hw1实验报告

52205903005 娄泽华

**hw1要求：**

UCI数据集: <http://archive.ics.uci.edu/ml/index.php>

任选一个数据集

任选一种ML算法：逻辑回归、决策树、神经网络、SVM等

源码+实验报告

交给助教

Deadline: 学期末考试前

# 一、选取数据集

选取模式识别文献中最著名的数据库：鸢尾花数据集（<http://archive.ics.uci.edu/ml/datasets/Iris>）。数据信息如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Set Characteristics:** | Multivariate | **Number of Instances:** | 150 | **Area:** | Life |
| **Attribute Characteristics:** | Real | **Number of Attributes:** | 4 | **Date Donated** | 1988-07-01 |
| **Associated Tasks:** | Classification | **Missing Values?** | No | **Number of Web Hits:** | 3764434 |

数据集包含3个类，每个类有50个实例，其中每个类表示一种鸢尾植物。一类与另两类是线性可分的；后者的两个类别不是线性可分的。

Attribute Information:

1. sepal length in cm

2. sepal width in cm

3. petal length in cm

4. petal width in cm

5. class:

-- Iris Setosa

-- Iris Versicolour

-- Iris Virginica

# 二、实验思路

采用SVM分类器分类，训练集与测试集的比例为7:3，SVM参数为：惩罚因子C=10.0，径向基（高斯）核函数kernel='rbf'，核函数系数gamma=0.10，训练停止误差tol=1e-3。

# 三、代码实现

## 3.1 预处理

4个列名（attributes）与索引的互相转换：索引转列名cn\_ind2type(ind)及列名转索引cn\_type2ind(t)；3种标记类型(class)与索引的互相转换：索引转类型名it\_ind2type (ind)及类型名转索引it\_type2ind (t)。

def cn\_ind2type(ind):  
 return col\_name[ind]  
  
  
def cn\_type2ind(t):  
 return col\_name\_indicator[t]  
  
  
def it\_ind2type(ind):  
 return iris\_type[ind]  
  
  
def it\_type2ind(t):  
 return iris\_type\_indicator[t]

读入数据，并将最后一列的标记转换为索引：

iris\_data = pd.read\_csv(iris\_path, delimiter=',', header=None, converters={4: it\_type2ind})

数据集分隔，测试集占比30%：

X\_train, X\_test, y\_train, y\_test = train\_test\_split(  
 X, y, test\_size=0.3, random\_state=0)

标准化：

sc = StandardScaler()  
sc.fit(X\_train)  
X\_train\_std = sc.transform(X\_train)  
X\_test\_std = sc.transform(X\_test)

展示三类样本的分布情况（选取其中2个维度）：

plt.scatter(X[y == 0, 0], X[y == 0, 1], color='r', marker='\*')  
plt.scatter(X[y == 1, 0], X[y == 1, 1], color='g', marker='s')  
plt.scatter(X[y == 2, 0], X[y == 2, 1], color='b', marker='v')

## 3.2训练

svm\_clf = svm.SVC(C=10.0, kernel='rbf', gamma=0.10, tol=1e-3)  
svm\_clf.fit(X\_train\_std, y\_train)

## 3.3预测

print\_accuracy(svm\_clf, X\_train\_std, y\_train, X\_test\_std, y\_test)

其中print\_accuracy按格式输出准确率，show\_accuracy计算预测结果的准确率。

def show\_accuracy(a, b, tip):  
 acc = a.ravel() == b.ravel()  
 print('%s accuracy: %.5f' % (tip, np.mean(acc)))  
  
  
def print\_accuracy(clf, x\_train, y\_train, x\_test, y\_test):  
 print('training score: %.5f' % (clf.score(x\_train, y\_train)))  
 print('testing score: %.5f' % (clf.score(x\_test, y\_test)))  
 show\_accuracy(clf.predict(x\_train), y\_train, 'training data')  
 show\_accuracy(clf.predict(x\_test), y\_test, 'testing data')

# 四、实验结果

可视化原始数据，选取轴为属性[2, 3]，即花瓣长度、花瓣宽度两个特征，可将数据集尽可能地分开。

col\_name = {  
 0: 'sepal length',  
 1: 'sepal width',  
 2: 'petal length',  
 3: 'petal width',  
 4: 'class',  
}



svm分类后的对比图



SVM分类在训练集和测试集上的准确率分别为：训练集上0.97143；测试集上0.97778。



# 五、实验心得

本次实验中，我在编程过程中锻炼了自己的数据预处理能力以及数据的可视化分析能力，并通过手动实现一些基本的数据计算函数，体会到了集成性较强库函数的功能也是完全可以根据公式与逻辑逐步实现的。我对SVM的一些参数有了新的理解，比如高斯核、训练停止误差等。此次实验的难点主要在于对数据的预处理及对SVM的理解上。