

模式识别导论上机题4-SVM

薛犇 1500012752

1. 程序实现说明

本次实验采用Matlab作为编程语言，使用的版本为2016b。

在实验的一开始，利用importdata函数读取hw4_data.txt中的数据，保存在向量 raw_data中。

```
raw_data = importdata('hw4_data.txt');
```

随后分离出样本x和对应的label: y，需要注意的是，因为是利用SVM处理数据，所以要把0值的y改为-1，这样才满足理论中的假设。

```
x = raw_data(:, 1:3);  
y = raw_data(:, 4);  
y(y==0)=-1;  
[n, d] = size(x);
```

然后设置一些超参数，因为我选择的是RBF核函数，所以除了C之外，还要设置高斯函数的参数 σ ，在之后的SMO算法中，我利用了终止条件三，也即对偶函数的差值作为判断优化算法是否终止的条件，所以这里也需要一个阈值。参数设置如下：

```
threshold = 20;  
C = 20;  
sigma = 20;
```

之后就开始SMO算法，随机挑选两个拉格朗日乘子 α_1, α_2 ,迭代算法如下：

$$E_i = f(x_i) - y_i = \left(\sum_{j=1}^n \alpha_j y_j K(x_i, x_j) + b \right) - y_i, i = 1, 2$$

$$\alpha_2^{new} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{K_{11} + K_{22} - 2K_{12}}$$

$$\alpha_2^{new} = \begin{cases} V, \alpha_2^{new} > V \\ U, \alpha_2^{new} < U \end{cases}$$

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new})$$

其中：

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

在代码实现中， E, K 分别被定义成如下的两个函数：

```
function res = E(a, y, x, i, n, sigma)
    res = 0;
    for j=1:n
        res = res + a(j)*y(j)*K(x(i,:), x(j,:), sigma);
    end
    res = res - y(i);
end

function res = K(xi, xj, sigma)
    res = exp(-0.5*sum((xi-xj).^2)/sigma^2);
end
```

α_2^{new} 的计算如下：

```
a2new = a(j)+y(j)*(E(a,y,x,i,n,sigma)-E(a,y,x,j,n,sigma))/(K(x(i,:),x(i,:),sigma)+K(x(j,:),x(j.
```



U, V 的设置则由如下代码完成：

```
U = 0;
V = C;
if y(i)~=y(j)
    U = max([0, a(j)-a(i)]);
    V = min([C, C-a(i)+a(j)]);
end
if y(i)==y(j)
    U = max([0, a(i)+a(j)-C]);
    V = min([C, a(i)+a(j)]);
end
if a2new > V
    a2new = V;
end
if a2new < U
    a2new = U;
end
```

与此同时，还要设计终止条件，采用对偶函数的差值作为终止条件，也即：

$$J(w) - W(\alpha) = \sum_{i=1}^n \alpha_i - 2W(\alpha) + C \sum_{i=1}^n \xi_i$$

$$\frac{J(w) - W(\alpha)}{J(w) + 1} < T$$

$$\xi_i = \max(0, 1 - y_i(\sum_{j=1}^n \alpha_j y_j K(x_i, x_j) + b))$$

实现的代码如下：

```
% get b first
sum_kernel = 0;
for k=1:n
    sum_kernel = sum_kernel + a(k)*y(k)*K(x(1,:),x(k,:),sigma);
end
b = (1-y(1)*sum_kernel)/y(1);

% then get ksi
ksi = zeros(1,n);
for k=1:n
    sum_kernel = 0;
    for t=1:n
        sum_kernel = sum_kernel + a(t)*y(t)*K(x(k,:),x(t,:),sigma);
    end
    ksi(k)=max([0, 1-y(k)*(sum_kernel+b)]);
end

% get W and J
newW = W(a,x,y,n,sigma)
judge = sum(a)-newW+C*sum(ksi);
if (judge-newW)/(judge+1) < threshold
    break;
end
```

最终收敛后，得到结果，退出优化的循环。

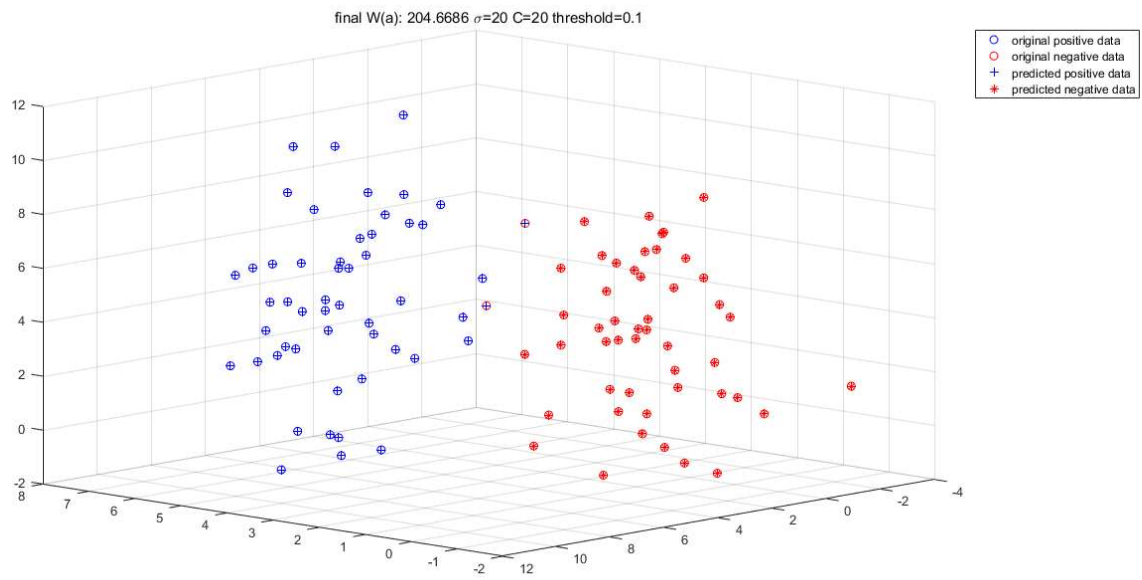
2. 实验结果分析

把得到的分类器作用于原来的样本，prediction的原理如下：

$$pred = w^T x + b = \sum_{j=1}^n \alpha_j y_j K(x, x_j) + b$$

也即，把样本代入核函数即可。

得到的结果如下图所示：



可以看出，分类错了两个原本是负样本的点。准确率为98%