K-means实验

PB18071477 敖旭扬

原理

给定数据集 $D=\{X_1,\dots,X_m\}$ 和 K，`K-means`（K均值）算法针对聚类所得簇划分 $C=\{C_1,C_2,\dots,C_K\}$ 最小化平方误差

$$E=\sum_{k=1}^{K}\sum_{\boldsymbol{x}\in C_k}||\boldsymbol{x}-\boldsymbol{\mu}_k||_2^2 \tag{1}$$

将数据集D划分成 K 个簇，其中

$$\boldsymbol{\mu}_k=\frac{1}{|C_k|}\sum_{\boldsymbol{x}\in C_k}\boldsymbol{x} \tag{2}$$

在本实现中，$\boldsymbol{\mu}$ 的初始值是从数据集合 D 中随机抽取 K 个样本点得到。之后把每个样本点归类为欧氏距离最近的 $\boldsymbol{\mu}_k$ 所在的那个簇，再重新计算 $\boldsymbol{\mu}$，重复迭代直到样本分类类别标签不再发生变化时结束训练。

性能度量使用**DBI**，定义为

$$DBI=\frac{1}{K}\sum_{i=1}^{K}\max_{j\neq i}\left(\frac{avg(C_i)+avg(C_j)}{d_{cen}(C_i,C_j)}\right) \tag{3}$$

其中

$$avg(C_k)=\frac{1}{|C_k|}\sum_{\boldsymbol{x}\in C_k}dist(\boldsymbol{\mu}_k,\boldsymbol{x}) \tag{4}$$

为簇内样本平均距离

$$d_{cen}(C_i,C_j)=dist(\boldsymbol{\mu}_i,\boldsymbol{\mu}_j) \tag{5}$$

为簇中心点距离

一般`DBI`越小，聚类划分效果越好。

编程实现

矩阵运算使用`python`的`numpy`库实现。

`K-means`算法如下：

```python
class KMeans:
    def fit(self, raw_X, show=False, normalize=True):
        X = copy.deepcopy(raw_X)
        m = X.shape[0]  # 样本数
        d = X.shape[1]  # 维度

        # 归一化数据
        if normalize:
            for i in range(d):
```

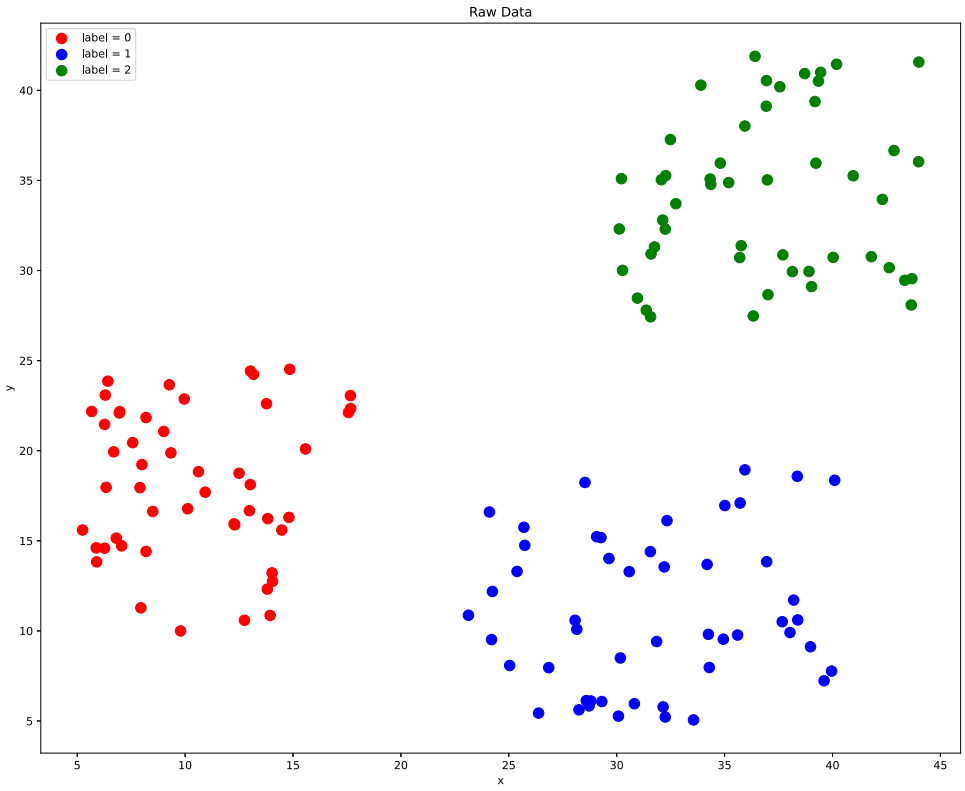
```
                max_Xi = np.max(np.abs(X[:, i]))
                X[:, i] = X[:, i]/max_Xi

        # 初始向量设置
        sample = random.sample(range(m), self.k)
        self.mu = copy.deepcopy(X[sample])
        raw_mu = copy.deepcopy(raw_X[sample])

        # 迭代优化
        self.saved = []
        self.label = np.zeros(m, dtype=int)
        while True:
            old_label = copy.deepcopy(self.label)
            for i in range(m):
                self.label[i] = np.argmin(self.dist(X[i], self.mu, 2))
            if show:
                label = copy.deepcopy(self.label)
                self.saved.append((label, raw_mu))
                raw_mu = self.calculate_mu(raw_X, self.label)
            self.mu = self.calculate_mu(X, self.label)
            if (old_label == self.label).all():
                # 样本分类类别标签不再发生变化时终止迭代
                break
        self.mu = self.calculate_mu(raw_X, self.label)
        if show:
            self.show_dynamic(raw_X, self.saved)
        return self.label, self.mu
```

完整实验源码见压缩包中的KMeans.py。

运算结果

实例

本次实验使用的数据分布如下：



原始数据和理想结果

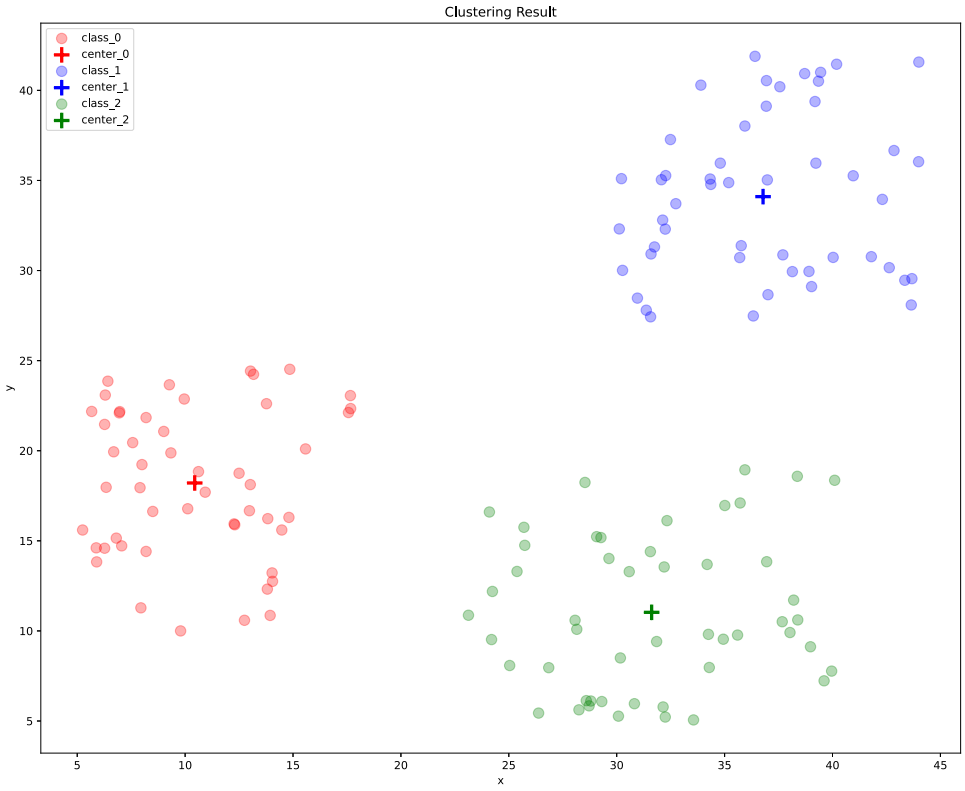
在主函数中调用下面的实例

```
# 训练模型
kmeans = KMeans(k=3)
label, mu = kmeans.fit(X, show=True)
```

命令行输出结果为

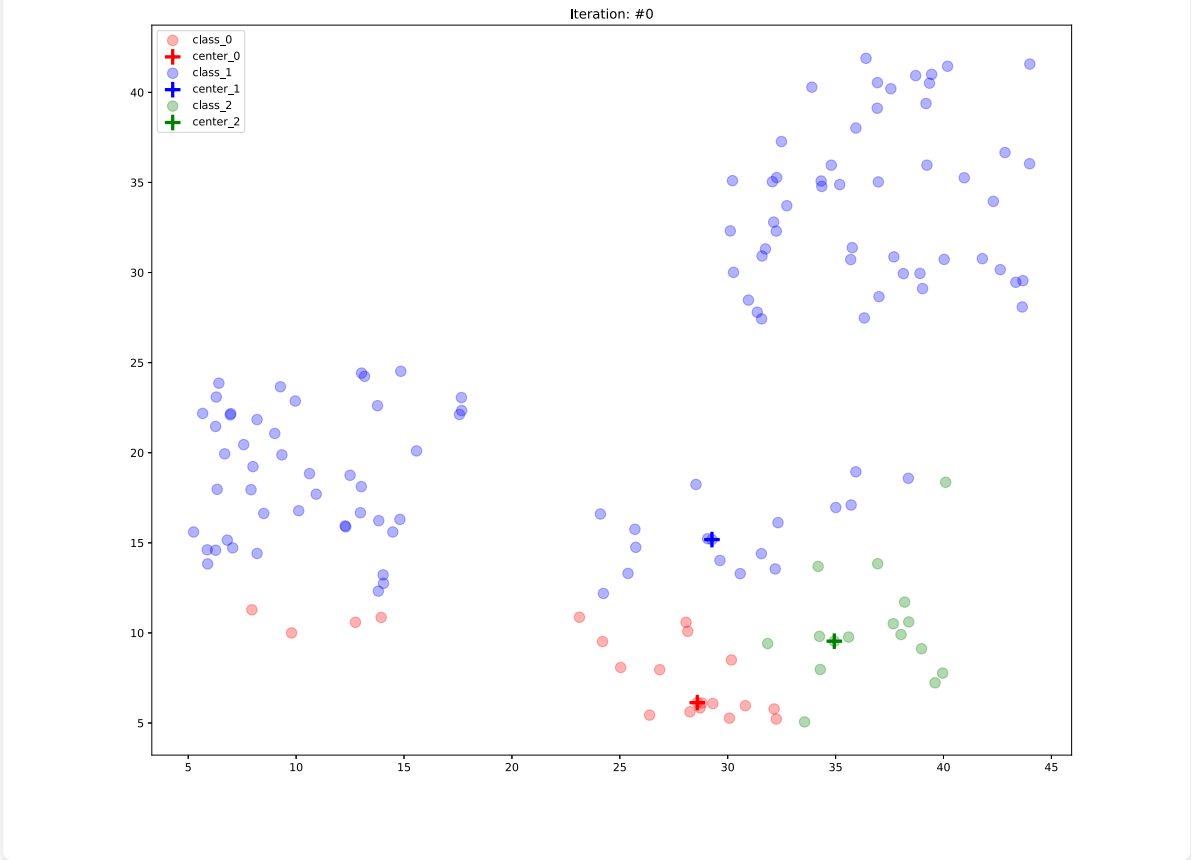
```
簇中心坐标：
 [[10.444  18.2102]
 [36.782  34.1022]
 [31.6182 11.0314]]
DBI = 0.4936
```

训练结果

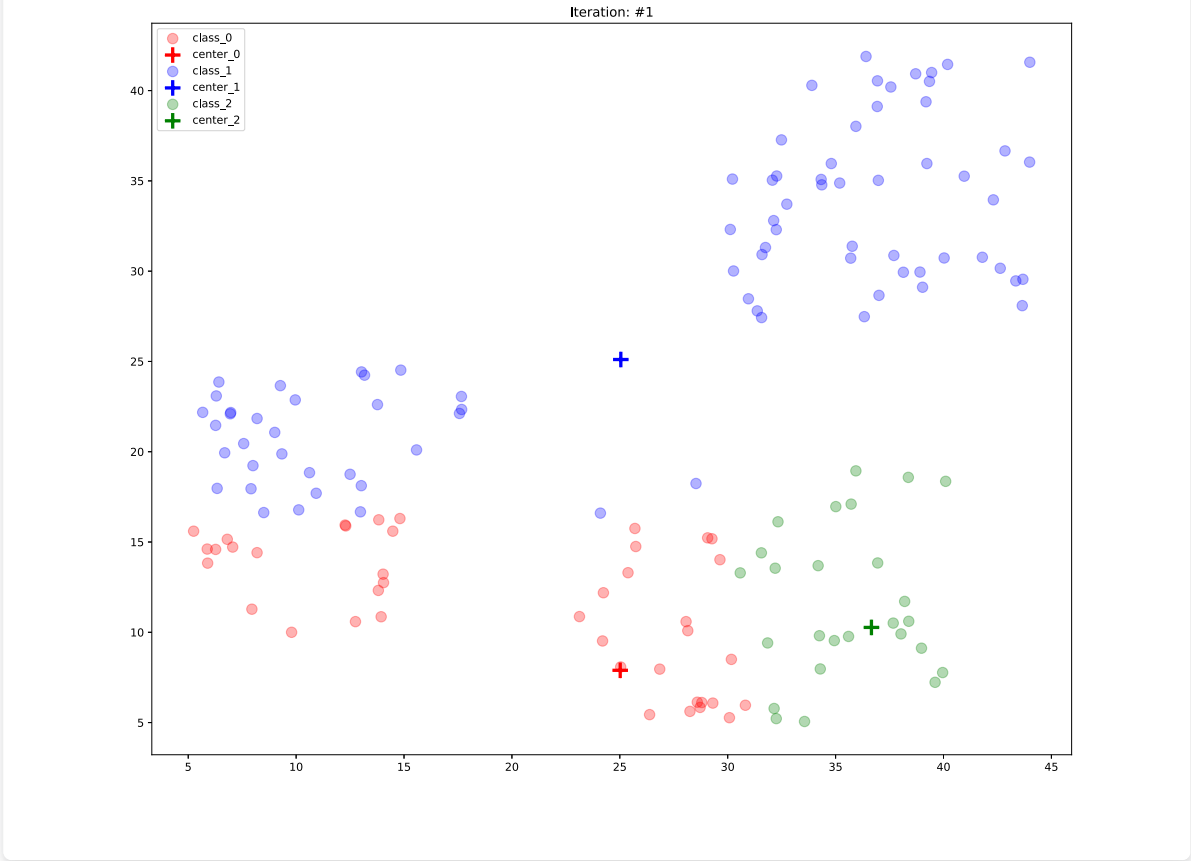


聚类结果

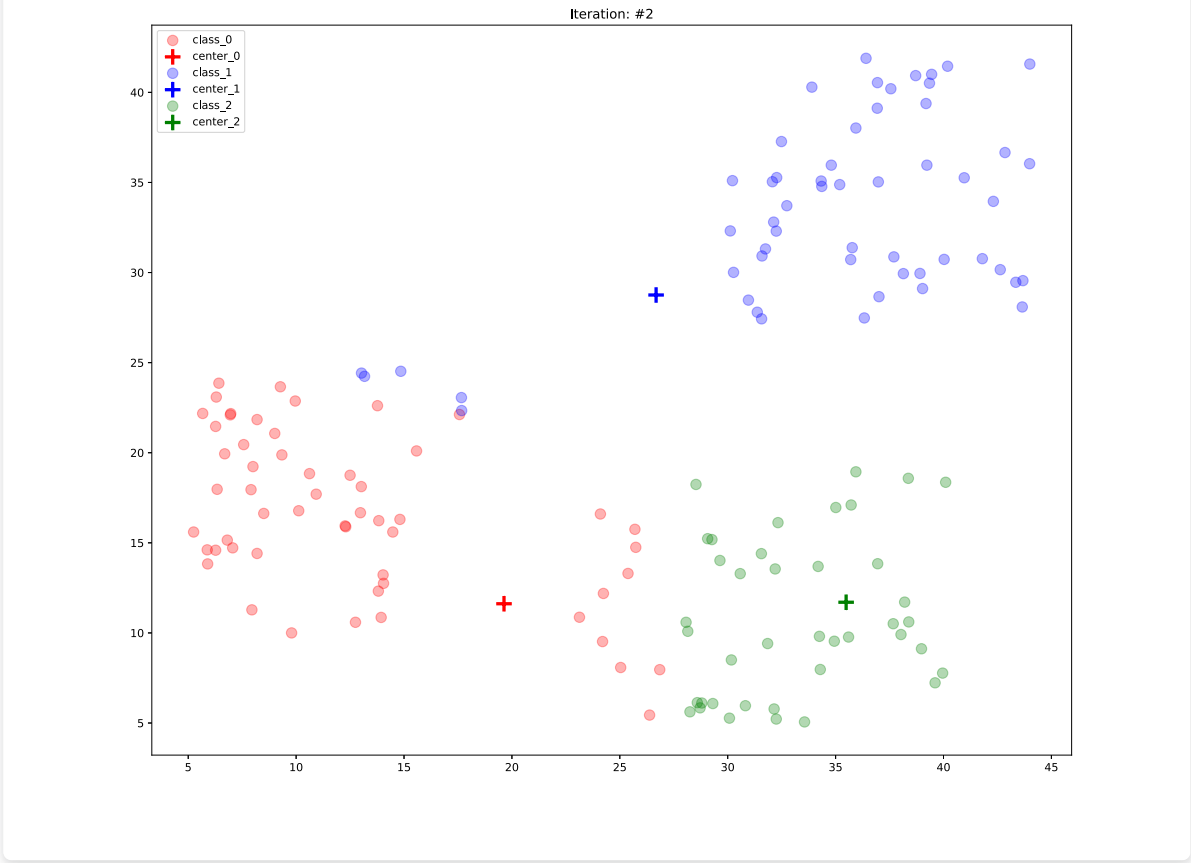
训练迭代过程（某一次的结果）



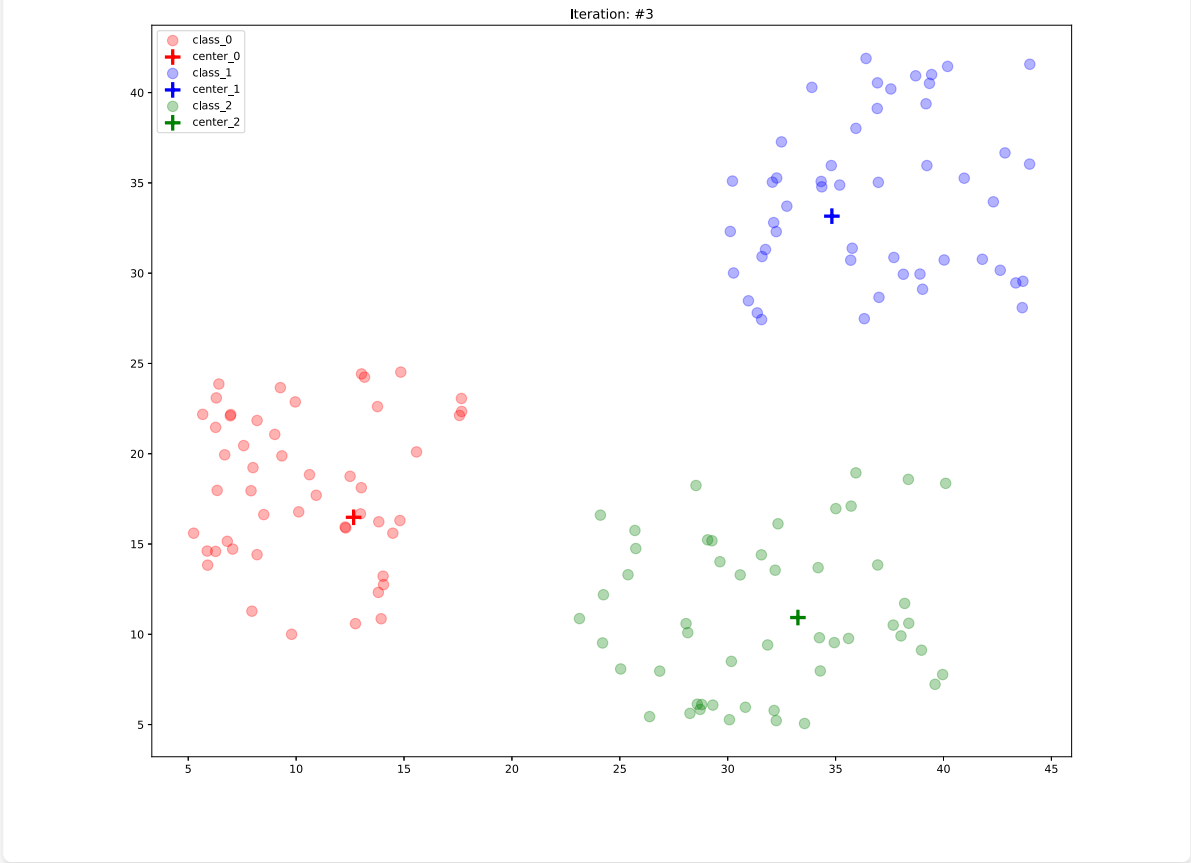
第0次迭代后



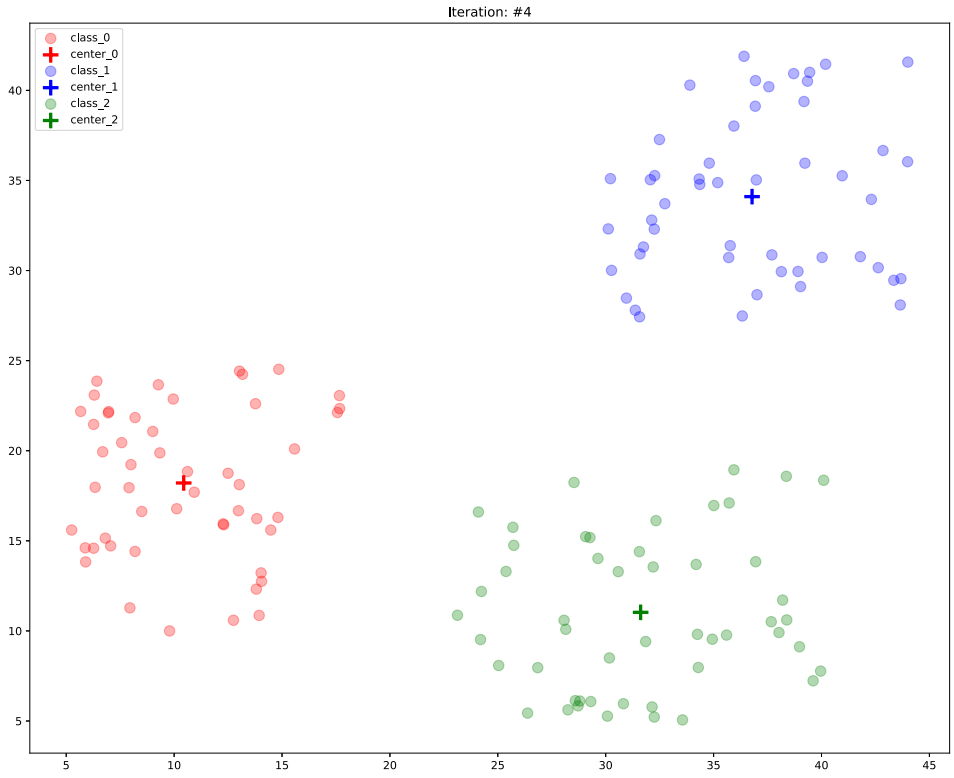
第1次迭代后



第2次迭代后



第3次迭代后



第4次迭代后，样本分类类别标签不变

总结

实验要求的 Baseline 为

```
1）输出聚类后簇中心点坐标
2）DBI值小于5
```

我训练出的结果中，聚类后簇中心点坐标为 $(10.444, 18.2102), (36.782, 34.1022), (31.6182, 11.0314)$ ，DBI值为 0.4936 ，效果良好。