

**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

学位类别工程硕士

指导教师

**2018年5月**

中图分类号：TP312 学校代码：10216

UDC：004 密级：公开

**工程硕士学位论文**

基于上下文的深度混合推荐系统的研究

|  |  |  |
| --- | --- | --- |
| 硕士研究生 | ： |  |
| 导师 | ： |  |
| 副导师 | ： |  |
| 申请学位 | ： | 工程硕士 |
| 工程领域 | ： | 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 and Technology

**Research of context-based deep hybrid recommender system**

by

Supervisor: Associate Professor

**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《基于上下文的深度混合推荐系统的研究》，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

《基于上下文的深度混合推荐系统的研究》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。ConvMF算法模型可以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，并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，提出了一种新颖的基于上下文的深度混合推荐算法(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，通过坐标下降算法和RMSProp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

The sparseness of user-to-project scoring data is one of the main factors in the deterioration of the recommended system. In order to deal with sparseness problems, some recommendation techniques consider auxiliary information to improve the accuracy of scoring prediction. When scoring data is particularly sparse, document-based modeling approaches improve accuracy by using textual data such as reviews, abstracts, or synopses. However, due to the inherent limitations of the term bag model, they still have difficulties in effectively using the contextual information of the document and can only understand the document shallowly. ConvMF algorithm model can capture the context information of the document and further improve the prediction accuracy of the score. However, the convolutional neural network in ConvMF ignores the feature's location information when analyzing the text, and loses the information strength of the same feature.

In order to solve the problems in the above models, this paper deeply researches the convolutional network and proposes a novel context-based deep hybrid recommendation algorithm (ConvMF with Segmeng-Max Pooling, ConvMF-S). The content of this article is mainly divided into the following sections:

First of all, this paper introduces the current research status of recommendation systems, and introduces various algorithms in the recommendation system, especially the most classic collaborative filtering technology in personalized recommendation systems. After fully understanding the principles of various algorithms, the advantages and disadvantages of these algorithms were analyzed in depth. Then introduced the application of deep learning in the recommendation system.

Secondly, this paper deeply studies the application of convolutional neural network in natural language processing, improves the existing ConvMF model, and proposes a new model ConvMF-S.

Again, this article has conducted an in-depth study of the optimization of the ConvMF-S model. First, the CNN embedded layer is initialized by pre-training the word vector model, and then the model is further optimized by the coordinate reduction algorithm and RMSProp.

Finally, this paper validates the improved model algorithm through relevant experiments, and compares it with other algorithms from various aspects, and analyzes and evaluates the experimental results.

Keywords: Deep Learning; Recommendation; NLP; Word Vector

目 录

[摘 要 I](#_Toc511074877)

[Abstract II](#_Toc511074878)

[第 1 章 绪 论 1](#_Toc511074879)

[1.1 课题背景及研究意义 1](#_Toc511074880)

[1.2 国内外研究现状 2](#_Toc511074881)

[1.3 现有算法存在的问题 4](#_Toc511074882)

[1.3.1 基于内容的算法 4](#_Toc511074883)

[1.3.2 协同过滤推荐算法 4](#_Toc511074884)

[1.3.3 基于知识的算法 4](#_Toc511074885)

[1.3.4 基于人口统计学的算法 5](#_Toc511074886)

[1.3.5 混合推荐算法 5](#_Toc511074887)

[1.4 论文研究内容 5](#_Toc511074888)

[1.5 论文的组织结构 6](#_Toc511074889)

[第 2 章 卷积矩阵分解相关理论研究 7](#_Toc511074890)

[2.1 协同过滤推荐算法 7](#_Toc511074892)

[2.2 卷积神经网络 10](#_Toc511074893)

[2.3 词向量相关研究 13](#_Toc511074894)

[2.4 推荐系统评估方法 14](#_Toc511074895)

[2.4.1预测评分准确度的评估方法 14](#_Toc511074896)

[2.4.1 分类准确度的评估方法 15](#_Toc511074897)

[2.5 本章小结 16](#_Toc511074898)

[第 3 章 卷积矩阵分解的模型改进 17](#_Toc511074899)

[3.1 引言 17](#_Toc511074901)

[3.2 卷积矩阵分解算法模型 17](#_Toc511074902)

[3.2.1 卷积矩阵分解模型构建 18](#_Toc511074903)

[3.2.2 卷积神经网络模型架构 19](#_Toc511074904)

[3.3 问题定义及改进措施 21](#_Toc511074905)

[3.4 本章小结 26](#_Toc511074906)

[第 4 章 预训练词向量及模型优化 27](#_Toc511074907)

[4.1 引言 27](#_Toc511074909)

[4.2 词向量 27](#_Toc511074910)

[4.3 Word2vec词向量模型 28](#_Toc511074911)

[4.3.1 Hierarchical Softmax优化 30](#_Toc511074912)

[4.3.2 Negative Sampling优化 30](#_Toc511074913)

[4.3训练词向量 31](#_Toc511074914)

[4.3.1数据集预处理 31](#_Toc511074915)

[4.4 词向量评估 32](#_Toc511074916)

[4.5 模型优化 33](#_Toc511074917)

[4.5.1 模型复杂度分析 36](#_Toc511074918)

[4.6 本章小结 36](#_Toc511074919)

[第 5 章 实验及结果分析 37](#_Toc511074920)

[5.1 实验环境及软件 37](#_Toc511074922)

[5.1.1 实验环境 37](#_Toc511074923)

[5.1.2 实验软件 37](#_Toc511074924)

[5.2 推荐质量评价方法 37](#_Toc511074925)

[5.3 实验数据 37](#_Toc511074926)

[5.4 实验步骤 38](#_Toc511074927)

[5.4.1 数据预处理 38](#_Toc511074928)

[5.4.2 实验细节 38](#_Toc511074929)

[5.4.3 实验参数选取 39](#_Toc511074930)

[5.5 实验分析 39](#_Toc511074931)

[5.5.1 三种算法在同一数据集上不同迭代次数的比较 39](#_Toc511074932)

[5.5.2 三种算法在数据集不同比例上的对比 40](#_Toc511074933)

[5.5.3 嵌入词向量对模型的影响 42](#_Toc511074934)

[5.6 本章小结 42](#_Toc511074935)

[结 论 43](#_Toc511074936)

[参考文献 45](#_Toc511074937)

[攻读硕士学位期间承担的科研任务与主要成果 50](#_Toc511074938)

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。如今，社会生活中各个领域都随处可见推荐系统的身影，它已经影响着人们生活的方方面面。比如，阿里巴巴和亚马逊的电商平台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；网易云音乐用推荐系统为用户推荐符合其口味的歌曲。

推荐系统的核心是推荐算法，随着科学的不断发展，科学家们先后提出了多种推荐算法。协调过滤算法、基于内容的推荐算法和混合推荐算法是目前应用较广的，其中协同过滤算法是所有推荐算法中最热门也是应用最广的。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。比如，亚马逊的电商平台使用的推荐系统，通过高质量的推荐广受好评，并由此给其带来了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，一些推荐技术中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。最近几年，很多科学工作者将深度学习和推荐系统结合，以解决推荐系统面临的诸多问题，提高推荐质量。

2006年Hinton等人在神经网络的基础上，提出了深度学习的概念。此后，深度学习飞速发展，成为机器学习领域中一个备受瞩目的研究方向。深度学习模型是一种复杂的模型，以海量数据为基础，从底层数据中抽取更高层的属性或特征。例如，具有多个隐藏层的多层感知机就是一种深度学习模型。近年来，深度学习在机器视觉和自然语言处理等方面取得了革命性的进展。

推荐系统中嵌入深度学习，可以改善推荐系统所面临的很多问题，并有效提高推荐效果。一方面，深度学习复杂的算法模型可以更加有效的利用用户和商品的海量数据，从大量数据中抽取更高级的特征，获得用户和商品的深层次表示。另一方面，深度学习可以利用多源的复杂的数据信息，从多种异构数据中抽取特征映射到一个相同空间，统一数据的特征表示。结合深度学习的推荐系统，可以有效利用多源的海量数据，缓解数据稀疏和冷启动造成的推荐系统预测精度下降的问题。

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具有进步意义，可以帮助提高推荐质量，改善人们的生活，使人们更容易获得其所需的信息。

## 国内外研究现状

目前，推荐系统的研究范围越来越广泛。根据推荐算法所使用信息和知识来源的不同，推荐算法可以被分成：基于内容的推荐，协同过滤推荐，基于知识的推荐，基于人口统计学的推荐和混合推荐。其中协同过滤算法是所有推荐算法中应用最成功的，该推荐的研究方向已经涉及到了生活中的方方面面。

协同过滤推荐算法[1],[2]以用户的历史行为数据为基础，通过挖掘用户的评分数据来发现其兴趣偏好，然后给用户推荐其所需要是商品。协同过滤推荐算法根据研究对象的不同有基于用户和基于项目的推荐算法，另外还有一种以算法模型为基础的推荐算法。

基于用户的协同算法[3]在评分矩阵中通过用户的评分挖掘与当前用户相似的用户群，从用户群中取相似度较高的个用户作为近邻用户集，再通过近邻用户对商品的评分，选取评分最高的个商品放入用户的推荐列表中。基于项目的协同过滤算法[4],[5]的基本原理与此类似，该算法通过不同用户对项目的评分计算项目的相似度，根据目标用户对项目的评分寻找其可能会喜欢的项目集，从项目集中选取评分最高的个项目推荐给用户。基于模型的协同过滤推荐算法[6]是利用用户的历史行为数据，如评价、收藏等，然后通过模型训练(如贝叶斯模型，聚类模型或神经网络模型)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)是一种以数学模型为基础的推荐算法。矩阵分解是将一个高维稀疏矩阵(评分矩阵)分解为两个低维矩阵(用户特征矩阵和项目特征矩阵)，用两个低维特征矩阵的乘积拟合原始评分矩阵，对没有评分的用户-项目进行预测。带权重的矩阵分解方法[7]和基于非负矩阵的矩阵分解方法[8]、基于矩阵局部性的矩阵分解方法[9]都是在此算法上进行的改进，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层次里的抽象特征表示。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(Convolutional neural network, CNN)和循环神经网络(Recurrent neural network, RNN)。其中卷积神经网络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池化层构成。多个模块通过相互堆叠，一个模块在另一个模块的上方，上面模块的输入作为下面模块的输入，这样构成的模型就是深度模型。卷积层的权值共享和池化层的子采样，都会减少下一个模块的数据量，使得CNN有一些抗扭曲的能力，即有一定的不变性。目前，卷积神经网络已经在计算机视觉领域[10]-[11][12]应用中取得非常好的成果。近年来，卷积神经网络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和项目特征。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，如主题模型(Latent Dirichlet Allocation, LDA)和堆叠降噪自动编码机(Stacked Denoising Auto-Encoders，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模型，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等人提出了将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]算法，从而在评分预测精度方面产生更准确推荐效果[22]。虽然许多结合深度学习的推荐算法被提出，但是基于深度学习的推荐算法的研究仍然非常少，不够成熟。

## 现有算法存在的问题

### 基于内容的算法

基于记忆(启发式)的方法和基于模型的方法是基于内容的算法中用到的两种方法。基于内容的推荐[23]推荐给用户的商品是与用户过去喜欢的商品相似的商品。通过挖掘用户过去喜欢的商品可以得到用户画像，进而能够找到最适合用户的推荐[24]。该算法根据用户的兴趣来进行建模，通过对物品属性维度的增加，可以将更准确的商品推荐给用户，获得更好的推荐质量。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。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

协同过滤推荐算法给用户推荐和他兴趣相似的用户过去喜欢的商品，该算法认为有对同一商品有相似评分的用户之间有着相似的兴趣爱好。协同过滤算法不需要对用户和商品进行分析建模，也不需要对物品的属性进行分析，这种方法是与领域无关的。利用该方法可以有效的帮助用户发现潜在兴趣爱好，但是由于该方法使用的是用户和物品的评分矩阵。所以该方法的推荐效果，依赖于用户对物品评分的多少以及评分的准确性。在大多数啊情况下，用户对物品的评分用一个二维矩阵来存储，过于稀疏的评分矩阵会降低推荐系统的准确性，而且新加入的用户由于没有评分存在冷启动的问题，部分用户的错误评分也会严重影响推荐结果的准确性。

### 基于知识的算法

基于知识的推荐[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，都需要向系统中输入其基本信息，系统通过分析基本信息就能获得用户画像和商品的属性特征。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。

### 基于人口统计学的算法

基于人口统计学的推荐[26]，是所有推荐算法中最容易实现的一种。该算法通过用户个人信息寻找相似用户，然后基于相似用户有着相似的兴趣偏好这一观点对当前用户进行商品推荐。通过相似用户进行推荐的思想和基于用户的协同过滤类似，不同的是协同过滤的相似度来源于用户的评分信息，基于人口统计学的算法，是根据系统中的用户个人信息包括用户的性别、年龄、所在城市、受教育程度、工作等各方面的情况，通过分析这些信息得到用户的画像。利用用户画像在系统中匹配与当前用户最相似的用户群，将相似用户群中评价最高的商品推荐给目标用户。该推荐算法需要的是用户的个人信息，而不是用户-商品的评分矩阵，因此该算法不存在冷启动的问题，只要能获得用户的画像信息就可以进行推荐。然而，用户画像所需要的信息都属于用户的隐私，收集起来比较困难，而且用户画像信息即使收集到了也会有很大的噪声。

### 混合推荐算法

单独使用上述推荐算法中的一种，都会存在某方面存在缺陷。为了在不同情况下都能尽可能的达到最优的推荐效果，有人提出将多个推荐算法一起使用的混合推荐算法[27]。关于如何合理的使用不同的推荐算法，目前有几种比较流行的组合方法。比如，按照一定的权重将几种不同的推荐算法通过线性组合的方式一起使用的加权组合方法(Weighted Hybridization)，为了达到最好的推荐效果，具体权重值的设置要在测试集上反复训练。数据量、用户和商品不同的情况下，不同的推荐算法在性能上可能有很大的差异，于是根据不同的情况可以自由切换不同推荐算法的切换混合(Switching Hybridization)方法被人提出。还有一种较为流行的混合推荐算法是为了让用户得到更全面的推荐，将多种推荐算法的推荐结果分区域显示给用户，称为分区混合(Mixed Hybridization)方法，该方法在电子商务领域应用最为广泛，因为推荐的商品更多更全面，用户更容易找到他们需要的东西。在推荐系统中加入深度学习的方法也可以算作一种混合推荐。

## 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。

本文工作可简要概括为：

(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,进一步提高推荐质量。

## 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，主要内容如下：

第1章 主要介绍了研究课题的相关背景与意义，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 对卷积矩阵分解的算法模型进行详细介绍，针对其中用卷积神经网络从文本数据中提取特征过程中的不足，提出了一种新的池化方法S-Max Pooling，在池化层的下采样过程中提取了多个特征信息并保留了特征的位置信息以及特征强度。

第4章 深入研究了词向量模型(word2vec)，并以IMDB电影评论为数据集训练词向量模型，获得词向量，通过实验验证了生成的词向量的有效性。对提出的ConvMF-S算法模型进一步研究，并对算法进行优化，计算模型算法复杂度。

第5章 在前文理论研究的基础上，实现算法模型。与其它推荐算法在movielens100k数据集上进行实验对比，通过实验数据分析论证了改进算法的优越性以及训练的词向量的可用性。

最后，根据研究成果，对本文研究课题做出总结，并指明了进一步的研究方向以及未来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

在种类繁多的推荐算法当中，基于协同过滤的算法、基于人口统计学的算法和基于内容的推荐算法是三种较为流行的推荐算法。其中，最流行、应用范围最广的是基于协同过滤的算法，该技术在不同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通过用户对商品的历史行为来进一步获得用户或商品的信息，然后根据获得的信息做出有效的推荐。一般来说，协同过滤算法根据使用信息的不同可以细分为三种：第一种是基于用户的协同过滤，第二种是基于项目的协同过滤，第三种是基于模型的协同过滤。

基于用户的协同过滤通过所有用户对物品的评分得到评分矩阵，利用余弦相似度或者欧式距离等计算方法，计算用户之间的相似度，找到与目标用户兴趣偏好相似的用户，即近邻用户群。近邻用户可能较多，一般是从中选择最相似的个用户。然后，基于这个近邻用户的历史偏好信息，寻找近邻用户中评分最高的个物品作为推荐列表向用户进行推荐。基于项目的协同过滤也是要通过相似度进行推荐，与基于用户的协同过滤不同的是，这里要计算的是物品的相似性，通过目标用户的历史兴趣偏好，找到用户最感兴趣的物品，再计算该物品与其他物品的相似度，将相似度最高的个物品放入用户的推荐列表。比如，在网上买了一本数据结构，下次登录的时候网站所展示的推荐列表就都是一些有关编程、操作系统的书籍，这里用到的就是基于项目的协同过滤算法。

项目的协同过滤算法中，考虑到物品具有短时间不变性，相似物品的计算可以在线下进行，提高了线上推荐的实时性，准确度也在可接受的范围，但是缺乏推荐的多样性，不能带给用户有惊喜、新颖的推荐。基于用户的协同过滤与基于项目的协同过滤不同，其用户和用户之间的相似度是在线计算的，且时间复杂度较高，推荐实时性较差。但是通过不同用户的偏好可以帮助目标用户发现一些新类别的物品，推荐多样性较高。对于注重实时的小型推荐系统可以使用基于项目的协同过滤算法，基于用户的协同过滤算法应用于大型的推荐系统效果更好。

上面两种协同过滤算法不需要获得用户或者商品的特征模型，也不需要用户或商品的统计信息，这两种方法是与领域无关的，可以有效的发现用户隐藏的兴趣偏好。然而，基于内容的协同过滤严重依赖用户评分矩阵中评分的多少和评分的质量。通常用户的兴趣偏好是通过用户-商品评分矩阵来表现的，该矩阵为稀疏的高维矩阵，且矩阵中部分用户对物品的错误评价会大大影响推荐结果的准确度。

在不同的协同过滤算法中，目前最为流行的协同过滤算法是基于模型的协同过滤，其推荐质量也被实验证明是协同过滤技术中最好的。矩阵分解算法是将高维的用户评分矩阵分解成两个低维矩阵，并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。

隐语义模型(Latent Factor Model, LFM)最开始在文本挖掘领域被提出，主要用于寻找文本的隐含语义，随着LFM在Netflix Prize推荐比赛中的突出表现，近年来隐语义模型得到了学者们越来越多的关注。后来人们根据隐语义模型的思想提出了矩阵分解技术，它也是实现隐语义模型使用最为广泛的一种方法。推荐领域大神Yehuda Koren利用矩阵分解技术勇夺Netflix Prize推荐比赛冠军。科学实验证明，在推荐系统中使用基于邻域的协同过滤算法(基于用户或基于项目的协同过滤算法)与使用基于模型的协同过滤(矩阵分解)算法相比，后者具有明显的优越性。如今推荐系统研究领域中最为流行的算法就是矩阵分解算法。

矩阵分解的核心思想是将高维且稀疏的评分矩阵分解为两个低维的特征矩阵，一个是用户的特征矩阵，另一个是物品的特征矩阵，利用这两个特征矩阵的乘积来拟合用户对物品的评分，然后优化两个矩阵，最终利用两个重构的低维矩阵来预测评分矩阵中的空值。

矩阵分解利用用户和物品的低维矩阵的乘积来预测用户的评分，改善了数据稀疏性问题。矩阵分解算法原理简单、易于编码实现、模型简单，通过随机梯度下降的方式不断迭代就可以训练处较好的模型。高维的评分矩阵分解为两个低维的特征矩阵不但节省了存储空间，也降低了时间复杂度和空间复杂度。虽然不断迭代的过程稍有费时，但是可以离线完成，在线计算预测评分时，可以直接使用离线训练得到的参数，以此可以完成实时的推荐任务。相比基于领域选取top N的推荐算法和基于内容的推荐算法，利用矩阵分解预测的评分具有较高的精度，同时具有很好的扩展性，很方便在两个低维特征矩阵上做一些改进研究。

推荐系统中常用的矩阵分解有非负矩阵分解(Nonnegative Matrix Factor, NMF)、奇异值分解(Singular Value Decomposition, SVD)、概率矩阵分解(PMF)等。Lee等人最早提出了非负矩阵分解算法[29]，非负矩阵分解算法要求矩阵中的所有元素均为非负数。该算法是多变量分析和线性代数的算法。非负矩阵分解实现简单，占用空间少，在大规模数据的处理上更加便捷，同时分解形式和分解的结果具有可解释性。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。

奇异值分解是一种提取信息的方法，其在推荐系统中的应用就是将用户对物品的评分矩阵进行分解，然后选择一些较大的奇异值来简约数据，去除噪声和冗余数据进行降维。这种方法简单直接，但有一个很大的问题就是要进行奇异值分解，评分矩阵必须是稠密的，即评分矩阵中不能有空值。有空白的时候评分矩阵没有办法直接去SVD分解。然而，现实世界中由于用户(user)数量和项目(item)数量都是极其庞大的所得到评分矩阵的维度也极大，但是因为用户的消费能力的有限性，对每个用户来说其消费的物品很少，产生的评分自然也极少。这样导致评分矩阵中存在大量的空值，数据稀疏性很大。传统的SVD要求矩阵中不能有空值，这就需要对评分矩阵中空值进行补全，常用的补全方法是用整个矩阵的平均值或用每个用户对物品评分的平均值补全其没有评分的空值，得到补全后的稠密评分矩阵，然后再进行矩阵分解。这里存在两个严重的问题：其一，补全空白值后大大的增加了数据量，严重增加了算法复杂度；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)其实是在New SVD的基础上添加了概率分布。其假设用户和项目的隐式特征向量都是服从高斯先验分布的。它是从概率的角度出发来学习用户和项目的特征向量，而其他的矩阵分解本质上还是在还原整个评分矩阵。

用户的特征矩阵满足均值为0，方差为的高斯分布，如式(2-1)所示。

|  |  |
| --- | --- |
|  |  |

项目的特征矩阵也满足均值为0，方差的高斯分布，如式(2-2)所示。

|  |  |
| --- | --- |
|  |  |

其中，表示变量满足均值为0，方差为的高斯分布。

假设已经观测到的评分数据的条件概率服从高斯先验分布，如式(2-3)所示。

|  |  |
| --- | --- |
|  |  |

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。

利用贝叶斯公式，用户和物品的隐式特征的后验概率如式(2-4)所示。

|  |  |
| --- | --- |
|  |  |

已知参数，，和现有的评分矩阵，可以通过求取对数和最大值得到和的隐式特征矩阵。

最后，PMF的概率模型如图2-1所示。



1. PMF矩阵分解概率模型图
   1. 卷积神经网络

卷积神经网络(CNN)，最开始应用于计算机视觉(computer vision)领域，后来CNN在机器视觉的各个方向取得了重大进展，Facebook将其应用于图像标注和自动驾驶。目前CNN已经成为计算机视觉系统的核心算法。

虽然CNN模型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(Natural language processing, 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。

Yoon Kim将卷积神经网络用于句子分类[30]，该卷积神经网络的模型结构相比用于图像处理的CNN要简单的多，只有一层网络。在卷积网络中使用了不同长度的卷积核来对文本矩阵进行卷积操作，其中卷积核的宽度设置为词向量的长度。然后用池化层对不同卷积核提取出的特征向量进行池化操作，最后每个卷积核提取的特征用一个特征向量表示，将不同的特征向量拼接起来得到了一个表示该句子的向量。Kalchbrenner等人提出了用于句子建模的卷积神经网络模型[31]，Chen Y等人提出的动态多池化卷积神经网络模型用于事件提取[32]，以发现句子层级的特征。Barzilay等人提出了一种针对文本建模的非线性非连续的卷积神经网络[33]，对卷积层进行了非线性化改造。Dong L等人提出的卷积神经网络模型针对同一输入做多重CNN分类，然后将结果进行组合[34]。Kalchbrenner提出了一种扩张的卷积神经网络(Dilated Convolution)[35]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，句子建模时以句子的字符级别的数据作为输入，之后随着卷积核所能覆盖范围的扩展，不断地交互信息，同时还能保证原始的输入信息不被丢失。文献[36]考虑到句子本身已有依存句法树信息，将其融入到了句子的建模中。文献[37]从原始数据训练CNN模型，不需要通过预训练得到word2vec或Glove等词向量表征，该CNN模型直接将one-hot编码的词向量作为输入进行卷积操作。为了减少卷积网络需要学习的参数的个数，作者对输入的词向量采用了类似词袋的表征方式以节省输入数据的存储空间。

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[38]，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

自然语言处理任务中使用的都是文本数据，而计算机目前只能识别数值型的数据，那么要处理NLP任务，首先就是要将文本中的字符转换为数值。NLP中最常用最直观的方法就是one-hot编码，每个词用一个长向量表示，向量的长度就是词汇表的大小，向量中只有一个位置的值为1，其余元素全部为0，值为1的索引代表当前词在词汇表中的位置。例如，“happy”用one-hot表示为[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 …]，则表示其出现在词汇表的第4个位置。若以这种方式存储不但浪费空间还会增加训练难度，所以一般都是采用稀疏的方法存储，比如上面的“happy”记为4(假设索引从1开始)。这种存储方式简洁易用，但是存在一个很重要的问题就是任意的两个词都是相互独立的，即使两个同义词也无法从向量中看出两个词的是否有关系，这种现象称为“词汇鸿沟”。one-hot只是一种词的表示方法，并不是Deep Learning中常用的词向量。用Distributed Representation表示的低维实数向量才是人们常用的词向量，如[0.792,-0.158,0.249,-0.315,…]。Distributed Representation是1986年由Hinton提出的，用Distributed Representation表示词，被称为“Word Embedding”，中文称为“词向量”或“词嵌入”。与one-hot表示方法相比，该方式表示的词向量，可以通过计算词向量之间的距离来判断词语之间的相似度，两个相似的词他们之间的词向量的距离也就越近，提高了向量语义上的精确度，而且解决了one-hot表示法所带来的维数灾难问题。

词向量是通过训练语言模型得到的，研究者们提出了寻多的产生词向量的算法模型。如，利用聚类算法挖掘单词与其上下文的关系的算法模型(布朗聚类)，神经网络语言模型，word2vec算法等。

* 1. 推荐系统评估方法

面对层出不穷的推荐算法，科学家们也先后提出了很多不同的方法，来帮助人们评估不同推荐系统的优劣[40]，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[41]。

1. 平均绝对值误差

平均绝对误差(mean absolute error, MAE)[42],[43]是预测评分准确度方法中最经典的一种，其定义如式(2-5)所示。

|  |  |
| --- | --- |
|  |  |

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。

后人在MAE的基础上提出了均方误差(mean squared error, MSE)如式(2-6)所示，均方根误差(root mean squared error, RMSE)如式(2-7)所示，标准平均绝对误差(normalized mean absolute error, NMAE)[44]如式(2-8)所示。

1. 均方误差

均方误差(Mean Squared Error, MSE)，用真实值与预测值的差值平方之后求平均。

|  |  |
| --- | --- |
|  |  |

1. 均方根误差

|  |  |
| --- | --- |
|  |  |

1. 标准平均绝对误差

|  |  |
| --- | --- |
|  |  |

其中，表示用户评分中的最大值，表示用户评分中的最小值。

由式(2-6)和式(2-7)可以看出MSE和RMSE将预测评分与用户真实评分的误差求了平方，所以用这两个作为评价指标会对较大的绝对误差做出更重的惩罚。相比之下，NMAE评估指标在MAE的基础上做了归一化处理，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。当用户只有两个选择时，适合用分类准确度作为评价指标。分类准确率适合在用户只有两个选择的情况下作为评价指标。准确率(precision)、召回率(recall)、F1指标是目前人们最常用的分类准确度指标。

1. 准确率

准确率在推荐系统中的定义为推荐列表中用户喜欢的产品和所有被推荐产品的比率。准确率表示对于一个推荐产品用户感兴趣的可能性。准确率的公式如式(2-9)。

|  |  |
| --- | --- |
|  |  |

其中，表示向用户u推荐的个物品的集合，T(u)表示数据集合中用户u真实评价过的物品的集合。

1. 召回率

召回率，又叫查全率。在推荐系统中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。召回率如式(2-10)，表示的是一个用户喜欢的产品被推荐的概率。

|  |  |
| --- | --- |
|  |  |

1. F1指标

准确率和召回率有时候会出现矛盾的情况，为了能更全面的评价推荐算法的好坏，必须使用准确率和召回率一起对推荐系统进行评价。Pazzan iM[45],[46]等人将二者结合起来提出了F1指标，它可以看作是准确率和召回率的一种加权平均，其最大值为1，最小值为0。F1指标定义如式(2-11)所示。

|  |  |
| --- | --- |
|  |  |

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[47]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. 卷积矩阵分解算法模型

现有的卷积神经网络不适合推荐任务，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，如图像识别，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。

卷积矩阵分解(ConvMF)正是将卷积神经网络(CNN)无缝集成到概率矩阵分解(PMF)中，而PMF通常用于推荐任务。CNN以文本文档为输入通过训练模型来优化权重矩阵，利用权重矩阵加高斯噪声来初始化项目特征向量用于PMF预测用户评分，综合模式遵循推荐目标，且有效地利用协作信息和上下文信息。即使评分数据非常稀疏，ConvMF也能准确地预测未知的评分。

### 卷积矩阵分解模型构建



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，左侧虚线框是PMF部分，右侧为卷积神经网络(CNN)部分。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，评分的条件分布如式(3-1)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，是指标函数，当用户u对物品i有评分时，其值为1，否则为0。

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，如式(3-2)。

|  |  |
| --- | --- |
|  |  |

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，2) 表示项目的文档，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，最终项目潜在模型的定义如式(3-3)和(3-4)所示。

|  |  |
| --- | --- |
|  |  |
|  |  |

对于W中的每个权重，使用最常用的零均值球面高斯先验来初始化，如式(3-5)所示。

|  |  |
| --- | --- |
|  |  |

项目的潜在模型的条件分布如下式(3-6)所示。

|  |  |
| --- | --- |
|  |  |

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自然语言处理领域有着广泛的应用。一般来说，卷积神经网络(CNN)应用于图像处理，循环神经网络(RNN)应用于自然语言处理。最近也有科学家将CNN用于自然语言处理任务中，如文本分类、语义分析、垃圾邮件检测和话题分类等。

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3-1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。

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，文档矩阵见式(3-7)。

|  |  |
| --- | --- |
|  |  |

其中，是文档的长度，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[51],[52]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式(3-8)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式(3-10)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如式(3-11)所示。

|  |  |
| --- | --- |
|  |  |

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如(3-12)所示。

|  |  |
| --- | --- |
|  |  |

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. 问题定义及改进措施

卷积神经网络(CNN)被广泛的应用于计算机视觉领域，并成为该领域的核心技术。后来，也有科学家将其用在自然语言处理(Natural Language Processing)上，并取得了引人瞩目的成绩。CNN模型在NLP任务中的典型网络结构如图3-2所示。



NLP中CNN模型的网络结构

卷积神经网络一般都是将一个矩阵作为输入的，在处理NLP任务时就需要将文本数据转换成矩阵，这个过程一般都是通过Word Embedding来处理的，处理后每个词转换成了对应的词向量，每个文本文件就转换成了一个二维的矩阵。假设输入的文本文档中包含个不同的单词，每个单词设置的词向量长度为d，那么最后输入卷积神经网络的是一个的二维矩阵。因为文档中句子长度不同，m的大小也可能不同，所以输入矩阵的大小是不确定的。其中Word Embedding的数学原理类似word2vec的数学原理，不同的是word2vec词向量一般都是通过预训练的方式得到的。CNN在处理图像时一般都会用到多通道，比如RGB(红绿蓝)通道，每个通道都可以做卷积运算，可以赋予相同或不同的权重。在自然语言处理任务中CNN也可以用到多通道，不同的是这里的多通道是指不同类的词向量的表征，例如word2vec和Glove可以看作是两个独立的通道。

卷积神经网络中的卷积操作是一个抽取特征的过程，特征抽取由特征抽取器(Filter)来完成，至于抽取多少个特征可以通过设置超参数来指定。每个特征抽取器上都可以看作有一个大小的矩阵从输入矩阵的第一个值开始不断的向后移动，每次移动的距离由步长(strides)决定。其中，是词向量的维度，是特征抽取器(Filter)指定的窗口(矩阵)的大小。在某个时刻时，窗口内的输入值通过神经网络的线性变换得到一个输出值，即该处的特征值。随着卷积核不断的往后移动，不断地产生特征值，当移动到最终的位置时，所有产生的特征值构成这个Filter的特征向量。这就是卷积神经网络中卷积的过程，对每个特征抽取器都做着同样的操作，最后形成个不同的特征抽取器。

池化层主要对卷积层后得到的特征向量进行进一步的特征提取，因为是下采样也具有一定的降维效果，提高模型训练效率。池化层后接flatten层然后是全连接层，通过全连接进一步降低维度。

卷积神经网络中最关键的两个步骤就是卷积和池化了。应用于自然语言处理的卷积神经网络一般都是设置3个大小不同的卷积核(即window size)来处理嵌入后的文本，然后针对每个卷积核提取的特征向量进行池化，即下采样操作。其中最常用池化操作就是Max Pooling Over Time了，对每个滤波器产生的向量只取最大值，作为最终代表该滤波器获得的最终特征，其它值全部抛弃。

取单个最大值的池化操作虽然可以有效降低模型的参数个数，降低模型过拟合的可能性。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：首先在文本分析中，特征出现的次数也是一个很重要的信息，比如我们常见的词频-逆文件频率(Term Frequency-Inverse Document Frequency, TF-IDF)，TF就是统计某个特征出现的次数的，越强的特征出现的次数自然也就越多。然后普通的Max Pooling不管特征出现多少次，只取一个唯一最大值，丢失了特征的强度信息。

其次，特征出现的位置也是一个很重的信息，比如一个名词可以作为主语出现在句子的开头，也可以作为宾语在句子结尾处出现，而出现位置的不同代表的意义也可能相差甚多，因此在特征提取时保留特征的位置信息将会更有助于更好的分析文本。在卷积层中局部感知野提取的特征向量中保留了特征的位置信息，然而在接下来的Max Pooling层中，对每个特征向量只取一个最大值，丢了部分重要的特征信息，同时特征之间的相对位置信息也没有保留。

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。如图3-4中，将Filter的特征向量切割成个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。对于特征向量中出现多次的一些强特征，也可以通过该方法进行捕获。

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个局部特征，那么每个局部特征向量的长度如式(3-13)所示。

|  |  |
| --- | --- |
|  |  |

其向量表示形式如式(3-14)。

|  |  |
| --- | --- |
|  |  |

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。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s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。再对模型进行优化，提高训练效率和推荐的准确性。

* 1. 词向量

词向量又称为词嵌入，是一类将词的语义信息映射到向量空间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，是词向量的维度。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某个的词语的情况下，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式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通过学习文本数据来训练模型然后用词向量的方式表征词的语义信息，通过训练 word2vec模型可以得到词向量，然后基于词向量进行聚类和词性分析将其用于自然语言处理的诸多工作中。这里得到的词向量是Distributed representation类型的词向量，并不是类似[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 …]这种one-hot representation词向量。这里代表每个词的向量是通过训练神经网络得到的，是一个维的实数向量(是模型训练前设置的超参数)。通过余弦相似度和欧式距离等算法计算词向量间的距离，根据词之间的距离来判断各个词之间在语义上的相似程度。如果用余弦距离求相似性，则得到的值越大，两个词的语义越是相似。如果用欧式距离计算词向量的距离，则得到的值越小，两个词的语义越相似。

## Word2vec词向量模型

在word2vec出现之前，已经有科学家用神经网络来训练词向量了，采用的神经网络一般都是三层结构，分别为输入(Input)层，隐藏(Hidden)层和输出层(softmax层)，如图4-1所示。



三层神经网络的语言模型

该神经网络模型存在一个很大的缺点，神经网络模型结构过于复杂，输出层上的softmax的函数，要计算所有单词的softmax概率，还要找到这些概率中最大的概率，导致计算量太大，模型训练缓慢，难以训练更多的数据。Word2vec针对这个模型进行了改进，将隐藏层(Hidden layer)改为投影层(Projection layer)，并将两个网络层之间的运算从“拼接”变成“叠加”，投影层的结点数不再是上下文单词数量和词向量维度的乘积，而是词向量的维度。Word2vec的这种网络结构使模型更加简单，训练起来也更容易，能训练更多的数据。Word2vec的两种变体分别为：连续词袋模型(Continuous Bag of Words, CBOW)如图4-2所示，Skip-gram模型如图4-3所示。CBOW模型的目标是根据上下文、、、来预测当前词语的概率。Skip-gram模型则与CBOW相反，它是根据当前词语来预测上下文、、、的概率。

|  |
| --- |
| 词袋模型(CBOW) |

Skip-gram模型

CBOW对小型的数据集处理效果较好，因为该算法在处理大部分的分布式信息时进行了平滑处理(例如，将一整段上下文信息视为一个单一的观察量)。相比之下，Skip-Gram模型作为CBOW的逆过程，在大型的数据样本集中效果更好，因为该算法将每个“上下文-目标词汇的组合”看作是一个新观察量。

原神经网络模型中Softmax计算过程过于耗费时间，word2vec在CBOW和Skip-gram两个模型的基础上，针对Sofmax的计算过程提出了两种优化方法， Hierarchical Softmax优化方法和Negative Sampling优化方法。

### Hierarchical Softmax优化

霍夫曼树利用树形结构实现了分层的softmax，即利用树形结构来代替隐藏层和输出层的神经元，霍夫曼树的叶子结点起到输出层神经元的作用，叶子结点的个数也就是词汇表的大小，而内部结点则起到隐藏神经元的作用。Hierachical softmax采用的树是二叉树。它将树上的叶子节点分配给词典里的词，而将从树根到叶子结点路径上的每个非叶子结点看作是二分类，路径上的二分类概率连乘的结果就是该叶子结点对应的词的概率。

|  |
| --- |
| 1. 霍夫曼树算法 |
| 霍夫曼树算法 | |
| 输入：权值()的个节点  输出：对应的霍夫曼树   1. 将()看作是有n棵数的深林，每个数仅有一个结点 2. 在森林中选择根节点权值最小的两棵树进行合并，得到一棵新的树，这两棵树分布作为新树的左右子树。新树的根节点权重为左右子树的根节点权重之和。 3. 将之前的根节点权重最小的两棵树从森林删除，并把新树加入森林。 4. 重复(2)和(3)直到森林里只有一棵树为止。 | |

一个全连接的softmax一次需要计算所有的个词，而Hierachical softmax只需要计算大约(即树根到该叶子节点的路径长度)个词，大大减少了计算的复杂度。

因为Huffman树对高频词会赋予更短的编码，这样高频词离根节点距离更近，计算复杂度也更低，模型训练速度加快，所以实际应用中，Hierachical softmax都是使用Huffman树，而不是其他的二叉树。

### Negative Sampling优化

与Hierachical softmax使用Huffman树不同，Negative Sampling采用的是随机负采样，每次只从词典里随机选取一些word作为当前词的负样本，而不是取所有的字典里的其它词作为负样本。随机负采样是Noise Constrastive Estimation(NCE)的一种简化版本。改进后的算法关注的不在是语言模型的求解问题，而是只关注词向量的问题。

## 4.3训练词向量

Word2vec模型的训练过程实际上是分为两部分的，第一部分是建立模型，第二部分是通过训练模型获取词向量。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，并不会用这个训练好的模型处理新的任务，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。

4.3.1数据集预处理

本节实验中使用IMDB电影评论数据集，其核心数据集包含50000条有标签的评论和50000条不带标签评论，其中带有标签的评论集又平均分为25000条训练集和25000条测试集。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，正面评论的得分高于5分。

数据集中每条评论都是以单个文件的形式进行保存，即训练集和测试集中各包含25000个txt文件。为了方便Python利用panda算法库对数据进行处理，首先合并数据并保存在tsv类型的文件中，共得到三个数据集文件，分别是25000条带标签的训练样本文件labeledTrainData.tsv，25000条带标签的测试样本文件testData.tsv以及50000条不带标签的训练样本文件unlabeledTrainData.tsv。大于5分的评分表示喜欢，小于5分的评分表示不喜欢，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文本中的文本内容作为第三列存入文件中，得到的数据集格式如表4-2所示。

其中，最左边的是行号，id代表的是文件名，文件名中“\_”前的数值代表不同电影的编号，后面的数是用户对电影的评分，sentiment代表用户的情绪，review是用户的评论。

1. 带标签数据集格式

|  |  |  |  |
| --- | --- | --- | --- |
| line-number | 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 … |

分析训练集评论数据，发现在评论中有一些字符对处理文本生成词向量没有多大的帮助，所以开始训练词向量之前需要进行数据清洗。首先利用BeautifulSoup工具包删除html标签。用正则表达式来删除标点和数字。然将单词全部转换为小写，将评论语句分割成独立的单词，去除重复的单词构建词汇表。最后，删除了一些经常出现但没什么用的词语，也就是停用词，比如“a”、“an”、“the”等。

## 词向量评估

用Distributed Representation表示词的低维实数向量，可以通过就余弦距离来判断词的相似性，可以用此方法来评估生成的词向量。向量之间余弦值越大，表示两个词的向量距离越近，词义越相近，即余弦相似度和词义相似度成正比关系。寻找与“man”最相似的词，得表4-3的结果，寻找与“awful”相似的词得到表4-4的结果。

1. man的语义相关词

|  |  |
| --- | --- |
| 单词 | 相似度 |
| woman | 0.6256189346313477 |
| lady | 0.5953349471092224 |
| lad | 0.576863169670105 |
| person | 0.5407935380935669 |
| farmer | 0.5382746458053589 |
| chap | 0.536788821220398 |
| soldier | 0.5292650461196899 |
| men | 0.5261573791503906 |
| monk | 0.5237958431243896 |
| guy | 0.5213091373443604 |

1. awful的语义相关词

|  |  |
| --- | --- |
| 单词 | 相似度 |
| terrible | 0.7551683187484741 |
| atrocious | 0.7340768575668335 |
| horrible | 0.7315883040428162 |
| dreadful | 0.7080680131912231 |
| abysmal | 0.7010548114776611 |
| horrendous | 0.6951696872711182 |
| appalling | 0.5292650461196899 |
| horrid | 0.6708598136901855 |
| amateurish | 0.6481891870498657 |
| embarrassing | 0.6306308507919312 |

从表4-3和表4-4的实验结果可以看出，训练出来的词向量能够区分不同的单词。

## 模型优化

卷积矩阵分解(ConvMF)模型中CNN模块的输出层(output layer)使用传统的tanh产生文档潜在矢量。在各种激活函数中tanh是sigmoid函数的一个变种，如公式(4-1)所示。



该激活函数当神经元的激活值在接近0或1时会饱和导致梯度消失，梯度消失会使模型收敛速度减慢，无法达到最优值。为了解决该问题，本文中使用了Maxout激活函数，其拟合能力非常强，它可以拟合任意的凸函数。Maxout函数具有计算简单，不容易饱和的优点，相比ReLU激活函数不容易出现模型崩溃的问题。

为了优化用户潜在模型和项目潜在模型，CNN权重和偏差等变量，本文中使用了最大后验(MAP)估计，如公式(4-2)所示。



由公式(4-2)进一步优化可以得到如式(4-3)的结果。



其中，代表的是，代表的是，代表的是。

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(比如当要估计的是一个维的向量的时候，坐标的方向则有个，分别是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,…,的顺序)对每一个参数进行更新(更新第个参数的时候，其它个参数保持不变)，直到收敛。

1. 坐标下降算法

|  |
| --- |
| 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。 |

坐标下降的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。应用于卷积矩阵分解模型就是假设和(或)是一个常量，公式(4-3)就变成了(或)的一个二次函数。然后，通过简单地将优化函数相对于(或)进行微分，以封闭形式分析计算特征向量(或)的最优解，如公式(4-4)和公式(4-5)所示。





其中，是具有的对角矩阵，作为其对角元素，是用户的向量。对于项目，和的定义和和类似。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，权重矩阵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，当和临时恒定时，可以解释为具有正则项的平方误差函数，如公式(4-6)所示。



为了优化，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。





训练卷积神经网络的过程中，使用了基于批量的RMSProp算法。它是从AdaGrad上优化得来的，也是一个自适应算法。

AdaGrad中的历史梯度如式所示。

RMSProp算法的累积平方梯度如式所示。



与公式(4-10)相比，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获取的多少。

|  |
| --- |
| 1. RMSProp算法 |
| 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 | |

神经网络都是非凸条件下的，使用RMSProp算法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。

### 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。在更新时计算文档潜在矢量。用于更新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。

## 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，并通过计算词向量的余弦距离验证了生成的词向量的有效性。最后对改进的卷积矩阵分解算法进一研究，对算法模型的训练过程进行优化。

1. 实验及结果分析
2. 1. 实验环境及软件

### 实验环境

本文所用实验环境：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。

### 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等算法库。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准确度的有效性，使用MovieLens 100k的数据集。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。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。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。评分区间为1-5。每一行数据格式为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。



## 实验步骤

### 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MoviLens100k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[49],[50]，本文对所有数据集的预处理描述如下：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。

### 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。针对卷积矩阵分解模型，本文对设置：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。

### 实验参数选取

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，本章用网格搜索算法来训练不同参数下的算法模型。

网格搜索是一种穷举方法，通过估计参数可能的取值然后利用交叉验证来得到最优参数的一种搜索算法。即，列出各个参数可能取到的值，对每个参数的取值进行排列组合，得到所有的组合结果。然后分别用每个组合训练模型，并利用交叉验证对该组参数的训练结果进行评估。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。

交叉验证算法中主要k折交叉验证(k-folder cross-validation)和留一法(leave-one-out cross-validation)这两种算法。其中，k折交叉验证就是将数据集划分为个子集，而这个子集要求大小相似内容互斥，如式所示。

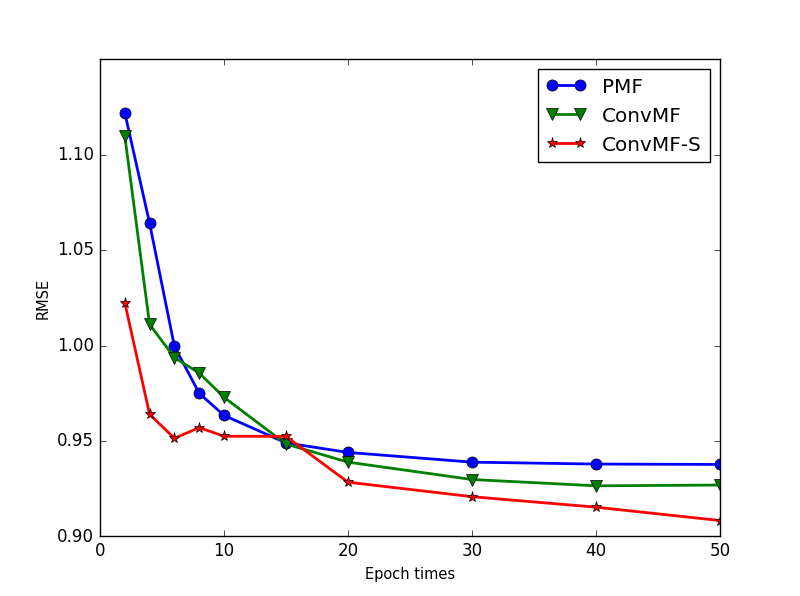


每次从个子集中选取一个数据集作为训练用的测试集，然后将剩余的-1个数据集进行合并作为训练集。其中，从1到分别取值，这样就得到了个不同的测试集和个不同的训练集，在不同的数据集上做训练，然后对次测试的结果求平均值作为最后的结果。留一法假设数据集中包含个样本，另折交叉验证法中的=。本文中使用的折交叉验证中的=10。

* 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，实验结果如图5-1所示。

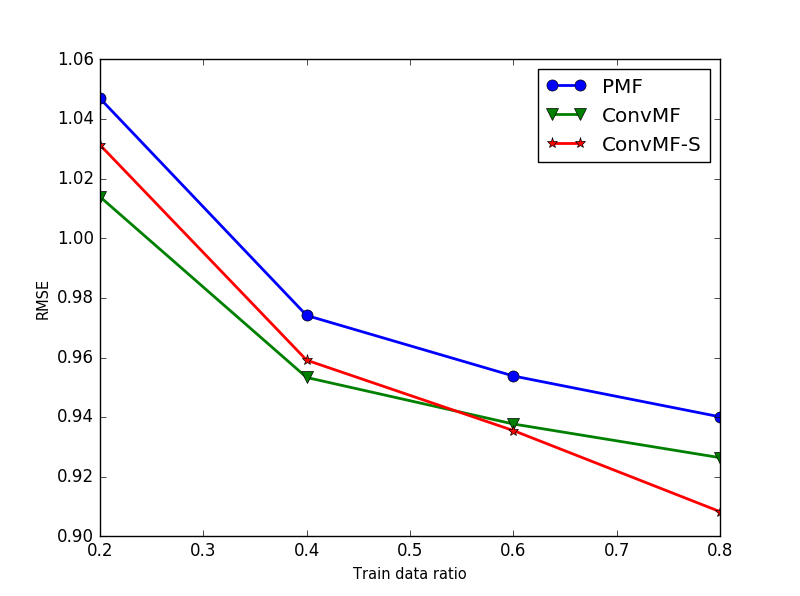


1. 三种推荐算法在ML-100K数据集上实验结果对比

由图5-1的折线图可以看出对于MovieLens100K数据集，概率矩阵分解算法(PMF)在第0到第15次的迭代过程中收敛速度很快，迭代15次后RMSE值趋于稳定，模型不在收敛。卷积矩阵分解算法(ConvMF)在前20次的迭代过程中收敛速度很快，迭代20之后模型模型依然在收敛只是收敛速度减慢，迭代到30次的时候模型趋于稳定RMSE值不在变化，此时RMSE的值小于PMF得到的最优值，说明加入对项目描述文档的分析后，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。改进的算法模型(ConvMF-S)在模型训练初始就要优于PMF和ConvMF算法，说明池化层分段选取特征有效提高了CNN对文档数据的分析能力。改进的算法最终迭代结果要优于其他两种算法也进一步证明了改进的模型可以有效提高推荐质量。

### 三种算法在数据集不同比例上的对比

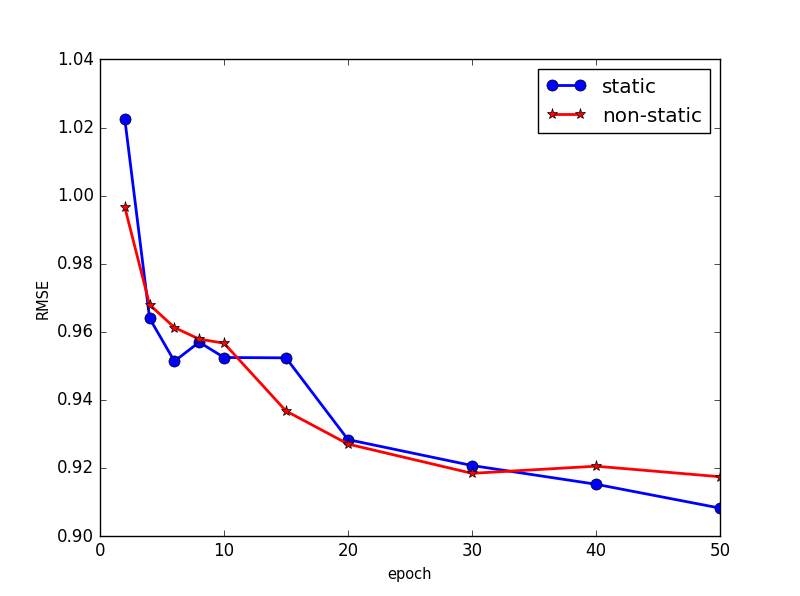
这一组实验通过对同一数据集选取不同的训练集比例来测试改进算法的推荐结果。从ML-100K数据集中按比例(20%, 40%, 60%, 80%)随机抽取数据集。然后将三个算法分别在不同比例的训练集上进行训练，实验结果如图5-2所示。

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。不同训练集比例的情况下，另两种算法的推荐效果都优于PMF算法，说明加入文档建模方法可以提高准确性。在训练集比例为20%和40%时原始的卷积矩阵分解算法比改进的算法要好，随着训练集占比的增多，改进的算法超过原始的卷积矩阵分解算法，且效果越来越好。

### 嵌入词向量对模型的影响

在Movielens100k数据集上训练未嵌入词向量的模型和嵌入词向量后的算法模型。实验结果如图5-3所示。从折线图5-3可以看出，是否嵌入词向量模型在训练初期并没有明显的差别，但是随着迭代次数的不断增加，未嵌入词向量的模型迭代至30次时已经基本稳定，加入预训练词向量的模型仍在收敛，且模型最终的结果要优于未嵌入词向量的模型。由此可见，嵌入预训练的词向量模型，可以提高卷积矩阵对文本分析的能力，提高推荐的准确性。同时，也进一步证明了第4章中所训练的词向量就有可用性。



1. 嵌入词向量对模型的影响
   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，并实现了另外两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推荐算法中的数据稀疏问题。

结 论

随着互联网的飞速发展，用户和商品数量呈爆发式增长，评分数据越来越稀疏。传统的推荐算法大部分都是挖掘评分矩阵中的用户和商品信息做出推荐的。特别是基于协同过滤的推荐算法在稀疏的数据集上做出的推荐结果质量很差。为了解决数据稀疏问题对推荐系统的影响，科学家们提出了多种方法来提高推荐的质量，其中通过利用商品的辅助信息来改善数据的稀疏性，可以有效提升模型表现。

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挖掘数据的本质特征，也可以通过从多源异构数据中进行自特征学习。将深度学习与推荐系统进行融合，可以有效利用多源异构数据，缓解数据稀疏问题。在此方向进行深入研究后发现，利用深度学习的方法来分析与商品有关的文本数据(例如评论，摘要)可以有效缓解数据稀疏问题，提高推荐质量。

本文介绍了一种基于上下文的深度推荐算法(ConvMF-S)。本文工作主要有以下几个方面：

1. 对卷积矩阵分解模型进行了深入研究，发现卷积神经网络在对文本信息进行分析时，忽视了特征的位置信息，对出现多次的强特征没有进行保留。在此基础上提出了卷积矩阵分解的改进算法(ConvMF-S)。改进的算法在提取多特征的同时保留了特征的位置信息和特征强度，加深了对文本信息的利用程度，改善了推荐结果。
2. 以IMDB数据集为基础，通过训练word2vec算法模型获得了词向量。通过实验分析证明了训练得到的词向量可以有效区分不同单词的词性。对基于上下文的深度混合推荐算法进行一步优化，提高了其运行效率。
3. 以深度学习框架为平台，实现算法并训练模型，通过从多个方面进行比较得到的实验数据，验证了改进模型的有效性。
4. 将预训练的词向量嵌入到卷积神经网路，对文本数据进行分析，然后结合推荐系统进行推荐。通过实验与为嵌入词向量的算法模型进行对比，证明了嵌入词量有助于卷积神经网络对文本数据的特征提取，同时进一步提高了推荐效果。

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. Resnick P, Iacovou N, Suchak M, et al. GroupLens:an open architecture for collaborative filtering of netnews[C]// ACM Conference on Computer Supported Cooperative Work. ACM, 1994:175-186. Xiaoyuan Su, and Taghi M. Khoshgoftaar. A Survey of Collaborative Filtering Techniques [J]. Advances in Artificial Intelligence 2009(2009):4.
2. Zhao Z D, Shang M S. User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop[C]// Third International Conference on Knowledge Discovery and Data Mining. IEEE Computer Society, 2010:478-481.
3. Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]// International Conference on World Wide Web. ACM, 2001:285-295.
4. Barragáns-Martínez A B, Costa-Montenegro E, Burguillo J C, et al. A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition[J]. Information Sciences, 2010, 180(22):4290-4311.
5. Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]// Proc. Conference of Uncertainty in Articial Intelligence. 1998:43-52.
6. Srebro N, Jaakkola T S. Weighted low-rank approximations[C]. international conference on machine learning, 2003: 720-727.
7. Lee D D, Seung H S. Algorithms for non-negative matrix factorization[C]// International Conference on Neural Information Processing Systems. MIT Press, 2000:535-541.
8. Lee J, Kim S, Lebanon G, et al. Local low-rank matrix approximation[C]// International Conference on Machine Learning. 2013:82-90.
9. Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.
10. Dan C C, Giusti A, Gambardella L M, et al. Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images[J]. Advances in Neural Information Processing Systems, 2012, 25:2852--2860.
11. Le Q V. Building high-level features using large scale unsupervised learning[J]. 2011:8595-8598.
12. Abdel-Hamid O, Mohamed A R, Jiang H, et al. Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2012:4277-4280.
13. Abdel-Hamid O, Li D, Dong Y. Exploring Convolutional Neural Network Structures and Optimization Techniques for Speech Recognition[C]// INTERSPEECH. 2013:1173-5.
14. Sainath T N, Mohamed A R, Kingsbury B, et al. Deep convolutional neural networks for LVCSR[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013:8614-8618.
15. Deng L, Abdel-Hamid O, Yu D. A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013:6669-6673.
16. Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering[C]// International Conference on Machine Learning. ACM, 2007:791-798.
17. Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2011:448-456.
18. Ling G, Lyu M R, King I. Ratings meet reviews, a combined approach to recommend[J]. 2014:105-112.
19. 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.
20. Salakhutdinov R, Mnih A. Probabilistic Matrix Factorization[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2007:1257-1264.
21. Wang H, Wang N, Yeung D Y. Collaborative Deep Learning for Recommender Systems[J]. 2014:1235-1244.
22. Aggarwal C C, Gates S C, Yu S L. System and method for generating taxonomies with applications to content-based recommendations: US, US 6360227 B1[P]. 2002.
23. Pazzani M J, Billsus D. Content-based recommendation systems[M]// The adaptive web. Springer-Verlag, 2007:325-341.
24. Robin Burke. Knowledge-Based Recommender Systems. Encyclopedia of Library and Information Science (ELIS), 69(32), 2000.
25. Korfiatis N, Poulos M. Using online consumer reviews as a source for demographic recommendations: A case study using online travel reviews[J]. Expert Systems with Applications, 2013, 40(14):5507-5515.
26. Burke R. Hybrid Recommender Systems: Survey and Experiments[J]. User Modeling and User-Adapted Interaction, 2002, 12(4):331-370.
27. Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]// Proc. Conference of Uncertainty in Articial Intelligence. 1998:43-52.
28. Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 1999, 401(6755): 788-791.
29. Kim Y. Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.
30. Kalchbrenner N, Grefenstette E, Blunsom P. A Convolutional Neural Network for Modelling Sentences[J]. Eprint Arxiv, 2014, 1.
31. 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.
32. 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.
33. 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.
34. Kalchbrenner N, Espeholt L, Simonyan K, et al. Neural Machine Translation in Linear Time[J]. 2016.
35. Ma M, Huang L, Xiang B, et al. Dependency-based Convolutional Neural Networks for Sentence Embedding[J]. 2015:174-179.
36. Johnson R, Zhang T. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks[J]. Eprint Arxiv, 2014.
37. Dieleman S, Schrauwen B. Deep content-based music recommendation[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2013:2643-2651.
38. Herlocker J L. Evaluating collaborative filtering recommender systems[J]. Acm Transactions on Information Systems, 2004, 22(1):5-53.
39. 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.
40. Gunawardana A, Shani G. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks[M]. JMLR.org, 2009, 10: 2935-2962.
41. Shardanand U, Maes P. Social information filtering: algorithms for automating “word of mouth”[C]// CHI'95 Conference Proceedings. 1995:210-217.
42. Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]// Proc. Conference of Uncertainty in Articial Intelligence. 1998:43-52.
43. Balabanović M, Shoham Y. Fab: content-based, collaborative recommendation[J]. Communication of the Acm, 1997, 40(3):66-72.
44. P Pazzani, Michael, Billsus, et al. Learning and Revising User Profiles: The Identification ofInteresting Web Sites[J]. Machine Learning, 1997, 27(3):313-331.
45. VAN RIJSBERGEN C J. Information retrieval[M]. MA, USA: Butterworth- Heinemann Newton, 1979.
46. Kim D, Park C, Oh J, et al. Deep Hybrid Recommender Systems via Exploiting Document Context and Statistics of Items ☆[J]. Information Sciences, 2017, 417.
47. Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation[C]// Conference on Empirical Methods in Natural Language Processing. 2014:1532-1543.
48. Wang C, Blei D M. Collaborative topic modeling for recommending scientific articles[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2011:448-456.
49. Wang H, Wang N, Yeung D Y. Collaborative Deep Learning for Recommender Systems[J]. 2014:1235-1244.
50. Kim Y. Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014:1746–1751.
51. Collobert R, Weston J, Bottou L, et al. Natural Language Processing (Almost) from Scratch[J]. Journal of Machine Learning Research, 2011: 2493-2537.

攻读硕士学位期间承担的科研任务与主要成果