基于 MapReduce 编程模型的 TFIDF 算法研究*

赵伟燕[1]，王静宇[2]
(1.内蒙古科技大学 信息工程学院，内蒙古 包头 014010；
2.内蒙古科技大学 信息办与网络中心，内蒙古 包头 014010)

摘 要：随着 Internet 等技术的飞速发展，信息处理已经成为人们获取有用信息不可或缺的工具，如何在海量信息中高效地获得有用信息至关重要，因此自动文本分类技术尤为重要。现有的文本分类算法在时间复杂性和空间复杂性上遇到瓶颈，不能满足人们的需求，为此提出了基于 Hadoop 分布式平台的 TFIDF 算法，给出了算法实现的具体流程，通过 MapReduce 编程实现了该算法，并在单机和集群模式下进行了对比实验，同时与传统串行算法进行了对比。实验证明，使用 TFIDF 文本分类算法可实现对海量数据的高速有效分类。

关键词：文本分类；MapReduce；并行化；TFIDF 算法
中图分类号：TP391.1 **文献标识码**：A **文章编号**：1674-7720(2013)04-0071-03

Research of TFIDF algorithm based on the MapReduce programming model

Zhao Weiyan[1], Wang Jingyu[2]
(1.Information Engineering College, Inner Mongolia University of Science and Technology, Baotou 014010, China;
2.Information Office and Network Center, Inner Mongolia University of Science and Technology, Baotou 014010, China)

Abstract: With the highspeed development of Internet, information processing has become an indispensable tool for peple obtain useful information. So automatic text classification technology is especially important. The existing classification algorithm in the time complexity and space complexity meet the bottleneck, and can't satisfy people's needs, this paper puts forward the TFIDF algorithm based on Hadoop distributed platform, and gives the specific process of the algorithm, through the MapReduce programming realized TFIDF classification algorithm, and compares with the traditional serial algorithm,also in single and cluster mode with contrast experiment, the experiment proved that, the use of text categorization algorithm TFIDF realize high-speed effective classification of mass data.

Key words: text classification; MapReduce; parallelization; TFIDF algorithm

当今信息时代，数据膨胀的速度已远远超过人工分析它们的能力，如何在海量数据中快速地获得所需信息至关重要，因此自动文本分类技术尤为重要。文本分类是指依据文本内容由计算机根据某种自动分类算法，把文本判定为预先定义好的类别[1]。文本分类是数据挖掘的关键技术，为了提高分类质量，首先要实现算法并行化。

近几十年来，一系列统计学习文本分类方法被提出[2]，国内外对文本分类算法的研究很多，但大都存在一些局限性，特别是缺乏对海量文本数据的挖掘。云计算的出现为算法并行化带来了新的契机，很多科研人员和机构都在投入研究云计算。Hadoop 平台发布以来，很多专业人员致力于利用它对海量数据进行挖掘，目前已经实现了一些基于该平台的算法。本文研究 TFIDF 文本分类算法，并通过 MapReduce 编程，在单机和集群模式下研究 TFIDF 算法的并行化并进行实验验证，并与传统算法进行对比实验，实验表明，改进的算法提高了分类速度，有效地解决了海量数据的分类问题。

1 TFIDF 算法的实现

TFIDF 是一种用于资讯检索与资讯探勘的常用加权技术。在某一个特定的文档中，词频(TF)指某一具体给定的词语在这个文档中出现的次数。对于在某一特定文档里的词语 t_i，其词频可以表示为：

* 基金项目：国家自然科学基金资助项目(61163025)；教育部春晖计划资助项目(Z2009-1-01044)

$$\mathrm{tf}_{i,j}=\frac{m_{i,j}}{\sum_k m_{k,j}} \tag{1}$$

其中，$m_{i,j}$ 是该词在文档 d_j 中出现的次数，$\sum_k m_{k,j}$ 是文档 d_j 中所有字词出现次数之和。

逆向文件频率(IDF)即对一个词语普遍重要性的度量。某一特定词语的 IDF 值，可由总文件数除以包含该词语的文件的数目，再将二者之商取对数：

$$\mathrm{idf}_i=\lg\frac{M}{|\{j:t_i\in d_j\}|} \tag{2}$$

上式中，M 表示所选语料库中文件的总数目，$|\{j:t_i\in d_j\}|$ 表示包含词语 t_i 的文件数目。

由式(1)和式(2)可得到，单词的权重公式为[3]：

$$\mathrm{tfidf}_{i,j}=\mathrm{tf}_{i,j}\times\mathrm{idf}_i \tag{3}$$

TFIDF 算法是 Rocchoi 算法的一种，每篇文档表示成由特征项组成的向量。TFIDF 算法将同一类的所有文档向量相加，得到每一个类 c_j 的特征向量 $\vec{c}_j$。利用 TFIDF 算法测试一篇文档所属的类别时，通过计算这篇文档的向量 $\vec{d}_i$ 和每个类特征向量 $\vec{c}_j$ 的相似度距离 $\mathrm{sim}(\vec{d}_i,\vec{c}_j)$，相似度距离最大的类向量所属的类就是测试文档的类别[4]。

首先，对每一个文档进行 TFIDF 向量化，然后为每一类文档建立一个原型向量 c_j，如下式：

$$\vec{c}_j=\alpha\frac{1}{|c_j|}\sum_{\vec{d}\in c_j}\frac{\vec{d}}{\left|\left|\vec{d}\right|\right|}-\beta\frac{1}{|D-C_j|}\sum_{\vec{d}\in D-C_j}\frac{\vec{d}}{\left|\left|\vec{d}\right|\right|} \tag{4}$$

式中，α 和 β 是调节正例、负例相对影响程度的参数，$|c_j|$ 表示属于类别 j 的文档个数，$\left|\left|\vec{d}\right|\right|$ 为 $\vec{d}$ 的欧式长度。

在这个原型向量的结果集中，每个类别被表示成一个向量，代表学习到的模型。利用此模型就可以对新文档 d' 进行分类。首先对新文档 d' 利用 TFIDF 权值进行向量化，用 d' 表示；然后分别计算代表每一个类别的原型向量 c_j 与 d' 的余弦相似度；最后取最大的相似度值所对应的类别为 d' 的归属类别，即测试结果[5]。余弦相似度的计算公式如下：

$$H_{\mathrm{TFIDF}}(d')=\arg\max_{c_j\in C}\cos(\vec{c}_j,\vec{d})=\arg\max_{c_j\in C}\frac{\vec{c}_j\bullet\vec{d}'}{\left|\left|\vec{c}_j\right|\right|\bullet\left|\left|\vec{d}'\right|\right|} \tag{5}$$

TFIDF 算法是有监督的文本分类算法，它的训练集是已标记的文档，并且随着训练集规模的增大，分类效率、精度均显著提高[6]。

2 MapReduce 编程模型

分布式文件系统(HDFS)和 MapReduce 编程模型是 Hadoop 的主要组成部分。Hadoop 是一个能够对大数据进行分布式处理的框架，能够把应用程序分割成许多小的工作单元，并且把这些单元放到任何集群节点上执行[7]。MapReduce 模型的计算流程如图 1 所示。

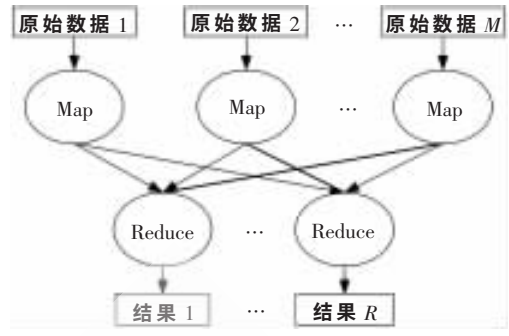


图 1 MapReduce 计算模型

分布式文件系统主要负责各节点上的数据的存储，并实现高吞吐的数据读写。MapReduce 计算模型的核心部分是 Map 和 Reduce 两个函数[8]。Map 的输入是 in_key 和 in_value，指明了 Map 需要处理的原始数据。Map 的输出结果是一组<key，value>对。系统对 Map 操作的结果进行归类处理。Reduce 的输入是(key，[value1… value m])。Reduce 的工作是将相同 key 的 value 值进行归并处理最终形成(key，final_value)的结果，所有的 Reduce 结果并在一起就是最终结果。其中 HDFS 和 MapReduce 的关系如图 2 所示。

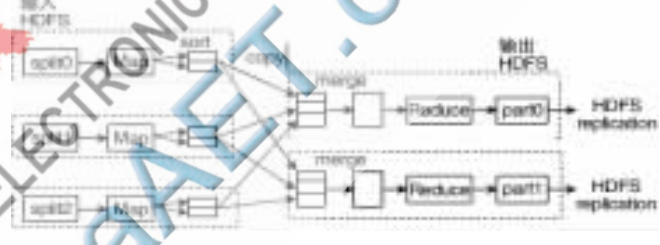


图 2 HDFS 和 MapReduce 的关系图

3 MapReduce 编程模型下的 TFIDF 算法

3.1 TFIDF 算法流程

Hadoop 分布式计算的核心思想就是任务的分割及并行运行。从 TFIDF 的计算公式可看出，它非常适合分布式计算求解。词频(TF)只与它所在文档的单词总数及它在此文档出现的次数有关。因此，可以通过数据分割，并行统计出文档中的词频 TF，加快计算速度。得到单词词频 TF 后，单词权重 TFIDF 的计算取决于包含此单词的文档个数。因此，只要能确定包含此单词的文档个数，即能以并行计算的方式实现 TFIDF 的求解。MapReduce 下计算 TFIDF 的整个处理流程如图 3 所示。主要包括统计每份文档中单词的出现次数、统计 TF 及计算单词的 TFIDF 值三个步骤。

3.2 实验与分析

实验选择 Sogou 文本分类语料库[6]作为数据集，文本分类语料库来源于 Sohu 新闻网站保存的大量经过编辑手工整理与分类的新闻语料与对应的分类信息。其分类体系包括几十个分类节点，汽车、财经、IT、健康、体育、旅游、教育、招聘、文化、军事等类别约十万篇文档。分布式环境包括 4 个节点：1 个 namenode、3 个 datanode。操作系统为 Ubuntu 12.04，内存 4 GB。

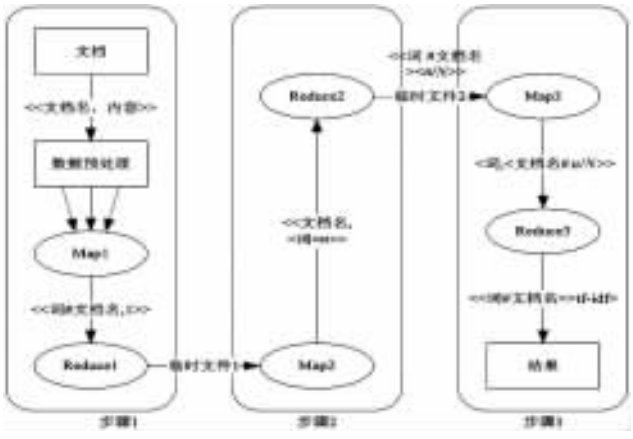


图 3 TFIDF 算法处理流程图

集群环境的配置：HDFS 的配置容量为 1.52 TB，4 个节点的 HDFS 配置如表 1 所示。

表 1 HDFS 配置容量

	Node1	Node2	Node3	Node4
HDFS 容量/GB	328.65	308.45	308.45	308.45
Map 任务数	0	2	2	2
Reduce 任务数	0	2	2	2

(1)单机与分布式环境下的对比

取数据集的 70%作为训练集，其余的 30%作为测试数据集进行实验。单机模式下的运行时间如表 2 所示。

表 2 单机模式下运行时间

数据集	训练时间/s	测试时间/s
文本分类语料库	1 025	362

Hadoop 下实现 Map/Reduce 编程。在 4 个节点上，Sogou 文本分类语料库数据集的运行时间如表 3 所示。

表 3 集群环境下运行时间

数据集	训练时间/s	测试时间/s
文本分类语料库	375	110

(2)MapReduce 编程的 TFIDF 算法和传统 TFIDF 算法

使用 MapReduce 下 TFIDF 算法与传统 TFIDF 算法进行对比实验，随着数据集大小的变化，结果如图 4 所示。

由图 4 可以看出，当数据量较小时，MapReduce 下 TFIDF 算法与传统 TFIDF 算法性能差距并不明显。由于

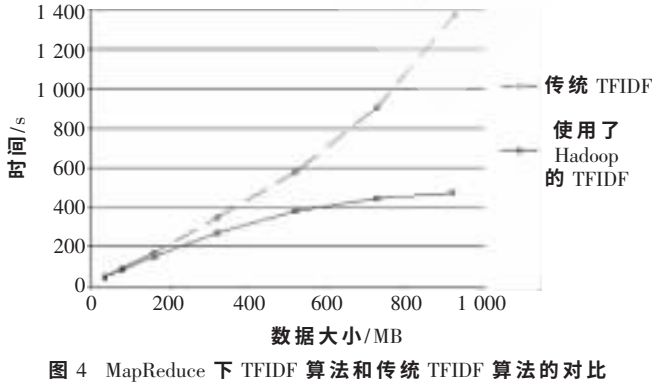
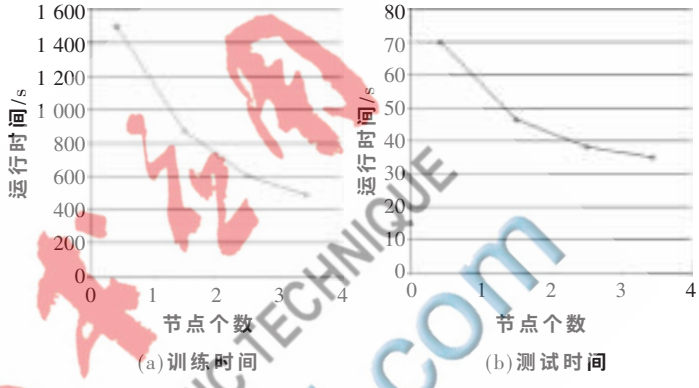


图 4 MapReduce 下 TFIDF 算法和传统 TFIDF 算法的对比

Hadoop 对数据进行的是分块处理，并且默认数据块大小为 64 MB，所以当存在很多小数据文件时，反而降低了运行速度，因此对小数据集 Hadoop 的优越性体现得并不明显。但是随着数据集增大，传统算法所需要的时间急剧增长，而应用了 Hadoop 框架的 TFIDF 算法所需要的时间只是呈线性增加，表现出了一定的算法优越性。

(3)不同节点数下的对应运行时间

图 5(a)和(b)分别显示了 Sogou 文本分类语料库随着节点数目由 1 增加到 4 时的训练时间和测试时间曲线。



(a)训练时间 (b)测试时间

图 5 训练时间与测试时间

本文通过在 Hadoop 平台下的 MapReduce 编程，对传统 TFIDF 算法进行了性能优化，并通过 3 组对比实验，验证了改进的 TFIDF 算法可取得更好的分类结果，可以很好地实现对海量数据的高效挖掘。

参考文献

[1] SEBASTIANI F.Text categorization[Z].Encyclopedia of Database Techologies and Applications，2005：683-687.

[2] Yang Yiming.An evaluation of statistical approaches to text categorizationg[J].Journal of Information Retrieval，1999，1(1/2)：67-68.

[3] 谢鑫军，何志均.一种单一表单工作流系统的设计和实现[J].计算机工程，1988，24(9)：53-55.

[4] 王宇.基于 TFIDF 的文本分类算法研究[D].郑州：郑州大学，2006.

[5] 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)：190-194.

[6] 搜狐研发中心.Sogou 文本分类语料库[OL].(2008-09)[2012-09-30].http://www.sogou.com/labs/dl/c.html.

[7] 刘鹏.实战 Hadoop-开启通向云计算的捷径[M].北京：电子工业出版社，2011.

[8] 李彬.基于 Hadoop 框架的 TF- IDF 算法改进[J].微型机与应用，2012，31(7)：14-16.

(收稿日期：2012-10-17)

作者简介：

赵伟燕，女，1987 年生，在读硕士研究生，主要研究方向：云计算。

王静宇，男，1976 年生，在读博士，副教授，主要研究方向：云计算，信息安全。