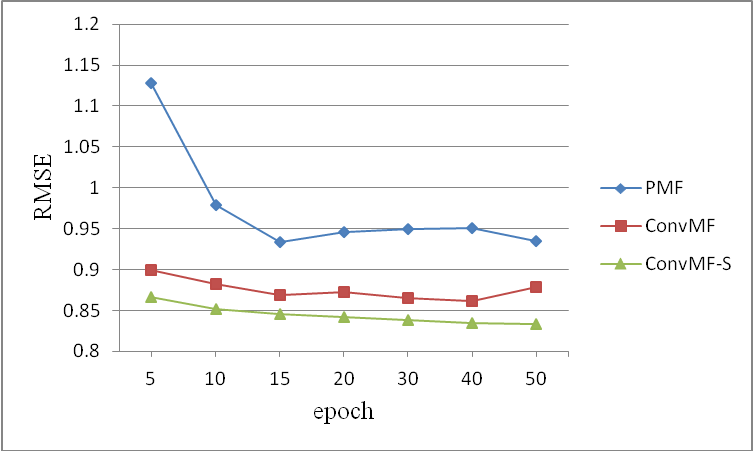


然后每次用k-1个子集作为训练集，余下的那个作为测试集，这样就可以得到k组训练集/测试集，从而可以做k次训练和测试，最终返回的是这k个测试结果的均值。留一法假设数据集中包含n个样本，另k折交叉验证法中的k=n。本文中使用的k折交叉验证中的k=10。

* 1. 实验分析

### 三种算法在同一数据集上不同迭代次数的比较

这一组是将本文提出的基于上下文的深度混合推荐系统与传统推荐算法PMF和卷积矩阵分解算法ConvMF在同一数据集上对不同迭代次数所得结果的对比。目的是在ML-100K数据集上验证改进的算法在收敛速度和最终的推荐质量上是否比另两个算法有所改善。

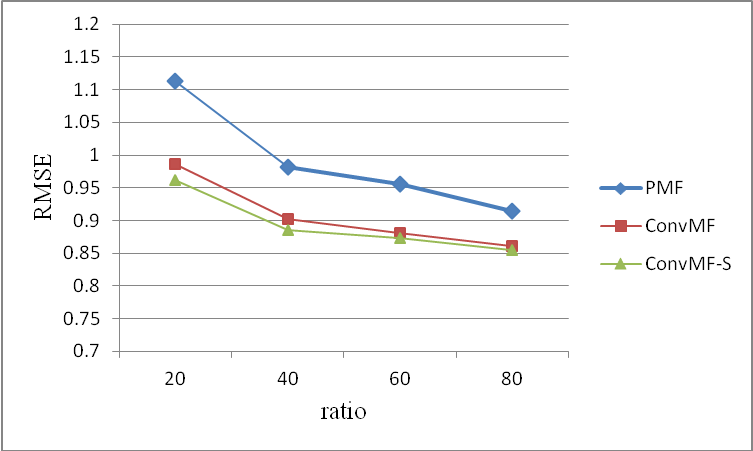


1. 三种推荐算法在ML-1M数据集上实验结果对比

由图5-1的折线图可以看出对于相对密集的MovieLens数据集，经典的协同过滤推荐算法(PMF)在迭代到第15次时达到最小值附近，RMSE值趋于稳定。加入项目文档分析模型后，推荐算法在迭代到第5次时就能达到很好的效果，随着迭代次数的增加，RMSE的值还在逐渐减小并最终优于PMF算法，说明加入项目文档分析模型，可以有效改善因评分数据稀疏所造成的推荐系统质量恶化的问题。而且改进的算法模型一直优于原有算法模型，说明改进是算法能更加有效的利用文本数据，并通过其提供推荐算法的准确性。

### 三种算法在数据集不同比例上的对比

这一组实验通过不同数据集来测试改进算法的推荐结果。通过从ML-100K数据集中随机抽样，生成不同稀疏度的七个附加训练集。然后将三个算法分别在不同的数据集上进行测试，其对比结果如下图所示。

1. 三种算法在ML-100K的不同训练集比例上进行比较

由图5-2的折线图可以看出，随着训练集所占百分比的越来越高，三种算法的RMSE值都是越来越小。很明显在训练集所占比例为20%时，PMF与另两种文档模型的推荐算法相比要差很多，且这种差值随着训练集所占比例的提高，越来越小，说明当数据集越是稀疏时，基于文档模型的推荐系统的效果更好。对比改进的算法模型与原算法模型也不难发现，在数据越是稀疏时，他们的差值越大，也同样说明了改进的模型，更有效的利用了文档数据。

* 1. 本章小结

本章开头首先介绍了实验的软件和硬件环境以及验证推荐算法质量所采用的评估标准，并根据前文中提出的改进算法，基于Python语言做出实验代码，以Movielens数据集为输入，并实现了另外另种推荐算法与本文中的改进算法进行对比，利用第二节的评估方法进行算法性能评估。由最后的实验结果分析可知，本文提出的算法在预测精度，收敛速度等方面都优于传统的协同过滤算法，并有效的改善了传统算法中的数据稀疏问题。

结 论

随着互联网的飞速发展，用户和商品项目爆发式增长，评分数据越来越稀疏。而传统的推荐算法大部分都是对评分矩阵进行相关计算然后做出推荐的。特别是基于协同过滤的推荐算法在稀疏的数据上做出的推荐结果越来越差。本文在此问题的基础上进行了广泛研究，发现结合深度学习，充分利用与推荐内容有关的辅助信息可以改善数据稀疏性问题，得到较好的推荐结果。

深度学习是近年来非常热门的研究领域，其在自然语言处理和图像处理等方面已经取得了非凡的成就。深度学习可以从海量数据样本中学习数据集的本质特征，也可以通过从多元异构数据中进行自特征学习，将不同数据映射到一个相同的隐空间以获得数据的同一表征。将深度学习与推荐系统进行融合，可以有效利用多元异构数据，缓解数据稀疏问题。本文在对此方面进行深入研究后发现，利用深度学习中的自然语言处理方面的方法去分析文本数据(例如评论，摘要)可以缓解数据稀疏问题，提高推荐算法的准确性。

本文介绍了一种基于文档的混合深度推荐算法(ConvMF-S)。本文工作主要有一下几个方面：

1. 对ConvMF进行了研究，特别是其中用CNN分析文本数据进行了深入研究，发现了其在特征处理时，丢失了特征的位置信息，对同一特征的多次出现没有保留其特征强度，在此基础上提出了算法的改进模型(ConvMF-S)。改进的算法不仅可以保留特征的相对位置信息，还能保留特征强度，更加充分的利用了文本数据。
2. 在Word2vec算法模型的基础上，利用IMDB数据集，进行了词向量的训练，并用实验证明了产生的词向量的有效性。并对基于文档的混合深度推荐进行了算法优化，提高了其运行效率。
3. 算法实现与实验结果分析。根据提出的基于文档的混合深度推荐算法，利用Python作为基础语言实现算法，并与传统协同过滤算法和原有模型进行对比，通过各方面的比较，证明了提出的改进算法的有效性。
4. 将预训练的词向量嵌入到推荐算法模型中，证明了本文训练的词向量可以有效提高推荐算法的质量，特别是在数据稀疏时。

算法的不足与未来的研究方向：

(1)由于该算法利用深度学习进行文本分析，当文本数据特别多或选取的维度较高时，模型过于复杂，执行效率不高。可以考虑利用分布式对其进行改进以提高训练效率。

(2)引入基于文档建模的方法以提高推荐效果后，可以考虑加入用户信息的分析方法，来提高用户潜在特征矩阵的质量，进一步提高推荐算法的推荐质量。

参考文献

1. Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl, Grouplens: an open architecture for collaborative filtering of netnews[C]. Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp. 175–186. ACM, 1994.
2. Xiaoyuan Su, and Taghi M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques [J]. Advances in Artificial Intelligence 2009(2009):4.
3. Zhi-Dan Zhao, and Ming-Sheng Shang. User-based collaborative-filtering recommendation algorithms on hadoop[C]. In Knowledge Discovery and Data Mining, 2010. WKDD '10. Third International Conference on, pp.478-481.IEEE, 2010.
4. Badrul Sarwar, George Karypis, Joseph Konstan, et al. Item-based collaborative filtering recommendation algorithms[C]. Proceedings of the 10th International Conference on World Wide Web. ACM, 2001.
5. Ana Belén Barragáns-Martínez, EnCosta-Montenegro, Juan C Burguillo, et al. A Hybrid Content-Based and Item-Based Collaborative Filtering Approach to Recommend TV Programs Enhanced with Singular Value Decomposition [J]. Information Sciences, 180(22):4290-4311, 2010.
6. John S. Breese., David Heckerman., and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering[C]. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI ’98), pp.43-52, 1998.
7. Nathan Srebro, and Tommi Jaakkola. Weighted low-rank approximations [C]. Proceedings of 20th International Conference on Machine Learning (ICML), vol.3, pp.720-727, 2003.
8. Daniel D. Lee, and H. Sebastian Seung. Algorithms for non-negative matrix factorization[C]. Advances in Neural Information Processing Systems (NIPS), 13(6):556-562, 2001.
9. Joonseok Lee, Seungyeon Kim, Guy Lebanon, et al. Local low-rank matrix approximation[C]. Proceedings of the 30th International Conference on Machine Learning (ICML), pp.82-90, 2013.
10. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition[C]. Proceedings of the IEEE, 86(11), pp.2278-2324, 1998.
11. Dan C. Ciresan, Alessandro Giusti, Luca M. Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images[C]. Proceedings of Advances in Neural Information Processing Systems (NIPS), pp.2852-2860, 2012.
12. Quoc V. Le, Marc’Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, and Andrew Y. Ng. Building high-level features using large scale unsupervised learning[C]. Proceedings of the 29th International Conference on Machine Learning (ICML), 2012.
13. Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, and Gerald Penn. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition[C]. Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference, pp.4277-4280, 2012.
14. Ossama Abdel-Hamid, Li Deng, and Dong Yu. Exploring convolutional neural network structures and optimization techniques for speech recognition[C]. Proceedings of Interspeech, 2013.
15. Tara N. Sainath, Abdel-rahman Mohamerd, Brian Kingsbury, and Bhuvana Ramabhadran. Deep convolutional neural networks for LVCSR[C]. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pp.8614-8618. IEEE, 2013.
16. Li Deng, Ossama Abdel-Hamin, and Dong Yu. A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion[C]. Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference, pp.6669-6673, 2013.
17. Ruslan R. Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering[C]. Proceedings of the 24th international conference on Machine learning, pp. 791–798. ACM, 2007.
18. C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’11, pages 448–456. ACM Press, August 2011.
19. G. Ling, M. R. Lyu, and I. King. Ratings meet reviews, a combined approach to recommend. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys ’14, pages 105–112, New York, NY, USA, 2014. ACM.
20. A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In International Conference on Machine Learning Workshop on Deep Learning for Audio, Speech, and Language Processing, 2013.
21. R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In Advances in Neural Information Processing Systems, volume 20, 2008.
22. H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’15, pages 1235–1244, New York, NY, USA, 2015. ACM.
23. Charu Chandra Aggarwal, Stephen C. Gates, and Philip Shi-lung Yu. System and method for generating taxonomies with applications to content-based recommendations: US, US6360227 [P], 2002.
24. Michael J. Pazzani, and Daniel Billsus. Content-Based Recommendation Systems [M]. The Adaptive Web, pp. 325-341. Springer Berlin Heidelberg, 2007.
25. Robin Burke. Knowledge-Based Recommender Systems. Encyclopedia of Library and Information Science (ELIS), 69(32), 2000.
26. Nikolaos Korfiatis, and Marios Poulos. Using online consumer reviews as a source for demographic recommendations: A case study using online travel reviews [J]. Expert Systems with Applications, 40(14):5507-5515, 2013.
27. Robin Burke. Hybrid Recommender Systems: Survey and Experiments [J]. User Modeling and User-Adapted Interaction (UMUAI), 12(4):331-370, 2012.
28. John S. Breese., David Heckerman., and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering[C]. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI ’98), pp.43-52, 1998.
29. Lee D D,Seung H S. Learning the parts of objects by non-negative matrix factorization.[J]. Nature,1999,401(6755).
30. A. van den Oord, S. Dieleman, and B. Schrauwen. Deepcontent-based music recommendation. In Advances in Neural Information Processing Systems 26, pages 2643–2651. Curran Associates, Inc., 2013.
31. HERLOCKER J L, KONSTAN J A, TERVEEN L G, et al. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information Systems, 2004, 22(1): 5-53.
32. 刘建国, 周涛, 郭强, 等. 个性化推荐系统评价方法综述[J]. 复杂系统与复杂性科学, 2009, 6(3): 1-10.
33. GUNAWARDANA A, SHANI G. A survey of accuracy evaluation metrics of recommendation tasks[J]. Journal of Machine Learning Research, 2009, 10: 2935-2962.
34. SHARDANAND U, MAES P. Social information filtering：algorithms for automating “word of mouth”[C]//Proceedings of ACM CHI’95 Conference on Human Factors in Computing Systems. New York: ACM Press, 1995: 210-217.
35. BREESE J S, HECKERMAN D, KADIE C. Empirical analysis of predictive algorithms for Collaborative filtering[C]//Proceedings of the 14th conference on Uncertainty in Artificial Intelligence. [S.l.]: [s.n.], 1998, 461(8): 43-52.
36. BALABANOVIC M, SHOHAM Y. Fab: content-based collaborative recommendation[J]. Comm ACM, 1997, 40(3): 66-72.
37. PAZZANI M J, BILLSUS D. Learning and revising user profiles: the identification of interesting Web sites[J]. Machine Learning, 1997, 27(3): 313-331.
38. VAN RIJSBERGEN C J. Information retrieval[M]. MA, USA: Butterworth- Heinemann Newton, 1979.
39. Donghyun Kim,Chanyoung Park,Jinoh Oh,Hwanjo Yu. Deep hybrid recommender systems via exploiting document context and statistics of items[J]. Information Sciences,2017,417.
40. Kim Y. Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.
41. Kalchbrenner N, Grefenstette E, Blunsom P. A Convolutional Neural Network for Modelling Sentences[J]. Eprint Arxiv, 2014, 1.
42. Chen Y, Xu L, Liu K, et al. Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks[C]// The, Meeting of the Association for Computational Linguistics. 2015.
43. Lei T, Barzilay R, Jaakkola T. Molding CNNs for text: non-linear, non-consecutive convolutions[J]. Indiana University Mathematics Journal, 2015, 58(3):págs. 1151-1186.
44. Dong L, Wei F, Zhou M, et al. Question Answering over Freebase with Multi-Column Convolutional Neural Networks[C]// Meeting of the Association for Computational Linguistics and the, International Joint Conference on Natural Language Processing. 2015:260-269.
45. Kalchbrenner N, Espeholt L, Simonyan K, et al. Neural Machine Translation in Linear Time[J]. 2016.
46. Ma M, Huang L, Xiang B, et al. Dependency-based Convolutional Neural Networks for Sentence Embedding[J]. 2015:174-179.
47. J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, 2014.
48. C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’11, pages 448–456. ACM Press, August 2011.
49. H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’15, pages 1235–1244, New York, NY, USA, 2015. ACM.
50. Y. Kim. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, 2014.
51. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. Journal of Machine Learning Research (JMLR), 12:2493–2537, Nov. 2011.

攻读硕士学位期间承担的科研任务与主要成果

致 谢

时光荏苒，岁月如梭。转眼间三年的研究生求学生活即将结束，还记得匆匆那年初到燕山大学时候的欣喜和激动，如今站在毕业的门槛上，回首往昔，奋斗和辛劳成为丝丝的记忆，甜美和欢笑也尘埃落定。燕山大学以其优良的学习风气、严谨的科研氛围教我求学，以其博大包容的情怀胸襟、浪漫充实的校园生活育我成长。值此毕业论文完成之际，我谨向所有关心、爱护、帮助我的人们表示最诚挚的感谢与最美好的祝愿。

首先，我要感谢我的研究生导师张大鹏老师。在我读研的三年中，张老师无论在学习上还是平时的生活中都特别照顾我，他学识渊博、治学严谨、循循善诱，给了我很多关怀。我从老师身上不仅学到了很多专业的知识，还学到了做人的道理，令我终身受用。在我写论文的过程中，张老师也给了我很多悉心的指导，从论文的选题到后来的研究工作再到最后的论文撰写，每一步都是在张老师的指导下完成的，倾注了张老师大量心血。

其次，我要感谢实验室的同门师兄弟们和我们班级的各位同学。三年时间里，无论是有关学术的正式交流还是生活中的非正式交流，从他们身上我学到了之前所欠缺的东西。每当我遇到困难，他们总是不遗余力的帮助我，并把所有的知识和学习资料毫无保留的分享出来。正式他们陪伴我读过了人生中最磨砺意志同时也是收获最多的三年。

再次，我要特别感谢我的父母和家人。他们不仅为我提供了温暖的港湾，还为我准备了随时远航的帆船。父母在物质上和精神上给了我最大的支持，也给了我最大的动力。感谢父母为我所做的一切。

最后，我要向燕山大学信息科学与工程学院的全体老师表示由衷的感谢！感谢各位老师的辛勤培养和谆谆教诲，感谢百忙之中评阅本论文和出席答辩的各位教授！感谢我身边的朋友、同学。同时对本文写作中所参考和引用的国内外研究成果、著作、文献的作者表示最真挚的感谢！

作者简介

|  |  |
| --- | --- |
|  | 姓名：刘建成 |
| 性别：男 |
| 民族：汉族 |
| 籍贯：河北省深州市 |
| 2011年9月考入沈阳理工大学计算机科学与技术专业。 |
| 2015年6月于沈阳理工大学取得工学学士学位。 |
| 2015年9月至今于燕山大学信息科学与工程学院攻读计算机技术专业工学硕士学位。 |