**实验三 聚类分析**

1. **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。

1. **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；

1. **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维数据集聚集成k个簇W1，W2，……，Wk，它们的质心依次为c1，c2，……，ck，其中ci=，ni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=

式中，(xj，)是xj与ci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c1，c2，……，ck。

（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：

开始

输入聚类个数k，数据个数n

初始化K个聚类中心

分配各个数据对象到距离最近的类中

重新计算各个聚类的中心

是否收敛

N

Y

输出聚类结果

结束

**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;

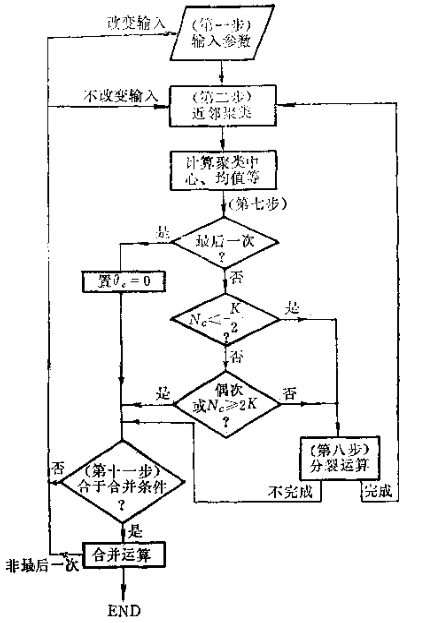
(c) 初始化分类, 按照与聚类中心距离最小的原则将各样本分类;

(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。

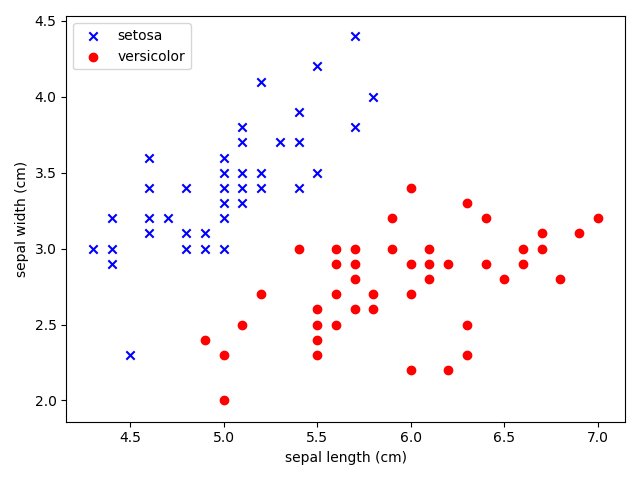
ISODATA算法示意图如下：



1. **实验过程**

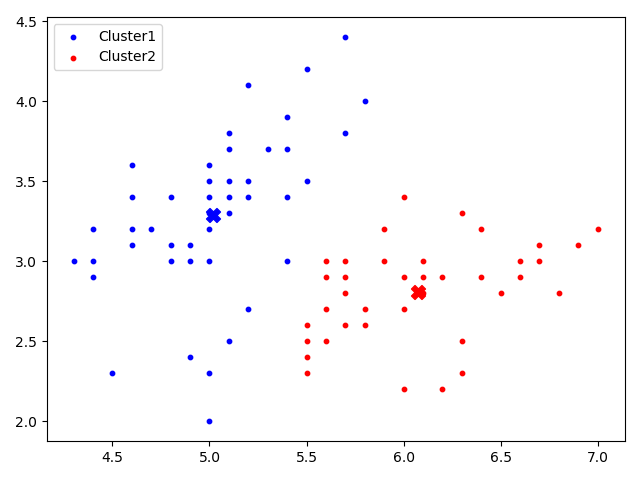
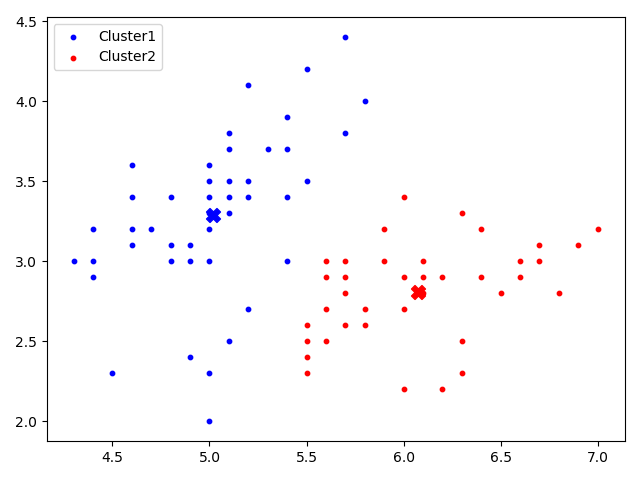
**1. k-means算法实验**

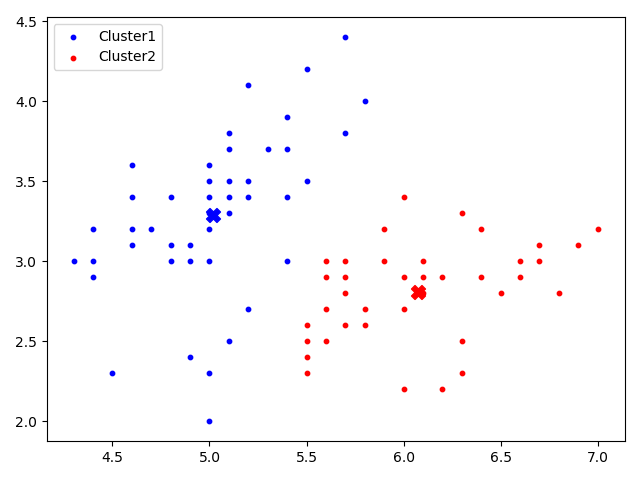
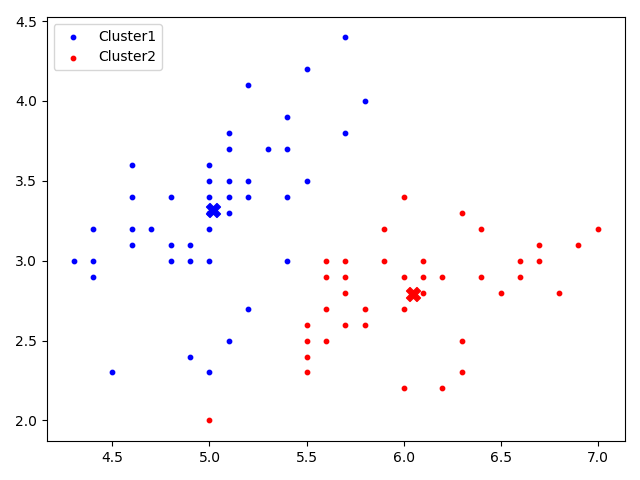
（1）选取“setosa”和“versicolor”两类进行聚类，并选取“sepal length (cm)”和 “sepal width (cm)”这两个特征。用matplotlib库画出这两类数据：



（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]

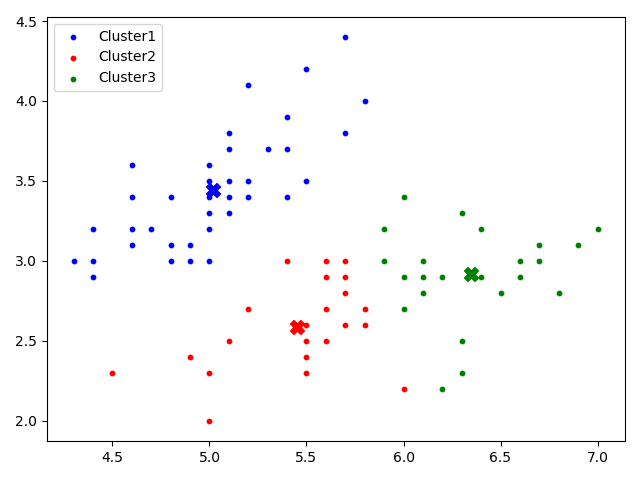
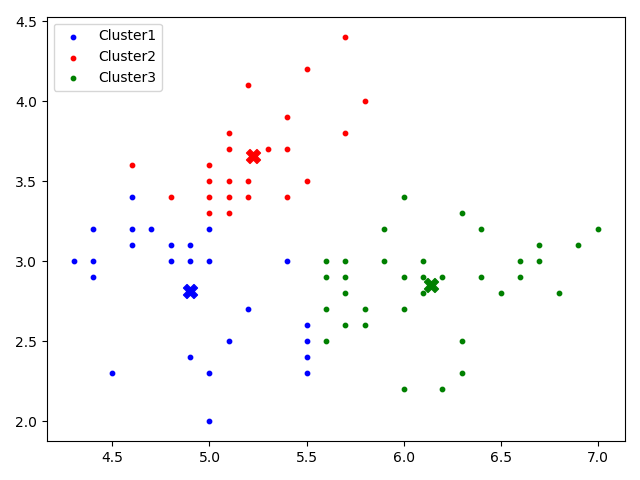
初始聚类中心：[4.765, 2.648], [5.414, 3.829] 初始聚类中心：[6.271, 4.041], [6.072, 2.519] 

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。

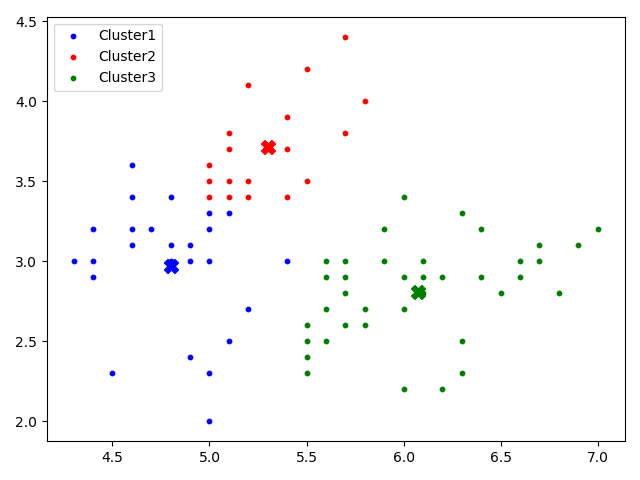
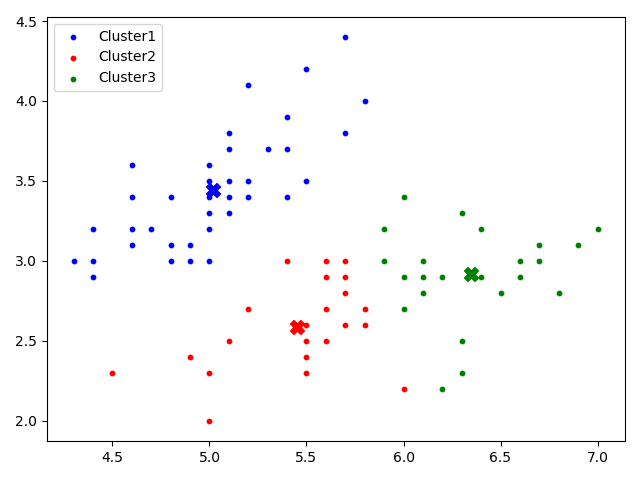
（b）设置k值为3，随机选取初始聚类中心，实验结果如下：

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 |  |
|  | 3.205 |

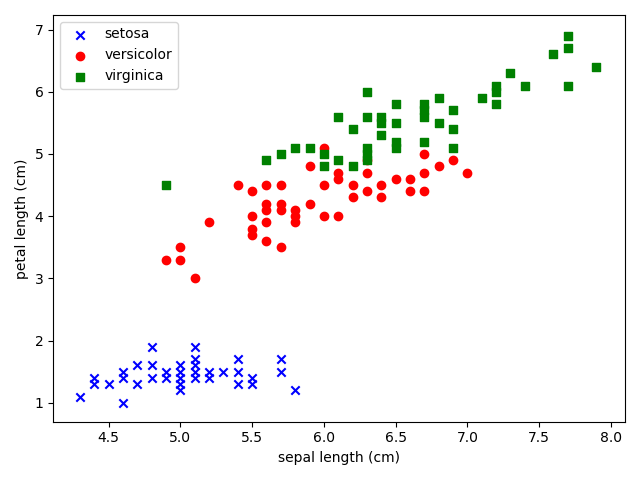
诸聚类域中样本数目：

|  |  |
| --- | --- |
| 聚类域 | 样本数目 |
|  | 50 |
|  | 50 |

诸聚类域内样本的距离方差：

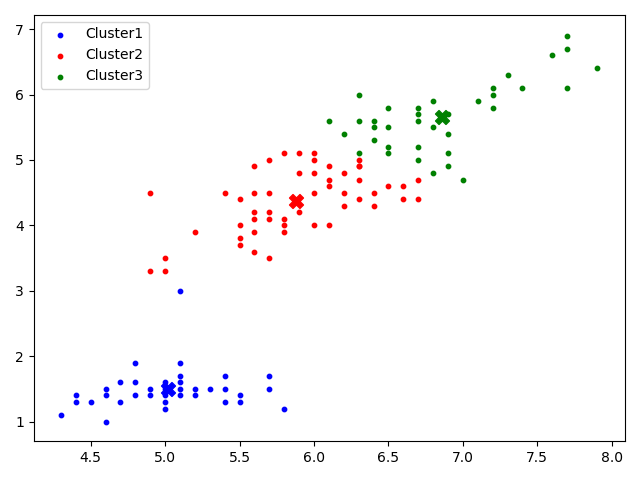
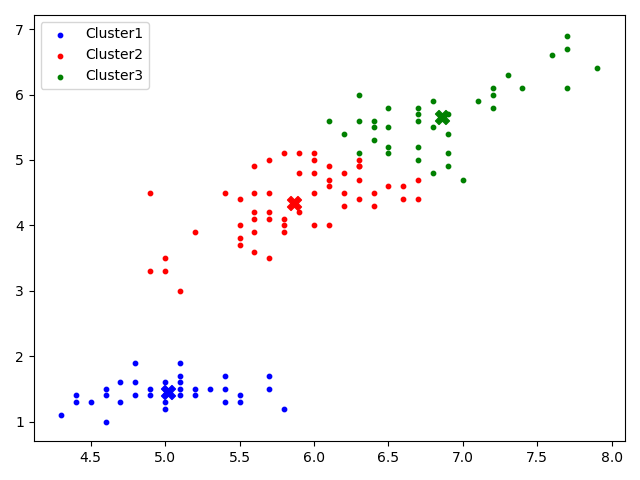
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 距离标准差  聚类域 |  |  |  |  |
|  | 0.349 | 0.377 | 0.172 | 0.106 |
|  | 0.511 | 0.311 | 0.465 | 0.196 |

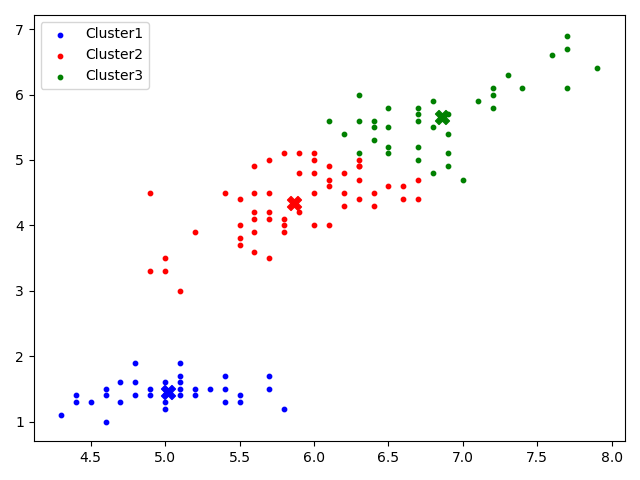
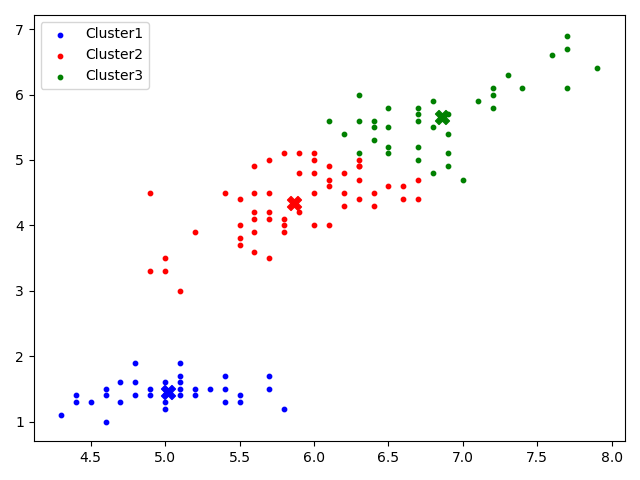
(2) 选取“setosa”、“versicolor”、”virginica” 三类进行聚类，并选取“'sepal length (cm)'”和 “petal length (cm)”这两个特征。用matplotlib库画出这三类数据：



可以看到三类模式样本形成相距较远、孤立的区域分布。

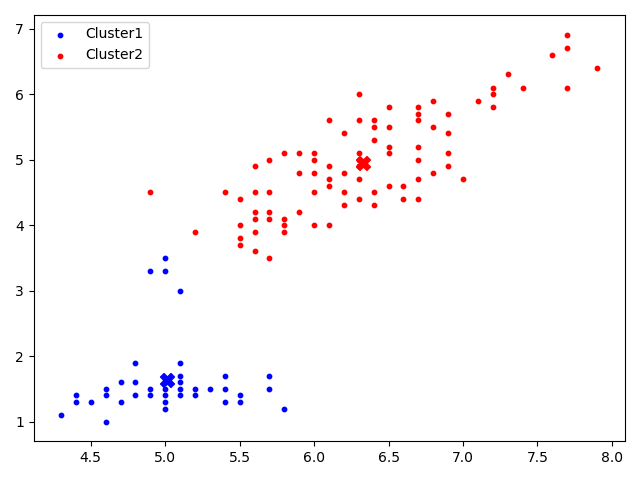
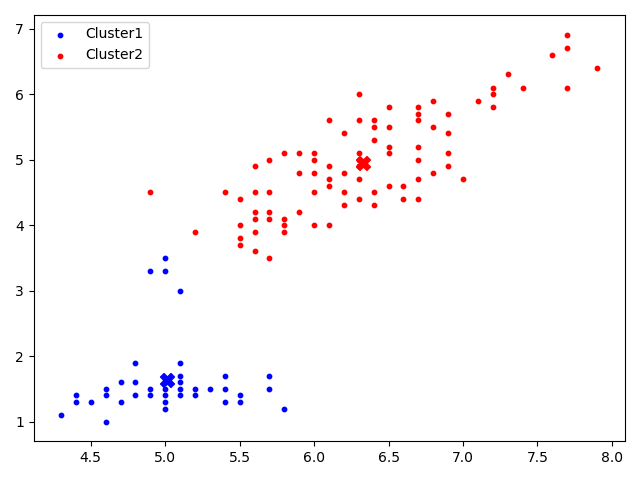
1. 取k=3，随机初始化聚类中心，多次实验结果如下：

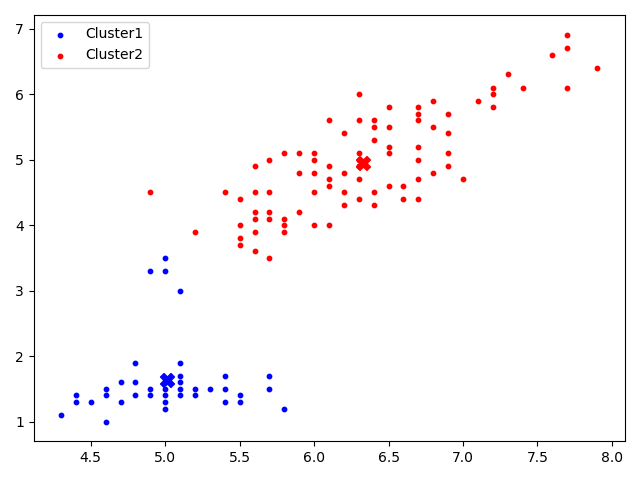
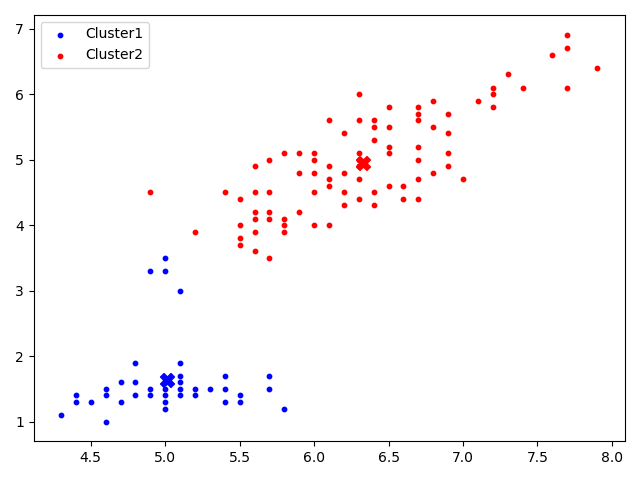




可以看到，在这种数据分布情况下，算法能够收敛到一个基本一致的聚类结果。此外，当k取2时，红色和绿色两个簇应该也是可以合并为一个簇的。

1. 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 |  |  |
|  | 4.988 | 3.346 |
|  |  | 1.788 |

诸聚类域中样本数目：

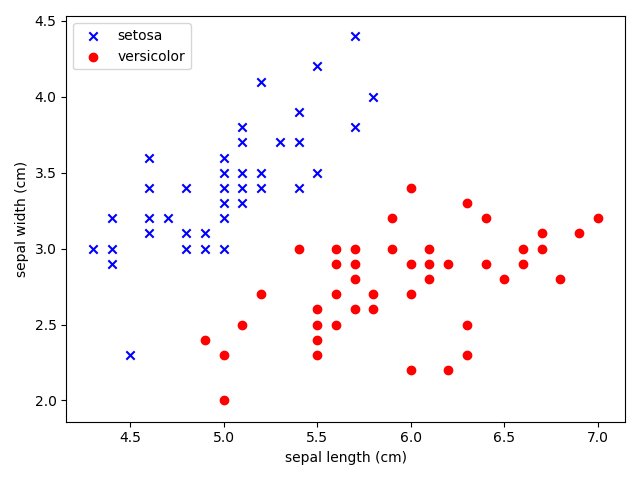
|  |  |
| --- | --- |
| 聚类域 | 样本数目 |
|  | 50 |
|  | 39 |
|  | 61 |

诸聚类域内样本的距离方差：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 距离标准差  聚类域 |  |  |  |  |
|  | 0.349 | 0.377 | 0.172 | 0.106 |
|  | 0.482 | 0.283 | 0.504 | 0.293 |
|  | 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：

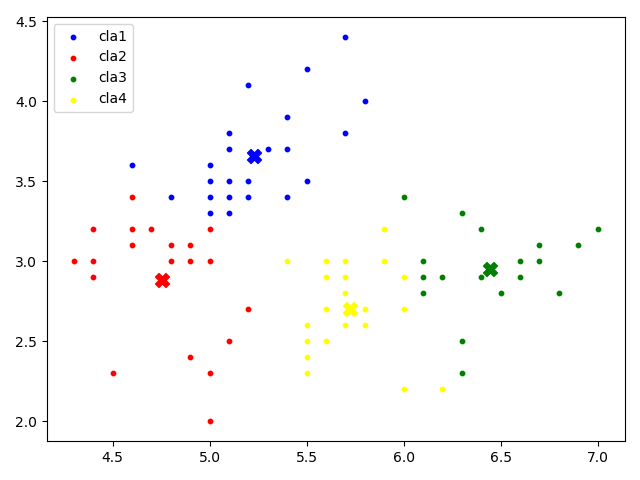
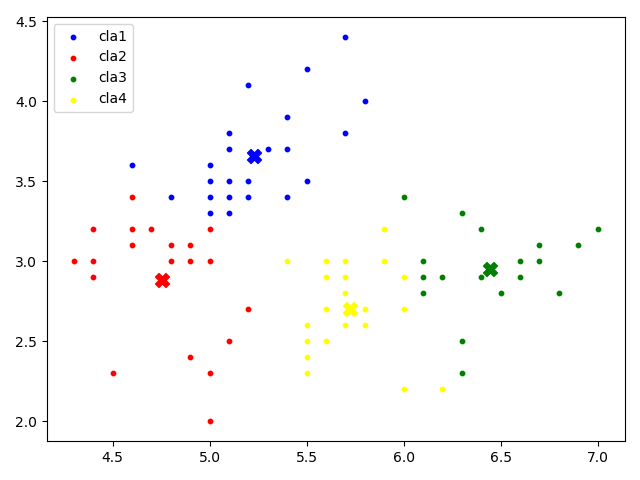


使用ISODATA算法对该数据进行聚类：

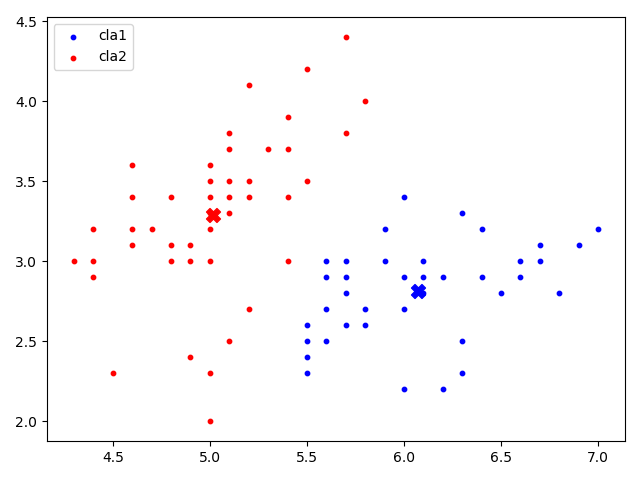
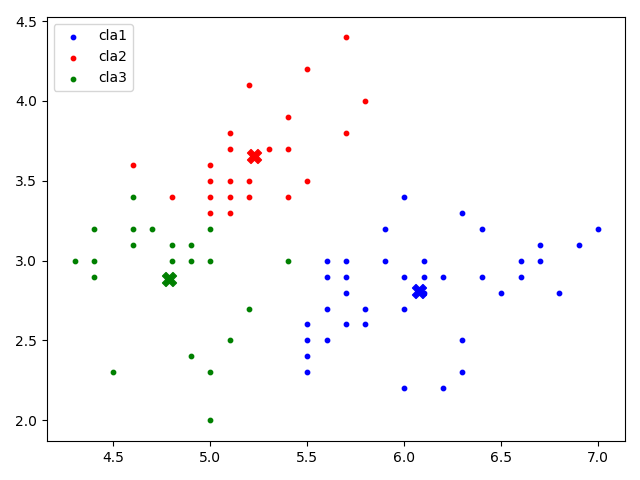
ISODATA算法的参数有：预期聚类中心数目、聚类域最少样本数目、聚类域中样本分布距离标准差、聚类中心最小距离、一次迭代可合并对数、初始聚类中心个数、迭代最大次数。

1. 设置分别为5、10、20、30，其他参数如下，得到实验结果。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 3 | 0.4 | 0.1 | 3 | 1 | 1000 |



(1) (2)

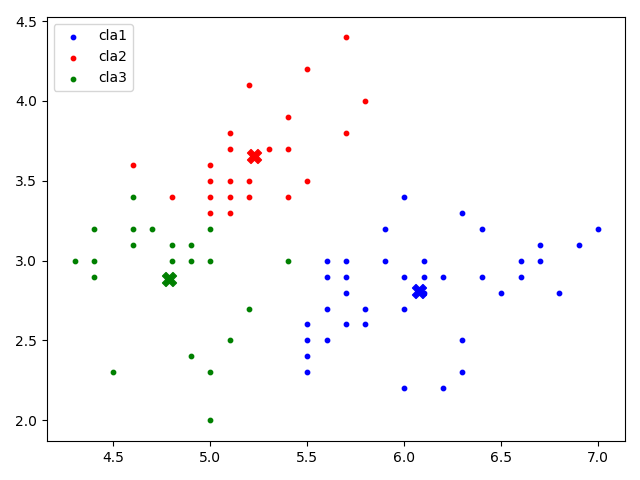
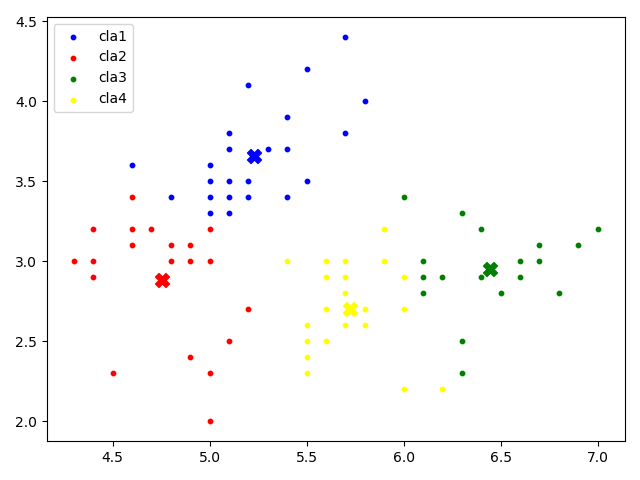


(3) (4)

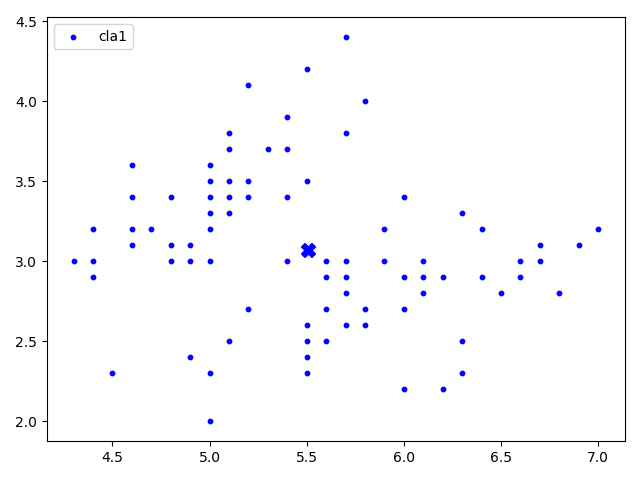
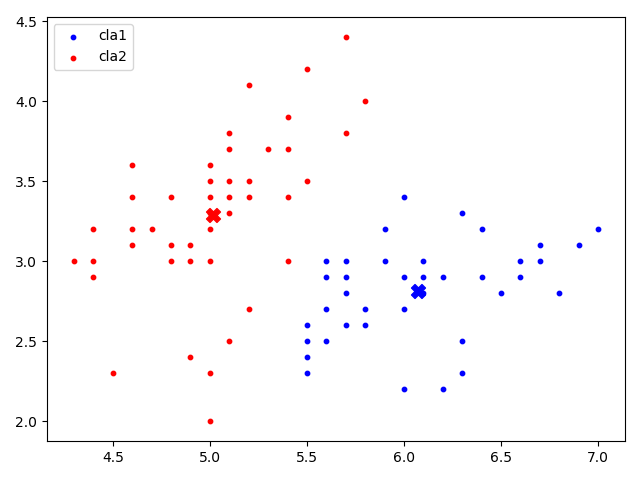
可以看到，当时，聚类域最少样本数目为5，ISODATA算法将该数据聚类为4类，随着的增大，由于增加的聚类域内最少样本数目的限制，聚类数逐渐减少。

1. 设置分别为0.4、0.5、0.6、1，其他参数如下，得到实验结果。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 3 | 10 | 0.1 | 3 | 1 | 1000 |



(1) (2)

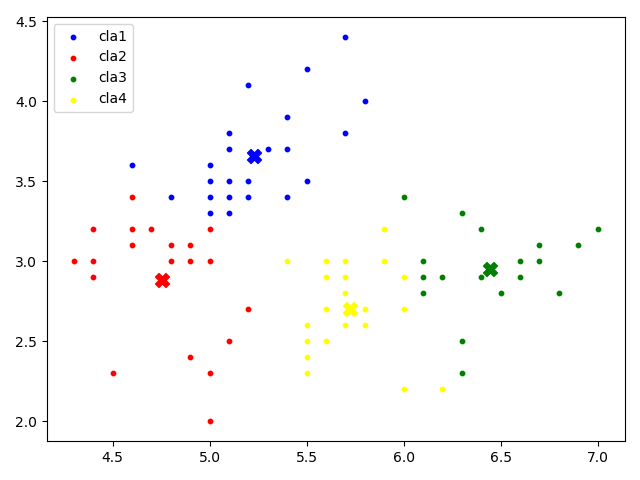
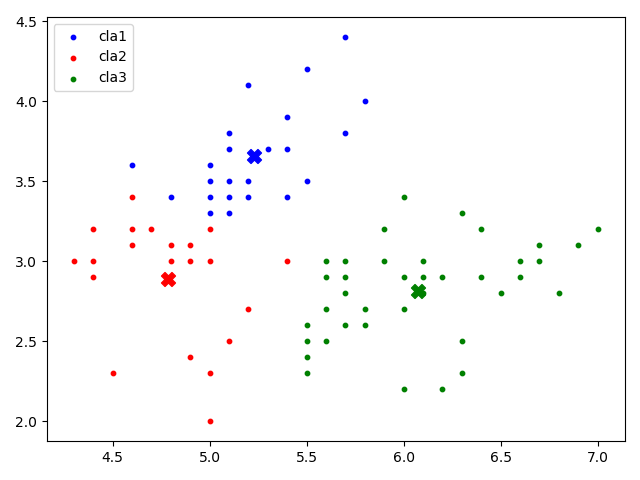


(3) (4)

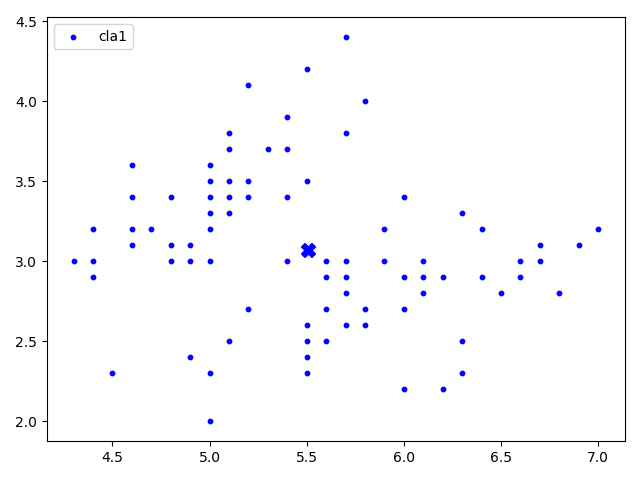
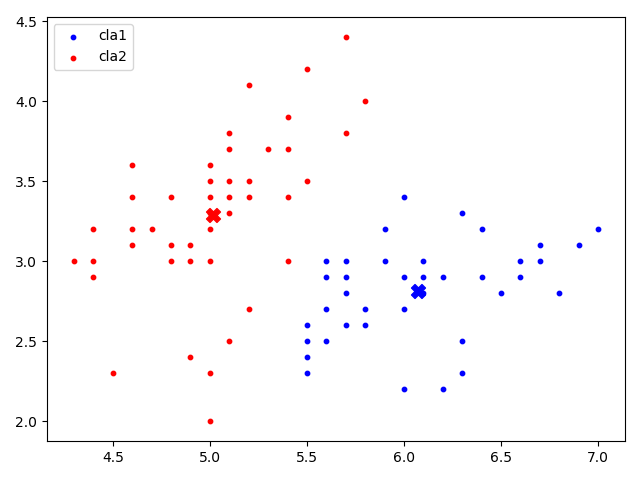
可以看到，当时，聚类域中样本分布距离标准差为0.4，如果对于某一类样本的分布距离标准差大于该值，则有可能进行分裂，ISODATA算法将该数据聚类为3类，随着的增大，聚类数逐渐减少。注意的设定因数据集而异。

1. 设置分别为0.1、0.5、0.6、0.7其他参数如下，得到实验结果。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 3 | 5 | 0.4 | 3 | 1 | 1000 |

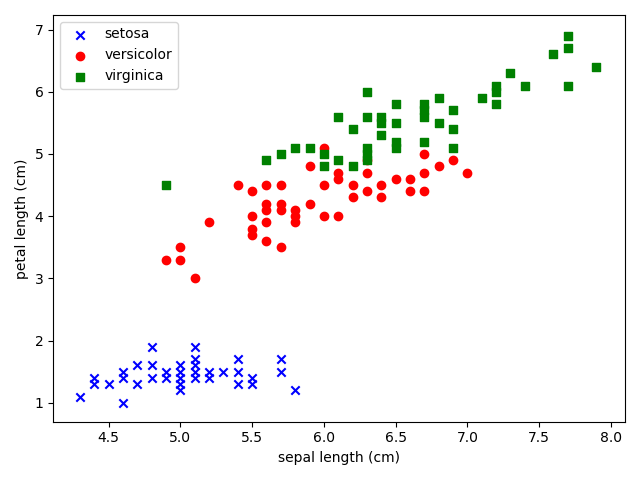
(1) (2)



(3) (4)

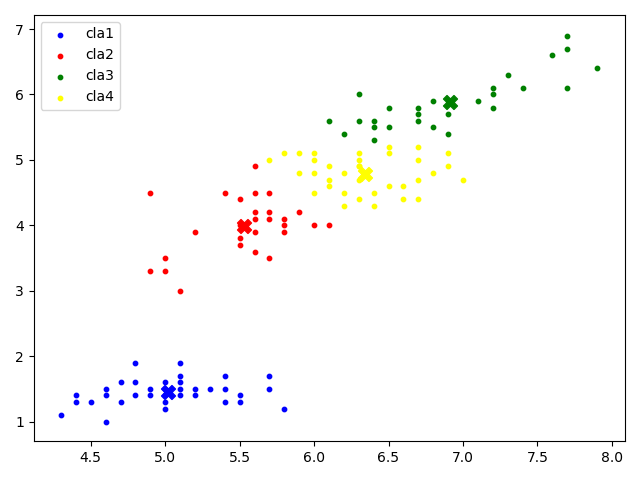
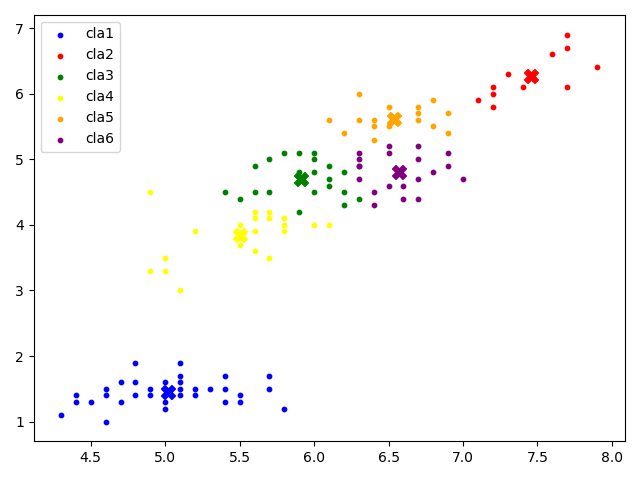
可以看到，当时，聚类中心最小距离为0.1，ISODATA算法将该数据聚类为4类，随着的增大，由于增加的聚类中心最小距离的限制，聚类数逐渐减少。

(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：

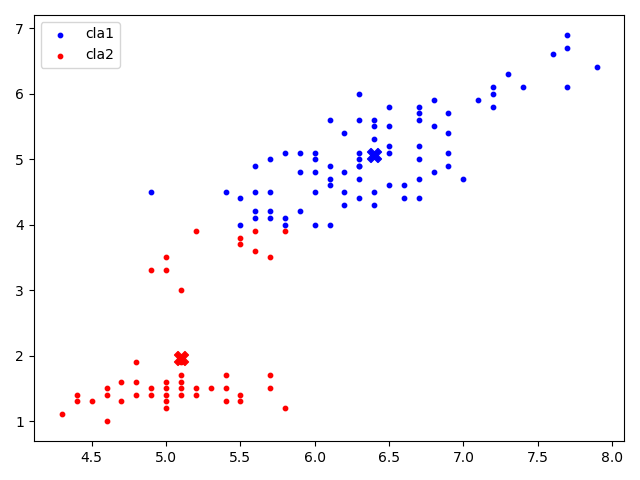
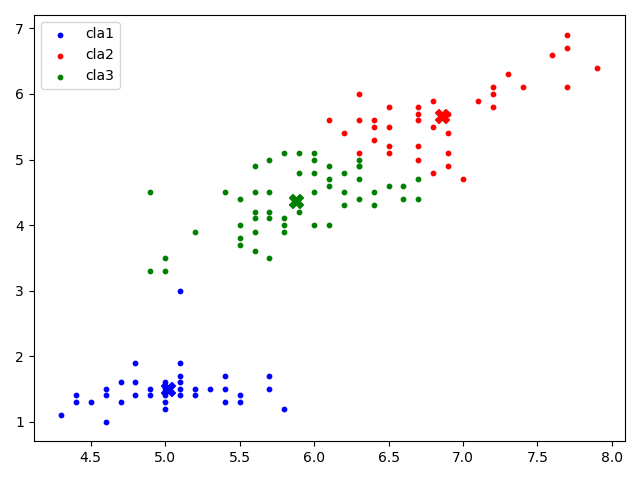


1. 设置分别为10、20、30、40，其他参数如下，得到实验结果。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 3 | 0.4 | 0.1 | 3 | 1 | 1000 |



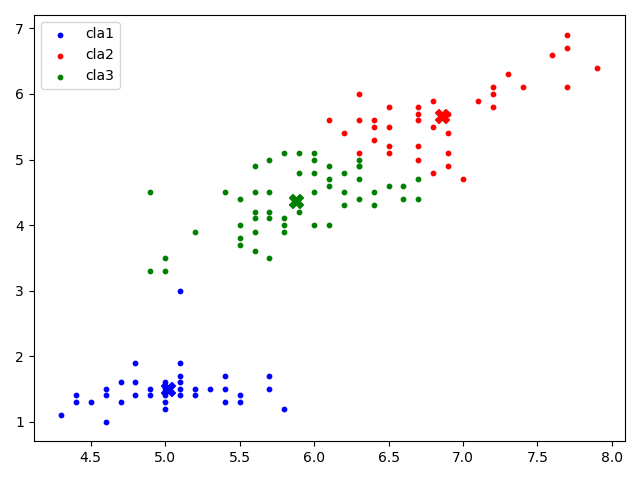
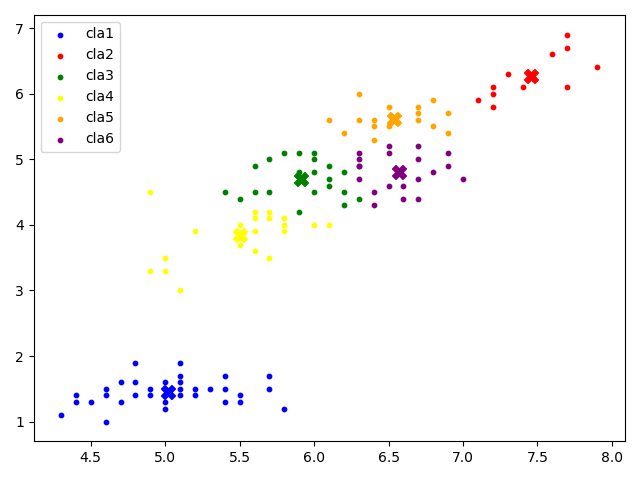
(1) (2)



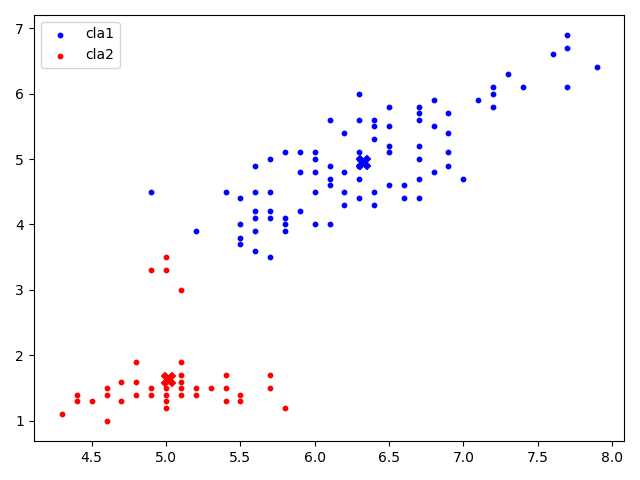
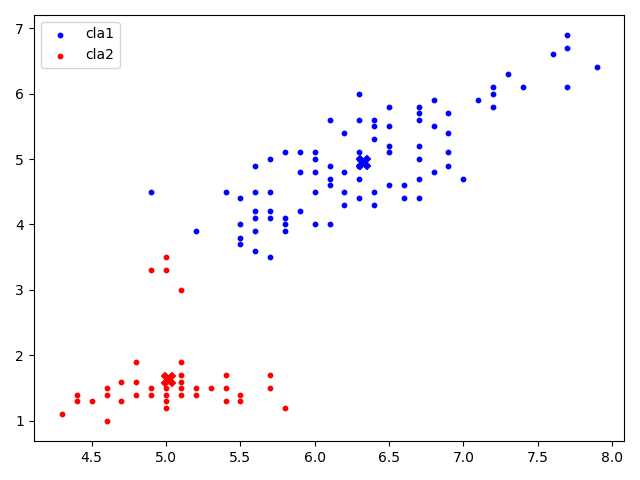
(3) (4)

1. 设置分别为0.4、0.6、0.8、1，其他参数如下，得到实验结果。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 3 | 10 | 0.1 | 3 | 1 | 1000 |



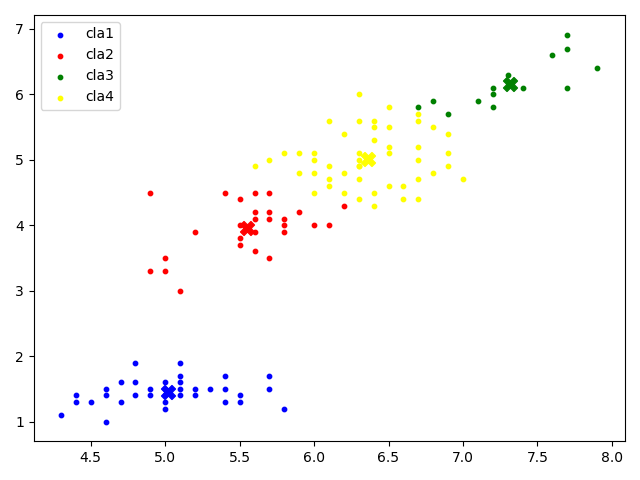
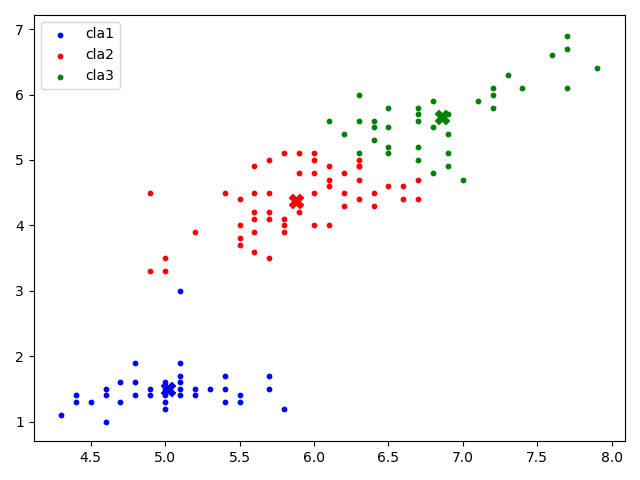
(1) (2)



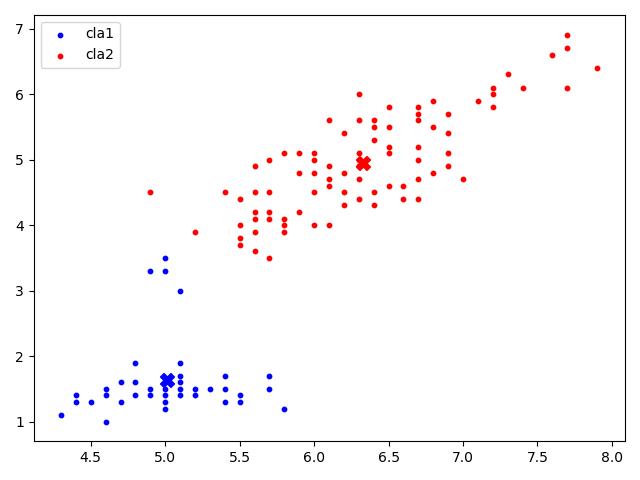
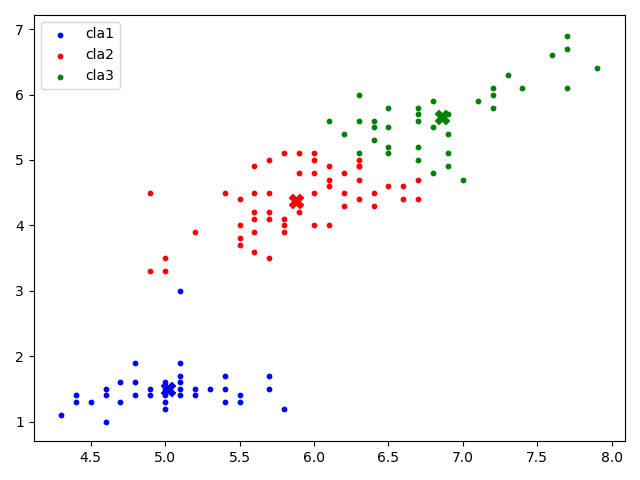
(3) (4)

（c）设置分别为0.5、0.6、0.7、0.8其他参数如下，得到实验结果。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 3 | 5 | 0.5 | 3 | 1 | 1000 |

(1) (2)



(3) (4)

**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。