

scikit-learn 实战之非监督学习

一、实验介绍

1.1 实验内容

非监督学习（英语：Unsupervised learning）是机器学习中十分重要的一个分支。这是本实验课程的第 3 章节，将带你了解更多聚类算法，并完成聚类算法对照实验。

1.2 实验知识点

- Mini Batch K-Means 等其他聚类算法
- 聚类算法对比

1.3 实验环境

- python2.7
- Xfce 终端
- ipython 终端

1.4 适合人群

本课程难度为一般，属于初级级别课程，适合具有 Python 基础和线性代数基础，并对机器学习中分类问题感兴趣的用戶。

1.5 代码获取

你可以通过下面命令将代码下载到实验楼环境中，作为参照对比进行学习。

```
$ wget http://labfile.oss.aliyuncs.com/courses/880/k_means_cluster.py
```

二、聚类算法对比

📖 scikit-learn 实战之非监督学习 (/courses/880)

我们已经了解了非监督学习中十分流行的 K-Means 聚类算法。它简单实用，易于实现。其实，Scikit-learn 中还包含有其他聚类算法，每一种算法都有自己的一些特长。

2.1 Mini Batch K-Means

实现方法：sklearn.cluster.MinibatchKMeans

Mini Batch K-Means 整体上和 K-Means 很相似，它是 K-Means 的一个变种形式。与 K-Means 不同的地方在于，其每次从全部数据集中抽样小数据集进行迭代。Mini Batch K-Means 算法在不对聚类效果造成较大影响的前提下，大大缩短了计算时间。

2.2 Affinity Propagation

实现方法：sklearn.cluster.AffinityPropagation

Affinity Propagation 又被称为亲和传播聚类。Affinity Propagation 是基于数据点进行消息传递的理念设计的。与 K-Means 等聚类算法不同的地方在于，亲和传播聚类不需要提前确定聚类的数量，即 K 值。但是运行效率较低。

2.3 Mean Shift

实现方法：sklearn.cluster.MeanShift

MeanShift 又被称为均值漂移聚类。Mean Shift 聚类的目的是找出最密集的区域，同样也是一个迭代过程。在聚类过程中，首先算出初始中心点的偏移均值，将该点移动到此偏移均值，然后以此为新的起始点，继续移动，直到满足最终的条件。Mean Shift 也引入了核函数，用于改善聚类效果。除此之外，Mean Shift 在图像分割，视频跟踪等领域也有较好的应用。

2.4 Spectral Clustering

实现方法：sklearn.cluster.SpectralClustering

Spectral Clustering 又被称为谱聚类。谱聚类同样也是一种比较常见的聚类方法，它是从图论中演化而来的。谱聚类一开始将特征空间中的点用边连接起来。其中，两个点距离越远，那么边所对应的权值越低。同样，距离越近，那么边对应的权值越高。最后，通过对

所有特征点组成的网络进行切分，让切分后的子图互相连接的边权重之和尽可能的低，而各子图内部边组成的权值和尽可能高，从而达到聚类的效果。谱聚类的好处是能够识别任意形状的样本空间，并且可以得到全局最优解。

2.5 Agglomerative Clustering

实现方法：sklearn.cluster.AgglomerativeClustering

Agglomerative Clustering 又被称为层次聚类。层次聚类算法是将所有的样本点自下而上合并组成一棵树的过程，它不再产生单一聚类，而是产生一个聚类层次。层次聚类通过计算各样本数据之间的距离来确定它们的相似性关系，一般情况下，距离越小代表相似度越高。最后，将相似度越高的样本归为一类，依次迭代，直到生成一棵树。由于层次聚类涉及到循环计算，所以时间复杂度比较高，运行速度较慢。

2.6 Birch 聚类

实现方法：sklearn.cluster.Birch

Birch 是英文 Balanced Iterative Reducing and Clustering Using Hierarchies 的简称，它的中文译名为「基于层次方法的平衡迭代规约和聚类」，名字实在太长。

Birch 引入了聚类特征树（CF树），先通过其他的聚类方法将其聚类成小的簇，然后再在簇间采用 CF 树对簇聚类。Birch 的优点是，只需要单次扫描数据集即可完成聚类，运行速度较快，特别适合大数据集。

2.7 DBSCAN

实现方法：sklearn.cluster.DBSCAN

DBSCAN 是英文 Density-based spatial clustering of applications with noise 的简称，它的中文译名为「基于空间密度与噪声应用的聚类方法」，名字同样很长。

DBSCAN 基于密度概念，要求聚类空间中的一定区域内所包含的样本数目不小于某一给定阈值。算法运行速度快，且能够有效处理特征空间中存在的噪声点。但是对于密度分布不均匀的样本集合，DBSCAN 的表现较差。

二、聚类算法对比

接下来，我们对上面提到的 8 中常见的聚类算法做一个对比。如下图所示，这里选择了一

个空间分布由三个团状图案组成的数据集。

🔗 scikit-learn 实战之非监督学习 (/courses/880)

首先，你需要打开终端，通过下面的链接获取这个数据集。

```
$ wget http://labfile.oss.aliyuncs.com/courses/880/data_blobs.csv
```

由于接下来的对比试验中，需要书写的代码量较大。所以，推荐通过编辑器来编写，以防止直接在终端中书写的格式缩进错误。

首先，打开在线环境桌面上的 gedit 文本编辑器。当然，如果你熟悉 Vim，同样可以打开 Gvim 书写。



第一步，导入本次实验需要的模块

```
from sklearn import cluster # 导入聚类模块
from matplotlib import pyplot as plt # 导入绘图模块
import pandas as pd # 导入数据处理模块
import numpy as np # 导入数值计算模块
```

然后，从 cluster 模块中，导入各聚类方法。如 K-Means 等方法需要提前确定类别数量，也就是 K 值。判断的方法很简单，如果聚类方法中包含 n_clusters= 参数，即代表需要提前指定。这里我们统一确定 $K = 3$ 。

对聚类方法依次命名
 cluster_names = ['KMeans', 'MiniBatchKMeans', 'AffinityPropagation', 'MeanShift', 'SpectralClustering', 'AgglomerativeClustering', 'Birch', 'DBSCAN']

确定聚类方法相应参数
 cluster_estimators = [
 cluster.KMeans(n_clusters=3),
 cluster.MinibatchKMeans(n_clusters=3),
 cluster.AffinityPropagation(),
 cluster.MeanShift(),
 cluster.SpectralClustering(n_clusters=3),
 cluster.AgglomerativeClustering(n_clusters=3),
 cluster.Birch(n_clusters=3),
 cluster.DBSCAN()
]

接下来，读取数据开始绘图。

```
# 读取数据集 csv 文件
data = pd.read_csv("data_blobs.csv", header=0)
X = data[['x', 'y']]
Y = data['class']

plot_num = 1 # 为绘制子图准备

# 不同的聚类方法依次运行
for name, algorithm in zip(cluster_names, cluster_estimators):

    algorithm.fit(X) # 聚类


    # 判断方法中是否由 labels_ 参数，并执行不同的命令
    if hasattr(algorithm, 'labels_'):
        algorithm.labels_.astype(np.int)
    else:
        algorithm.predict(X)

    # 绘制子图
    plt.subplot(2, len(cluster_estimators) / 2, plot_num)
    plt.scatter(data['x'], data['y'], c=algorithm.labels_)

    # 判断方法中是否由 cluster_centers_ 参数，并执行不同的命令
    if hasattr(algorithm, 'cluster_centers_'):
        centers = algorithm.cluster_centers_
        plt.scatter(centers[:, 0], centers[:, 1], marker="p", edgecolors="red")

    # 绘制图标题
    plt.title(name)
    plot_num += 1

plt.show() # 显示图
```

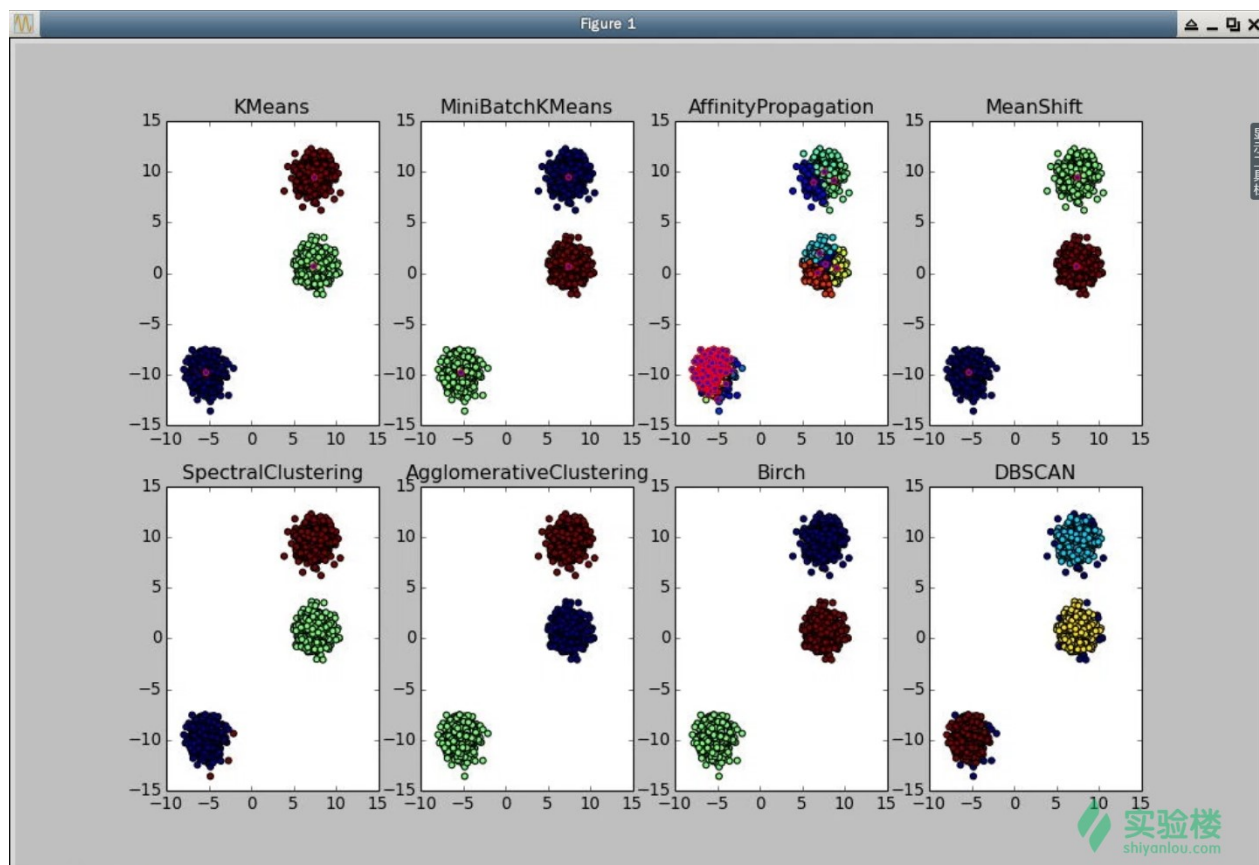
接下来，点击编辑器上方的保存按钮，重命名文件（任意名并以 .py 结尾），选择默认  scikit-learn 实战之非监督学习 (/courses/880) 目录即可。



然后，从桌面上打开终端，执行该文件。



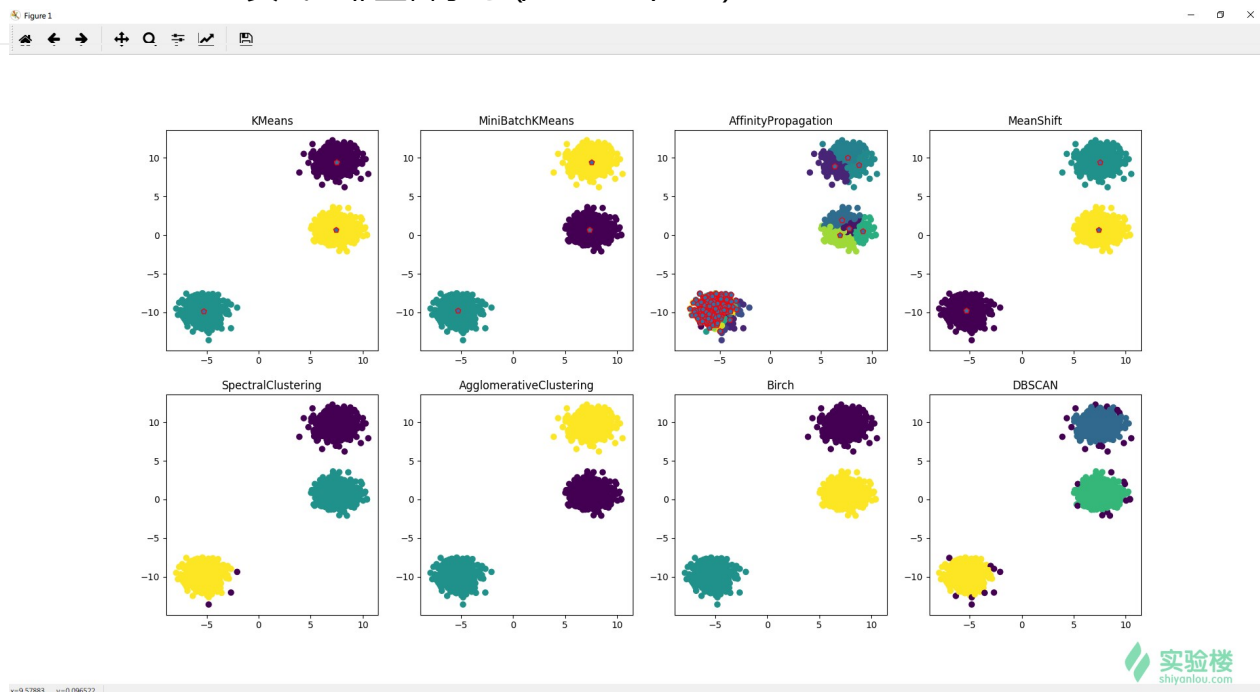
全部输出图像如下方所示。



由于 8 张子图一起显示，所以在实验楼的在线环境中看起来不是特别清楚。下面这一张是

线下环境测试时，绘制的高清大图。

🔗 scikit-learn 实战之非监督学习 (/courses/880)

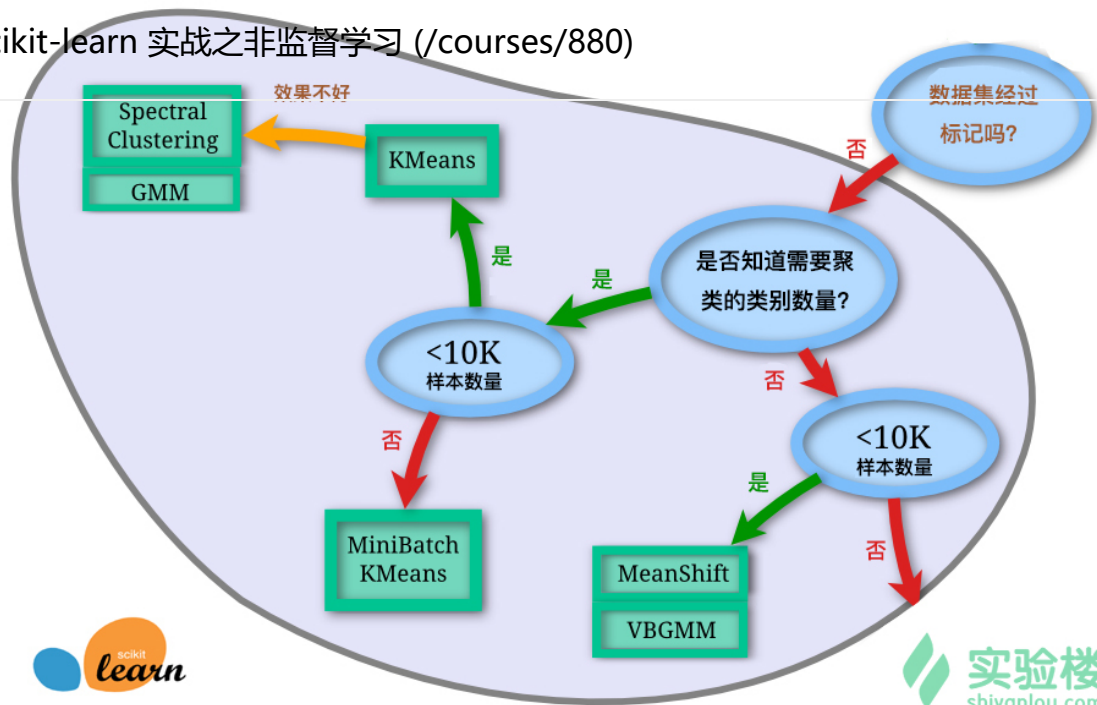


在我们指定 `n_clusters=3` 的方法中，除了 `SpectralClustering` 出现了三个特征点飘逸，其他几种方法的结果几乎是一致的。

除此之外，在没有指定 `n_clusters` 的聚类方法中，`Mean Shift` 对于此数据集的适应性较好。而亲和传播聚类方法在默认参数下，竟然确定出了几十个类别。

三、聚类算法选择

聚类方法这么多，在实际运用中我们该怎样选择呢？其实，`scikit-learn` 提供了一张选择判断图供大家参考。



四、实验总结

本次实验，了解了除 K-Means 之外的其他几种聚类方法。并通过对比实验，测试了不同方法对于同一数据集的聚类效果。由于测试集相对简单，所以这里的目的并不是对比各类方法的好坏。每一类方法都有自己的优点和缺点，它们针对不同测试集的适应性也会不一样，不能单纯地用好或不好来以偏概全。实验的目的是了解和学会简单使用这些方法，如果要想进一步将机器学习方法用于实践，还需要深入了解背后的数学理论才是关键。

五、课后习题

1. 通过调整 `n_clusters` 参数值，尝试将数据集聚为其他不同数量的类别，并查看不同方法的效果。
2. 深入了解各方法的数学原理，并尝试学习调整 scikit-learn 方法里包含的其他参数。

*本课程内容，由作者授权实验楼发布，未经允许，禁止转载、下载及非法传播。

上一节：K 值选择与聚类评估 (/courses/880/labs/3189/document)

下一节：主成分分析（PCA 降维） (/courses/880/labs/3191/document)