**Kmedian算法**

基于python实现Kmedian算法对鸢尾花数据进行聚类

**源代码：**

import pandas as pd

import matplotlib.pyplot as plt

import KMedians

import numpy as np

Iris\_types = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'] # 花类型

Iris\_data = pd.read\_csv('dataSet/Iris.csv')

x\_axis = 'PetalLengthCm' # 花瓣长度

y\_axis = 'PetalWidthCm' # 花瓣宽度

# x\_axis = 'SepalLengthCm' # 花萼长度

# y\_axis = 'SepalWidthCm' # 花萼宽度

examples\_num = Iris\_data.shape[0] # 样本数量

train\_data = Iris\_data[[x\_axis, y\_axis]].values.reshape(examples\_num, 2) # 整理数据

# 归一化

min\_vals = train\_data.min(0)

max\_vals = train\_data.max(0)

ranges = max\_vals - min\_vals

normal\_data = np.zeros(np.shape(train\_data))

nums = train\_data.shape[0]

normal\_data = train\_data - np.tile(min\_vals, (nums, 1))

normal\_data = normal\_data / np.tile(ranges, (nums, 1))

# 训练参数

k = 3 # 簇数

max\_iterations = 50 # 最大迭代次数

centroids, cluster = KMedians.kmedians(normal\_data, k, max\_iterations)

plt.figure(figsize=(12, 5), dpi=80)

# 第一幅图是已知标签或全部数据

plt.subplot(1, 2, 1)

for Iris\_type in Iris\_types:

plt.scatter(Iris\_data[x\_axis], Iris\_data[y\_axis], c='black')

plt.title('raw')

# 第二幅图是聚类结果

plt.subplot(1, 2, 2)

for centroid\_id, centroid in enumerate(centroids): # 非聚类中心

current\_examples\_index = (cluster == centroid\_id).flatten()

plt.scatter(normal\_data[current\_examples\_index, 0], normal\_data[current\_examples\_index, 1])

for centroid\_id, centroid in enumerate(centroids): # 聚类中心

plt.scatter(centroid[0], centroid[1], c='red', marker='x')

plt.title('label kemans')

plt.show()

**运行（测试）过程及结果：**

