Machine Learning – SVM

7.2

I use the R package 'e1071' for svm classification, which us the R interface of the LIBSVM.

For details, I use linear kernel and set the cost© coefficient to be 10.

```
library(e1071)
```

```
## Loading required package: class
```

```
data = data.frame(V1 = c(1, 2, 3, 2, 3), V2 = c(2, 3, 3, 1, 2), y =
as.factor(c(1,
    1, 1, -1, -1)))
model = svm(y ~ ., data = data, kernel = "linear", cost = 10, scale =
F)
summary(model)
```

```
##
## Call:
## svm(formula = y ~ ., data = data, kernel = "linear", cost = 10,
##     scale = F)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  10
##       gamma:  0.5
##
## Number of Support Vectors:  3
##
##  ( 2 1 )
##
##
## Number of Classes:  2
##
## Levels:
##  -1 1
```

We can see the classification precision as below:

```
pred <- predict(model, data)
table(pred, data$y)
```

```
##
## pred -1 1
##   -1  2 0
##   1   0 3
```

All training points are classified into the right class(their original class).

I calculate the coefficients for the classification's hyperplane using the returned values of the model.

```
(w = t(model$coefs) %*% model$SV)
```

```
##      V1 V2
## [1,] -1  2
```

```
(b = model$rho)
```

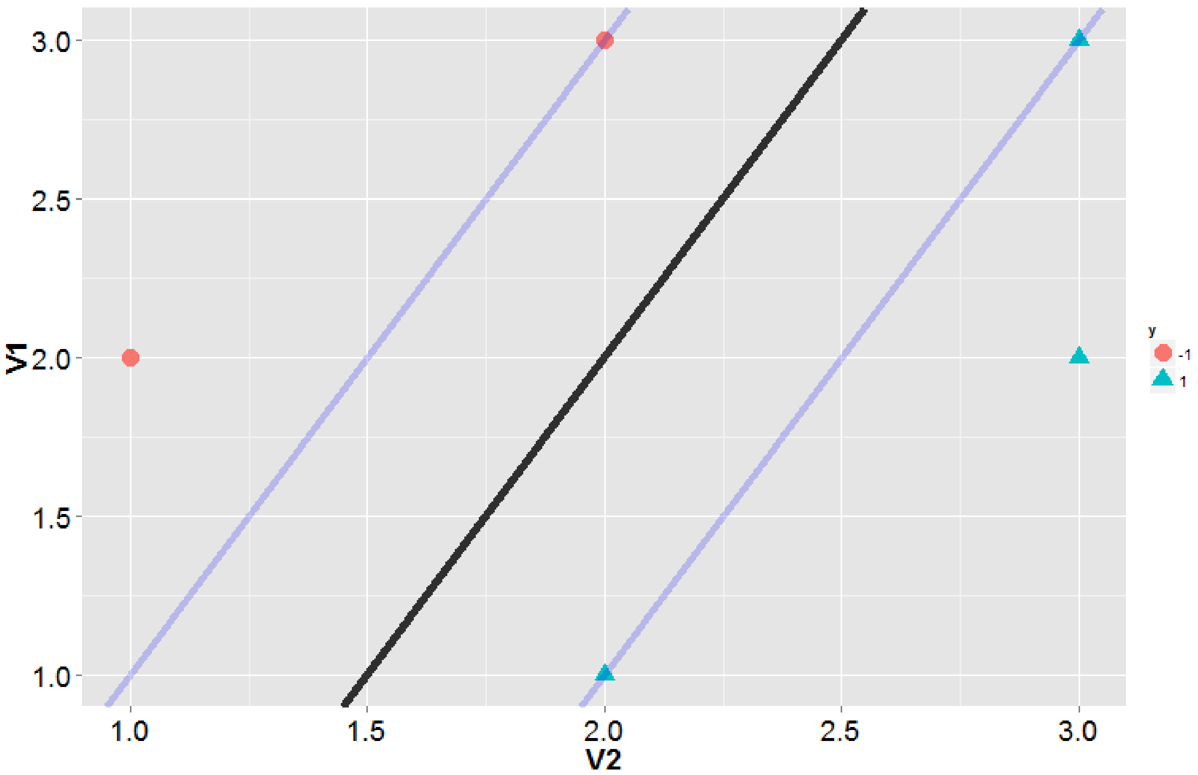
```
## [1] 2
```

The classification hyperplane is: $-X_1 + 2X_2 = 2$

The margin hyoerplanes are: $-X_1 + 2X_2 = 2 + 1 = 3$ and $-X_1 + 2X_2 = 2 - 1 = 1$

Then we draw them on one picture:

```
library(ggplot2)
p <- ggplot(data, aes(V1, V2), group = y) + geom_point(aes(colour = y,
shape = y),
    size = 5, xlim = c(0, 4))
p <- p + geom_abline(intercept = 2/2, slope = 1/2, colour = "black",
size = 2,
    alpha = 0.8) + coord_flip()
p <- p + geom_abline(intercept = (2 + 1)/2, slope = 1/2, colour =
"blue", size = 1.5,
    alpha = 0.2) + coord_flip()
p <- p + geom_abline(intercept = (2 - 1)/2, slope = 1/2, colour =
"blue", size = 1.5,
    alpha = 0.2) + coord_flip()
p + theme(axis.text = element_text(size = rel(1.5), colour = "black"),
axis.title.y = element_text(size = rel(1.5),
    angle = 90, face = "bold"), axis.title.x = element_text(size =
rel(1.5),
    face = "bold"), plot.title = element_text(size = rel(1.8), face =
"bold"),
    strip.text = element_text(size = 15, face = "bold", hjust = 0.5,
vjust = 0.5))
```



7.3

Problem Description:

$$
\begin{aligned}
&min_{w,b,\xi} \qquad \frac{1}{2}\left\|w\right\|^2 + C\sum_{i=1}^{N}\xi^2 \\
&s.t \qquad y_i(w\cdot x_i + b) \geq 1-\xi_i, i = 1,2,\cdots,N \\
&\qquad\qquad\quad \xi_i \geq 0, i = 1,2,\cdots,N
\end{aligned}
$$

First we write down the Lagrange Multiplier:

$$L = 1/2\left\|w\right\|^2 + C\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N}\alpha_i[y_i(w\cdot x_i + b) - 1 + \xi_i] + \sum_{i=1}^{N}\beta_i\xi_i$$

The dual problem is :

$$max_{\alpha,\beta}min_{w,b,\xi}\, L(w,b,\xi,\alpha,\beta)$$

We first solve the min problem. By using the first order condition, we have:

$$
\begin{aligned}
&\nabla w = w - \sum_{i=1}^{N}\alpha_i y_i x_i = 0 \\
&\nabla b = \sum_{i=1}^{N}\alpha_i y_i = 0 \\
&\nabla \xi_i = 2C\xi_i - \alpha_i + beta_i = 0
\end{aligned}
$$

By substituting the condition into the L function, we get:

$$
\begin{aligned}
&L = 1/2\sum_{i=1}^{N}\sum_{i=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j) - \sum_{i=1}^{N}\alpha_i y_i[(\sum_{j=1}^{N}\alpha_j y_j x_j)x_i + b] - \sum_{i=1}^{N}\alpha_i \\
&= -1/2\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j) - \sum_{i=1}^{N}\frac{(\alpha_i - \beta_i)^2}{2C}
\end{aligned}
$$

And it subjects to the restrictions:

$\sum_{i=1}^{N} \alpha_i y_i = 0$
$\alpha_i \geq 0$
$\alpha_i \geq \beta_i \geq 0$