计算机、材料与Continua CMC,第 62 卷,第 1 期,第 217-231 页,2020 年

基于TF-IDF的新闻文本主题聚类优化方法 Spark 上的算法

周卓1 ,覃娇华1, *, 徐宇翔1 , 谭云1 , 刘强 1和 Neal N. Xiong2

摘要:针对大数据背景下单机架构下文本主题聚类处理速度慢的问题,本文以新闻文本为研究对象,提出基于Spark大数据平台的LDA文本主题聚类算法。由于Spark下的TF-IDF（词频-逆文档频率)算法对于词映射是不可逆的,因此映射后的词索引无法追溯到原始词。本文提出了一种Spark下的TF-IDF优化方法,以保证文本单词能够得到恢复。首先利用本文提出的TF-IDF算法结合CountVectorizer提取文本特征,然后将特征输入到LDA（Latent Dirichlet Allocation)主题模型中进行训练。

最后得到文本主题聚类结果,实验结果表明,对于大数据样本,基于Spark的LDA主题模型聚类处理速度有所提升,同时与基于词频输入的LDA主题模型相比,本文提出的模型困惑度有所降低。

关键词:新闻文本主题聚类,spark平台,countvectorizer算法,TF-IDF算法,潜在狄利克雷分配模型。

1 简介

文本主题挖掘与聚类技术已经存在很长时间了。
近年来,文本主题聚类技术受到越来越多的研究,对经典挖掘算法的改进是当前学术界的研究热点之一。常用的文本处理方法包括检索、情感分析、主题提取等。Dumais 等人[Dumais,Furnas and Landauer（1998）]提出了一种潜在语义分析（LSA)算法进行文本分析,但是该算法无法解决多义性问题,而且在奇异值分解中使用了大量迭代,大规模数据处理成本巨大。Hofmann 提出的 PLSA 算法[Hofmann（1999）]引入了隐式主题变量,一定程度上解决了 LSA 算法的多义性问题。其缺点是 PLSA 的参数个数随着文本量和词数的增加而线性增加,

[1] 中南林业科技大学计算机科学与信息技术学院,长沙 410114

[2] 东北州立大学数学与计算机科学系,OK,74464,美国。

[*] 通讯作者: 秦娇华.邮箱:qinjiaohua@163.com。

中央军事委员会。 doi:10.32604/cmc.2020.06431 www.techscience.com/cmc

实际操作中容易造成过拟合。为了解决这个问题,Blei 等人。 [Blei, Ng and Jordan (2003)]提出了潜在狄利克雷分配（LDA)算法,该算法引入了通过三层贝叶斯模型对文本集、主题层和特征词层的控制。模型的超参数解决了PLSA参数过多的问题。该模型算法在文本主题挖掘和聚类方面取得了巨大成功。近年来,许多研究者在LDA算法的基础上做了大量的研究。姚等人。 [Yao, Song and Peng (2011)]将LDA模型应用到SVM框架中对文本主题进行建模,取得了良好的分类效果。

张等人。 [Zhang,Sun和Ding(2011)]将微博的社交关系与微博文本相结合,提出了一种基于LDA模型的微博生成模型,用于挖掘微博的主题。 Wang等人基于LDA主题模型和Gibbs算法。 [Wang,Gao and Chen (2015)]估计文本的主题概率分布,使用JS（Jensen-Shannon)距离作为文本相似性度量,并使用层次聚类方法进行聚类,以获得更高的聚类纯度和Fscore值。施等人。 [Shi and Cong (2016)]利用最大相关最小冗余（mRMR)方法过滤非活跃词,并结合LDA主题模型提出mRMR_LDA算法,提高了文本聚类的准确性。曲等人。 [Qu, Chen and Cheng (2018)]使用LDA主题模型进行主题挖掘,基于Ward和K-means的聚类算法对科学报告文档进行聚类,取得了良好的性能。

Backenroth 等 [Backenroth, He, Kiryluk et al. (2018)] 将 LDA 主题模型应用于人类遗传学,以细胞类型和组织特异性的方式预测功能性非编码遗传变异,其预测分辨率比前人更高。张等 [Zhang, Lu and Du (2018)] 提出了词合并和 LDA 主题模型（WMF_LDA）,统一了领域词和同义词的映射,并根据词性对文本进行过滤,最终对主题进行建模。该方法减少了建模时间开销,提高了聚类速度。

Spark大数据平台基于分布式内存计算框架,在1.X版本中使用弹性分布式数据集（RDD)作为并行数据结构。尽管有研究人员提出了对缓存机制的优化

spark2.X引入的分布式DataSet RDD,对RDD进行了优化,
提供了强大的分布式计算引擎,逐渐取代了以前的RDD。
Spark的计算框架包括Hadoop的MapReduce框架,并且已经
在Hadoop基础上进行了改进,最大的优点是克服了
Hadoop 只能处理离线数据,磁盘 IO 开销较大。它比
Hadoop在数据处理方面速度提升了100倍[http://spark.apache.org/]。作为
Spark 已被用于解决日益重要的分布式计算系统
复杂的问题。近年来,随着Hadoop等大数据处理平台的出现
和 Spark 的出现,研究人员开始考虑并行化传统
机器学习算法并将其应用于大数据处理平台。对于
例如,张等人。 [张,张,郑等。 （2016)实现了并行化
Hadoop MapReduce 分布式计算框架上的 LDA 主题模型,
解决了单台机器无法解析隐藏主题的问题
大规模语料库中的信息。本文主要研究LDA的应用

Spark大数据处理平台中以新闻文本为研究对象的主题模型首先,我们利用中文分词工具对新闻文本进行分词。其次,提出了一种优化的TF-IDF算法来提取新闻文本特征,并将其输入到基于Spark的LDA模型中。最后,实验结果表明,基于Spark的LDA主题模型的聚类处理速度比单机环境更快。同时,与基于LDA输入词频的LDA主题模型相比,模型复杂度有所降低。

2 相关工作

2.1 新闻文本预处理

新闻通常由文本、图片、视频、链接信息等组成,每篇新闻文本的长度不一。由于新闻文本具有实时性强、数据量大、噪声数据多、主题多等特点,导致新闻文本表现形式各异,核心主题被大量无效信息湮没。因此在新闻文本主题模型提取过程中,需要过滤掉无效信息,提高文本内容质量,以增加核心主题的提取准确率。

新闻文本的预处理主要包括对采集到的新闻文本进行无意义文本的过滤、中文分词、停用词过滤、单个词过滤等,具体步骤如图1所示。

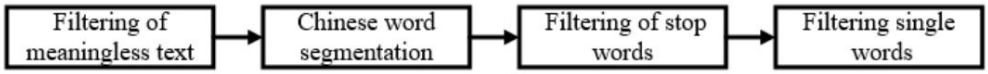

图1:新闻文本预处理步骤

过滤无意义文字:有些新闻文本字数太少,不足以表达其主题,应该被删除。

中文分词:与英文不同,中文不能按照空格来分词,所以需要先按空格分词。

过滤停用词:文本中存在大量没有实际意义、不能凸显特定主题,仅起到连接句子或加强语气作用的词,这些词被称为停用词,如“哦”、“啊”、“我们”等。停用词的过滤不仅可以降低词矩阵的维数,提高关键词提取的准确率,还可以大大减少计算次数和计算规模,提高时间效率。主要包括以下三类:

(1)语气词:如“啊”、“是啊”、“哦”、“而且”、“但是”等。这些词只起到加强语气、增加句子连贯性的作用,并不是文本主题的影响因素。

(2) 出现频率较低且后跟单个词的词:新闻文本的主题会在文本中多次出现,一些出现频率很低的词和单个词在分词后可能会出现。我们认为这个类与主题相关性不够,所以应该对其进行过滤。

(3)特定词汇:与上述两种情况不同,新闻文本中存在一些与主题无关的固定词汇。例如,新闻中会出现“某记者报道”、“某网首发”、“版权”等新闻附加词汇。这些词应该被删除。

2.2 向量空间模型

经过分词之后,一篇新闻文本就变成了若干个单词的集合,但这仍然不能被计算机识别。因此需要通过一些手段将这些非结构化数据转换成计算机可以识别的结构化数据。在文本分析中,向量空间模型是一种广泛使用的处理方法。向量空间模型又称词袋模型,它将文本单词的权重映射到一个高维向量上,这样对文本单词的处理问题就变成了对向量空间的处理。在处理单词权重方面,有词频方法、词频-反文档频率方法等。

定义1:假设文本集合D中有n条文本,即

dd d {.,.D} n ;所有单词经过排重,生成m个词

W,其中W ww w = { , }构成同义词库 m ; 如公式 (1)所示, tf j (i,)为

表示一个函数F ,该函数表示单词wi在文档d j中出现的频率

; wij是单词wi在文档d中的权重 杰。

词频法认为反映主题的词在文本中会重复出现。基于这个思想,一个词在文本中出现的次数就等于对应词汇的权重。

$$j\ F(i,\ ) = (\ tf\ ^{\omega}\text{我}j) \tag{1}$$

由于词频法只考虑词频一个因素,会发现有些词频较高的词并不是文档的主题词,因此研究者将逆文档频率因素加入到词频统计中,称为词频-逆文档频率法。它认为一个词的重要性随着它在文件中出现的次数成正比增加,但同时随着它在语料库中出现的频率成反比下降。也就是说,一个词在文本中出现的次数越多,在所有文本集合中出现的次数越少,那么这个词就越有可能是主题词。TF-IDF是文本处理中最广泛的统计度量之一。例如,刘玉玲等人在加密外包数据的可验证搜索算法中使用TF-IDF作为关键词权重[Liu, Peng and Wang (2018)]。TF-IDF计算如公式(2)所示,其中Num w D (·)表示包含单词wi的文档总数

我∈

$$tf_{jj}{}^{=*+(}\quad (\ ,\ )\ \text{日志}\ \frac{n}{\text{宽度 D}\ {}^{1}_{\text{我})\in}} \tag{2}$$

2.3 LDA模型

LDA [Hofmann (1999)] 模型是继 LSA 模型和 PLSA 模型之后的一项文档模型技术。LDA 模型不仅可以根据主题词汇生成文档,它还是一种无监督的机器学习技术,可以在大型文档集或语料库中识别主题。

定义2:在定义一的基础上,假设新闻文本分布有k
t k 。LDA 模型认为每个主题都分布在主题之间,并且每个主题都是参数为 θ 的多项式分布,如式（1)所示。 （3）。 td |, ~多((3)

θ θ

每个主题由多个关键词混合组成,每个关键词也服从参数为φ的多项式分布,如公式（4)所示。wd | , ~ Multi ((4)

φ φ

参数θ和参数φ分别服从具有超参数α和超参数β的狄利克雷分布,如公式（5)和公式（6)所示。

θ 狄利克雷( ) (5) α

φ ~ 狄利克雷() β (6)

LDA利用联合概率分布计算给定观测变量值下隐含变量的条件分布。核心部分如式(1)所示。 （7）。

p wd | pwt ptd | *(|) (7)

p wd ( )是单词 w 出现在文本 d 下的概率， pwt (| k)表示在主题tk出现的前提下,单词w出现的概率ptd (|) k
表示在文档d出现的前提下,主语tk出现的概率。
图2示出了三个矩阵:文档-词矩阵是每个文本的词特征,文档-词矩阵被拆分为文档-主题矩阵和主题-词矩阵的乘积,即对应于等式的右侧（7）。

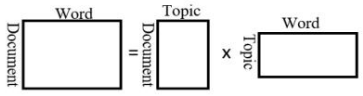

图2： LDA主题模式示意图

2.4 Spark平台

图3是Spark的生态系统结构。 Spark包括核心部分和官方的四个子模块:Spark SQL、图计算模块GraphX、机器学习模型库MLlib、实时流数据处理模型Spark Streaming。该子模块只关注数据的计算,而底层的数据存储是

仍然由 Hadoop 的分布式文件系统 (HDFS) 承载。 Spark MLlib [Meng、Bradley、Yavuz 等人。 (2016)]是Spark为Spark上的分布式机器学习提供的一组接口。它提供了大多数机器学习算法的库,例如分类、回归、协同过滤、聚类和底层优化。

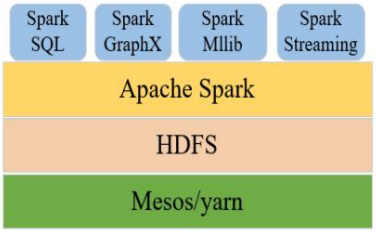

图3： Spark生态系统结构

如图4所示,是Spark的集群结构图。这个集群由四部分组成:客户端、驱动程序、管理节点Master和计算节点Worker。Client主要用于向Driver端提交程序,Driver创建有向非循环调度器（DAGScheduler)和任务调度器（TaskScheduler）。另外Driver端完成RDD生成,将RDD划分为有向无环图,生成任务,并接受管理节点Master的指令将任务发送给计算节点Worker执行。

管理节点Master主要进行资源调度。计算节点Worker负责使用执行器执行任务。计算节点worker通过心跳机制与管理节点保持联系,worker定期向master反馈资源和运行状态,管理节点通知driver何时将任务分发给资源空闲的worker。

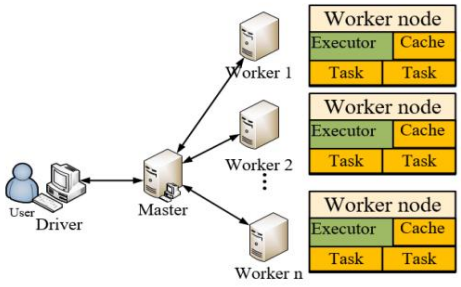

图4： Spark集群结构图

3 基于Spark平台的LDA主题模型聚类

3.1 Spark分布式处理

与 Hadoop 中的 MapReduce 类似,Spark 机器学习系统始终优先拆分数据并将其分发到集群中的计算节点。每个计算节点计算自己的数据,然后将结果聚合回用户,这适用于 Spark 中的所有计算。图 5 是主题聚类的模型

Spark集群环境下的新闻文本模型。用户只需将所有新闻文本集合提供给Spark的Driver端即可。 Driver端根据预设的并行参数将数据分发到各个计算节点,各个节点将

有新闻文本的子集,每个计算节点对新闻文本进行处理和计算,包括文本的分词、词语的索引建立、词语的量化、主题的聚类等,每个计算节点计算出的主题模型的结果都会聚合到驱动端。

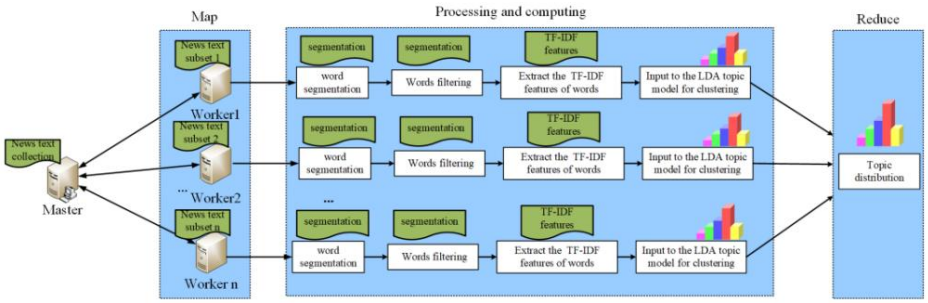

图5： Spark集群环境下新闻文本主题模型聚类模型

3.2 基于TF-IDF优化算法的Spark EM LDA算法

在本节中,我们使用以下数据和函数来表达我们的算法。

数据声明

文本:所有新闻文本;

sentenceData:文本语句的集合 ;

wordsData:所有语句切分之后的单词集合featurizedData:经过TF-IDF训练后的向量化结果; ;

定义函数:

Segment():分割文本到sentenceData

Tokenizer():将所有语句转换成单词

HashingTF:TF-IDF哈希算法,将wordData转换为TF-IDF特征

CVmodel:CountVectorizerModel 的实例

在Spark平台中,Spark上的TF-IDF算法如算法1所示。第一步是将新闻文本分割成语句集合,即sentenceData。

步骤2初始化一个tokenizer,主要作用是识别输入分词后的文本句子,步骤3中， sentenceData会经过Tokenizer转换

将所有语句分词之后得到的单词集合即wordsData。
在第4步中， wordData将通过HashingTF转换为TF-IDF特征:
指定哈希值的数量。第 5 步是保存模型。

算法1:Spark上的TF-IDF算法

输入:所有新闻文本

输出:单词 TF-IDF 特征

1:SentenceData Segment (text)//分段文本到sentenceData

2:初始标记器

3: wordsData tokenizer.input(SentenceData)//句子转换成单词

4: featurizedData HashingTF.input((wordsData, Hashing_num).transform)

//通过输入单词数和哈希值转换为TF-IDF特征

5:保存模型

从上述算法可以知道,TF-IDF算法在单词的向量化表示中使用了hashmap,即每个单词映射到一个哈希索引。

由于Hash的生成不可逆,TF-IDF一旦将词转换成索引,索引就不再与原词一一对应,而且每次文本主题聚类的文本数量和文本长度都不一样,导致第3行无法准确确定需要转换的词的Hash个数。TF-IDF算法虽然适合文本聚类,但依然存在词的追踪和词Hash个数自适应调整的不足。针对此情况,我们在TF-IDF计算过程中加入了CountVectorizer算法,CountVectorizer模型在训练过程中会自动统计文本词数,避免了TF-IDF算法需要手动指定词的Hash个数的问题。另外CountVectorizer可以保存词的索引与词的对应关系,避免了TF-IDF算法中词的索引无法追溯到词的问题。结合CountVectorizer算法的TF-IDF算法如算法2所示。

算法2:TF-IDF算法结合CountVectorizer

输入:所有新闻文本

输出:单词IDF特征,索引-单词映射表

1: SentenceData Segment (text) //将文本分段到sentenceData

2:初始分词器

3: wordsData tokenizer.input（SentenceData）

//句子转换成单词

4:初始 Cvmodel CountVectorizerModel

//初始 CountVectorizerModel

5: wordcount_features Cvmodel.fit(wordsData)

//将单词转化为词频特征6:word=Cvmodel.word//构建索引-单词映射表

7:initial idfModel //初始TF-IDF模型

8: featurizedData idfModel.input （“wordcount_features”）

//计算TF-IDF特征

9:保存模型

算法第1行到第3行与算法1一致。算法第四行初始化CountVectorizerModel,并用它对第二行的单词进行词频统计。这不仅取代了Spark上TF-IDF算法第三行对单词进行哈希处理并进行词频的功能

统计的同时,还添加了索引词映射。第5~8行将CountVectorizerModel生成的词频特征信息转化为TF-IDF特征信息,并构建索引词映射表。

Spark MLlib提供了EM和Online[Hoffman,Bach,and Blei（2010）]两种实现LDA主题模型的方法,它们使用相同的数据输入,但其内部实现原理有所不同,如表1所示。

表 1： Spark LDA 的两种实现

Spark LDA模型	实现方法	参数预测方法	储存方法
EM LDA	基于 Spark GraphX	吉布斯抽样	存储在顶点
在线LDA	采样方法	贝叶斯变分判断	存储在矩阵中

由于Spark GraphX的数据存储是分布式的存储结构,这使得EM LDA可以做大规模的分布式计算。对于LDA来说,单词和文本是两种主要的数据类型,因此Spark GraphX构建了单词节点和文本节点来保存单词和文本。
单词节点存储一个单词以及该单词属于每个主题的概率;文档节点存储文档以及该文档属于每个主题的概率。 Spark GraphX构建文本-单词映射算法如算法3所示。对于每个文本,当一个单词出现在文本中时,GraphX在相应的单词节点和文本节点之间构建一个EdgeRDD,这意味着它将它们连接在一起形成一条边。例如,如表所示。 2、 d1和d 2代表两个新闻文本,

w1, w2、w3 、 w4代表4个单词,表中数据表示该单词在对应新闻文本中的词频。Spark EM LDA根据词与文本的关系构建GraphX图,如图6所示。在构建好图点边数据后,Spark EM LDA根据预设的全局超参数对数据进行迭代计算。

算法3:Spark GraphX构建文本-词映射算法

参数说明:
n :文本总数
m : 分词后的总词数
Matrix:词频矩阵

1: 对于来自[0, 1 n −]的文档i：
2:对于文档i中的单词j , j来自[1,m]:
3: 生成一条边(, [ ][ ]) ij 矩阵 ij作为 EdgeRDD 的元素
4:结束
5:结束

表 2： Spark GraphX 构建文本-单词关系示例

	w_1	w_2	w_3	w_4	.....
d_1	2	0	4	3	…
d_2	0	2	5	3	.....
…	…	.....	…	…	.....

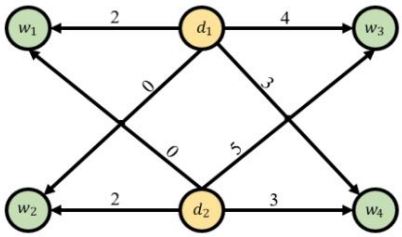

图 6:文本-单词图

4 实验与结果分析

4.1 数据集

搜狗实验室的数据集提供了搜狐新闻数据[https://www.sogou.com/labs/resource/cs.php]。新闻内容来自2012年6月至7月国内、国际、体育、社交、娱乐等18个频道的新闻数据,解压后约2GB。数据集分词工具使用开源Hanlp[http://hanlp.linrunsoft.com/]。 Hanlp是一个由一系列模型和算法组成的Java工具包。提供完整的分词、词法分析、句法分析、语义理解等功能。

其底层算法经过精心优化,快速分词模式下可达每秒2000万词。

4.2 系统环境

实验采用阿里云的ECS弹性云服务器,将Spark单机安装到Cluster中,硬件、软件及系统配置信息如表3所示。

表3:实验硬件及软件配置信息

节点	中央处理器	记忆	操作系统	火花版
母版 (1套)	8 核	32 GB	CentOS7.2	Spark2.3.0
工人（5人一组）	8核	32 GB	CentOS7.2	Spark2.3.0

4.2 实验参数及评价指标

实验中的参数主要包括集群主题数k、Dirichlet参数α和β、最大迭代次数以及Spark集群检查点间隔,如下：

k:聚类中心个数,由于主题聚类属于无监督聚类,因此该参数设置没有绝对的判断标准。有可能

在合理范围内。

α:Spark中的docConcentration,是指分布在主题上的文档的先验参数。在Spark中,该参数必须大于1。当数据量较小时,该参数对聚类影响很大。随着数据规模

增大,参数对聚类效果的影响逐渐减小。

β:Spark中的topicConcentration,主题在词上的先验分布参数。 Spark中该参数必须大于1。其对聚类的影响与Dirichlet参数α相同。

maxIterations:EM算法的最大迭代次数。最大迭代次数取决于数据集的大小。当数据量较大时,建议迭代次数在50~100次之间。

Checkpoint:在Spark中,当迭代次数较多时,Checkpoint间隔可以帮助减少Spark中的shuffle文件大小,有利于故障的恢复。我们使用困惑度测量作为实验指标。困惑度是一种信息熵,是自然语言处理中模型质量的衡量标准。它是概率模型的不确定性,换句话说,困惑度越低,模型越好。如方程式所示。 (8)具体。

$$\text{困惑} = \frac{1}{\text{经验值}} \frac{\sum_{i=1}^{M} \log_{\text{减}}(\quad)}{\sum_{i=1}^{M} \text{否}_i} \tag{8}$$

其中M表示测试新闻文本集中的文本数量， Ni表示第i个文本单词的数量， p w(·)i表示该文本的概率。

4.2 实验过程及结果分析

本文由两组实验组成,包括七个子实验。第一组实验包括实验1和实验2,用于比较本文算法与传统的基于词频的LDA算法的优缺点。第二组实验包括实验2到实验7,比较不同规模数据集下单机架构和基于集群的情况下算法的聚类时间。数据集是从搜狗数据集中过滤出来的。实验设计如表 1 所示。 4.

表 4:实验设计

实验数据集数量		实验平台及算法
1	10000条新闻数据	单机架构下基于词频的LDA算法
2	10000条新闻数据	
3	10000条新闻数据	本文单机架构中的LDA算法
4	50000条新闻数据	本文基于Spark集群的LDA算法
5	50000条新闻数据	本文LDA算法在单机架构上的应用
6	10万条新闻数据	本文基于Spark集群的LDA算法
7	10万条新闻数据	本文LDA算法在单机架构上的应用
		本文基于Spark集群的LDA算法

Hanlp分词算法后的实验集结果如图7所示。每个文本被分成一行并附有文本标签。图8是结合CountVectorizer的TF-IDF算法得到的索引-词映射关系表,其中(a)是单机环境下计算的本地索引-词映射,(b)是分布式索引-在 Spark 集群上计算的单词映射。

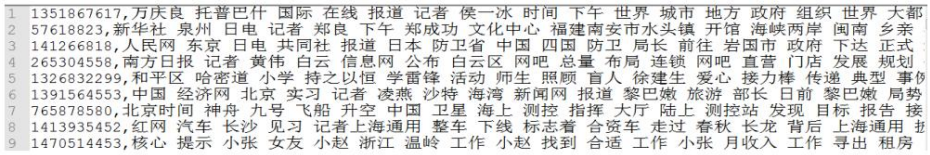

图7新闻文本分词实验结果

(A) (二)

图 8:索引字映射

在分词建词时,文本词会被向量化。LDA通过接收词频特征或TF-IDF特征信息,可以计算出每个主题下的主题词以及每篇新闻文本的主题分布信息。如表5所示,在TF-IDF特征输入、主题个数k=8、α=7、β=2的情况下,每个主题的前4个核心关键词的权重信息。图9为该情况下随机抽取8篇新闻时的主题分布示意图。

表 5:主题-主题词-主题分布

话题	主题词	受试者体重
0	[164、116、6、35]	[0.0045335, 0.0042228, 0.0037616, 0.0037323]
1	[44、72、133、151]	[0.0036255, 0.0035607, 0.0026286, 0.0025112]
2	[8、43、60、171]	[0.0085596,0.0048892,0.0041579,0.0029936]
3	[9, 0, 51, 211]	[0.0127042, 0.0050826, 0.0031506, 0.0029446]
4	[14, 16, 13, 37]	[0.0052911,0.0041124,0.0038823,0.0037967]
5	[318、338、630、796]	[0.0037961, 0.0023438, 0.0022488, 0.0019364]
6	[0, 18, 20, 15]	[0.0086184, 0.0056217, 0.0047544, 0.0046929]
7	[1, 3, 12, 5]	[0.0249111,0.0209015,0.0130717,0.0116296]

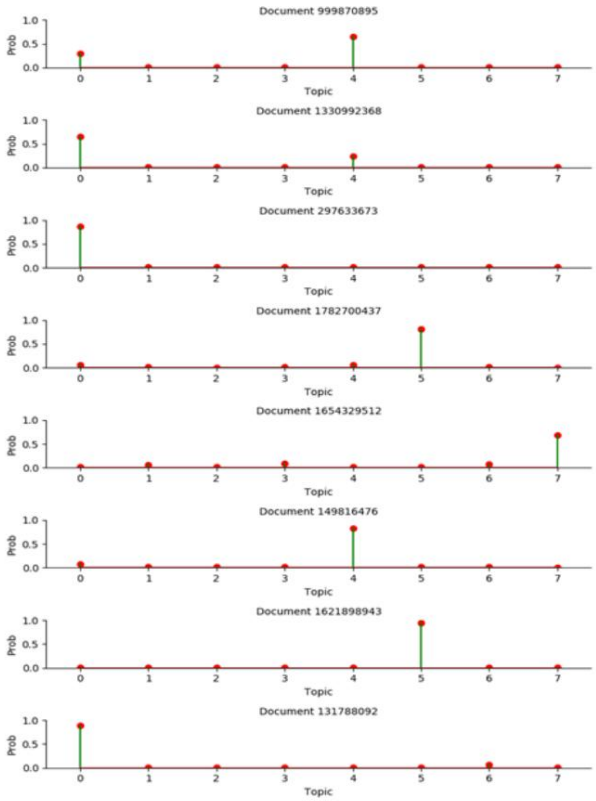

图9:新闻文本主题分布

第一组实验取10 000条新闻文本,主题数为8个,迭代次数为20次,通过对比词频特征输入的LDA主题模型算法与本文算法的结果,发现本文算法在相似度上明显较高,在混淆度上也有所降低,如表6所示。

表 6:困惑度比较

算法	对数（困惑度）
基于词频的LDA	26.811657551355566
本文LDA算法	21.777937111042842

对于第二组实验,当数据量分别为1万、5万、10万,主题聚类个数为8个时,对比了单机环境与Spark集群下的聚类时间,具体实验参数及结果如表7所示。从表中可以看出,当数据量为1万时,集群环境下主题的聚类时间比单机集群长,这是由于集群中节点之间的通信花费了大量的时间。当数据量为5万时,集群环境下主题的聚类时间比单机集群长。

集群环境大约相当于单机主题聚类时间,并且当数据量上升到10万时,集群环境下的主题聚类时间明显低于单机环境。

表7:不同尺度数据下的聚类时间结果

数据大小	数据块	α	β	最大迭代次数	平台	聚类时间（第二）
10000	8	7	2	20	单机	25.61
10000	8	7	2	20	Spark 集群	39.32
50000	32	7	2	三十	单机	92.31
50000	32	7	2	30	Spark集群	94.35
100000	64	7	2	60	单机	436.76
100000	64	7	2	60	Spark 集群	292.65

5 结论

本文提出将TF-IDF特征作为LDA主题模型的输入,并在Spark上将CountVectorizer算法融入TF-IDF算法中,优化了词语转化为向量时不可逆的问题。在

聚类效果方面,与基于词频特征输入的LDA主题聚类相比,本文的LDA主题模型的困惑度明显降低。并且,通过在单机和Spark分布式集群上对不同规模数据集进行测试,得出结论:虽然单机环境下小样本数据的聚类速度快于Spark分布式集群环境,但是随着数据规模的增加,单机处理速度较慢。Spark分布式集群下处理速度有明显提升。但是基于Spark分布式集群的LDA算法也并非完美无缺,由于Spark是基于内存的分布式计算平台,当数据量增加一倍时会消耗大量的内存资源。在后续工作中将进一步研究Spark集群资源的优化。

致谢:湖南省教育厅科研项目（编号:18A174、18C0262）、国家自然科学基金项目（编号:61772561）、湖南省重点研发计划（编号:2018NK2012)湖南省学位与研究生教育改革项目（209号）、中南林业大学研究生教育教学改革项目（2019JG013）。

参考

巴肯罗斯,D.;他,Z。基里卢克,K.;博埃瓦,V.;佩图科娃,L.等人。 (2018) FUN-LDA:用于预测非编码变异的组织特异性功能影响的潜在狄利克雷分配模型。方法和应用。美国人类遗传学杂志,卷。 102,没有。 5,第 920-942 页。

布莱,D.;吴,A.； Jordan, M. (2003):潜在狄利克雷分配。机械学报
学习研究,第 3 卷,第 5 期,第 993-1022 页。

杜迈斯,ST;弗纳斯,G.;兰道尔,T.； Deerwester, S. (1998):使用潜在语义分析来改进信息检索。计算中的人为因素会议,第 281-285 页。

HanLP v1.2.8 链接（2019）:http://hanlp.linrunsoft.com/。

Hofmann, T. (1999):概率潜在语义索引。第十五次人工智能不确定性会议,卷。 51,没有。 2,第 50-57 页。

Hoffman, M.; Bach, F.; Blei, D. (2010):潜在狄利克雷分配的在线学习。
神经信息处理系统的进展,卷。 23,第 856-864 页。

石庆伟;丛绍义(2016): 基于 mRMR 和 LDA 主题模型的文本分类。计算机工程与应用,第 52 卷,第 5 期,第 127-133 页。

刘,YL;彭,H。 Wang, J. (2018):可验证多样性排名搜索
加密的外包数据。计算机、材料与Continua,卷。 55,没有。 1,第 37-57 页。

Liu, XW; Zhu, X.; Li, M.; Wang, L.; Tang, C. 等(2018): 后期融合不完全多视图聚类。IEEE模式分析与机器智能学报,第 41 卷,第 10 期。

Meng, X.;Bradley, J.;Yavuz, B.;Sparks, E.;Venkataraman, S. 等人(2016):MLlib:Apache Spark 中的机器学习。《机器学习研究杂志》,第 17 卷,第 1 期。
34,第 1-7 页。

曲,JY;陈,Z.;郑燕宁(2018):基于主题模式的科技报告文献聚类方法研究。图书馆和信息服务,卷。

4,第 113-120 页。

石庆伟;丛书扬(2016): 基于 mRMR 和 LDA 主题模型的文本分类。计算机工程与应用,第 52 卷,第 5 期,第 127-133 页。

Sougou数据集链接（2019）:https://www.sogou.com/labs/resource/cs.php。

Spark 主页（2019）:http://spark.apache.org/。

王平, 高伟, 陈晓梅(2015): 基于LDA模型的文本聚类研究.情报科学, vol. 1, pp. 63-68.

姚庆忠;宋志玲;彭晨曦（2011）:基于 LDA 模型的文本分类。
计算机工程与应用,第 47 卷,第 13 期,第 150-153 页。

张,CY;孙,JB;丁YQ （2011）:基于MB-LDA模型的微博主题挖掘。计算机研究与发展杂志,卷。 48,没有。第 10 页。
1795-1802。

张L。卢,TL;杜YH (2019) 基于WMF_LDA主题模型的文本相似度计算。 http://www.arocmag.com/article/02-2019-10-028.html。

Zhang, Z.; Zhang, XF; Zheng, N.; Gui, MJ (2016): 基于 Hadoop 平台的 LDA 算法并行化。计算机工程与科学,第 38 卷,第 2 期,第 231-239 页。