# 并行 MapReduce 模型下的一种改进型 KNN 分类算法

韦泽鲲，夏靖波，付 凯，申 建，陈 珍
(空军工程大学信息与导航学院，西安，710077)

**摘要** 大数据时代带来数据处理模式的变革，依托 Hadoop 分布式编程框架处理大数据问题是当前该领域的研究热点之一。为解决海量数据挖掘中的分类问题，提出基于一种双度量中心索引 KNN 分类算法。该算法在针对存在类别域的交叉或重叠较多的大数据，先对训练集进行中心点的确定，通过计算分类集与训练集中心点的欧式距离，确定最相似的 3 个类别，然后以余弦距离为度量，通过索引选择找出 $K$ 个近邻点，经过 MapReduce 编程框架对 KNN 并行计算加以实现。最后在 UCI 数据库进行比较验证，结果表明提出的并行化改进算法在准确率略有提高的基础上，运算效率得到了极大提高。

**关键词** 大数据；Hadoop；数据挖掘；双度量中心索引；MapReduce
**DOI** 10.3969/j.issn.1009-3516.2017.01.016
**中图分类号** TN391　　**文献标志码** A　　**文章编号** 1009-3516(2016)06-0092-07

# A Modified Bi-Measurement Central Index KNN Classification Algorithm Based on MapReduce

WEI Zekun, XIA Jingbo, FU Kai, SHEN Jian, CHEN Zhen
(Information and Navigation College, Air Force Engineering University, Xi′an 710077, China)

**Abstract**: Big data era has a revolution on the data processing mode, and the way dealing with bigdata by Hadoop distributed framework becomes one of the most popular research topics. Cloud computing model of clusters covers the shortage of the large amount of calculation and time-consuming of traditional non-distributed algorithm, meanwhile huge amounts of unstructured data increases the difficulty of data utilization. Aimed at the problem of solving the mass classification in data mining, this essay puts forward a algorithm, i. e. Bi-Measurement Central Index KNN Classification. And the algorithm mainly deals with in the field of the cross or overlap data. First, the essay is to find center of training data, then calculate the Euclidean distance between classifying data and training sites, and determine the most similar to the three categories. In addition, the essay selects k nearest neighbor points by the cosine distance metric, and computes the results by MapReduce. Finally, the UCI database is compared with and verified. The results show that though the amplitude of improving the accuracy by the proposed algorithm is not very great, the efficiency of the algorithm is greatly improved.

**收稿日期**：2015-12-01
**基金项目**：陕西省科技计划自然基金重点项目(2012JZ8005)
**作者简介**：韦泽鲲(1992—)，男，陕西西安人，硕士生，主要从事网络态势感知，大数据挖掘研究. E-mail：50217711k@sina.com
**引用格式**：*韦泽鲲，夏靖波，付凯，等. 并行 MapReduce 模型下的一种改进型 KNN 分类算法[J]. 空军工程大学学报(自然科学版)，2017，18(1)：92-98. WEI Zekun, XIA Jingbo, FU Kai, et al. A Modified Bi-Measurement Central Index KNN Classification Algorithm Based on MapReduce[J]. Journal of Air Force Engineering University(Natural Science Edition)，2017，18(1)：92-98.*

**Key words**：big data；Hadoop；data mining techniques；bi-measurement central index

随着大数据[1]时代的到来，人类社会的信息量成 PB 级别不断增加。越来越多的移动终端为接入网络带来了极其庞大的海量非结构化的数据，从这些数据中挖掘出有用知识的难度也不断增加，传统的分布式数据挖掘[2]已经越来越难以突破瓶颈。云计算下的并行化数据挖掘[3]的发展与应用是解决当下问题的必然趋势。

在数据挖掘领域中，分类算法是基础与主干研究内容之一，分类算法主要研究的是对事物的主动分类与归档问题，目的是对训练数据集进行挖掘，得以发现对目标数据的类别。云计算环境下经典的分类算法有基于 K-最近邻算法、Naïve Bayes 分类算法、SVM 分类算法、基于神经网络的算法、基于范例的推理和决策树的分类算法。其中，K 最近邻(KNN)[4]是一种适用于分类和回归的非参数分类算法。传统的 KNN 算法简单，直接，但是复杂度较高，需要单次的执行需要 $O(MN)$ 次运算。随着数据规模的增大和实时性要求的提高，该算法的准确性、伸缩性以及效率都不能满足海量动态数据场景下的应用的需求。而 Hadoop 的诞生很好地解决了海量数据的并行式计算和分布式处理的问题。Hadoop[5-8]是源于 Apache 基金会的一个开源项目，旨在解决大数据存储、处理以及管理问题。目前 Hadoop2.0，Yarn[6]以及基于内存计算的 Spark[7]是当前学术界、商业界最为流行的云计算法平台。Hadoop 核心由 HDFS、MapReduce(Yarn)[8]构成，是一个分布式系统基础架构，是一个可开发和运行处理大规模数据的软件平台。因此，我们采用 MapReduce 体系结构来处理该问题。

# 1　改进型双度量中心索引 BC-KNN 分类算法

## 1.1　传统 KNN 算法

KNN 算法是一种基于实例学习，或者懒惰学习的算法，其训练过程与一般分类不同，是一种局部近似的被动过程，边测试边训练边建立分类模型。该算法是一种经典的统计模式识别算法，因其简单而被广泛应用。KNN[7]最近邻算法的基本思想为：先计算每个测试样本点到其他样本点的距离，搜寻出该样本空间距离最邻近的 $K$ 个训练样本，即 $K$ 个最近邻，选出这些临近点中比例最大的点簇的类，判决为该测试样本的类别。这里，我们定义训练数据集为 $Y$ 类，各类记为 $C_i(1 \leqslant i \leqslant M)$，每个数据集样本都有 $N$ 个属性，待测数据为 $X$，具体步骤如下：

计算每一个样本与其他所有训练样本的距离(这里选用欧氏距离)，距离计算公为：

$$\mathrm{Distance}(X,Y) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \tag{1}$$

选出与待测样本最为相似的 $K$ 个训练样本，距离越大，则相似度越小，其计算公式为：

$$\mathrm{Sim}(X,Y) = \frac{1}{1+\mathrm{Distance}(X,Y)} \tag{2}$$

式中：$K$ 值得确定根据先验经验以及样本的不同而不同，一般先采用一个初始值，然后根据实验不断调整，最后确定最优的 $K$ 值。

在新的 $K$ 最近邻样本中，求出其与待测样本的总相似度：

$$\begin{aligned}&\mathrm{Sim}(X,C_i) = \sum_{j=1}^{K}\mathrm{Sim}(X,Y_j) * \delta(Y_j,C_i)\\&\delta(Y_j,C_i) = \begin{cases}1, & \text{If } Y_j \in C_i\\0, & \text{If } Y_j \notin C_i\end{cases}\end{aligned} \tag{3}$$

式中 $\delta(Y_j,C_i)$ 为类别属性函数，如果 $Y_j \in C_i$ 则为 1，$Y_j \notin C_i s$ 为 0。

判决出的总相似度最大的类，即为待测样本的类别。

$$\mathrm{Result}(X) = \mathrm{argmax}(\mathrm{Sim}(X,C_i)) \tag{4}$$

传统的 KNN 算法存在很多不足之处，首先当了类间分布极为不均时，如有些类别属性距离过大但是所占比重高，需要在计算向量的欧氏距离之前需要将特征向量进行归一化处理，或者分配相应的权重。其次，KNN 算法的计算复杂度较高，每次距离对比计算复杂度为 $O(MN)$，当样本空间巨大时，处理时间成本较高。最后一点，当数据集结构和类型差异较大时，采取不同的距离公式会随着不同数据集产生一定的误差，对样本类别判别的准确率会有一定影响。

## 1.2　改进型 KNN 算法

本算法针对的是基于多维的、非结构化的海量规模数据，以集群并行计算的方式进行处理，Bi-Measurement Central Index（双度量中心索引）KNN 算法改进了查找最近邻点的搜索策略，算法复杂度将得以降低，同时两种相似性度量模型将会提高其准确性。文献[9]通过改变对近邻点的搜索策略，提出了一种改进型的 KNN 算法。该算法在对最近邻的选择过程中，放弃传统算法中遍历所有样本的做法，而是通过逐渐逼近的思想来寻找最近

邻点,在计算复杂度和计算效率上能明显地优于传统的 KNN 算法。文献[10],提出多层差分 KNN 算法,根据类域对已知样本进行分层,既避免了传统改进算法中剪辑样本带来的判别误差,又大大降低了无效的计算量。

文献[11]Tinghuai Ma 等人提出了一种基于中心向量的快速 KNN 分类策略,以牺牲较少的准确性来换取效率的极大提高。文献[12]Rina S. Jain 针对多示例学习归类未知的问题,把特征选择加入到 KNN 算法中,比较了 Bayesian -KNN,Citation-KNN, Bayesian Citation-KNN 3 种算法,提出改进型 Bayesian Citation-KNN 在有监督与无监督 KNN 分类预测的普适性的思想。这对多源异构数据的分类预测具有很大的借鉴意义。文献[13] Yan-Ming Zhang 等人提出一种快速 KNN 图分类算法,通过 LSH(Location Sensitive Hash,LSH)位置敏感哈希函数对图数据集进行预处理,并分成若干等量的小数据集,然后分别对其进行传统 KNN 图算法相似度分类,实验结果效率和准确度都有较大提高。

本文针对逐渐逼近来寻找最近邻点的思想,在本算法的训练阶段,先得出数据样本属性字段的特征向量,计算出其中心向量点。

图 1 中,计算训练集中不同类数据各点到其它点欧式距离之和,对比排序,选取欧式距离最小的即为中心向量点。

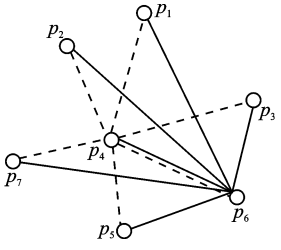


图 1 训练样本各类别中心点的确定

Fig. 1 Determination of the center point of each class of training samples

训练阶段,先计算待测点到各个类别中心点的距离,如有 7 个类别则计算 7 次相似性度量值,选取相似度最小的中心点作为索引的初始向量在训练集中随机选取一个训练向量 $\boldsymbol{P}_0$。此训练过程提前进行,不计入总的消耗时间中。以 $\boldsymbol{P}_0$ 为基点;计算在该向量空间中距离该向量最近的 $K$ 个向量,然后在这 $K$ 个向量中选取距离待测向量 $\boldsymbol{X}$ 最近的向量,以此向量为基点;选取距离 $\boldsymbol{P}_1$ 向量最近的 $K$ 个向量,然后在这 $K$ 个向量中选取距离待测向量 $\boldsymbol{X}$ 最近的向量 $\boldsymbol{P}_2$(如果 $\boldsymbol{P}_i$ 与 $\boldsymbol{P}_{i-1}$ 重合,则停止)……依次做迭代,已经做过距离测量的向量,不做重复计算,一直找到距离测试向量的距离最短的前 $K$ 个向量。

在图 2 中,以映射到三维空间为例,对于测试向量,随机找出某特征向量,记为 $\boldsymbol{P}_0$,依照上述方法,找出 $\boldsymbol{P}_0,\boldsymbol{P}_1,\ldots,\boldsymbol{P}_n$,如果 $K=4$,则在 $\boldsymbol{P}_n$ 附近测量找出最近邻的 4 个点。

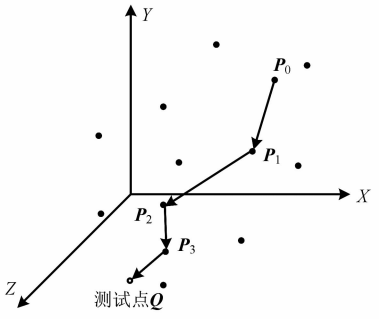


图 2 通过搜索选择最近邻点

Fig. 2 Search nearest neighbor points by searching algorithm

由于初始向量的选取可能造成基点最近邻的 $K$ 个向量都与待测向量距离较远,导致类别判定错误。因此在分类一开始,从训练样本 $M$ 个类中随机选取一个初始向量作为开始搜索的基点, 初始向量选择类别中心向量点作为基点,这样结果会产生 $\boldsymbol{G}_1,\boldsymbol{G}_2,\ldots,\boldsymbol{G}_m$ 个簇集,每个类的簇集中包含 $K$ 个向量点。因为不存在跨类现象,因此这些向量没有重合的,即产生了 $KM$ 次向量运算(以上所求的距离公式为欧氏距离)。将待测样本与这 $M$ 个簇集的所有点进行距离测算,这里采取的是余弦距离[9],因为在多维空间之中,2 个向量方向越相近,其余弦距离[9],即余弦相似度就越大。选出 $K$ 个最近邻,并用传统的 KNN 算法进行类别判别。

双向随机含义在于随机选取训练样本上的点,求其与待测样本的欧氏距离,算出点对之间的绝对距离,得出相似度值。然后再求待测样本与上一步所得训练样本的余弦距离,因为在多维空间之中,2 个向量方向越相近,其余弦距离[9],即余弦相似度就越大。之后找出 2 种相似度之间点的交集,进行标记,最后根据 $K$ 值判决出其类别。

本文结合了欧氏距离和余弦距离的特点,欧氏距离能够体现个体数值的绝对差异,余弦距离修正了可能存在的度量标准不统一的问题。中心向量的选择和双度量距离的交集的求解正是本算法创新点之一。

首先,分析计算各个数据样本属性字段的特征向量;

在训练集中($Y_i,1\leqslant \boldsymbol{i}\leqslant M$)的每个类别中找出中心点,通过上述迭代欧氏距离测量的方法,找出

各个类中距离待测样本最近邻的 $K$ 个样本点，记为 $\boldsymbol{G}_1,\boldsymbol{G}_2,\ldots,\boldsymbol{G}_m$，(这里 $m=3$)分别求出每个点与待测点之间的相似度，所用到的计算公式如下；

$$\text{Distance}(X,Y)=\sqrt{\sum_{i=1}^{N}(x_i-y_i)^2} \tag{5}$$

$$\text{Sim}(X,Y)_{Euclid}=\frac{1}{1+\text{Distance}(X,Y)} \tag{6}$$

计算待测点与 $\boldsymbol{G}_1,\boldsymbol{G}_2,\ldots,\boldsymbol{G}_m$ 这 $M$ 个簇集中各个点的余弦距离以及相似度，所用到的公式如下；

$$\text{Sim}(X,Y)_{\cos}=\text{Sim}(d_i,d_j)=\frac{\sum_{m=1}^{m}w_{im}w_{jm}}{\sqrt{\sum_{m=1}^{m}w_{im}^2}\sqrt{\sum_{m=1}^{m}w_{jm}^2}} \tag{7}$$

式中：$\boldsymbol{d}_i,\boldsymbol{d}_j$ 表示样本 $i,j(X,Y)$ 对应的特征向量；$\boldsymbol{w}_{im},\boldsymbol{w}_{jm}$ 表示 $i,j$ 的特征向量的第 $m$ 维属性的特征值(权重)；

求出满足最近邻条件的 2 种相似度的训练样本的交集，欧氏距离相似度产生 $km$ 个，取出前 $k$ 个，余弦相似度产生为 $k$ 个。

$$\{Y_j\mid 1\leqslant j\leqslant k\}=\text{Result}(Y_j,C_i)_{\text{Euclid}}\cap\text{Result}(Y_{j'},C_{i'})_{\cos} \tag{8}$$

判决出总相似度最大的类别。

$$\text{Sim}(X,C_i)=\sum_{j=1}^{K}\text{Sim}(X,Y_j)\delta(Y_j,C_i)$$
$$\delta(Y_j,C_i)=\begin{cases}1,\text{If } Y_j\in C_i\\0,\text{If } Y_j\notin C_i\end{cases} \tag{9}$$

$$\text{Result}(X)=\text{argmax}(\text{Sim}(X,C_i)) \tag{10}$$

本文中改进型 KNN 算法创新点在于采用训练集中心点确立索引起始点最近邻点的策略，大大降低了运算的复杂度，但会牺牲部分准确性；同时根据对比数据集的特点，采取不同的 2 种距离函数，结合 2 种距离模型，集成优点，得出 2 组相似度的，将低于相似度阀值的数据剔除，并取交集得出各类最近邻与测试样本的总相似度，判决相似度最大的类，提高准确性。

## 2　BC-KNN 算法的并行化设计与实现

### 2.1　Mapreduce 分布式框架概述

MapReduce 是 Hadoop 的核心编程框架[14-15]，主要是用于解决大规模数据集的并行计算问题。其核心思想为把大数据集按块进行分割(一般大小为 16～64 MB)，分别分配给集群的不同节点去完成，最后合并返回处理结果。该计算模式将一般的数据计算过程分为 Map 和 Reduce 2 个阶段，将数据表述为键值对＜key，value＞的形式，通过多个高次函数的串接，将数据的计算转化为一些列函数的执行。处理过程见图 3。

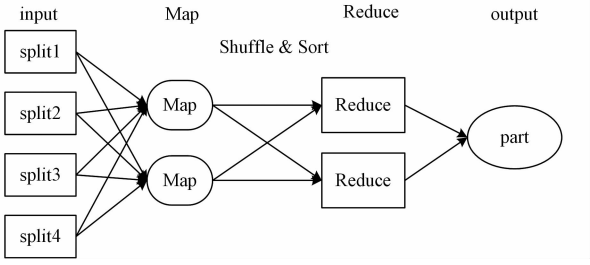


图 3　MapReduce 流程图

Fig. 3　Diagram of MapReduce

MapReduce 过程主要分为 Map 与 Reduce 2 个阶段，可用以下 2 个公式表述：

$$\text{Map}:<k1,v1>\rightarrow\text{list}(<k2,v2>) \tag{11}$$

$$\text{Reduce}:<k2,\text{list}(v2)>\rightarrow\text{list}(<k3,v3>) \tag{12}$$

用户首先读入数据，系统对数据进行分块，如图 3 中的 split 0～split $n$，这些数据块经处理生成＜$k$，$v$＞键值对。由用户编写的 Map 函数读入一系列键值对＜$k1$，$v1$＞，经过分析处理产生一组中间的＜$k2$，$v2$＞键值对，再由 Partitioner()类将这些中间结果指定的区分地写到输出文件中，随后 Combiner()类会对其进行合并，产生过于 $k2$ 的键值 list($v2$)，这样 Map 部分的处理就算基本完成。上一步的输出结果会作为输入进入 Reduce 函数中，经过组合(shuffle)、排序(sort)、聚集(reduce)3 个阶段[13]，对各个节点各个分区上的键值对进行分析处理，最后形成相对较小的键值对的集合 list(＜$k3$，$v3$＞)，即产生我们需要的结果。总而言之，MapReduce 这种“分而治之”[16]的思想非常适合大规模集群架构，是大数据处理的极佳选择。

### 2.2　基于 MapReduce 的 BC-KNN 算法

BC-KNN 算法的并行化实现主要解决 MapReduce 的 Map 函数和 Reduce 函数 2 个问题。本文中增加了 Combine 函数，Combine 类的作用就是在 Map 类执行结束后预先执行一次小规模的 Reduce 操作，以降低 Reduce 的运算负担。

Map 函数中，先设定参数 $k$，遍历训练样本，将属性值存入 TrainData，类别存入 TrainLabel，读入测试样本，存入 TestData；接着分别计算待测点与各类中心点的欧式距离，选取最小的 3 个中心点，采取搜索紧邻点的方式计算从 3 个类别中计算出 $3k$ 个样本点到待测点的欧式相似度。

Combine 函数中，计算上一步中待测点 $3k$ 个样本点的余弦相似度，遍历训练组集，将所得相似度 $S$

与优先级队列中的最小相似度进行比较。若 $S<=S_{max}$,则舍弃,遍历下一组。若 $S>S_{max}$,删除优先级队列中最小相似度的样本,将当前训练样本加入入优先级队列,由此得出 $k$ 个样本级。同理遍历欧氏相似度,取出前 $k$ 个优先级样本集。

Reduce 函数中,将欧式相似度前 $k$ 个与余弦相似度前 $k$ 个相交,得出 $t(t<=k)$ 个样本级,判断其类别。

MapReduce 函数中数据类型可以简化为<key,value>,Reducer 的输入类型是由 Map 类的输出类型决定的,因此设计类型时要注意传递性。本实验中 Map 输出将 key 值定义为距离,value 为类别。Context 实例为 Map 方法提供的,以便在系统内部的上下文中读写。

3.2.1 Map 函数设计

部分核心伪代码如下:

```
Input:Text key,Vector value
    Output:<Text,Vector>Context context
    //已经计算得出各类别中心点 Central[i]
    For(i=1: 7){
    Distance=EuclidDistance(d, Central[i]);
    SimEuclid=1/(1+Distance);}
    //选取最小 3 个类别,依次遍历算氏距离
    For(i=1:3){
    For(j=1:CentralTrainData[i]. Length)
    EuclidData [ ]. append ( SimEuclid. CentralTrainData [i] [j]);}
    For(i=0:k*3-1)
    context. write(EuclidData[i], distLable);
```

3.2.2 Combine 函数设计

```
Input:Text key,Vector value
    Output:Text key,Vector value,Context context
    For all key and value do
    Distance=CosDistance(d,EuclidData[ ]);
    Sort(CosData[k]);
```

3.2.3 Reduce 函数设计

```
Input:Text key,Vector value
    Output:Text key,Vector value,Context context
    ArrayList=CosData[k]∩Euclid[k];
    NewArrayList result;
    for(i=0:k) Result. add(key,ArrayList. get(i));
    context. write(key. Tradition KNN(result));
    context. write(key, Lable);
```

## 3 实验分析

文中实验所用的计算机操作系统为 Linux CentOS 7,配置如下:机器内存 1.8 G,CPU 位 Pentium(R)Dual-Core CPUE5300@2.60GHz x2,硬盘容量为 157.8 GB。其中,Hadoop 版本为 2.6.0,JAVA 版本为 JDK1.8.0。实验所用的开发环境为 Eclipse-luna-X86_64,并安装有 MapReduce 插件,编程语言为 java。Hadoop 集群具体配置与分工见图 4。

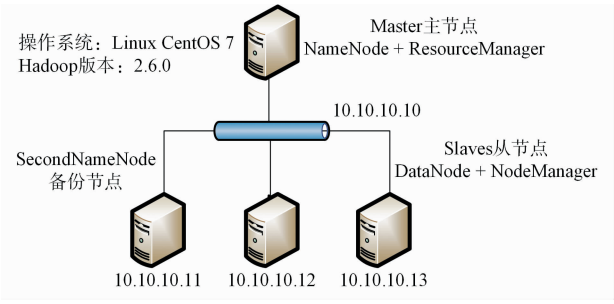


图 4 Hadoop v2.0(Yarn)集群配置及网络拓扑图

Fig. 4 Hadoop v2.0(Yarn) clusters configuration and network topology

实验数据采用 UCI 数据库的 Covertype Data,该数据集是依据地质变量来预测森林植被类型的机器学习数据集,其中样本数据为 581 012 个,每个样本维度为 54 维,对 Spruce/Fir(云杉/冷杉)、Lodgepole Pine(美国黑松)、Ponderosa Pine(美国黄松)、Cottonwood/Willow(三角叶杨/柳树)、Aspen(山杨)、Douglas-fir(道格拉斯冷杉)、Krummholz(高山矮曲林)这七种植被(编号依次由 1~7)的海拔、方向、斜坡、水文距离、不同时刻山坡情况、火源距离、土壤类型等 54 维度进行定量测算统计,文件大小为 75.2 MB。本实验中在数据集每种植被选取10 000 个作为训练样本,另外各选择 200 个共计1 400。通过查询 NameNode,DataNode 和 ResourceManager 上日志文件的查询,统计得出计算消耗时间。

图 5 为不同 K 值时对应的传统 KNN 算法与本文提出的 BC-KNN 算法的准确率对比图,通过 $K$ 值由 3~7 取不同值时计算得到的准确率数据可看出,$K$ 的取值对准确率有一定的影响。针对本实验数据集和实验算法,当 $K=4$ 时,集群运算分类的准确性达到最高。针对伪分布式的 Hadoop 集群(单一节点),针对不同 $K$ 值时传统 KNN 与 BC-KNN 算法耗费的时间进行实验,结果见表 1。通过实验数据可得出针对 covtype 样本,同等 $K$ 值条件下,BC-KNN 算法耗费时间远远小于传统 KNN 算法

的。优化的最近邻搜索算法弥补了传统 KNN 算法针对海量数据时计算量巨大的不足[18]，降低了计算量，提高了计算效率。同时可以看出随着 $K$ 值的增大，KNN 算法消耗的时间都不断增加，因而算法的优化在于在一定的 $K$ 值范围内，提高准确率，降低耗费时间。

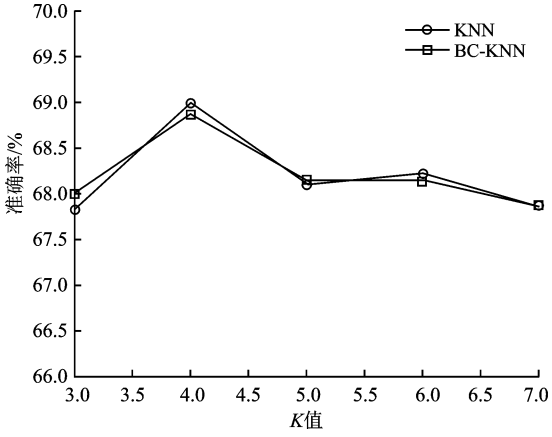


图 5　不同 $K$ 值时集群分类准确率趋势图

Fig. 5　$K$ value of the cluster classification accuracy trend chart

**表 1　传统 KNN 与 BC-KNN 算法耗费时间对比**

**Tab. 1　Time consuming contrast between KNN and BC-KNN**

| $K$ | 传统 KNN 耗费时间/s | BC-KNN 耗费时间/s |
|---|---|---|
| 3 | 2 650.725 | 920.983 2 |
| 4 | 2 684.344 | 929.737 9 |
| 5 | 2 689.038 | 926.004 5 |
| 6 | 2 733.851 | 926.029 7 |
| 7 | 2 765.808 | 956.931 5 |

Hadoop 单机时集群处理的优势就在于并行计算会将成比例的降低消耗的时间，当实验数据足够大时，集群节点足够多时，可忽略 namenode 与 datanode(ResourceManager 与 NodeManager)之间通信所需的开销，时间复杂度为 $O(ns)/p$，其中 $n$ 为训练样本数目，$s$ 为每个样本特征属性数目，$p$ 为节点总的 map 任务的个数。本文改进型 KNN 算法在不同数目节点以及单机下进行消耗时间对比分析，由图 6 是本文改进型 KNN 算法在不同数目节点的实验环境下，消耗的时间对比。除却 Master 与 Slavers 互相通信与 Task 分配管理以及心跳机制的同步，需要耗费一定的计算时间资源外，节点数量越大，消耗时间越小，且其消耗时间与节点数目近似成倍数递减。

Hadoop 集群节点数目小于 2 个时，消耗的时间要比单机下消耗的时间大的多。造成节点少时消耗反而巨大的原因有如下 2 点：①Master 与 Slavers 互相通信与 Task 分配管理以及心跳机制的同步，需要耗费一定的计算时间资源；②网络传输速度会对集群的运行时间产生一定影响，单机时本地内存数据传输，不受网络传输影响。而随着节点数目的增加，Hadoop 集群的优势显现出来，当节点数超过 3 个后，消耗时间不断降低，结果数据与实验预期大致相同。

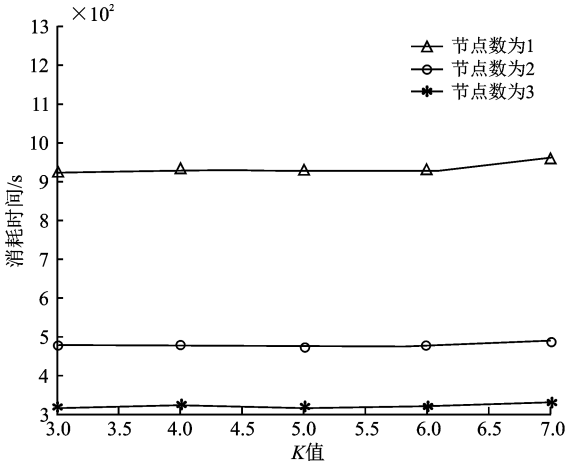


图 6　BR-KNN 算法不同节点数目时消耗时间对比图

Fig. 6　Chart of time consumption with different node number by BC-KNN algorithm

Hadoop 天生为处理大型文件而诞生，处理过多小文件会同时存在占用大量的内存空间，严重影响 NameNode 的性能，同时执行过程中频繁访问 DataNode，需不停地生成和释放相关资源。这样频繁调度极为耗费 Hadoop 框架的处理能力，成为影响计算性能的瓶颈[19]。本实验中因实验条件有限，计算节点数目较少成为实验不足之处，在下一步实验中，笔者将增加实验集群规模，采用规模更大、更加异构化的实验数据进行大数据的挖掘实验，同时对于异构类型数据将引入分层计算分类的思想，以进一步减少计算量，提高运算速度。

## 4　结语

随着云计算的深度和广度不断加深与扩展，对于大数据的高效准确的挖掘变得越来越具有价值，而开源平台 Hadoop、Spark 的大范围应用与推广，基于 MapReduce 的并行挖掘算法亟需被优化更新，使其更贴合大数据模式。本文提出的 BC-KNN 算法，相比于传统 KNN 算法，在保证分类准确率不降低且略有提高的条件下，运算效率极大提高，本算法以及实验具有一定的理论意义和实际应用价值。

**参考文献(References)：**

[1]　VIKTOR MAYER-SCHÖNBERGER，KENNETH

CUKIER. BIG DATA [M]. Hodder Export. 2013.

[2] JAIN Saloni, ZHANG Yanqing. Real-Time Social Network Data Mining For Predicting the Path For A Disaster[J]. Georgia State University. Computer Science, 2015, 64, (4): 827-836.

[3] LIU L, ZHANG J. Privacy Preserving Data Mining for Numerical Matricers, Social Networks, and Big Data [D]. Kentucky, USA: University of Kentucky, 2015.

[4] VeniVidiVicipedia. k-nearest neighbors algorithm [EB/OL]. [2017-2-11]. https//en. m. wikipeida. org/wiki/K-nearest_neighbor_algorithm. html.

[5] The Apache Software Foundation. Apache Hadoop [EB/OL]. [2017-2-11]. http:// Hadoop. apahce. org/index. html.

[6] The Apache Software Foundation. Apache Hadoop YARN [EB/OL]. [2017-2-11]. http://hadoop. apache. org /docs/current2/ hadoop-yarn/ hadoop-yarn-site/ YARN. html.

[7] LAMPI J. Large-Scale Distributed Data Management and Processing Using R, Hadoop and MapReduce. [D]. Oulu, Finland: University of Oulu, 2015.

[8] HUBREGTSEN T S, WANDSWORTH B I, KINGDOM U. Evaluation of Different Storage Systems for Apache Spark and Apache Hadoop[D]. Delft, Holland: Delft University of Technology, 2015.

[9] 钱强,庞林斌,高尚. 一种基于改进型 KNN 算法的文本分类方法[J]. 江苏科技大学学报(自然科学版), 2013, 27(4): 381-385.
QIAN Q, PANG L B, GAO S. A Text Classification Method Based on Improved KNN Algorithm [J]. Journal of Jiangsu University of Science and Technology(Natural Science Edition), 2013, 27(4): 381-385. (in Chinese)

[10] 耿丽娟,李星毅. 用于大数据分类的 KNN 算法研究[J]. 计算机应用研究, 2014, 31(5): 1342-1345.
GENG L J, LI X Y. Improvements of KNN Algorithm for Big Data Classification [J]. 2014, 31(5): 1342-1345. (in Chinese)

[11] MA T H, YAN Y G, TIAN W. A Fast KNN Algorithm Based on Centre Vector Condensing[J]. Journal of Computational Information Systems, 2012, 8(23): 9792-9798.

[12] JAIN S. Study of Different Multi-Instance LearningkNN Algorithms [J]. International Journal of Computer Applications Technology and Research, 2014, 3(7): 460-463.

[13] ZHANG Y M, HUANG K, GENG G. Fast kNN Graph Construction with Locality Sensitive Hashing [C]// The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. 2013: 660-674.

[14] VERMA A, ZEA N, CHO B, et al. Breaking the MapReduce Stage Barrier [C]//Proc of 2010 IEEE Int Conf on Cluster Computing (CLUSTER'10). Piscataway, NJ: IEEE, 2010: 235-244.

[15] FAN L, LI H, LI C F. The Improvement and Implementation of Distributed Item-Based Collaborative Filtering Algorithm on Hadoop[C]// Proceedings of the 34th Chinese Control Conference. 2015: 9078-9083.

[16] PERERA S, GUNARATHNE T. Hadoop MapRecuce 实战手册[M]. 杨卓荦,译. 北京:人民邮电出版社, 2015.

[17] 王晓华. MapReduce2.0 源码分析与编程实战[M]. 北京:人民邮电出版社, 2014.
WANG X H. MapReduce 2.0 Code Analysis and Experiment [M]. Beijing: Posts and Telecom Press, 2014. (in Chinese)

[18] 鲁婷,王浩,姚宏亮. 一种基于中心文档的 KNN 中文文本分类算法[J]. 计算机工程与应用, 2011, 47(2): 41-44.
LU T, WANG H, YAO H L. K-Nearest Neighbor Chinese Text Categorization Algorithm Based on Center Documents[J]. Computer Engineering and Applications, 2011, 47(2): 41-44. (in Chinese)

[19] TOM WHITE. Hadoop: The Definitive Guide [M]. Beijing: O'Reilly, 2015.

(编辑:徐楠楠)