

## 实验 4: K-均值聚类

### 1 实验要求

#### 实验任务

使用 Hadoop MapReduce 实现课堂上介绍的“K-Means 聚类算法”。

---

**Algorithm 1:** K-Means 算法

---

**Input:** Dataset  $D = \{x_1, x_2, \dots, x_m\}$ ; 聚类簇数  $k$ ;

**Output:** 簇划分  $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$

从  $D$  中随机选择  $k$  个样本作为初始均值向量  $\{\mu_1, \mu_2, \dots, \mu_k\}$

**repeat**

    令  $C_i = \emptyset (1 \leq i \leq k)$

**for**  $j = 1, 2, \dots, m$  **do**

        计算样本  $x_j$  与各均值向量  $\mu_i (1 \leq i \leq k)$  的距离:  $d_{ji} = \|x_j - \mu_i\|_2$ ;

        根据距离最近的均值向量确定  $x_j$  的簇标记:  $\lambda_j = \arg \min_{i \in \{1, 2, \dots, k\}} d_{ji}$ ;

        将样本  $x_j$  划入相应的簇:  $C_{\lambda_j} = C_{\lambda_j} \cup \{x_j\}$ ;

**for**  $i = 1, 2, \dots, k$  **do**

        计算新均值向量:  $\mu'_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$ ;

**if**  $\mu'_i \neq \mu_i$  **then**

            将均值向量  $\mu_i$  更新为  $\mu'_i$

**else**

            保持当前均值向量不变

**until** 当前均值向量均未更新;

---

注意为了便于验证实验结果, 本次实验不需要随机选择初始聚类中心, 而是使用给定的初始聚类中心。同时, 由于给出了初始聚类中心聚类簇数也是确定的, 在本次实验中  $k = 20$ 。而实验的基本任务便是在 K-Means 算法完成后输出最终得到的聚类中心。

#### 输出格式

本次实验的基本任务是输出 K-Means 聚类算法完成后生成的聚类中心, 共有 20 个聚类中心。输出格式如下

聚类中心 ID(0-19)\TAB 聚类中心向量 (向量的各分量以 , 分隔)

下图展示了输出文件的一个片段:

```
0 4.49367,4.501744,4.49628,4.49701,4.4969,4.500036,4.500506,4.491124,4.499306,4.503578
1 14.50264,14.500742,14.495192,14.506466,14.495812,14.50276,14.497252,14.501858,14.496668,14.497848
2 24.501768,24.50761,24.50257,24.495495,24.497847,24.506466,24.502674,24.507177,24.501446,24.507666
```

## 选作内容

该部分选作内容不做要求，仅供感兴趣的同学尝试练习。

上一个任务中我们利用 K-Means 算法对数据进行聚类，最终得到各个簇的聚类中心。那么如何将属于不同簇的数据划分到不同的文件中？这是很自然进一步要考虑的问题。

选作部分的任务就是将所属不同聚类的数据划分到不同的文件中。即产生  $k$  个输出文件，每个文件对应一个簇，文件中存储属于该簇的数据。

**提示：** Hadoop 的 "MultipleOutputs"

## 2 实验数据

本次实验的数据包括一个数据集和一个存储初始聚类中心的文件。数据集中包含 10000000 条数据，每条数据就是一个向量。数据集中的数据有 11 维，其中第 1 维是该数据条目的 ID，后 10 维是向量的各个分量。

下图展示了数据集和初始聚类中心文件的一个片段：

1 4 1 6 7 4 1 6 1 7 5	0 11,11,6,4,12,5,4,12,6,9
2 3 2 5 6 7 7 2 7 0 3	1 22,18,15,14,20,17,19,16,13,14
3 7 1 0 9 4 8 4 6 7 1	2 31,30,32,26,32,32,28,32,24,32
4 3 9 2 0 8 0 3 9 6 9	3 37,36,36,37,37,33,41,39,36,33
5 8 6 9 6 5 3 6 0 2 0	4 50,45,46,50,47,43,50,46,46,43
6 5 3 3 5 0 8 0 0 6 9	5 57,53,56,55,59,57,62,53,58,54
7 6 7 9 8 0 3 5 0 0 2	6 69,63,67,67,67,71,70,69,69,65
8 8 5 7 5 6 1 4 6 2 6	7 78,82,81,77,82,73,81,81,77,76
9 3 5 1 2 0 2 4 2 9 8	8 92,92,92,88,89,88,88,84,91,87
10 3 7 0 4 3 7 7 9 4 9	9 95,97,93,94,95,94,102,97,95,98

(a) dataset

(b) initial centers

**单机测试样例：**提供完整数据集的一个子集作为单机测试样例，单机测试样例中有 10000 条数据。该数据集主要供本地调试使用。

**完整数据集：**完整数据集位于集群中的 HDFS 存储上，存储位置为 `hdfs://master001:9000/data/kmeans`

注意最终每个小组的程序必须在课程指定集群上运行，而且输入数据集是完整数据集。结果输出到集群的 HDFS 上。

## 3 提示

- 每一轮 K-Means 运算，即计算一次新的聚类中心作为一个 MapReduce 任务。需要在 Job Driver 中控制迭代，在下一轮任务中需要利用上一轮任务产生的聚类中心（利用 Distributed Cache 或 HDFS 文件系统接口）。直到聚类中心不再变化时，终止迭代。

- 为了维持 MapReduce 任务迭代运行，注意处理输出文件夹（删除或更换）和本次生成的聚类中心。
- 可以使用自定义 Writable 的方式减少字符串和浮点数组之间的反复转换。

## 示例代码

由于本次实验可能需要使用 HDFS 文件系统接口，下面提供一些使用该接口的示例代码以供参考，更详细的功能请同学们自行查阅文档。

---

```
// 删除文件或目录
public static void deletePath(String pathStr, boolean isDeleteDir) throws IOException {
    if(isDeleteDir) pathStr = pathStr.substring(0, pathStr.lastIndexOf('/'));
    Path path = new Path(pathStr);
    Configuration configuration = new Configuration();
    // 获取 HDFS 文件系统
    FileSystem fileSystem = path.getFileSystem(configuration);
    fileSystem.delete(path, true);
}
```

---

---

```
// 将一个文件的内容拷贝到另一个文件中
public static void copyFile(String from_path, String to_path) throws IOException {
    Path path_from = new Path(from_path);
    Path path_to = new Path(to_path);
    Configuration configuration = new Configuration();
    // 获取 HDFS 文件系统
    FileSystem fileSystem = path.getFileSystem(configuration);

    FSDataInputStream inputStream = fileSystem.open(path_from);
    LineReader lineReader = new LineReader(inputStream, configuration);

    FSDataOutputStream outputStream = fileSystem.create(path_to);

    Text line = new Text();

    while(lineReader.readLine(line) > 0) {
        String str = line.toString() + "\n";
        outputStream.write(str.getBytes());
    }

    lineReader.close();
    outputStream.close();
}
```

---

## 4 实验报告要求

在最后提交的压缩包中，除了包含源代码、JAR 包、JAR 包执行方式说明，还需要包含一个实验报告。实验报告中请包含：

1. Map 和 Reduce 的设计思路（含 Key、Value 类型）。
2. MapReduce 中 Map 和 Reduce 的伪代码（或者带注释的实际代码，如果使用实际代码，请做好排版）。
3. 输出结果文件的部分截图。输出结果文件在 HDFS 上的路径（某些情况下助教会检查 HDFS 上的输出文件）。
4. 请在报告中包含在集群上执行作业后，Yarn Resource Manager 的 WebUI 执行报告内容。注意只需要 K-Means 任务中最后一个任务的执行报告内容即可，如果实验中包含了选做内容，也需要选做任务的执行报告。