动手实现 Cluster.KMeans

一、数据预处理

1.1 数据集简介

特征信息：https://kdd.ics.uci.edu/databases/kddcup99/kddcup.names

数据集链接：KDD Cup 1999 Data

数据集描述了网络连接情况，含有网络类型的标记。本项目选用 10% 数据集，约 5×10^5 个样本。每个样本有 41 个属性，1 个标签，属于有真实标记的聚类任务。41 个属性共分为 5 类，分别是：

1. normal：normal
2. probe：ipsweep mscan nmap portsweep saint satan
3. dos：apache2 back land mailbomb neptune pod processtable smurf teardrop udpstorm
4. u2r：buffer_overflow httptunnel loadmodule perl ps rootkit sqlattack xterm
5. r2l：ftp_write guess_passwd imap multihop named phf sendmail snmpgetattack snmpguess spy warezclient warezmaster worm xlock xsnoop

我们尝试将所有样本进行聚类。

1.2 数据缺失值处理

数据集完整，无需进行数据缺失值处理。

1.3 离散型数据处理

假设所有属性均为有序属性。采用**闵可夫斯基距离**进行距离计算，因此我们需要知道每一个特征的**序属性**。对于离散型特征，我们使用 `LabelEncoder` 进行编码，编码范围为 $[0, \lambda - 1]$，其中 λ 为当前离散属性的种类数。

二、模型实现细节

2.1 数据类型设计

实现逻辑对标标准包的调用语句。下方是调用 sklearn 包的语句

```
std_model = KMeans(n_clusters=5, n_init=10)
std_model.fit(X)
std_result = std_model.labels_
```

于是我们定义 `fit(X: pd.DataFrame) -> None` 方法进行参数训练，`label_: pd.Series` 字段存储每一个样本的聚类结果标签，`n_clusters: int` 字段存储当前模型的簇数，`n_init: int` 字段存储质心初始化次数，再补充一个 `centroids: pd.DataFrame` 字段存储质心数据。

2.2 算法设计

参考 EM 算法的流程，共四步：

- 初始化质心坐标。即初始化聚类的中心，中心数量取决于 `n_clusters`
- E-step。将所有样本划分到最近的中心，得到 `n_clusters` 个簇
- M-step。计算新的簇中心，更新 `centroids` 字段
- 迭代终点。质心坐标收敛、达到最大迭代次数，满足其一即可。

最终算法的时间复杂度为 $O(mk)$，其中 m 为样本数，k 为簇数。

三、模型测试与评估

3.1 采用外部指标

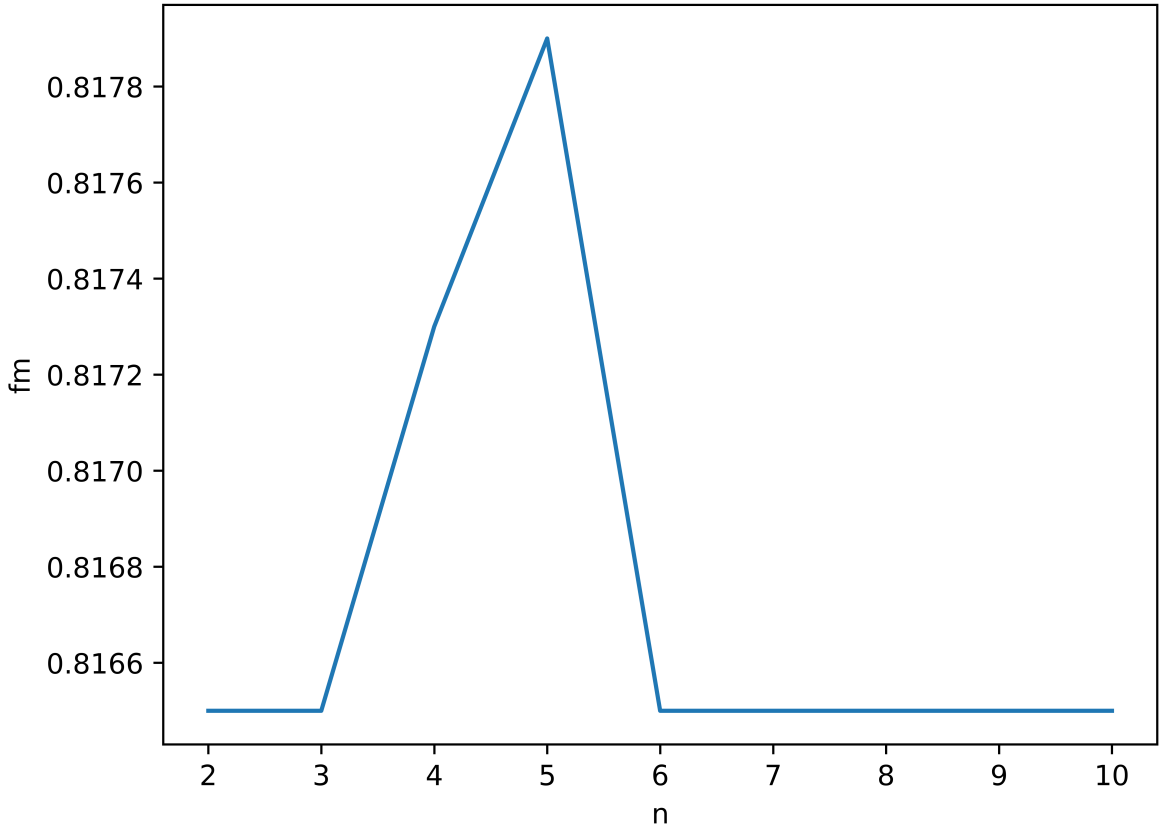
由于每一个样本的真实标记已知，因此我们可以将真实标记看做“参考模型”，考虑两两样本的聚类结果，将当前模型与参考模型的比对结果作为指标，得到以下三种外部指标：

- JC 指数：$JC = \dfrac{a}{a+b+c}$
- FM 指数：$\sqrt{\dfrac{a}{a+b} \cdot \dfrac{a}{a+c}}$
- RI 指数：$\dfrac{2(a+d)}{m(m-1)}$

上述 a, b, c, d 分别对应两样本在两种模型下分类相同、有一个不同、全都不同的样本数量，显然 $a+b+c+d=m(m-1)/2$。三种指标的取值均在 $[0,1]$ 之间，且越大越好。

四、实验结果

4.1 超参数选择



通过簇数的更新，可以得到上图所示的数据。最终我们选择簇数 `n_cluster` 为 5 作为最佳实践，这也与真实数据的簇数相等。

4.2 大数据性能测试

将完整数据集作为聚类数据，即 5×10^6 条样本数据，最终算法执行时间与标准包执行时间差异明显。自定义聚类模型的执行时间约为标准包执行时间的 15 倍。但聚类效果较优，FM 指数为 0.75，而标准包的 FM 指数为 0.64

```
now_model = KMeans(n_clusters=5, n_init=10)
now_model.fit(X)
```

[12] ✓ 21.3s → 标准包执行时间

```
KMeans
KMeans(n_clusters=5, n_init=10)
```

```
now_my_model = MyKMeans(n_clusters=5, n_init=10)
now_my_model.fit(X)
```

[13] ✓ 5m 25.6s → 自定义模型执行时间

```
big_y = LE.fit_transform(fulldata['outcome']).astype(np.int32)
fm_std = fowlkes_mallows_score(big_y, now_model.labels_)
fm_my = fowlkes_mallows_score(big_y, now_my_model.labels_)

print(f"标准包 FM 指数: {fm_std:.4f}")
print(f"自实现 FM 指数: {fm_my:.4f}")
```

[31] ✓ 2.7s

```
标准包 FM 指数: 0.6449
自实现 FM 指数: 0.7517
```

→ 性能度量

五、参考资料

机器学习 周志华

目前流行和先进的聚类算法有哪些?

聚类分析-kddcup99数据集

基于pyspark的对KDD-99数据集的聚类分析实验

聚类算法之K-means算法

入侵检测之KDDCUP99数据集分析

六、总结与反思