Support Vector Machines

Jing-Mao Ho

1 The Primal Form

We have seen how a perceptron learning algorithm can find a separating hyperlane that dichotomizes the given data. If the data are linearly separable, then a PLA guarantees a hyperplane. Note that theoretically there are infinite separating hyperplanes. Therefore, here arises a question: how do we know, among those candidates, which is the best one? First of all, we should define what 'best" means here. By "best," people usually mean the hyperplane that neither overfits nor underfits the given set of data. In the case of linear classification, the best hyperplane should be the one that has the largest distance to the closest data points of both groups. We call the closet data points to the separating hyperplane *support vectors*.

Next, let's specify the data and the notation. Assume

$$\mathbf{X}_{n\times k}=\begin{bmatrix} x_{01} & x_{11} & x_{11} & \cdots & x_{k1} \\ x_{02} & x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{0n} & x_{1n} & x_{2n} & \cdots & x_{kn} \end{bmatrix}, \widehat{\boldsymbol{\beta}}_{k\times 1}=\begin{bmatrix} \widehat{\beta}_1 \\ \widehat{\beta}_2 \\ \vdots \\ \widehat{\beta}_k \end{bmatrix}, \widehat{\boldsymbol{\beta}}_{\mathbf{o}}=[\widehat{\boldsymbol{\beta}}_{\mathbf{o}}], \mathbf{Y}_{n\times 1}=\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Since the goal of a support vector machine is to find the best hyperplane, the idea is built on the perceptron learning algorithm. Recall that a perceptron, $\widehat{f(x_i)}=\widehat{y_i}=\text{sign}\left(\mathbf{X}\cdot\widehat{\boldsymbol{\beta}}_{\text{PLA}}+\widehat{\boldsymbol{\beta}}_{0\text{PLA}}\right)$, aims at obtaining $\widehat{\boldsymbol{\beta}}_{\text{PLA}}$ so that $\widehat{y_i}$ is equal to y_i. For an SVM, the goal is to find the separating hyperplane that has the largest "margin." This means that now the goal is to get $\max_{\widehat{\boldsymbol{\beta}}} \text{margin}(\widehat{\boldsymbol{\beta}})$. Then we can derive that

$$\begin{aligned}
\max_{\widehat{\boldsymbol{\beta}}\widehat{\boldsymbol{\beta}}_0} \text{margin}(\widehat{\boldsymbol{\beta}}) &= \max_{\widehat{\boldsymbol{\beta}}\widehat{\boldsymbol{\beta}}_0} \min_{x_i} \left[\text{dist}(x_i, \mathbf{X}\widehat{\boldsymbol{\beta}}+\widehat{\boldsymbol{\beta}}_0)\right] \\
&= \max_{\widehat{\boldsymbol{\beta}}\widehat{\boldsymbol{\beta}}_0} \min_{x_i} \left[\frac{|\mathbf{X}\widehat{\boldsymbol{\beta}}+\widehat{\boldsymbol{\beta}}_0|}{\left\|\widehat{\boldsymbol{\beta}}\right\|}\right] \\
&= \max_{\widehat{\boldsymbol{\beta}}\widehat{\boldsymbol{\beta}}_0} \min_{x_i} \left[\frac{y_i(\mathbf{X}\widehat{\boldsymbol{\beta}}+\widehat{\boldsymbol{\beta}}_0)}{\|\boldsymbol{\beta}\|}\right] \qquad \left(\text{Let } y_i(\mathbf{X}\widehat{\boldsymbol{\beta}}+\widehat{\boldsymbol{\beta}}_0)=1\right) \\
&= \max_{\widehat{\boldsymbol{\beta}}\widehat{\boldsymbol{\beta}}_0} \frac{1}{\left\|\widehat{\boldsymbol{\beta}}\right\|} \qquad \text{subject to } y_i(\mathbf{X}\widehat{\boldsymbol{\beta}}+\widehat{\boldsymbol{\beta}}_0)\geq 1 \\
&= \min_{\widehat{\boldsymbol{\beta}}\widehat{\boldsymbol{\beta}}_0} \left\|\widehat{\boldsymbol{\beta}}\right\|^2 \qquad \text{subject to } y_i(\mathbf{X}\widehat{\boldsymbol{\beta}}+\widehat{\boldsymbol{\beta}}_0)\geq 1
\end{aligned}$$

$$= \min_{\widehat{\boldsymbol{\beta}}\,\widehat{\boldsymbol{\beta}}_0} \frac{1}{2} \left\| \widehat{\boldsymbol{\beta}}_{\text{SVM}} \right\|^2 \qquad \text{subject to } y_i(\mathbf{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0) \geq 1$$
$$= \min_{\widehat{\boldsymbol{\beta}}\,\widehat{\boldsymbol{\beta}}_0} \frac{1}{2} \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}} \qquad \text{subject to } y_i(\mathbf{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0) \geq 1$$

The primal form of the support vector machine is $\min_{\widehat{\boldsymbol{\beta}}\,\widehat{\boldsymbol{\beta}}_0} \frac{1}{2} \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}}$ (subject to $y_i(\mathbf{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0) \geq 1$) because this is an optimization problem that can be solved by a quadratic programming (QP) algorithm. The general form of a QP problem is:

$$\min_{\boldsymbol{\omega}} \frac{1}{2} \boldsymbol{\omega}^T \mathbf{D}\, \boldsymbol{\omega} + \mathbf{W}^T \boldsymbol{\omega}$$
$$\boldsymbol{\omega} \in \mathbf{R}^n$$
$$\text{subject to } \mathbf{A}\boldsymbol{\omega} \geq z$$

Given this form of the quadratic programming problem, we next transform $\min_{\widehat{\boldsymbol{\beta}}\,\widehat{\boldsymbol{\beta}}_0} \frac{1}{2} \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}}$ (subject to $y_i(\mathbf{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0) \geq 1$) into a QP problem. So let

$$\boldsymbol{\omega} = \begin{bmatrix} \widehat{\boldsymbol{\beta}}_0 \\ \widehat{\boldsymbol{\beta}} \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \mathbf{W} = \mathbf{0}$$
$$\mathbf{A} = y_i \cdot \begin{bmatrix} 1 & \mathbf{X} \end{bmatrix}, z = 1$$

Finally, solving the QP problem will obtain the minimized ω. This helps us get $\widehat{\boldsymbol{\beta}}_{\text{SVM}}$.

2 The Dual Form

To solve $\min_{\widehat{\boldsymbol{\beta}}\,\widehat{\boldsymbol{\beta}}_0} \frac{1}{2} \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}}$ (subject to $t_i(\mathbf{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0) \geq 1$), one often needs to transform the feature space $\mathbf{X} \in \mathbf{R}^k$ into higher dimensional space $\Phi(\mathbf{X}) = \mathbf{Z} \in \mathbf{R}^m$. This transformation is to facilitate an SVM to obtain $\widehat{\boldsymbol{\beta}}_{\text{SVM}}$ because data might not be able to separated in lower dimensional space. However, this transformation can increase the computational complexity in that new dimension m might be much greater than the original dimension k. Therefore, if we can make the QP problem independent of the dimensionality, then the computation will be simpler. To do so, we need to transform the original QP problem (*primal form*) into another, which is the *dual form* QP problem.
The new problem now is $\min_{\widehat{\boldsymbol{\beta}}\,\widehat{\boldsymbol{\beta}}_0} \frac{1}{2} \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}}$ (subject to $t_i(\mathbf{Z}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0) \geq 1$) Next, we need to conduct several transformation. First of all, define a *Lagrange function* $\mathcal{L}$ accompanied by a *Lagrange multiplier* α:

$$\mathcal{L}(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\beta}}_0, \boldsymbol{\alpha}) = \frac{1}{2} \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}} - \left[\sum_{i=1}^{k} \alpha_i \cdot y_i(\mathbf{Z}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0)) \right]$$

Second, we can derive that

$$\begin{aligned}\min_{\widehat{\boldsymbol{\beta}}\,\widehat{\beta}_0} \frac{1}{2}\widehat{\boldsymbol{\beta}}^T\widehat{\boldsymbol{\beta}} &= \min_{\widehat{\boldsymbol{\beta}}\,\widehat{\beta}_0} \left(\max_{\alpha_i \geq 0} \mathcal{L}(\widehat{\boldsymbol{\beta}}, \widehat{\beta}_0, \boldsymbol{\alpha})\right) \\ &\geq \min_{\widehat{\boldsymbol{\beta}}\,\widehat{\beta}_0} \left(\mathcal{L}(\widehat{\boldsymbol{\beta}}, \widehat{\beta}_0, \boldsymbol{\alpha})\right) \\ &\geq \max_{\alpha_i \geq 0} \left(\min_{\widehat{\boldsymbol{\beta}}\,\widehat{\beta}_0} \mathcal{L}(\widehat{\boldsymbol{\beta}}, \widehat{\beta}_0, \boldsymbol{\alpha})\right)\end{aligned}$$

Third, because we are to minimize the Lagrange function, it's useful to get the first order condition. We take the partial derivative of $\mathcal{L}$ with respect to $\widehat{\beta}_0$:

$$\begin{aligned}&\mathcal{L}(\widehat{\boldsymbol{\beta}}, \widehat{\beta}_0, \boldsymbol{\alpha})\widehat{\beta}_0 = 0 \\ \Rightarrow &-\sum_{i=1}^{n} \alpha_i \cdot y_i = 0 \\ \Rightarrow &\sum_{i=1}^{n} \alpha_i \cdot y_i = 0\end{aligned}$$

Based on this result, we can simplify the equation $\max_{\alpha_i \geq 0} \left(\min_{\widehat{\boldsymbol{\beta}}\,\widehat{\beta}_0} \mathcal{L}(\widehat{\boldsymbol{\beta}}, \widehat{\beta}_0, \boldsymbol{\alpha})\right)$:

$$\begin{aligned}&\max_{\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i \cdot y_i = 0} \left(\min_{\widehat{\boldsymbol{\beta}}\,\widehat{\beta}_0} \mathcal{L}(\widehat{\boldsymbol{\beta}}, \widehat{\beta}_0, \boldsymbol{\alpha})\right) \\ =&\max_{\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i \cdot y_i = 0} \left[\frac{1}{2}\widehat{\boldsymbol{\beta}}^T\widehat{\boldsymbol{\beta}} - \left(\sum_{i=1}^{k} \alpha_i \cdot y_i(\mathbf{Z}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0)\right)\right] \\ =&\max_{\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i \cdot y_i = 0} \left[\frac{1}{2}\widehat{\boldsymbol{\beta}}^T\widehat{\boldsymbol{\beta}} + \left(\sum_{i=1}^{k} \alpha_i \cdot \left(1 - (y_i(\mathbf{Z}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0))\right)\right)\right] \\ =&\max_{\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i \cdot y_i = 0} \left[\frac{1}{2}\widehat{\boldsymbol{\beta}}^T\widehat{\boldsymbol{\beta}} + \sum_{i=1}^{k} \alpha_i \cdot \left(1 - (y_i(\mathbf{Z}\widehat{\boldsymbol{\beta}})\right) - \underbrace{\sum_{i=1}^{k} \alpha_i y_i}_{0} \cdot \widehat{\boldsymbol{\beta}}_0\right] \\ =&\max_{\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i \cdot y_i = 0} \left[\frac{1}{2}\widehat{\boldsymbol{\beta}}^T\widehat{\boldsymbol{\beta}} + \sum_{i=1}^{k} \alpha_i \cdot \left(1 - (y_i(\mathbf{Z}\widehat{\boldsymbol{\beta}})\right)\right]\end{aligned}$$

Fourth, take the partial derivative of $\mathcal{L}$ with respect to $\widehat{\boldsymbol{\beta}}$:

$$\begin{aligned}
&\mathcal{L}(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\beta}}_0, \boldsymbol{\alpha})\widehat{\beta} = 0 \\
\Rightarrow &\widehat{\beta} - \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \mathbf{Z} = 0 \\
\Rightarrow &\sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \mathbf{Z} = \widehat{\boldsymbol{\beta}}
\end{aligned}$$

Then we can make use of this result to simplify the equation $\max_{\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i \cdot y_i = 0} \left[\frac{1}{2} \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}} + \sum_{i=1}^{k} \alpha_i \cdot \left(1 - (y_i(\mathbf{Z}\widehat{\boldsymbol{\beta}}) \right) \right]$:

$$\begin{aligned}
&\max_{\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i \cdot y_i = 0, \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \mathbf{Z} = \widehat{\boldsymbol{\beta}}} \left[\frac{1}{2} \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}} + \sum_{i=1}^{n} \alpha_i \cdot \left(1 - (y_i(\mathbf{Z}\widehat{\boldsymbol{\beta}}) \right) \right] \\
= &\max_{\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i \cdot y_i = 0, \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \mathbf{Z} = \widehat{\boldsymbol{\beta}}} \left[\frac{1}{2} \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}} + \sum_{i=1}^{n} \alpha_i - \underbrace{\sum_{i=1}^{k} \alpha y_i \mathbf{Z}}_{\widehat{\boldsymbol{\beta}}} \widehat{\boldsymbol{\beta}} \right] \\
= &\max_{\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i \cdot y_i = 0, \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \mathbf{Z} = \widehat{\boldsymbol{\beta}}} \left[\frac{1}{2} \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}} + \sum_{i=1}^{n} \alpha_i - \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}} \right] \\
= &\max_{\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i \cdot y_i = 0, \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \mathbf{Z} = \widehat{\boldsymbol{\beta}}} \left[-\frac{1}{2} \widehat{\boldsymbol{\beta}}^T \widehat{\boldsymbol{\beta}} + \sum_{i=1}^{n} \alpha_i \right] \\
= &\max_{\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i \cdot y_i = 0, \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \mathbf{Z} = \widehat{\boldsymbol{\beta}}} \left[-\frac{1}{2} \left\| \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \mathbf{Z} = \widehat{\boldsymbol{\beta}} \right\|^2 + \sum_{i=1}^{n} \alpha_i \right] \\
= &\min_{\alpha} \left[\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{Z}^T \mathbf{Z} - \sum_{i=1}^{n} \alpha_i \right] \\
&\text{subject to} \sum_{i=1}^{n} y_i \alpha_i = 0; \alpha_i \geq 0
\end{aligned}$$

There are KKT conditions to be satisfied:

- Condition for the primal form: $(y_i(\mathbf{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0) \geq 1$
- Condition for the dual form: $\alpha \geq 0$
- First order condition:

$$\begin{aligned}
\sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \mathbf{Z} &= \widehat{\boldsymbol{\beta}} \\
\sum_{i=1}^{n} \alpha_i \cdot y_i &= 0
\end{aligned}$$

- $\alpha(1 - y_i(\mathbf{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0)) = 0$

The KKT conditions are importantly informative. When $\alpha > 0$, the pair $(\mathbf{Z}_i, y_i)$ are on the boundary. Therefore, they are called *support vectors*. Most important, the dual form of the SVM is also as QP problem. Recall that the form of a QP problem is

$$\begin{aligned}
&\min_{\boldsymbol{\omega}} \frac{1}{2}\boldsymbol{\omega}^T \mathbf{D}\, \boldsymbol{\omega} + \mathbf{W}^T \boldsymbol{\omega} \\
&\boldsymbol{\omega} \in \mathbf{R}^n \\
&\text{subject to } \mathbf{A}\boldsymbol{\omega} \geq z
\end{aligned}$$

To solve the dual form, let

$$\begin{aligned}
&\boldsymbol{\omega} = \boldsymbol{\alpha} \\
&\mathbf{D} = y_i y_j \mathbf{Z}^T \mathbf{Z} \\
&\mathbf{W} = \mathbf{-1} \\
&\mathbf{A} = \mathbf{Y} \\
&z = 0
\end{aligned}$$

To sum up,

$$\begin{aligned}
&\min_{\alpha} \left[\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{Z}^T \mathbf{Z} - \sum_{i=1}^{n} \alpha_i \right] \quad \text{subject to} \sum_{i=1}^{n} y_i \alpha_i = 0; \alpha_i \geq 0 \\
&= \min_{\boldsymbol{\alpha}} \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{D}\, \boldsymbol{\alpha} - \boldsymbol{\alpha} \quad \text{subject to } \mathbf{Y}\boldsymbol{\alpha} = 0 \\
&\Rightarrow \widehat{\boldsymbol{\beta}} = \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \mathbf{Z}
\end{aligned}$$

3 Illustrating SVMs in R

I use an R built-in package, "e1071," to illustrate how SVMs work. First, we import the package

```
> library(e1071)
```

Like what I do while explaining the perceptron learning algorithm, the Iris data set will be used in the case SVMs.

```
> data(iris)
> df<-iris
> df$Species<-as.character(df$Species)
> df$Species[df$Species=="setosa"]<-"+1"
> df$Species[df$Species!="+1"]<-"-1"
> df$Species<- as.integer(df$Species)
> df$Species<- as.factor(df$Species)
```

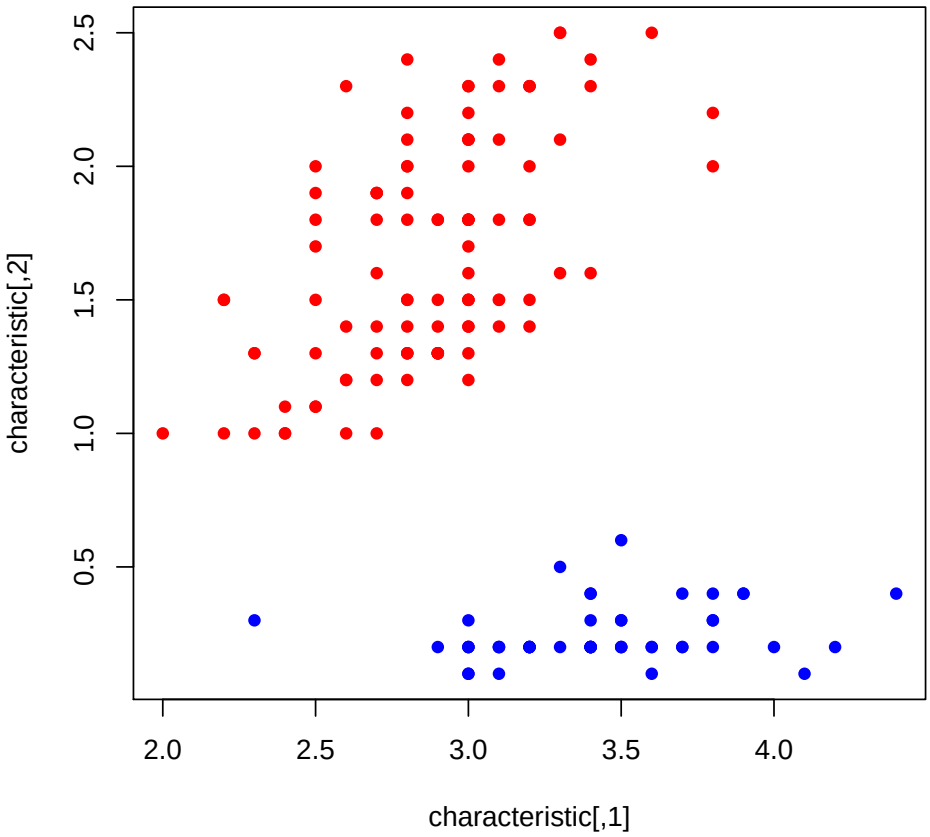
Now we call the function to get the hyperplane:

```
> hyperplane_svm <- svm(Species ~ Sepal.Width + Petal.Width,
+                       data=df, kernel="linear",scale=F)
> summary(hyperplane_svm)
```

```
Call:
svm(formula = Species ~ Sepal.Width + Petal.Width, data = df, kernel = "linear",
    scale = F)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  1
      gamma:  0.5

Number of Support Vectors:  8

 ( 4 4 )


Number of Classes:  2

Levels:
 -1 1
```

Next, we would like to plot the Iris data to get a sense of what it looks like:

```
> characteristic <- cbind(df$Sepal.Width,df$Petal.Width)
> plot(characteristic,cex=0.2)
> points(subset(characteristic,df$Species==1),col="blue",pch=16)
> points(subset(characteristic,df$Species==-1),col="red",pch=16)
```



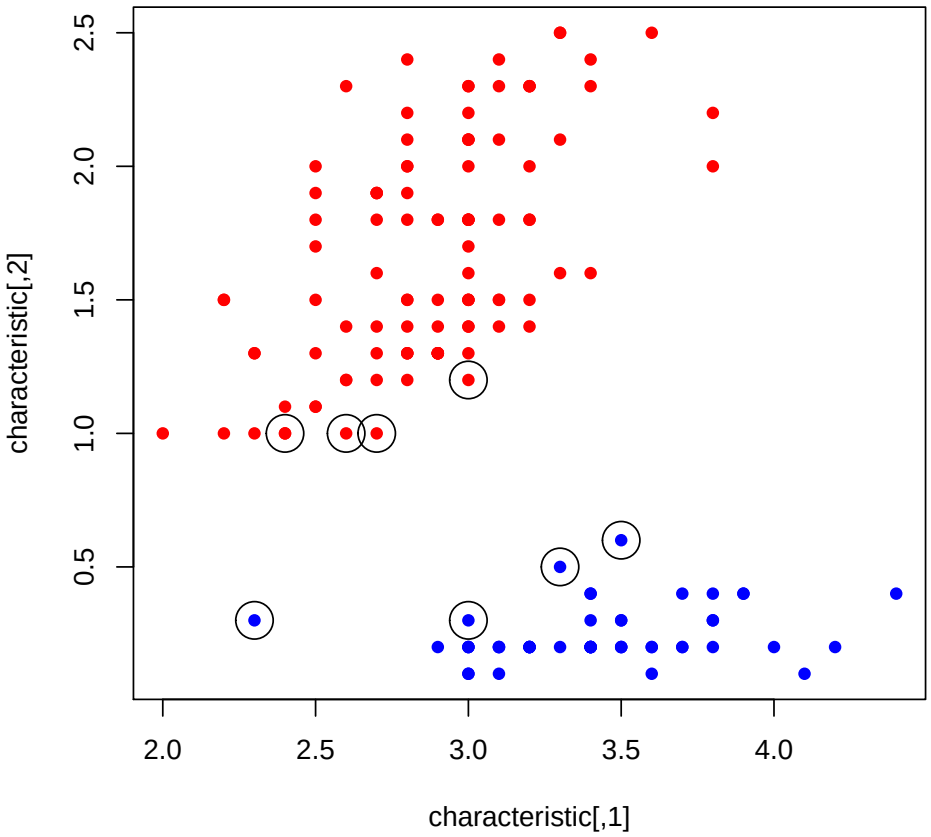
We also would like to know what the support vectors are:

```
> print((rownames(df))[hyperplane_svm$index])
```

```
[1] "24" "42" "44" "46" "68" "80" "82" "96"
```

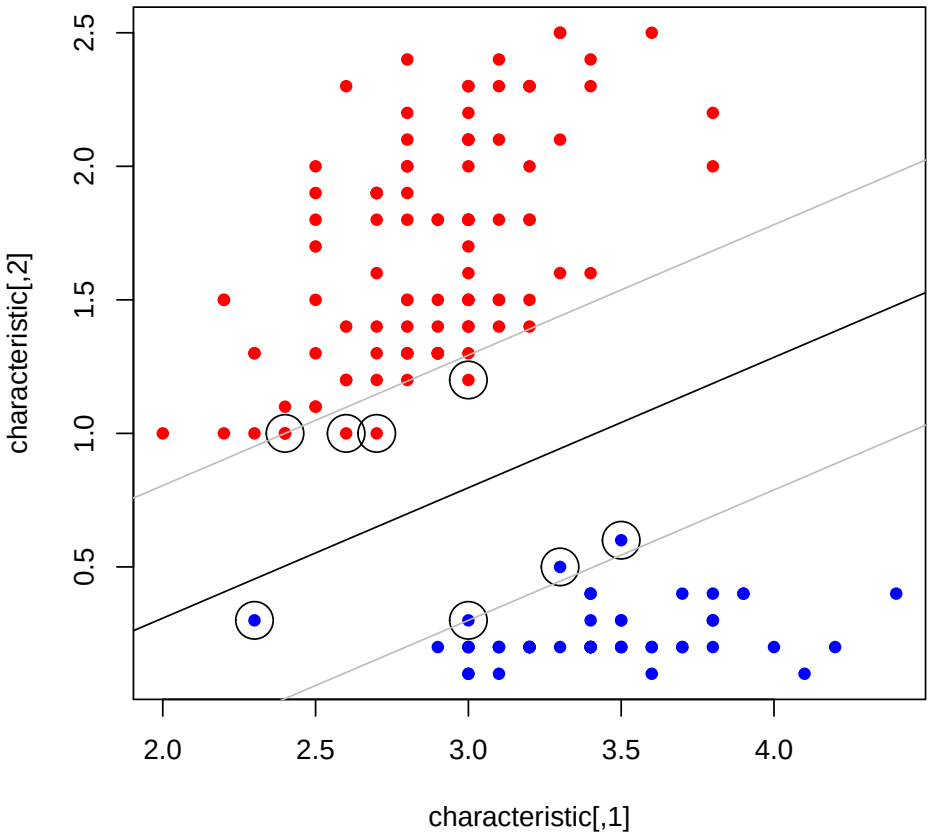
Identify the support vectors:

```
> plot(characteristic,cex=0.2)
> points(subset(characteristic,df$Species==1),col="blue",pch=16)
> points(subset(characteristic,df$Species==-1),col="red",pch=16)
> points(df$Sepal.Width[hyperplane_svm$index],
+        df$Petal.Width[hyperplane_svm$index],cex=3,col=rgb(0,0,0))
```



Finally, we show the hyperplane, which is a line in 2-dimension space.

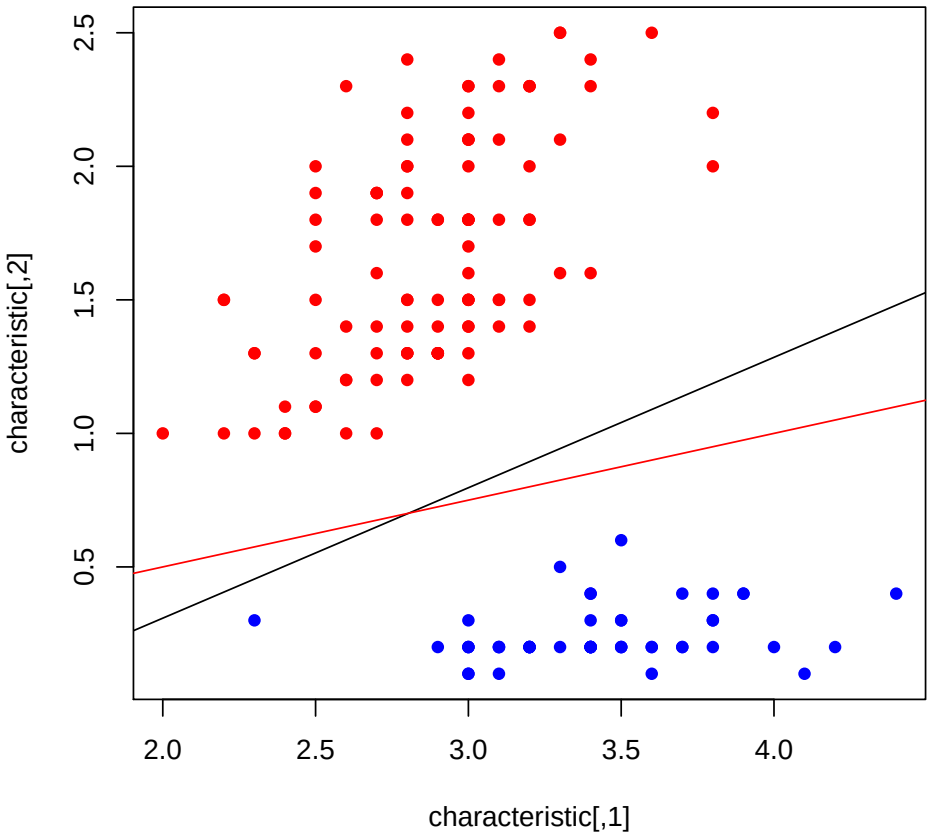
```
> plot(characteristic,cex=0.2)
> points(subset(characteristic,df$Species==1),col="blue",pch=16)
> points(subset(characteristic,df$Species==-1),col="red",pch=16)
> points(df$Sepal.Width[hyperplane_svm$index],
+        df$Petal.Width[hyperplane_svm$index],cex=3,col=rgb(0,0,0))
> beta.0 <- -hyperplane_svm$rho
> beta.1 <- sum(hyperplane_svm$coefs*df$Sepal.Width[hyperplane_svm$index])
> beta.2 <- sum(hyperplane_svm$coefs*df$Petal.Width[hyperplane_svm$index])
> abline(-beta.0/beta.2,-beta.1/beta.2,col="black")
> abline((-beta.0-1.0)/beta.2,-beta.1/beta.2,col="gray")
> abline((-beta.0+1.0)/beta.2,-beta.1/beta.2,col="gray")
```



4 Comparing Perceptrons and SVMS

I would like to compare and contrast the hyperlanes obtained from the perceptron learning algorithm and the support vector machine.

```
> plot(characteristic,cex=0.2)
> points(subset(characteristic,df$Species==1),col="blue",pch=16)
> points(subset(characteristic,df$Species==-1),col="red",pch=16)
> beta.0 <- -hyperplane_svm$rho
> beta.1 <- sum(hyperplane_svm$coefs*df$Sepal.Width[hyperplane_svm$index])
> beta.2 <- sum(hyperplane_svm$coefs*df$Petal.Width[hyperplane_svm$index])
> abline(-beta.0/beta.2,-beta.1/beta.2,col="black")
> abline(0.0/-1.2,-0.3/-1.2,col="red")
```



The red line is the hyperplane generated by the perceptron algorithm, and the black the SVM.