模式识别实验报告

实验分工

- 陆 鸿 *161540110*

  代码编写、实验实施与分析

- 李 卉 *161540103*

  实验原理分析、实验分析

实验一　编程实现K-means算法

算法思想

1 从n个数据对象中随机选择k个对象作为初始簇中心；

2 计算剩下的对象到k个簇中心的相异度（距离），将这些对象分别划归到相异度最低的簇中；

3 根据聚类结果，重新计算k个簇各自的中心，计算方法是取各个簇中所有的元素各自维度的算数平均值；

4 按照新的中心重新聚类

5 重复第（4）步，直到聚类结果不再变化；

6 输出聚类结果。

程序结构

```
|_ _ _ _K-means
||_ _ _ _ loadDataSet(filename)
||_ _ _ _ distE(A,B)  //欧氏距离
||_ _ _ _ disM(A,B)   //曼哈顿距离
||_ _ _ _ randCenter(Matrix, k)
||_ _ _ _ KMeans(Matrix,k)
||_ _ _ _ Main
```

数据集

UCI数据集，链接[perfume Data Set]

> Abstract: This data consists of odors of 20 different perfumes. Data was obtained by using a handheld odor meter (OMX-GR sensor) per second for 28 seconds period. 该数据集用于香水的分类实验

数据集处理

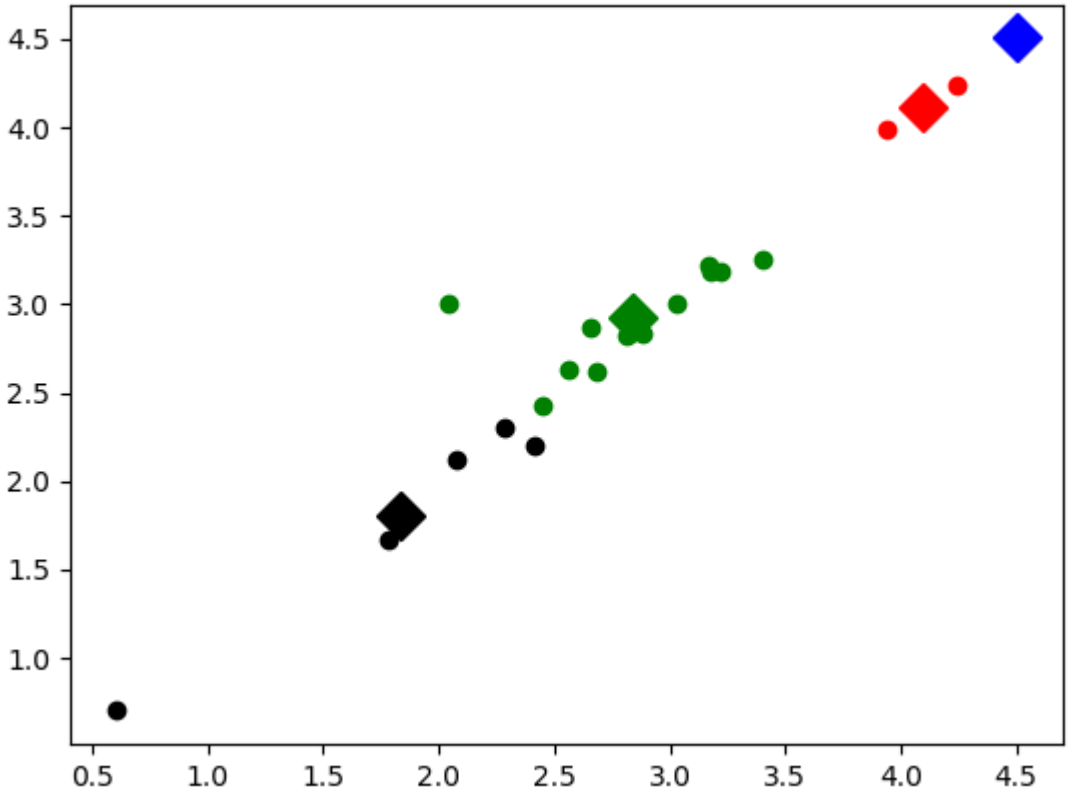
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	ajayeb	63,529	63,529	64,528	64,528	64,528	64,528	64,526	64,526	64,528	64,528	64,528	64,528	64,528	64,528	64,528	64,528	64,526	64,526	63,514	64,526	64,526	64,526
2	aday jmal	61,445	61,434	60,423	61,434	60,412	61,421	60,435	61,434	60,447	60,449	61,455	61,470	61,470	61,469	61,459	61,443	60,435	61,445	61,445	61,459	61,457	61,469
3	amreaj	56,040	56,040	56,040	57,040	57,040	56,040	56,040	56,040	55,040	57,042	57,042	57,042	57,042	57,042	57,042	57,042	57,042	58,041	58,041	58,041	57,042	57,042
4	aood	73,094	72,090	72,087	73,086	72,083	72,083	72,079	72,079	72,079	72,076	72,075	72,072	72,072	72,072	72,069	71,069	71,069	71,069	71,073	71,076	71,076	71,076
5	asgar_ali	68,230	68,230	68,230	68,231	68,231	68,231	68,231	68,238	68,231	68,231	68,231	68,230	68,230	68,231	68,231	68,230	68,230	68,230	68,223	68,223	68,223	68,223
6	bukhoor	71,046	70,046	70,046	70,046	70,046	70,046	70,046	70,048	70,048	70,048	70,046	70,046	70,048	70,049	70,048	70,048	70,049	70,049	70,049	70,049	70,049	70,049
7	bukhoor	71,046	70,046	70,046	70,046	70,046	70,046	70,046	70,048	70,048	70,048	70,046	70,046	70,048	70,049	70,048	70,048	70,049	70,049	70,049	70,049	70,049	70,049
8	dehenalaoc	68,132	69,132	69,132	69,131	68,127	69,127	68,122	69,122	69,117	69,122	68,122	68,122	68,123	68,127	68,128	68,128	68,132	68,132	68,132	68,132	68,132	68,128
9	junaid	72,542	72,541	71,528	71,514	71,514	71,514	71,514	72,513	71,514	71,514	71,514	72,527	72,527	72,527	72,527	71,514	72,513	71,501	71,501	71,501	71,501	71,514
10	kausar	72,586	72,586	72,586	72,586	72,586	72,586	72,586	72,586	72,586	72,586	72,586	72,572	72,571	72,571	72,572	72,571	72,557	72,557	72,557	72,557	72,557	72,557
11	rose	62,999	62,999	62,999	62,999	62,999	62,999	62,999	63,999	62,999	62,999	62,999	66,999	62,999	62,999	62,999	62,999	62,999	62,999	61,999	61,999	62,999	62,999
12	solidmusk	46,015	46,015	46,015	46,015	46,015	46,015	46,015	46,015	46,015	46,015	46,015	46,015	46,015	46,015	48,016	48,016	48,016	48,016	48,016	48,016	48,016	48,016
13	TeaTreeOil	82,372	82,361	82,351	82,351	82,331	82,351	82,341	82,361	82,362	82,372	82,361	82,351	82,372	82,361	82,341	82,351	82,383	82,394	82,383	82,372	82,361	82,341
14	raspberry	78,999	78,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999	79,999
15	RoseMusk	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056	85,056
16	strawberry	71,999	71,999	71,999	71,999	71,999	72,999	71,999	71,999	71,999	71,999	71,999	72,999	71,999	72,999	71,999	71,999	71,999	71,999	71,999	70,999	71,999	71,999
17	onstructed	66,680	66,663	67,678	67,643	66,626	67,624	67,624	67,608	67,622	67,624	67,624	66,626	66,626	67,641	67,639	67,624	67,624	67,622	67,606	71,577	71,545	71,530
18	rolina_herr	61,053	61,056	61,056	62,059	62,059	62,059	62,059	63,059	62,059	62,059	61,060	62,059	62,059	62,059	62,059	63,059	62,056	63,056	63,055	63,052	62,050	64,052
19	dh_ma'alat	66,218	66,218	66,218	66,219	66,219	66,219	66,219	66,226	66,226	66,219	66,218	66,219	66,220	66,226	66,226	66,226	66,226	66,226	66,219	66,220	66,226	66,226
20	constructec	67,149	67,149	67,149	66,145	66,145	66,145	66,145	66,145	66,145	66,145	66,144	66,145	66,144	66,144	66,144	66,144	66,144	66,144	66,144	66,144	66,144	66,144

如上图所示，数据列出了20种香水的28个时刻的气味值，不同的气味值表示了香水之间的区别，所以可以用这个数据集进行分类操作，我将数据集分为两份，分别作为两个维度的值，并且利用 $f(x) = \frac{x-40000}{10000}$ 来对数据进行归一化，使数据缩小到0-1的范围。

实验结果

实验变量取值：测度标准（欧式距离），K = 4

其中菱形表示簇中心点，其余的各个分簇颜色各不相同。



实验二　相似性测度比较

算法：K-means

数据集与实验一相同

选择的测度距离比较

欧氏距离公式与特点：

公式：$D_e = \sqrt{\sum (\vec{a} - \vec{b})^2}$

特点：欧式距离表明了每个特征点对于最终的计算结果的贡献是相等的，如果我们不做相应的权值设置的话，欧式距离会将目标特征的不同属性之间的差别等同看待，这一点有些时候不能满足实际要求。没有考虑到总体变异对于距离远近的影响。

马氏距离（曼哈顿距离）公式与特点：

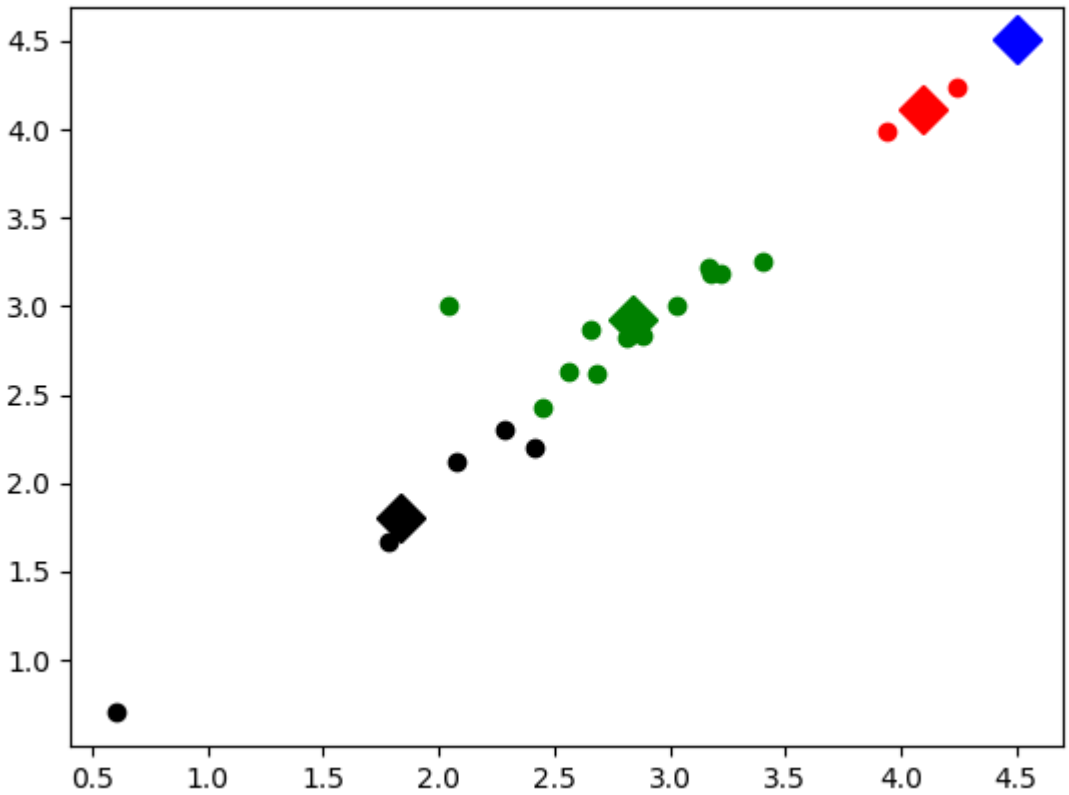
公式：$D_m = \sum |\vec{a} - \vec{b}|$

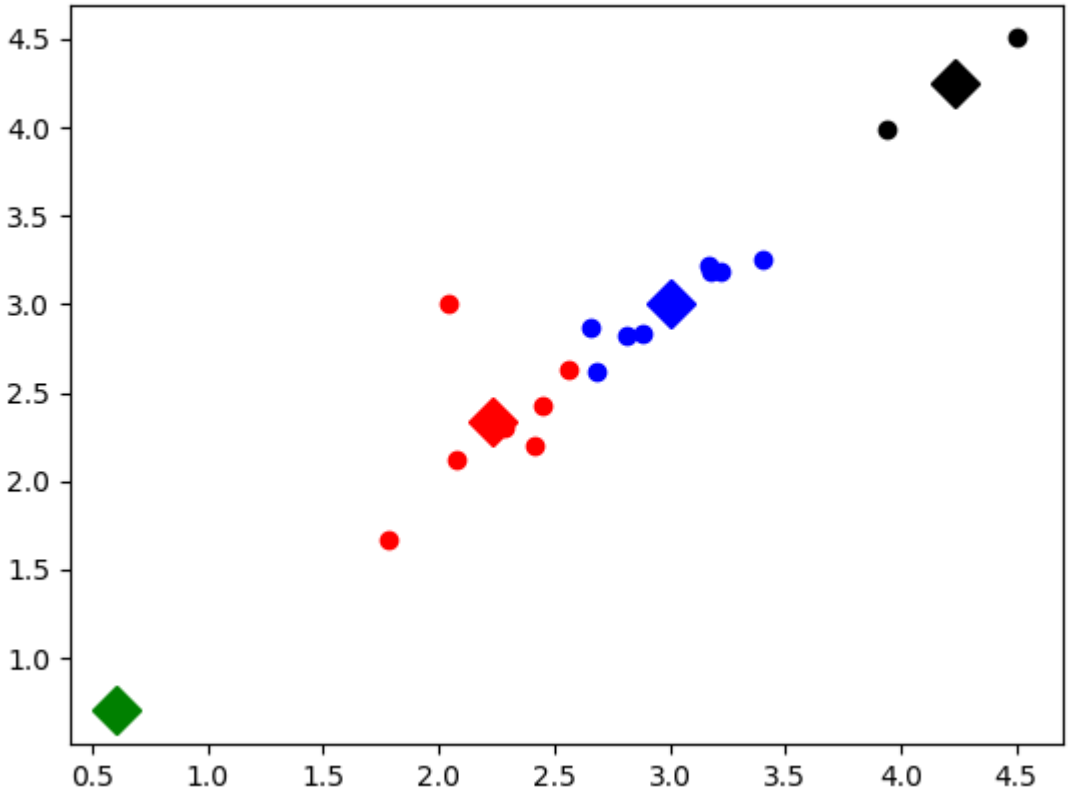
特点：在实验中我感觉马氏距离具有较高的稳定性，因为马氏距离与测量测度单位无关，他只关心两个数据点之间的联系，只考虑各种特征之间的联系。但是马氏距离有些时候不会顺利算出。

下面为两种测度距离的分类效果：

> 欧氏距离

> 通过实验发现，连续输出10次实验结果，欧氏距离作为测度时稳定性较差

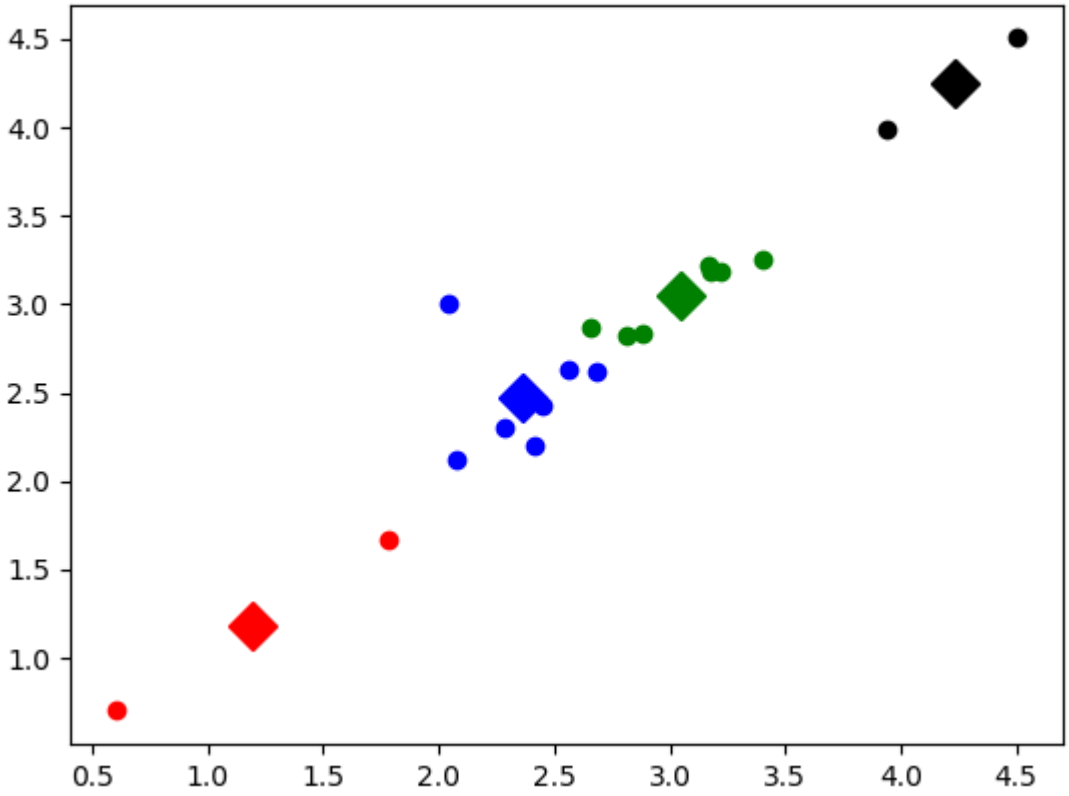


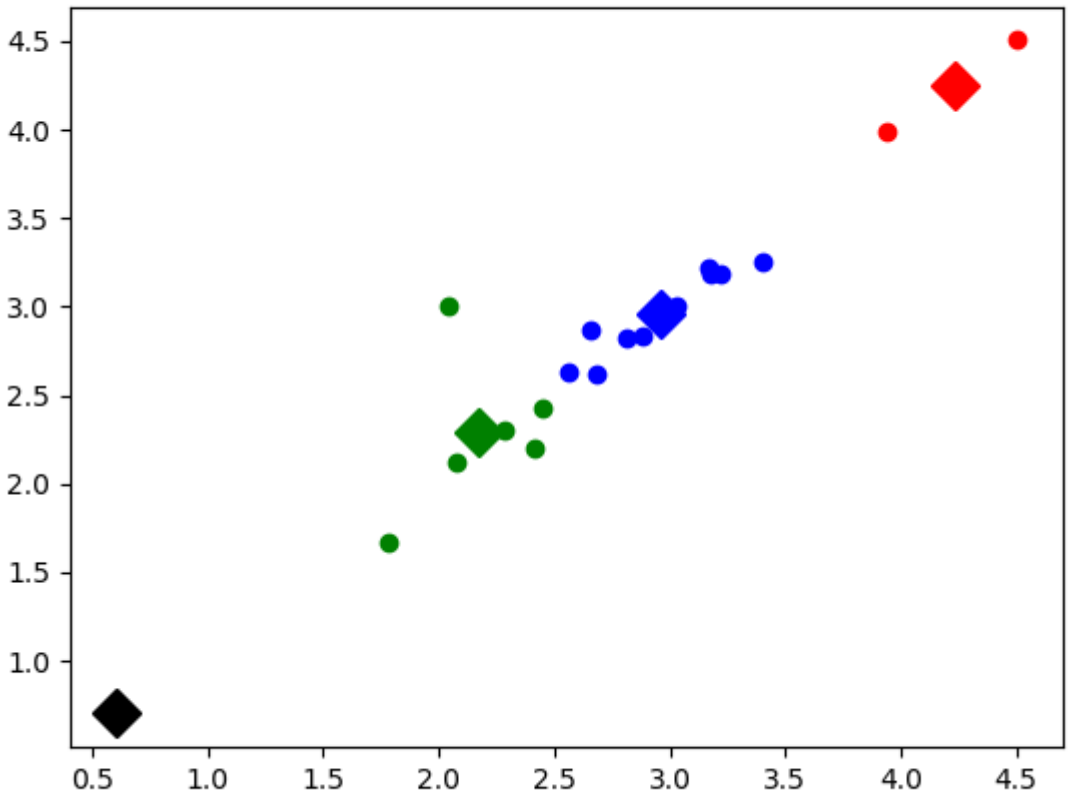


> 曼哈顿距离
>
> 由实验连续输出结果可以得到曼哈顿的计算结果比价稳定，但是有些时候计算不出结果，这也是由于该距离测度过于夸大了细小的特征值造成的。

> 簇中心计算不出的情况：

```
[[3.43782 3.42407]
 [2.47968 2.70958]
 [    nan     nan]
 [1.83288 1.8012 ]]
[[3.5856     3.572975  ]
 [2.53267778 2.63487778]
 [       nan        nan]
 [1.48736667 1.49896667]]
```


4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5


4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5

实验三　有监督学习分类算法的实验与分析

算法：决策树算法与SVM算法

数据集：鸢尾花数据集

算法简介

决策树

该算法根据数据的属性采用树状结构来建立决策模型，决策树模型常常用来解决分类与回归问题。

如何将数据快速的分类呢？选择一个合适的特征来作为判断节点是十分重要的，既可以快速的分类又可以减少树的深度，为了使分类后的数据集比较的纯，所以我们首先选择一种数据纯度函数。

我选择了基尼系数来作为评判的指标，其计算的是数据的不纯度。

首先介绍基尼系数的公式为 $Gini(D) = 1 - \sum_i^c p_i^2$, 从公式中我们可以看出当数据集中数据混合的程度越高，基尼指数也就越高。当数据集 D 只有一种数据类型，那么基尼指数的值为最低 0，所以每个节点我们要尽量挑选基尼系数低的指标。

支持向量机

简单来说, SVM的分类思想就是给定给一个包含正例和反例的样本集合，寻找一个超平面来对样本根据正例和反例进行分割，并且尽最大努力使分开的两个类别有最大间隔，这样才使得分隔具有更高的可信度，而且对于未知的新样本才有很好的分类预测能力和泛化能力。

为了寻找这样的分隔，需要找到两个和这个超平面平行和距离相等的超平面，在这两个超平面上的样本点也就是理论上离分隔超平面最近的点，是它们的存在决定了这两个超平面的位置，支撑起了分界线。

它们就是所谓的支持向量，这也正是支持向量机名字的由来。通过这两个超平面可以自然的得到间隔的计算方法，然后只需要最大化这个间隔即可。

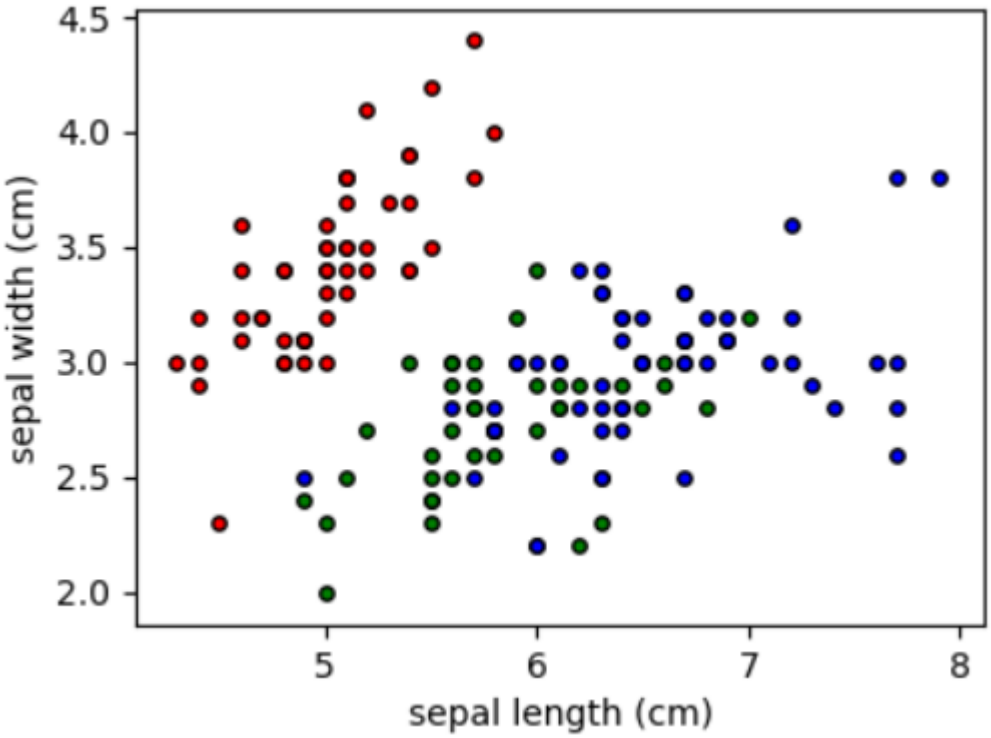
此外，为了处理线性不可分的问题，需要将一个低维的样本集映射到高维才可以变成线性可分这样才能使用SVM工作。但是，如果拿到低维数据直接映射到高维的话，维度的数目会呈现爆炸性增长，所以需要引入核函数（kernel function）。核函数的思想是寻找一个函数，这个函数使得在低维空间中进行计算的结果和映射到高维空间中计算内积<Φ(x1), Φ(x2)>的结果相同。这样就避开直接在高维空间中进行计算，而最后的结果却是等价的。

实验过程

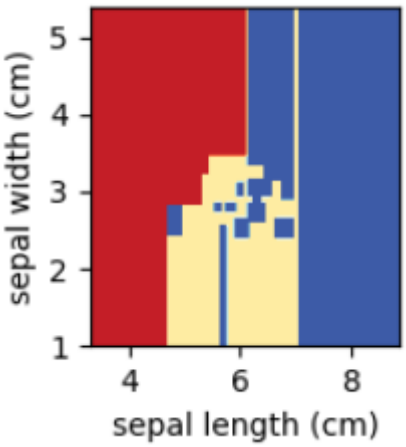
选择sepal length 与 sepal width 来给鸢尾花进行分类，该数据集中共有三种不同的花种，由于数据类数不多，选用两种特征分类是可以的。

决策树模型

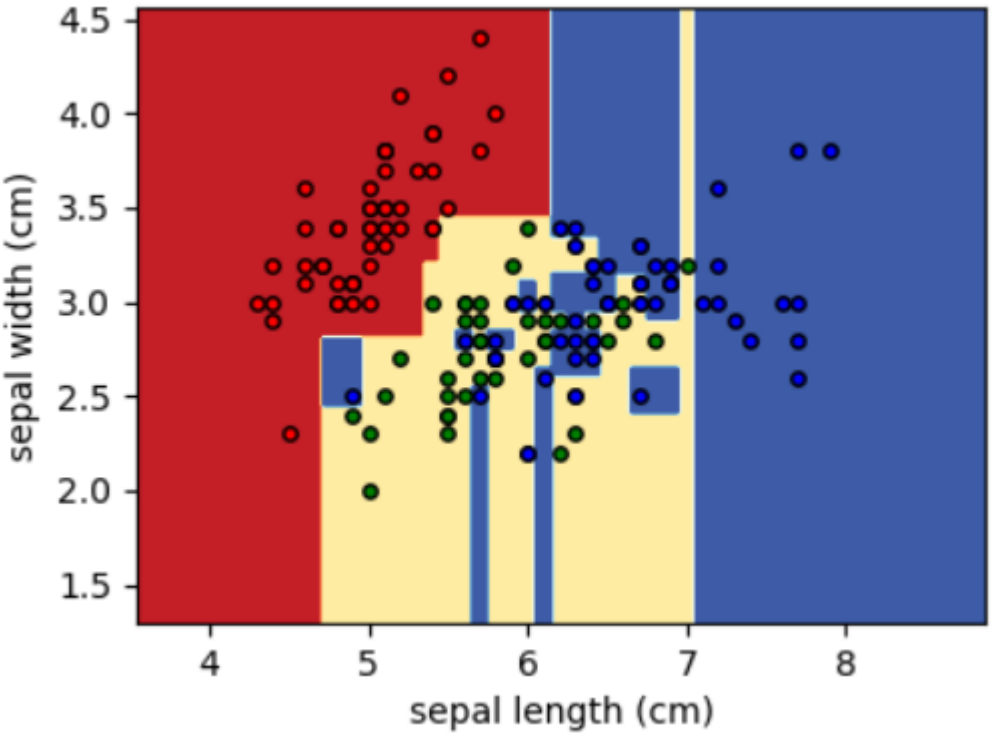
> 在sepal length 与 sepal width 两个关系形成的预测模型中，首先将其特征点分布描述出来，如下图所示，我分别用红色、绿色和蓝色来表示鸢尾花的不同类别。



> 用决策树训练出了decision boundary 如下图所示，通过与SVM的训练结果会发现，决策树过于依赖历史数据，并且有过拟合的趋势，我通过搜索发现解决过拟合可以用"剪枝"的方法实现，但是由于只使用了两种属性来进行分类，所以对于特征值的依赖性是一定存在的。

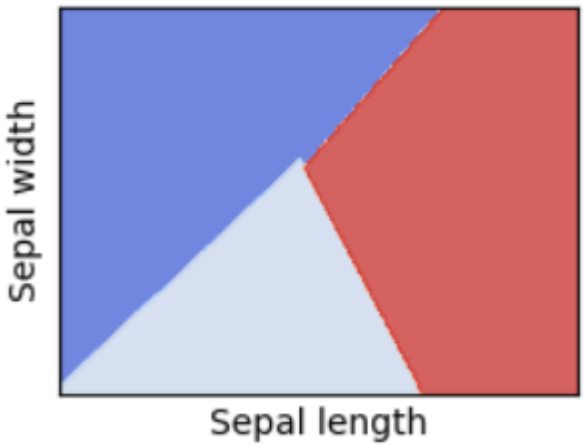


> 将特征点与decision boundary拟合，可以发现决策树对于历史数据的依赖性存在过拟合的问题。

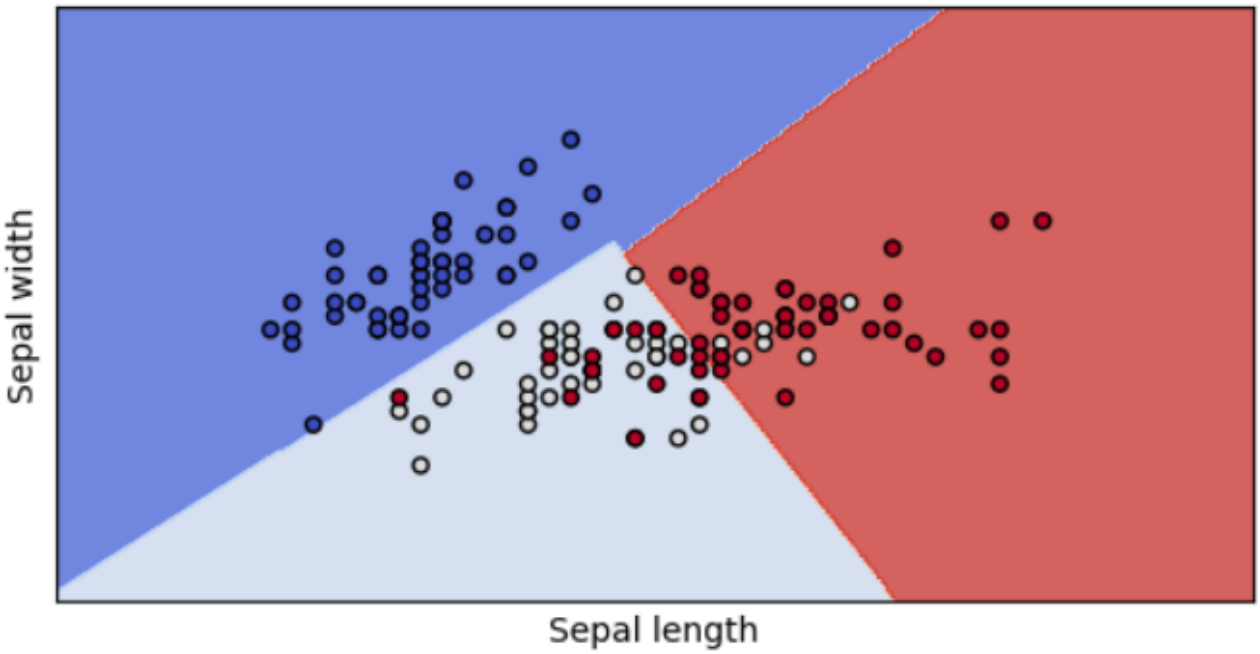


SVM模型

> 用SVM训练出了decision boundary 如下图所示，通过训练结果会发现，拥有线性内核的SVM分类器的boundary具有普适性，虽然对于训练样本的正确率不高，但是由于其特性，能在三类鸢尾花中划清大致的界限。



> 将特征点与decision boundary拟合。



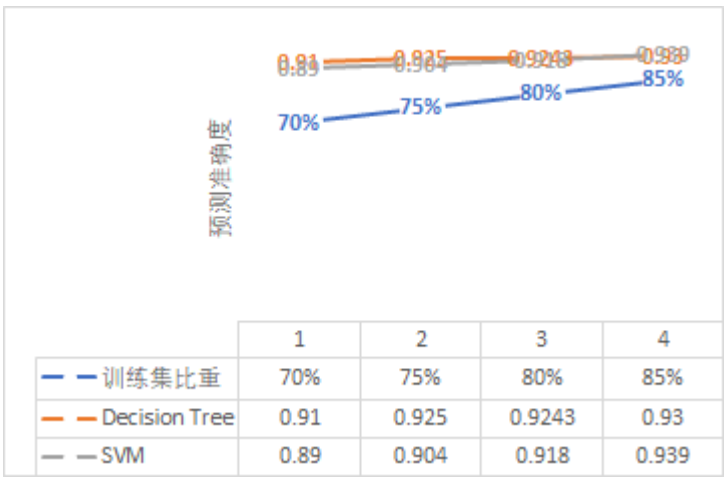
两种监督学习的算法训练出来的模型已经确立。下面使用test集去判断他们的优劣。

实验分析

根据上一部分训练出来的decision boundary的模型，我么得到以下的结果

	Decision Tree	SVM
平均准确率	0.91	0.89

上述结果是在训练集为原训练集的70%数据量基础上的，后面我将数据集的比重调整进行了结果比较。



> 蓝色线条表示训练集占数据的总比值，橙色表示决策树的准确率，灰色线表示SVM的分类结果

个人觉得实验结果符合原理与自己的预期，对于低数据量的工作而言，对于数据分布而言，SVM的预测准确性总体上是比决策树好的，因为决策树如图所示会达到一个阈值，由于依赖历史数据，其训练出来的模型确实存在过拟合的问题，也确实影响了预测的准确性。

对于SVM来讲，应该是能达到一个较好的分类效果，只是这个数据集中样本数量过小，相比较决策树来讲其分类界面没有过于准确，若数据集数量大的话，其分类效果会高于决策树算法。

实验代码

见附件