**实验三 SVM**

**小组成员：**计科 1802 张继伟（201808010829）

计科 1802 谢正宇（201808010824）

计科 1801 樊佳婷（201808010816）

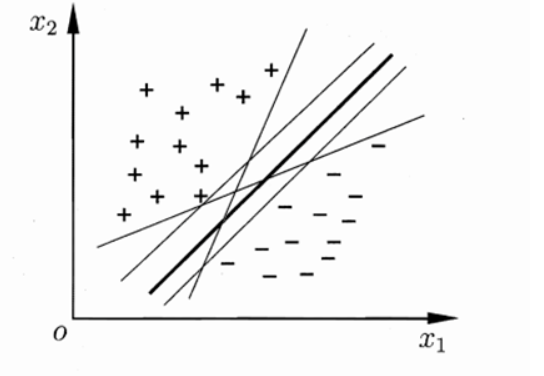
计科 1801 刘怡聪（201808010813）

计科 1801 孙晶铭（201808010808）

**实验完成日期：**2020 年 12 月 04 日

1. **实验描述**
2. 实验原理：**SV**M--支持向量机

给定训练样本集 D ={ (X1 ,Yl)，（X2，Y2)..., (Xm,Ym)}, Yi ∈{-1,+1}，分类学习最基本的想法就是基于训练集D在样本空间中找到一个划分超平面、将不同类别的样本分开。



原则上，是去找位于两类训练样本"正中间"的划分超平面，即图 6.1中黑色最粗的那个，因为该划分超平面对训练样本局部扰动的"容忍性“最好，鲁棒性最强，泛化能力最强。

例如，由于训练集的局限性或噪声的因素，训练集外的样本可能比图6.1中的训练样本更接近两个类的分隔界，这将使许多划分超平面出现错误，而黑色最粗的那个超平面受影响最小。换言之，这个划分超平面所产生的分类结果是最鲁棒的，对未见示例的泛化能力最强。

在样本空间中，划分超平面可通过如下线性方程来描述:;(1)

x=(x1,x2…xd)为输入数据，维度为d。

w=()为法向量，决定了超平面的方向。

 b为位移项，决定了超平面与原点之间的距离。

显然，分类超平面可被法向量ω和位移b确定，下面我们将其记为(w,b)。

样本空间中任意点 x到超平面(w,b)的距离可写为：

r=；（2）

假设超平面(w,b)能将训练样本正确分类，即对于，

若;

若;

所以，令：

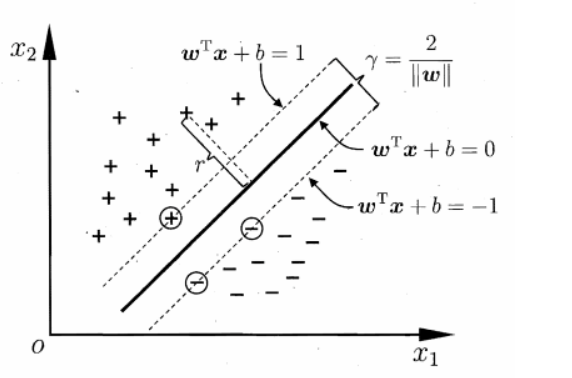
;

;(3)

如图所示，距离超平面最近的这几个训练样本点使(3)的等号成立，它们被称为"支持向量"，两个异类支持向量到超平面的距离之和为：

r=;（4）

其中，γ被称为"间隔"。



欲找到具有"最大间隔" (maximum margin) 的划分超平面，也就是要找到能满足式(6.3)中约束的参数w和b，使得γ最大，即：

max;

s.t. ,i=1,2,…,m;(5)

显然，为了最大化间隔γ，仅需最小化||w||。

于是，式(5)可重写为：

min

s.t. ,i=1,2,…,m;(6)

**上述（6）是支持向量机的基本型**。

1. 数据集处理

Iris.data的数据格式如下：共5列，前4列为样本特征，第5列为类别，分别有三种类别Iris-setosa， Iris-versicolor， Iris-virginica。

注意：因为在分类中类别标签必须为数字量，所以应将Iris.data中的第5列的类别（字符串）转换为num。

1. **实验及结果分析**
2. 开发语言及运行环境；

Visual Studio 2019及anaconda

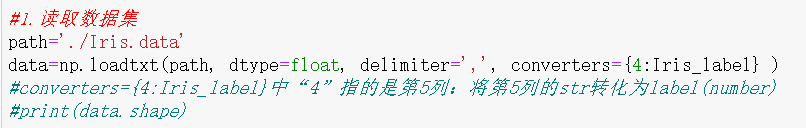
1. 实验的具体步骤；

**导入SVM模块**

首先在使用SVM时，需先从sklearn包中导入SVM模块。



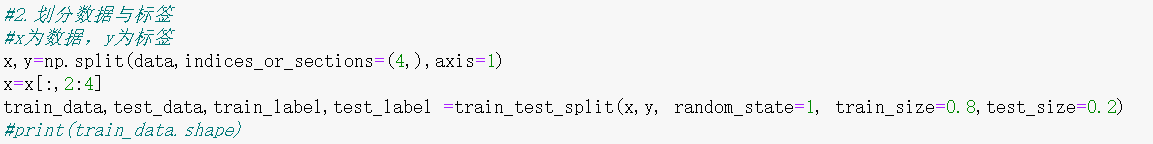
**读取数据集**

**定义的转换函数**

可实现将类别Iris-setosa,Iris-versicolor,Iris-virginica映射成0,1,2。



**划分训练样本与测试样本**



split(数据，分割位置，轴=1（水平分割） or 0（垂直分割）)。

sklearn.model\_selection.train\_test\_split随机划分训练集与测试集。train\_test\_split(train\_data,train\_label,test\_size=数字, random\_state=0)

参数解释：

　    　train\_data：所要划分的样本特征集

　    　train\_label：所要划分的样本类别

　    　test\_size：样本占比，如果是整数的话就是样本的数量

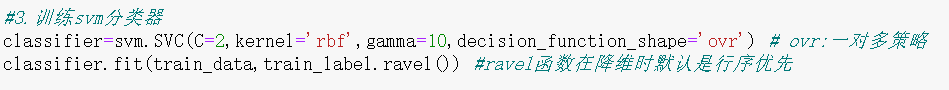
**注意：**

--  test\_size:测试样本占比。 默认情况下，该值设置为0.25。 默认值将在版本0.21中更改。 只有train\_size没有指定时，它将保持0.25，否则它将补充指定的train\_size，例如train\_size=0.6,则test\_size默认为0.4。

-- train\_size:训练样本占比。

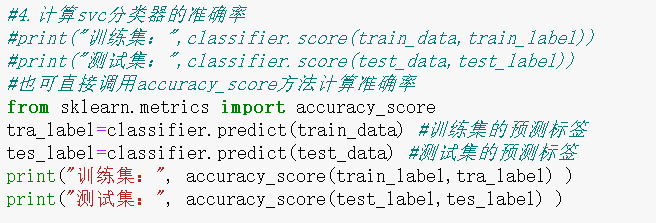
random\_state：是随机数的种子。随机数种子：其实就是该组随机数的编号，在需要重复试验的时候，保证得到一组一样的随机数。比如你每次都填1，其他参数一样的情况下你得到的随机数组是一样的。但填0或不填，每次都会不一样。随机数的产生取决于种子，随机数和种子之间的关系遵从以下两个规则：种子不同，产生不同的随机数；种子相同，即使实例不同也产生相同的随机数。

**训练SVM分类器**

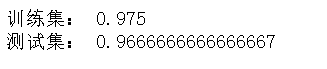
kernel='linear'时，为线性核，C越大分类效果越好，但有可能会过拟合（defaul C=1）。kernel='rbf'时（default），为高斯核，gamma值越小，分类界面越连续；gamma值越大，分类界面越“散”，分类效果越好，但有可能会过拟合。

decision\_function\_shape='ovr'时，为one v rest（一对多），即一个类别与其他类别进行划分，decision\_function\_shape='ovo'时，为one v one（一对一），即将类别两两之间进行划分，用二分类的方法模拟多分类的结果。

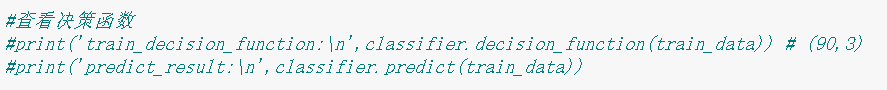
**计算分类准确率**



结果：



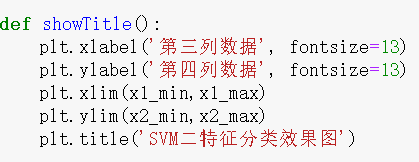
**查看决策函数**

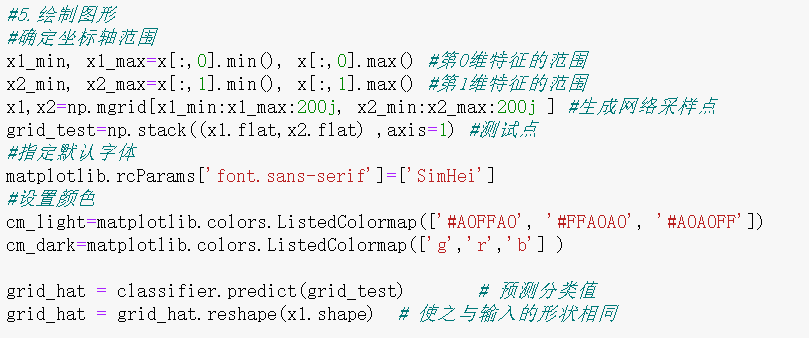


**绘制图形**

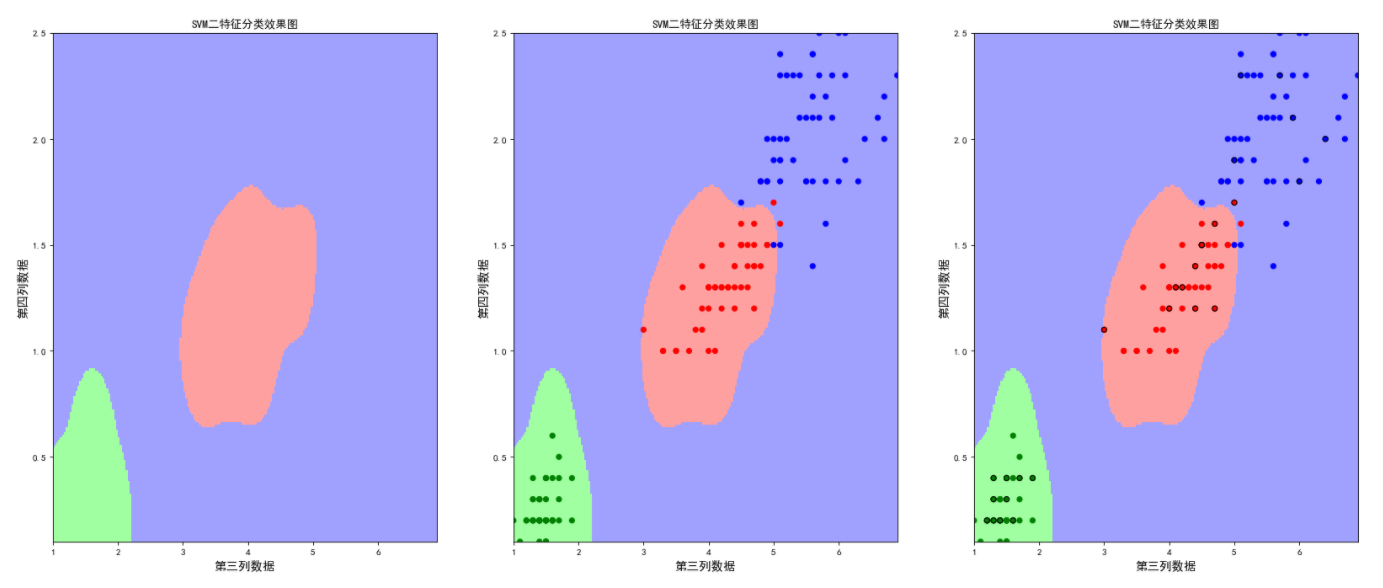
根据第三列和第四列数据绘制分类图形（二维平面图形）

确定坐标轴范围、字体、背景颜色





1. 根据实验数据集，按实验要求给出相应的结果（截图）；



1. 对实验结果进行简要分析。

通过图形表示来看，整体训练的效果良好，无论是训练集还是测试集，训练结果的准确率都比较高。

1. **实验心得**

**计科 1802 张继伟（201808010829）：**

支持向量机（support vector machines, SVM）是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机；SVM还包括核技巧，这使它成为实质上的非线性分类器。SVM的的学习策略就是间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。SVM的的学习算法就是求解凸二次规划的最优化算法。通过本次实验，我基本明白了支持向量机的基本原理，通过自我动手对机器学习充满强烈兴趣，虽然实验中遇到许多困难，但是通过博客查询等网络手段还是可以有效解决问题。

**计科 1802 谢正宇（201808010824）：**

本次实验是一个支持向量机（SVM）的实验。老师上课的听的稀里糊涂，在做了这个实验之后感觉明白了许多。有了svm这一部分python有可以直接用的第三方库，所以我们重点设置合适的参数。首先是核函数的选取。当kernel=&apos;linear&apos;时，为线性核，C越大分类效果越好，但有可能会过拟合（defaul C=1）。而当kernel=&apos;rbf&apos;时（default），为高斯核，gamma值越小，分类界面越连续；gamma值越大，分类界面越“散”，分类效果越好，但有可能会过拟合。我们最终选取的是高斯核。其次关于决策函数对应关系问题，当decision\_function\_shape=&apos;ovr&apos;时，为one v rest（一对多），即一个类别与其他类别进行划分，而当decision\_function\_shape=&apos;ovo&apos;时，为one v one（一对一），即将类别两两之间进行划分，用二分类的方法模拟多分类的结果。最后数据可视化部分也是用的python的numpy库。总的来说，本次实验让我更深入的理解了核函数和支持向量机。

**计科 1801 樊佳婷（201808010816）：**

SVM 是一个非常优雅的算法，具有完善的数学理论。SVM（Support Vector Machines）——支持向量机是在所有知名的数据挖掘算法中最健壮，最准确的方法之一，它属于二分类算法，可以支持线性和非线性的分类。本次实验学习策略便是SVM间隔最大化，最终可转化为一个凸二次规划问题的求解。核函数的选取，kernel=&apos;linear&apos;时，为线性核，C越大分类效果越好，但有可能会过拟合（defaul C=1）。而当kernel=&apos;rbf&apos;时（default），为高斯核，gamma值越小，分类界面越连续；gamma值越大，分类界面越“散”，分类效果越好，但有可能会过拟合。对训练数据和测试数据的标记，在训练和预测的时候，数据的组成部分里是一直都有标记的，对每一组数据进行标记，而不是想当然的对某一行或者某一列进行标记，标记矩阵的维数要和输入的训练数据和测试数据的维数相同，要不然就不能对所有的样本进行标记。

**计科 1801 刘怡聪（201808010813）：**

本次机器学习实验我负责代码的编写，因为Python中的sklearn库也集成了SVM算法，所以在Python中一样可以使用支持向量机做分类。熟悉了利用SVM进行数据分类的代码编写过程，从导入数据，编写转换函数，训练SVM模型和根据测试准确率进行调参，并通过可视化展示来分析代码可改进的地方。SVM的核心是寻找对训练样本局部扰动的"容忍性“最好，鲁棒性最强，泛化能力最强的划分超平面，所以我们修改SVC函数的参数，最终选择高斯核，一对多策略进行训练，最后得到一个比较高的准确率。在支持向量机的算法中，也涉及到了对偶问题、拉格朗日乘子法、核函数、软间隔和正则化等，也学习到了很多知识。

**计科 1801 孙晶铭（201808010808）：**

SVM学习:  
SVM本身是一种典型的二分类器，  
常用的有三种方法：  
1.一对多  
也就是“一对其余”的方式，就是每次仍然解一个两类分类的问题。这样对于n个样本会得到n个分类器。但是这种方式可能会出现分类重叠现象或者不可分类现象，而且由于“其余”的数据集过大，这样其实就人为造成了“数据偏斜”的问题  
2.一对一  
每次选择一个类作为正样本，负样本只用选其余的一个类，这样就避免了数据偏斜的问题。  
很明显可以看出这种方法训练出的分类个数是k\*(k-1)/2，虽然分类器的个数比上面多了，但是训练阶段所用的总时间却比“一类对其余”方法少很多。  
这种方法可能使多个分类器指向同一个类别，所以可以采用“投票”的方式确定哪个类别：哪个分类器获得的票数多就是哪个分类器。  
这种方式也会有分类重叠的现象，但是不会有不可分类的情况，因为不可能所有类别的票数都是0。  
但是也很容易发现这种方法是分类器的数目呈平方级上升。  
3.DAG SVM  
假设有1、2、3、4、5五个类，那么可以按照如下方式训练分类器（ 这是一个有向无环图，因此这种方法也叫做DAG SVM）  
这种方式减少了分类器的数量，分类速度飞快，而且也没有分类重叠和不可分类现象。  
但是假如一开始的分类器回答错误，那么后面的分类器没有办法纠正，错误会一直向下累积。  
为了减少这种错误累积，根节点的选取至关重要。

1. **程序文件名的清单**

lab3.ipynb

machineLearing1.py

iris.data