DOI:10.15923/j.cnki.cn22-1382/t.2017.3.10

一种分布式 LDA 主题模型方法

蒋 权, 董亚则*, 刘 凯, 庞海龙

(长春工业大学 计算机科学与工程学院, 吉林 长春 130012)

摘 要: 基于 Spark 分布式计算框架,采用 Gibbs 抽样方法研究分布式 LDA 主题模型挖掘方法。在 Spark 平台进行大规模数据集处理实验。

关键词: LDA 主题模型; Spark; Gibbs; 分布式计算; 主题建模

中图分类号: TP 301.6 **文献标志码:** A **文章编号:** 1674-1374(2017)03-0265-05

A distributed LDA topic modeling

JIANG Quan, DONG Yaze*, LIU Kai, PANG Hailong

(School of Computer Science & Engineering, Changchun University of Technology, Changchun 130012, China)

Abstract: Based on spark distributed computing framework, we apply Gibbs sampling to study the distributed latent topic information (LDA) mining method. In Spark platform, experiment based on large data sets is carried.

Key words: LDA topic model; Spark; Gibbs sampling; distributed computing; topic modeling.

0 引 言

近年来,大数据技术迅猛发展,各类信息资源的存储量都呈现海量特征[1],其中以文本数据的不断增长最为显著,文本主题分析旨在确定文本的主题以及推断出相应的主题分布,界定主题的外延,追踪主题的转换等,分析结果对文本聚类分析、文本特征生成预测任务、文章自动生成等领域有着非常重要的价值。LDA(Latent Dirichlet Allocation)模型是 Blei[2]等在 2003 年提出的一种三层贝叶斯全概率生成的主题挖掘模型,现在已经广泛应用在信息检索、文本分类、词云推荐等文本相关领域。

Spark[3]是一个使用简单的大数据处理的分布式计算框架,基于它开发的应用程序能够很容易地运行在成百上千台由一堆便宜的商用机器组成的超大集群上,其提出一种新的抽象数据结构 RDD(Resilient Distributed Datasets)[4]以一种可靠容错的方式能够处理 T 级别的数据集,极大地简化了分布式程序设计。

文中提出了一种分布式 LDA 主题模型挖掘方法,并实现了基于 Spark 计算框架的分布式

收稿日期: 2016-11-22

基金项目: 吉林省教育厅"十二五"科学技术研究基金资助项目(2014125, 2014131); 吉林省自然科学基金资助项目(20130101060JC)

作者简介: 蒋 权(1993—),男,汉族,湖北仙桃人,长春工业大学硕士研究生,主要从事数据挖掘、主题模型方向研究,E-mail:13944878813@163.com. *通讯作者:董亚则(1982—),女,汉族,吉林德惠人,长春工业大学讲师,博士,主要从事智能计算方向研究,E-mail:dongyaze@ccut.edu.cn.

LDA 文本主题挖掘模型。通过真实数据的实验验证，该方法在处理大规模数据集时，不仅能获得接近线性的加速比，而且大大提高了挖掘潜在主题信息的准确性，对主题建模效果也有所提高。

1 Spark 数据处理过程

Spark 不仅仅局限于 Map 和 Reduce 编程，它提供了更强大的内存计算模型，这样用户能够快速在内存中对数据集进行多次迭代计算。Spark 计算模型需要处理的数据都会分区存储像 Hadoop 的分布式文件存储系统 HDFS 以键值对的形式存在。

Spark 数据处理流程如图 1 所示。

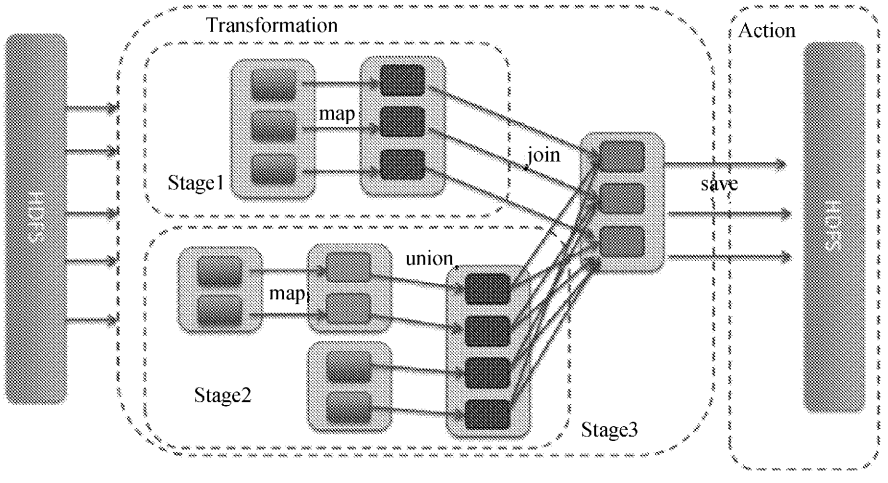

图 1 Spark 数据处理流程

Spark 数据处理流程主要分两个层面：首先，RDD 实现了以操作本地集合的方式操作分布式数据集的抽象实现，并且数据集合都通过缓存到内存中，每次对 RDD 数据集的操作之后的结果都可以存放到内存中，省去了 MapReduce 框架中由于 Shuffle 操作所引发的大量磁盘 IO。这样的处理对于迭代运算比较常见的机器学习算法、交互式数据挖掘来说，效率提升比较大。其次，RDD 上面执行的算子(Operator)主要有 Transformation 和 Action 两大类。在转换方面支持算子有 map、join、groupBy 和 filter 等，而在操作方面支持算子有 reduce、collect、count 等 save。

2 LDA 模型基本思想

LDA 模型是一个包含词、主题和文档三层结构的贝叶斯无监督的概率模型，是典型的文档主题生成模型，是一种对文本数据集潜藏的主题信息进行建模的方法[2,5]，如图 2 所示。

它假设文档属于多个隐含主题上的混合分布，各个主题之间是一个固定词表上的混合分布。令 M 表示文档数目，K 表示主题数目，N_m 表示第 m 个文档的单词数目，文档的生成过程描述如下：

$$\theta_d \sim \mathrm{Dir}(\alpha)$$
$$\varphi_k \sim \mathrm{Dir}(\beta) \tag{1}$$

1)以先验概率 $p(d_i)$选择一篇文档 d_i；

2)从 Dirichlet 分布 α 中取样生成文档 d_i 的主题分布 θ_i；

3)从主题的多项式分布 θ_i 中取样生成文档 d_i 的第 j 个词的主题 $z_{i,j}$；

4)从 Dirichlet 分布 β 中取样生成主题 $z_{i,j}$ 对应的词语分布 $\varphi_{z_{i,j}}$；

5)从词语多项式 $\varphi_{z_{i,j}}$ 中采样最终生成词语 $w_{i,j}$。

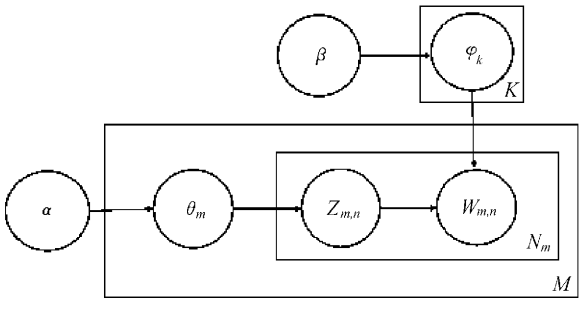

图 2 LDA 的图模型

3　分布式 LDA 算法

提出的基于 Spark 计算框架的分布式 LDA 主题挖掘模型，其算法原理流程如图 3 所示。

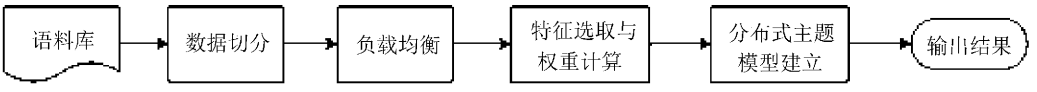

图 3　分布式 LDA 算法原理

首先介绍分布式 LDA 主题建模的思想以及如何使用 Spark 进行主题建模的实现。

3.1　分布式 LDA 主题建模的思想

通过使用 Gibbs sampling 抽取算法[6]，很快能够得到目标的分布情况，但面对巨大的语料集时，数据的维度是难以想象的，并且其中一篇文档中矩阵 θ 将在巨大的数据集中变化的不明显，这样就出现了计算的瓶颈。根据公式：

$$p(z_n^{(m)}=k \mid Z^{-(m,n)},\theta^{-(m,n)},\varphi^{-(m,n)},w^{(m)},\alpha,\beta) \tag{2}$$

可以分析出，独立计算隐藏变量 $z_n^{(m)}$，通过把每一篇文档 $w^{(n)}$ 分发到每一个计算节点，并且分布式的计算 $z_n^{(m)}$，在给定 φ 的前提下，算法的效率将会大大提高。

建立文档的主题模型，其主要的思想为：

1)需要初始化各项参数，比如 γ 和 θ；

2)数据集 X 被按照文档分割成 p 份，例如，有 m 篇文档，分割后每个小数据可以用 X_i 表示，其中 $X_i \in p$；

3)对每个小数据集计算局部估计量和该结点的 log 函数期望；

4)对每个小数据集分别进行一次 Gibbs 采样，这样避免需要不断对文档-主题和主题-词的概率中的状态矩阵进行记录；

5)聚合得到全局的估计量并循环估计参数 γ 和 θ，直到模型的参数收敛；

6)输出训练完成的模型。

但是，不要忽略了一个问题，在 Spark 分布式计算平台都是以集群的方式来处理计算分配任务的，每一个分布式处理的集群都会给 RDD 分配满足全部大数据集的计算机资源，处理的单元根本就不会考虑小批处理文件是不是有效地利用了所分配的资源。Nallapati[7]等虽然在减少网络传输时间开销和网络负载方面做了很多工作，但是他们主要的研究报告结果显示，在 4 台计算机上运行 4 条线程的情况下得到了大约 2.0 的加速比，所以算法在并行效果上也不是很理想。为了解决上面的问题，我们在分布式 LDA 算法设计采用了如下负载均衡策略：

1)将分割后的数据块 X_i 中相同行数据块分配到一个计算节点上，因为数据集中文档的个数通常比词汇表中的词个数要多，仅仅只看是否还有未处理的文档来进行线程调度，这样能保证线程大致可以同时完成，节约计算资源。

2)在计算矩阵 $n_k^{(t)}$ 的传输过程中，必然会产生网络开销，为了尽量减少这部分的网络开销，尽可能地保持 $n_k^{(t)}$ 中数据量相同，也就是说，做到词表中词数尽可能地分配平均。

3.2　特征权重计算

采用 TF-IDF 权重评价方式进行权重计算[8]。权重计算公式如下：

$$TF-IDF=TF_{ij}*IDF_i=\frac{N_w}{N}*\log\frac{D}{DF_i} \tag{3}$$

$$IDF_i=\log\left(\frac{\mid D\mid}{1+\mid\{j\mid w_i\in d_j\}\mid}\right) \tag{4}$$

$$TF_{ij}=\begin{cases}0, & f_{ij}=0\\ \dfrac{f_{ij}}{\max(f_{ij}\mid w_i d_j)}\end{cases} \tag{5}$$

式中：N ——特定文本中单词的数量；

N_w ——词在特定文本中出现的频数；

D ——整个语料集中文本的总数；

DF_i ——整个语料库中出现特定词的文本总数量。

3.3　分布式 LDA 主题模型算法的实现

前面介绍了分布式 LDA 算法避免大量网络传输和计算消耗所作出的改进，并简单地介绍主题特征权重计算的基本方法。但是，前面的改进策略同样会大大增加迭代的次数，所以，我们结合前面介绍的内容给出基于 Spark 的分布式 LDA

主题模型算法建立的过程(见图3),分布式LDA主题建模算法执行过程如下:

Input:ω_m,α,β,K
Output:θ,φ
zero all count variables,$n_m^{(k)}$,$n_k^{(t)}$,n_m,n_k
for all ducuments ω_m(m∈[1,M]) do
for all words n in ω_m do
sample topic index $z_{m,n}=k\sim Mult(1/K)$
increment topic index $n_m^{(k)}+=1$
increment counts and sums:$n_m+=1$
$n_k^{(t)}+=1$
$n_k+=1$
end for
// load balance
set maximum number of executions:$max_{exec}=3$
set number of element in RDD X_i:$num_{X_i}=0$,$sum_{num}=0$
for all data blocks id i i∈[1,P] in X_i do
compute the difference between max and min : $d_i=max_{num}-min_{num}$
add all d_i: $sum_{num}+=d_i$
end for
select optimal solution:min_{sum}
end for
while(parameters converge and not reach maximum number of iterations) do
for all documents ω_m(m∈[1,M]) do
for all words n in ω_m do
$n_m^{(k)}-=1$;$n_m-=1$;$n_k-=1$
k=p(z_i|z→i,w)
$n_m^{(k)}+=1$;$n_m+=1$;$n_k+=1$
end for
end for
end while

4 实验分析

4.1 测试数据以及实验环境

实验使用的数据是由搜狗实验室提供的网络新闻数据集,来自将近20个栏目的全网新闻数据SogouCA,数据大小为630 MB。在使用数据集之前,必须进行数据预处理工作:分词处理采用的分词工具是基于词典的分词算法mmseg;取出停用词(stopwords)和在数据集中出现次数少于5次的词。停用词是指语气助词和代词等常用词,尽管在文本中出现的次数很多,但对于主题的发现帮助不大。

实验环境是由3台虚拟机来搭建集群环境,每台虚拟机的硬件配置为8 cores、8G内存、50G磁盘,操作系统使用的是centos6.5。Spark版本为1.5.2,hadoop版本为2.6.2。

4.2 实验结果分析

实验采用困惑度(Perplexity)指标对实验的结果进行度量。比较LDA模型(LDA)和分布式LDA模型(Spark-LDA)的Perplexity,它是度量概率模型性能的常用指标,同样也是业界主题建模常用的衡量方法[9],Perplexity定义如下:

$$\text{Perplexity}(D_{\text{test}})=\exp\left\{-\frac{\sum_{d=1}^{M=1}\log p(w_d)}{\sum_{d=1}^{M}N_d}\right\} \quad (6)$$

根据上面的计算公式,我们在对数据集不同分块个数 $p=(2,4,6,8)$ 对两模型进行对比实验,如图4所示。

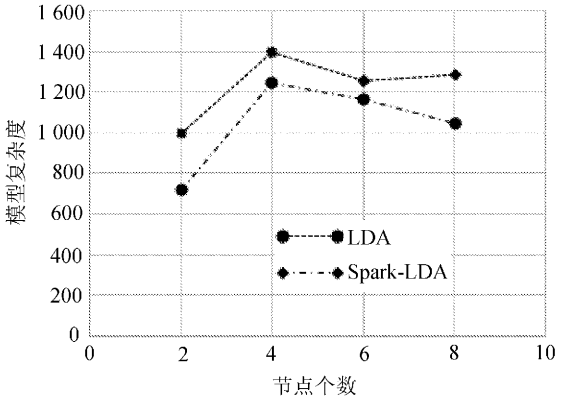

图4 LDA与Spark-LDA模型的困惑度对比

Perplexity表示主题模型对于观测数据的预测能力,取值越小就表示模型的性能越好,相应的模型的推广度越高,结果很明显,Spark-LDA在每分块上的困惑度都要比LDA低,并且从图中可以看出,数据集切分块数越多,模型的效果越好。

下面一组实验是比较LDA和Spark-LDA算法在处理不同大小的数据集上的能力,我们分别用不同大小的文本量(100,200,300,400,500,600)/MB进行对比实验,计算时间与文档数的关系如图5所示。

可以看出,在处理不同的任务量时,LDA和Spark-LDA的处理时间都是随着任务量增加而增加的,但是,Spark-LDA所需要的时间都比LDA要少,并且是呈线性增加的,这说明Spark-LDA在处理大数据量文档集时性能比较稳定。

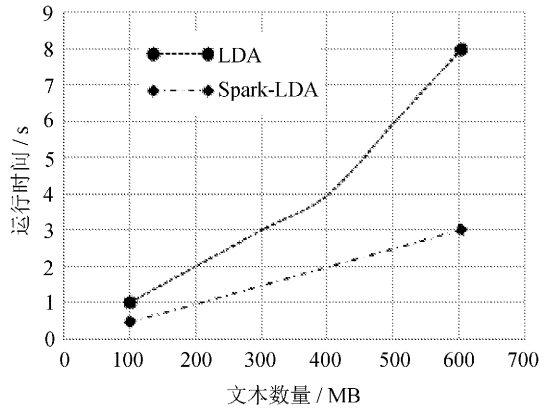

图 5　不同文本规模的计算时间

5　结　语

研究并且实现了 LDA 主题模型建立方法在 Spark 分布式计算框架上的分布式实现，通过数据合理的切分和一种负载均衡的方法，在不降低主题模型的精准度前提下大幅度减少计算时间和网络的消耗。经过真实数据的验证，该方法在处理规模较大数据集时能够得到较低并且接近线性的加速比，由于 Spark 是非常适合这种大量迭代时计算的平台，给模型天然的赋予了比较好的可扩展性，这对于解决海量文本数据中挖掘潜在主题的问题提供了很重要的参考依据。

今后的工作将主要对 LDA 主题模型及其扩展模型与经典的数据挖掘算法结合，进行更深层次挖掘潜在主题信息以及主题演化分布情况。

参考文献：

[1] M Hirzel, H Andrade, B Gedik, et al. IBM streams processing language: analyzing big data in motion [J]. IBM Journal of Research and Development, 2013,57(5):1-7.

[2] D M Blei, A Y Ng, M Jordan. Latent dirichlet dirichlet allocation [J]. The Journal of Machine Learning Research,2003,3:993-1022.

[3] M Zaharia, M Chowdhury, M J Franklin, et al. Spark: cluster computing with working sets [J]. Usenix Conference on Hot Topics in Cloud Computing,2010,15(1):1765-1773.

[4] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, et al. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing [C]//NSDI 2012. Best Paper Award and Honorable Mention for Community Award. 2012.

[5] Y A Ghassabeh, F Rudzicz, H A Moghaddam. Fast incremental LDA feature extraction [J]. Pattern Recognition,2015,48(6):1999-2012.

[6] Liu Shukui, Wu Ziyan, Zhang Yubing. Ldentification of physical parameters and damage localing with markov chain monte carlo method based on Gibbs sampling [J]. Journal of Vibration and Shock,2011,30(10):203-207.

[7] Nallapati, R Cohen, W Lafferty, et al. Parallelized variational EM for latent dirichlet allocation: an experimental evaluation of speed and scalability[C]// Proceeding of Seventh IEEE International Conference on Data Mining Workshop on High Performance Data Mining, Omaha, NE, USA,2007:349-354.

[8] H C Wu, R W P Luk, K F Wong, et al. Interpreting TF-IDF term weights as making relevance decisions[J]. Acm Transactions on Information Systems,2008,26(3):55-59.

[9] T L Griffiths, M Steyvers. Finding scientific topics [J]. Proc of the National Academy of Sciences of United States of America,2004,101:5228-5235.