

Nama : Laila Sirri Hayati

NIM : 19611050

Kelas : C

## Support Vector Machines (SVM)

1. Berikut merupakan *syntax* yang digunakan dalam kasus ini :

```
source("http://bioconductor.org/bioclite.R/")
bioclite("rgl")
library(rgl)
library(misc3d)
library(devtools)
library(rgl)
library(e1071)

n = 10
nbaru = 5
kelas = sample(2, n, replace = T)
kelas
dat = data.frame(kelas=factor(kelas), matrix(rnorm(n*3, rep(kelas, each=3)), ncol=3, byrow=T))
dat
fit = svm(kelas ~., data = dat)
summary(fit)
plot3d(dat[, -1], col=dat$kelas)

datbaru.list = lapply(dat[, -1], function(x) seq(min(x), max(x), len=nbaru))
datbaru.list
datbaru = expand.grid(datbaru.list)
datbaru.pred = predict(fit, newdata=datbaru, decision.values=T)
datbaru.dv = attr(datbaru.pred, 'decision.values')
datbaru.dv = array(datbaru.dv, dim=rep(nbaru,3))
```

2. Langkah pertama, meng-*install* dan mengaktifkan *packages* yang akan digunakan dalam SVM.

```
> library(rgl)
> library(misc3d)
> library(devtools)
> library(rgl)
> library(e1071)
```

3. Selanjutnya, membuat data sebanyak 10, dan data baru untuk prediksi.

```
> n = 10
> nbaru = 5
```

4. Kemudian, membuat grup untuk data yang telah dibuat sebelumnya

```
> kelas = sample(2, n, replace = T)
> kelas
[1] 1 1 2 1 2 2 1 2 2 2
```

5. Setelah itu, mendefinisikan data tersebut.

```
> dat = data.frame(kelas=factor(kelas), matrix(rnorm(n*3, rep(kelas, each=3)), ncol=3, byrow=T))
> dat
  kelas      X1      X2      X3
1     1 1.86585847 1.42283468 1.8079176
2     1 1.28023276 1.52969481 2.3338642
3     2 0.90390589 2.09072583 1.4861314
4     1 0.33739472 0.03375036 0.5190489
5     2 0.07799870 1.35164896 0.3818214
6     2 3.09174954 2.60836431 2.0827643
7     1 0.57325633 0.02730675 0.2457501
8     2 0.04093945 2.81070568 0.8308958
9     2 3.20126418 2.76899728 2.8383093
10    2 2.34230282 0.40111536 0.9494184
```

6. Selanjutnya membuat *hyperplane* yang berguna untuk pemisah/kelas dari data.

```
> fit = svm(kelas ~., data = dat)
> fit

Call:
svm(formula = kelas ~ ., data = dat)

Parameters:
  SVM-Type:  C-classification
 SVM-Kernel: radial
      cost:  1

Number of Support Vectors:  9
```

7. Kemudian melihat hasil SVM dengan perintah `summary()`.

```
> summary(fit)

Call:
svm(formula = kelas ~ ., data = dat)

Parameters:
  SVM-Type:  C-classification
 SVM-Kernel: radial
      cost:  1

Number of Support Vectors:  9

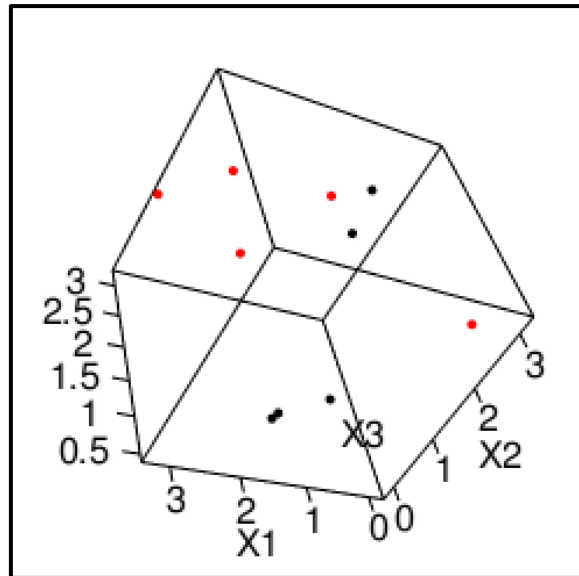
( 4 5 )

Number of Classes:  2

Levels:
 1 2
```

8. Lalu melakukan visualisasi data dalam 3 dimensi.

```
> plot3d(dat[, -1], col=dat$kelas)
```



9. Setelah itu, membuat 5 data baru yang nantinya akan sebagai data test dan akan diprediksi kelasnya juga.

```
> plot3d(dat[, -1], col=dat$kelas)
> datbaru.list = lapply(dat[, -1], function(x) seq(min(x), max(x), len=nbaru))
> datbaru.list
$X1
[1] 0.04093945 0.83102064 1.62110182 2.41118300 3.20126418
$X2
[1] 0.02730675 0.72315648 1.41900621 2.11485594 2.81070568
$X3
[1] 0.2457501 0.8938899 1.5420297 2.1901695 2.8383093
```

10. Lalu membuat data.frame dari fungsi `expand.grid()`.

```
> datbaru = expand.grid(datbaru.list)
> datbaru
      X1      X2      X3
1  0.04093945 0.02730675 0.2457501
2  0.83102064 0.02730675 0.2457501
3  1.62110182 0.02730675 0.2457501
4  2.41118300 0.02730675 0.2457501
5  3.20126418 0.02730675 0.2457501
6  0.04093945 0.72315648 0.2457501
7  0.83102064 0.72315648 0.2457501
8  1.62110182 0.72315648 0.2457501
9  2.41118300 0.72315648 0.2457501
10 3.20126418 0.72315648 0.2457501
11 0.04093945 1.41900621 0.2457501
12 0.83102064 1.41900621 0.2457501
13 1.62110182 1.41900621 0.2457501
14 2.41118300 1.41900621 0.2457501
15 3.20126418 1.41900621 0.2457501
16 0.04093945 2.11485594 0.2457501
17 0.83102064 2.11485594 0.2457501
18 1.62110182 2.11485594 0.2457501
19 2.41118300 2.11485594 0.2457501
20 3.20126418 2.11485594 0.2457501
21 0.04093945 2.81070568 0.2457501
```

11. Selanjutnya, memprediksi data tersebut.

```
> datbaru.pred = predict(fit, newdata=datbaru, decision.values=T)
> datbaru.pred
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
1 1 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
2 2 2 1 1 1 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2
45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
2 2 2 2 2 2 1 1 1 2 2 1 1 2 2 2 2 2 2 2 2 2
67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88
2 2 2 2 2 2 2 2 2 1 1 1 2 2 1 1 2 2 2 2 2 2
89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110
2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 1 1 1 1 2
111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
attr(,"decision.values")
1/2
1 0.608383605
2 0.524214031
3 0.060416222
4 -0.467650837
5 -0.746029624
```

12. Setelah itu diberikan label untuk *decision values*-nya dengan disusun sebagai data array.

```
> datbaru.dv = attr(datbaru.pred,'decision.values')
> datbaru.dv = array(datbaru.dv, dim=rep(nbaru,3))
> datbaru.dv
, , 1
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.60838361 0.029625053 -0.6752835 -1.0831731 -1.0839833
[2,] 0.52421403 -0.007487807 -0.6611978 -1.0541931 -1.0730541
[3,] 0.06041622 -0.297959923 -0.7146712 -0.9602329 -0.9625128
[4,] -0.46765084 -0.645433939 -0.8039441 -0.8746770 -0.8400411
[5,] -0.74602962 -0.815239886 -0.8268330 -0.7984769 -0.7435045
, , 2
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.53715489 0.02288696 -0.6481038 -1.0816618 -1.1278801
[2,] 0.46166314 0.01932473 -0.5913192 -1.0280340 -1.1211958
[3,] 0.01215155 -0.25254664 -0.6296918 -0.9339690 -1.0284657
[4,] -0.51603235 -0.62581227 -0.7655144 -0.9011908 -0.9524906
[5,] -0.80158090 -0.84314944 -0.8629060 -0.8949817 -0.9006766
, , 3
```

13. Terakhir membuat plot *isosurface* ke dalam *decision boundary*.

```
> contour3d(datbaru.dv, level=0, x=datbaru.list$X1, y=datbaru.list$X2, z=datbaru.list$X3, add=T)
```

