

大数据机器学习系统研究进展

黄宜华^{1,2}

1.南京大学计算机软件新技术国家重点实验室 南京 210023;

2.南京大学PASA大数据技术实验室 南京 210023

摘要

要实现高效的大数据机器学习,需要构建一个能同时支持机器学习算法设计和大规模数据处理的一体化大数据机器学习系统。研究设计高效、可扩展且易于使用的大数据机器学习系统面临诸多技术挑战。近年来,大数据浪潮的兴起,推动了大数据机器学习的迅猛发展,使大数据机器学习系统成为大数据领域的一个热点研究问题。介绍了国内外大数据机器学习系统的基本概念、基本研究问题、技术特征、系统分类以及典型系统;在此基础上,进一步介绍了本实验室研究设计的一个跨平台统一大数据机器学习系统——Octopus(大章鱼)。

关键词

大数据;机器学习;分布并行计算;大数据处理平台

Research Progress on Big Data Machine Learning System

Huang Yihua^{1,2}

1.National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China;

2.PASA Big Data Laboratory, Nanjing University, Nanjing 210023, China

Abstract

To achieve efficient big data machine learning, we need to construct a unified big data machine learning system to support both machine learning algorithm design and big data processing. Designing an efficient, scalable and easy-to-use big data machine learning system still faces a number of challenges. Recently, the upsurge of big data technology has promoted rapid development of big data machine learning, making big data machine learning system to become a research hotspot. The basic concepts, research issues, technical characteristics, categories, and typical systems for big data machine learning system, were reviewed. Then a unified and cross-platform big data machine learning system, Octopus, was presented.

Key words

big data, machine learning, distributed and parallel computing, big data processing platform

1 大数据机器学习系统研究背景

近年来,大数据技术在全球发展迅猛,掀起了巨大的研究热潮,引起全球业界、学术界和各国政府的高度关注。随着计算机和信息技术的迅猛发展和普及应用,行业应用数据呈爆炸性增长。动辄达到数百TB甚至数PB规模的行业/企业大数据已经远远超出了传统计算技术和信息系统的处理能力。与此同时,大数据往往隐含着很多在小数据量时不具备的深度知识和价值,大数据智能化分析挖掘将为行业/企业带来巨大的商业价值,实现多种高附加值的增值服务,从而提升行业/企业生产管理决策水平和经济效益。

大数据分析挖掘处理主要分为简单分析和智能化复杂分析两大类。简单分析主要采用类似于传统数据库OLAP的处理技术和方法,用SQL完成各种常规的查询统计分析;而大数据的深度价值仅通过简单分析是难以发现的,通常需要使用基于机器学习和数据挖掘的智能化复杂分析才能实现。

机器学习和数据分析是将大数据转换成有用知识的关键技术,并且有研究表明,在很多情况下,处理的数据规模越大,机器学习模型的效果会越好^[1~3]。目前,国内外业界和学术界专家普遍认同的观点是,越来越多的海量数据资源加上越来越强大的计算能力,已经成为推动大数据时代人工智能技术和应用发展的动力,将基于大数据的机器学习和人工智能推上了新一轮发展浪潮,让大数据机器学习(big data machine learning)成为全球业界和学术界高度关注的热点研究领域。随着大数据时代的来临,Google、Facebook、微软、百度、腾讯等国内外著名企业均纷纷成

立专门的基于大数据的机器学习与人工智能研发机构,深入系统地研究基于大数据的机器学习和智能化计算技术。

由于大数据机器学习和数据挖掘等智能计算技术在大数据智能化分析处理应用中具有极其重要的作用,在2014年12月中国计算机学会(CCF)大数据专家委员会上百位大数据相关领域学者和技术专家投票推选出的“2015年大数据十大热点技术与发展趋势”中,结合机器学习等智能计算技术的大数据分析技术被推选为大数据领域第一大研究热点和发展趋势^[4]。

由于大数据机器学习在具体实现时通常需要使用分布式和并行化大数据处理技术方法,也有人将大数据机器学习称为“分布式机器学习”(distributed machine learning)或“大规模机器学习”(large-scale machine learning)。

大数据机器学习,不仅是机器学习和算法设计问题,还是一个大规模系统问题。它既不是单纯的机器学习,也不是单纯的大数据处理技术所能解决的问题,而是一个同时涉及机器学习和大数据处理两个主要方面的交叉性研究课题。一方面,它仍然需要继续关注机器学习的方法和算法本身,即需要继续研究新的或改进的学习模型和学习方法,以不断提升分析预测结果的准确性;与此同时,由于数据规模巨大,大数据机器学习会使几乎所有的传统串行化机器学习算法难以在可接受的时间内完成计算,从而使得算法在实际应用场景中失效。因此,大数据机器学习在关注机器学习方法和算法研究的同时,还要关注如何结合分布式和并行化的大数据处理技术,以便在可接受的时间内完成计算。为了能有效完成大数据机器学习过程,需要研究并构建兼具机器学习和大规模分布并行计算处理能力的一体化系统。

因此,领域内出现了“大数据机器学习

系统”或者“分布式学习系统”的概念,并进行了诸多大数据机器学习系统的研究与开发工作。

2 大数据机器学习系统的技术特征

参考文献[5,6]专门介绍了大数据机器学习系统的技术特征。

如图1所示,一个大数据机器学习系统会同时涉及机器学习和大数据处理两方面的诸多复杂技术问题,包括机器学习方面的模型、训练、精度问题以及大数据处理方面的分布式存储、并行化计算、网络通信、局部性计算、任务调度、容错等诸多因素。这些因素互相影响,交织在一起,大大增加了系统设计的复杂性。因此,大数据机器学习已经不仅仅是一个算法研究问题,而是需要针对大数据集,考虑从底层的大数据分布存储到中层的大数据并行化计算,再到上层的机器学习算法,设计一种一体化的支撑系统,形成易于为数据分析程序员和机器学习研究者使用的、完整的大数据机器学习系统。

一个理想的大数据机器学习系统通常需要具备以下几个方面的技术要素和特征^[5~7]。

- 应当从整个学习的生命周期/流水

线来考虑,包括训练数据和特征的提取、并行学习算法的设计、训练模型和参数的查询管理、分布式训练计算过程,都应在一个一体化的学习系统平台上完成。

- 应提供多种并行训练模式,支持不同的机器学习模型和算法。
- 需要提供对底层系统的抽象,以实现
- 对底层通用大数据处理引擎的支持,并提供数据科学中常用的编程语言接口(API)。
- 应该拥有开放和丰富的生态、广泛的应用和快速的进化能力。

在上述技术特征中,一个非常重要的思路是,要通过系统抽象来降低系统设计的复杂性。如图2所示,一个设计良好的大数据机器学习系统,应当通过定义特定的机器学习编程计算和系统抽象接口,将上层机器学习和底层分布式系统解耦开来,将机器学习实现在现有的大数据计算平台之上,而不需要考虑底层系统层面的因素,以此实现底层大数据处理平台对上层用户的透明化,让上层用户从诸多底层的分布和并行化大数据编程计算细节中解放出来,以便他们致力于上层的机器学习模型和算法的设计实现。通过编程计算和系统抽象层API,向上提供各种机器学习编程计算接口以及学习模型和训练数据的表示,向下由底层分布式系统负责处理并提供高效的分布和并行化计算实现。

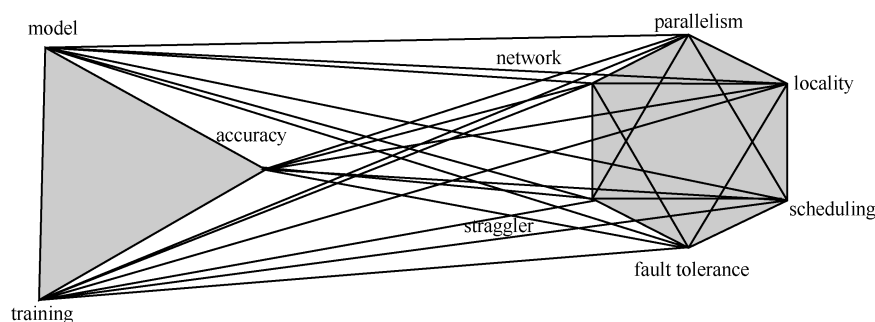


图1 大数据机器学习系统所涉及的复杂因素^[5]

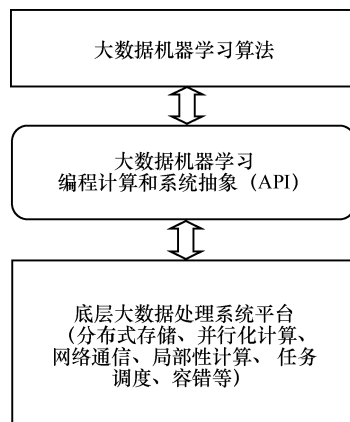


图2 大数据机器学习系统抽象

3 大数据机器学习系统的主要研究问题

知名的Apache Flink大数据分析系统研究者在2014年VLDB会议主题报告和论文中^[8]，从数据分析的复杂程度和数据规模的维度，考察了现有的相关研究工作和系统，如图3所示。现有的系统要么主要用于小规模复杂分析，要么主要用于大规模

的简单统计分析，缺少既具有复杂数据分析能力又具有强大的大数据处理能力的大数据分析系统；文章作者甚至认为，对于行业大数据分析人员而言，现有的工具还处于“石器时代”^[8]。

与此同时，著名的UC Berkeley AMPLab在研究基于Spark的机器学习库MLBase^[9]时，从计算性能和系统易用性两个重要维度，考察了现有的大数据机器学习研究工作和系统，如图4所示。面向机器学习和数据分析时，目前已有的工作和系统，绝大多数都未能同时具备大规模分析处理能力和良好的系统易用性。

因此，大数据机器学习除了需要继续关注和研究传统意义上的学习方法和算法问题，以不断提高学习精度外，还需要重点关注和研究解决大数据场景下所特有的两大技术问题：一是大数据复杂分析时的计算性能问题；二是大数据机器学习系统的可编程性和易用性问题。

(1) 大数据复杂分析时的计算性能问题
在计算性能方面，大规模数据集给很



图3 研究者Apache Flink提出的分析维度和研究现状^[8]

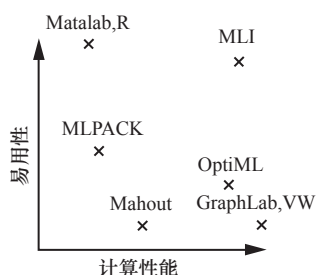


图4 Spark系统研究者提出的分析维度和研究现状^[9]

多传统串行的机器学习和数据分析挖掘算法带来很大的挑战,需要研究解决面向大数据机器学习和数据分析的高效计算方法和系统。在数据集较小时,很多复杂度在 $O(n \log n)$ 、 $O(n^2)$ 甚至 $O(n^3)$ 的传统串行化机器学习算法都可以有效工作;但当数据规模增长到极大尺度时,现有的串行化算法将花费难以接受的时间开销,使得算法在实际应用场景中失效,这给业界大量实际的大数据复杂分析应用带来很多新的挑战和问题^[10]。正如微软全球副总裁陆奇博士在2012年指出的,“大数据使得现有的大多数机器学习算法失效,面向大数据处理时这些算法都需要重写”¹。

在大型互联网企业,大规模机器学习算法经常用于处理十亿至千亿级别的样本以及一亿至数十亿数据特征的大规模数据集。例如,Google著名的Seti太空搜索项目需要进行高达千亿样本、十亿特征数据的大规模机器学习²,腾讯Peacock主题模型分析系统需要进行高达十亿文档、百万词汇、百万主题的主题模型训练,而仅一个百万词汇乘以百万主题的矩阵,其数据存储量即高达3 TB,如果再考虑十亿文档乘以百万主题的矩阵,其数据量更是高达3 PB^[3]。如此大量的训练样本,加上机器学习算法本身的复杂性,导致难以在传统的串行化计算平台上、在可接受的时间内完成如此巨大的复杂分析计算任务,因而

带来了十分突出的计算性能问题。因此,大数据机器学习算法和系统需要研究解决大规模场景下高效的分布式和并行化算法设计以及计算问题,以保证算法和系统可以在可接受的时间内完成大规模数据的学习和训练。

(2) 大数据机器学习系统的可编程性和易用性问题

为了满足急剧增长的大数据处理需求,过去几年来,全球推出了Hadoop^[11]和Spark^[12]等多种主流大数据处理技术和系统平台。这些主流技术和平台的出现为大数据的处理提供了良好的技术手段和方法。Hadoop的出现使得大数据处理技术和平台从无到有,而基于内存计算的Spark系统的出现使得大数据分析计算从慢到快。然而,现有的大数据处理技术和系统平台都存在很大的易用性问题,难以为普通的程序员掌握和使用,尤其是面向复杂的大规模机器学习和数据分析时,这个问题更为突出。

大数据机器学习和数据分析的并行化程序设计,比传统的串行化算法设计复杂很多、对程序员的技术要求更高。对于普通的数据分析程序员或机器学习研究者来说,在特定的大数据编程模型和平台下进行并行化机器学习和数据分析算法设计,需要掌握很多分布式系统背景知识和并行程序设计技巧,这对他们来说难度较大,经常要把实际工作的大部分时间花费在底层复杂的并行化和分布式编程和调试上,给上层数据分析和算法设计工作带来很大的不便和困难,导致普通程序员与现有的各种大数据处理平台之间存在一个难以逾越的鸿沟^[8, 9, 13~15]。

另一方面,即使对于诸如Google、百度等具有强大技术力量的互联网企业程序员来说,随着多种大数据处理与编程平台的出现以及各种分析应用问题需求的不同,

1
<http://www.china-cloud.com/zt/2012iCOME/>

2
<http://google.research.blogspot.com/2010/04/lessons-learned-developing-practical.html>

他们也常常需要针对不同平台,重复编写和测试各种并行化机器学习和数据分析算法,这种在多个大数据平台上重复编写和尝试数以百计的机器学习算法的工作负担也很重。正如阿里巴巴集团副总裁、高级研究员章文嵩博士在中国计算机学会第一届大数据学术会议上所指出的,“淘宝内部需要使用的并行化机器学习算法很多,随着新型的并行化计算框架的逐步出现,需要将其逐个实现到MPI、MapReduce、Spark等并行计算框架上;如果后面出现更先进的框架,又面临着将这些算法重新实现一遍的负担”。

针对大数据的机器学习和数据分析,现有的大数据处理技术和平台存在很大的可编程性和易用性问题,导致普通程序员和常规的程序设计方法与现有的大数据分析处理编程技术之间存在着一个很大的鸿沟。正如CCF大数据专家委员会《2015年大数据发展趋势预测报告》指出的那样,“由于现有的大数据平台易用性差,而垂直应用行业的数据分析又涉及领域专家知识和领域建模,目前在大数据行业分析应用与通用的大数据技术之间存在很大的鸿沟,缺少相互的交叉融合”^[4]。

因此,大数据机器学习系统要解决的一个重要问题是,让不易使用的大数据处理技术和平台变得易于使用,要填平普通程序员和常规程序设计方法与现有的大数据处理技术之间的鸿沟,尽可能让普通程序员用常规程序设计方法便能有效完成大数据的复杂分析处理。

根据Google Seti项目研究人员在开发实际的大型机器学习系统上的经验,在面向大数据时,考虑到用户开发效率问题,系统易用性与提高机器学习精度几乎同等重要,有时甚至比精度问题更为重要,“也许过去学术界很少关心设计一个精度稍差、但有更好易用性和系统可靠性的学习算

法,但在实际应用中,这会体现出非常重要的价值”。

4 大数据机器学习系统的分类

近几年来,随着大数据的来临,基于大数据的机器学习方法和系统成为业界和学术界普遍关注的研究热点。为了提供有效的大数据机器学习和数据分析手段,业界和学术界在尝试多种途径和方法,在不同的并行计算模型和平台下,在实际的数据分析挖掘中得到应用。

较早出现以及现有的一些大数据机器学习算法和系统大多采用较为低层的紧耦合、定制化方法构建。这些算法和系统大都针对特定的应用,选择特定的平台和模型,针对特定的机器学习模型和特定的计算模式,从串行算法和原型到分布并行化算法和原型,自底层向上进行紧耦合和定制化的开发和优化。尽管这可以最大化利用系统资源以达到最佳的性能,但这种低层的紧耦合定制化实现方法,将学习和系统混杂在一起,实现难度大,算法和系统也难以调试和维护^[5,6]。

从前述的大数据机器学习系统特征来看,一个设计良好的大数据机器学习系统应当考虑高层的系统层抽象,向上为程序员提供易于使用的高层机器学习算法编程接口,向下基于现有的通用化大数据处理平台提供大规模数据的分布和并行化计算能力。为此,近年来已经开始出现基于各种高层编程计算和系统抽象设计大数据机器学习系统的研究工作。

在高层编程计算和系统抽象上,目前的大数据机器学习系统大致可分为3种主要的编程计算和系统抽象方法,分别是基于矩阵模型的抽象、基于图模型的抽象以及基于参数模型的抽象。

(1) 基于矩阵模型的抽象

基于矩阵模型的抽象基于这样一个事实：大多数机器学习和数据分析算法都可以表示为矩阵或向量代数计算，因此可以构建一个基于矩阵计算模型的机器学习系统，允许程序员直接基于矩阵计算快速构建机器学习和数据分析算法。最典型的基于矩阵模型的大数据机器学习系统尚不多见，UC Berkeley AMP Lab推出的Spark MLlib是一个试图基于矩阵计算模型构建上层各种机器学习算法的系统³；最近Spark系统提出了一个DataFrame的数据抽象机制，允许基于矩阵和表等数据结构，设计各种机器学习和数据分析算法。此外，国内由南京大学PASA大数据实验室研究开发的“Octopus（大章鱼）”系统是一个以大规模矩阵模型为中心的跨平台大数据机器学习系统⁴。

(2) 基于图模型的抽象

虽然矩阵可以表示大多数机器学习和数据分析算法，但对于一些基于图模型的社会网络分析问题，基于矩阵模型进行计算并非最有效的方法（虽然图和矩阵表示具有理论上的等价性）。矩阵模型对于图数据的表达不如图模型，且通常会面临稀疏性的问题。因此，一些社会网络分析类的实际应用需要基于更为高效的图模型的学习和分析方法，为此出现了相应的基于图模型的大数据机器学习系统。最典型的基于图模型的系统是美国卡耐基梅隆大学（CMU）推出的GraphLab系统^[16]以及基于Spark构建的GraphX系统。

(3) 基于参数模型的抽象

基于矩阵模型和图模型的系统主要是从数据表示模型的角度来构建系统，这两种方法的特点是更接近于实际学习问题中自然化的数据表示和计算模型，因此对于数据分析人员方便快速地构建机器学习和数据分析算法比较自然和直观。但是，

大数据机器学习常常还需要涉及大规模模型，例如，近几年发展迅猛的深度神经网络算法，常常需要通过对大量模型参数的调优来提高学习精度，在这种情况下，上述两种方法会缺少灵活性，难以让算法设计者通过控制模型参数进行算法优化。为此，有研究者提出了一种基于模型参数的抽象方法，即把所有机器学习算法抽象为对学习过程中一组模型参数的管理和控制，并提供对大规模场景下大量模型参数的有效管理和访问。目前参数模型最典型的方法是最初由美国卡耐基梅隆大学的Li Mu等人提出的、后在很多系统中得到进一步应用的Parameter Server^[17, 18]。

实际的系统也可能会是一种兼有上述多种类型的混合系统。此外，也有不少系统从并行模式角度来区分，分为数据并行和模型并行两种方式，且一些系统只提供数据并行方式，另一些系统会同时提供数据并行和模型并行两种方式。

³ <https://spark.apache.org/docs/1.2.0/mllib-guide.html>

⁴ <http://pasa-bigdata.nju.edu.cn/octopus/>

5 典型大数据学习方法和系统介绍

5.1 基于特定平台的定制式并行化机器学习算法与算法库

随着Hadoop MapReduce^[11]和Spark^[12]大数据并行计算与编程平台的出现和成熟应用，Hadoop和Spark已经成为目前大数据分析处理的主流平台。为了解决大规模机器学习问题，有大量的研究工作致力于基于Hadoop MapReduce和Spark以及传统的MPI并行计算框架，完成各种并行化机器学习和数据挖掘算法的设计。在诸如TPDS、JPDC、IPDPS、ICPP、ICPADS、IEEE BigData等大数据和分布并行计算领域的知名期刊和国际会议上，这些并行化算法研究工作不胜枚举。由于需要同时顾及上层机器学习算法

设计和底层分布并行化大数据处理系统层细节, 总体而言, 基于上述不同的并行化编程方法和平台进行并行化机器学习算法设计, 仍属于一种专业性较强、技术要求较高、过程较为繁琐的工作, 难以为缺少分布式和并行计算知识背景的普通程序员使用。

为了让普通数据分析程序员能够直接使用并行化机器学习算法, 目前普遍的一种做法是, 在不同的并行化计算平台上, 由专业的机器学习算法设计者实现并行化机器学习算法, 提供一个机器学习和数据挖掘工具包以供一般的数据分析和应用开发工程师直接使用, 如Hadoop下的Mahout⁵以及Spark环境下的MLlib。

Mahout的主要任务是设计并提供一些基于MapReduce的可扩展的机器学习领域经典算法库, 包括聚类、分类、推荐过滤、频繁项集挖掘等。这些并行化算法都是基于Hadoop MapReduce平台完成的, 用户可直接调用Mahout算法库实现好的算法。然而, Mahout提供的并行化机器学习算法数量有限, 而且作为标准的软件包, 其所提供的算法几乎都是标准的, 在算法精度和性能上不一定能满足用户的需要。进一步, MapReduce的设计目标是解决数据密集但计算逻辑相对简单的批处理问题。然而, 大多数机器学习算法的计算流程都比较复杂, 模型的训练过程往往需要多次迭代计算(如梯度下降算法)、有中间数据集需要共享等。MapReduce模型处理这类问题时, 额外的调度和初始化开销导致其处理性能偏低。

为了克服MapReduce在计算性能上的缺陷, 不断地有新的并行化计算模型和框架出现, UC Berkeley AMP实验室推出的、目前已成为Apache开源项目的Spark^[12]系统是目前最有影响力的大数据处理平台之一。Spark使用基于内存计算的并行化计算模型RDD(resilient

distributed dataset)^[12], 提供了一个强大的分布式内存并行计算引擎, 实现了优异的计算性能, 同时还保持与Hadoop平台在底层分布式数据存储上的兼容性。在Spark执行引擎上, AMP实验室实现了很多机器学习算法, 并逐步整理成Spark下的一个机器学习算法库项目MLlib。MLlib目前也包含一些矩阵操作, 并希望基于矩阵的表示设计开发一些统计包和机器学习算法库。为了加速上层计算, MLlib底层通过Breeze使用了BLAS(basic linear algebra subprogram)单机的矩阵函数库。BLAS是很多更高层的数学函数库和数学编程语言(如LAPACK和MATLAB等)的基本构成单元。BLAS和LAPACK是当下普遍使用的线性代数函数库, 主要定义了一些线性代数中常用的运算操作和方程求解、矩阵分解的函数。此外, 基于MLlib的工作基础, UC Berkeley还计划研发MLBase^[9]项目, 该项目增加了机器学习模型自动选择和参数自动优化等功能, 提供了以目标为导向的高层机器学习方法接口。

提供机器学习算法库在一定程度上减轻了程序员进行机器学习算法设计的负担。但即使如此, 程序员编写程序时仍然需要熟悉具体的并行编程计算模型和平台, 而且由于实际数据分析应用的需求不同, 很多时候工具包所提供的通用算法在学习精度和计算性能上可能不能满足需求, 需要程序员定制和改进某个并行化机器学习算法或者开发新的算法, 这对普通数据分析程序员仍然是很大的挑战。

5.2 结合传统数据分析平台的大数据机器学习系统

虽然有了基于大数据处理平台的机器学习算法设计方法, 能够较好地解决一体

5
[http://mahout.
apache.org/](http://mahout.apache.org/)

化的大数据存储、计算以及并行化算法设计问题,但仅有机器学习系统仍然不能很好地解决各个平台对终端用户存在的可编程性和易用性不够的问题。为此,还需要进一步研究解决这些问题。

从可编程性和易用性角度来说,对于行业数据分析师,最熟悉使用的分析语言和环境通常是R、Python、MATLAB等系统。R是目前在数据分析应用领域最广为使用的数据分析、统计计算及制图的开源软件系统,提供了大量的专业模块和实用工具。为了尽可能缩小R语言环境与现有大数据平台间的鸿沟,工业界和研究界已经尝试在R中利用分布式并行计算引擎来处理大数据。最早的工作和系统RHadoop⁶,是由Revolution Analytics发起的一个开源项目,其目标是将统计语言R与Hadoop结合起来,目前该项目包括3个R package(包),分别为支持用R语言编写MapReduce应用的rmr、用于R语言访问HDFS的rhdfs以及用于R语言访问HBase的rhbase。其中,Hadoop主要用来存储和处理底层的海量数据,用R语言替代Java语言完成MapReduce算法的设计实现。

类似地,UC Berkeley AMP实验室在2014年1月也推出了一个称为SparkR的项目⁷。SparkR也是作为一个R的扩展包,为R用户提供一个轻量级的、在R环境里使用Spark RDD API编写程序的接口。它允许用户在R的shell环境里交互式地向Spark集群提交运行作业。

然而,目前的RHadoop和SparkR都还存在一个同样的问题:仍要求用户熟悉MapReduce或Spark RDD的编程框架和程序结构,然后将自己的MapReduce或Spark程序实现到基于R的编程接口上,这和Hadoop或Spark上写应用程序没有太大的区别,只是编程接口用R语言封装了一下。此外,这些工作都是基于单一平台,无

法解决跨平台统一大数据机器学习算法设计的问题。

5.3 基于特定平台的大数据机器学习系统

为了克服前述做法的缺陷,近年来,学术界和业界开始尝试总结机器学习算法设计的共同特性,结合大规模学习时所需要考虑的底层分布式数据存储和并行化计算等系统问题,专门研究能同时兼顾并支持大数据机器学习和大数据分布并行处理的一体化大数据机器学习系统。

在国内外的机器学习和大数据技术方面的会议中,已经频繁出现大数据机器学习系统相关的研究工作,如Spark MLlib、IBM的SystemML、Apache Flink、GraphLab、Parameter Server、Petuum等;此外,腾讯、百度等国内著名互联网企业也推出了不同的面向大数据的分布式机器学习系统,如腾讯的Peacock和Mariana大规模机器学习系统、百度的ELF和百度机器学习云平台BML。

(1) Spark MLlib

MLlib与Spark环境整合后可完整解决大数据的分布式存储、并行化计算以及上层的机器学习算法设计和使用问题,因此,可将其视为一个基本的机器学习系统。目前其已经向上层提供多种优化方法、分类算法、回归算法、推荐算法、聚类算法。但其算法数目和接口有限,难以完全满足用户各种各样的需求,且作为一个算法库,用户难以对内部算法进行深层定制优化。因此,其在灵活性方面仍然存在不足。

(2) SystemML

SystemML^[14,15]是由IBM Waston Research Center 和IBM Almaden Research Center联合研发的一款大数据

6
<https://github.com/RevolutionAnalytics/RHadoop>

7
Spark R. <https://github.com/amplab-extras/SparkR-pkg>

学习系统。对用户提供了一个类似于R语言的高层声明式语言,基于这种语言编写的程序可以被自动编译转化为MapReduce作业在Hadoop集群上运行。这种高层语言提供了大量的监督和非监督的机器学习算法所需要的线性代数操作算子,包括高层算子HOP (high-level operator) 和底层算子LOP (low-level operator)。SystemML最大的优势是其具有较好的可编程性和易用性,用户不需要具备任何分布式系统的概念或编程经验,即可利用其写出可扩展的机器学习算法;其最大的缺点在于底层计算平台单一,只采用了MapReduce作为底层分布式执行平台,而底层的MapReduce作业本身执行性能并不高,近年来涌现出的高效分布式计算框架如Spark、Flink等,在众多性能指标上远远高于MapReduce。

(3) Apache Flink

Apache Flink^[19]是由欧洲的多名研究者和多家资助单位(如EIT ICT Labs、DFG2、IBM、Oracle、HP等)联合研发的一款开源的并行化数据分析软件栈,项目早期的名称是Stratosphere。经过一段时间的发展,目前已经成为Apache开源项目。Flink同样意识到,数据分析人员在分布式系统上编写大数据分析算法时,需要有大量的精力耗费在分布式程序调试以及底层调优上。为了使数据分析人员能够在无需顾及各种并行化优化问题的情况下就可以进行深度数据分析,Flink提供了一款数据分析栈式软件。它向用户提供了一种专用的脚本式语言MeteorScript,并且会自动将基于该语言开发的程序转换到底层的分布式计算平台Nephele上^[20]。由于其专用性语言不够普及,目前系统对语言接口又做了更多的扩充,支持Java、Scala等语言。Flink项目的优势在于其从上到下提供了一整套完整的栈式解决方案,试图为

用户提供易于使用的数据分析系统,同时通过一些编译优化手段尽可能地提升程序执行的性能;其缺点在于其底层是专用的单一化计算平台,与目前普遍使用的主流大数据平台Hadoop和Spark尚未能集成使用,且上层的语言接口层也没有包含现在数据分析师广为使用的R、Python等语言接口。

(4) GraphLab

GraphLab^[16]是CMU开发的一个以顶点为计算单元的大规模图处理系统,是一个基于图模型抽象的机器学习系统。设计初衷主要是解决具有以下特点的机器学习问题:有局部依赖的稀疏数据集、迭代可收敛、异步执行。为了实现这个目标,GraphLab把数据之间的依赖关系抽象成Graph结构,以顶点为计算单元,将算法的执行过程抽象成每个顶点上的GAS (gather、apply、scatter)过程,其并行的核心思想是多个顶点同时执行。GraphLab的优点是能够高效地处理大规模图算法问题或者可归结为图问题的机器学习和数据挖掘算法问题;其缺点在于提供的接口细节比较复杂,对于普通的数据分析程序员而言,有较大的使用难度。

(5) Parameter Server与Petuum

很多机器学习算法常常要解决学习训练过程中模型参数的高效存储与更新问题。为了有效应对和满足大数据场景下这类机器学习算法的需要,研究者提出了一种称为Parameter Server的框架^[17],提供了一个分布式全局模型参数存储和访问接口,模型参数存储在多台服务器(server)中,工作节点(worker)可以通过网络访问API方便地读取全局参数。

Li Mu等人开发了一套基于Parameter Server框架的分布式机器学习系统^[17],该系统由一个服务器组(server group)和多个工作组(worker group)构成。其中,

服务器组中包括一个服务器管理(server manager)节点和多个服务器节点。每个服务器节点存储部分全局共享参数;服务器管理节点用来存储服务器节点的元信息,并通过心跳机制管理所有服务器。在该系统中,每个工作组包含一个任务调度器(task scheduler)和多个工作节点,工作节点只与服务器节点通信获取全局参数以及推送局部更新,不同的工作组可以同时运行不同的应用。Parameter Server的优点是为大规模机器学习提供了非常灵活的模型参数调优和控制机制;缺点是缺少对大规模机器学习时的数据及编程计算模型的高层抽象,使用较为繁琐,通常比较适合于机器学习算法研究者或者需要通过调整参数深度优化机器学习算法的数据分析程序员使用。

国际著名的机器学习专家、美国卡耐基梅隆大学机器学习系Eric Xing教授是大数据机器学习系统的倡导者。他认为,传统的机器学习研究过于简化机器学习算法本身的问题以及大规模数据机器学习存在的系统问题^[21]。由于大规模机器学习存在很多新的挑战,近几年来主要致力于大数据机器学习系统的研究,并领导其SAILING 实验室研究实现了大数据学习系统Petuum^[18, 21]。Petuum也是一个基于Parameter Server框架的系统,关键组件包括Parameter Server和调度器。Petuum的Parameter Server中使用SSP(stale synchronous parallel)一致性模型,允许计算速度慢的节点从本机缓存中读取陈旧的参数值,以此大大节省网络开销和网络等待时间,使得慢机器的大部分时间用在计算而不是等待上。Petuum在使用上需要算法设计者对分布式系统概念和系统有一定了解,其易用性仍有一定的限制,且其底层数据存储和并行计算框架与上层的学习算法层以紧耦合的方式全部自

己实现,系统实现复杂性和代价较大。

(6) 腾讯Peacock与Mariana深度学习平台

Peacock^[3]是腾讯公司研发的一个大规模LDA主题模型训练系统。该系统通过并行计算可对10亿×1亿级别的大规模矩阵进行分解,从而从海量文档样本数据中学习10万~100万量级的隐含语义。为了完成大规模处理,Peacock基于吉布斯采样的LDA训练算法进行了并行化设计,并设计实现了一个完整的具有大规模样本数据处理能力的训练系统。Peacock已广泛应用在腾讯的文本语义理解、QQ群推荐、用户商业兴趣挖掘、相似用户扩展、广告点击率转化率预估等多个业务数据中,是一个专为LDA并行化计算而定制的大规模训练系统,不是一个通用化的大数据机器学习系统。

为了提供更为广泛的大规模并行化机器学习处理能力,腾讯研究构建了一个称为Mariana的深度学习平台^[22],该平台由3套大规模深度学习系统构成,包括基于多GPU的深度神经网络并行计算系统Mariana DNN、基于多GPU的深度卷积神经网络并行计算系统Mariana CNN以及基于CPU集群的深度神经网络并行计算系统Mariana Cluster。Mariana可提供数据并行和模型并行计算,基于GPU和CPU集群提升模型规模,加速训练性能。其中,Mariana DNN在腾讯内部用于微信语音识别声学模型训练,可训练超过1万小时的语音数据、超过40亿的数据样本以及超过5 000万的参数,相对于单GPU而言,6GPU可实现4.6倍的加速比;而Mariana CNN用于微信图像识别,可训练2 000个以上的分类、300万以上的数据样本以及超过6 000万的参数,相对于单GPU而言,4GPU可实现2.5倍的加速比,在图文类效果广告点击率提升方面也取得初步应用;Mariana Cluster实现了一个基

于Parameter Server模型的大规模通用化机器学习和训练系统,主要用于进行大规模广告并行化训练,完成广告点击率预估模型训练和广告点击性能优化。

(7) 百度ELF与百度机器学习云平台BML

百度公司研发了一个大规模分布式机器学习框架和系统ELF (essential learning framework)^[23]。ELF是一个基于Parameter Server模型的通用化大规模机器学习系统,可允许用户方便快速地设计实现大数据机器学习算法,在系统设计上吸收了Hadoop、Spark和MPI等大数据平台的优点,用类似于Spark的全内存DAG计算引擎,可基于数据流的编程模式,通过高度抽象的编程接口,让用户方便地完成各种机器学习算法的并行化设计和快速计算。

在ELF的基础上,百度进一步开发了一个机器学习云平台BML (Baidu machine learning),该平台支持丰富的机器学习算法,可支持20多种大规模并行机器学习算法,提供包括数据预处理算法、分类算法、聚类算法、主题模型、推荐算法、深度学习、序列模型、在线学习在内的各种机器学习算法支持,并通过分布和并行化计算实现优异的计算性能。BML在百度内部的业务系统中经历了线上大规模使用部署考验,承载公司内各种重要的在线业务线应用,包括凤巢广告CTR 预估、搜索LTR排名等。

6 跨平台统一大数据机器学习系统Octopus的研究设计

6.1 Octopus的基本设计思想

上述绝大多数大数据机器学习方法和系统都是基于特定平台构建的,难以集成

和兼容现有和未来出现的多种大数据处理引擎和平台。现实世界中的各种大数据分析处理应用通常会有不同的分析处理需求和特征,例如,有些可能是极大规模数据的离线分析处理,有些可能是要求高实时性响应的联机分析处理,这些不同的分析需求要求底层有不同特性的大数据处理平台支持;此外,随着大数据处理技术和平台的不断发展,目前和未来不断有新的大数据编程方法和处理平台出现。因此,企业内需要将其原有平台上已经开发好的机器学习和数据分析算法,改写和移植到新的平台上,这将给企业带来很大的重复性劳动和开发负担。

为此,一个理想的大数据机器学习系统还需要拥有能够支持现有和未来出现的不同大数据处理平台的能力,实现跨平台大数据机器学习和数据分析算法的设计能力,达到“Write Once, Run Anywhere”的跨平台算法设计和运行目标。

针对大数据机器学习系统需要重点研究解决的计算性能以及可编程性与易用性问题,并考虑上述的跨平台需求特性,南京大学PASA大数据实验室正在研究设计一个跨平台大数据机器学习的统一编程模型和系统平台。该系统基于矩阵编程计算模型,结合R编程语言和编程方法,设计提供一个跨平台的统一编程计算框架,最终研究实现一个跨平台大数据机器学习系统Octopus (大章鱼)。

分析发现,机器学习和数据挖掘算法中的主体计算大多可表示为矩阵或向量运算,这通常也是算法中最耗时的部分。矩阵操作也是机器学习研究者用来描述问题和算法最为自然和常用的方式,基于矩阵可以表示和刻画大多数实际应用中涉及的机器学习和数据分析算法问题。基于这样的事实,为了给大数据机器学习和数据分析提供一种自然和统一的编程计

算模型和编程方法,类似于MapReduce中基于数据记录列表的抽象编程计算模型,将研究建立一种基于矩阵模型的抽象编程计算模型,以此作为大数据机器学习和数据分析算法设计的统一编程计算模型和接口。

Octopus是一个高层的大数据机器学习和数据分析统一编程模型和系统平台,允许数据分析和大数据应用开发人员轻松地设计和开发各种大数据机器学习和数据分析算法与应用程序。通过提供基于矩阵的统一编程计算模型,使用基于R语言的数据分析程序设计语言和程序设计方法,允许用户方便地编写和运行常规的R语言程序,而无需了解底层大数据平台的分布和并行化编程计算知识,使底层的分布并行计算框架和大数据平台对用户完全透明;底层平台上,通过良好的系统层抽象,可以快速集成Hadoop和Spark等通用大数据并行计算框架和系统平台,而且程序仅需编写一次,不需要有任何修改即可根据需要进行平滑运行于任何一个平台,从而实现“Write Once, Run Anywhere”的跨平台特性。

6.2 基于矩阵模型的统一编程和计算模型

分析发现,一方面,大数据机器学习算法中的主体计算很多可表示为矩阵或向量运算(向量可视为退化为1维的特殊矩阵),通常这也是算法中最耗时的部分;另一方面,矩阵操作通常也是机器学习研究者用来描述问题和算法最为自然和常用的方式。例如,训练样本特征值的归一化往往是通过将矩阵行或列求和后再除以该求和的值;KNN算法(最近邻算法)的核心操作是矩阵相减和矩阵元素求平方操作;PCA(主成分分析)算法可以通过矩阵相

乘或矩阵的SVD分解进行求解;最小二乘法的解析法涉及矩阵的相乘和求逆操作;互联网企业推荐系统中,相似性分析的核心是高达数亿用户乘以数亿物品的矩阵分解计算问题;而腾讯Peacock主题模型训练系统则涉及数十亿文档乘以数百万词汇的巨大矩阵计算问题;在信息检索领域,著名的PageRank算法也可以表示成矩阵的迭代相乘;社会网络中可以通过计算朋友关系矩阵的幂,从而进行基于共同好友的推荐;电子商务中可以通过对用户商品的购买记录进行矩阵分解,得到商品和用户的聚类并分析其潜在语义主题模型。很多其他的数据分析和大数据科学计算问题也常常要基于矩阵模型完成问题的描述和分析计算过程。因此,基于矩阵可以表示和刻画大多数实际应用中涉及的机器学习和数据分析算法问题。

基于这样的事实,为了给大数据机器学习提供一种统一的编程计算模型以及编程方法,类似于MapReduce采用了基于数据记录列表的抽象编程计算模型,研究建立了一种基于矩阵模型的抽象编程计算模型,以此作为大数据机器学习算法设计的统一编程计算模型和接口。

这种抽象矩阵编程计算模型将成为统一机器学习算法设计和编程计算的核心,它具有如下3个重要作用。

- 基于机器学习和数据挖掘算法中主体计算大多可表示为矩阵或向量运算的事实,为数据分析用户提供一种自然和统一化的大数据机器学习和数据分析算法建模和表示方法。
- 作为一个隔离和解耦上层数据分析算法与下层各种大数据平台的中间接口,实现底层平台对上层程序员及其程序的透明性,以此提高统一平台对上层程序员的易用性。
- 为实现统一大数据机器学习系统开

放式集成框架提供一种接口标准。针对任何一个拟集成使用的底层大数据平台,只要遵照所建立的统一矩阵编程接口,实现一个完成该接口程序计算任务的适配器或插件,即可以博采众长和兼容并蓄的开放式框架,连接和集成使用各种主流的大数据平台,使得上层的机器学习算法具有处理大规模数据的能力。

6.3 Octopus软件框架和系统设计

如图5所示,Octopus是一种基于良好的系统抽象的层次化系统。其最底层是分布式文件系统,如HDFS和分布式内存文件系统Tachyon,它们被用来存储和索引大规模的矩阵数据。在存储层之上,Octopus可以用多种大数据计算引擎和单机R引擎来执行不同规模的矩阵操作。

Octopus提供给用户的编程API是基于R语言的高层矩阵计算接口,基于该接口,用户不需要了解分布和并行计算系统知识,就可以很容易地设计实现大数据机器学习和数据分析算法或应用。

6.4 Octopus系统的技术特征

Octopus具有以下技术特征。

(1) 易于使用的高层编程API

Octopus提供给用户一组基于R语言的大规模矩阵运算API,称为OctMatrix。这些API提供各种大规模分布式矩阵计算操作,其形式上与标准R语言中的单机矩阵/向量操作API很相似。程序员可基于这些矩阵运算接口,用R语言快速编写各种机器学习和数据分析算法。除了最常用的大规模矩阵乘法,Octopus也提供了其他

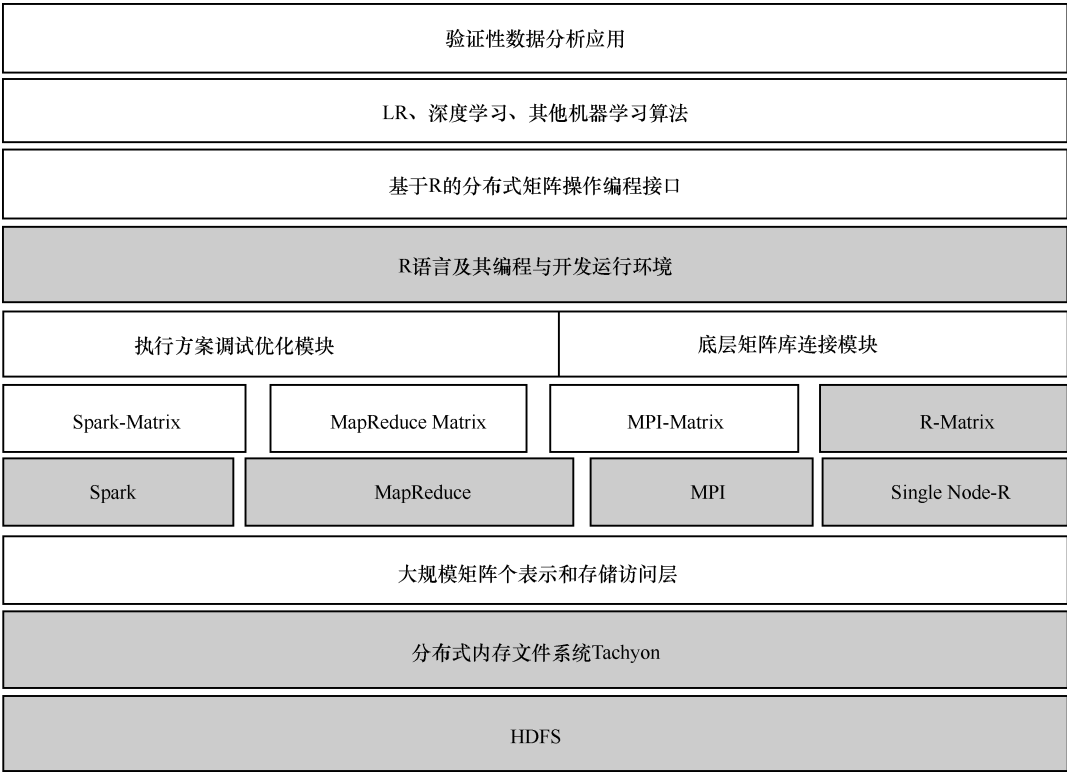


图5 Octopus (大章鱼) 软件系统框架

各种矩阵操作,如矩阵与矩阵的加法和减法、矩阵元素级别的乘法和除法、子矩阵运算等。这些API定义了高层矩阵操作符(operator)和操作(operation),因此熟悉R编程语言和编程方法的用户可以很容易地用其编程实现大数据机器学习和数据分析算法,且不需要了解底层大数据处理平台及其并行化编程方法。

(2) 一次编写,随处运行

用Octopus编写完成的算法和程序,可以运行在不同的底层大数据计算引擎和平台上。用OctMatrix API实现的机器学习和数据分析算法,用户可以在单机R上用小数据进行测试,不需要修改代码就可以用大数据在底层的大数据计算引擎和平台上执行,只需要简单切换底层的大数据计算引擎如Spark、Hadoop MapReduce或MPI即可。此外,Octopus通过提供通用接口可实现对底层不同文件系统的集成,并实现基于不同文件系统的大规模矩阵数据存储访问接口,包括HDFS、Tachyon以及基于单机本地的文件系统。

(3) 无缝融合R生态系统

Octopus可运行于标准的R环境下,实现与R环境的无缝融合,因此可以利用R生态系统中的丰富资源,比如第三方R包。除了传统的矩阵/向量函数,Octopus也在OctMatrix中提供apply函数,传入apply函数的参数可以是任意的R函数,包括UDF(user defined function)。当OctMatrix运行在分布式环境中时,函数参数可以在集群中被应用于OctMatrix的每个元素、每行或每列,函数参数在集群的每个节点并行执行。

6.5 Octopus中分布式矩阵计算优化

大规模数据通常会导致整个数据分析计算时间过长,难以在可接受的时间内

完成分析任务,为此需要借助于分布式和并行化计算技术实现大规模数据分析计算时的加速和性能提升,以提高数据分析的响应时间。由于采用大规模矩阵作为大数据机器学习和数据分析的编程计算接口,在底层大数据平台上的大规模矩阵运算的性能将直接关系到上层算法执行的性能。为此,大规模矩阵计算性能优化是需要重点研究解决的关键技术问题之一。

在所有矩阵计算中,矩阵乘法是使用最多且最为耗时的计算,许多矩阵的因子分解操作也都可以由矩阵乘法近似实现。因此,矩阵乘法的优化是整个矩阵计算优化中最为重要的问题。两个大规模矩阵进行分布和并行化相乘运算时,一个重要问题是如何合理划分矩阵数据,以便利用多个计算节点分布和并行化地完成计算。不同大小和形状的矩阵,其划分方法可能会造成计算性能上的差异,例如,方形矩阵与长条形矩阵需要考虑不同的划分方法,而一个大矩阵与一个小矩阵相乘时,当小矩阵可存放在单节点的内存中时,对小矩阵可不进行划分,而改用广播方式将小矩阵发送到每个计算节点上,以此完成两个矩阵的运算,这样可以避免大量的网络数据传输。因此,需要根据矩阵的大小和形状,合理划分矩阵,以便以最小的计算代价完成乘法运算。

除了不同形状和大小的矩阵划分策略优化外,被划分后落到每个计算节点上分布处理的子矩阵计算也需要进行优化。矩阵乘法是典型的计算密集型任务并且存在许多单机的高性能的线性代数库,如BLAS、Lapack和MKL。由于在JVM中执行线性代数计算性能较低,Octopus将计算密集的矩阵计算从JVM中通过JNILoader装载到本地线性代数库(如BLAS、Lapack)中执行,这样可显著加速

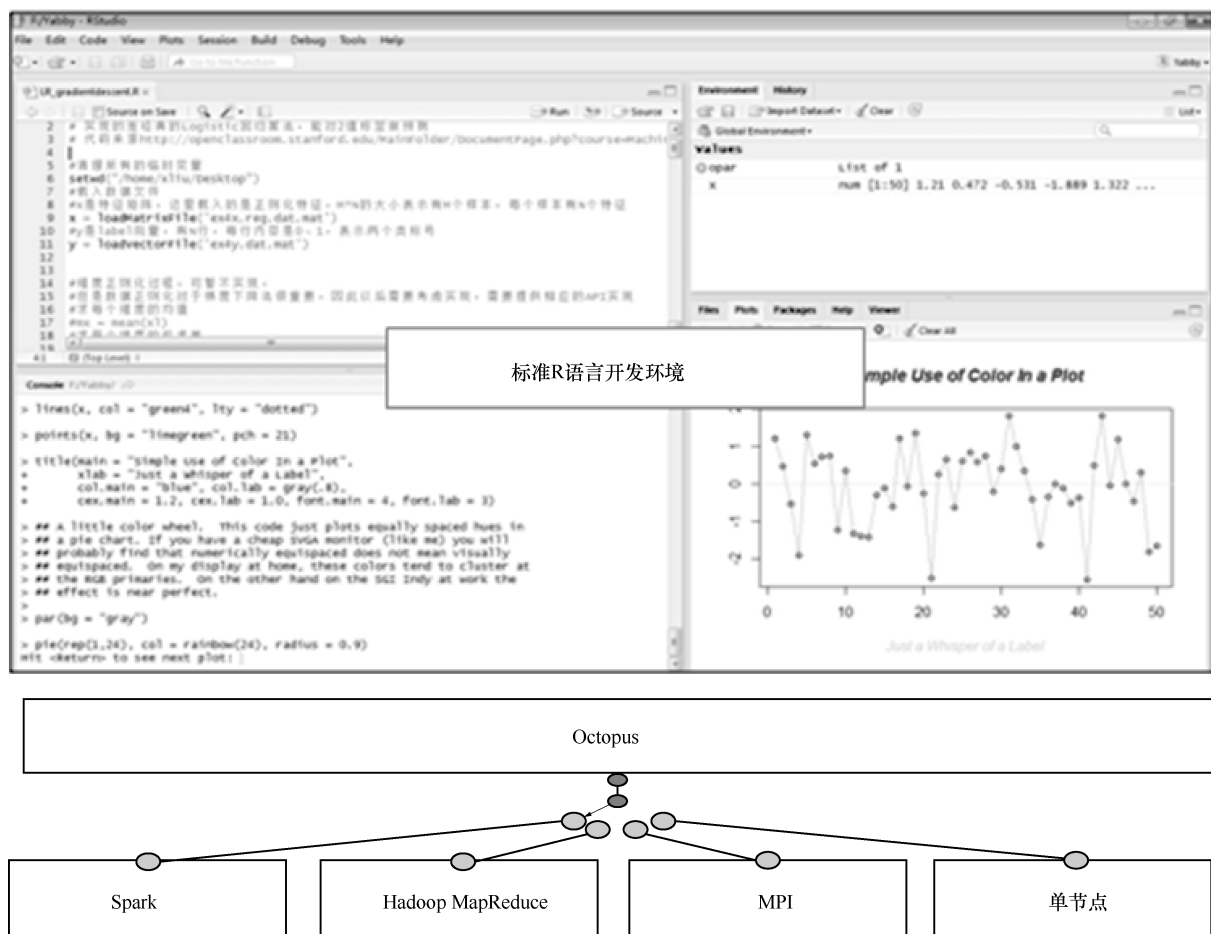


图6 基于R语言和Octopus的跨平台统一大数据机器学习系统

子矩阵的计算性能。

6.6 Octopus系统的编程使用

图6显示了Octopus系统的使用方式。Octopus使用标准的R编程和开发环境，允许用户使用R语言，并基于大规模矩阵计算模型编写各种机器学习和数据分析算法。系统可实现与Spark、Hadoop MapReduce和MPI的集成，底层可无缝切换运行于不同的大数据平台上。基于Octopus所设计实现的算法或程序代码，无需修改即可平滑切换并运行于上述任意一个底层大数据计算引擎和平台上，只要简单地修改程序代码中的计算引擎类型选

择参数即可。

基于R语言和初步研究设计的Octopus，设计实现了多个典型的机器学习算法，包括LR、SVM、聚类、深度学习等，以验证研究设计的原型系统的有效性。

图7是基于常规R语言所实现的Linear Regression算法与基于Octopus所实现的Linear Regression算法的示例代码。由代码比较可见，两者在形式上非常接近，仅在有关矩阵运算调用的API上有微小差异。

由图7可见，在R语言中基于矩阵进行机器学习的算法设计较为简洁和方便，Linear Regression代码仅仅需要20行左右即可实现完整的算法。


```

1 linear_regression <- function(x, label)
2 {
3   x <- cbind(x, 1)
4   beta <- solve(t(x) %*% x) %*% t(x) %*% label
5   class(beta) <- "lm_param"
6   beta
7 }
8
9
10 predict.lm_param <- function(model, x)
11 {
12   x <- cbind(model, 1)
13   x %*% model
14 }

```

(a) 基于常规R语言

```

1 linear_regression <- function(x, label)
2 {
3   x <- cbind2(x, ones(ncol = 1, nrow = dim(x)[1]))
4   beta <- inv(t(x) %*% x) %*% t(x) %*% label
5   res <- list("param" = beta)
6   class(res) <- "oct_lm_param"
7   res
8 }
9
10
11 predict.oct_lm_param <- function(model, x)
12 {
13   x <- cbind2(x, ones(ncol = 1, nrow = dim(x)[1]))
14   x %*% (model$param)
15 }

```

(b) 基于Octopus

图7 基于Octopus和常规R语言的Linear Regression算法代码比较

7 结束语

近几年来,大数据技术的发展推动了大数据机器学习和智能计算技术的发展热潮。大数据机器学习不仅是一个单纯的机器学习问题,更是一个大规模的复杂系统问题;是一个同时涉及机器学习和大数据处理两个领域的交叉研究课题。要实现有效的大数据机器学习处理,需要构建一个能同时支持机器学习算法设计和大规模数据处理的一体化大数据机器学习系统。

本文介绍了国内外大数据机器学习系统的基本概念、基本研究问题、技术特征、

系统分类以及典型系统。在此基础上,进一步介绍了研究设计的跨平台统一大数据机器学习原型系统Octopus。基于大多数机器学习和数据分析算法可表示为矩阵运算的事实,Octopus采用矩阵模型作为大数据机器学习和数据分析抽象编程计算模型,提供了一个基于矩阵的高层编程模型和接口,并基于R语言和开发环境向用户提供了一个矩阵运算R扩展包,为用户提供可扩展性好且易于使用的矩阵运算操作,允许用户基于所提供的大规模矩阵运算操作,快速设计实现各种机器学习和数据分析算法。Octopus能在底层无缝地集成和使用不同的大数据计算引擎和平台,完成大数据机器学习算法的分布和并行化执

行,并支持单节点R环境以及Spark、Hadoop MapReduce和MPI等多种大数据计算引擎和平台,并能实现这些平台间的无缝切换,实现“Write Once, Run Anywhere”的跨平台特征。就我们所知,Octopus是目前世界上第一个具有跨平台特性,同时还能实现底层大数据平台对上层程序员透明化的大数据机器学习系统研究工作。

正如CCF大数据专家委员会发布的2014年《中国大数据技术与产业发展白皮书》中所说,目前大数据机器学习系统尚处在一个初期的探索和研究阶段,尽管国内外已经有不少研究开发工作,但研究设计高效、可扩展且易于使用的大数据机器学习系统仍面临诸多的技术挑战。因此,大数据机器学习系统将是目前和未来几年的热点研究领域,工业界和学术界都将持续地投入相当多的资源进行深入的研究开发工作。

同样,虽然初步研究设计了跨平台大数据机器学习系统Octopus,但其中仍然有大量需要进一步深入研究解决和完善的技术问题,例如大规模矩阵运算的深度优化、稀疏矩阵的存储管理和计算优化、异构大数据处理平台环境下不同矩阵计算时的平台自动选择、基于矩阵计算表达式和计算流图的计算优化等。此外,仅有矩阵计算模型还不能满足所有的大数据机器学习计算需求,还需要考虑其他计算模型,如图模型和参数模型(parameter server)的混合使用,形成一个能满足各种大数据机器学习算法设计需求的综合系统。

致谢

本文工作受到江苏省科技支撑计划基金项目(No.BE2014131)和江苏省软件技

术产业协同创新中心的资助。

参考文献

- [1] Banko M, Brill E. Scaling to very large corpora for natural language disambiguation. Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL), Toulouse, France, 2001: 26~33
- [2] Brants T, Popat C A, Xu P, *et al.* Large language models in machine translation. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning, Prague, Czech Republic, 2007: 858~867
- [3] Wang Y, Zhao X M, Sun Z L, *et al.* Peacock: learning long-tail topic features for industrial applications. ACM Transactions on Intelligent Systems and Technology, 2014, 9(4)
- [4] 中国计算机学会大数据专家委员会. 2015年中国大数据发展趋势预测. 中国计算机学会通讯, 2015, 11(1): 48~52
CCF Task Force on Big Data. Forecast for the development trend of big data in 2015. Communications of the China Computer Federation (CCCF), 2015, 11(1): 48~52
- [5] Gonzalez J E. Emerging systems for large-scale machine learning. Proceedings of Tutorial on International Conference for Machine Learning(ICML) 2014, Beijing, China, 2014
- [6] 中国计算机学会大数据专家委员会. 2014年中国大数据技术与产业发展白皮书. 2014中国大数据技术大会, 北京, 中国, 2014
CCF Task Force on Big Data. White paper of China's big data technology and industrial development in 2014. Proceedings of Big Data Conference China, Beijing, China, 2014
- [7] Boehm M, Tatikonda S, Reinwald B, *et al.* Hybrid parallelization strategies

- for large-scale machine learning in systemML. Proceedings of the VLDB Endowment, Hangzhou, China, 2014
- [8] Markl V. Breaking the chains: on declarative data analysis and data independence in the big data era. Proceedings of the VLDB Endowment, Hangzhou, China, 2014
- [9] Kraska T T. MLbase: a distributed machine-learning system. Proceedings of the 6th Conference on Innovative Data Systems Research(CIDR), Asilomar, CA, USA, 2013
- [10] Fan W F, Geerts F, Neven F. Making queries tractable on big data with preprocessing: through the eyes of complexity theory. Proceedings of the VLDB Endowment, Trento, Italy, 2013: 685~696
- [11] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. Communications of the ACM, 2004, 51(1): 107~113
- [12] Zaharia M, Chowdhury M, Das T, *et al.* Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation(NSDI), San Jose, CA, USA, 2012: 141~146
- [13] Venkataraman S, Bodzsar E, Roy I, *et al.* Presto: distributed machine learning and graph processing with sparse matrices. Proceedings of the 8th ACM European Conference on Computer Systems(EuroSys), Prague, Czech Republic, 2013: 197~210
- [14] Ghoting A, Krishnamurthy R, Pednault E, *et al.* SystemML: declarative machine learning on MapReduce. Proceedings of International Conference on Data Engineering (ICDE), Hannover, Germany, 2011: 231~242
- [15] Boehm M, Tatikonda S, Reinwald B, *et al.* Hybrid parallelization strategies for large-scale machine learning in SystemML. Proceedings of the VLDB Endowment, Hangzhou, China, 2014
- [16] Low Y, Bickson D, Gonzalez J, *et al.* Distributed graphLab: a framework for machine learning and data mining in the cloud. Proceedings of the VLDB Endowment, Istanbul, Turkey, 2012: 716~727
- [17] Li M, Andersen G D, Park W J, *et al.* Scaling distributed machine learning with the parameter server. Proceedings of Operating Systems Design and Implementation (OSDI), Broomfield, CD, USA, 2014: 583~598
- [18] Ho Q, Cipar J, Cui H, *et al.* More effective distributed ml via a stale synchronous parallel parameter server. Proceedings of Advances in Neural Information Processing Systems (NIPS), Nevada, USA, 2013: 1223~1231
- [19] Alexandrov A, Bergmann R, Ewen S, *et al.* The stratosphere platform for big data analytics. Vldb Journal, 2014, 23(6): 939~964
- [20] Battré D, Ewen S, Hueske F, *et al.* Nephele/PACTs: a programming model and execution framework for web-scale analytical processing. Proceedings of ACM Symposium on Cloud Computing(SoCC), Indianapolis, Indiana, USA, 2010: 119~130
- [21] Dai W, Wei J, Zheng X, *et al.* Petuum: a framework for iterative-convergent distributed ML. Proceedings of Advances in Neural Information Processing Systems 26, Big Learning Workshop, California, USA, 2013
- [22] 邹永强. Mariana—腾讯深度学习平台的进展与应用. 2014年中国大数据技术大会, 北京, 中国, 2014
- Zou Y Q. Marina—the progress and application of deep learning platform of Tencent. Proceedings of Database

Technology Conference China 2015,
Beijing, China, 2014
[23] 刘伟. 百度机器学习云平台. 2015年中国数据
库技术大会, 北京, 中国, 2015

Liu W. Machine learning cloud platform
of Baidu. Proceedings of Database
Technology Conference China 2015,
Beijing, China, 2015

作者简介



黄宜华, 男, 博士, 南京大学计算机系教授、博士生导师, 中国计算机学会大数据专家委员会委员、副秘书长, 江苏省计算机学会大数据专家委员会主任, CCF高级会员, 南京大学PASA大数据技术实验室学术带头人。主要研究方向为大数据并行处理、大数据机器学习、Web信息抽取与挖掘技术, 在国内外学术刊物和国际会议上发表学术论文60多篇, 撰写并出版大数据处理方向书籍两部, 是近年来出版的《深入理解大数据—大数据处理与编程实践》一书的作者。目前正在大数据并行化算法、大数据系统平台和应用方面开展深入的研究工作, 主持或参与多项国家级和省部级科研项目, 并在大数据领域开展了与Intel、UC Berkeley AMP Lab、微软亚洲研究院、百度、中兴通讯等国内外知名企业和研究机构的合作研究工作。

收稿日期: 2015-05-03; 修回日期: 2015-05-11
基金项目: 江苏省科技支撑计划基金资助项目 (No.BE2014131)
Foundation Item: Jiangsu Province Science and Technology Support Program (No.BE2014131)
论文引用格式: 黄宜华. 大数据机器学习系统研究进展. 大数据, 2015004
Huang Y H. Research progress on big data machine learning system. Big Data Research, 2015004