一种基于MapReduce的文本聚类方法研究

李 钊[1,2,3,4] 李 晓[3,4] 王春梅[2,3] 李 诚[3,4] 杨 春[3,4]

(北京交通大学软件学院 北京 100044)[1] (山东省计算中心(国家超级计算济南中心) 济南 250014)[2]
(山东省计算机网络重点实验室 济南 250014)[3] (山东省电子政务大数据工程技术研究中心 济南 250014)[4]

摘 要 在文本聚类中,相似性度量是影响聚类效果的重要因素。常用的相似性度量测度,如欧氏距离、相关系数等,只能描述文本间的低阶相关性,而文本间的关系非常复杂,基于低阶相关测度的聚类效果不太理想。一些基于复杂测度的文本聚类方法已被提出,但随着数据规模的扩展,文本聚类的计算量不断增加,传统的聚类方法已不适用于大规模文本聚类。针对上述问题,提出一种基于MapReduce的分布式聚类方法,该方法对传统K-means算法进行了改进,采用了基于信息损失量的相似性度量。为进一步提高聚类的效率,将该方法与基于MapReduce的主成分分析方法相结合,以降低文本特征向量的维数。实例分析表明,提出的大规模文本聚类方法的聚类性能比已有的聚类方法更好。

关键词 文本聚类,MapReduce,K-means,信息损失

中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.1.053

Text Clustering Method Study Based on MapReduce

LI Zhao[1,2,3,4] LI Xiao[3,4] WANG Chun-mei[2,3] LI Cheng[3,4] YANG Chun[3,4]

(School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China)[1]
(Shandong Computer Science Center(National Supercomputing Center in Jinan), Jinan 250014, China)[2]
(Shandong Provincial Key Laboratory of Computer Network, Jinan 250014, China)[3]
(Shandong E-Government Big Data Engineering Technology Research Center, Jinan 250014, China)[4]

Abstract Text clustering is the key technology of text organization, information extraction and topic retrieval. Appropriate similarity measure selection is an important task of clustering, which has great affection on the clustering results. Classical similarity measures, such as distance function and the correlation coefficient, can only describe the linear relationship between documents. However, clustering results based on classical clustering methods are usually unsatisfactory due to the complicated relationship among text documents. Some complicated clustering methods have been studied. But, with the growing scale of text data, the computational cost increases markedly with the increase of dataset size. Classical clustering methods are out of work in dealing with large scale dataset clustering problems. In this paper, a distributed clustering method based on MapReduce was proposed to deal with large scale text clustering. Furthermore, we proposed an improved version of k-means algorithm, which utilizes information loss as the similarity function. For improving clustering speed, parallel PCA method based on MapReduce was used to reduce the document vector dimension. The experimental results demonstrate that the proposed method is more efficient for text clustering than classic clustering methods.

Keywords Text clustering, MapReduce, K-means, Information loss

1 引言

文本聚类是Web数据挖掘的重要研究内容。作为一种无监督的机器学习方法,聚类技术可以将大量文本信息组成少数有意义的簇[1]。目前已提出很多聚类方法,如K-means、层次聚类、Kohonen神经网络、Fisher聚类、模糊聚类方法等,这些聚类方法在很多领域得到重用,如自然语言处理、多文档自动文摘、搜索引擎、用户兴趣模式挖掘、数字图书馆、文件自动归档等[2]。K-means聚类方法由于其简单易用而成为当前应用最广的一种聚类方法,聚类算法的核心是文本间的相似性度量[3],相似度测度直接影响聚类结果。常规的K-means聚类方法通常以欧氏距离、Jffreys距离或相关系数作为距离测度。欧氏距离在一定程度上放大了较大元素误差在距离测度中的作用,Jffreys距离则在一定程度上放大了较小元素误差的作用[4],相关系数主要用来衡量变量之间的线性相关性,因此,常规的K-means聚类方法很难得到最优的聚类结果。

到稿日期:2015-06-01 返修日期:2015-10-24 本文受国家自然科学基金项目(61472230),山东省科技发展计划(2013GZC20102)资助。
李 钊(1975—),男,博士生,主要研究方向为软件工程、大数据挖掘、云计算,E-mail:liz@sdas.org;李 晓 女,硕士,工程师,主要研究方向为大数据挖掘、机器学习;王春梅 女,硕士,副研究员,主要研究方向为云计算、大数据;李 诚 男,硕士,工程师,主要研究方向为大数据挖掘、云计算;杨 春 女,工程师,主要研究方向为大数据、云计算。

Web 内容非常丰富，Web 内容之间的关系也非常复杂，很难通过简单的相关测度进行度量。为提高 Web 文本聚类的效果，一些基于复杂测度的文本聚类方法已被提出[5,6]，并取得了好的聚类结果，但聚类的计算量也随之增加，基于复杂测度的大规模文本聚类方法是目前文本分析亟需解决的难题。

向量空间模型通常是文本的表示形式，一般文本特征向量会有上千维，高维文本特征不但会增加文本聚类的计算量，而且会给聚类带来噪音，影响聚类精度。因此，文本的特征提取在文本聚类分析中非常重要，聚类的特征提取是一种无监督的方法，常用的有文档频率法、词频法、TF-IDF 法、主成分分析法、非负矩阵分解、潜在语义分析等[7,8]。其中，主成分分析是最常用的特征提取方法之一，在实践中得到了广泛应用。对于大规模文本聚类问题，常规的主成分分析方法不再适用，为解决大规模文本特征提取问题，本文将采用基于 MapReudce的并行 PCA 方法予以实现。

常规聚类算法大都基于集中式计算模式[9]，比如构建统一的数据仓库或数据中心，然而在现实中，面对数据信息量不断增大、存储形式向分布式存储转变、数据类型不一致性等情况，集中式聚类算法的不足之处越来越明显。有效的并行算法和实现技术是实现大规模数据挖掘的关键。很多并行挖掘算法以不同技术实现，如多线程、MPI 技术、MapReduce 技术、工作流技术等，不同的实现技术有不同的性能和使用特性，MPI 模式适用于计算密集型问题，特别适用于仿真，但编程复杂度较高，对运行环境的时延要求高，容错性较差。MapReduce是面向信息检索领域而提出的一种适于数据分析的云技术，适合于数据密集型的并行数据挖掘[10-12]。基于 MapReduce 的编程模式不仅可实现分布式聚类，而且具有良好的可扩展性和容错性。分布式聚类算法的核心是在集群的各个节点上进行局部聚类，之后将局部聚类结果进行合并，以获得全局聚类结果。因此本文基于分布式聚类的思想，以集中式的 K-means[13,14]算法为基础，提出一种基于 MapReduce 的改进 K-means 算法，结合基于 MapReduce 的主成分分析特征提取方法，实现大规模文本聚类问题。最后，通过 Web 文本聚类实例来验证算法的有效性。

本文第 2 节简要叙述了 MapReduce 框架的基本知识；第 3 节给出了基于 MapReduce 的改进的 K-means 算法的详细设计；第 4 节构建了基于改进 K-means 的文本聚类的一般框架；第 5 节运用改进的算法进行聚类分析以验证聚类效果；最后总结全文。

2 MapReduce 分布式编程模型

MapReduce 是一种用于处理和生成大规模数据集的分布式编程模型[11]。MapReduce 编程主要分为两个过程，一个是 Map 过程，另一个是 Reduce 过程。通过数据划分的方式将计算任务分解，每个计算节点通过 Map 操作处理该计算节点的数据子集，产生中间结果的键值对集合；然后通过分区操作确保具有相同键的数据映射到同一分区，Shuffle 操作实现中间结果在集群的传递；最后，Reduce 操作对不同分区的中间结果进行处理，从而生成最终结果。典型的 MapReduce 运算流程如图 1 所示。

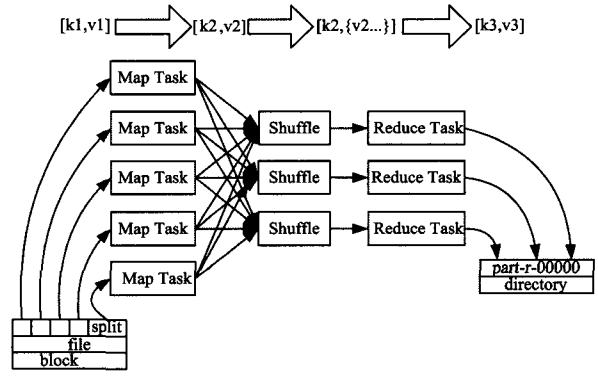


图 1 MapReduce 算法流程

3 基于 MapReduce 的聚类算法

3.1 K-means 算法

K-means 是一种典型的基于划分的方法。以文本聚类为例，当把 n 篇文档聚成 k 类时，首先随机选择 k 篇文档作为初始聚类中心$\{C_1, C_2, \cdots, C_k\}$，计算其余文档与聚类中心之间的距离，将每个文档赋给与之距离最小的簇；然后根据每个簇中的所有对象计算新的聚类中心，以更新后的聚类中心作为输入，将每个文档重新赋给距离其最小的簇，重复这一过程，直到簇的划分不再发生变化。算法的基本流程如图 2 所示。

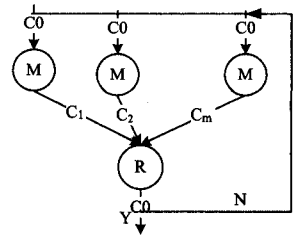


图 2 计算最终聚类中心的迭代过程

1)设定聚类数目为 k，截尾阈值为 ε，并从原数据集合 D 中随机选取 k 个对象作为初始聚类中心。

2)计算数据集合 D 中其余对象与已有的 k 个聚类中心的距离，将每个对象划分到与其距离最小的聚类中。

3)重复步骤 2)，直到 D 中所有对象都被划分到 k 个聚类中。

4)重新计算并更新当前每个聚类的中心。

5)对相邻两次聚类的结果进行比较，如果两次生成的聚类中心差值小于指定的阈值 ε，则聚类结束，输出聚类的结果；否则转步骤 2)。

3.2 基于信息损失的相似性测度

给定一个目标集合，基于信息瓶颈原理[6]的聚类方法是在所有的聚类中寻找使目标类与特征之间的信息损失达到最小的聚类。设在目标空间 X 和特征空间 Y 上的联合概率分布为 $p(x,y)$，信息瓶颈理论是找一个聚类$\hat{X}$在给定聚类质量的约束条件下使信息损失 $I(X;Y)-I(\hat{X};Y)$ 达到最小。$I(X;\hat{X})$是 X 和$\hat{X}$之间的互信息：

$$I(X;\hat{X})=\sum_{x,\hat{x}} p(x)p(\hat{x}|x)\log\frac{p(\hat{x}|x)}{p(\hat{x})} \quad (1)$$

信息瓶颈理论源于 Shannon 的率失真理论[6]，它提供了

在给定失真约束的条件下分类数的下限，给定一个随机变量 X 和失真测度 $d(x_1,x_2)$。Shannon 的率失真理论是指在使平均失真最小的情况下，可以仅用 R 个字节表示变量 X，失真率函数表示为

$$D(R)=\min_{p(\hat{x}|x):I(x;\hat{x})\leqslant R} Ed(x,\hat{x}) \tag{2}$$

其中 $Ed(x,\hat{x})=\sum_{x,\hat{x}} p(x)p(\hat{x}|x)d(x,\hat{x})$。

由聚类 $\hat{X}$ 而导致的变量 X 与 Y 之间的互信息损失可看作平均的失真测度：

$$\begin{aligned} d(x,\hat{x})&=I(X;Y)-I(\hat{X};Y)\\ &=\sum_{x,\hat{x},y} p(x,\hat{x},y)\log\frac{p(y|x)}{p(x)}-\\ &\quad\sum_{x,\hat{x},y} p(x,\hat{x},y)\log\frac{p(y|\hat{x})}{p(y)}\\ &=ED(p(x,\hat{x})\parallel p(y|\hat{x})) \end{aligned} \tag{3}$$

其中，$D(f\parallel g)=E_f\log(f/g)$ 是 K-L 散度。可得到失真函数为

$$D(R)=\min_{p(\hat{x}|x):I(x;\hat{x})\leqslant R} (I(X;Y)-I(\hat{x};Y)) \tag{4}$$

它正是信息瓶颈理论提出的最小化标准，即找到一个聚类使目标空间与特征空间的互信息损失最小。

设 c_1 和 c_2 是聚成的两个类，由于两个类聚到一起而导致的信息损失可表示为

$$d(c_1,c_2)=I(c_1;Y)-I(c_1,c_2;Y) \tag{5}$$

通过标准的信息论运算可得：

$$\begin{aligned} d(c_1,c_2)&=\sum_{y,i=1,2} p(c_i,y)\log\frac{p(c_i,y)}{p(y)p(c_i)}-\sum_y p(c_1\cup c_2,y)\\ &\quad\log\frac{p(c_1\cup c_2,y)}{p(y|c_1\cup c_2)} \end{aligned} \tag{6}$$

其中 $p(c_i)=\frac{c_i}{|X|}$，$|c_i|$ 表示类 c_i 的势，$|X|$ 表示目标空间的势，$p(c_1\cup c_2)=|c_1\cup c_2|/|X|$。

在基于信息熵的概率合并中，假定两类是相互独立的，因此，合并后的概率形式为两类概率分布的和：

$$p(y|c_1\cup c_2)=\frac{1}{|c_1\cup c_2|}=\sum_{i=1,2}\frac{|c_i|}{|c_1\cup c_2|}p(y|c_i) \tag{7}$$

用信息损失量 $d(c_1,c_2)$ 代替 K-means 聚类中的欧氏距离，以度量两个文档之间的相关性。

3.3 基于主成分分析的特征选择算法

主成分分析(Principal Component Tnalysis，PCA)方法是在数据空间中找一组向量，使这一组向量尽可能地解释原始数据的方差，从而将原始数据从 n 维空间降到 m 维($m<n$)，降维后的数据保存了原始数据的主要信息。

分布式 PCA 算法的基本流程如图 3 所示。

分布式 PCA 算法的基本原理如下：对于输入数据，用 P_i ($1\leqslant i\leqslant n$)表示输入数据集，算法最终映射的维度为 t。

①局部 PCA

第一步：对于集群的每个节点，针对各自分配的数据 P_i，采用奇异值分解算法(SVD)$P_i=U_iD_i(E_i)^{\mathrm{T}}$，计算 $D_i^{(t)}$；

$D_i^{(t)}$：仅包含矩阵 D_i 的第 t 个对角元素，其余元素为 0 的矩阵。

第二步：根据 $P_i^{(t)}=U_iD_i^{(t)}(E_i)^t$，计算 $E_i^{(t)}$；

$E_i^{(t)}$：矩阵 E_i 的第 t 列组成的矩阵。

第三步：根据 $P_i^{(t)}$ 计算 $S_i^{(t)}$，$S_i^{(t)}=(P_i^{(t)})^TP_i^{(t)}$。

②全局 PCA

第四步：综合各个节点的 $S_i^{(t)}$，计算 $S^{(t)}$，$S^{(t)}=\sum_{i=1}^{n}S_i^{(t)}$。

第五步：计算协方差矩阵 $S^{(t)}$ 的特征值和特征向量，$S^{(t)}=E\Lambda E^{\mathrm{T}}$。

第六步：通过 $E^{(t)}$ 以及第二步计算得到的 $P_i^{(t)}$，分布式计算 $\hat{P}_i$，$\hat{P}_i=P_i^{(t)}E^{(t)}(E^{(t)})^{\mathrm{T}}$。

第七步：综合第六步的结果，输出 $\hat{P}^T=[\hat{P}_1^T,\cdots,\hat{P}_n^T]$，根据 PCA 算法的结果，选择第一主成分 F_1 作为特征提取的依据，$F_1=a_{11}x_1+a_{21}x_2+\cdots+a_{1m}x_m$。

根据第一主成分系数 a_{ij} 的绝对值的大小判断特征词的重要程度，按照重要度对特征词进行排序，依照重要度递减的规则选取一定数量的特征词，实现特征选择。

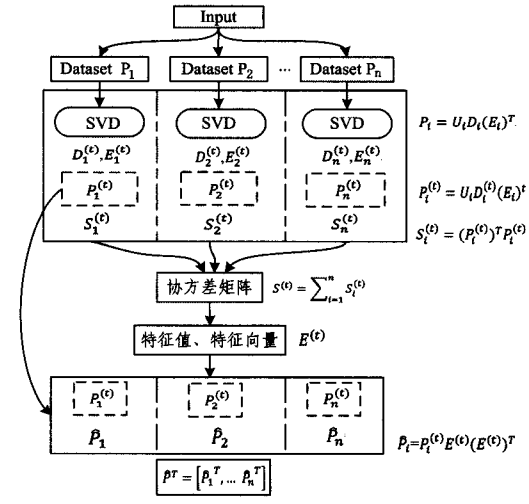


图 3 分布式 PCA 基本流程

3.4 基于 MapReduce 的改进 K-means 算法设计

MapReduce 最主要的工作是设计和实现 Map 和 Reduce 函数，包括输入和输出〈*key*，*value*〉键值对的类型以及 Map 和 Reduce 函数的具体逻辑。K-means 聚类的核心过程是计算每个样本与聚类中心的距离与分配不同样本到距离其最近的聚类中心，其操作之间是相互独立的，因此这一步骤可以并行地执行，并行 K-means 算法在每次迭代中分别执行相同的 Map 和 Reduce 操作。

以文本聚类为例，根据空间向量模型的定义将文档表示成词的向量。首先随机选择 k 个向量作为初始聚类中心，并将这 k 个中心点存储在 HDFS 上的一个文件中作为全局变量，每次迭代由两部分组成：Map 函数和 Reduce 函数。

(1)Map 函数

Map 函数输入的〈*key*，*value*〉选择 MapReduce 框架默认的输入格式，即 *key* 是当前样本相对于输入数据文件起始点的偏移量，*value* 是文档的向量表示。首先，采用基于信息损失的相似性度量式(6)计算当前的文档向量与 k 个初始中心的相似性，找出与其相似性最大的聚类中心的标号；然后输出〈*key*，*value*〉，其中 key_1 是所属聚类中心的标号，$value_1$ 是当前文档的向量表示。

(2)Reduce 函数

Reduce 函数的输入是 Map 函数的输出，因此 Reduce 接收到的键值对为$\langle key_1, list\{value_1, value_1 \cdots\}\rangle$，$key_1$ 是所属聚类中心的标号，$value_1$ 是由各个 Map 函数传输的同属于一个聚类中心的文档向量组成的 *List*。Reduce 主要根据式(7)完成聚类中心的更新，具体执行过程为：

首先对 *List* 中的文档向量各维度的值进行累加，然后将累加值除以总的 *List* 中样本个数，即得新的聚类中心点坐标。

根据 Reduce 的输出结果，得到新的中心点坐标，并更新到 HDFS 上的文件中，然后进行下一次迭代，直到算法收敛。

4 基于改进 K-means 的文本聚类

根据文本聚类的基本流程构建 K-means 聚类的框架如下。

(1)数据处理

要对文档进行聚类，必须使用向量空间模型将文档表示成向量的形式，向量空间模型的形式为

$$v(d_i)=(T_{i1}:w_{i1};T_{i2}:w_{i2};\cdots;T_{in}:w_{in}) \tag{8}$$

其中，T_{ij} 是词条，w_{ij} 是对应词条的权重。将文档进行中文分词之后计算词的 *TF-IDF* 值，用 *TF-IDF* 值代替词条的权重组成文档向量。为实现信息损失量的计算，需要对各文档对应的 *TF-IDF* 值进行归一化处理，使得向量满足概率表达形式，即

$$w_{ij}=\frac{w_{ij}}{w_{i1}+w_{i2}+\cdots+w_{in}} \tag{9}$$

(2)特征提取

提取出文档中最能表示文档内容的词汇即关键词，可以采用两种方法：1)可以设置 *TF-IDF* 的阈值，仅输出大于阈值的词，这样可以略去意义较小甚至无意义的词，此方法不可预知关键词的个数；2)可以调用自然语言处理工具 FudanNLP 的接口进行关键词提取，此方法可以设置提取关键词的个数，FudanNLP 采用一种类似于 PageRank 的算法来提取关键词。本文采用分布式 PCA 算法进行特征选择。

(3)相似性度量

在文本聚类中，对于不同的数据集，选用不同的相似性度量，聚类质量会受到很大影响，因此要尝试不同的相似性度量方法，选择聚类效果较好的一种。本文采用基于信息损失的相似性度量。

(4)选择模型

K-means 聚类需要预先确定聚类的数目，初始值的选择对聚类质量也存在影响，因此可以采用合适的算法生成初始类，可以借助 Canopy 算法设置合适的阈值来生成初始聚类中心。对于文本聚类而言，可以采用基于 LDA(Latent Dirichlet Allocation)的主题发现算法，以主题的个数确定聚类的数目，同时主题可以作为类别标签。

(5)模型评估

设定测度函数，判断算法迭代终止条件，采用均方差作为测度函数，均方差定义如下：

$$E=\sum_{i=1}^{k}\sum_{p\in C_i}\|p-m_i\|^2 \tag{10}$$

其中，p 表示数据集中的对象即一个文本向量，m_i 为簇 C_i 的均值，k 为聚类数目。当测度函数的值开始收敛到某一固定值时，K-means 算法终止。

5 实验结果及分析

5.1 实验环境及实验数据

(1)集群环境部署

实验搭建的 Hadoop 集群共有 4 台机器，其中 1 台为 master，其余 3 台为 slave，每台机器 CPU 为 Intel(R) core (TM)双核，内存 8GB，硬盘 500GB，每台机器的操作系统是 Ubuntu11.04，JDK 版本是 1.6。

(2)实验数据

本文实验数据集有两个，第一个数据集来源于亿云垂直搜索引擎项目中采用 Nutch 分别从新浪、搜狐、腾讯等网站上爬取的保存在分布式数据库 Hbase 中的网页数据。通过搜索引擎的自动分类模块，先将爬取的网页信息进行分类，从中抽取经济、体育、医疗、教育、交通 5 类网页，每类网页对应的数目如表 1 所列。

表 1 网页类别及样本量

类别	数量
经济	6738
体育	8256
医疗	9384
教育	5065
交通	18095

第二个数据集来源于搜狐的新闻语料。选取了财经、IT、健康、体育、旅游 5 类数据，每个类别包含 1990 篇文档，共计 9950 篇。

5.2 聚类分析

为了说明 K-means 算法中采用信息损失的相似性度量比采用欧氏距离更具优势，将两种相似性度量方法的聚类结果进行对比，分别在 Hadoop 集群上测试改进的 K-means 与采用欧氏距离为测度的 K-means 算法的聚类效果，分别采用两个数据集进行聚类分析。首先给出初始聚类个数 $K=5$，针对第一个数据集从 Hbase 数据库读取数据进行词频统计，分词工具采用中科院的 ICTCLAS 分词器，对词性进行标注，保留名词、动词、副词，去除连词、介词等意义较小的词；随后计算每个词的 *TF-IDF* 值，设置 *TF-IDF* 阈值为 0.048，低于此阈值的词条将被忽略，使用向量空间模型将文档表示成向量的形式，向量每个维度的值是对应词的 *TF-IDF* 值，随机选定 5 个聚类中心，通过信息损失相似性度量计算剩余文档与选定初始聚类中心向量的相似度值，将每篇文档划分到与其相似性最大的类中，经过迭代输出最终的聚类结果。

为了测试 PCA 特征选择对聚类结果的影响，对比采用 PCA 进行特征选择的 K-means 算法的聚类效果与仅采用停用词、*TF-IDF* 阈值等初步降维处理的 K-means 聚类结果。对第二个数据集搜狐新闻语料进行分词、去除停用词、设置 *TF-IDF* 阈值，初步对特征词进行降维后，采用基于欧氏距离测度的 K-means 算法进行聚类。

针对初步降维后的文档 *TF-IDF* 值组成的 $N\times M$ 矩阵(其中 N 表示样本文档的个数，M 表示特征词的个数即向量的维度)进行 PCA 分析，提取第一主成分对应的载荷系数，按照各特征词对应的载荷系数的绝对值的大小对特征词进行排

序，选取前 Z 个特征词（$Z<M$，Z 值可以设定）实现对向量的降维操作，对降维后的数据进行 K-means 聚类。

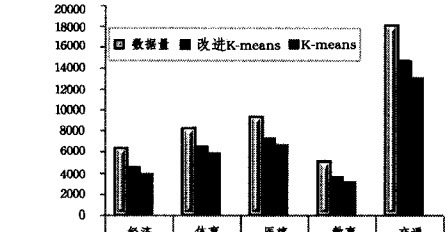
5.3 结果分析（一）

在对数据集一进行聚类实验的过程中，先使用单个节点对数据集进行聚类，然后分别使用 2～4 个节点测试分布式文本聚类的效率，结果如表 2 所列。

表 2 聚类时间变化

节点数	聚类时间(min)
1	45.03
2	20.48
3	14.04
4	11.29

从表 2 可以看出，对比单节点，基于 MapReduce 的分布式文本聚类在节点增多时计算速度明显加快，计算时间按线性递减，与单机内存计算相比，分布式计算的聚类结果没有明显变化。分布式文本聚类能明显提高聚类的效率，为了验证改进的聚类算法相比传统 K-means 算法在并行框架下的优越性，在相同的 Hadoop 集群、相同文本数量和相同文本向量的维度下，比较两种算法的各个类别的正确聚类情况以及整个聚类过程的执行时间、迭代次数。两种算法在同一 Hadoop 集群中的聚类结果对比如图 4 和表 3 所示。



	经济	体育	医疗	教育	交通
数据量	6378	8256	9384	5065	18095
改进K-means	4565	6494	7289	3576	14678
K-means	3945	5845	6668	3069	13056
提高百分比	10%	8%	7%	10%	9%

图 4 测试数据集一聚类结果对比

表 3 测试集一聚类执行时间及迭代次数

算法	迭代次数	执行时间(min)
改进 K-means	39	37.788
K-means	43	25.592

从图 4 的聚类模拟结果可以看出，改进 K-means 算法在正确聚类的网页数目上优于基于欧氏距离的分布式 K-means 算法，正确归类数目提高了 7%～10%。

从表 3 可以看出，在终止条件相同的情况下，改进的 K-means 算法由于计算复杂度高，花费的时间要比传递的 K-means 算法多。

5.4 结果分析（二）

为了验证采用 PCA 进行特征选择可以有效地减少 K-means 的迭代次数，加快收敛速度，针对搜狐新闻测试数据，采用初步降维处理，*TF-IDF* 阈值设为 0.055，此时根据得到的文本向量进行聚类，继而采用 PCA 算法对初步降维后文本向量进行二次降维处理。设定提取特征词的个数，在相同的 Hadoop 集群、相同文本数量下，比较两种降维处理的各个类别的正确聚类情况以及整个聚类过程的执行时间、迭代次数。两种算法在同一 Hadoop 集群中的聚类结果对比如图 5 和表 4 所示。由表 4 的结果可以看出，经过 PCA 降维处理过的数据会更快地趋向收敛，迭代次数明显减少。

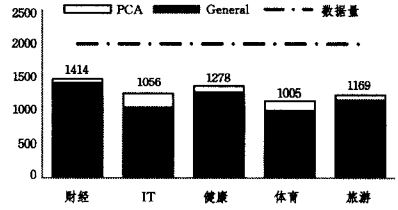


图 5 测试数据集二聚类结果对比

表 4 测试集二聚类执行时间及迭代次数

算法	迭代次数	执行时间(min)
PCA	10	5.19
General	15	7.92

图 5 显示了在数据集二上进行聚类的效果，图中上方的虚线表示数据集包含的 5 类样本的数量，都是 1990 篇。从图中可以看出选择合适的特征选择算法对聚类结果的准确率存在一定影响。

结束语 本文提出了改进的 K-means 算法，旨在提高分布式聚类的效率与查准率。首先利用数据预处理去掉对网页内容影响较小的词条，其次通过 PCA 特征选择在尽量保证网页信息的情况下减少最终参与聚类的文档向量的维度。实验结果表明，该算法能够在保证聚类质量的基础上提高算法效率。

参 考 文 献

[1] Zhang Ren-yuan, Shibata T. An analog on-line-learning K-means processor employing fully parallel self-converging circuitry[J]. Analog Integrated Circuits and Signal Processing, 2013, 75(2): 267-277

[2] Sathiyakumari K, Preamsudha V, Manimekalai G, et al. A Survey on Various Approaches in Document Clustering [J]. International Journal of Computer Technology and Applications, 2011, 2(5): 1534-1539

[3] Xiang Xiao-jun, Gao Yang, Shang Lin, et al. Parallel Text Categorization of Massive Text Based on Hadoop [J]. Computer Science, 2011, 38(10): 184-187(in Chinese)
向小军，高阳，商琳，等. 基于 Hadoop 平台的海量文本分类的并行化[J]. 计算机科学，2011，38(10)：184-187

[4] Kannungo T, Mount D M, Netanyahu N S, et al. An Efficient K-Means Clustering Algorithm: Analysis And Implementation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7): 881-891

[5] Wang Da, Mazumdar A, Womell G W. A Rate-Distortion Theory For Permutation Spaces[C]// IEEE International Symposium on Information Theory Proceedings. 2013: 2562-2566

[6] Sun Zhan-quan, Geoffrey F, Gu Wei-dong, et al. A parallel clustering method combined information bottleneck theory and centroid-based clustering [J]. The Journal of Supercomputing, 2014, 69(1): 452-467

[7] Lu Shi-jian, Chen Tao, Tian Shang-xuan, et al. Scene text extraction based on edges and support vector regression[J]. International Journal on Document Analysis and Recognition, 2015, 18(2): 125-135

（下转第 269 页）

CVRP 的标准案例，测试结果表明改进后的 ILS 能够在增加少量计算时间的基础上大幅度提高求解的质量。将改进后的 ILS 算法与量子进化算法和蜂群算法进行了比较，进一步验证了改进后 ILS 算法的有效性。

参 考 文 献

[1] Toth P, Vigo D. The vehicle routing problem [M]// The vehicle routing problem: Society for Industrial and Applied Mathematic Philadelphia. 2002: 245-247

[2] Gendreau M, Potvin J Y. The Vehicle Routing Problem: Latest Advances and New Challenges [M]. New York: Springer, 2008: 143-169

[3] Laporte G. Fifty years of vehicle routing [J]. Transportation Science, 2009, 43(4): 408-416

[4] Vidal T, Crainic T G, Gendreau M, et al. Heuristics for multi-attribute vehicle routing problems: A survey [J]. European Journal of Operational Research, 2013, 231(1): 1-21

[5] Gendreau M, Potvin J Y. Handbook of Metaheuristics (2nd Edition) [M]. New York: Springer, 2010: 368-370

[6] Chen Ping, Huang Hou-kuan, Dong Xing-ye. A Multi-Operator Based Iterated Local Search Algorithm for the Capacitated Vehicle Routing Problem [J]. Journal of Beijing Jiaotong University (Natural Science), 2009, 33(2): 1-5 (in Chinese)
陈萍，黄厚宽，董兴业. 基于多邻域的车辆路径优化迭代局部搜索算法[J]. 北京交通大学学报(自然科学版)，2009，33(2)：1-5

[7] Penna P H V, Subramanian A, Ochi L S. An Iterated Local Search heuristic for the Heterogeneous Fleet Vehicle Routing Problem [J]. Journal of Heuristics, 2013, 19(2): 201-232

[8] Subramanian A, Penna P H V, Uchoa E, et al. A hybrid algorithm for the Heterogeneous Fleet Vehicle Routing Problem [J]. European Journal of Operational Research, 2012(221): 285-295

[9] Subramanian A, Drummond L M A, Bentes C, et al. A parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery [J]. Computers & Operations Research, 2010, 37(11): 1899-1911

[10] Michallet J, Prins C, Amodeo L, et al. Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services [J]. Computers & Operations Research, 2014, 41(1): 196-207

[11] Schrimpf G, Scheider J, Stamm-Wilbrandt H, et al. Record breaking optimization results using the ruin and recreate principle [J]. Journal of Computational Physics, 2000, 159(2): 139-171

[12] Shaw P. Using constraint programming and local search methods to solve vehicle routing problems [C] // Principles and Practice of Constraint Programming — CP98. Springer, 1998: 417-431

[13] Pisinger D, Ropke S. A general heuristic for vehicle routing problems [J]. Computers & Operations Research, 2007, 34(8): 2403-2435

[14] Clarke G, Wright J. Scheduling of vehicles from a central depot to a number of delivery points [J]. Operations Research, 1964, 12(4): 568-581

[15] Groër C, Golden B, Wasil E. A library of local search heuristics for the vehicle routing problem [J]. Mathematical Programming Computation, 2010, 2(2): 79-101

[16] Hou Yan-e, Dang Lan-xue, Kong Yun-feng, et al. Design and Application of Metaheuristic Framework for School Bus Routing Problem [J]. Journal of Chinese Computer Systems, 2014, 35(7): 1625-1631 (in Chinese)
侯彦娥，党兰学，孔云峰，等. 校车路径问题元启发算法框架设计及应用[J]. 小型微型计算机系统，2014，35(7)：1625-1631

[17] Zhao Yan-wei, Peng Dian-jun, Zhang Jing-ling, et al. Quantum volutionary algorithm for capacitated vehicle routing problem [J]. System Engineering Theory and Practice, 2009, 29(2): 159-166 (in Chinese)
赵燕伟，彭典军，张景玲，等. 有能力约束车辆路径问题的量子进化算法[J]. 系统工程理论与实践，2009，29(2)：159-166

[18] Wang Zhi-gang, Xia Hui-ming. An artificial bee colony algorithm for the vehicle routing problem [J]. Computer Engineering and Science, 2014, 36(6): 1088-1095 (in Chinese)
王志刚，夏慧明. 求解车辆路径问题的人工蜂群算法[J]. 计算机工程与科学，2014，36(6)：1088-1095

(上接第 250 页)

[8] Bellot P, Bonnefoy L, Bouvier V, et al. Large Scale Text Mining Approaches for Information Retrieval and Extraction [M] // Innovations in Intelligent Machines. 2014: 3-45

[9] Zhu Ye-xing, Li Yan-ling, Cui Meng-tian. Clustering Algorithm CARDBK Improved from K-means Algorithm [J]. Computer Science, 2015, 42(3): 201-205 (in Chinese)
朱烨行，李艳玲，崔梦天. 一种改进 K-means 算法的聚类算法 CARDBK[J]. 计算机科学，2015，42(3)：201-205

[10] Brecheisen S, Krieegel H P, Kroger P, et al. Visually mining through cluster hierarchies [C] // International Conference on Data Mining. Lake Buena Vista, FL, 2004: 400-412

[11] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters [C] // Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation, 2004(6): 137-150

[12] Lee K, Lee Y, Choi H. Parallel Data Processing with Map-Reduce: A Survey [J]. ACM SIGMOD Record, 2011, 40(4): 11-20

[13] Kanungo T, Mount M D, Neanyahu N S, et al. A Local Search Approximation Algorithm for k-Means Clustering [J]. Computational Geometry Theory & Applications, 2004, 28(2): 89-112

[14] Xiong Zhong-yang, Chen Ruo-tian, Zhang Yu-fang. Effective method for cluster centers' initialization in K-means clustering [J]. Application Research of Computers, 2011(11): 4188-4189 (in Chinese)
熊忠阳，陈若田，张玉芳. 一种有效的 k-Means 聚类中心初始化方法[J]. 计算机应用研究，2011(11)：4188-4189

[15] Younis O, Fahmy S. HEED: A Hybrid, Energy-efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks [J]. IEEE Transactions on Mobile Computing, 2004, 3(4): 366-379