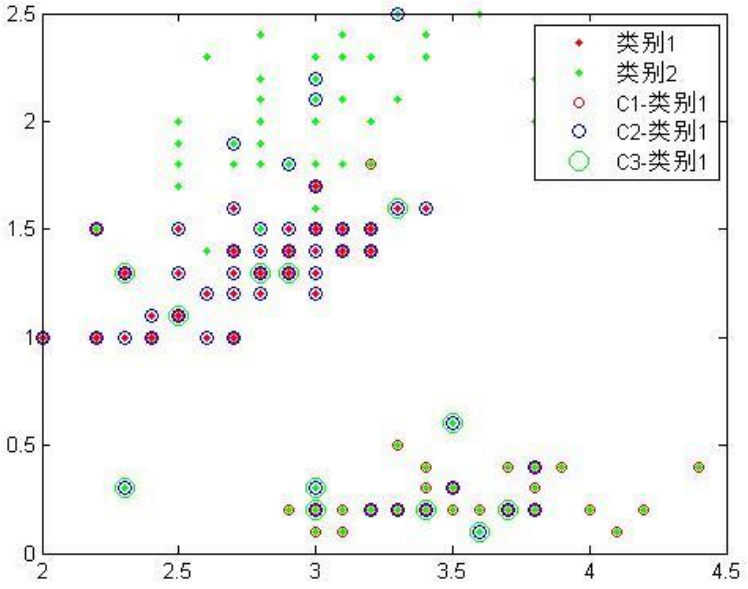
1）Boosting 算法 其一

代码见 T3_KNN.m

代码里用的 KNN 作弱分类器 C1,C2,C3。其中一次的结果如下：

代码用 45 组数据作为训练集，105 组数据作为测试集。可以看出，C1 的正确率为 55.24%，C2 正确率为 67.62%，C3 正确率为 76.19%,都属于弱分类器。在经过 Boosting 算法优化后变为 94.29%的强分类器。



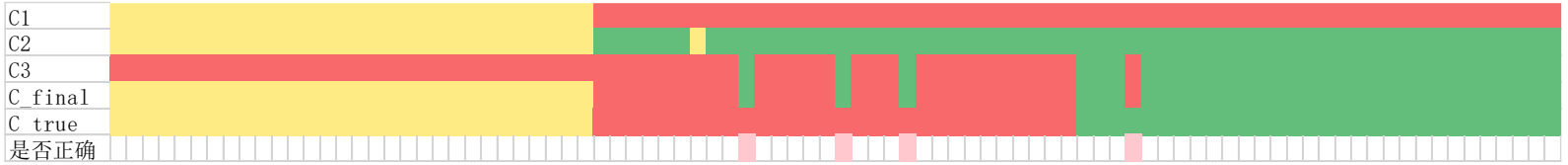

```
>> T3_KNN

ans =

    0.5524

ans =

    0.6762

ans =

    0.7619

ans =

    0.9429
```

2）Boosting 算法 其二

代码见 T3_KNN_3.m

与上面不同的是，我们在此不简化类别，而是用原来的 3 种植物类别，分别为'setosa','versicolor','virginica'。代码里用的 KNN 作弱分类器 C1,C2,C3。其中一次的结果如下：代码用 60 组数据作为训练集，90 组数据作为测试集。可以看出，C1 的正确率为 66.67%，C2 正确率为 66.67%，C3 正确率为 62.22%，都属于弱分类器。在经过 Boosting 算法优化后变为 95.56%的强分类器。所有的 90 组测试数据如右图所示，可以看出，只有 4 组数据分类错误。

```
ans =

    0.6667


ans =

    0.6667


ans =

    0.6222


ans =

    0.9556
```



3) Adaboost 分类算法

代码见 Adaboost_1.m

Step1:代码先是导入数据，并且分出训练集和测试集(各 75 组数据)；

Step2:初始化权重 w

Step3:在迭代中训练若干弱分类器：选择训练集中权重最大的数据点以及随机选择的其他若干个数据点多次训练分类器，比较e(t)大小，选择其中e(t)最小的分类器`svmStruct_j(t)`。直到融合这些弱分类器的强分类器对训练集的识别率P(t)高于0.9时停止。

Step4:最后，利用前面迭代得出的若干弱分类器，对测试集进行测试

最后输出的结果为：

P:对训练集的识别率

per:若干个独立弱分类器的识别率

PER:Adaboost 方法，最终得出的识别率

其中一次的结果如下：

```
P =
    0.6800    0.6267    0.6800    0.7867    0.9200
per =
    0.5733    0.6933    0.5333    0.6667    0.6400
PER =
    0.9200
```

其中另一次的结果如下：

```
P =
    0.6933    0.6133    0.6400    0.7467    0.6667    0.7467    0.8400    0.8667    0.9067
per =
    0.5733    0.7067    0.5600    0.6800    0.5467    0.5733    0.6267    0.5733    0.6533
PER =
    0.9200
```

调试中遇到的主要错误：

最开始，本代码只在第一次迭代时对权重进行了归一化。使得最后的权重矩阵 w(t,:)全为 0。这是个致命错误，会造成无法正常更新权重。

最开始发现错误后，将归一化写成了

```
for i=1:length(groups_train)
    w(t,i)=w(t,i)/ sum(w(t,:));% normalize the weights
end
```

这样写是错误的。因为求和项一直在变，不能实现归一化。因此可以改为：

```
w(t,:)=w(t,:)/sum(w(t,:));% normalize the weights
```