基于Spark的并行化协同深度推荐模型

贾晓光
JIA Xiaoguang

燕山大学 信息科学与工程学院,河北 秦皇岛 066004
School of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei 066004, China

JIA Xiaoguang. Parallel collaborative depth recommendation model based on Spark. Computer Engineering and Applications, 2018, 54(14):71-76.

Abstract: Due to the strong feature learning ability and robustness in model fitting, Collaborative Deep Learning(CDL) solves the problem that the performance falls sharply in the case of data sparsity in recommendation systems. However, recommendation system cannot deal with big data because of the difficulty to maintain the training of model, which can bring up many problems that hard to predict. To solve the problem mentioned above, this paper proposes a model called "CDL with item private node"(CDL-i) on the basis of collaborative depth learning and its parallelization method. The private network node is added to improve the self-coding network in traditional CDL. In the case of sharing network parameters of the model, private bias for each projects are added to make the network more target to the content of the project learning parameters, so as to improve the model of project content detection in the recommendation system performance. In addition, the algorithm is parallelized by splitting the model, and a parallel training method of CDL-i is also proposed. The parallelized model is applied in Spark distributed cluster, and the parameters of each part of the model are optimized in parallel to enhance the data scale and extensibility of the model. The effectiveness and efficiency of the proposed algorithm are verified by experiments on multiple real data sets.

Key words: deep learning; recommender system; Collaborative Deep Learning(CDL); Spark

摘 要:协同深度学习(Collaborative Deep Learning,CDL)利用神经网络极强的特征学习能力和模型拟合鲁棒性,解决了推荐系统在数据稀疏的情况下性能表现急剧下降的问题。但当推荐系统面临大量数据时,导致模型训练变得难以维护,进而出现多种不可预料的问题。为解决上述问题,对协同深度学习及其并行化方法进行了研究,提出了一种针对项目内容学习优化的改进模型协同深度推荐(CDL with item private node,CDL-i),通过对传统CDL中的自编码网络进行改进,增加私有网络节点,在模型的网络参数共享情况下,为每个项目添加私有偏置项,使网络能够更针对性地学习到项目内容参数,改进了模型在推荐系统中对项目内容的探测性能。同时对算法进行并行化改进,通过对模型进行拆分,提出一种并行训练CDL-i的方法,将其移植到Spark分布式集群上,并行地对模型各部分参数进行训练优化,增强模型所能处理数据的规模和扩展性。通过在多个真实数据集上的实验,验证了提出的并行深度推荐算法的有效性和高效性。

关键词:深度学习;推荐系统;协同深度学习;Spark
文献标志码:A **中图分类号:**TP312 doi:10.3778/j.issn.1002-8331.1711-0152

1 引言

当代,由于信息的高速增长,海量数据的迅速产生,推荐系统(Recommender System)对及时获取有效信息起着至关重要的作用,改变了人们与信息的交互方式[1]。推荐系统通过对海量数据进行识别、区分与排序,帮人们从海量信息中筛选出其最感兴趣、最有用的部分,大大简化了信息选择的过程。随着硬件计算能力的提升和互联网的深度发展,人们面对的场景也更加多样化,数据的规模和复杂程度也一日千里。在此背景下深度学习表现出惊人的学习能力,在许多领域取得了长足

基金项目:秦皇岛市科学技术研究与发展计划(No.201701B008)。
作者简介:贾晓光(1981—),男,助理研究员,研究领域为云资源分配,大数据处理,E-mail:jxg@ysu.edu.cn。
收稿日期:2017-11-10 **修回日期:**2018-02-08 **文章编号:**1002-8331(2018)14-0071-06

的进步甚至是颠覆性进展,使机器智能化程度迅速提升。

深度学习在于建立并模拟人脑进行分析学习的神经网络,模仿人脑的机制来解释信息,例如图像、声音和文本[2]。使用深度学习,如自编码神经网络等,可以学习用户和商品的特征,进而在习得的特征上进行进一步的模型训练,或将习得的特征应用到传统的推荐方法中[3]。推荐系统为深度学习算法提供了新的应用场景,而深度学习则为推荐系统的发展开辟了新的方向,注入新的活力。

Aaron等人2013年将深度学习应用到音乐推荐中,在针对音乐作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[4],更是将两者的结合推向一个新的高潮。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[5]。Liang提出了一种概率评分下的自编码神经网络模型用于无监督特征学习,通过自编码网络发现用户隐性特征,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[6]。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[7]。

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[8]。类似于Aaron等人的工作,基于音乐内容的推荐结合深度学习,业界利用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[9-10]。业界也有将用户行为看作序列化信息或将推荐整个过程当作序列化信息,利用RNN对序列化数据进行学习和探测的研究[11-13]。也有研究利用自编码神经网络,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[14-15]。

随着信息的进一步丰富,数据膨胀速度越来越快,已经可以预见到不久的将来硬件性能的提升将无法满足数据量的快速膨胀,成为当代复杂运算的瓶颈;而由于推荐系统多次迭代的特性,也对计算性能提出了更加苛刻的要求。当深度学习应用到推荐场景下,面对海量计算数据时,基于分布式计算平台的协同深度学习不可避免地成为解决计算资源不足、数据量庞大及海量复杂运算等问题的可行选择,成为海量数据复杂推荐场景下的重要解决方案。与此同时,互联网中的数据越来越丰富,冗余信息越来越多,从中提取出有效信息变得愈加困难,研究基于深度学习的推荐问题及其算法,并引入并行学习方法,具有非常重要的现实意义。

2 基于Spark的改进CDL-i模型

2.1 基于CDL的CDL-i模型设计

类似于文献[8]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 J 个项目,用 $J\times S$ 维矩阵 $\boldsymbol{X}_c$ 表示。$\boldsymbol{X}_c$ 中的第 j 行表示对应项目的内容向量,用 S 维向量 $\boldsymbol{X}_{c,j*}$ 表示。对于 I 个用户,对 J 个项目的评分可用 $I\times J$ 维度的评分矩阵 $\boldsymbol{R}=[R_{ij}]^{I\times J}$ 表示。当第 i 个用户对第 j 个项目有正向反馈时,令 $R_{ij}=1$;否则为0。

本文采用图1所示的自编码神经网络作为无监督学习网络算法,分别由输入层、隐含层和输出层组成。其中输入层为数据直接表示,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。

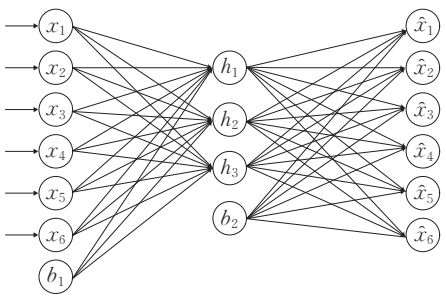

图1 自编码神经网络结构

激活函数使用Sigmoid函数,其函数表达式为:

$$y=\frac{1}{1+\exp(-x)} \tag{1}$$

一般的,一个自编码网络包含编码部分和解码部分,两部分的形式化描述为:

$$\text{Encoding:}\xi=f(W_1x+b_1) \tag{2}$$

$$\text{Decoding:}\hat{x}=f(W_2\xi+b_2) \tag{3}$$

其中 f 为非线性激活函数,权值矩阵分别为 $\boldsymbol{W}_1\in\mathbb{R}^{k\times m}$、$\boldsymbol{W}_2\in\mathbb{R}^{m\times k}$,偏置项为 $\boldsymbol{b}_1\in\mathbb{R}^{k\times 1}$、$\boldsymbol{b}_2\in\mathbb{R}^{1\times k}$,隐含层输出为 $\boldsymbol{\xi}\in\mathbb{R}^{k\times 1}$。给定一组输入数据 $\{x\}_{i=1}^n$,则重构误差可表示为 $\sum_{i=1}^{n}\|\hat{x}_i-x\|^2$。整个自编码神经网络最终目标为通过最小化重构误差,得到恰当的 $\boldsymbol{W}_1$、$\boldsymbol{W}_2$ 及 $\boldsymbol{b}_1$、$\boldsymbol{b}_2$。整个网络的目标函数可表示为 $\min_{\boldsymbol{W}_1,\boldsymbol{b}_1,\boldsymbol{W}_2,\boldsymbol{b}_2}\sum_{i=1}^{n}\|\hat{x}_i-x_i\|^2$。

自编码神经网络中 $\boldsymbol{X}_0$、$\boldsymbol{X}_c$、$\boldsymbol{X}_l$ 分别表示加入噪声的输入、原始数据输入和自动编机第 l 层网络的输出。$\boldsymbol{X}_0$、$\boldsymbol{X}_c$ 为 $J\times S$ 维矩阵,$\boldsymbol{X}_l$ 为 $J\times K_l$ 维度矩阵。类似于 $\boldsymbol{X}_c$,$\boldsymbol{X}_l$ 的第 j 行表示为 $\boldsymbol{X}_{l,j*}$。$\boldsymbol{W}_l$ 和 $\boldsymbol{b}_l$ 分别为第 l 层网络的权值矩阵和偏置向量。$\boldsymbol{W}_{l,*n}$ 表示权值矩阵 $\boldsymbol{W}_l$ 的第 n 列。神经网络的总层数为 L。$\boldsymbol{W}^+$ 是权值矩阵和偏置向量的简称。

为了在评分时充分利用推荐内容的已知信息,本文引入了针对隐含信息的矩阵分解算法。在使用矩阵分解模型进行推荐前,首先计算模型的相关参数,即隐式

空间下的用户向量和项目向量。通常利用已有的用户对项目的评分数据构成矩阵来进行矩阵分解,通过优化式(4)来计算各自隐含向量的参数。

$$\min_{U,V}\sum_{i,j}(r_{ij}-\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{v}_j)^2+\lambda_u\|\boldsymbol{u}_i\|^2+\lambda_v\|\boldsymbol{v}_j\|^2 \tag{4}$$

其中 $\boldsymbol{U}=(\boldsymbol{u}_i)_{i=1}^{I}$,$\boldsymbol{V}=(\boldsymbol{v}_j)_{j=1}^{J}$,$\lambda_u$ 和 λ_v 分别为正则化参数。

如图2所示,通过上述已有矩阵项对评分矩阵进行分解,利用分解得到的隐含向量进行矩阵乘法操作,可以实现缺失数据填充。通过不断对隐含向量的学习与调整,网络可以得到越来越精准的缺失填充值,从而获得精准推荐。

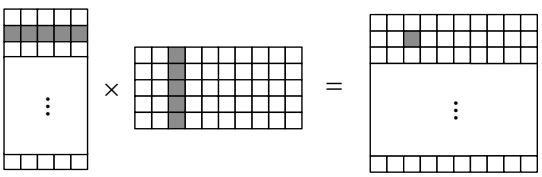

图2　隐含信息矩阵分解

利用上述原理,本文在原CDL的基础上提出了拥有项目私有偏置节点的模型CDL-i,其整体结构如图3所示。通过堆叠降噪自编码神经网络(Stacked Denoising Auto-Encoder,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。

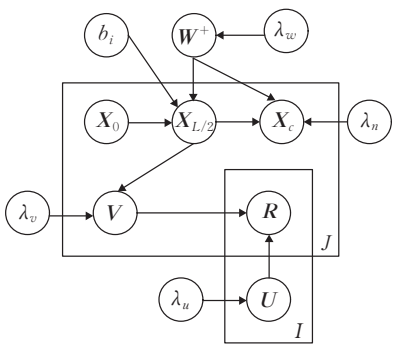

图3　CDL-i概率图模型

本文所提出的CDL-i模型在不增加算法复杂度和模型训练时长的基础上,加入了隐含信息反馈校准机制,使得模型能够个性化地感知项目内容偏好,对项目内容进行个性化画像建模,增加模型对不同类型内容的偏置,进而增加模型对不同类别的项目内容的敏感性与拟合度。

2.2　CDL-i合理性及算法复杂度分析

(1)合理性分析

自编码网络作为一种无监督学习模型,因其强大的拟合性能,在有限内容拟合与探测上具有令人满意的性能。当数据样本规模巨大且分布边界不清晰时,特定地为每个样本增加偏置项,可以记录不同阶段网络的学习状态,均和网络学习残差,能够得到特定样本在网络拟合过程中的专有特色。由式(2)(3)可知,当增加专有偏置节点后,能够显著影响上层网络向下层网络的输出结果,同理由神经网络反向传播,可得网络残差在专有节点上得以累计表征,在往复迭代学习过程中,节点伴随样本一一对应学习,在当前网络状态下,更具有针对性地进行网络损失拟合,进而实现对不同样本的专注性训练,并且对网络整体分布影响较小。因此,在多层网络结构中,对项目添加一一对应能够更好地在当前网络状态下进行数据样本的拟合表征学习。

(2)算法复杂度分析

在单层网络中,有参数矩阵 W 为 (m,n),则正向传播过程中,时间复杂度为 $O(m\times n)$,而通常在自编码神经网络中,样本维度远高于隐层维度,则算法时间复杂度上限为 $O(m\times m)$。而增加网络偏置节点,则矩阵变为 $(m,n+1)$,随后对于不同样本增加item节点,网络计算复杂度更新为 $O((m+1)\times(n+1))$。转化为 $O(mn+m+n+1)$。在自编码神经网络中,n 达不到 m 的低阶无穷小,但在宏观意义上可视 n 为 m 低阶参数,故而算法复杂度约等于 $O(m\times(n+1))$,同理可得算法复杂度上限为 $O(m\times m)$。

由此可得,本文CDL-i在未改变算法复杂度上限的情况下,提升算法针对项目内容拟合性能,增强不同项目下模型的探测灵敏度,具有改进意义。结合后文对模型的并行化研究,使得模型效率性能均有相当的提高。

3　基于Spark的CDL-i并行实现

CDL-i的并行训练采用了混合模式,将模型拆分为SDAE与PMF在Spark框架下进行分布式训练。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。CDL-i并行示意图如图4所示。

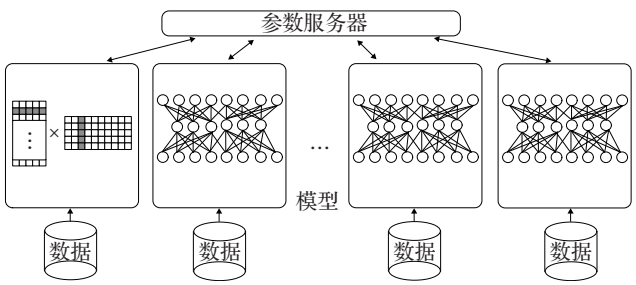

图4　CDL-i并行示意图

在算法初始化阶段,根据数据所在进行分发,同时为避免数据倾斜导致的模型训练时间异常增加,在将数据载入内存后,对数据分区进行随机分片与分发,确保所有计算节点拥有近似均衡的负载。此外在数据读入阶段,利用Spark对数据操作特性,进行了分区内的Shuffle操作,确保单模型副本的数据随机性。

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。

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,由节点进行自我更新。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。算法流程如下所示。

输入:集群配置参数 P_{cluster}、模型配置参数 P_{model}、训练超参数 T_{tr}、数据分布式存储状态 S_{DB}

输出:推荐结果集 R

(1)依据 P_{cluster} 申请分布式计算资源并启动job

(2)根据 S_{DB} 对节点进行模型分割,各节点按照 P_{model} 进行模型初始化

(3)while(当前迭代次数未超过阈值or模型未收敛)

{

① 依据初始 P_{model} 和 T_{tr} 进行模型并行训练;

② if(训练达到交互条件) then 与对应的PMF交互参数;

③ if(SDAE请求到达or满足更新条件) then 重新分发 T_{tr};

}

(4)由Master进行各节点参数收集,分别将SDAE和PMF参数进行平均融合;

(5)由Master利用一定量数据进行调优(fine-turn)并输出最终结果集 R;

在上述算法第(4)步中,参数模型融合本文选择网络参数取平均的方法,即各模型副本参数矩阵在Master上进行取均值操作;第(5)步的结果集 R 包括SDAE和PMF以及对应的偏置结点。由于推荐数据存在高度稀疏的特点,针对稀疏数据带来的错误信息与无效计算等问题,在对CDL-i进行并行化过程中,对模型进行了稀疏优化,在神经网络训练的输入层和输出层进行稀疏化操作,通过稀疏指示器函数对数据和网络对应节点进行筛选,针对性地进行模型训练,不仅避免了未知数据0填充带来的错误信息,同时在达到同样模型同样收敛的情况下,减少网络训练的计算量。

4　实验与结果分析

实验中使用的集群拥有6台节点机,任务提交配置为48个节点,调度节点内存为30 GB,节点运行内存为30 GB,数据由HDFS读入。Spark版本2.0.1。

4.1　数据集和预处理

类似于CDL,本文使用3个数据集进行实验,分别是CiteULike和Netflix。CiteULike-a包括了5 551名用户和16 980篇文献作为项目推荐集;CiteULike-t包括了7 947名用户和25 975篇文献作为推荐集。实验中,当用户 u 收藏了文献 v,则将 $\text{pair}(u,v)$ 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,CiteULike-a包含有210 537个收藏操作,转换为评分矩阵后,数据稀疏度为0.218%;CiteULike-t包含了142 807个收藏,对应的评分矩阵中仅含有0.065%的数据。对于Netflix Prize数据集,本文仿照文献[24]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,保留了407 261名用户和9 228部电影。最终的数据集包含了15 348 808个用户评分,数据稀疏度为0.408%。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。

4.2　CDL-i模型性能对比

在基于Top-N的推荐中,本文在MAP@300的测度上,对不同层次的CDL-i和CDL进行对比实验,结果如图5所示。实验表明,在推荐准确率方面,CDL-i相比SDAE拥有明显优势。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。

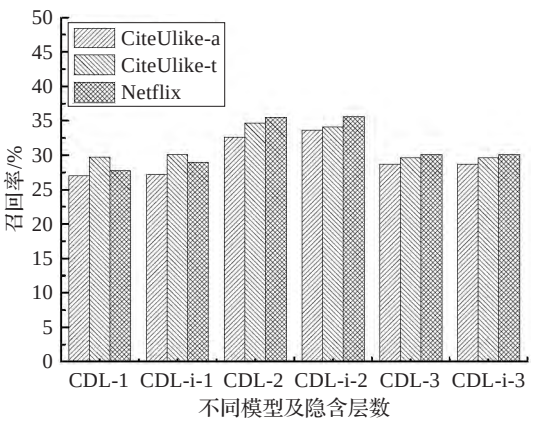

图5　CDL-i算法表现

CDL-i的对比实验如图6所示,图中数据分别由10组实验数据取平均后的结果,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能够更早地接近收敛值,表明了对CDL改进能够在更好提取项目内容的基础上提高算法推荐性能,能够在同样的情况下,取得相对更加良好的推荐表现。

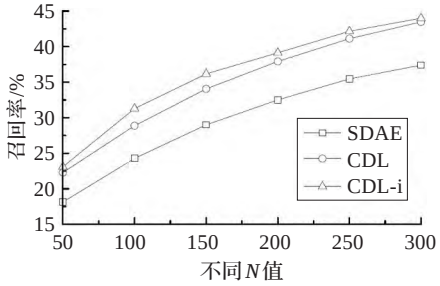

图6 CDL-i算法表现

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。

4.3 并行CDL-i实验

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,且前文实验中,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实验性能分析,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。同时进行了不同规模集群下的实验,对比不同规模集群算法并行加速比。

4.3.1 单机实验结果分析

单机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7所示。其中Scala Data表示使用Scala脚本对不同规模数据进行处理的时间开销,Spark Data表示使用Spark单节点进行数据处理的时间开销。Scala Model和Spark Model分别表示Scala实现的算法与Spark并行算法的时间开销。图8中展示不同需求不同实现方式下的算法开销差异。

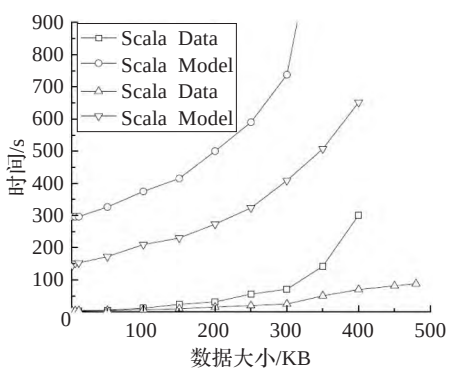

图7 Scala和单机Spark时间开销

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单机版实现有2%~5%的精度损失,在业界误差定义中,此范围可被接受为误差范围。

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。

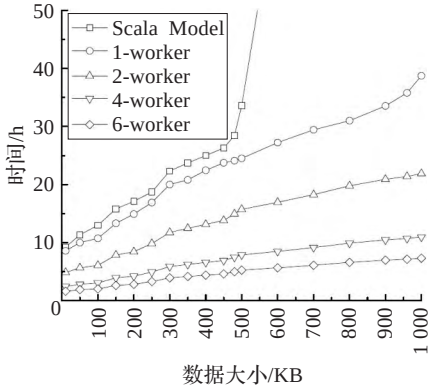

图8 算法并行性能对比

4.3.2 并行实验结果分析

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。实验结果如图8所示。

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。可以看到在算法计算量巨大时,多节点运算相较于单节点运算具有明显性能优势,且不同节点间在同等数据规模下,具有一定的比例关系。

随着数据量的不断增大,可以看出当数据量达到300 000条时,单机系统的时间开销出现了跳跃性增长,而随着数据量的继续增大,系统时间开销呈现不可控的飞速上涨,当数据规模达到单节点处理极限时,系统时间开销变得无法估量。

与单机算法对比,首先分布式算法能够处理的数据量上限远远大于单机算法;其次,分布式算法时间开销明显优于单节点算法时间开销,随着节点数的增长,时间开销优势逐渐凸显;最后由图线对比得出,在数据量逐渐增大的过程中,系统时间开销随数据量增长而稳定增长,具有一定的线性趋势,且在当前数据量下,并未探测出集群计算容量极限,相比单机不可控时间增长,具有明显优势。

5 结束语

协同过滤作为一类被广泛应用推荐系统中并且非常成功的方法,取得了良好的表现。传统方法无法充分利用实际业务场景数据,在没有历史数据或信息量欠缺的情况下,算法表现急剧下降。为应对这一问题,本文提出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。通过对CDL-i进行并行化,将其迁移到Spark分布式平台,使得算法运算时间极大缩短。

参考文献:

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

[2] Lecun Y,Bengio Y,Hinton G.Deep learning[J].Nature,2015,521(7553):436-444.

[3] 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.

[4] 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.

[5] Zhang J,Cai H,Huang T,et al.A distributional representation model for collaborative filtering[J].Computer Science,2015,20(7):1-7.

[6] Liang H,Baldwin T.A probabilistic rating auto-encoder for personalized recommender systems[C]//ACM International Conference on Information and Knowledge Management,Melbourne,Australia,2015:1863-1866.

[7] Salakhutdinov R,Mnih A,Hinton G.Restricted Boltzmann machines for collaborative filtering[C]//Proceedings of the Twenty-Fourth International Conference on Machine Learning,Corvallis,Oregon,USA,2007:791-798.

[8] Wang H,Wang N,Yeung D Y.Collaborative deep learning for recommender systems[C]//Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,Sydney,Australia,2015:1235-1244.

[9] 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.

[10] 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,2010:339-344.

[11] Devooght R,Bersini H.Collaborative filtering with recurrent neural networks[J].arXiv preprint arXiv:160807400,2016.

[12] 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.

[13] Wu C,Wang J,Liu J,et al.Recurrent neural network based recommendation for time heterogeneous feedback[J].Knowledge-Based Systems,2016:90-103.

[14] Ouyang Y,Liu W,Rong W,et al.Autoencoder-based collaborative filtering[M].Heidelberg:Springer International Publishing,2014.

[15] 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.

[16] Wang C,Blei D M.Collaborative topic modeling for recommending scientific articles[C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,San Diego,CA,USA,2011:448-456.

[17] Salakhutdinov R,Mnih A.Bayesian probabilistic matrix factorization using Markov chain Monte Carlo[C]//Proceedings of the International Conference on Machine Learning,Helsinki,Finland,2008:880-887.

[18] Salakhutdinov R,Mnih A.Probabilistic matrix factorization[C]//International Conference on Neural Information Processing Systems,2007.

[19] Hu Y,Koren Y,Volinsky C.Collaborative filtering for implicit feedback datasets[C]//Eighth IEEE International Conference on Data Mining,Pisa,Italy,2008:263-272.

[20] Agarwal D,Chen B C.Regression-based latent factor models[C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,Paris,France,2009:19-28.

[21] Koren Y,Bell R,Volinsky C.Matrix factorization techniques for recommender systems[J].Computer,2009,42(8):30-37.

[22] Herlocker J L,Konstan J A,Borchers A,et al.An algorithmic framework for performing collaborative filtering[C]//Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval,Berkeley,CA,USA,1999:230-237.

[23] 黎文阳.大数据处理模型Apache Spark研究[J].现代计算机:普及版,2015,13(3):55-60.

[24] Zhou K,Zha H.Learning binary codes for collaborative filtering[C]//Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,Beijing,China,2012:498-506.