

**课 程 实 验 报 告**

**课程名称： 大数据分析**

**专业班级：**

**学 号：**

**姓 名：**

**指导教师：**

**报告日期：**

**计算机科学与技术学院**

**目录**

[实验一 wordCount算法及其实现 1](#_Toc57104249)

**[1.1实验目的](#_Toc57104250)** [1](#_Toc57104250)

**[1.2 实验内容](#_Toc57104251)** [1](#_Toc57104251)

**[1.3 实验过程](#_Toc57104252)** [1](#_Toc57104252)

[1.3.1 编程思路 1](#_Toc57104253)

[1.3.2 遇到的问题及解决方式 1](#_Toc57104254)

[1.3.3 实验测试与结果分析 1](#_Toc57104255)

**[1.4 实验总结](#_Toc57104256)** [1](#_Toc57104256)

[实验二 PageRank算法及其实现 2](#_Toc57104257)

**[2.1实验目的](#_Toc57104258)** [2](#_Toc57104258)

**[2.2 实验内容](#_Toc57104259)** [2](#_Toc57104259)

**[2.3 实验过程](#_Toc57104260)** [2](#_Toc57104260)

[2.3.1 编程思路 2](#_Toc57104261)

[2.3.2 遇到的问题及解决方式 2](#_Toc57104262)

[2.3.3 实验测试与结果分析 2](#_Toc57104263)

**[2.4 实验总结](#_Toc57104264)** [2](#_Toc57104264)

[实验三 关系挖掘实验 3](#_Toc57104265)

**[3.1 实验内容](#_Toc57104266)** [3](#_Toc57104266)

**[3.2 实验过程](#_Toc57104267)** [3](#_Toc57104267)

[3.2.1 编程思路 3](#_Toc57104268)

[3.2.2 遇到的问题及解决方式 3](#_Toc57104269)

[3.2.3 实验测试与结果分析 3](#_Toc57104270)

**[3.3 实验总结](#_Toc57104271)** [3](#_Toc57104271)

[实验四 kmeans算法及其实现 4](#_Toc57104272)

**[4.1实验目的](#_Toc57104273)** [4](#_Toc57104273)

**[4.2 实验内容](#_Toc57104274)** [4](#_Toc57104274)

**[4.3 实验过程](#_Toc57104275)** [5](#_Toc57104275)

[4.3.1 编程思路 5](#_Toc57104276)

[4.3.2 遇到的问题及解决方式 5](#_Toc57104277)

[4.3.3 实验测试与结果分析 5](#_Toc57104278)

**[4.4 实验总结](#_Toc57104279)** [5](#_Toc57104279)

# 实验一 wordCount算法及其实现

## **1.1实验目的**

1、理解map-reduce算法思想与流程；

2、应用map-reduce思想解决wordCount问题；

3、（可选）掌握并应用combine与shuffle过程。

## **1.2 实验内容**

提供9个预处理过的源文件（source01-09）模拟9个分布式节点，每个源文件中包含一百万个由英文、数字和字符（不包括逗号）构成的单词，单词由逗号与换行符分割。

要求应用map-reduce思想，模拟9个map节点与3个reduce节点实现wordCount功能，输出对应的map文件和最终的reduce结果文件。由于源文件较大，要求使用多线程来模拟分布式节点。

学有余力的同学可以在map-reduce的基础上添加combine与shuffle过程，并可以计算线程运行时间来考察这些过程对算法整体的影响。

提示：实现shuffle过程时应保证每个reduce节点的工作量尽量相当，来减少整体运行时间。

## **1.3 实验过程**

### 1.3.1 编程思路

### 1.3.2 遇到的问题及解决方式

### 1.3.3 实验测试与结果分析

## **1.4 实验总结**

# 实验二 PageRank算法及其实现

## **2.1实验目的**

1、学习pagerank算法并熟悉其推导过程；

2、实现pagerank算法，理解阻尼系数的作用；

3、将pagerank算法运用于实际，并对结果进行分析。

## **2.2 实验内容**

提供的数据集包含邮件内容（emails.csv），人名与id映射（persons.csv），别名信息（aliases.csv），emails文件中只考虑MetadataTo和MetadataFrom两列，分别表示收件人和寄件人姓名，但这些姓名包含许多别名，思考如何对邮件中人名进行统一并映射到唯一id？（提供预处理代码preprocess.py以供参考）。

完成这些后，即可由寄件人和收件人为节点构造有向图，不考虑重复边，编写pagerank算法的代码，根据每个节点的入度计算其pagerank值，迭代直到误差小于10-8

实验进阶版考虑加入teleport β，用以对概率转移矩阵进行修正，解决dead ends和spider trap的问题。

输出人名id及其对应的pagerank值。

## **2.3 实验过程**

### 2.3.1 编程思路

### 2.3.2 遇到的问题及解决方式

### 2.3.3 实验测试与结果分析

## **2.4 实验总结**

# 实验三 关系挖掘实验

## **3.1 实验内容**

**必做：**

1. 实验内容

编程实现Apriori算法，要求使用给定的数据文件进行实验，获得频繁项集以及关联规则。

2. 实验要求

以Groceries.csv作为输入文件

输出1~3阶频繁项集与关联规则，各个频繁项的支持度，各个规则的置信度，各阶频繁项集的数量以及关联规则的总数

固定参数以方便检查，频繁项集的最小支持度为0.005，关联规则的最小置信度为0.5

**加分项：**

1. 实验内容

在Apriori算法的基础上，要求使用pcy或pcy的几种变式multiHash、multiStage等算法对二阶频繁项集的计算阶段进行优化。

2. 实验要求

以Groceries.csv作为输入文件

输出1~4阶频繁项集与关联规则，各个频繁项的支持度，各个规则的置信度，各阶频繁项集的数量以及关联规则的总数

输出pcy或pcy变式算法中的vector的值，以bit位的形式输出

 参数不变，频繁项集的最小支持度为0.005，关联规则的最小置信度为0.5

## **3.2 实验过程**

### 3.2.1 编程思路

### 3.2.2 遇到的问题及解决方式

### 3.2.3 实验测试与结果分析

## **3.3 实验总结**

# 实验四 kmeans算法及其实现

## **4.1实验目的**

1、加深对聚类算法的理解,进一步认识聚类算法的实现；

2、分析kmeans流程,探究聚类算法院里；

3、掌握kmeans算法核心要点；

4、将kmeans算法运用于实际，并掌握其度量好坏方式。

## **4.2 实验内容**

提供葡萄酒识别数据集，数据集已经被归一化。同学可以思考数据集为什么被归一化，如果没有被归一化，实验结果是怎么样的，以及为什么这样。

同时葡萄酒数据集中已经按照类别给出了1、2、3种葡萄酒数据，在cvs文件中的第一列标注了出来，大家可以将聚类好的数据与标的数据做对比。

编写kmeans算法，算法的输入是葡萄酒数据集，葡萄酒数据集一共13维数据，代表着葡萄酒的13维特征，请在欧式距离下对葡萄酒的所有数据进行聚类，聚类的数量K值为3。

在本次实验中，最终评价kmean算法的精准度有两种，第一是葡萄酒数据集已经给出的三个聚类，和自己运行的三个聚类做准确度判断。第二个是计算所有数据点到各自质心距离的平方和。请各位同学在实验中计算出这两个值。

实验进阶部分：在聚类之后，任选两个维度，以三种不同的颜色对自己聚类的结果进行标注，最终以二维平面中点图的形式来展示三个质心和所有的样本点。效果展示图可如图1.1所示。



图4.1 葡萄酒数据集在黄酮和总酚维度下聚类图像（SSE为距离平方和，Acc为准确率）

## **4.3 实验过程**

### 4.3.1 编程思路

### 4.3.2 遇到的问题及解决方式

### 4.3.3 实验测试与结果分析

## **4.4 实验总结**