Support Vector Machines

Jing-Mao Ho

1 The Primal Form

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*. The algorithm that is able to identify the support vectors of a given data set in order to find the "best" hyperplane is called *support vector machine*.

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

$$\mathbf{X}_{n\times m} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1m} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \cdots & x_{nm} \end{bmatrix}, \widehat{\boldsymbol{\beta}}_{m\times 1} = \begin{bmatrix} \widehat{\beta}_1 \\ \widehat{\beta}_2 \\ \vdots \\ \widehat{\beta}_m \end{bmatrix}, \widehat{\boldsymbol{\beta}}_0 = [\widehat{\boldsymbol{\beta}}_0], \mathbf{Y}_{n\times 1} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

Since the goal of a support vector machine is to find the best hyperplane, the idea is somewhat similar to, and actually 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 a separating hyperplane that has the largest "margin." This means that now the goal is to get $\max\limits_{\widehat{\boldsymbol{\beta}}} \text{margin}(\widehat{\boldsymbol{\beta}})$. Then we can derive that

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

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

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:

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

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

$$\begin{aligned} &\boldsymbol{\omega}_{(m+1)\times 1} = \begin{bmatrix} \widehat{\boldsymbol{\beta}}_{\mathbf{0}} \\ \widehat{\boldsymbol{\beta}} \end{bmatrix}, \\ &\mathbf{D}_{(m+1)\times(m+1)} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \\ &\mathbf{W}_{1\times(m+1)} = \mathbf{0} \\ &\mathbf{A}_{n\times(m+1)} = y_i \cdot \begin{bmatrix} 1 & \mathbf{X} \end{bmatrix}, \\ &\mathbf{z}_{n\times 1} = \mathbf{1} \end{aligned}$$

Finally, solving the QP problem will obtain the minimized $\boldsymbol{\omega}$. 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 $y_i(\mathbf{X}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0) \geq 1$), one often needs to transform the feature space $\mathbf{X} \in \mathbf{R}^m$ into higher dimensional space $\Phi(\mathbf{X}) = \mathbf{Z} \in \mathbf{R}^k$. 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 $y_i(\mathbf{Z}\widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}_0) \geq 1$) 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

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

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 $\widehat{\beta}_0$:

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

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

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

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:

$$\sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \mathbf{Z} = \widehat{\boldsymbol{\beta}}$$

$$\sum_{i=1}^{n} \alpha_i \cdot y_i = 0$$

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

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

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

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 Implementing SVMs Using R

The key to implementing a support vector machine is to solve a quadratic programming problem. Now the question is how to use programming languages to solve QP problems. As a matter of fact, most languages have packages or libraries that we can use to do QP equations. Given these capacities, it is straighforward to code support vector machines.

Now I start with the data:

```
> data(iris)
> df<-iris
```

Then we take a partial look at the data:

```
> head(df)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

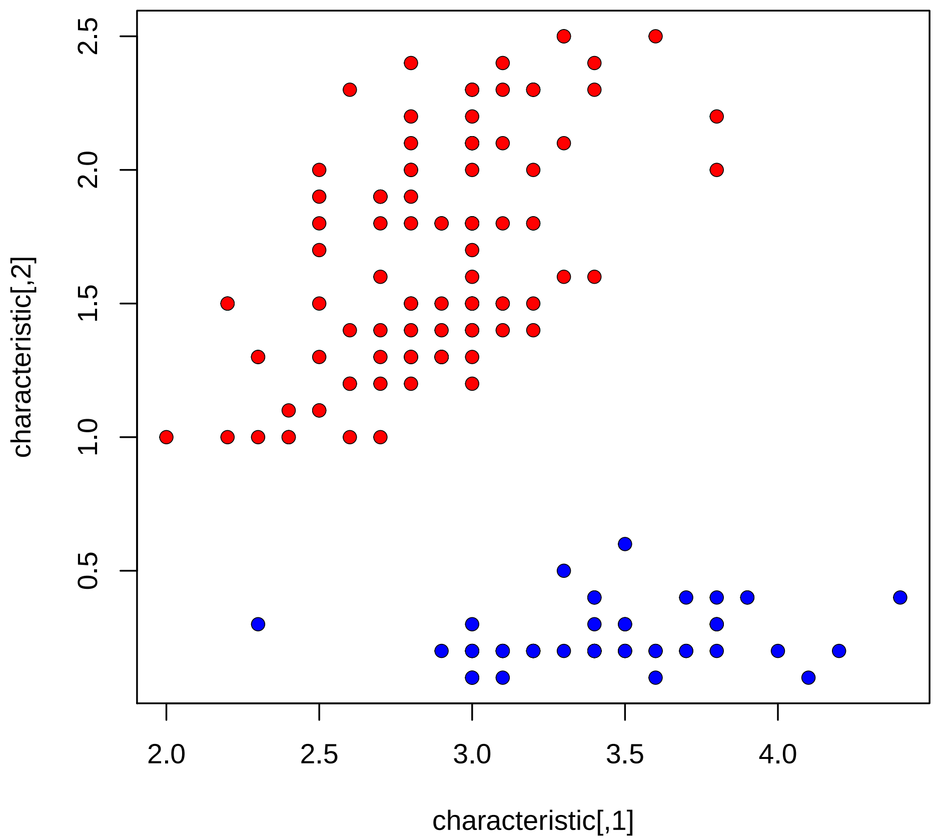
Columns 1 to 4 are flowers' characteristics. The last column, "Species," is a categorical variable:

```
> summary(df$Species)
```

```
    setosa versicolor  virginica
        50         50         50
```

The two separable groups are (1) versicolor and virginica and (2) setosa. Let's plot these two groups in terms of their petal width and sepal width:

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



Since the goal to solve QP problems, we need to import the library that can do this task:

```
> library(quadprog)
```

Now, let's construct all the components of the QP equation:

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

So we need to construct **D**, **W**, **A**, and **z**.
First, **D** is an identity matrix:

```
> D <- diag(1,3,3)
```

Second, **W** is a zero vector

```
> W <- rep(0,3)
```

Third, $\mathbf{A} = y_i[1 \quad \mathbf{X}]$

```
> A <- matrix(cbind(df[,"Sepal.Width"],df[,"Petal.Width"],rep(-1,nrow(df))),nrow=nrow(df),ncol=3)
> A <- A*df$Species
```

Last, $\mathbf{z}$ is an identity vector.

```
> z<-rep(1,nrow(df))
```

Once all the components have been specified, we can call a function from the library to solve the QP problem:

```
> beta = solve.QP(D, W, t(A), z)
```

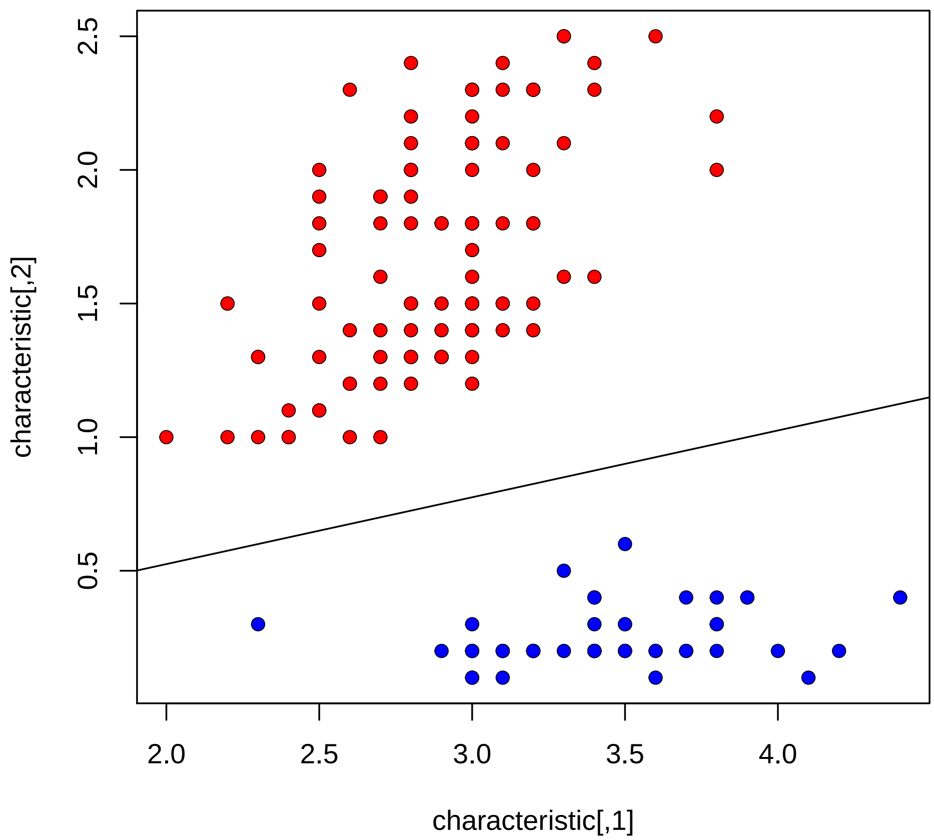
Then the answer, $\boldsymbol{\omega}$, can be obtained by calling:

```
> beta$solution
```

```
> 0.83333333 -3.33333333 -0.08333333
```

Finally, plot the hyperplane (line) based on our solutions to the QP problem:

```
> characteristic <- cbind(df$Sepal.Width,df$Petal.Width)
> plot(characteristic)
> points(subset(characteristic,df$Species=="setosa"),col="blue",pch=16)
> points(subset(characteristic,df$Species!="setosa"),col="red",pch=16)
> abline(-0.08333333/-3.33333333,0.83333333/3.33333333,col="black")
```



characteristic[,2]
0.5
1.0
1.5
2.0
2.5
2.0
2.5
3.0
3.5
4.0
characteristic[,1]