机器学习概论 实验报告

Lab4：K-Means

2020 年 12 月 25 日

目录

1 算法介绍

K-Means 是一种 原型聚类, 此类算法假设聚类结构能够通过一组原型刻画.

1.1 K-Means 理论基础

给定数据集给定数据集 $D=\{x_1,\cdots,x_m\}$, kmeans 算法能够针对聚类所得簇划分 $\mathcal{C}=\{C_1,\cdots,C_k\}$, 最小化平方误差, 将数据集 D 划分为 k 个簇.

$$E=\sum_{c=1}^{k}\sum_{x\in C_c}=\|x-\mu_c\|_2^2,\quad \mu_c=\frac{1}{|C|}\sum_{x\in C_c}x$$

这里 k 的值(也就是 kmeans 中的 k), 是人为规定的. 此外数据之间的相似度又欧氏距离进行度量, 两个样本欧氏距离越近, 相似度越高.

1.2 K-Means 算法

Algorithm 1 K-MEANS(H,x)

Require: 样本集 $D=\{x_1,\cdots,x_m\}$
Require: 聚类簇数 k

function K-MEANS(D)
  repeat
    令 $C_i=\varnothing(1\le i\le k)$
    for $j=1,\cdots,m$ **do**
      计算样本 x_j 与各均值向量距离 $d_{ji}=\|x_j-\mu_i\|_2$
      求最近簇标记: $\lambda_j=\arg\min_{i\in\{1,2,\cdots,k\}}d_{ji}$
      将样本 x_j 划入相应簇: $C_{\lambda_j}=C_{\lambda_j}\cup\{x_j\}$
    for $i=1,\cdots,k$ **do**
      计算新的均值向量: $\mu_i'=\frac{1}{|C_i|}\sum\limits_{x\in C_i}x$
      如果不同则更新.
  until 当前均值向量均未更新
  return 簇划分结果

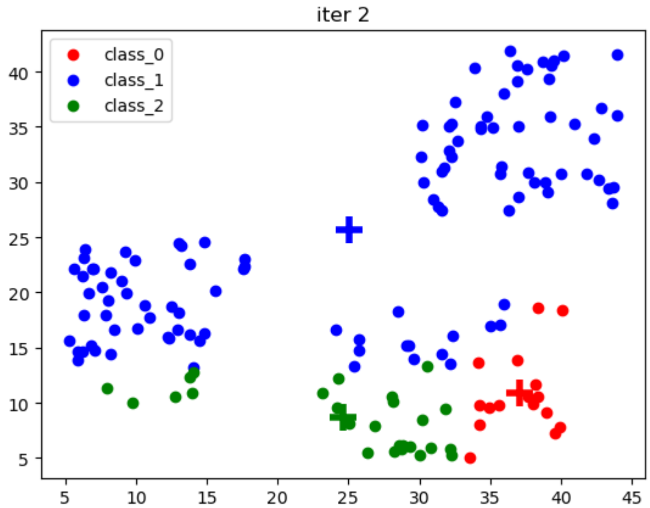

图 1: 迭代中的 K-Means 算法

1.3 K-Means 细节处理

这些细节后面会有相应的实现, 并做一些简单的测试.

1.3.1 初始向量设置

初始化向量有两种基本方法:

- 随机生成 k 个样本点
- 从数据集中随机抽取 k 个样本点

1.3.2 迭代终止条件

迭代终止条件有两种基本方法:

- 簇中心向量不再发生变化
- 样本分类类别标签不再发生变化

2 度量指标

本实验选用了 DBI 作为度量指标(Davies-Bouldin Index).
首先计算簇内样本平均距离:

$$avg(C) = \frac{1}{|C|}\sum_{i=1}^{|C|} dist(\mu, x_i)$$

再计算簇中心距离:

$$d_{cen}(C_i, C_j) = dist(\mu_i, \mu_j)$$

即可算出 DBI:

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{avg(C_i)+avg(C_j)}{d_{cen}(C_i, C_j)}\right)$$

3 实验结果

以下按照初始向量设置方法和迭代终止条件的组合形成了 4 种实验结果, 由于数据集过于简单, 故取结果的平均值作为依据.

3.1 算法效果

这里展示一个算法运行过程的截图, 以直观地了解这个算法. 图中加号表示的是簇中心.

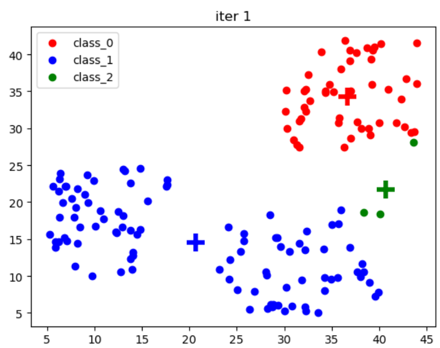

图 2: K-Means 算法运行过程 iter 1

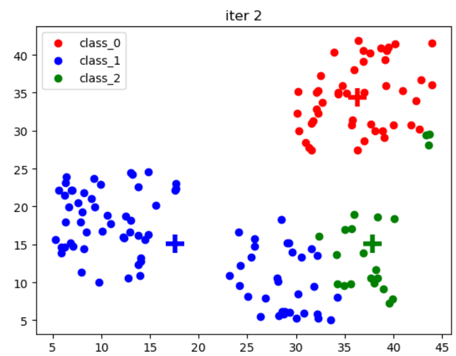

图 3: K-Means 算法运行过程 iter 2

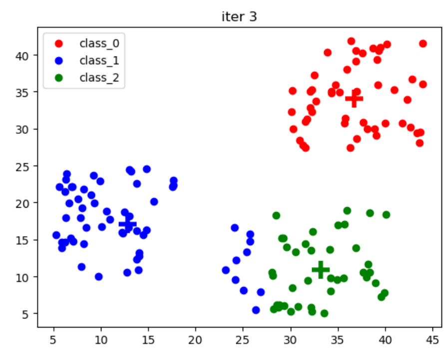

图 4: K-Means 算法运行过程 iter 3

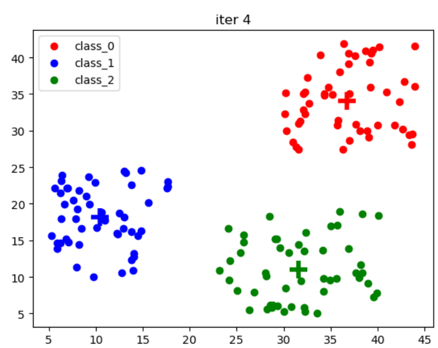

图 5: K-Means 算法运行过程 iter 4

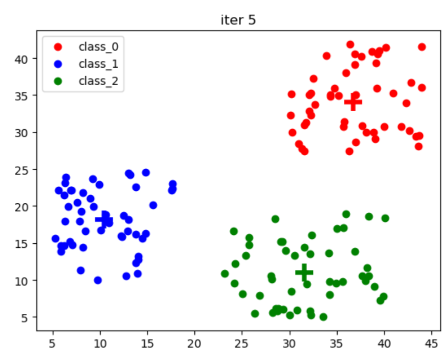

图 6: K-Means 算法运行过程 iter 5

3.2 迭代次数

迭代次数是非常重要的, 因为 k-means 是一个懒惰学习的算法, 每次都要等到数据集来了才做 ”训练”, 所以经过多少次迭代才能收敛是一个很重要的指标.

这里我针对了 4 种情况生成了结果, 如下表:

初始均值 / 迭代终止条件	随机生成	从样本抽样
簇中心向量不变	5.227	4.513
样本分类标签不变	5.517	4.713

表 1: 迭代次数

此处有一些明显的特征:

- 随机生成初始向量的平均迭代次数 比 从样本抽样的平均迭代次数 多了 1 左右. 这是很自然的, 从样本抽样能够保证至少其为样本内的点, 而随机生成可能生成在了很远的地方之类的, 但是经过一次迭代就能马上步入正轨, 因此结果上看起来相差了 1 左右是很合理的.
- 簇中心向量不变来终止 和 样本分类标签不变来终止的结果相差不多. 这主要是因为数据集过于简单, 无法体现其区别.

3.3 簇中心向量

从图中可以直观地看出来, 四种条件下的聚类的结果基本上是一样的:

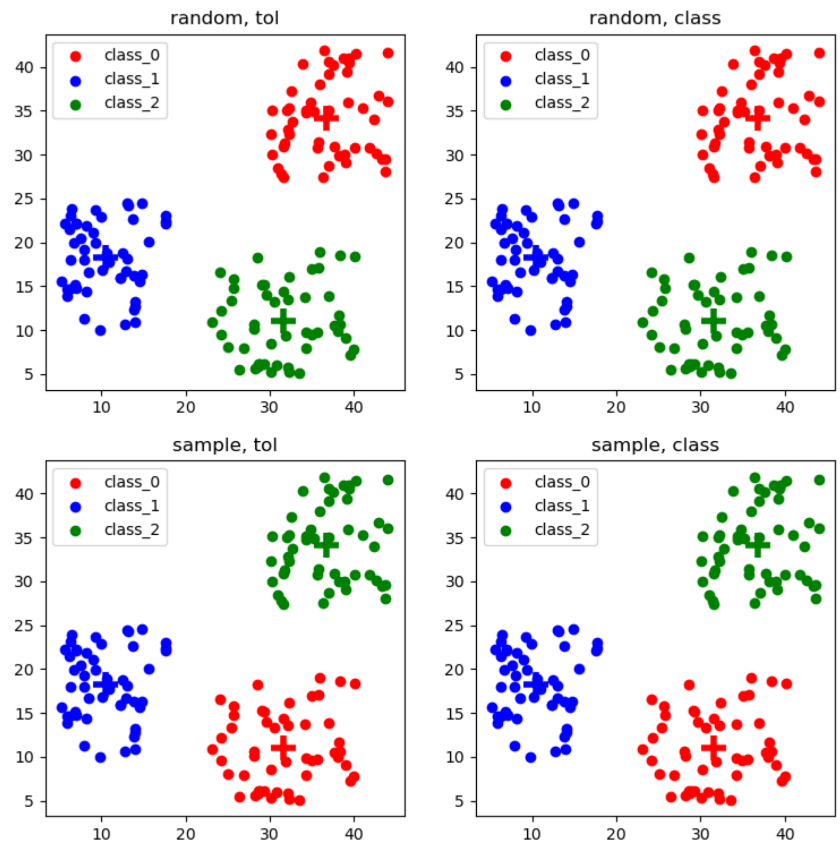

图 7: 四种条件下的聚类结果

而数值上, 其聚类中心如下:

```
====================[random, tol]====================
center: [[36.782  34.1022]
 [10.444  18.2102]
 [31.6182 11.0314]]

====================[random, class]====================
center: [[36.782  34.1022]
 [10.444  18.2102]
 [31.6182 11.0314]]

====================[sample, tol]====================
center: [[31.6182 11.0314]
 [10.444  18.2102]
 [36.782  34.1022]]

====================[sample, class]====================
center: [[31.6182 11.0314]
 [10.444  18.2102]
 [36.782  34.1022]]
```

图 8: 四种条件下的聚类中心

3.4 DBI 值

迭代终止条件 \ 初始均值	随机生成	从样本抽样
簇中心向量不变	0.534	0.520
样本分类标签不变	0.494	0.496

表 2: 迭代次数

同样有一些值得说明的特征:

- DBI 值考虑的是结果, 因此不论初始数据如何生成, 其结果相差并不多
- 簇中心向量不变来终止 比 样本分类标签不变来终止 的记过差一点, 这是很自然的, 当样本分类标签不变的时候, 显然划分得更好一些些, 但二者没有差太多, 这是因为数据集过于简单.

4 实验总结

本次实验实现了 K-Means 算法. 在实验最开始的时候, 其实出现了一些小问题, 例如, 有的类别簇里可能暂时没有任何样本点, 这时候我采取的是重新为该簇生成一个中心. 但整体还是十分顺畅的, 且最终迭代过程的图还是具有一定观赏价值的.