

**硕 士 学 位 论 文**

MASTER’S DISSERTATION

论文题目

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

关键词：深度学习；推荐系统；协同深度学习；Spark集群

# Abstract

Collaborative Deep Learning relieves the problem that Conventional CF-based methods use the ratings given to items by users as the sole source of information for learning to make recommendation, while CF-based methods to degrade significantly in their recommendation performance because of the ratings are very sparse. CDL use the ability of Neural network that learn the features from data and to fit a robust model automatically. It introduces a new way to solve the problems that the recommendation system meets the very sparse data. Meanwhile, Recommendation System can not hold the Big Data because of the Rating Matrix is too large to load in the memory and the process of training for the content model is hard to predict. As a result, the Distribute Platform for RS is more and more necessary.

To solve the problem mentioned above, We propose in this paper a model based on CDL called "CDL with Item private Node"(CDL-i), And train CDL-i with a parallel method on Apache Spark. When training the CDL-i model, more than one dataset was used. we transfer CDL-i to Spark to investigate it's scalability.

First of this paper, we introduce the theory of the Parallelized CDL-i, including CDL, Autoencoder, PMF, state of parallelized machine learning and Apache Spark.

Then, we develop a modified collaborative deep learning CDL-i. Meanwhile, we introduce a method to show that how to transfer CDL-i to Spark. By the way, a framework based Spark for training machine learning model with pipeline Logic was developed.

At last, extensive experiments on real-world datasets show that our modified CDL-i can improve the performance of original CDL. And Parallel experiments can tell us the parallelized CDL was scalable and effective. As the result, CDL-i can perform a better precision, and has the ability to hold the data which has a huge size. The framework helps the algorithm easy to use.

Keywords: Deep Learning; Recommendation; CDL; Spark Cluster

目 录

[摘 要 I](#_Toc482799492)

[Abstract II](#_Toc482799493)

[第1章 绪 论 1](#_Toc482799494)

[1.1 课题背景及研究意义 1](#_Toc482799495)

[1.2 国内外研究现状 2](#_Toc482799496)

[1.3 现有算法存在的问题 4](#_Toc482799497)

[1.4 论文研究内容 5](#_Toc482799498)

[1.5 论文的组织结构 6](#_Toc482799499)

[第2章 协同深度推荐相关理论研究 7](#_Toc482799500)

[2.1 协同深度推荐 7](#_Toc482799502)

[2.1.1 自动编码机 7](#_Toc482799503)

[2.1.2 矩阵分解 10](#_Toc482799504)

[2.2 机器学习的并行加速 11](#_Toc482799505)

[2.3 基于Spark的分布式计算框架 12](#_Toc482799506)

[2.3.1 Spark分布式平台 13](#_Toc482799507)

[2.3.2 Spark MLLib 14](#_Toc482799508)

[2.3.3 Scala Reflection 15](#_Toc482799509)

[2.4 本章小结 16](#_Toc482799510)

[第3章 协同深度推荐的模型改进 17](#_Toc482799511)

[3.1 引言 17](#_Toc482799513)

[3.2 问题定义及模型改进 17](#_Toc482799514)

[3.2.1 自动编码机的概率图模型 18](#_Toc482799515)

[3.2.2 概率矩阵分解的概率图模型 20](#_Toc482799516)

[3.2.3 CDL-i的概率图模型 21](#_Toc482799517)

[3.3 CDL-i算法描述 22](#_Toc482799518)

[3.4 CDL-i实验分析 24](#_Toc482799519)

[3.4.1 实验环境 24](#_Toc482799520)

[3.4.2 数据集描述 24](#_Toc482799521)

[3.4.3 实验结果分析 25](#_Toc482799522)

[3.5 本章小结 33](#_Toc482799523)

[第4章 协同深度推荐模型的并行化 34](#_Toc482799524)

[4.1 引言 34](#_Toc482799526)

[4.2 数据并行 34](#_Toc482799527)

[4.3 模型并行 35](#_Toc482799528)

[4.4 协同深度推荐(CDL-i)并行 36](#_Toc482799529)

[4.5 并行化CDL-i的实验分析 40](#_Toc482799530)

[4.5.1 实验环境 40](#_Toc482799531)

[4.5.2 数据准备 40](#_Toc482799532)

[4.5.3 单机试验结果分析 40](#_Toc482799533)

[4.5.4 并行试验结果分析 41](#_Toc482799534)

[4.6 本章小结 42](#_Toc482799535)

[第5章 基于Spark的分布式机器学习框架 43](#_Toc482799536)

[5.1 引言 43](#_Toc482799538)

[5.2 框架建立动机 43](#_Toc482799539)

[5.3 框架算法简述 44](#_Toc482799540)

[5.4 数据处理与算法支持逻辑简述 46](#_Toc482799541)

[5.5 框架扩展性介绍 49](#_Toc482799542)

[5.6 Spark机器学习框架下协同深度学习推荐算法 50](#_Toc482799543)

[5.7 本章小结 52](#_Toc482799544)

[结 论 53](#_Toc482799545)

[参考文献 55](#_Toc482799546)

[攻读硕士学位期间承担的科研任务与主要成果 59](#_Toc482799547)

[致 谢 60](#_Toc482799548)

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，有效的解决了“信息过载”问题。目前，不同类型的推荐系统已经应用于许多领域，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其感兴趣的商品；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，主要包括协调过滤、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使用了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。很快深度学习成为机器学习领域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，深度学习可以通过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，深度学习通过从多远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的信息和知识来源，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[Resnick 1994, Su 2009]寻找与目标用户有相似兴趣的用户群体，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可以分为三个子类：基于用户的推荐(User-based CF Recommendation)，基于商品的推荐(Item-based CF Recommendation)和基于模型的推荐(Model-based CF Recommendation)。

基于用户的协同过滤算法[Zhao 2010]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-紧邻算法)，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算法[Sarwar 2001, Barragáns-Martínez 2010]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算法[Breese 1998]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中最热门的一种。矩阵分解方法通过用户-项目评分矩阵分解成一个用户相关的低秩矩阵(用户特征矩阵)和一个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，同时对没有评分的用户-项目进行预测。一些改进工作已经被应用到该方法上，如带权重的矩阵分解方法[Nathan 2003]、基于非负矩阵的矩阵分解方法[Lee 2011]，和基于矩阵局部性的矩阵分解方法[Joonseok 2013]等，这些方法都展示了其在推荐系统上的效果。

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。深度学习是机器学习近年来关注度非常热的研究领域，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。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，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[LeCun 1998; Ciresan 2012; Le 2012]应用中取得非常好的结果。近年来，CNN在自然语言处理方向也取得了很好地发展[Abdel-Hamid 2012,2013; Sainath 2013; Deng 2013]

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 2007]提出了基于玻尔兹曼机的协同过滤算法，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方法(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。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。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)，从而在评分预测精度方面产生更准确的潜在模型，但是基于深度学习的推荐算法研究还非常少，不够系统。

## 现有算法存在的问题

目前，对推荐算法的研究越来越广泛。根据推荐算法所使用的信息和知识来源，推荐算法被分成多种类别。主流的推荐算法主要被分成下面五个类别：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。

1. 基于内容的推荐算法[Aggarwal 2002]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，能够找打最适合用户的推荐[Pazzani 2007]。这里建立用户画像的方法，可以使用基于记忆(启发式)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对用户的兴趣可以很好的建模，并通过对物品属性维度的增加，获得更好的推荐精度。它的局限性在于，用户无法得到商品推荐直到他给足够的商品进行评分，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。
2. 协同过滤推荐算法[Resnick 1994, Su 2009]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盒准确性。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。
3. 基于知识的推荐算法[Burke 2000]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。它不需要的样本用户，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商品知识库。
4. 基于人口统计学的推荐算法[Korfiatis 2013]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。
5. 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。现在往往将多个方法混合在一起，从而达到更好的推荐效果[Burke 2002]。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。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，具体权重值在测试集上反复试验，从而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

本文基于协同深度学习算法[[16](#_ENREF_16)]3其模型进行改进，提出了模型CDL-i，并进行了协同深度学习算法的并行化研究，依托Apache Spark分布式计算平台，将协同深度学习算法并行化。提高了推荐系统对数据信息的使用程度，同时提高了推荐系统在大数据时代下的可扩展性。

CDL通过联合自动编码机对项目内容的深度表示学习和协同过滤评分矩阵的反馈信息，构建了一种多层的贝叶斯模型。本文在CDL基础上针对其项目内容探测较为粗犷的不足进行了改进，提出CDL-i模型。并在真实数据集上结合多组实验证明相对于传统推荐，CDL-i带来的提升。随后本文介绍了算法的并行化改进及移植到Spark平台下的成果，同时实现了一套基于Spark分布式平台的Pipeline学习框架。

本文贡献可简要概括为：

1. 对协同深度学习进行改进，基于CDL算法提出了CDL-i模型。
2. 依托分布式计算平台Spark，对算法进行并行化改进，赋予其较强扩展性。
3. 实现并完善一套基于Spark的Pipeline分布式机器学习框架，简化机器学习工作流程。

## 论文的组织结构

本文对协同深度学习的并行化进行了研究，主要内容有：

第2章介绍课题相关理论，包括协同深度学习(CDL)、自动编码机、概率矩阵分解；机器学习算法模型并行化现状与思想；介绍了基于Spark的分布式并行技术。

第3章详细分析CDL算法，并对多种模式下的算法模型进行探索与实验对比。介绍了基于CDL的改进模型CDL-i。并通过在真实数据集上的多组实验证明改进所带来的提升。

第4章对CDL-i算法进行了并行化研究与改进，详细描述其移植到Spark分布式计算框架下实现并行训练的思想。通过对数据一系列处理，验证海量数据下并行化CDL的可扩展性，并与单机性能做了对比分析。

第5章，基于前文的理论基础与算法实现，构建了一种基于Spark平台的Pipeline分布式机器学习框架，并行版本的CDL-i算法作为其中一个算法模块被融入其中，简化了算法的应用流程。

最后作出结论与展望。CDL在推荐准确率上较大优势，改进CDL-i也为推荐性能带来提升。将算法并行化后，模型在海量数据情景下有较强扩展能力。对机器学习流程进行抽象与统一，实现的Pipeline框架提升了算法的易用性。但算法仍有改进空间，框架实现仍有些许不足。

1. 协同深度推荐相关理论研究

## 协同过滤推荐算法

推荐算法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。协同过滤技术可以分成基于邻域的协同过滤方法和基于模型的协同过滤方法两种。而在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方法中最热门的一种。矩阵分解方法通过和用户-商品评分矩阵分解成一个用户相关的低秩矩阵(用户特征矩阵)和一个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-商品对进行预测。

基于矩阵分解的方法中，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，好的用户商品特征表示能带来更好的预测结果。

推荐算法具有非常多的应用场景和商业价值，因此对推荐算法值得好好研究。推荐算法种类很多，但目前应用最广泛的应该是协同过滤类别的推荐算法。

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，只要找出相似用户喜欢的物品，并预测目标用户对对应物品的评分，就可以找到评分最高的若干个物品推荐给用户。而基于项目(item-based)的协同过滤和基于用户的协同过滤类似，只不过这时我们转向找到物品和物品之间的相似度，只有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。比如你网上买了一本机器学习的书，网站上马上就会推荐一堆机器学习，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。

基于用户的协同过滤需要在线找用户和用户之间的相似度关系，计算复杂度肯定会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，由于考虑的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，但是推荐的多样性来说，就很难带给用户惊喜了。一般对于小型的推荐系统来说，基于项目的协同过滤肯定是主流。但是如果是大型的推荐系统来说，则可以考虑基于用户的协同过滤，当然更加可以考虑我们的第三种类型，基于模型的协同过滤。

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。用矩阵分解做协同过滤是目前使用也很广泛的一种方法。由于传统的奇异值分解SVD要求矩阵不能有缺失数据，必须是稠密的，而我们的用户物品评分矩阵是一个很典型的稀疏矩阵，直接使用传统的SVD到协同过滤是比较复杂的。

协同过滤推荐算法[Resnick 1994, Su 2009]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。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可以分成三个子类：基于用户的推荐(User-based CF Recommendation)，基于商品的推荐(Item-based CF Recommendation)和基于模型的推荐(Model-based CF Recommendation)。

基于用户的协同过滤推荐算法[Zhao 2010]的基本原理是，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算法[Sarwar 2001, Barragáns-Martínez 2010]的基本原理与此类似，它使用所有用户对商品或者信息的偏好，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算法[Breese 1998]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。

1.简单介绍协同过滤推荐算法

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，而且也不要求商品的描述是机器可理解的，因此这种方法是领域无关的，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。

协同过滤算法是目前最普遍使用的独立算法之一。然而，基于矩阵分解的协同过滤存在数据稀疏性和冷启动等问题，在学习用户和商品特征方面，存在一定的局限性。

2.引出基于矩阵分解PMF的推荐，并详细介绍

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。

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。矩阵分解的核心思想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，Item评分信息学习到的用户特征矩阵P和物品特征矩阵Q，通过重构的低维矩阵预测用户对产品的评分。

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。

矩阵分解方法将高维User-Item评分矩阵映射为两个低维用户和物品矩阵，解决了数据稀疏性问题。使用矩阵分解具有以下优点：

1. 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，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。

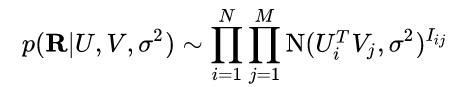
1. 预测的精度比较高，预测准确率要高于基于领域的协同过滤以及内容过滤等方法。
2. 非常好的扩展性，很方便在用户特征向量和物品特征向量中添加其它因素

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的的用户。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。

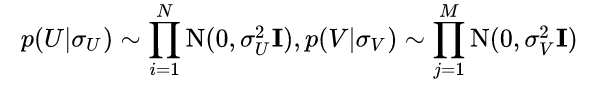
假设有N个用户和M个物品，那么就形成了一个N x M评分矩阵R，通常R非常稀疏，只有不到1%的元素是已知的，而我们要估计出缺失元素的值。PMF假设评分矩阵中的元素Ri,j是由用户的潜在偏好向量Ui和物品的潜在属性向量Vj的内积决定的，即：



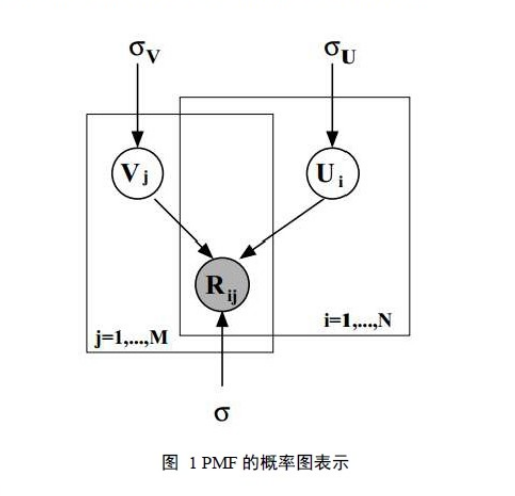
其中N表示正态分布。则观测到的评分矩阵条件概率为：



Ii,j是指示函数，若观测到Ri,j则其值为1，否则为0。再假设用户偏好向量和物品偏好向量也都服从正态分布，即：



最后，下图用概率图模型表示PMF：



## 卷积神经网络

1. 介绍神经网络的基本知识

当我们听到CNN时，一般会想到计算机视觉(computer vision)。CNN在图像分类中取得了重大的突破，也是从Facebook的图像标注到自动驾驶等，当今计算机视觉系统的核心。

虽然CNN最初是为计算机视觉开发的，但是CNN的关键思想已经积极应用于信息检索和自然语言处理(NLP)，如搜索查询检索，句子建模和分类和其他传统的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的词的含义。在语言模型里n-gram模型将连续的两个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，实际训练时的参数时一个非常大的问题，因为假设你有20000个词，加入bi-gram实际上你就要有400000000个词，这样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例如参数权重等，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句法差异 - 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），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。为了解决上述问题，我们采用卷积神经网络（CNN），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，作为更简洁表示的池化（或子采样）层，局部特征来自上一层，其通常是卷积层。

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，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用。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，但总是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。在以句子分类中，每一种卷积核可以用来检测某一种含义的词组，如果这种类型的含义的词语出现了，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。

1. 介绍卷积神经网络在推荐中应用(这部分可以参考原论文)

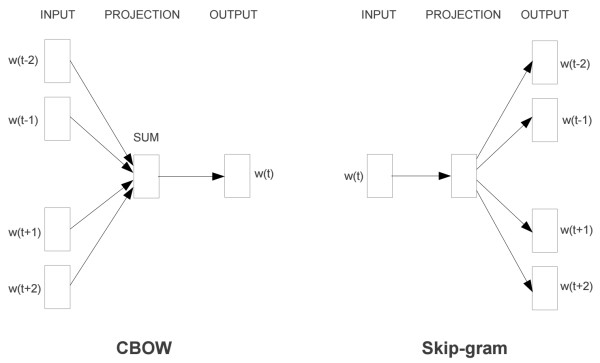
然而，CNN尚未被积极应用于推荐系统领域。据我们所知，van den Oord等人首先将CNN 应用于音乐推荐，通过声学分析观点利用CNN分析了歌曲，并提出了一种基于声学CNN获得的项目潜在模型来预测评分模型。然而，他们的CNN模型，专为声信号处理而设计，不适合处理文档。文档和声学信号对周围特征的质量有固有的差异。一定时间的信号固有地类似于其周围的信号，即具有轻微时间差的信号，而文档中某个位置处的一个字与周围的字有很大的语义差异。周围特征之间的这种相似度的差异影响局部特征的质量，最终需要不同的CNN架构。此外，该模型还没有充分反映协同信息。具体来说，项目潜在模型主要由通过CNN进行音频信号分析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），WMF这是处理隐式反馈数据集的常规基于MF的协同过滤之一。

## Word embedding

Word embedding的训练方法大致可以分为两类：一类是无监督或弱监督的预训练；一类是端对端（end to end）的有监督训练。无监督或弱监督的预训练以word2vec和auto-encoder为代表。

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用词向量的方式表征词的语义信息，即通过一个嵌入空间使得语义上相似的单词在该空间内距离很近。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，也就是把原先词所在空间嵌入到一个新的空间中去。

Word2Vec模型中，主要有Skip-Gram和CBOW两种模型，从直观上理解，Skip-Gram是给定input word来预测上下文。而CBOW是给定上下文，来预测input word。

Word2Vec模型实际上分为了两部分，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)的思想和相似，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中实际上就是我们试图去学习的”word vectors”。

Word2vec和auto-encoder这类模型的特点是，不需要大量的人工标记样本就可以得到质量不错的embedding向量。不过因为缺少了任务导向，可能和我们要解决的问题还有一定的距离。因此，人们往往会在得到预训练的embedding向量后，用少量人工标注的样本去fine-tune整个模型。

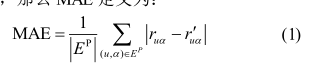
相比之下，端对端的有监督模型在最近几年里越来越受人们的关注。与无监督模型相比，端对端的模型在结构上往往更加复杂。同时，也因为有着明确的任务导向，端对端模型学习到的embedding向量往往更加准确。例如，通过一个embedding层和若干个卷积层连接而成的深度神经网络以实现对句子的情感分类，可以学习到语义更丰富的词向量表达。

## 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。[24-26]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：

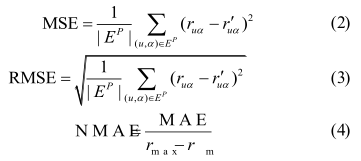
1. 预测评分的准确度

顾明思议，这类方法评价的是推荐算法所预测的用户可能评分和用户的实际评分的贴近程度。预测评分的准确度的方法目前也有很多种，其中最经典的方法是平均绝对值误差(mean absolute error,MAE)[30-32]，其定义为：

其中rua表示用户u对商品a的真实评分，rua’表示用户u对商品a的预测评分，Ep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。

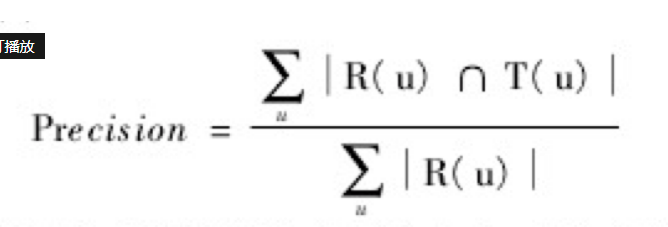
此外，平均平方误差(mean squared error,MSE)、均方根误差(root mean squared error, RMSE)以及标准平均绝对误差【15】(normalized meand absolute error,NMAE)都是和平均绝对误差类似的方法。它们分别定义为



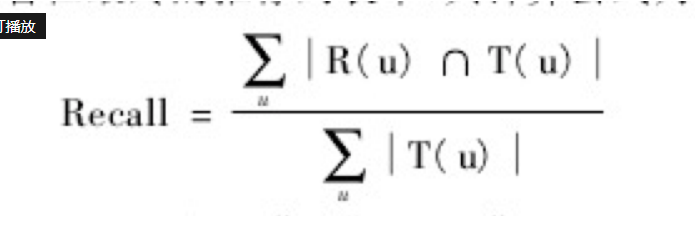
(4)式中的rmax和rmin分别为用户评分区间的最大值和最小值。由于MSE和RMSE对预测值和真实值的绝对误差做了平方，所以它们对比较大的绝对误差有更重的处罚。NMAE对结果在评分区间上做了归一化，所以可以用来评价同一推荐算法在不同数据集上的表现。

1. 分类准确度

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。

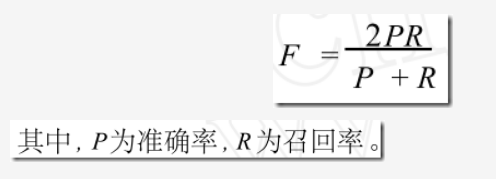
文献【38-41】最先把准确率和召回率用在了推荐系统的评价中。准确率定义为系统的推荐列表中用户喜欢的产品和所有被推荐产品的比率。准确率表示用户对一个被推荐产品感兴趣的可能性。假设R(u)表示向用户u推荐的N个物品的集合，T(u)表示数据集合中用户u真实评价过的物品的集合，则准确率的公式表示如下：

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。



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[44-45]等把二者综合考虑提出了F1指标。F1指标定义为：



## 本章小结

1. 协同深度推荐的模型改进

## 引言

深度学习的蓬勃发展向人们展示了其强大的特征学习能力，在推荐系统中的应用逐渐引起人们的关注。利用深度学习学到的项目内容特征作为推荐系统的辅助信息，提高推荐算法的性能已经有了许多成果，而神经网络本身作为推荐系统的基础算法直接应用于推荐相关研究却很少。

协同深度学习基于神经网络，同时利用神经网络的特征学习能力和神经网络能够拟合任意复杂数据分布的能力，联合对数据进行分析和学习得到推荐结果的模型。CDL可以自动的提取项目内容的高度抽象特征，并且捕获项目或者用户间隐含关系和相似性。CDL首次将深度学习模型与推荐系统利用统一的贝叶斯视角整合在一起，使得学习到的项目的附加信息可以很容易的被解释。

受启发于CDAE[[37](#_ENREF_37)]，本文在原CDL的基础上对SDAE进行改进，提出了CDL-i(CDL with Item private Node, CDL-i)，CDL-i在CDL基础上在SDAE的输入层为每个项目添加私有的偏置单元，在模型的网络参数共享情况下，为每个项目添加属于自己的私有偏差项，进而能够使网络更加针对性地学习到项目内容参数。

## 问题定义及模型改进

类似于[[16](#_ENREF_16)]中使用训练数据的隐式反馈提升推荐系统的性能，本文定义全量数据集包含个项目，用维矩阵表示。中的第行表示对应项目的内容向量，用维向量表示。对于个用户，对个项目的评分可用维度的评分矩阵表示。当第个用户对第个项目有正向反馈时，令；否则为0。

自动编码机中、、分别表示加入噪声的输入、原始数据输入和自动编机第层网络的输出。、为维矩阵，为维度矩阵。类似于，的第行表示为。和分别为第层网络的权值矩阵和偏置向量。表示权值矩阵的第列。神经网络的总层数为。是权值矩阵和偏置向量的简称。

本文在原CDL的基础上提出了拥有项目私有偏置结点的模型CDL-i(CDL with Item private Node, CDL-i)，CDL-i对SDAE的输入层为每个项目添加特有的偏置单元，在模型的网络参数共享情况下，为每个项目添加属于独有的偏置量，使网络能够更针对性地学习到不同项目的内容参数。

CDL-i的训练与学习使用了类似CDL的算法，需要从数据集中分别对个项目进行学习，此过程中自动编码机对内容进行学习和特征提取。将得到的内容向量与偏置向量整合后作为推荐系统的项目信息。利用矩阵分解进行评分预测，通过对计算评分进行排序，按分数由高到低进行推荐。此外，模型训练过程中，对数据进行正样本抽样，使得正负样本间距加大，提高模型表现，通过推荐Top-N结果对模型进行评估。

### 自动编码机的概率图模型

更一般的，类比于文献[[38-40](#_ENREF_38)]，我们可以利用以下步骤定义广义堆叠降噪自动编码机(Generalized Bayesian SDAE)：

1、对于SDAE的每一层网络：

1. 记当前列为*n*，对每一列的权值矩阵进行服从如式(3-1)分布的采样。

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

1. 对第*l*层偏差向量，采样使之服从如式(3-2)的先验分布。

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

1. 对第*l*层数据矩阵的每行数据*j*，采样使之服从形如式(3-3)的分布。

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

2、对于每一条输入数据，利用对式(3-4)的抽样获得其初始数据。

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

为对自动编码机进行求解，本文选择反向传播(Backpropgation, BP)来进行学习和优化，此外目标函数学习使用了梯度下降(Gradient Descent)思想。另外，为提高算法运行效率，我们采用随机梯度下降算法(Stochastic Gradient Descent, SGD)，即计算每一条样本，实时更新学习模型。

自动编码机模型可具体表示为正向传播、反向传播以及权值更新。

正向传播操作如式(3-5)和(3-6)所示。

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

反向传播如式(3-7)、式(3-8)、式(3-9)和式(3-10)所示。

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

权值更新如式(3-11)所示。

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

其中式中符号表示含义如表3-1所示。

|  |  |
| --- | --- |
| 1. 式中符号含义 | |
| 符号 | 含义 |
|  | 列向量，输入数据 |
|  | ，矩阵，权值矩阵 |
|  | ，列向量，权值向量 |
|  | 激活函数 |
|  | 列向量，隐藏层输出 |
|  | 列向量，输出 |
|  | 指示器向量 |
|  | ，列向量，误差 |
|  | ，矩阵，更新值 |
|  | 常数，步长 |
|  | 正则化参数 |

其中指示器向量取值满足式(3-12)。

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

本文对数据进行添加噪声的操作通过对数据进行概率采样完成，采样过程满足式(3-13)。其中为噪声比率，为数据置信度参数，本文实验中由于对数据已进行归一化处理，故而如无特殊描述，式(3-13)中取值1，即直接使用数据向量原始值进行训练。

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

本文中添加的神经元与偏置神经元地位相等，作用相当，不同的是私有神经元为每条数据特有，独立训练与调整，偏置神经元为模型网络共享。SDAE添加偏差与私有神经元的结构如图3-1。其中训练数据在进入模型前分别进行添加噪声处理、添加偏置维度操作和增加私有神经元操作。

|  |
| --- |
|  |
| 1. 堆叠降噪自动编码机实例 |

广义自动编码机，根据定义可以表述为概率图模型，形如图3-2。其中、作为、事件的参数条件，分别决定两个概率事件的表现。

|  |
| --- |
|  |
| 1. 堆叠降噪自动编码机图模型 |

### 概率矩阵分解的概率图模型

类似于自动编码机，本文对概率矩阵分解模型的初始化通过一系列操作完成。对于矩阵分解应用于协同过滤可以定义为如图3-3的概率图模型[[31](#_ENREF_31)]，由概率图模型设计PMF的模型生成过程，利用高斯分布通过对各向量参数进行初始化抽样，得到表3-1。其中是的准确化参数，用来衡量对应评分的置信度。本文中取值由式(3-23)决定。

|  |
| --- |
|  |
| 1. 概率矩阵分解概率图模型 |

当c为1时，模型的最大后验估计符合式(2-5)，当c越大时，表示评分可信度越高。实际观测中，如果评分为0，则可能有两种情况，用户i对项目j不感兴趣或者未进行过评分反馈。实验中我们对评分值是否为零，使用一致性策略，描述为令c取值满足式(3-23)，其中a>b>0，a和b作为超参数，调整模型对待预测值的预测。本文在构建概率矩阵模型后，利用坐标下降法对模型参数进行学习，可以在模型收敛后用来拟合模型。

|  |
| --- |
| 1. PMF模型定义 |
| 概率矩阵分解 | |
| 1. 对每个用户i，初始化用户隐含向量为 2. 对每个项目j，初始化项目向量隐含向量为 3. 对每个用户-项目评分对，初始化反馈值为 | |

### CDL-i的概率图模型

类似于文献[16]，本文使用SDAE作为算法的组成部分，并为模型添加私有项目节点。一般化的协同深度过滤构建过程可被描述为：

1. 对堆叠降噪自编码网络的每层网络分别做1)、2)、3)操作。
2. 对当前层网络权值矩阵记，对权值矩阵的每一列阵进行采样，使之服从式(3-14)所示分布。

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

1. 对第层偏差向量，采样使之服从如式(3-15)描述分布。

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

1. 对第层数据矩阵的每行数据，按式(3-16)进行采样。

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

1. 对于每一条输入数据，进行初始化操作。操作过程分别依照1)、2)、3)完成操作。
2. 初始化模型输入，使之服从分布(3-17)。

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

1. 为项目偏置结点初始化，使之服从式(3-18)。私有结点权值为项目独有，在网络内并无共享。

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

1. 创建项目的隐含偏置向量服从式(3-19)，并由式(3-20)得到项目隐含向量。

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

1. 为用户创建隐含向量，对每位用户按照式(3-21)进行初始化操作。

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

1. 创建评分矩阵初始值，对每对用户项目评分，按照式(3-22)进行采样操作。

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

此处的分别为模型的超参数，则是对于评分矩阵中每对用户-项目评分矩阵的置信度。类似[[30](#_ENREF_30)]，本文按式(3-23)对进行赋值。

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

其中作为联络用户评分和内容信息的通道，配合的使用，能够学习到项目间隐含关系的高效表达。考虑到效率，设定为无穷大。CDL-i的概率图如图3-4。

## CDL-i算法描述

在按照图3-2、图3-3、图3-4利用相关公式构建模型后，得到了加入根据经验设定先验概率的初始化模型，其中本文使用和CDL一致的训练算法，分别对各部分进行学习和优化，最后对整个模型进行调优，模型收敛后得到学习完成的模型。

CDL-i对CDL进行模型上的改进，算法求解使用与CDL一致的方法和步骤。CDL-i训练中当给定用户和项目隐含向量时，使用坐标梯度下降对PMF的权值及项目对应的偏置向量进行调整；在完成一次PMF的调整后，根据PMF的结果对SDAE进行优化，再对调整后的项目偏置向量进行剥离得到项目隐式表达，可得到SDAE对应输出，利用反向传播对神经网络进行调整。本文中CDL-i的目标函数使用与CDL相同的目标函数。

|  |
| --- |
|  |
| 1. CDL-i概率图模型 |

算法的求解为计算概率图模型在给定数据下的最大后验概率，等同于计算目标函数的联合最大似然值，其中设定参数为无穷大，对于PMF模型训练的调整策略如式(3-24)和式(3-25)所示。

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

其中，，，是用户的所有评分组成的列向量。代表前文提到的置信度参数。

在CDL算法中SDAE部分的参数更新服从传统反向传播方法，权值矩阵更新法则如式(3-26)所示，偏置向量更新如式(3-27)所示。

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

算法进行预测的步骤如式(3-28)所示，取近似后得到形如(3-29)所示公式。

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

由前文理论基础，可以得到CDL-i算法主要步骤，如表3-2所示。其中项目私有结点于模型训练前加入，不同项目间的初始化结点满足独立同分布条件，结点权值与模型共同计算。结点更新规则依从于式(3-26)。

|  |
| --- |
| 1. CDL-i算法示意 |
| 输入：增加私有单元INode的项目数据、用户评分矩阵 | |
| 1. 按照图3-4初始化算法模型 2. 按照式(3-14~3-23)对模型参数进行初始化 3. 配置超参数 4. 对某批次数据，分别由自动编码机计算项目隐含向量和，并利用评分反馈信息，根据(3-24)(3-25)进行模型优化 5. 对当前批次数据，得到PMF计算结果，根据式(3-26)(3-27)对SDAE进行优化，更新参数 6. 重复(4)、(5)直至模型收敛 | |

## CDL-i实验分析

### 实验环境

本章所用环境为惠普工作站ML310e v3，CPU为Intel Xeon E3-1220 3.10GHz，内存4G，硬盘500G。系统为Ubuntu 14.04.1 64位，JDK版本为1.8.0\_60，Scala版本为2.11.8，Python版本2.7.6，MXNet版本为0.9.1。

### 数据集描述

类似于CDL，本文使用三个数据集进行试验，分别是，和。前两个数据集来自，分别独立地取自。是一个文献标记网站，允许用户创建个性化的个人图书文献库。当一篇文献被某用户选中，在该用户的个人图书馆中，将会包含文献的摘要、题目和关键词等信息。本文所使用数据集为收集并过滤后的结果，其中数据集作者手动的选择了273个种子标签，并且收集了所有包含这些标签的文献。但其中的用户数据有删选和过滤，当一位用户在个人图书馆中收藏的文献低于三篇时，不被数据集所包括。

筛选后的结果各有不同，包括了5551名用户和16980篇文献作为项目推荐集；包括了7947名用户和25975篇文献作为推荐集。实验中，当用户u收藏了文献v，则将pair(u,v)置1，类似于评分矩阵中的“喜欢”对应分值。在用户-项目评分对上，包含有210537个收藏操作，亦即数据集转换为评分矩阵后，数据稀疏度为0.218%；包含了142807个收藏，对应的可以计算其评分矩阵中仅含有0.065%的数据。

最后一个数据集，来自于，为奈飞公司提供的推荐大奖赛数据集，数据集中包括了用户对电影的评分。实验中，为与前两个数据集起点一致，本文仿照[[42](#_ENREF_42)]的方法将用户评分转换为隐式反馈操作。本文仅考虑正向积极反馈，即评分为5分的用户-项目对。在剔除积极评价次数低于3次的用户和没有得到内容信息的电影后，我们保留了407261名用户和9228部电影。在此条件下，我们的数据集包含了15348808个用户评分，同前两者计算数据稀疏度为0.408%。

### 实验结果分析

* + - 1. 深度学习模型与传统推荐结果对比

在本目实验中，由于传统机器学习并未针对Top-N推荐进行优化。为确保量纲一致，我们利用平均绝对误差(Mean Absolute Error，MAE)和平均根方差差(Root Mean Squared Error，RSME)作为衡量算法效果的度量值。

其中平均绝对值误差MAE的计算如式(3-30)所示。

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

RMSE的计算由式(3-31)给出。

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

相应的，在实验中基于自动编码机进行推荐的对比试验中，本文分别使用了用户和项目作为基准推荐维度，以用户为基准维度的自动编码机对用户向量进行特征提取，由低维度特征向量重构出评分矩阵对应维度数据，进行填充，以填充数据进行推荐，以项目为基准维度的自动编码机则是对项目向量进行特征学习完成推荐。本文使用表示基于用户的自动编码机推荐，表示基于项目的DAE推荐。

表3-3中数据表明，自动编码机做数据填充，应用于个性化推荐，表现良好。在推荐系统中的表现优于传统模型。且实验中模型训练的时间表现也优于许多模型。神经网络对数据提取的抽样特征，提高了推荐系统的算法表现，降低了数据中冗余信息的比例，剔除了数据中无用的隐性联系，对推荐系统表现有一定的提升。实验中所有模型的隐含向量大小设定为100。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. 推荐结果对比 | | | | | |
| 评价指标 | NMF | PMF |  |  | CDL-i |
| MAE | 0.7891 | 0.8292 | 0.7970 | **0.7666** | 0.8173 |
| RMSE | 1.0198 | 1.0097 | 0.9926 | **0.9656** | 0.9946 |

其中NMF类似于PMF，基本思想可以简单描述为：对于任意给定的一个非负矩阵A，NMF算法能够寻找到一个非负矩阵U和一个非负矩阵V，使得满足 ，从而将一个非负的矩阵分解为左右两个非负矩阵的乘积。通过对有缺失值的原始矩阵进行分解，得到低维隐含向量，再有低维向量的乘积进行原始数据的填充从而完成对数据的预测，由预测值进行推荐。NMF和PMF类似能够发现数据中隐藏的特征，在众多领域的应用中发挥着重要作用。是推荐领域一类重要算法的代表。

自动编码机通过数据填充进行推荐，表现优秀的同时，数据还展现出了CDL-i在对项目推荐及预测打分的情境下表现并不理想，整体表现介于协同过滤与DAE之间。这是因为CDL-i对数据进行过滤与筛选，致使在模型学习的阶段丢掉了评分的具体信息。此外CDL-i是针对Top-N推荐进行优化学习，加入的先验条件仅考虑到了用户对项目喜欢与否的绝对判断，在对非一般用户的Top-N推荐上能够取得良好的成绩的同时，对此类用户的评分预测上无法学习到用户对所有项目的偏好信息，尽管有项目内容信息辅助，由于使用了简单的推荐模型，并只关注与用户强偏好项目，致使在探测用户与项目的隐式交互反馈方面出现不足。

* + - 1. 针对自动编码机中不同结构的结果分析

自动编码机的网络规模由人为设定，一般与数据规模相关。但其网络激活函数与数据无关，由设计人员在设计网络结构时主观确定。故而在本文章中我们测试了不同网络组件下的模型性能。网络激活函数配置如表3-4所示。

其中线性映射函数配置为恒等函数。此时，网络中的每一层网络退化为线性回归模型。当映射函数为Sigmoid时，单层网络实际上是逻辑回归模型，只是各结点对应模型的参数不同。式(2-1)给出了逻辑回归所使用的Sigmoid映射。SquareLoss可根据前文定义自编码重构误差函数求得。

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 网络激活函数配置 | | | |
| 模型代码 | 隐含层 | 输出层 | 损失函数 |
| M1 | Identity | Identity | Square |
| M2 | Identity | Sigmoid | Logistic |
| M3 | Sigmoid | Identity | Square |
| M4 | Sigmoid | Sigmoid | Logistic |

反正切映射函数Tanh类似于Sigmoid。相比逻辑回归映射，其变化幅度更巨大，值域范围也扩展至(-1，1)。作用于Sigmoid类似，其非线性映射函数的特性，被用于表示神经元的激活与否。

本目试验中，使用针对Top-N推荐优化的评估指标，平均期望准确率(Mean Average Precision，MAP)和召回率(Recall)。

平均期望准确率的定义分为两部分，分别为准确率和准确率均值。定义由式(3-32)(3-33)给出。计算得到准确率均值后，对所有用户的AP@N计算均值可得MAP。

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

式(3-34)给出了本文中召回率的定义。

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

式中为在Top-N推荐中，给出的推荐结果；为在测试集中用户对项目的采纳情况，表示用户在评分中给出正向反馈的项目集合。表示一个指示器函数，当项目在Top-k中被采纳则为1，否则为0。

在NetFlix数据集上，不同网络结构推荐结果如表3-5所示。

实验表明非线性映射函数极大的提高了模型推荐表现。均衡考虑效果和实现难度，本文随后的实验将基于Sigmoid作为网络神经元的激活函数。

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Top-N推荐中不同结构DAE表现 | | | |
| 模型代码 | MAP@1 | MAP@5 | MAP@10 |
| M1 | 0.1301 | 0.0695 | 0.0571 |
| M2 | **0.2608** | **0.1431** | **0.1199** |
| M3 | 0.2000 | 0.1162 | 0.1011 |
| M4 | 0.2474 | 0.1370 | 0.1143 |

图3-5、图3-6、图3-7中同时展示了在对数据加入噪声的情况下，不同水平下模型的表现，实验中噪声比例分别为0%，20%，40%，60%，80%，100%。

|  |
| --- |
|  |
| 1. 不同结构的SDAE在Top-N中推荐表现 |
|  |
| 1. 不同结构的SDAE在Top-N中推荐表现 |

实验表明当噪声比例在40%左右时模型表现最佳，20%和60%其次，噪声比例在80%时性能表现仅优于全噪声的情况，而100%噪声的情况下，算法表现最差。这是因为在0.4比例时，SDAE在学习数据特征时，被迫对噪声进行过滤，而此时数据中信息含量相对丰富，足以支持模型对噪声进行过滤并重现数据，随着噪声比例的不断提升，噪声对模型的影响越来越大，模型无法习得有效的特征表达还原数据。而当训练数据为全噪声时，SDAE等同于随机非线性填充模型，故而表现最差。当噪声比例为0时，尽管数据信息量丰富，但推荐系统中的反馈数据无法保证数据完全拟合某种分布，用户在反馈信息时的行为表现受多种主观因素的影响，进而对模型分析拟合用户产生噪声信息负反馈，当训练数据噪声为0时，由于模型无法判断有意义数据与噪声数据，导致模型的过拟合，影响算法的最终表现。

|  |
| --- |
|  |
| 1. 不同结构的SDAE在Top-N中推荐表现 |

当网络结构固定时，我们探索了不同隐层维度与最终效果的关系。实验数据如图3-8所示。

由实验数据可以看出，在一定范围内，网络表现与隐层维度规模相关，但到达一定程度后，性能不再提升。综合考虑性能与效率，本文选择50作为网络隐层规模参数。隐层维度的大小在一定数量后，对模型表现贡献减弱，推测是因为实验中推荐所使用数据分布相对简单，神经网络对数据的探索已接近实际分布水平，此时加大隐层结点抑或增大网络层数，均难再获取更多信息与知识来支撑模型表现的提升。故而在推荐系统神经网络隐层维度不必无限扩增。同时本文还测试不同层数网络算法表现，实验数据表明，网络层数达到一定程度，模型表现不再提升，进一步印证猜想。

此外图中数据可以得到MAP@1时，算法表现最佳，这意味着当系统在只为用户推荐一个项目时，用户满意度相对较高。本文分析其原因在于对数据的选取，以及用户对推荐项目的不同测度感受，在模型对用户二值化数据学习后，能够很容易的得到用户偏好倾向，进而发现相对应的项目为用户推荐。但随着推荐数量的逐渐上升，项目内容有下降趋势，模型难以判断Top倾向，进而退化为传统推荐手段，根据数据运算结果进行填充，利用项目排序进行推荐。故而推荐结果N值的增大，会导致推荐平均准确率有不同程度的下降。

|  |
| --- |
|  |
| 1. SDAE中隐含向量维度大小对推荐结果的影响 |

为验证猜想，本文统计了在Top-N推荐中不同N值对召回率和准确率的影响，实验结果如图3-9所示。

|  |
| --- |
|  |
| 1. N值对推荐结果的影响 |

由实验可以推测算法模型的表现随着N值的上升而逐渐上升，但随着N值得持续增大，模型表现逐渐稳定，表明算法推荐表现已然收敛，且与N值变动关系减弱。考虑到用户对项目不同的主观感受，N值不必取值过大，当N取值为所有项目数目，则推荐变得毫无意义，实验中也出现当N值相对项目数目J达到同阶，模型推荐性能急剧下降，此时模型的本质退化为统计用户在当前数据集下的感兴趣项目占项目全量的比例，丧失推荐意义。此外，我们对不同层数的堆叠自编码机进行了对比，实验显示在推荐数据集下，堆叠网络表现并未有明显提升。据此推断，推荐系统中用户与项目的隐含关系较为简单，在充分拟合数据的情况下，对网络深度要求并不苛刻，随后的实验中，本文将进一步验证猜想。

* + - 1. CDL-i实验结果

在基于Top-N的推荐中，本目使用前文表现较好的SDAE结构M2、M4在MAP@300的测度上，与CDL-i进行对比试验，结果如表3-6所示。

实验表明，在推荐准确率方面，CDL-i相比SDAE有明显优势。主要原因在于：

1. CDL-i的训练结合了项目内容信息，在对进行用户与项目隐含关系发现的训练中，掌握到更本质和精确的项目信息。结合协同过滤对用户的画像，可以针对性的调整模型参数，进而提升了模型推荐准确率。
2. CDL-i在目标函数优化中，针对Top-N进行优化，在数据进行二值降噪处理后，数据所保留信息更利于CDL-i进行推荐，而不利于SDAE进行数据补全。

综上两点，可以发现CDL-i在推荐准确率上具有明显优势，充分利用了项目内容的辅助信息，提升推荐性能表现。

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Top-N推荐中SDAE与CDL表现 | | | |
| 算法模型 |  |  |  |
| CDL-i | **0.514** | **0.453** | **0.312** |
| M2 | 0.416 | 0.373 | 0.223 |
| M4 | 0.407 | 0.279 | 0.167 |

此外，CDL-i由SDAE作为组成部分，神经网络的性能对模型的最终表现也会产生影响，结合前文假设，本文对不同网络层数的SDAE组成的CDL-i模型进行了对比试验，表现结果如图3-10所示。

实验显示，相对于一层隐层网络，隐含层数目的增多对模型表现具有积极意义，但层数继续增加，模型表现却并未如预期般提高。

据此推断，推荐系统中用户与项目的隐含关系较为简单，五层网络已然能够充分拟合，当隐层达到两层，模型推荐表现已接近稳定。同时也考虑到本文所使用数据中，项目内容信息较为简单，特别经过数据预处理后，数据信息量降低，模型可探测知识减少，进而在深层网络上学习效果未能有明显提高。同样的现象，在SDAE的实验中也被观测到。结合训练效率、模型规模与实现难度，本文随后的实验基于两层隐层网络，结合输入输出层，网络层数设定为5层。

|  |
| --- |
|  |
| 1. CDL-i算法表现 |

CDL-i的对比试验如图3-11所示，实验表明在同N条件下，CDL表现优于SDAE。在N的变化过程中CDL-i能够更快接近收敛值，表明了对CDL改进能够在更好提取项目内容的基础上提高了算法推荐表现。图中数据由多组实验数据做取平均操作后的结果，源数据实验包括对项目内容向量的抽样密度对比，对不同模型结构的结果对比，不同数据集下的结果对比。综合看来，CDL-i对CDL推荐结果有提升作用。

|  |
| --- |
|  |
| 1. N值对不同结构CDL推荐结果的影响 |

## 本章小结

本章详细描述了协同深度推荐算法CDL-i的改进、基础、构造和训练。详述了算法思路与流程。并进行了多组实验，不仅对比了神经网络中自编码机在推荐系统中的应用，也对比了不同结构模型的推荐表现。此外通过CDL-i与对比算法在推荐召回结果上对比，验证了CDL-i在推荐系统上具有良好表现，在训练过程中具有良好的收敛能力。

1. 协同深度推荐模型的并行化

## 引言

机器学习作为人工智能的主体与核心，各种机器学习算法已经应用到各个领域，如语音识别、图片分类、推荐系统、手写输入识别、漏洞发现等领域。极大提高了计算机智能化程度和各种系统的性能。

随着大数据时代的来临，处理海量数据，建立大规模计算模型成为必然需求。现有数据种类繁多，许多数据有非常稀疏的特点，比如推荐系统关系矩阵、文本词袋模型等。在实际应用场景下，数据量已然达到快速处理有困难的程度，而当数据稀疏度上升，数据量和机器学习算法的计算量更随之几何级上升。故而海量数据场景下的算法并行变得十分必要。随着数据量呈井喷式增长，数据复杂度越来越高，数据多样性日渐丰富，客观上对机器学习算法的性能与可扩展性提出了更高要求。目前主流的机器学习并行算法主要分为三大类，分别是基于数据并行的机器学习、基于模型并行的机器学习和混合模式下的机器学习算法。

前文介绍的协同深度推荐模型CDL-i，取得了较好的算法表现。但注意到训练过程中算法对内存消耗较大，且运行时间较长。为应对越来越庞大的数据，基于上文算法基础，本文对模型进行拆分，提出一种并行化训练CDL-i的方法。通过类似EM的步骤，并行地对模型各部分参数进行训练优化。

## 数据并行

基于数据并行的机器学习算法主要思想为算法模型基本不做改动，将数据进行切分，分别使用部分数据训练多个模型实例，对多个分片的数据并行训练，最后将多个模型进行融合达到充分提升算法性能的目的。基于数据并行的思想实现简单且可控，可扩展性较好，但需要机器学习算法的模型具有可融合性或可叠加性。而且数据层面需要数据满足独立同分布，以及数据倾斜可控。基本结构如图4-1所示[[43](#_ENREF_43)]4。

基于数据并行的机器学习算法，在并行训练模型的过程中，常需要进行通信交互来完成参数交换，此功能被逻辑抽象为一个叫参数服务器的部分(Parameter Server)。并行训练过程中，各模型副本独立训练，训练过程相对封闭，在参数服务器的帮助下，各结点能够将模型副本训练的状态进行发布，同时也可以获取其他模型的训练状态方便模型进行更新。参数服务器对各结点调度，负责模型参数的储存、转发，利用更新参数，然后将更新后的参数增量对各节点进行分发，各训练结点接受参数修正，对模型进行更新迭代，在新状态起点下开始进一步的训练。

|  |
| --- |
|  |
| 1. 数据并行框架图 |

数据并行有同步异步之分，同步模式中，算法与参数服务器进行实时参数交互，模型训练过程中实时性更强；异步模式在一段过程中与服务器交互较少，在一定量的计算完成后，首先进行小批量模型融合，再进行通信进行参数更新。两种模式中异步模式在计算资源、通信带宽以及数据分片等方面具有较高利用率，同时对异常数据具有一定的鲁棒性。异步模式整体性能优于同步模式，但对运算结点性能要求较高，空间占用较大，且更新策略的制定一定程度上受主观影响。

值得注意的是，参数服务器作为逻辑概念，可以以独立模型存储的形式存在；可以依附于某训练终端结点，亦可以是以模型拆分后的分片分布于若干个运算结点，甚至可以是模型与数据并存的结点。

## 模型并行

基于模型并行的机器学习算法主要思想为将模型进行拆分，由几个训练单元分别持有，分别计算模型的不同部分。当一部分算法的输入依赖于另一部分的输出时，产生通信开销。训练完成后，对模型进行组装、融合等操作得到最终模型。基于模型并行的思想实现难度相对较大，且需要模型组件独立可分互不影响或组件部分间的可计算性。但相比数据并行，模型并行对模型拆分，所需计算开销可控且精简，可以充分利用训练数据以及模型期望较佳等特性。模型并行结构如图4-2所示[[43](#_ENREF_43)]3。

|  |
| --- |
|  |
| 1. 模型并行框架图 |

多数情况下，模型并行带来的通信开销和同步消耗超过数据并行，因此加速比也不及数据并行。 但对于单机内存无法容纳的大模型来说，模型并行是一个很好的选择，模型并行能够承担超大体积模型的训练，当数据并行无法容纳模型时，模型并行仍能正常工作，在扩展性上一定程度的由于数据并行。

令人遗憾的是，数据并行和模型并行都不能无限扩展。数据并行的训练程序太多时，不得不减小学习率，以保证训练过程的平稳；模型并行的分片太多时，神经元输出值的交换量会急剧增加，效率大幅下降。

现在业界更多情况下综合两者，同时进行模型并行和数据并行。混合模式下，系统分别将数据与模型进行拆分，不仅拥有极强的可扩展性，而且对模型体积拥有控制能力，能够充分利用现有计算资源，且得到的模型期望可预测。但实现难度大，并行过程中通信开销大，模型学习过程中易出现不稳定等缺点。

## 协同深度推荐(CDL-i)并行

本文中，CDL-i的并行训练采用了混合模式，将模型拆分为SDAE与PMF进行分布式训练。其中SDAE基于数据并行模式，将SDAE进行模块化封装，对数据进行切片，每个分布式节点维护一个完整的网络系统，在一定的迭代次数或者PMF达到特定状态后，进行参数更新。

值得注意的是本文SDAE进行数据并行训练基于神经网络性质：“相同初值的神经网络，对同分布下不同数据样本拟合，最终所拟合数据分布相近。”本文以此为基础，在初始化时保证网络参数配置相同，使用同分布数据的不同分片进行训练。由调度管理结点负责将相同的初始化策略和参数配置进行分发，保证一致的模型参数，随后利用结点各自分得的数据训练。数据模型具有实现容易，配置简单，易维护等特点。但进行模型融合时，需要保证各节点数据足够达到模型收敛或者接近收敛，数据分布相差也不应过大。进而可以保证模型饱和，充分训练。

基于混合模式并行，受启发于EM算法的操作，借鉴小批量梯度下降(Mini-Batch Gradient Decent)的思想，本文提出了将PMF和SDAE拆分并行训练的思路。CDL-i中，PMF和SDAE分别基于数据并行，保证结点上具有完整模型，但CDL-i作为整个模型，于集群训练中是分布式存在的。

下面详细介绍CDL-i的并行化训练流程。算法逻辑图如图4-3所示。使用一批数据，只进行PMF的训练，调整权值。当一批数据训练完成后，固定PMF参数，仍使用当前数据对SDAE进行训练和学习。这样降低了数据对内存的开销，同时降低了IO次数。虽然对计算精度有一定影响，但提升了模型训练效率。

|  |
| --- |
|  |
| 1. CDL-i并行示意图 |

当模型训练开始，manager首先进行map操作，将初始化参数分发至各worker节点，包括模型规模配置，正则化参数，随机数种子等。此后每个worker独立读取数据，利用随机数种子分别初始化对应的各项参数，其中由相同的随机种子保证各worker上神经网络初始状态一致。随后各worker分别利用各自数据分片训练模型，SDAE读取项目内容信息数据，PMF使用用户评分数据。不断迭代模型训练过程，至少保证所分得数据分片中的每一例样本计算一遍。算法流程图如图4-4。图中SDAE与PMF基于如图4-3所示逻辑同样并行训练。

|  |
| --- |
|  |
| 1. CDL-i并行算法流程图 |

各worker利用一部分数据进行模型训练后，进行Reduce操作，将各节点计算结果进行融合。其中约减操作定义为worker将计算所得模型参数上传至 manager，由manager对各参数取平均，分别对SDAE和PMF分发对应模型的融合参数，由结点进行自我更新。至此完成系统的一次迭代。

对系统进行迭代，保证所有worker均完全训练，至少保证训练集中所有数据均参与到系统模型的训练中。最终在manager上得到一个训练完毕的模型。CDL-i并行算法简要步骤如表4-1所示。

|  |
| --- |
| 1. CDL-i并行算法示意 |
| 输入：集群配置参数、模型配置参数、训练超参数、数据分布式存储路径 | |
| 1. 利用集群配置参数，向集群申请资源、启动任务 2. 根据数据分布式存储状态对结点进行模型分发，各结点按照分发所得参数进行模型初始化 3. 并行训练模型：   按照配置参数进行模型训练，当SDAE训练达到交互条件，向自身所对应的PMF发起参数更新请求，并交互参数  PMF等到SDAE的请求或达到自身参数更新条件，更新模型参数，并将项目隐式表示参数进行分发。完成参数更新。   1. 重复步骤3)，直到模型收敛或者到达设定最大迭代次数 2. 由Master进行各结点参数收集，分别将SDAE和PMF参数进行平均融合。 3. 由Master利用一定量数据进行调优(fine-turn)，达到模型收敛或最大迭代。 | |

值得指出的是，实际操作中可以根据集群实际情况，对数据进行多种划分模式，包括但不限于差异化节点数目、节点间不同数据条数、各次迭代处理的数据条数以及系统整体对数据比例划分。

系统的结构包括一个负责分发job、收集数据、调度任务的Manager，以及若干个负责具体计算的worker。值得指出的是各Worker上所分得模型由Manager根据数据所在物理位置进行分发，不同Worker上的模型不一定相同。

其中，分布式系统的各计算节点相同模型的初始值一致。每计算一定量的数据，对模型的参数进行收集归并，不断迭代得到最终权值。此外，本文针对自动编码机低效处理稀疏数据的缺陷，提出使用指示器函数对算法进行改进，避免计算过程中无效计算和无效存储开销。指示器函数如式(4-1)所述。

通过指示器函数的选择，在计算中，系统只关注有效值及其对应参数，训练时有效避免无效计算与操作。在损失一部分模型推荐精度的前提下，模型训练速度得到提高。

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

## 并行化CDL-i的实验分析

### 实验环境

集群采用中国科学院计算技术研究所智能信息处理重点实验室计算平台，拥有6台结点机，任务提交配置为48个Core，Driver-memory为30GB，Excutor-memory为30GB，数据由HDFS读入。Spark版本2.0.1。框架代码版本0.6.0。

### 数据准备

为使数据达到海量级别，本文在用户维度上进行了复制，对数据进行扩展，进而得到海量数据。值得注意的是，在数据并行的思路中，项目的数目直接影响到模型的规模。故而在当前实验环境下，本文只探讨当用户量海量时，集群对海量数据处理的效率。

### 单机试验结果分析

本文算法基于Spark和Scala实现，由于Spark本身具有数据的落盘操作，可以对海量数据的运算及储存进行优化，且前文试验中，单机运行算法进行训练及预测推荐耗时冗长。故而本节进行了单机计算资源下的试验性能分析，比较纯语言线性库的实现和基于Spark分布式平台优化的实现在计算资源有限的情况下的优劣。实验中统计了对不同规模数据进行处理的时间以及模型训练中对模型进行一次迭代产生的时间开销，实验结果如图4-5所示。

实验证明了基于Spark实现的算法在精度上略有损失，推荐精度、评分预测等维度上表现相较于高精度的单机版实现有2~5%的精度损失，在业界误差定义中，此范围的损失是可以接受的。在速度上，Spark平台优化表现出优势，在同样的数据量和同等资源开销的情况下，Spark相对单机算法有着1.3倍的速度提升。而在模型扩展性方面，相较于单机版算法出现的内存溢出、计算时间指数级上涨，基于Spark的算法有着稳定的时间开销。在单机资源应用中，Spark充分利用磁盘操作在损失一定计算时间的代价下极大的扩展了可应对的数据规模。实验证明基于Spark的算法在固定资源条件下，相较于单机算法，可处理的数据量维度上有着强势提升，极大地提高了单机计算可提供的计算能力。

|  |
| --- |
|  |
| 1. 数据量对计算时间的影响 |

### 并行试验结果分析

前文进行了单机条件下，Spark并行算法和单机算法性能对比。实验证明了在单机条件下，基于Spark的并行算法性能已经远超过单机朴素算法。为充分验证并行算法在数据规模可扩展性的能力，本文进行了Spark集群条件下的实验。同时与单机算法、单机Spark算法和不同集群规模下的算法表现进行了对比。

在实验中从数据中进行抽样或复制确保对数据规模进行控制，保证在实验过程中在同样数据条件下，测试不同的集群规模。由图4-6看到在数据量较小时，并行算法时间开销相对较大，变化趋势与数据量并无明显联系，这是因为数据量较小时，分布式计算中平台通讯开销所占系统时间开销比例较大。

随着数据量的不断增大，可以看出到当数据量达到300000条时，系统时间开销开始有明显的增长，此时系统的计算时间占比逐渐变大，并且接近单机可承载模型极限容量，计算时间的变化对系统总时间开销的影响开始显现出来，同时由于单机在接近极限容量时，操作系统自动进行内存的页面置换也使得模型训练时间迅速增长。与单机算法对比，首先分布式的算法能够处理的数据量上限远远大于单机算法；其次由图线对比得出，在数据量逐渐增大的过程中，系统时间开销随数据量增大而线性增长，相比SAE爆炸性增长的时间曲线，具有明显优势。

|  |
| --- |
|  |
| 1. 算法并行性能对比 |

## 本章小结

本章介绍了将协同深度学习迁移到分布式平台下进行并行训练的过程。首先分析了算法并行化的动机与背景；其次简述了当前机器学习算法的并行现状；随后简述了主流算法并行的思想。紧接着，本文提出了针对协同深度学习CDL的并行化改进，并且在计算中对网络做了针对数据稀疏性的优化操作。最后在不同的集群配置下，验证了CDL并行能够提高模型的可扩展性，能够应对海量数据场景下的推荐任务。

1. 基于Spark的分布式机器学习框架

## 引言

数据挖掘与分析与大数据的结合已经成为无可争议的时代趋势，虽然Spark平台提供了MLLib算法库和丰富的大数据处理API，但实际情况下的数据仍然不能直接满足算法模型训练的额需求，数据预处理、特征提取等流程十分繁杂。而不同场景下的应用往往意味着完全不同的处理逻辑与需求，致使需要开发人员进行大量低效且重复的编码工作。

在微博研发中心算法平台部的协助下，本文构建了一款基于Spark的机器学习平台。旨在为广大开发者(包括但不限于数据分析师、数据科学家、算法工程师)提供统一、易用、高效、完善的机器学习相关功能和服务。针对常见的机器学习流程与多种复杂操作，进行了模块化开发。提高了算法训练效率、简化了数据训练的流程逻辑。

当前框架依托于Spark分布式平台，基于Scala反射实现了统一灵活的机器学习框架。框架内包含了多种数据处理逻辑、多种算法模型、多种算法组合方式、调参策略和模型可解释性导出等逻辑，并可通过配置文件对相关逻辑进行定制，以及协同分布式计算平台其他框架进行协同化操作。

## 框架建立动机

机器学习已经越来越多的深入业界的每个角落，在当前信息技术的发展中起了极大的助推作用。然而机器学习算法相对传统应用程序计算量大、迭代次数多、逻辑复杂且多变，在数据量井喷式发展的当代，机器学习算法的并行处理成为一种高效的应对方案。而数据量逐渐膨胀到单机无法承受的地步时，分布式计算不可避免的进入了人们的视野，并迅速成为大数据信息智能的核心技术手段。

在Spark并不成熟的时期，为了应对大数据场景，业务逻辑由Hadoop分布式数据处理平台完成，得到处理完成的数据，交由特定的高性能计算机进行模型训练与算法优化。随后将特定模型文件部署线上由算法进行业务干预。Hadoop平台吞吐量大，但响应效率低下，且相对于复杂的数据处理逻辑和机器学习大量迭代场景略显吃力。并且多个处理流程基于多种大数据平台，需要人工参与，导致效率低下且对异常控制的减弱。

随着Apache Spark大数据分布式平台的不断成熟，其有向无环图带来的数据处理效率的提升、高可扩展性、基于内存的分布式计算和丰富的组件库等特性正在迅速成为大数据处理的综合平台。基于丰富的开发接口与方法以及Scala语言特性和灵活的开发范式，本文实现了一种基于Spark的机器学习框架，将多种业务逻辑统一在有限的几个入口中，对多个机器学习算法进行封装统一化操作，使得由数据到训练样本，再到模型训练，最后到可读模型保存的一系列流程均可通过配置文件完成，简化了机器学习应用场景的业务逻辑，提高了使用效率。目前框架容纳了多种数据操作和算法模型，其主要构成如图5-1所示，CDL在框架中与其他算法地位相当。

|  |
| --- |
|  |
| 1. 算法框架中的CDL算法 |

## 框架算法简述

2. 线性回归(Linear Regression)，线性最小二乘法是回归问题中最常见的范式，作为一种线性方法可以被定义如式(5-1)所示。

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

同时损失函数采用平方损失函数，如式(5-2)所示。

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

不同回归方法是通过不同的正则项来进行区分的。在没有正则项的情况下是普通最小二乘法或者线性最小二乘法回归，L1正则意味着Lasso回归，而L2正则项则代表着岭回归。所有的这些方法都是通过均方差来作为损失函数的。

框架中采用随机梯度下降对线性回归模型进行优化，从而可以拟合简单的有标签数据，进行监督学习。

1. 基于L-BFGS的逻辑回归，逻辑回归作为最基础也最常用的机器学习算法，由于其实现简单，效果良好，被广泛应用于各种环境下。如果将逻辑回归函数定义为逻辑损失，则逻辑回归算法同样可以被归类为线性模型。

框架中包含了多种优化方法的逻辑回归算法，其中L-BFGS作为一种拟牛顿法被Spark所采用，通过求解海森矩阵对算法模型进行优化和计算。

逻辑回归对数据拟合有良好的表现，更多的我们利用逻辑回归对数据进行拟合后，去除其分类阈值，通过算法学习到的参数对不同维度进行加权，供其他模型使用，取得了良好的数据表现。

1. 基于SGD的逻辑回归，当数据量巨大，数据表现复杂的情况下，对逻辑回归模型求解将导致巨大的计算开销。为了提高框架算法的工作范围，我们引入了基于随机梯度下降(Stochastic Gradient Descent, SGD)的逻辑回归算法。

尽管基于SGD的逻辑回归效果相比L-BFGS较差，且模型收敛速度慢，但在计算资源有限或者数据量极大的情况下，仍然有着极佳的表现。框架中逻辑回归默认使用了L2正则化。

1. 基于Tron的逻辑回归置信域算法(Trust Region Methods, Tron)是一种求解非线性优化问题的主流方法，具有效率高，可靠性强，收敛稳定等特性。不同于梯度下降方法的寻找梯度，Tron在进行参数更新前优先计算合适的试探步长(Trial Step)，随后根据双折线法求解优化方向。

框架引入基于Tron的逻辑回归算法，是在当数据量不是特别巨大时，提升模型训练效率和结果。

1. 基于XGBoost的逻辑回归，框架整合了DMLC发布的XGBoost算法库，并实现了基于XGBoost的逻辑回归算法，可以在GPU上进行计算，同时具有在Yarn集群资源管理失效时，仍然利用集群计算资源进行模型求解。
2. 朴素贝叶斯(Naive Bayes)，是一种基于数据各个特征间相互独立的简单多分类算法。算法通过统计所有分类标签下的条件概率分布，利用贝叶斯公式计算各维度下数据标签的条件概率分布，随后利用观测数据的维度与其条件概率分布计算数据所对应的分类标签的概率，并通过此概率值进行分类预测。故而朴素贝叶斯的训练十分高效，只需要对数据集进行一次迭代即可完成模型的训练。其中为了提高算法健壮性和正确性，对算法加入了平滑参数。
3. 频繁模式挖掘(Frequent Pattern Mining)作为数据挖掘算法中极重要的一类算法，应用极度广泛，多年活跃于数据挖掘的研究中。它可以挖掘频繁项目、频繁项集和频繁子序列或者其他的频繁子集。框架在应对关联关系或者隐式拓扑关系的场景下，其他机器学习算法并不能很好的适应需求，故而加入了频繁模式挖掘算法。
4. 隐式狄利克雷分布LDA，概率主题模型常被用来发现一个文档集中的不同主题。隐式狄利克雷分布模型-LDA，作为一个典型的概率主题模型，能够用来对文档进行主题聚类，常被用来探索发现文档主题并对文档进行聚类。此外LDA不同于其他聚类模型的一个文档智能被聚类到一个主题中，LDA允许文档具有多个主题，结果以概率形式呈现。为了进行深入的文本主题分析，框架融入了LDA算法模型。
5. 决策树(Decision Tree)作为一种根据数据构建核心为树状结构的分类器算法，能够在决策树生长过程中，根据信息熵增益或信息混淆程度来决定在特定维度上如何决策，进而能够自动地根据数据特性生成特定维度的分割节点，来对数据进行决策分析，预测和回归。在训练决策树的过程中，可以理解为不同维度特征的自动组合，进而可以求得简单的特征组合表征，在特征工程中，以GBDT的形式有着重要的应用。
6. 基于SGD的GBDT，梯度下降决策树(GBDT)，以决策树作为基础模型组件，通过梯度下降的策略进行Boosting，作为经典的集合算法，能够将多棵决策树学习到的只是进行融合，进而达到更加优秀的算法表现。在GBDT能够充分学习数据中包含知识的过程中，GBDT能够对数据不同维度自动地组合，得到优良的交叉特征。在GBDT的叶子节点被广泛应用的同时，提取数据在GBDT中的决策路径，进而进行特征组合也越来越成为其重要应用场景。本框架实现了基于随机梯度下降的梯度下降决策树，训练完毕后得到模型参数，对数据根据其决策路径进行特征交叉再编码，得到基于信息熵、信息增益等计算结果的交叉特征，能够极大地提高数据特征丰富程度，为下一步算法模型的性能提升奠定了坚实基础。

## 数据处理与算法支持逻辑简述

由于Spark平台MLLib的多种算法以及自实现的算法，对数据格式有着一定的要求，而实际生产数据又多种多样。同时在优化算法表现方面，不同算法有着不同的需求定义，不同场景下的数据也有着不同的分布表现。此外生产数据往往含有噪声而导致数据不洁净，算法计算出现偏差甚至直接无法使用。故而往往实际生产数据需要经过一系列的处理才能作为训练样本进行算法训练求解。基于此框架整合了多种数据处理功能，并且加入了一系列提升框架鲁棒性的支撑模块。

* 1. 数据离散化，数据离散化在简化模型运算，提升模型表现上有着重要应用，框架中对原始数据进行离散化处理，生成基于LibSVM格式的训练样本输入给算法进行训练，包含了单点映射编码(One-hot-encoding)、数据分段(PieceWise)、最大值截取(Max-Interception encoding)等多种离散手段。

单点映射离散编码(One-Hot-Encoding)作为业界常用的数据特征计算方法，是将原始数据的一个维度通过一系列计算映射到多个维度中，目前框架支持对数离散化(以2为底；以10为底；自然对数等)、基本运算离散化(包括加减乘除、取余等)、自定义公式离散化等一系列单点映射操作。例如当某条数据中的一个维度为17983146，将其取以10为底的对数后结果为8，离散化后的表示为00000001。

数据分段编码常用于无法直接进行计算的特征。通过数据分段映射，实现特征的离散化。例如某条数据一个维度取四季变化春夏秋冬，当值为夏时，可将其表示为0100。

最大值截取，可以防止数据异常点对模型的负影响，可通过配置维度取值最大值来将所有大于最大值的数据设定为最大值，从而保证模型参数的稳定，方差可控。在异常检测领用有较广泛的应用。

数据离散化可以将数据拆分为更加精细的表示，将一系列的无法计算的特征进行预处理，提高算法效率与精度。

* 1. 数据定比例抽样，在许多场景下，数据的正负样本分布并不均衡。业界常采用对样本增加权重或者对数据按一定比例采样来实现平衡数据分布的目的。此时，对数据的同分布采样将变得至关重要。

框架通过对训练数据正负样本进行统计，利用分布式WordCount的思想，将数据特征作为Key，统计数据的倾斜程度和样本比例。进一步按照统计所得比例对数据进行采样，保证采样所得训练样本与原始数据同分布，进而降低算法模型受数据倾斜的影响。

此外对数据进行采样，降低了数据样本的数目，一定程度上提升了模型训练速度。

* 1. 样本生成与特征交叉，框架在对数据进行一系列预处理后，在训练算法模型前需要将数据整理成为训练样本，使得算法能够接受。框架使用业界常用的LibSVM格式，在对数据进行抽样、离散化、降噪、截取后，将特征空间下表示的数据整理成LibSVM格式，进行算法训练。此功能单独抽取后，通过配置文件进行配置，实现可定制。

此外在某种特殊的场景下，不同维度间需要计算联合表示或者交叉维度，框架提供了基于特征空间的特征交叉，基于分布式平台可以提供上亿维度的特征交叉计算，随后整理为训练样本。

* 1. 模型训练交叉验证与自动化调参，Spark MLLib提供了算法模型的自动化评估功能，框架引入其对算法模型进行评估。在配置文件中指定待测试的模型参数与评估指标，框架将自动化的对模型训练结果进行评估，得到最佳表现模型，并给出用户定义指标下最佳模型的表现和当前模型训练所使用的参数组合。
  2. 训练模型可解释版本保存，Spark MLLib自带的模型保存接口，保存格式为Apache Parquet格式。尽管效率极高且容易维护，但对用户不友好，在不依赖相关依赖库的情况下解析复杂，对模型分析造成困扰。基于此，框架摒弃了使用Spark高度抽象的API进行模型保存，转而自定义模型文件，在算法训练模型完毕后，对模型进行拆分解读，按照特定格式进行储存。虽然开发难度更大且在存储效率上有一定损失，但得到的模型文件解析容易、人工可读，进而便于跨平台开发和模型分析。并且在脱离Spark MLLib的环境下重构模型进行预测十分容易。
  3. 特征权重评估，在推荐重排序的应用场景下，往往需要由算法模型计算的参数进一步计算得到各维度对最终结果的影响程度，为不同维度设置权重。故而框架囊括了对算法导出模型的分析与统计，并根据业务需求支持维度自动计算，节约业务分析成本，提高机器学习流相关工作的效率。
  4. 鲁棒性功能添加

包括但不限于：样本通过多路径读取；数据清洗；样本过滤；自定义字段分隔符；特征维度不限量翻番；离散特征支持列表参数；自定义文件参数；卡方检验。

此部分作为框架功能性代码，提供框架附加功能需求或者鲁棒性支持，亦或提高框架使用便捷性或者针对特定数据下算法表现进行性能表征的优化。作为支撑功能，不属于框架核心逻辑，在此不再赘述。

## 框架扩展性介绍

为了统一框架逻辑，本文在框架实现过程中使用Scala语言中的特质(trait)来进行操作方法上的统一。特质相当于一种特殊的抽象类，在Scala中特质支持多重继承，进而可以在不影响业务逻辑情况下引入框架统一的操作符定义。其次将算法对象进行自定义命名，利用Scala运行时反射来实现运行时多种逻辑的统一。并利用配置文件获得对自定义算法逻辑的暴露，在不破坏框架统一性操作的前提下实现新算法与框架的整合。

MLRunnable特质的声明：

trait MLRunnable {

def run(spark: SparkSession, input: AnyRef, conf: Map[String, String]): AnyRef

}

派生自定义算法逻辑：

object myObject extends MLRunnable{

def run(spark:SparkSession, input:AnyRef, conf:Map[String,String]) = {

//my code

0.asInstanceOf[AnyRef]

}

}

算法配置操作，通过pipeline.xml配置文件读入算法配置参数，进而对算法进行定制：

<process name="taskName">

<objectname>packageName.objectName</objectname>

<args>

<arg name="key1">value1</arg>

<arg name="key2">value2</arg>

</args>

</process>

暴露算法逻辑：

在依照上面的规范完成开发后，只需将pakcageName.objectName指定在pipeline.xml的<objectname>字段中即可。

<objectname>myObjectName</objectname>

至此，自定义算法作为模块通过框架扩展性支持，在逻辑整体保持不变的情况下，融合进机器学习框架。值得指出的是，作为Spark平台的应用级框架，受限于Spark不同模块的访问权限，对Spark特定的开发功能模块，需要在同一命名空间下实现，比如CDL中的SDAE实现了两种模式，分别是自定义优化模块和调用Spark MLLib中Optimization模块进行优化，故而基于Spark的SDAE版本必须的访问必须满足Spark一些相关访问限制。算法操作的优化算法既可以自定义，也可以在保证命名空间可访问的情况下通过Spark平台中的Optimization功能实现基于Spark原生模块的分布式高效操作。

## Spark机器学习框架下协同深度学习推荐算法

利用框架MLRunnable接口，能够很容易化将算法融合进框架。在完成算法逻辑的编码与测试后，我们将算法进行了模块化并与框架进行了融合，能够依托框架的多种特性对算法的训练、使用提供了更好的支持和更方便的操作。框架中算法基本运算逻辑如图5-2所示。

|  |
| --- |
|  |
| 1. 混合并行示意图 |

在利用Spark MLLib实现并改进已有CDL算法基础上，本文利用框架反射逻辑将本文中改进协同深度学习算法模块化，融合进框架，作为一个具有一定完成度的算法模块使用。其中算法模块由两个独立模块SDAE和PMF融合而成，CDL优化部分被实现为可以调用自定义优化算法和Spark MLLib标配优化算法。模型并行算法如图5-3。

利用Spark RDD或Dataset对海量数据进行处理，整理生成样本，供给算法模块训练使用，其中Spark的配置基于Yarn平台，数据存储于HDFS。在人工对数据进行预处理，并生成样本并配置框架配置文件后，启动框架。

|  |
| --- |
|  |
| 1. 并行CDL模型并行流程图 |

框架根据配置文件启动工作任务，配置集群参数。随后进入机器学习处理逻辑，首先框架负责读写样本，整理为算法模型输入；随后框架根据配置文件配置自动调用CDL算法，如果配置文件中有对样本的抽样操作配置，则进行对样本的抽样后调用算法逻辑；训练期间，由Spark进行数据分配，Yarn进行负责集群资源调度，根据配置文件制定参数，由Spark MLLib进行多个模型的共同训练，得到最终优选模型，并返回参数；如果有特定需求，根据配置文件，框架会按照制定格式导出模型文件。本文中由图4-4可得CDL宏观数据并行流程，图5-2中的Worker作为逻辑上的概念，可以是单机、单机多卡、小规模集群等设备，进而为算法副本并行提供基础。

## 本章小结

本章介绍了在本文课题研究过程中实现的基于Spark的机器学习框架，结合前文CDL-i并行训练的思路，将CDL-i移植到Spark分布式平台下实现，并与框架融合。框架实现了对数据一系列处理，算法的训练，模型评估与模型的生成与导出。同时基于Spark的分布式机器学习框架包含了多种成熟算法，能够将多种业务逻辑统一在有限的入口中，对多个机器学习算法进行封装统一化操作，使得由数据到训练样本，再到模型训练，最后到可读模型保存的一系列流程均可通过配置文件完成，简化了机器学习算法在不同应用场景下的操作流程，降低人工干预程度，提高了算法训练效率与流程鲁棒性。

结 论

协同过滤作为一类被广泛应用于推荐系统中且非常成功的方法，达到了良好的表现。通常，传统方法无法充分利用实际业务场景数据，仅能在数据稀疏度不高的场景下正常推荐，但当面对新入条目，在没有历史数据或历史数据稀少，信息量欠缺的情况下，算法表现急剧下降。为应对所述局限与不足，业界引入了多种方案来应对，通过使用一些评分外信息或附加信息来提升模型表现。当代，神经网络以极强的特征提取能力和模型拟合鲁棒性，为机器学习高速发展提供了新的方向，在多个领域表现出了颠覆性优势。在数据上深度学习极强的拟合能力，为推荐系统在数据极度稀疏情况下提供了新的思路。

深度学习模型训练具有海量参数，需要漫长的训练过程去寻找模型最佳状态；此外随着数据量迅速上涨，传统推荐系统在应对海量数据已十分吃力，结合深度学习模型后，计算量更是远高于以往。在无法快速计算和模型迭代的情况下，推荐模型会出现评分矩阵以及项目内容模型的训练变得难以维护的不足，进而引发多种不可预料的问题。为应对数据量的扩张，以及模型改进带来的海量计算操作，引入分布式计算的必要性日渐凸显。

本文介绍了一种基于协同深度学习的改进模型协同深度推荐(CDL-i)，并对改进后的协同深度推荐进行并行化研究。本文工作主要涉及以下几个方面：

1. 对CDL进行改进，提出协同深度推荐模型(CDL-i)。考虑到项目内容类别的差异，为CDL引入了项目私有化结点，进行差异性训练。CDL-i对每个项目保留独有的偏置权重，提高了算法在对内容探测的表现，在向用户推荐不同项目时，根据结合项目私有化偏置量，改善推荐结果。
2. 在CDL-i的基础上，对算法的并行化进行了研究、实验与分析。提出了一种并行CDL-i的训练方法，将模型拆分并行训练，最终在同等数据规模下缩短了训练时间。同时对模型采用数据并行对数据进行分片，协同计算各子模型，提高模型容量和算法可扩展性。
3. 本文结合Spark分布式计算平台，实现算法并行化训练，得到一系列基于客观数据集的实验数据，通过对数据进行分析对比，验证模型改进的有效性，证明并行研究对算法表现的提升，体现了分布式计算在推荐系统中的意义，得到高可扩展性的并行化CDL-i模型。
4. 此外在对CDL-i并行化研究的过程中实现了一个依托Spark平台的Pipeline机器学习框架，将多种算法操作统一在一种逻辑之下，整合了从源数据到样本生成，模型训练到参数调整，参数评估与模型导出等一系列数据研究流程，形成一套pipeline逻辑，将多数操作自动化，降低人为干预程度，为机器学习算法的使用提供便捷，提高算法使用效率与系统运行鲁棒性。

算法的不足之处与未来研究方向：

1. 由于算法中使用了Top-N推荐作为目标函数训练模型，未能对全局推荐优化，导致了在推荐场景的长尾序列中仍有较大改进空间，且算法仍依赖于协同过滤反馈信息训练，形成了一种不完全的冷启动(Incomplete Cold Start, ICS)场景，在面对完全冷启动(Complete Cold Start, CCS)时仍有不足。接下来的工作将充分考虑数据全量信息，引入用户画像信息结合文中的项目内容探测，对算法进行优化。
2. 深度学习模型的训练成本较高，未来的研究将着力于提高模型训练效率方面，边缘化概率求解自动编码机将是一种可行的方案。

参考文献

[1] Nagarnaik P, Thomas A. Survey on recommendation system methods[C]//International Conference on Electronics and Communication Systems, Coimbatore, India, Piscataway: IEEE, 2015:1603-1608.

攻读硕士学位期间承担的科研任务与主要成果

致 谢

转眼匆匆数年，我的研究生生涯也将要结束，人生中最美好的三年青春时光在母校燕山大学中度过，这个海滨城市中的美丽校园在我的生命中有着不同寻常的意义。行文至此，我的论文也已接近尾声，在此向在我的研究生生涯中给予过我关心、帮助和指导的各位老师和同学表示由衷的感谢！

本文的研究工作是在导师张大鹏副教授的悉心指导下完成的，论文的选题、研究和写作过程中都得到了张老师的精心指导。一直以来，张老师在学术研究、实际工作和生活学习中给予我极大的关心和鼓励。张老师深厚的理论水平、严谨的治学态度、渊博的专业知识、丰富的科研经验、敏锐的科学思维令我深感敬佩；张老师精益求精的工作作风，诲人不倦的高尚师德，严以律己、宽以待人的崇高风范，朴实无华、平易近人的人格魅力对我影响深远。正是因为有了导师严格、无私、高质量的教导，我才能在这三年的学习过程中汲取知识、提升能力。本论文从选题到完成，每一步都是在张老师的指导下完成的，本文成文倾注了张老师的心血，张老师的指导是本文成文的必要条件。同时我还要感谢计算所庄福振老师，庄老师在研究方面都给了我莫大的帮助，同时庄老师的学术态度和敬业精神也令我钦佩。本文的许多工作是在两位老师的帮助与指导下完成的。同时还要感谢王新生教授，在多方面给予我帮助。在此，谨向三位老师表示崇高的敬意和衷心的感谢！

其次，我要感谢实验室的同门师兄弟们。无论是项目方面还是课题方面，他们总是尽全力帮助我解决问题，并把全部的知识和资源分享出来，毫无保留。研究生的三年离不开他们的帮衬与照顾，在此向他们表示感谢！同时要感谢我的父母，一路走来，是家人给了我莫大的关怀与支持，可以使我可以无顾虑的求学。

随后，衷心感谢燕山大学信息科学与工程学院的全体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。同时向本领域知识产出者表示敬意！

最后，再次感谢张大鹏老师关心和付出！

作者简介

|  |  |
| --- | --- |
| D:\Documents\照片\微信图片_20180226180030.jpg | 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。 |