实验三 聚类分析

一、实验目的

掌握聚类分析的基本思想。通过对聚类方法的学习，掌握距离聚类的概念，掌握相似性测度和聚类准则，掌握最近邻聚类方法，最大最小距离聚类方法，系统聚类方法，K 均值算法，迭代自组织算法，聚类结果的评价。

二、实验内容

1. 数据集采用 Iris 鸢尾花数据，使用聚类算法实现两两分类、三类一起分类；
2. 采用算法：K 均值、ISODATA 算法；并讨论不同的初始类中心选择方法对聚类结果的影响；

三、实验原理

1. K-means 聚类算法

（1）算法思想

K-means 算法根据输入参数 k，将数据集划分成 k 个簇。算法采用迭代更新的方法：在每一轮中，依据 k 个参照点将其周围的点分别组成 k 个簇，而每个簇的质心被视为下一轮迭代的参照点。迭代使得选取的参照点越来越近真实的簇质心，所以聚类效果越来越好。

K-means 聚类算法的原理是：首先随机选取 k 个点作为初始聚类中心，然后计算各个样本到聚类中心的距离，把样本分到离它最近的那个聚类中心所在的簇；对调整后的新簇计算新的聚类中心，如果相邻两次的聚类中心没有任何变化，说明样本调整结束，这时某个误差平方和函数已经达到最小，聚类准则函数已经收敛，算法结束。

该算法的结果受到所选取的聚类中心的数目和其初始位置，以及模式样本的几何性质及读入次序等的影响。在实际应用中需要试探不同的 K 值和选择不同的聚类中心起始值。如果模式样本形成几个相距较远的小块孤立的区域分布，一般都能得到收敛结果。

（2）算法步骤

将 d 维数据集$\mathrm{X}=\{x_j|x_j\epsilon R^d, i=1,2,\dots,N\}$聚集成 k 个簇 W_1，W_2，……，W_k，它们的质心依次为 c_1，c_2，……，c_k，其中 $c_i=\frac{1}{n}\sum_{x\in w_i}x$，$n_i$ 是簇 Wi 中数据点的个数。聚类结果的好坏用目标函数 J 表示：

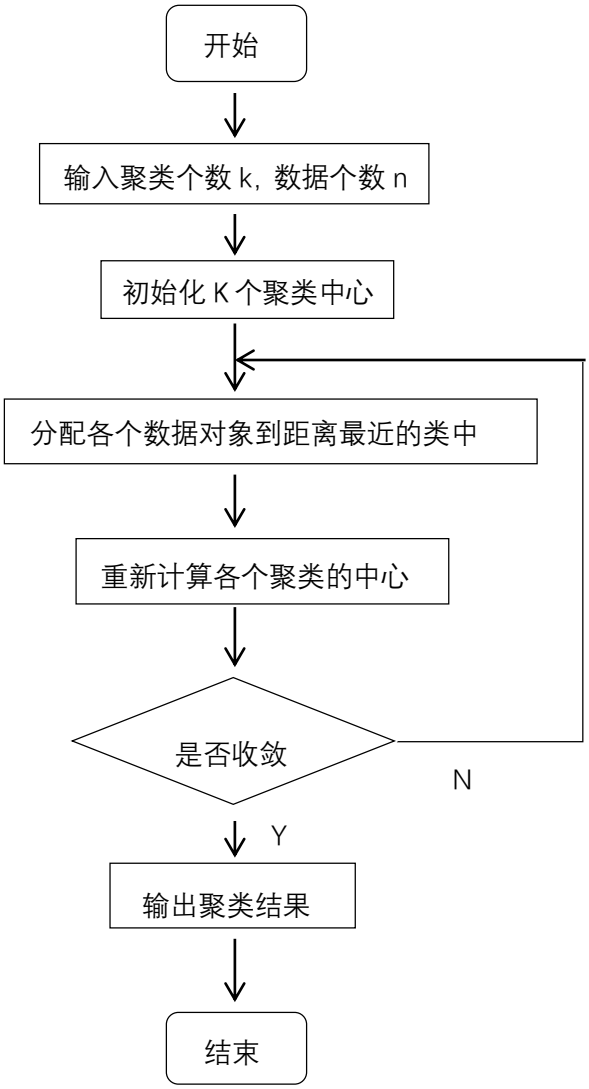
$$J=\sum_{i=1}^{k}\sum_{j=1}^{n_j}d_{ij}(x_j,c_i)$$

式中，$d_{ij}(x_j, c_i)$是 x_j 与 c_i 之间的欧氏距离。目标函数 J 其实是每个数据点与所在簇的质心的距离之和，所以 J 值越小，簇就越紧凑、越相对独立。因此，算法通过不断优化 J 的取值来寻求好的聚类方案，当 J 取极小值时，对应的聚类方法即为最优方案。

K-means 算法步骤如下：

（a）从 X 中随机选择 k 个初始参照 c_1，c_2，……，c_k。
（b）在第 n 次迭代中，对任意一个样本，求其到 k 个中心的距离，将该样本归到距离最短的中心所在的类。
（c）利用均值等方法更新该类的中心值。
（d）对于所有的 k 个聚类中心，如果利用（2）（3）的迭代法更新后，值保持不变（目标函数收敛），则迭代结束，否则继续迭代。
（e）输出聚类结果。

K-means 算法的流程如下图所示：



2. ISODATA 聚类算法

(1) 算法思想

ISODATA 算法是一种聚类划分算法，称为迭代自组织数据分析或动态聚类。通过设定初始参数而引入人机对话环节，并使用归并与分裂的机制，当某两类聚类中心距离小于某一阈值时，将它们合并为一类，当某类标准差大于某一阈值或其样本数目超过某一阈值时，将其分为两类。在某类样本数目少于某阈值时，需将其取消。如此，根据初始聚类中心和设定的类别数目等参数迭代，最终得到一个比较理想的分类结果。

(2) 算法步骤

ISODATA 算法步骤如下:

(a) 设置聚类分析控制参数;

(b) 将准备分类的样本值读入;

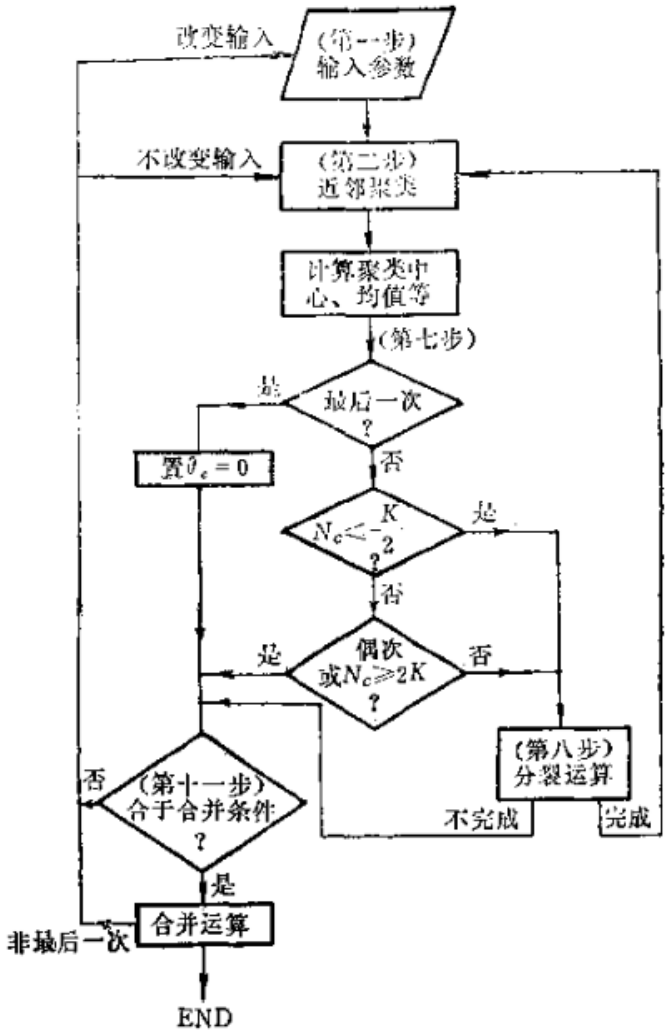
(c) 初始化分类，按照与聚类中心距离最小的原则将各样本分类;

(d) 类分裂，如果在同一类中样本分布太过密集或者类的数目太过少，这说明在这一空间上一定还存在不止一个的集群中心，从而需要将该类进行分裂操作，具体来说，就是设置类内各样本分布标准差上限，如果同类中样本距离超过此限度将被分裂,否则保留，然后再次转到第二步;

(e) 类合并，如果两类相隔太近，说明这两类中的样本分类的必要性不充分，根据一定条件将其合并，具体来说，就是设置类与类之间的距离下限，如果低于此下限则合并两类，或者是某一类中的样本数目过于少而不足以成为一类时，也可以考虑将该类合并到其他类中去，然后再次转到第二步;

(f) 如此往复的进行分类、判断、分裂或合并操作，如果达到了预计的分类效果，或者操作次数已经达到一定数目，则完成算法。

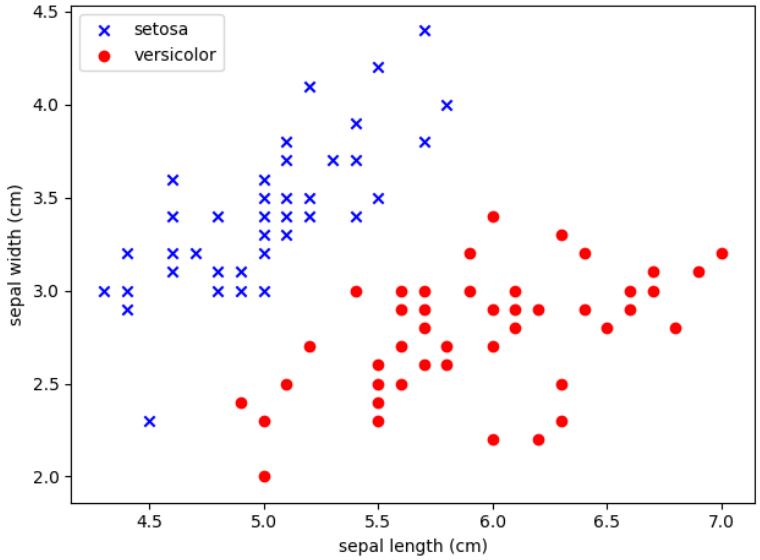
ISODATA 算法示意图如下:



四、实验过程

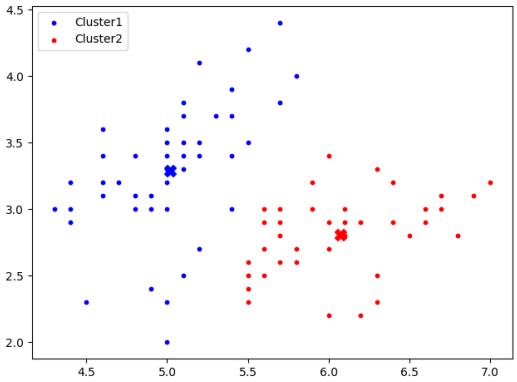
1. k-means 算法实验

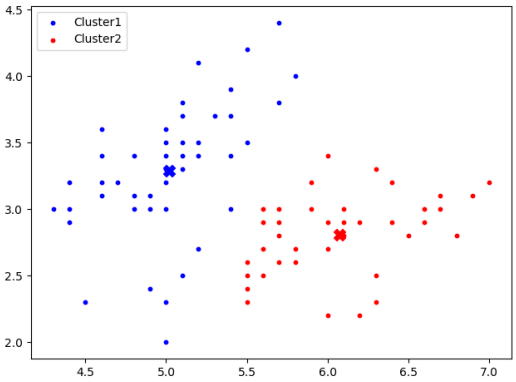
（1）选取“setosa”和“versicolor”两类进行聚类，并选取“sepal length (cm)”和 “sepal width (cm)”这两个特征。用 matplotlib 库画出这两类数据：



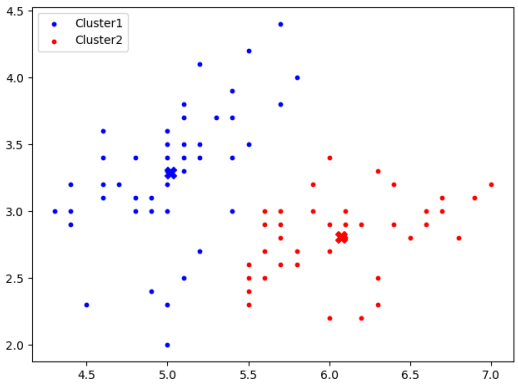
（a）使用 K-means 算法对该数据进行聚类，设置 k 值为 2，随机选取初始聚类中心，实验结果如下：

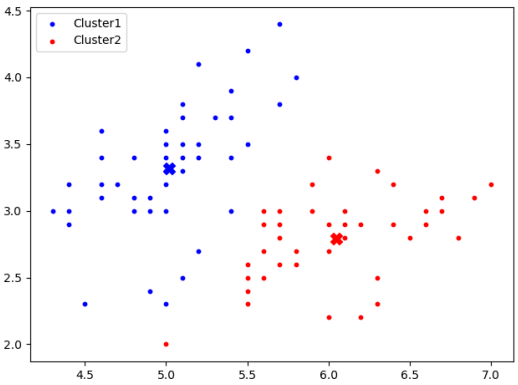
初始聚类中心：[6.743, 3.36], [4.51, 3.882]　　初始聚类中心：[5.012, 2.589], [4.756, 2.523]





初始聚类中心：[4.765, 2.648], [5.414, 3.829]　初始聚类中心：[6.271, 4.041], [6.072, 2.519]



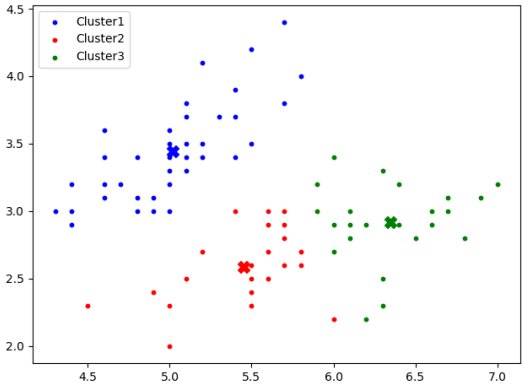


可以看到，在这种数据分布情况下，两类模式样本间距较远（或者说有较孤立的区域分布），在不同的聚类中心初始值情况下，能够得到基本一样的收敛结果，最后的两个聚类中心基本一致。

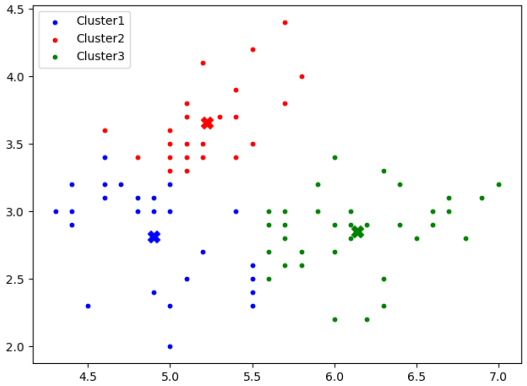
（b）设置 k 值为 3，随机选取初始聚类中心，实验结果如下：

初始聚类中心：

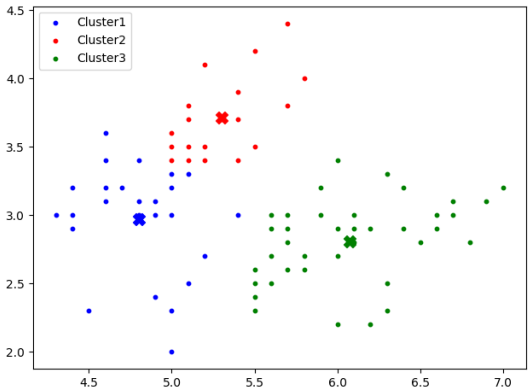
[4.507, 3.314], [6.408, 2.592], [6.8161, 2.600]



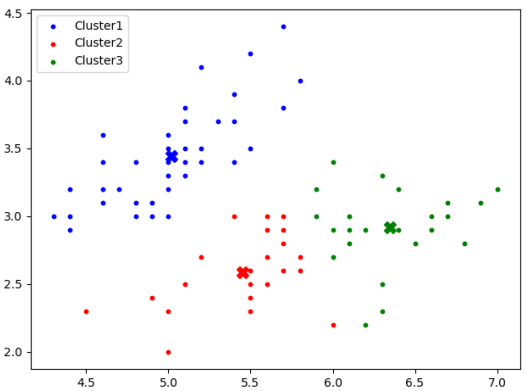
[5.277, 3.352], [6.925, 4.325], [5.576, 3.647]



[4.762, 3.261], [5.068, 2.471], [5.008, 3.162]



[5.42, 3.382], [6.264, 2.633], [6.575, 2.056]



可以看到，在这种数据分布情况下，由于该数据中的模式样本并未形成相距较远、孤立的区域分布（考虑设置 k 值为 3），所以算法最终的收敛结果受初始聚类中心的影响比较大，容易陷入局部最小值。另外，k-means 算法中 k 值的选择是指定的，不同的 k 得到的结果会有很大的不同，以该数据为例，当 k=3 时，可以看到上面第 4 个图中红色和绿色两个簇应该是可以合并成一个簇的。

(c) 选取“setosa”和“versicolor”两类进行聚类，并选取全部特征作为数据，由于 4 维特征向量的模式不能直观地看清聚类的效果，因此在评价聚类效果时，常用聚类中心之间的距离、诸聚类域中样本数目、诸聚类域内样本的距离方差这 3 个指标来综合考虑。下面是使用 K-means 算法对该数据进行聚类的结果（k=2）：

初始聚类中心：[4.505 2.801 1.48　0.909]，　[5.178 3.706 1.378 1.384]
聚类中心之间的距离：

聚类中心	z_2
z_1	3.205

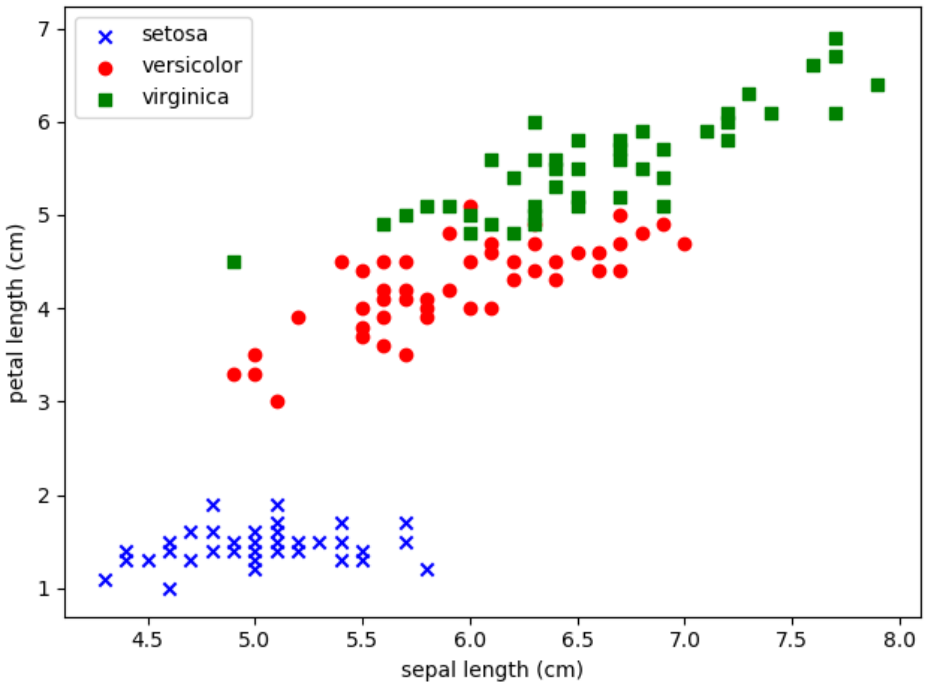
诸聚类域中样本数目：

聚类域	样本数目
S_1	50
S_2	50

诸聚类域内样本的距离方差：

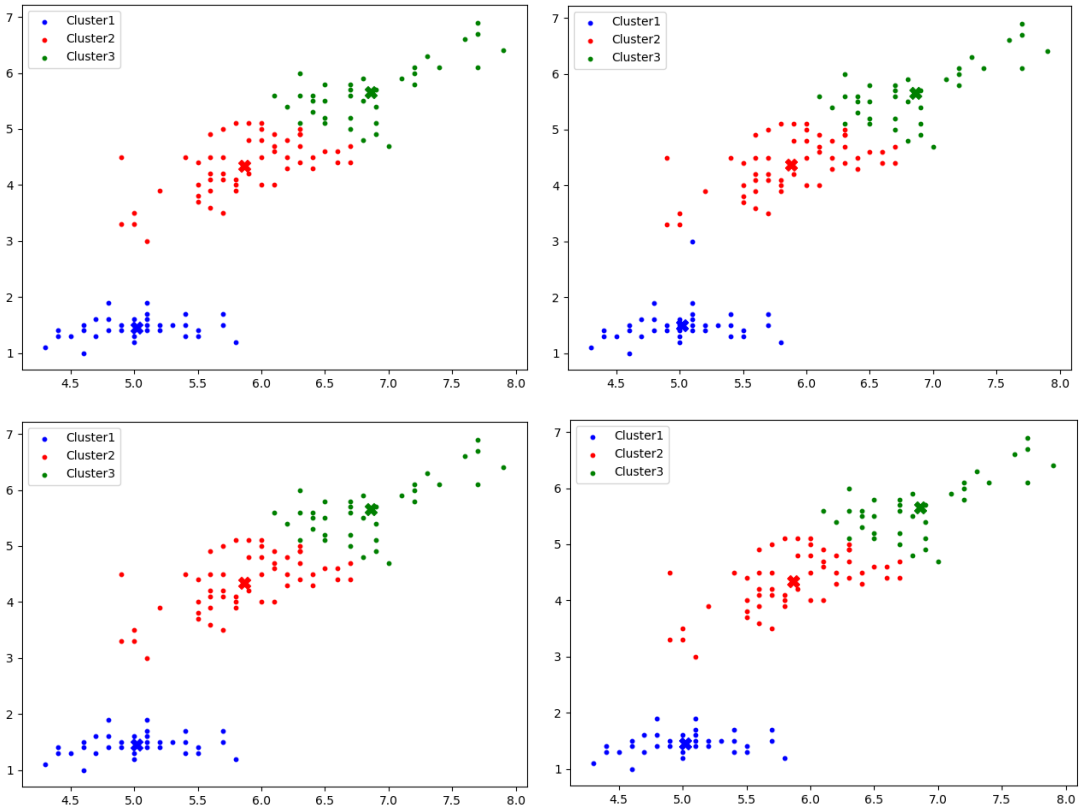
距离标准差 / 聚类域	σ_1	σ_2	σ_3	σ_4
S_1	0.349	0.377	0.172	0.106
S_2	0.511	0.311	0.465	0.196

(2) 选取“setosa”、“versicolor”、”virginica” 三类进行聚类，并选取“'sepal length (cm)'”和 “petal length (cm)”这两个特征。用 matplotlib 库画出这三类数据：



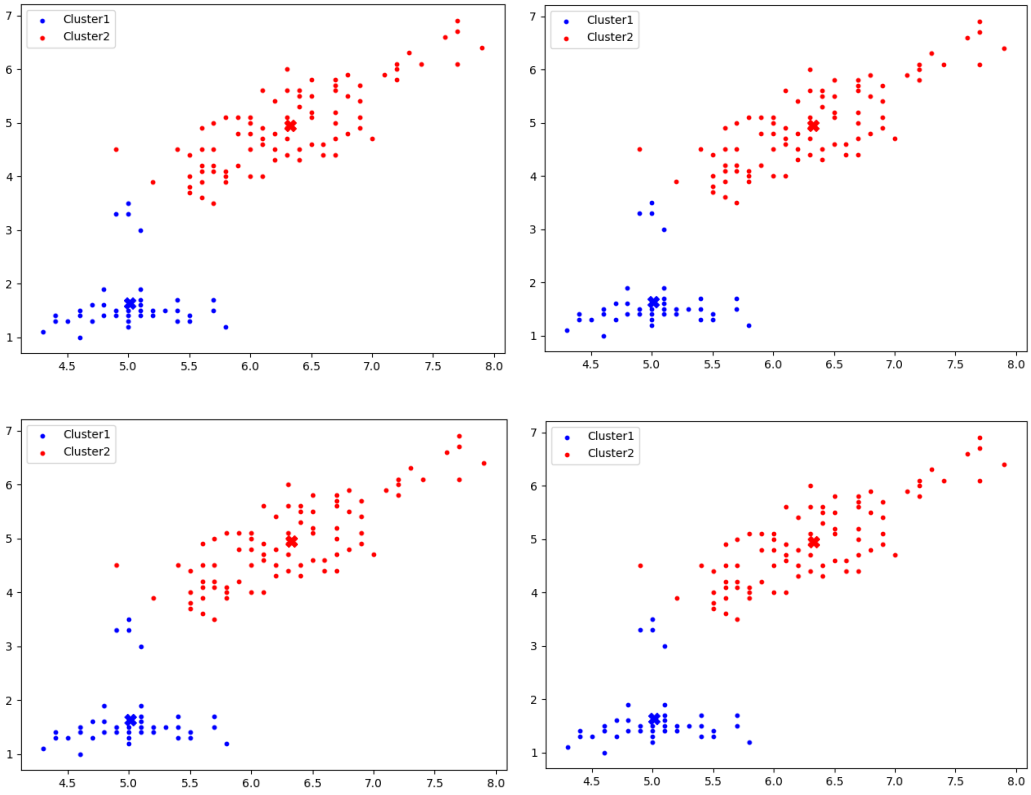
可以看到三类模式样本形成相距较远、孤立的区域分布。

(a) 取 k=3，随机初始化聚类中心，多次实验结果如下：



可以看到，在这种数据分布情况下，算法能够收敛到一个基本一致的聚类结果。此外，当 k 取 2 时，红色和绿色两个簇应该也是可以合并为一个簇的。

(b) 取 k=2，随机初始化聚类中心，多次实验结果如下:



可以看到，4 次聚类的结果完全一致，算法基本能得到一个很好的聚类结果，但是也存在局限性，即无法对非球状的数据分布进行很好地聚类，比如图中 Cluster2 中上面的 4 个蓝色点。

(c) 选取“setosa”、“versicolor”、”virginica”三类进行聚类，并选取全部特征作为数据，由于 4 维特征向量的模式不能直观地看清聚类的效果，因此在评价聚类效果时，常用聚类中心之间的距离、诸聚类域中样本数目、诸聚类域内样本的距离方差这 3 个指标来综合考虑。下面是使用 K-means 算法对该数据进行聚类的结果（k=3）：

初始聚类中心：[4.505 2.801 1.48　0.909]，　[5.178 3.706 1.378 1.384]
聚类中心之间的距离：

聚类中心	z_2	z_3
z_1	4.988	3.346
z_2		1.788

诸聚类域中样本数目：

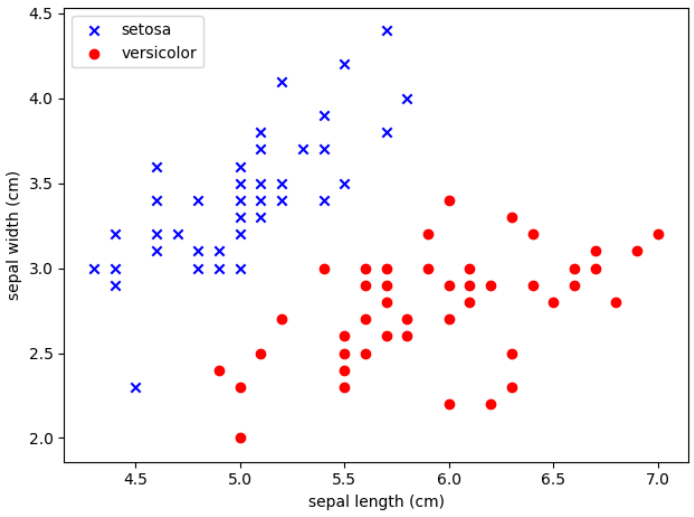
聚类域	样本数目
S_1	50
S_2	39
S_3	61

诸聚类域内样本的距离方差：

距离标准差 / 聚类域	σ_1	σ_2	σ_3	σ_4
S_1	0.349	0.377	0.172	0.106
S_2	0.482	0.283	0.504	0.293
S_3	0.444	0.29	0.507	0.297

2. ISODATA 算法实验

（1）选取“setosa”和“versicolor”两类进行聚类，并选取“sepal length (cm)”和 “sepal width (cm)”这两个特征。用 matplotlib 库画出这两类数据：

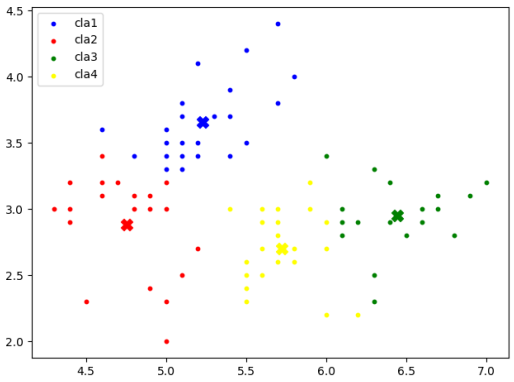


使用 ISODATA 算法对该数据进行聚类：

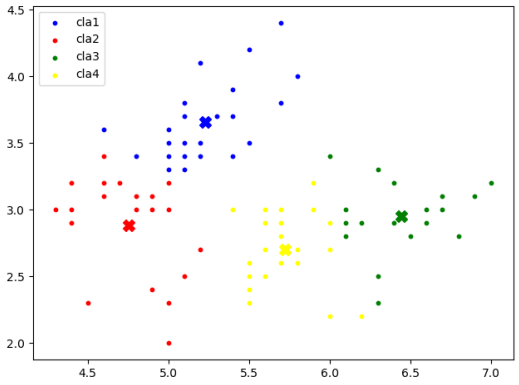
ISODATA 算法的参数有：预期聚类中心数目K、聚类域最少样本数目θ_N、聚类域中样本分布距离标准差θ_S、聚类中心最小距离θ_C、一次迭代可合并对数L、初始聚类中心个数N_C、迭代最大次数I。

(a) 设置θ_N分别为 5、10、20、30，其他参数如下，得到实验结果。

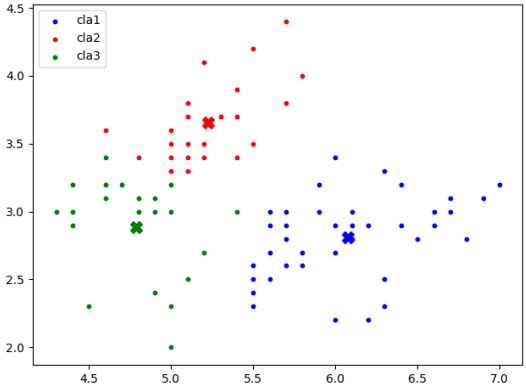
K	θ_S	θ_C	L	N_C	I
3	0.4	0.1	3	1	1000



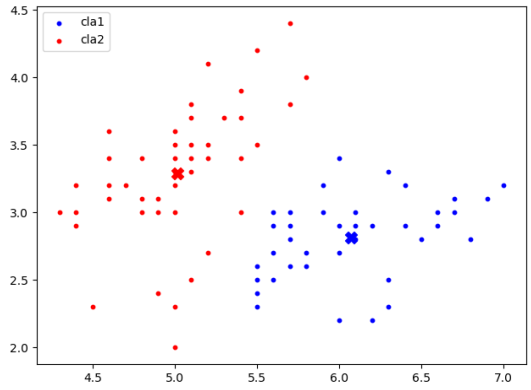
(1) $\theta_N = 5$



(2) $\theta_N = 10$



(3) $\theta_N = 20$

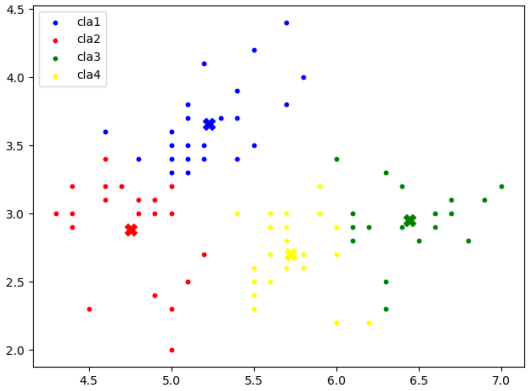


(4) $\theta_N = 20$

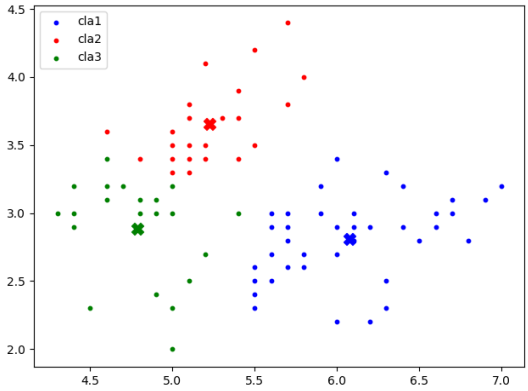
可以看到，当$\theta_N = 5$时，聚类域最少样本数目为 5，ISODATA 算法将该数据聚类为 4 类，随着θ_N的增大，由于增加的聚类域内最少样本数目的限制，聚类数逐渐减少。

(b) 设置θ_S分别为 0.4、0.5、0.6、1，其他参数如下，得到实验结果。

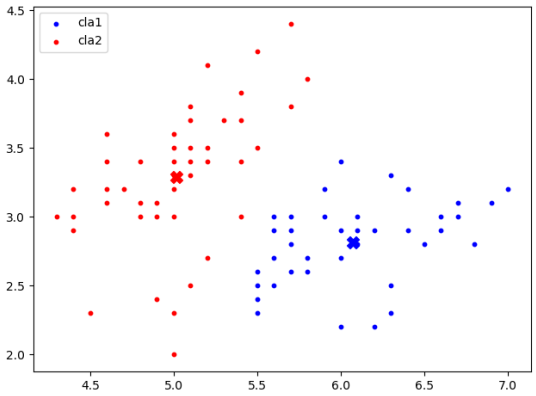
K	θ_N	θ_C	L	N_C	I
3	10	0.1	3	1	1000



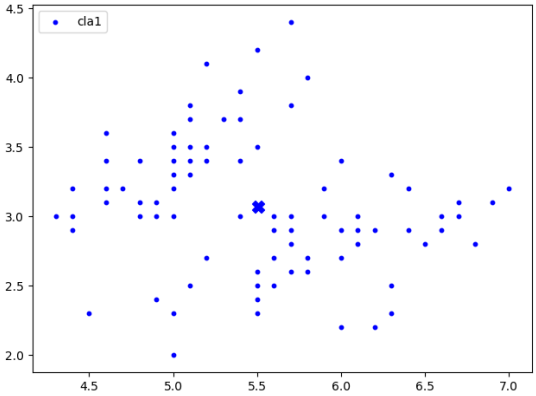
(1) $\theta_S = 0.4$



(2) $\theta_S = 0.5$



(3) $\theta_S = 0.6$

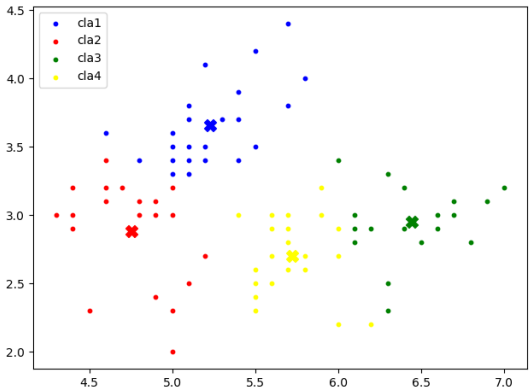


(4) $\theta_S = 0.7$

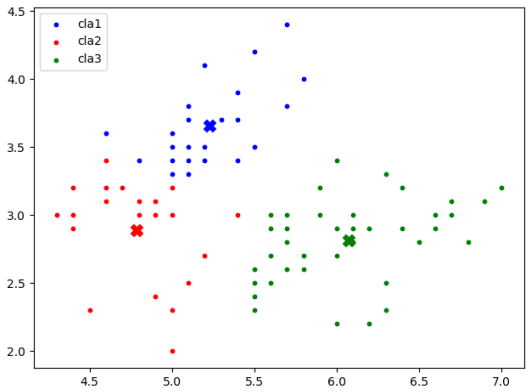
可以看到，当$\theta_S = 0.4$时，聚类域中样本分布距离标准差为 0.4，如果对于某一类样本的分布距离标准差大于该值，则有可能进行分裂，ISODATA 算法将该数据聚类为 3 类，随着θ_S的增大，聚类数逐渐减少。注意θ_S的设定因数据集而异。

(c) 设置θ_C分别为 0.1、0.5、0.6、0.7 其他参数如下，得到实验结果。

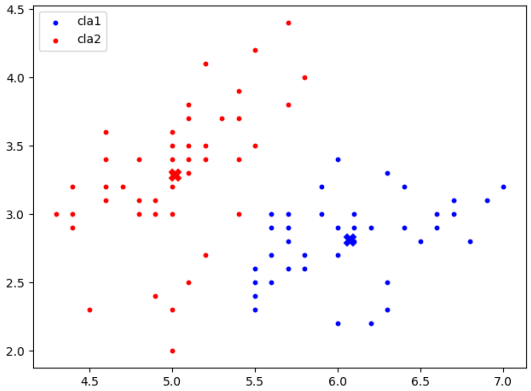
K	θ_N	θ_S	L	N_C	I
3	5	0.4	3	1	1000



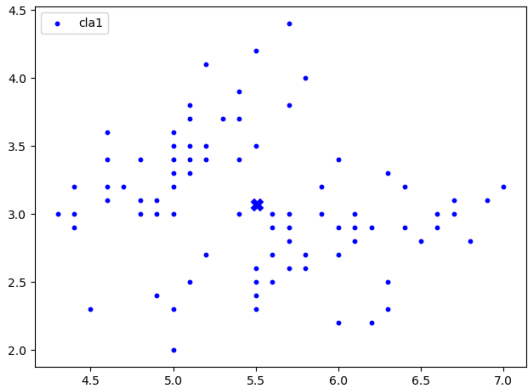
(1) $\theta_C = 0.1$



(2) $\theta_C = 0.5$



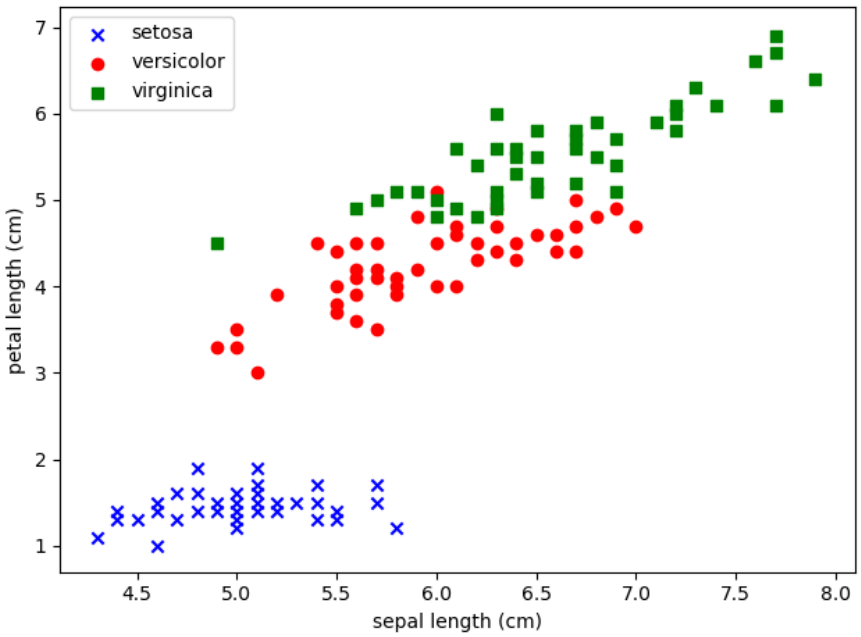
(3) $\theta_C = 0.6$



(4) $\theta_C = 0.7$

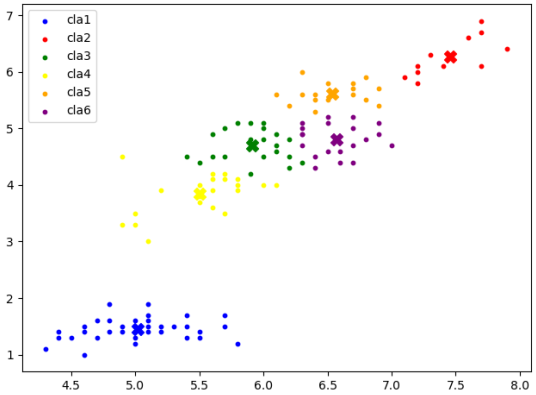
可以看到，当$\theta_C = 0.1$时，聚类中心最小距离为 0.1，ISODATA 算法将该数据聚类为 4 类，随着θ_C的增大，由于增加的聚类中心最小距离的限制，聚类数逐渐减少。

(2) 选取“setosa”、“versicolor”、”virginica” 三类进行聚类，并选取“'sepal length (cm)'”和 “petal length (cm)”这两个特征。用 matplotlib 库画出这两类数据：

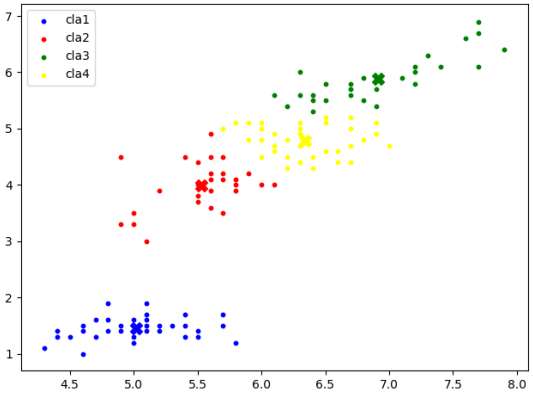


(a) 设置θ_N分别为 10、20、30、40，其他参数如下，得到实验结果。

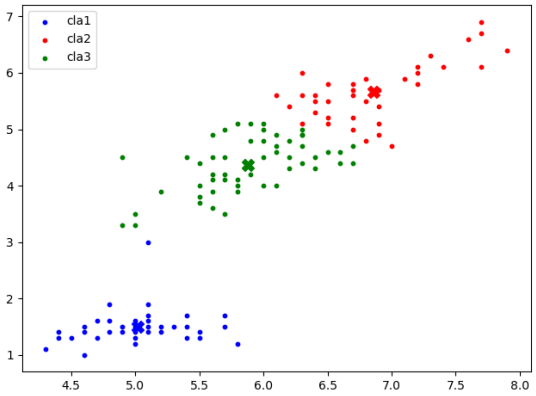
K	θ_S	θ_C	L	N_C	I
3	0.4	0.1	3	1	1000



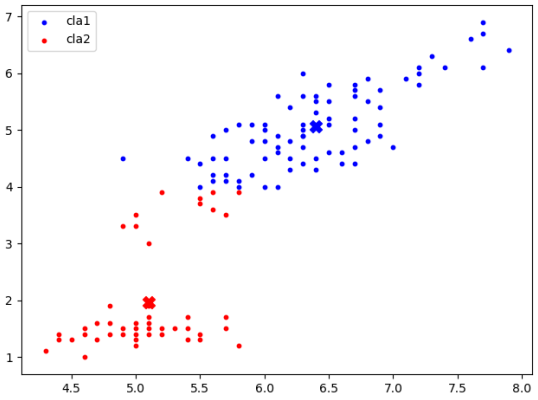
(1) $\theta_N = 10$



(2) $\theta_N = 20$



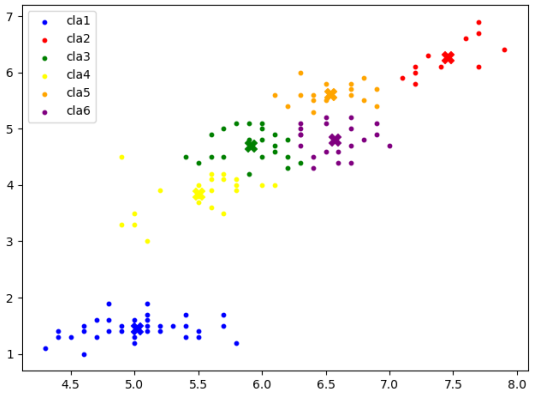
(3) $\theta_N = 30$



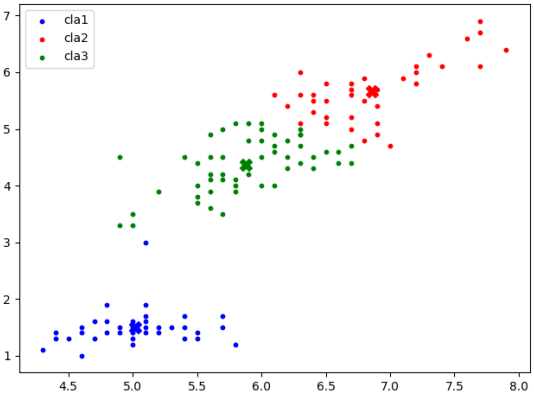
(4) $\theta_N = 40$

(b) 设置θ_S分别为 0.4、0.6、0.8、1，其他参数如下，得到实验结果。

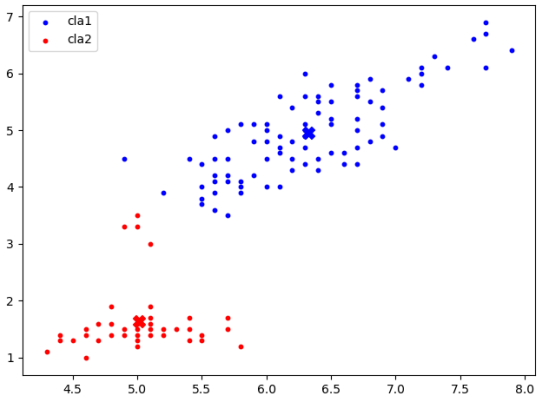
K	θ_N	θ_C	L	N_C	I
3	10	0.1	3	1	1000



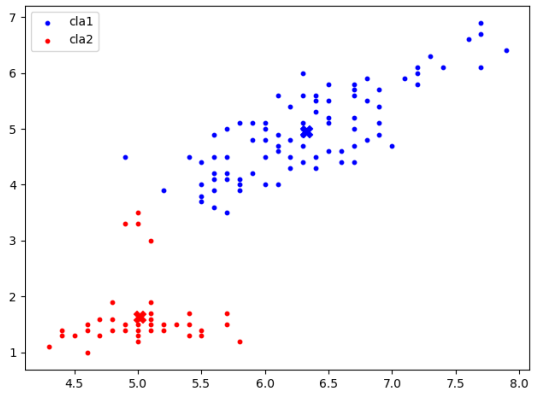
(1) $\theta_S = 0.4$



(2) $\theta_S = 0.6$



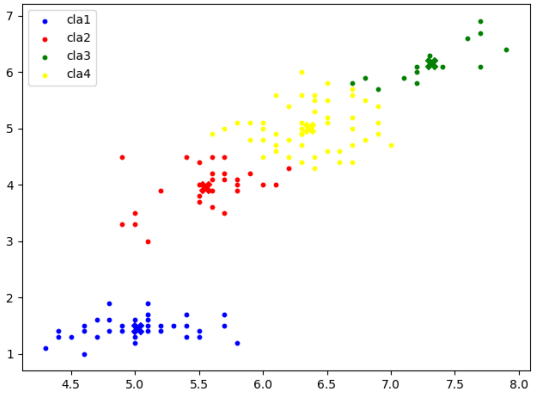
(3) $\theta_S = 0.8$



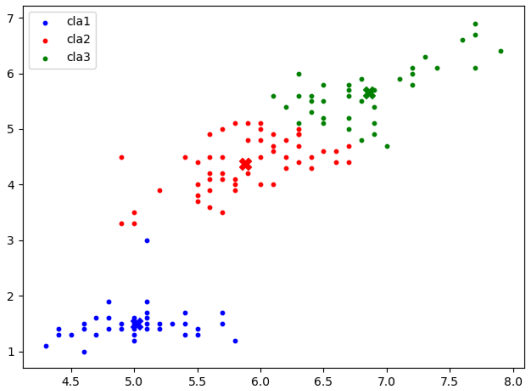
(4) $\theta_S = 1$

（c）设置θ_C分别为 0.5、0.6、0.7、0.8 其他参数如下，得到实验结果。

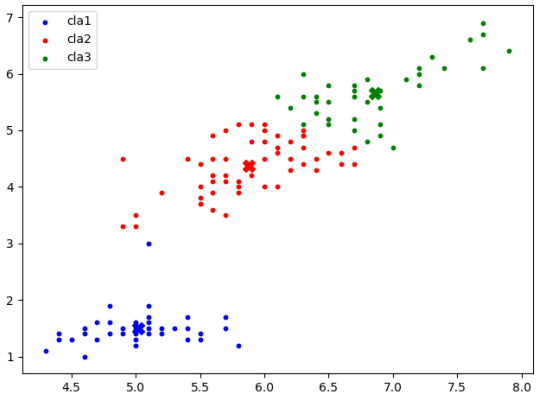
K	θ_N	θ_S	L	N_C	I
3	5	0.5	3	1	1000



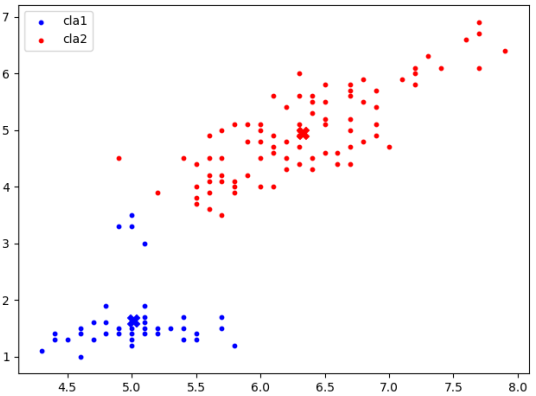
(1) $\theta_C = 0.5$



(2) $\theta_C = 0.6$



(3) $\theta_C = 0.7$



(4) $\theta_C = 0.8$

五、实验结论

根据前面的分析，K-means 算法的聚类效果好，初始中心选取对聚类效果的影响大。

ISODATA 算法的聚类效果更好，初始中心的选取对聚类效果的影响较 K-means 算法小。当聚类中心数未知时，建议使用 ISODATA 算法，并在迭代过程中合理调节参数。当聚类中心数已知时，建议使用 K-means 算法，能得到略逊于 ISODATA 的聚类效果，同时满足聚类需求，又能避免 ISODATA 中众多参数的设置问题。

通过本次实验，掌握了聚类分析的基本思想、相似性测度和聚类准则，进一步熟悉了最近邻聚类方法、K 均值算法、迭代自组织算法以及聚类结果的评价。