【华子】机器学习024-无监督学习模型的性能评估--轮廓系数 -

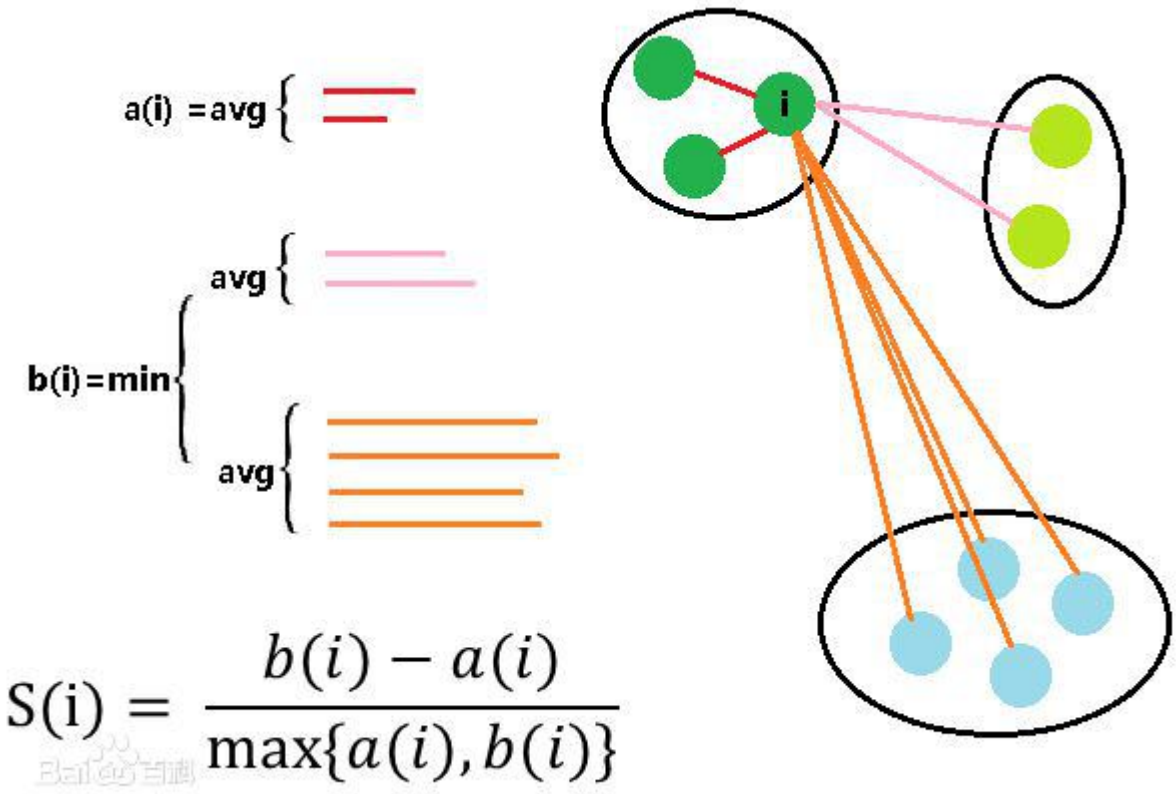
(本文所使用的Python库和版本号: Python 3.6, Numpy 1.14, scikit-learn 0.19, matplotlib 2.2)

前面我们学习过监督学习模型的性能评估，由于数据集有标记，所以我们可以将模型预测值和真实的标记做比较，计算两者之间的差异，从而来评估监督学习模型的好坏。

但是，对于无监督学习模型，由于没有标记数据，我们该怎么样评估一个模型的好坏了？显然，此时我们不能采用和监督学习模型一样的评估方式了，而要另辟蹊径。

1. 度量聚类模型的好坏---轮廓系数

有很多种度量聚类模型的算法，其中一个比较好用的算法就是轮廓系数（Silhouette Coefficient）指标。这个指标度量模型将数据集分类的离散程度，即判断数据集是否分离的合理，判断一个集群中的数据点是不是足够紧密（即内聚度），一个集群中的点和其他集群中的点相隔是否足够远（即分离度），**故而轮廓系数结合了内聚度和分离度这两种因素**，可以用来在相同原始数据的基础上用来评价不同算法，或者算法不同运行方式对聚类结果所产生的影响。



2. 使用轮廓系数评估K-means模型

首先是用pandas加载数据集，查看数据集加载是否正确。

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
np.random.seed(37) # 使得每次运行得到的随机数都一样
# 准备数据集
data_path='./024-data_perf.txt'
df=pd.read_csv(data_path,header=None)
print(df.info()) # 查看数据信息，确保没有错误
dataset=df.values # 数据加载没有问题
```

然后我随机的构建一个K-means模型，用这个模型来训练数据集，并用轮廓系数来评估该模型的优虐，代码如下：

```
from sklearn.cluster import KMeans
# 构建一个聚类模型，此处用K-means算法
model=KMeans(init='k-means++',n_clusters=3,n_init=10)
# 原始K-means算法最开始随机选取数据集中K个点作为聚类中心，
# 分类结果会因为初始点的选取不同而有所区别
# 而K-means++算法改变这种随机选取方法，能显著的改善分类结果的最终误差
# 此处我随机的指定n_cluster=3，看看评估结果
model.fit(dataset)
```

```
# 使用轮廓系数评估模型的优虐
from sklearn.metrics import silhouette_score
# 参数：数据集，聚类标签，欧氏距离，参与评估的样本数
si_score=silhouette_score(dataset,model.labels_,
                          metric='euclidean',sample_size=len(dataset))
print('si_score: {:.4f}'.format(si_score))
```

----------**输**--------**出**----------

si_score: 0.5572

----------**完**----------

参考资料:

1, Python机器学习经典实例，Prateek Joshi著，陶俊杰，陈小莉译