哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

大数据分析

实验二

(2019 年度春季学期)

姓　　名　朱明彦
学　　号　1160300314
学　　院　计算机学院
教　　师　杨东华、王金宝

计算机科学与技术学院

目录

实验二 聚类与分类

第 1 章 实验目的

掌握对数据使用聚类分析和分类分析，并理解其在大数据环境下的实现方式。

第 2 章 实验环境

- Ubuntu 16.04
- Hadoop 2.7.1

第 3 章 实验过程及结果

3.1 聚类分析

3.1.1 KMeans 聚类分析

主要思想 利用两类 Mapper 和 Reducer，其中第一对 Mapper-Reducer 主要用于中心点的选择，即初始化等工作；第二对 Mapper-Reducer 主要用于中心点的选择，即初始化等工作；第二对 Mapper-Reducer 主要用作迭代过程。

对于 K 值的选择，参考 CMU 在 2014 年春季的 10-605 [1]，使用 8 或者 12 作为聚类中心数。

第一类 Mapper

- 输入：原始数据
- 输出：`(1,` 原始数据中的一条`)`，共 K 个。
- 随机选择 K 个元素作为初始化的聚簇中心点，利用 `run` 函数实现。

由于此处仅仅需要 K 个元素作为初始化的聚簇中心点，所以只能使用 1 个第一类 Mapper 处理原始数据。

第一类 Reducer

- 输入：`(1, [`$c_0, c_1, \ldots, c_{k-1}$`])`，其中 $c_i, i \in \{0, 1, k-1\}$ 为原始数据中的一条。
- 输出：`(`i`,` c_i `+ \t + '-1')`，其中 i 为聚簇编号，\t 为制表符，加法为定义在 String 上的加法，即字符串的连接。

对于第一类 Reducer 而言，其输入的元组 Key 均为 1，所以仅有 1 个第一类 Reducer。

第二类 Mapper

- 输入：原始数据
- 输出：(clusterCenterID, $v;minDis$)，其中 Key 为 clusterCenterID，即该元组距离最近的聚类中心的编号；Value 为 $v;minDis$，其中 v 为该条原始数据，$minDis$ 为该原始数据与最近的聚类中心的欧式距离，二者以英文分号“;”分割。

第二类 Reducer

- 输入：(clusterCenterID, [$v_0;minDis_0$, $v_1;minDis_1$, ...])
- 输出：(clusterCenterID, new_c + \t + disSum)，其中 new_c 为属于该聚簇的计算出的新的聚类中心，disSum 为所有属于该聚簇的元素到该中心的距离和，用于判断 Kmeans 迭代收敛。

最终的 Kmeans 实现步骤如下，相关结果如图3.1所示。

1. 使用 1 个第一类 Mapper 随机取 K 个聚类中心，利用 Reducer 将结果存入 HDFS。
2. 读入上一轮（或者随机取的 K 个元素）中心点，并利用 Configuration 保存中心点。
3. 利用第二类 Mapper 计算每个元素所属的聚簇。
4. 利用第二类 Reducer 重新计算聚簇中心。
5. 如果收敛，算法结束；否则重新返回第 2 步。

3.1.2 GMM（混合高斯模型）聚类分析

3.2 分类分析

3.2.1 朴素贝叶斯

原理 由于使用的数据每一维特征都是连续型的数据，所以其处理与离散型的朴素贝叶斯处理有所不同。**因此，假设数据的每一维都符合高斯分布，而高斯分布的均值和方差均通过训练数据中的均值和方差来代替**。数据第 i 维取值为 x_i 的类条件概率为：

$$P(X_i = x_i|Y = y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

其中 y_j 为第 j 类，σ_{ij}, μ_{ij} 分别为第 j 类第 i 维样本数据的均值和方差。具体实现时，利用了两类不同的 Mapper 和 Reducer，其中第一对 Mapper 和 Reducer 主要用来计算训练数据中类别的先验；而第二对 Mapper 和 Reducer 用于处理计算后验并确定每一个样本所属的类别。

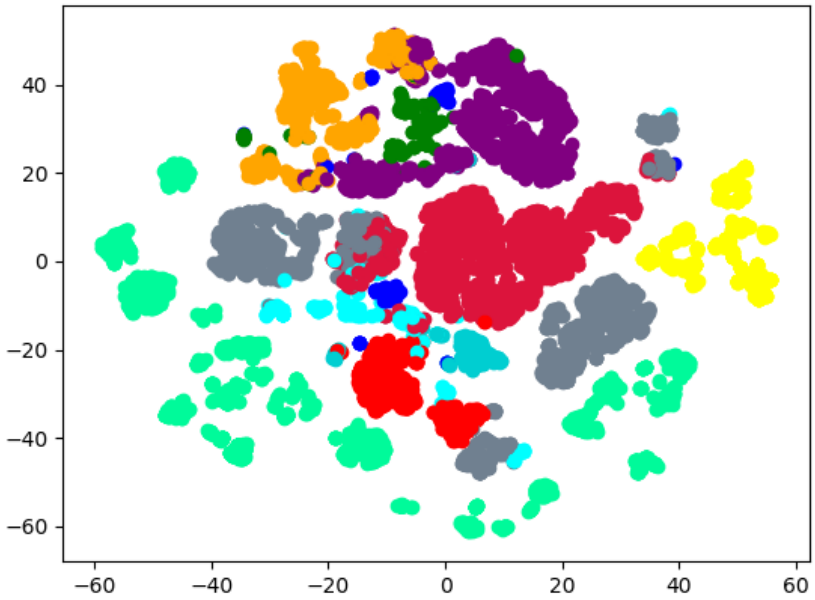

图 3.1: KMeans 聚类结果

第一类 Mapper

- 输入：训练数据
- 输出：(label_k，v_k)，其中 label 为该样本中标记的类别编号，k 为属性的第 k 维，v_k 为该样本第 k 维属性的取值。

第一类 Reducer

- 输入：(label_k，[$v_{k0}, v_{k1}, \ldots$])
- 输出：(label_k，$mean_k$ + \t + var_k)，即计算出属于 label 类的第 k 维训练数据的均值和方差，另加法为字符串的连接。

第二类 Mapper

- 输入：测试数据
- 输出：(compute_label，label)，其中 compute_label 为朴素贝叶斯得到的类别编号，而 label 为数据中原本标注的类别编号。

第二类 Reducer

- 输入：(compute_label，[label$_0$，label$_1$，...])

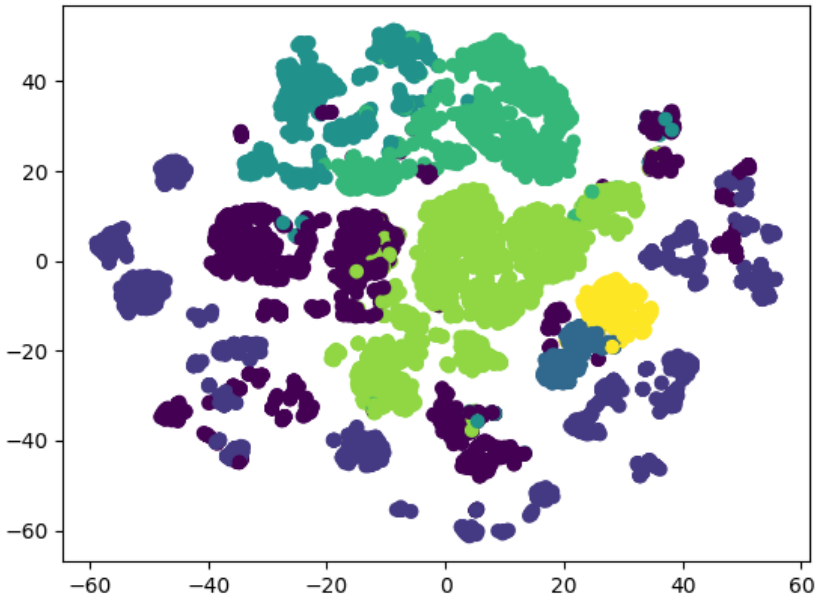

图 3.2: GMM 聚类结果

- 输出：`(compute_label, correct + \t + wrong)`，其中 `correct` 为正确分类样本数目，`wrong` 为错误分类数目。

3.2.2 逻辑回归

第 4 章 实验心得

可以分享您在实验环境搭建、程序编写和调试以及结果分析过程中遇到的问题和解决方法。

A 参考文献

参考文献

[1] K-Means Clustering on MapReduce, CMU 10-605 2014 Spring.