分布式环境下话题发现算法性能分析

邓璐[1]，贾焰[1]，方滨兴[2]，周斌[1]，张涛[1]，刘心[1]
（1. 国防科技大学计算机学院，湖南 长沙 410073；2. 北京邮电大学计算机学院，北京 100876）

摘 要：社交网络成为现在人们生活的一种重要方式，越来越多的人选择通过社交网络表达观点、抒发心情。在海量的数据下，快速发现讨论的内容得到越来越多的研究者的关注，随即出现了大量的话题发现算法。在大规模新浪微博数据环境下，针对 3 种经典分布式话题发现算法，结合社交网络平台的特点提出了分析性能的测试方案，并根据测试方案比较与分析了 3 种算法的性能，指出了各算法的优缺点，为后续应用提供参考。

关键词：话题发现；分布式环境；社交网络；性能分析
中图分类号：TP391
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2018136

Performance analysis of topic detection algorithms in distributed environment

DENG Lu[1], JIA Yan[1], FANG Binxing[2], ZHOU Bin[1], ZHANG Tao[1], LIU Xin[1]
1. College of Computer, National University of Defense Technology, Changsha 410073, China
2. College of Computer, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract: Social network has become a way of life, therefore more and more people choose social network to express their views and feelings. Quickly find what people are talking about in big data gets more and more attention. And a lot of related methods of topic detection spring up in this situation. The performance analysis project was proposed based on the characteristics of social network. According to the project, the performances of some typical topic detection algorithms were tested and compared in large-scale data of Sina Weibo. What's more, the advantages and disadvantages of these algorithms were pointed out so as to provide references for later applications.

Key words: topic detection, distributed environment, social network, performance analysis

1 引言

互联网正在逐步发展为无处不在的计算平台和信息传播平台，这为基于互联网的社交网络服务应用的快速发展提供了契机。中国互联网络信息中心发布的统计信息显示：截至 2017 年 12 月，中国的网民规模达到了 7.72 亿[1]。这些用户每天浏览、关注、发布数以千万计的信息，因此，如何在如此巨大的数据规模下快速发现人们讨论的内容是一个基础和流行的研究问题。

很多研究都是基于这个问题展开的，且大多数已有工作都来源于话题发现和追踪（TDT, topic detection and tracking）这个任务。话题发现自 1996 年以来就是 TDT 的子任务之一，它是将新闻、广播、TV 和其他媒体的信息流看作处理对象，将信息流中涉及的事件放入相关话题的过程。发展初期的信息通常由官方媒体发布，因此数据规模通常较小，单个机器节点就可以快速地完成处理。由于互

收稿日期：2017−11−08；**修回日期**：2018−07−03
基金项目：国家自然科学基金资助项目（No.61502517, No.61472433, No.61732004, No.61732022）；国家重点研发计划课题基金资助项目（No.0708068118002, No.2017YFB0803303）
Foundation Items: The National Natural Science Foundation of China (No.61502517, No.61472433, No.61732004, No.61732022), The National Key Research and Development Program of China (No.0708068118002, No.2017YFB0803303)

联网的快速发展以及移动终端使用的普及，各类社交媒体得到越来越多人的青睐，因此相关学者将研究重点放到邮件、论坛、微博等新型媒体。这类媒体的一个重要特征就是任意用户都可以发布信息内容，不再局限于官方媒体，而是由用户共同营造话题，用户的参与感更强。这就导致了话题发现任务需要处理的数据规模大幅度增加，大多数情况下单个机器节点已经很难满足快速完成任务的要求，分布式话题发现方法或将话题发现方法并行化是现在话题发现任务的必然趋势。

本文首先回顾了主流的话题发现算法，在此基础上，选择了 3 种代表性的分布式话题发现算法：MrLDA 算法，Mahout LDA 算法和 Spark LDA 算法，简单介绍了 3 种算法的工作原理。然后以运行时间和加速比为衡量指标，测试 3 种算法在不同数据规模、不同集群规模、不同迭代次数下的执行情况。最后对 3 种算法的性能进行对比分析，指出各自的优缺点，为后续应用提供参考。此项研究在实际的工程应用中有很大意义。

2　主流话题发现算法介绍

话题发现对于提供有效网络舆情信息、舆情监控和竞争情报等方面具有重大意义。各类话题发现方法都基于特定话题模型，而话题模型是研究话题发现的基础，目前主要有以下 2 类话题发现方法。

第一类是基于向量空间模型的方法。它是相对传统的模型化文档方法，将文档看作词袋，每个文档看作词汇空间中的向量。TFIDF 方法是用来衡量词项权重比较常见的方法，它的基本思想是,如果某个词汇或短语在一篇文档中出现的频率较高，并且在其他文档中很少出现，则认为该词汇或短语具有很好的类别区分能力，适合表示文档。2 个文档的相似性可以通过多种方法计算，一般采用余弦相似性方法，通过 2 个文档向量夹角的余弦值衡量。很多基于距离的方法可以被应用到话题发现中，比如 Spherical k-Means algorithm (Sk-Mean)[2]、Fuzzy Spherical k-Means (FSk-Means)[3-4]等。Makkonen 等[5]将单一的事件向量分解为 4 个子向量，用 4 种不同类型的词汇表征，分别是人物机构指示词、地点位置指示词、时间日期指示词和事件指示词。将时间表达式进行形式化，并利用本体知识对地点信息进行扩展，进而应用在话题发现中。Wu 等[6]提出一个重建文本向量的高效方法检测新话题，该方法通过 Jaccard 相似系数和逆向频率计算文本向量每一维的重要程度，基于重要程度重建文本向量，提高了文本聚类和关键话题抽取的准确性。

第二类是基于概率模型的方法。概率话题模型[7-10]是发现隐藏话题的有效统计模型，得到越来越多研究者的青睐。其中，最具有代表性的是 LDA 模型[8]。它是在 pLSI 模型的基础上改进的，认为文档是在话题上的多项式分布，话题是在词汇上的多项式分布，从而构造了文档的产生式模型：依据概率分布依次抽取话题和词汇，迭代地产生出文档中的每一个词汇，需要参数估计的方法得到最终模型，比如 variational inference[8, 11]、collapsed Gibbs sampling[12]等。该模型也成功地应用于 Twitter 数据，Ramage 等[13]提出一种基于标签 LDA（labeled LDA）的半监督学习模型，可将 Twitter 消息按主旨、风格、状态和社会角色这 4 个维度进行分类，以便用户浏览感兴趣的信息。Chen 等[14]采用 lifelong learing 方法，在训练模型时融入 must-link 集合和 cannot-link 集合：1) 对于多个文档集合利用经典的话题模型得到每个文档对应的话题集合，基于该集合获取词汇间的 must-link 集合；2) 在测试数据上将 must-link 集合融入基于知识的话题模型（KBTM, knowledge-based topic model），根据求得的话题集合抽取其中的 cannot-link 集合；3) 将 must-link 集合和 cannot-link 集合重新应用到测试数据中，得到最终的话题集合。Lin 等[15]提出一个双稀疏话题模型，该模型同时考虑了文档—主题分布的稀疏性和主题—词汇的稀疏性，是建立在一篇文档一般只包含几个主题、一个主题所使用的词汇也相对有限而不是分布在整个词汇表的基础上，将文档的生成过程依托于固定主题集合和固定词汇集合。

相对于基于向量空间模型的方法，基于概率模型的方法的应用更为广泛，特别是基于 LDA 模型的方法，在话题发现、情感分析等多个领域都得到了认可。本文针对 LDA 模型，选择了 3 种典型的分布式算法：MrLDA 算法[16]，Mahout LDA 算法[17]和 Spark LDA[18]，通过分析 3 种算法在不同条件下的性能情况，评估不同算法的优势。

3　3 种算法的原理

LDA 模型[8]是一个三层结构的贝叶斯模型，是处理语料库中文档的产生式概率模型。文档由随机潜在的话题表示，话题则是词汇上的分布。LDA 模

型的具体描述如下：一篇文档 $d=w_1,\cdots,w_N$ 是由数量为 N 的词汇组成，话题分布 θ_d 是基于参数 α 的 Dirichlet 分布，文档 d 中词汇对应的话题序列 $z=z_1,\cdots,z_N$ 是基于话题分布 θ_d 产生的，任意词汇 w_i 是根据分布 $p(w_i|z_i,\beta)$产生的。参数 α 和 β 是先验参数，整个语料库采样一次。参数 θ 是文档级别的变量，每个文档采样一次。变量 z 和 w 是词汇级别的变量，每个文档下的每个词汇采样一次。在完成 LDA 模型的训练后，可以通过推导方法求取文档的话题分布。

3.1 MrLDA 算法原理

MrLDA[16]是基于 Hadoop MapReduce 框架实现的大规模分布式 LDA 话题发现模型。其主要思想是通过对变分法的分布式化计算变分参数，迭代计算全局 LDA 模型的参数，从而提高 LDA 模型在分布式环境下的运行效率。

MrLDA 算法的迭代过程如图 1 所示，分为 3 个部分：在并行的 Mapper 中计算特定文档的变分参数；在并行的 Reducer 中计算特定话题参数；在 Driver 中更新全局参数，同时监控算法是否收敛，判断是否结束迭代。

3.2 Mahout LDA 算法原理

Mahout 是一个强大的数据挖掘工具，也是一个分布式机器学习方法的集合。Mahout LDA[17]基于 Hadoop MapReduce 框架，将变分法和 Gibbs 采样相结合来计算参数，提高了算法可处理的数据量级和算法本身的性能。

Mahout LDA 算法的本质是贝叶斯公式和 EM 算法的结合。Mahout 程序利用 CVBo 算法来计算 LDA 模型，在 Map 过程中对向量 ***docTopic*** 和矩阵 ***docTopicModel*** 反复迭代求解，算出每个文档的 ***docTopicModel*** 并且在更新 writeModel 阶段将 ***docTopicModel*** 矩阵进行向量的相加。执行完所有的 Map 过程后将整个数据集的 ***docTopicModel*** 聚合，最终在 cleanup 过程中将话题的索引作为 key 值，矩阵 ***docTopicModel*** 作为 value 值写入 Reduce 过程。

3.3 Spark LDA 算法原理

Spark LDA[18]在 Spark 机器学习库 MLlib 上实现了 2 个版本的 LDA，分别为 Spark EM LDA 和 Spark Online LDA。Spark EM LDA 建立在 Spark 框架下，通过 GraphX 实现的 LDA 模型的算法，利用对图的边和顶点数据的操作训练模型，并使用 Gibbs 采样原理估计模型参数，将训练的话题—词汇模型存储在 GraphX 图顶点上，属于分布式存储方式。Spark Online LDA 建立在抽样模式的基础上，每次训练模型是通过抽取一些文档实现的，最终模型是多次训练后的结果，参数估计采用贝叶斯变分的方法，利用矩阵存储话题—词汇模型，属于本地模型。Spark EM LDA 在训练时，shuffle 量非常大，极大地影响速度，同时，每轮迭代完毕后更新模型，导致收敛速度较慢。Spark Online LDA 使用矩阵存储模型，矩阵规模直接限制训练文档集的主题数和词的数

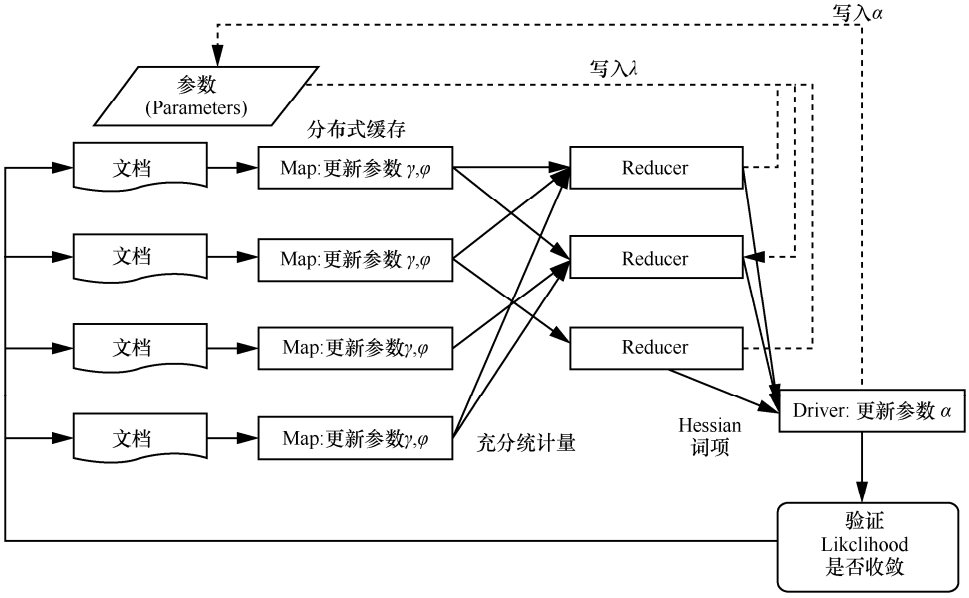


图 1 MrLDA 算法原理

目，且在每次训练完抽样文本后更新模型。因而 Spark Online LDA 模型更新更及时，收敛速度更快。Online LDA Optimizer 通过在小批量数据上迭代采样实现 Online 变分推断，比较节省内存。而 EMLDAOptimizer 得到的结果是建立在整个数据集基础上的，更为全面，所以这里选择 Spark EM LDA 进行实验。

Spark EM LDA 实现的核心是 GraphX 以文档到词汇作为边，以词频作为边数据，把语料库构造成图，把对语料库中每篇文档的每个词汇的操作转化为对图中每条边上的操作。GraphX 把文档—话题矩阵和话题—词汇矩阵存储在文档顶点和词汇顶点上，把词频信息存储在边上。它把整个文档的聚类结果矩阵、模型矩阵和语料库词频矩阵都表示在图结构中，将 LDA 算法处理过程转化为对边的遍历处理过程。

4　测试计划

4.1　测试环境

本文实验是在腾讯云上实现的，租用 128 台服务器节点，如图 2 所示。每个节点的软件环境如下：CentOS6.5, Ubuntu14.04, Jdk1.8, Hadoop2.6,0, Spark1.6.2。其中，有一个主节点的配置是 8 核处理器，64 GB 内存，500 GB 硬盘，1 Mbit/s 带宽；其余节点的配置是 8 核处理器，32 GB 内存，100 GB 硬盘，1 Mbit/s 带宽。

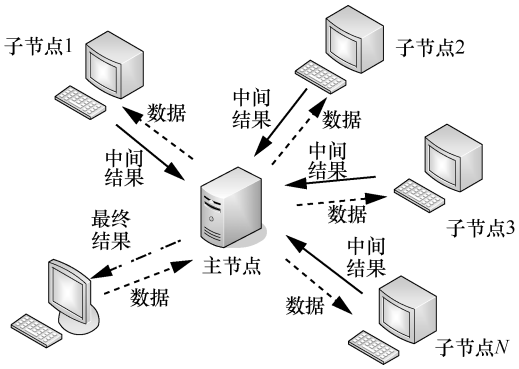


图 2　测试分布式环境示意

4.2　数据集

测试数据来自新浪微博的真实数据，爬取了 2015 年 8 月的微博内容，经过预处理，去除字数较少的博文，最终得到约 1 900 万条左右的博文数据，数据规模约为 12 GB。3 种算法都是基于 LDA 模型的，所以涉及话题数目 K 的设定，这里统一将话题数目 K 设置为 20。

4.3　测试指标

本文主要从加速比和运行时间这 2 个指标对 3 种不同的分布式话题发现算法进行测试。

加速比：测试 3 种算法在不同集群规模上的加速比。

运行时间：测试 3 种算法在不同数据规模、不同迭代次数、不同集群规模上从开始运行到结束运行的时间之差。

4.4　测试方案

1) 集群规模：随着集群规模的增加，算法的运行时间会呈现一定程度的减小，但是否会随着集群规模的不断增加，呈现不断减小的趋势是一个值得探索的问题。根据算法的实际情况，有选择地对 3 种算法节点数设定为 1、4、8、16、32、64、128 的集群规模，研究在同一数据规模和迭代次数条件下算法的运行时间和加速比。

2) 数据规模：计算 3 种算法执行不同规模测试数据的运行时间。由于 3 种算法的执行原理不同，对数据规模的适用情况也会有所差异，数据规模的选取可能会对算法的性能产生影响。分别对 3 种算法选取 100 万、1 000 万和 1 亿条博文的数据规模，研究在同一迭代次数和集群规模条件下算法的运行时间。

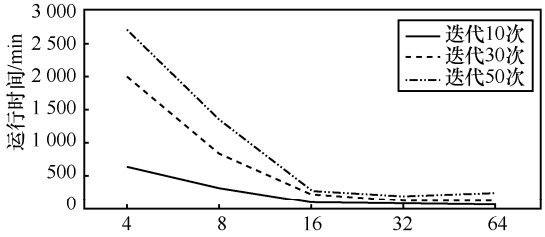
3) 迭代次数：3 种算法都需要反复迭代处理，迭代次数太少会影响产生话题的质量，迭代次数过多会影响算法效率，不同的迭代次数对算法的性能有一定的影响。分别对 3 种算法设定 10、30、50 的迭代次数，研究在同一数据规模和集群规模条件下的运行时间。

5　3 种算法的性能

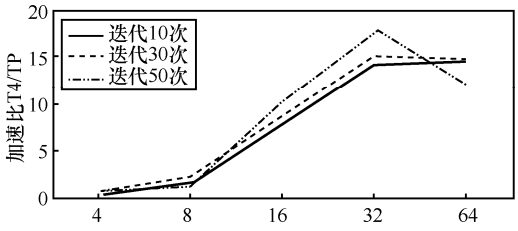
5.1　MrLDA 算法性能

由于在单个节点上运行 MrLDA 算法的时间过长，因此集群规模最小设定为 4 个服务器节点。这里展示了 MrLDA 算法在百万数据级别，不同迭代次数和不同集群规模下的运行时间和加速比，如图 3 所示。

1) 集群规模：从运行时间这个指标来看，对于 100 万条博文规模的数据，MrLDA 算法在 16 个节点时出现量级上的降低，之后时间降落没有很明显。从加速比指标来看，加速比在 32 个节点时，都达到最高值或变化不大。



(a) 算法运行时间随集群规模的变化情况



(b) 算法加速比随集群规模的变化情况

图 3　MrLDA 算法百万级别下的测试

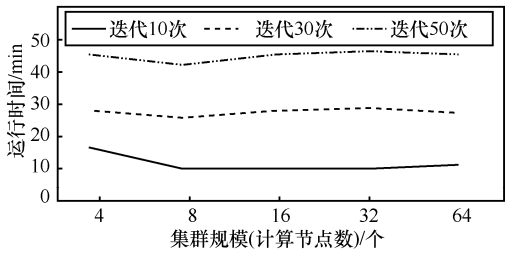
2) 数据规模：MrLDA 算法由于自身算法的局限性，没有基于内存消耗做优化处理，因此实际能够处理的数据量有限，在执行千万级别以上的数据规模时会出现错误警告，不适合处理超过百万级别的数据规模。

3) 迭代次数：随着迭代次数的增加，运行时间呈现变大趋势。迭代次数越大时，随着集群规模的增加，其相应运行时间的下降幅度越大，加速比的上升趋势越明显。MrLDA 算法在 30 次迭代时未出现收敛，在 50 次迭代时出现收敛。

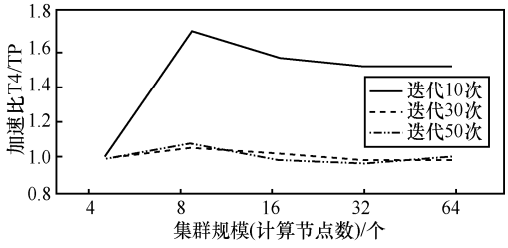
5.2　Mahout LDA 算法性能

与 MrLDA 算法类似，Mahout LDA 在单个服务器节点时间同样过长，因此集群规模最小设定为 4 个服务器节点。这里分别展示了 Mahout LDA 算法在百万、千万、亿级别的数据规模，不同迭代次数和不同集群规模下的运行时间和加速比，如图 4~图 6 所示。

1) 集群规模：Mahout LDA 在百万条博文规模上 8 个节点的运行时间最短，加速比最大。对于数据量为百万级别的博文，最优集群规模是 8 个节点左右。而以千万条规模和亿条规模条件为前提时，算法在 64 个节点处出现了拐点，由于数据规模为百万级别时，过早出现了性能的“瓶颈”，因此只测试到 64 台服务器节点规模。而数据规模为千万和亿级别时，测试到 64 台服务器节点时，运行时间仍呈现下降趋势，加速比仍出现上升趋势，所以对于这 2 个数据规模级别，测试了 128 台服务器节点下的执行情况。

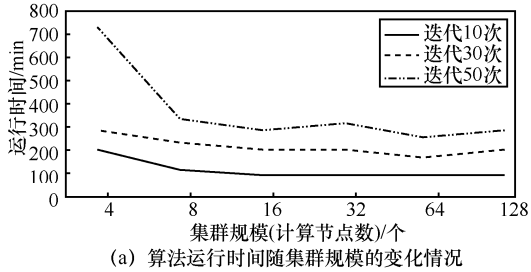


(a) 算法运行时间随集群规模的变化情况

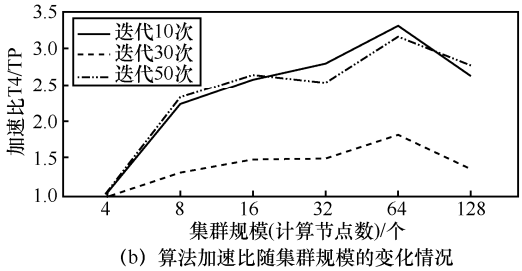


(b) 算法加速比随集群规模的变化情况

图 4　Mahout LDA 算法百万级别下的测试

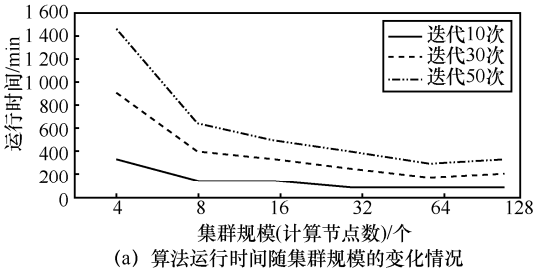


(a) 算法运行时间随集群规模的变化情况

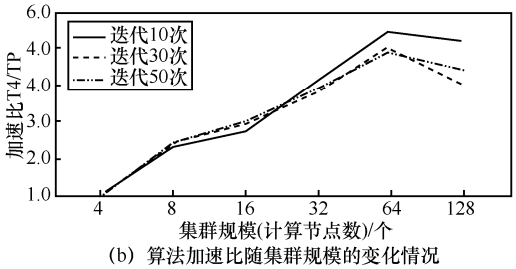


(b) 算法加速比随集群规模的变化情况

图 5　Mahout LDA 算法千万级别下的测试



(a) 算法运行时间随集群规模的变化情况



(b) 算法加速比随集群规模的变化情况

图 6　Mahout LDA 算法亿级别下的测试

2) 数据规模：在相同条件下，数据规模越大，算法的运行时间越长，加速比越大。即随着数据规模的增加，虽然运行时间相应变长，但分布式实现带来的优势也越显著。

3) 迭代次数：与 MrLDA 算法类似，随着迭代次数的增加，运行时间呈现大幅度增长。而在不同数据规模、不同集群规模、不同迭代次数下，Mahout LDA 算法均未达到收敛。

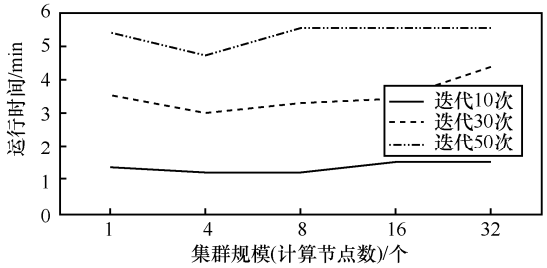
5.3 Spark LDA 算法性能

由于 Spark LDA 算法的普遍执行时间较短，因此对于该算法的实验是从服务器节点数量为 1 时开始测试的。这里分别展示了 Spark LDA 算法在百万、千万、亿级别的数据规模，不同迭代次数和不同集群规模下的运行时间和加速比，如图 7~图 9 所示。

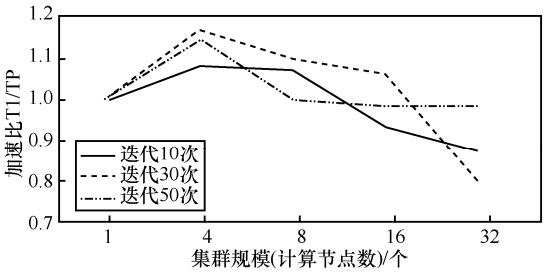
1) 集群规模：Spark LDA 在百万、千万、亿级别数据规模上，均在 4 台服务器节点处运行时间最短，加速比最大。由于在 3 种数据规模级别下，该算法均过早出现了性能的“瓶颈”，因此只测试到 32 台服务器节点规模。

2) 数据规模：随着数据规模的增加，执行时间呈量级增加趋势。千万级数据以下时间可以控制在 30 min 内，亿级别数据的时间消耗明显与百万级别数据和千万级别数据不在一个量级上。这应该是与算法的并行框架——Spark 有关，该框架将中间结果写入内存，随着数据规模的增大，中间结果也大幅度增加，内存无法负荷，所以时间呈现大规模增加趋势。在 3 种数据规模下，Spark LDA 的加速比均在 0.7~1.3 这个较小区间变动，即并不是数据规模越大，分布式实现的优势越明显，而是整体上保持稳定。

3) 迭代次数：与 Mahout LDA 算法类似，随着迭代次数的增加，算法的运行时间呈现大幅度增长趋势。而在不同数据规模、不同集群规模、不同迭代次数下，Spark LDA 算法均未达到收敛，这也论证了该算法在每轮迭代完毕后更新模型，导致收敛速度较慢的这个原理。

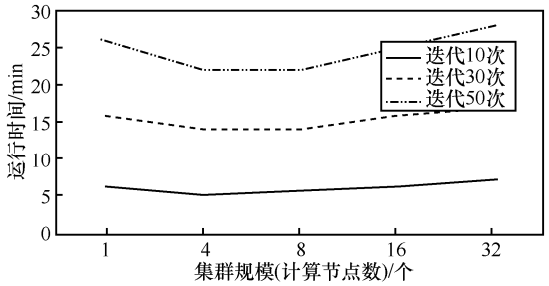


(a) 算法运行时间随集群规模的变化情况

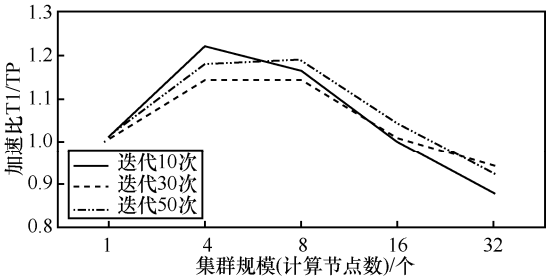


(b) 算法加速比随集群规模的变化情况

图 7　Spark LDA 算法百万级别下的测试

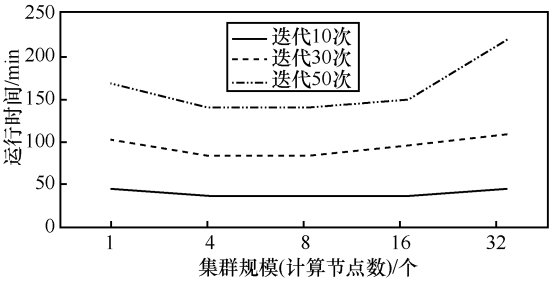


(a) 算法运行时间随集群规模的变化情况

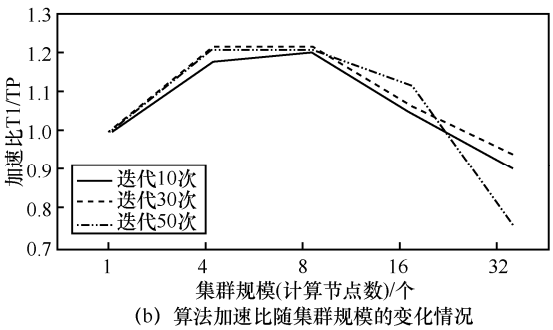


(b) 算法加速比随集群规模的变化情况

图 8　Spark LDA 算法千万级别下的测试



(a) 算法运行时间随集群规模的变化情况



(b) 算法加速比随集群规模的变化情况

图 9　Spark LDA 算法亿级别下的测试

5.4 3 种算法的分析对比

表 1~表 3 分别展示在不同数据规模下，MrLDA、Mahout LDA 和 Spark LDA 算法的运行时间，其中，加粗数值表示在当前条件下，算法可以达到收敛状态。

3 种算法基于的并行框架不同，在一定程度上可以很好地解释实验结果。MapReduce 框架在复杂的挖掘算法中往往需要多个 MapReduce 作业才能完成，多个作业之间存在着冗余的磁盘读写开销和多次资源申请过程，使基于 MapReduce 的算法实现存在严重的性能问题，MrLDA 算法和 Mahout 算法都基于 MapReduce 框架，这就使在相同条件下，2 种算法的运行时间均处于较大数量级，如当数据规模为百万级别，集群规模为 16 个节点，迭代次数为 30 时，MrLDA 算法的运行时间达到 220.90 min，Mahout LDA 算法的运行时间达到 27.20 min，如表 1 所示。

Spark 框架得益于其在迭代计算和内存计算上的优势，可以自动调度复杂的计算任务，避免中间结果的磁盘读写和资源申请过程，在一定程度上提高了算法的性能。Spark LDA 基于 Spark 框架，由于它将中间结果写入内存，因此运行时间处于较小数量级，同样是在数据规模为百万级别，集群规模为 16 个节点，迭代次数为 30 的条件下，Spark LDA 的运行时间仅为 3.40 min，如表 1 所示。但是随着数据规模的增加，内存没有办法负荷，所以它的时间呈大幅度上升趋势，例如，在迭代次数为 50 的条件下，Mahout LDA 算法在亿级别和千万级别下的运行时间比值较小，而 Spark LDA 算法在亿级别和千万级别下的运行时间比值呈现大规模增长趋势，如表 2 和表 3 所示，这充分说明 Spark LDA 算法在巨大规模数据处理上的劣势。

表 1 MrLDA、Mahout LDA 和 Spark LDA 在不同集群规模和迭代次数下的运行时间

集群规模	迭代次数								
	10			30			50		
	MrLDA/min	Mahout LDA/min	SparkLDA/min	MrLDA/min	Mahout LDA/min	SparkLDA/min	MrLDA/min	Mahout LDA/min	SparkLDA/min
4	632.73	16.90	1.30	1 981.47	27.52	3.10	**2 714.32**	44.21	4.80
8	303.60	9.99	1.30	807.37	25.95	3.30	**1 334.95**	41.06	6.50
16	73.33	10.92	1.50	220.90	27.20	3.40	**252.38**	44.25	5.60
32	41.63	11.26	1.60	126.50	28.33	4.50	**145.78**	45.40	5.60

表 2 千万级别下 Mahout LDA 和 Spark LDA 在不同集群规模和迭代次数下的运行时间

集群规模	迭代次数					
	10		30		50	
	Mahout LDA/min	Spark LDA/min	Mahout LDA/min	Spark LDA/min	Mahout LDA/min	Spark LDA/min
4	159.07	5.30	244.13	14.00	684.24	22.00
8	71.55	5.60	187.18	14.00	293.10	22.00
16	62.52	6.50	164.67	16.00	260.21	25.00
32	57.19	7.40	164.09	17.00	271.38	28.00

表 3 亿级别下 Mahout LDA 和 Spark LDA 在不同集群规模和迭代次数下的运行时间

集群规模	迭代次数					
	10		30		50	
	Mahout LDA/min	Spark LDA/min	Mahout LDA/min	Spark LDA/min	Mahout LDA/min	Spark LDA/min
4	338.07	34.00	903.15	84.00	1454.67	138.00
8	145.56	33.00	384.55	84.00	618.98	138.00
16	125.15	38.00	329.17	96.00	493.88	150.00
32	82.71	44.00	233.39	108.00	378.67	216.00

在参数推导方面，MrLDA、Mahout LDA 和 Spark LDA 分别基于变分方法、变分方法和 Gibbs 采样方法。变分方法既能推断隐变量，也能推断未知参数，其难点在于公式演算比较复杂，和 Gibbs 采样方法相比，其具有不易计算但运行效率高的特点。而 Gibbs 采样方法的特点是容易计算但速度慢。MrLDA 基于变分方法，所以其运行效率应该最高，但是由于每一轮迭代后都会更新超参数并计算相应的 Likelihood，在一定程度上影响了算法的执行时间。Spark LDA 基于 Gibbs 采样方法，本来计算时间应该相对较长，但是其主要在内存中完成，节省了大量中间结果的磁盘读写时间，所以时间反而相对较快。

在数据规模方面，随着数据规模的增大，增加计算节点数量，算法的运行时间会相应减少，且数据规模越大，效果越明显。然而不同算法自身原理并不相同，对应数据规模的适用情况也会有所区别：MrLDA 由于自身算法的局限，没有基于内存消耗做优化处理，因此实际能够处理的数据量有限，在处理超过百万级别规模数据时存在问题。Mahout LDA 随着数据规模增大，时间增长幅度较小，比较适合处理大规模数据。例如，在集群规模为 32，迭代次数为 30 的条件下，Mahout LDA 算法在千万级别下的运行时间为 164.09 min，亿级别下的运行时间为 233.39 min，增长幅度较小，如表 2 和表 3 所示。Spark LDA 随着数据规模增大，时间增长幅度较大，在亿级数据规模上相较前 2 种算法仍存在较小的运行时间优势，但是不适合处理超过亿级别的数据，而在百万规模和千万规模上优势明显，更适合处理千万级规模以下数据。例如，在集群规模为 32，迭代次数为 50 的条件下，SparkLDA 在百万数据规模和千万数据规模的运行时间远小于 Mahout LDA，而在亿数据规模下的运行时间优势显著减小，时间量级明显增加，如表 1~表 3 所示。

在迭代次数方面，迭代次数与算法的运行时间呈线性相关。迭代次数太少会导致模型尚未收敛，影响话题的质量，而迭代次数越大，其计算的资源消耗越高，运行时间也会越长。MrLDA 算法在百万数据规模，50 次迭代设置下出现收敛，对应的话题发现质量较好，而其他情况均未出现收敛，在一定程度上影响了算法话题发现的质量，如表 1 所示。

在集群规模方面，在相同迭代次数和数据规模下，随着集群规模的增加，不同算法的运行时间出现先大幅降低后减少速率逐渐变缓，直至基本无变化或呈上升趋势，而加速比呈现先增加后基本无变化或呈下降趋势，且不同算法对应的拐点出现有所不同。这是因为在处理大规模数据时，当前现有的主流算法存在着瓶颈，当集群规模达到一定数量时，再增加计算节点也无法提高算法的性能。这在 MrLDA、Mahout LDA 和 Spark LDA 算法中均有体现，例如，MrLDA 算法的加速比在百万级别的 32 节点出现拐点；Mahout LDA 算法的加速比在百万级别的 8 节点出现拐点，在千万级别和亿级别的 64 节点出现拐点；Spark LDA 算法的加速比均在百万、千万、亿级别的 4 节点出现最优值。

分布式话题模型的训练在最近几年得到越来越多研究者的关注，涌现了一大批相关模型。AliasLDA 针对 Gibbs 采样过程进行优化，基于 Alias Table 使 k 个话题的采样时间复杂度由原来的 $O(k)$ 降低到 $O(1)$。LightLDA 则针对采用的分布进行修改，将原来建立在话题上的词汇、文档联合分布变为 2 个独立的采样过程且交替进行，分别只与词汇和文档相关。WarpLDA 做了更多的工程级别的优化，让 LightLDA 更快。LDA* 则解决了顽健的采样性能以及词分布倾斜这 2 个难题，取得了更好的性能。LDA* 构建于腾讯开源的系统 Angel 之上，得益于 Angel 开放的参数服务器架构、良好的扩展性以及优秀的编程接口设计，它可以轻松处理 TB 级别的数据和十亿维度的话题模型。这些都是近几年比较有代表性的分布式话题模型，也是以后分布式话题模型测试的努力方向。

6　结束语

本文针对社交网络中的话题发现问题，在简单介绍了 MrLDA、Mahout LDA 和 Spark LDA 算法基本原理的基础上，测试了这 3 种典型分布式算法在面向不同数据规模、不同迭代次数、不同集群规模条件下的运行时间和加速比，并给出了不同情况下 3 种算法的数据规模适用性、迭代次数的设置以及集群规模的建议，由此可以为实际应用场景下，不同的现实需求选取较优算法提供参考。下一步，将面向评价指标——困惑度，从话题质量的角度分析各个算法的优劣。

参考文献：

[1]　中国互联网络信息中心. 第 41 次《中国互联网络发展状况统计报

告》[R]. 2018.
China Internet Network Information Center. The 41th statistical report on Internet development in China[R]. 2018.

[2] DHILLON I S, MODHA D S. Concept decompositions for large sparse text data using clustering[C]//Machine Learning. 2001:143-175.

[3] KUMMAMURU K, DHAWALE A, KRISHNAPURAM R. Fuzzy co-clustering of documents and keywords[C]//The IEEE International Conference on Fuzzy Systems. 2003:772-777.

[4] ZHAO Y, KARYPIS G. Soft clustering criterion functions for partitional document clustering:a summary of results[C]//Thirteenth ACM International Conference on Information & Knowledge Management. 2004: 246-247.

[5] MAKKONEN J, AHONENMYKA H, SALMENKIVI M. Topic detection and tracking with spatio-temporal evidence[C]// European Conference on Ir Research. 2003: 251-265.

[6] WU C, WANG B. Extracting topics based on Word2Vec and improved jaccard similarity coefficient[C]//IEEE Second International Conference on Data Science in Cyberspace. 2017: 389-397.

[7] HOFMANN T. Probabilistic latent semantic indexing[C]//International ACM SIGIR Conference on Research and Development in Information Retrieval. 1999: 50-57.

[8] BLEI D M, NG A Y, JORDAN M I. Latent dirichlet allocation[J]. J Machine Learning Research Archive, 2003, 3:993-1022.

[9] STEYVERS M , GRIFFITHS T. Probabilistic topic models[J]. Handbook of Latent Semantic Analysis, 2007, 427(7): 424-440.

[10] BLEI D, CARIN L, DUNSON D. Probabilistic topic models[C]//ACM SIGKDD International Conference Tutorials. 2011: 1.

[11] BERNHARD S, JOHN P, THOMAS H. A collapsed variational bayesian inference algorithm for latent dirichlet allocation[C]// The Twentieth Conference on Neural Information Processing Systems. 2006:1353-1360.

[12] GRIFFITHS T L, STEYVERS M. Finding scientific topics[J]. National Academy of Sciences of the United States of America, 2004: 5228-5235.

[13] RAMAGE D. Characterizing microblogs with topic models[C]// International AAAI Conference on Weblogs and Social Media. 2010:130-137.

[14] CHEN Z, LIU B. Mining topics in documents:standing on the shoulders of big data[C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014: 1116-1125.

[15] LIN T, TIAN W, MEI Q, et al. The dual-sparse topic model:mining focused topics and focused terms in short text[C]//International Conference on World Wide Web. 2014:539-550.

[16] ZHAI K, BOYD G J, ASADI N, et al. MrLDA:a flexible large scale topic modeling package using variational inference in MapReduce[C]// International Conference on World Wide Web. 2012:879-888.

[17] ARONSSON F. Large scale cluster analysis with Hadoop and Mahout[J]. Technology & Engineering. 2015.

[18] MENG X R, BRADLEY J, BURAK Y, et al. MLlib: machine learning in apache spark[J]. Journal of Machine Learning Research, 2015, 17(1):1235-1241.

[作者简介]

邓璐（1989−），女，湖北松滋人，国防科技大学博士生，主要研究方向为社交网络分析、数据挖掘、复杂网络等。

贾焰（1960−），女，四川成都人，国防科技大学教授、博士生导师，主要研究方向为社交网络分析、信息安全等。

方滨兴（1960−），男，江西万年人，中国工程院院士，北京邮电大学教授、博士生导师，主要研究方向为计算机体系结构、计算机网络与信息安全。

周斌（1971−），男，江西南昌人，国防科技大学教授、博士生导师，主要研究方向为社交网络分析、信息安全等。

张涛（1993−），女，湖南常德人，国防科技大学硕士生，主要研究方向为社交网络分析。

刘心（1993−），女，湖南长沙人，国防科技大学硕士生，主要研究方向为社交网络分析。