燕山大学

**信息科学与工程学院**

**硕士研究生课题中期考核报告**

|  |  |
| --- | --- |
| 课 题 名 称： | 协同深度推荐算法并行化研究 |
| 研 究 生 姓 名： | 申恩兆 |
| 导 师 姓 名： | 张大鹏 副教授 |
| 学 科 专 业： | 计算机科学与技术 |

燕山大学信息科学与工程学院

2017年11月21日

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 姓 名 | 性别 | 年龄 | | 入学时间 | | 专 业 | |
| 申恩兆 | 男 | 24 | | 2014.9 | | 计算机科学与技术 | |
| 课题来源 | 自选课题 | | | | | | |
| 研究方向 | 机器学习与推荐系统 | | | | | | |
| 课题名称 | 协同深度推荐算法并行化研究 | | | | | | |
| 考核时间 | 2016年11月21日 | | | | | | |
| 课题中期  考核评  审组成员 | 姓名 | | 职称 | | 姓名 | | 职称 |
| 金顺福 | | 教授 | |  | |  |
| 宫继兵 | | 副教授 | |  | |  |
| 孙胜涛 | | 副教授 | |  | |  |
|  | |  | |  | |  |
| 评审组对课题进展及报告的评议：  评议组组长（签字）：  年 月 日 | | | | | | | |
| 专业负责人意见：  专业负责人（签字）：  年 月 日 | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 一.中期考核报告（课题进展情况，所取得的阶段性成果，主要参考文献目录，不少于5000字，可另附页）  1、综述  推荐系统是信息过滤系统的一个子集，旨在通过商品特性或用户社交环境，推断用户的偏好，向用户展示其关注的信息。目前，已有许多方法被提出并应用到推荐系统中，但这些方法主要基于协同过滤、矩阵分解等技术，这些技术难以分析出复杂项目中隐含的信息关联性。  推荐算法主要有基于协同过滤(Collaborative filtering，CF)的推荐，基于人口统计学的推荐和基于内容的推荐等[[1](#_ENREF_1)]。其中，协同过滤是当下应用最广泛的推荐方法之一，该技术在不同方向上均引起研究者广泛关注。协同过滤的本质就是利用其它用户的历史记录协助对目标用户的判断进行推荐[[2](#_ENREF_2)]。协同过滤技术主要分为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[[3](#_ENREF_3)]。在不同协同过滤中，矩阵分解(Matrix factorization, MF)是基于模型协同过滤方法中最热门的一种。矩阵分解通过对用户-商品评分矩阵分解，得到一个低秩用户特征矩阵和一个低秩商品特征矩阵，并使用得到的低秩矩阵乘积来拟合原评分矩阵，对未观察到的用户-商品对进行预测。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[[4](#_ENREF_4)]、基于非负矩阵的矩阵分解方法[[5](#_ENREF_5)]，和基于矩阵局部性的矩阵分解方法[[6](#_ENREF_6)]等，这些方法都展示了其在推荐系统上的效果。基于矩阵分解的方法中，重要的因素是其所学习到的用户特征矩阵和商品特征矩阵，良好的特征能带来更好的预测结果。  随着社交媒体的流行，越来越多研究者开始研究利用了用户间社交关系的社会化推荐系统，有效的利用社交网络用户关系信息可以提高推荐的效果。Hao Ma通过共享相同的用户低秩矩阵，将用户-商品评分矩阵和用户的社交信任网络结合在一起，提出了一个具有可解释性的概率因子分析模型[[7](#_ENREF_7)]。类似的，Hao Ma在矩阵分解的优化函数中引入了基于社会化的正则化项，该正则化项使得在对目标用户进行预测时，跟目标用户有较高行为相似度的好友的特征因子被更加关注[[8](#_ENREF_8)]。社会化正则化项有效的利用了用户的社交关系，提高了推荐效果。  近年来，深度学习技术已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良好数据特征表示。将深度学习技术应用到推荐系统进行特征学习，能够学习到更具代表性的用户特征和商品特征。随着深度学习研究的深入，不断有研究人员将深度学习和推荐系统结合起来，已有一些工作致力于将深度学习应用到推荐系统中。其中Aaron等人2013年将深度学习应用到音乐推荐中，在针对音乐做为项目的推荐场景下，使用深度学习技术对项目内容进行探测与表达，结合传统分类技术与推荐技术生成推荐结果[[9](#_ENREF_9)]，更是将两者的结合推向一个新的高潮。  个性化推荐系统中存在一些问题，主要包括稀疏性问题，冷启动问题和可扩展问题等，深度学习可以在一定程度上解决个性化推荐领域中的冷启动以及稀疏性问题。通过发现数据中的高级抽象特征，推断项目间、用户间以及用户与项目间的内在联系，进而缓解推荐系统的局限性。Zhang利用用户和项目通过神经网络构建分布式向量，利用用户向量与项目向量的分布式表达对神经网络进行训练，进而提升系统推荐效果[[10](#_ENREF_10)]。Liang提出了一种概率评分下的自动编码机模型用于无监督特征学习，通过自编码机发现用户隐性特征，生成对用户偏好的隐式向量表达，结合最近邻模型协同过滤提供个性化推荐[[11](#_ENREF_11)]。Salakhutdinov提出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[[12](#_ENREF_12)]。为了有效利用外部信息，Wang提出了一个层次贝叶斯模型，称之为协同深度学习(Collaborative Deep Learning, CDL)，使用深度学习来学习内容信息，并将之与协同过滤相结合来得到评分矩阵[[13](#_ENREF_13)]。类似于Aaron等人的工作，基于音乐内容的推荐结合深度学习，利用DBN进行内容特征提取进而提升推荐结果准确性，取得了令人满意的进展[[14](#_ENREF_14), [15](#_ENREF_15)]。业界也有将用户行为看做序列化信息或将推荐整个过程当做序列化信息，利用RNN对序列化数据进行学习和探测的研究[[16](#_ENREF_16), [17](#_ENREF_17), [18](#_ENREF_18)]。也有研究利用自动编码机，对特征矩阵进行降维，进而实现数据填充或者推荐算法运行效率的提高[[19](#_ENREF_19), [20](#_ENREF_20)]。  已有的工作证明，深度学习在推荐系统中学习到的特征具有代表性，对于解决推荐系统中数据稀疏问题有效，可以提高学习性能。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学习到有效特征信息变得十分困难。目前基于深度学习的推荐算法还比较少，但初步的研究成果已经表明它们在实际数据应用中的有效性。  2、课题进展  2.1 并行自动编码机  自动编码机(Auto-Encoder, AE)是一种无监督学习算法[[21](#_ENREF_21)]，作为神经网络的一种变体也被称作自编码神经网络，使用反向传播(Back-propagation)算法，将输入值作为输出目标，比如。自动编码机主要由两部分组成，分别是编码部分和解码部分。训练中，编码部分和解码部分同时优化，使得。亦即，它试图逼近一个恒等函数，使得输出接近于输入。自动编码机逻辑结构如图2-1所示。   |  | | --- | |  | | 1. 自动编码机逻辑结构 |   一般的，自动编码机隐含层节点少于输入层节点，迫使其学习输入数据的压缩表示，换言之自动编码机需要由较少的隐层神经元输出重构出原始数据。通常情况下，数据中隐含一些特定结构，使得不同维度间彼此相关，那么就使得算法可以发现输入数据中的相关性，进而得到数据的有效低维表示，即数据压缩表示，或者称之为去除冗余信息的特征表示。图2-2为我们展示了一个简单自动编码机的结构。   |  | | --- | |  | | 1. 简单的自动编码机结构 |   图2-2中的自动编码机，作为前馈神经网络，分别由输入层、隐含层和输出层组成。其中输入层为数据线性表示层，隐含层和输出层中每个节点代表神经网络的一个神经元。神经元由上层网络输出的加权和通过激活函数得到输出，并提供下一层网络使用。神经元可以简单表示为如图2-3所示。   |  | | --- | |  | | 1. 神经元 |   激活函数一般为非线性函数，通常使用Sigmoid函数，其函数表达式如式(2-1)：   |  |  | | --- | --- | |  |  |   一般的，自动编码机编码部分和解码部分可以概括为式(2-2)和(2-3)。   |  |  | | --- | --- | |  |  | |  |  |   其中为非线性激活函数，权值矩阵分别为、，偏置项为、，隐含层输出为。给定一组输入数据，则重构误差可表示为。整个自动编码机最终目标为通过最小化重构误差，得到恰当的、及、。整个网络的目标函数可表示为。  至此给出了自动编码机(Auto-Encoder)的详细定义与计算，并给出了其求解目标函数。其中当多个自动编码机进行堆叠，上一层编码层(Encoder)的输出为下一层编码机的输入时，我们将他们并称为堆叠自动编码机(Stacked Auto-Encoder, SAE)或者栈式自动编码机，如图2-4所示。   |  | | --- | |  | | 1. 堆叠自动编码机 |   堆叠的自动编码机相对朴素自动编码机可以学习更加深层次的抽象表达特征，业界已经证明了在神经网络充分训练的前提下，神经网络对函数的表征能力与网络层数成正相关[[22](#_ENREF_22)]77，简单来说就是神经网络层数越深，就越能够提取出相比原始数据更加抽象和一般的特征，加强模型的鲁棒性和数据拟合表现。  对应的，如果为自动编码机输入的数据是经过噪声处理的，而最终学习结果为纯净数据；那么训练学习的过程中自动编码机被迫将噪声过滤，进而获得鲁棒性更强的数据特征表示[[23](#_ENREF_23)]1096-1103。我们称这种自动编码机为降噪自动编码机(Denoise Auto-Encoder, DAE)。其结构可简单表示为图2-5。  在使用自动编码机的过程中，常常将堆叠和降噪两种模式同时使用，一般称之为堆叠降噪自动编码机(Stacked Denoising Auto-Encoder, SDAE)。SDAE不仅拥有自动编码机的对数据学习近似恒等表达的能力；同时具有SAE提取高度抽象特征能力与DAE获得数据鲁棒特征的能力。SDAE能够获得更加健壮的恒等函数和特征表达，提取特征也更加鲁棒，具有较强泛化性能。相对的，SDAE学习过程较为缓慢，同时面临梯度爆炸和梯度弥散的问题。故而当单层AE足够获取所需特征时，往往倾向浅层网络结构。   |  | | --- | |  | | 1. 降噪自动编码机 |   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。 系统框图如图2-6所示。   |  | | --- | | 框图四版.png | | 1. 系统架构图 |  2.2 CDL-i算法描述与实现 类似于CDL[[13](#_ENREF_13)]，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。改进后模型本课题命名为CDL-i，其中SDAE部分示意图如图2-7所示。   |  | | --- | |  | | 1. 改进后SDAE示意图 |   CDL-i的对比试验如图2-8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 |   2.3 并行环境下算法实现  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2-9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。  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2-10所示。   |  | | --- | |  | | 1. 数据量对计算时间的影响 |   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。  2.4基于Spark的分布式机器学习框架  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。  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2-11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 |   已有算法模型简述    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，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。 3. 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。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。 4. 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。 5. 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。 6. 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。 7. 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。 8. 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。   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，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。   2. 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。   3. 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。   4. 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。   5. 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。   6. 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。  3、阶段性成果  经过前期的研究，已经取得以下的成果：  (1)了解了深度学习在推荐系统应用的国内外研究现状；  (2)实现了分布式平台下的并行算法，并基于已有算法进行了修改，在CDL算法下提出了CDL-i模型。  (3)基于Spark的分布式机器学习框架基本成型，融合进数个算法模型及支持模块。  参考文献  [1] 许海玲, 吴潇, 李晓东, et al. 互联网推荐系统比较研究 [J]. 软件学报, 2009, 20(2): 350-362.  [2] Schafer J B, Dan F, Herlocker J, et al. Collaborative Filtering Recommender Systems [J]. The Adaptive Web, 2007: 291-324.  [3] Ekstrand M D, Riedl J T, Konstan J A. Collaborative Filtering Recommender Systems [J]. Foundations and Trends® in Human-Computer Interaction, 2007, 4(2): 81-173.  [4] Srebro N, Jaakkola T. Weighted Low-Rank Approximations[C]//International Conference on Machine Learning, Washington, DC, USA, 2003:720-727.  [5] Li B, Yang Q, Xue X. Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction[C]//IJCAI 2009, Proceedings of the International Joint Conference on Artificial Intelligence, Pasadena, California, Usa, July, 2009:2052-2057.  [6] Lee J, Kim S, Lebanon G, et al. Local low-rank matrix approximation[C]//International Conference on Machine Learning, Atlanta, GA, USA, , 2013:82-90.  [7] Ma H, Yang H, Lyu M R, et al. SoRec:social recommendation using probabilistic matrix factorization[C]//ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, Usa, October, 2008:931-940.  [8] Ma H, Zhou D, Liu C, et al. Recommender systems with social regularization[C]//Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February, 2011:287-296.  [9] Oord A V D, Dieleman S, Schrauwen B. Deep content-based music recommendation [J]. Advances in Neural Information Processing Systems, 2013, 26(6): 2643-2651.  [10] Zhang J, Cai H, Huang T, et al. A Distributional Representation Model For Collaborative Filtering [J]. Computer Science, 2015, 20(7): 1-7.  [11] Liang H, Baldwin T. A Probabilistic Rating Auto-encoder for Personalized Recommender Systems[C]//ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 2015:1863-1866.  [12] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering[C]//Machine Learning, Proceedings of the Twenty-Fourth International Conference, Corvallis, Oregon, USA, 2007:791-798.  [13] Wang H, Wang N, Yeung D Y. Collaborative Deep Learning for Recommender Systems[C]//Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 2015:1235-1244.  [14] Wang X, Wang Y. Improving Content-based and Hybrid Music Recommendation using Deep Learning[C]//Proceedings of the 22nd ACM international conference on Multimedia, Orlando, Florida, USA, 2014:627-636.  [15] Hamel P, Eck D. Learning Features from Music Audio with Deep Belief Networks[C]//International Society for Music Information Retrieval Conference, Ismir 2010, Utrecht, Netherlands, August, 2010:339-344.  [16] Wu C, Wang J, Liu J, et al. Recurrent neural network based recommendation for time heterogeneous feedback [J]. Knowledge-Based Systems, 2016, 109: 90-103.  [17] Devooght R, Bersini H. Collaborative Filtering with Recurrent Neural Networks [J]. arXiv preprint arXiv:160807400, 2016, 1(1): 1-7.  [18] Dai H, Wang Y, Trivedi R, et al. Recurrent Coevolutionary Feature Embedding Processes for Recommendation [J]. arXiv preprint arXiv:160903675, 2016, 1(1): 1-7.  [19] Ouyang Y, Liu W, Rong W, et al. Autoencoder-Based Collaborative Filtering [M]. Heidelberg: Springer International Publishing, 2014.  [20] Sedhain S, Menon A K, Sanner S, et al. AutoRec:Autoencoders Meet Collaborative Filtering[C]//Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 2015:111-112.  [21] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks [J]. Science, 2006, 313(5786): 504-507.  [22] Mitchell T, 曾华军, 张银奎. 机器学习 [M]. 北京: 机械工业出版社, 2003.  [23] Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders[C]//Proceedings of the 25th international conference on Machine learning, Helsinki, Finland, 2008:1096-1103.  二.后期重点解决的问题及拟采用的研究方法，实验方案（可另附页）  (1) 重点要解决的问题是提升算法运行效率和算法并行拆分问题，拟采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。  (2) 完成对比实验和可行性实验。包括但不限于CDL-i与基线算法的对比，CDL-i与CDL的对比，课题模型在不同评测维度下的对比。Python实现和Spark实现对比，单机实现和并行算法效率对比等。  (3) 完善机器学习框架。  (4) 撰写毕业论文，准备答辩工作。  三.中期考核报告记录：  1.问题一:如何对模型进行拆分，实现并行?  分为两种方案进行尝试，其一是对CDL-i进行模型拆分，不同节点分别训练不同部分，通过网络通讯同步参数，实现整体模型的并行。其二对CDL-i算法中的矩阵进行拆分并行，进而实现全模型的并行。第一种实现效率不高且对硬件要求较苛刻，但实现相对容易。第二种实现上会有极大挑战，且模型训练过程中不可控因素会迅速变多，但效率较高，且对硬件条件要求相对宽松。现拟采用对模型进行数据并行验证算法并行的可行性，进一步尝试第一种方案实现；在条件允许情况下，进行第二种方案的研究与实验。  2.问题二：数据集规模，如何处理数据，如何保证集群计算优势的体现？  拟采用Netflix数据集进行实验，原因有三。一，本数据集在业界具有极高声誉，在其上进行实验验证，得到的数据比较有说服力；二，数据集规模较大，对神经网络的训练，相比其他数据集能够使模型更加充分训练；三，数据间关系相对简单，在随后海量数据实验过程中，对数据的复制比较容易，且数据规模极度可控。Netflix数据集在经过初步筛选后，得到的规模为40W级别用户，9K级别的项目，合计1500W量级数据。初步设想随后数据规模在用户或项目级别复制，得到亿级别数据。  记录人：  年 月 日 |
|  |