中研院統計所2015統計研習營
2015/07/01~03

Data Mining

(9) 分類 Classification

(Supervised Learning)

吳漢銘
淡江大學 數學系
資料科學與數理統計組

http://www.hmwu.idv.tw

Outlines

- Introduction
- Classification of Genes, Tissues or Samples (Supervised Learning)
- Performance Measures
- Methods
  - K-Nearest Neighbors (KNN) (*k*最近鄰居法)
  - Classification Tree (Decision Tree) (分類樹、決策樹)
  - Linear Discriminant Analysis (LDA) (線性區別分析)
  - Support Vector Machine (SVM) (支持向量機)

R Packages

- **CRAN Task View: Machine Learning & Statistical Learning**

http://cran.r-project.org/web/views/MachineLearning.html

- **knn (最近k鄰居分類法)**

class: Functions for Classification
http://cran.r-project.org/web/packages/class/index.html

- **Decision Tree (決策樹)**

C50: C5.0 Decision Trees and Rule-Based Models
http://cran.r-project.org/web/packages/C50/index.html
rpart: Recursive Partitioning and Regression Trees
http://cran.r-project.org/web/packages/rpart/index.html

- **lda (線性區別分析)**

MASS: Support Functions and Datasets for Venables and Ripley's MASS
http://cran.r-project.org/web/packages/MASS/index.html

- **svm (支持向量機)**

e1071: Misc Functions of the Department of Statistics (e1071), TU Wien
http://cran.r-project.org/web/packages/e1071/index.html

What is Classification?

- **Classification**
  - Clustering (unsupervised learning)
    (群集分析、非監督式學習)
  - Discriminant Analysis (supervised learning, classification)
    (區別分析、監督式學習、分類法則)

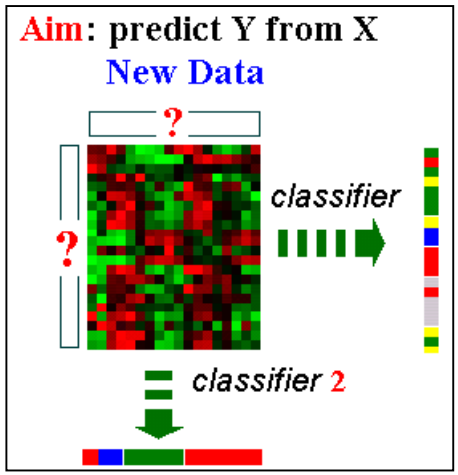
- **Discriminant Analysis**
  - It focuses on situations where the different groups (clusters) are known a priori.
  - Decision rules are provided in classifying a multivariate observation into one of the known groups.

Class Prediction Analysis

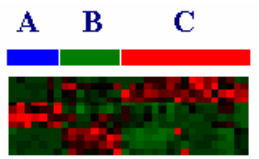
- Class prediction analysis is designed to predict the value, or "class", of an individual parameter in an uncharacteristic sample or set of samples.

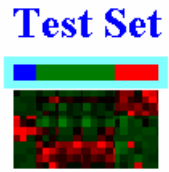
- **Examples**
  - Predict cancer types using genomic expression profiling.
  - Predict the class/phenotype/parameter of a sample.
  - Identify genes that discriminate well among classes
  - Identify samples that could be potential outliers.

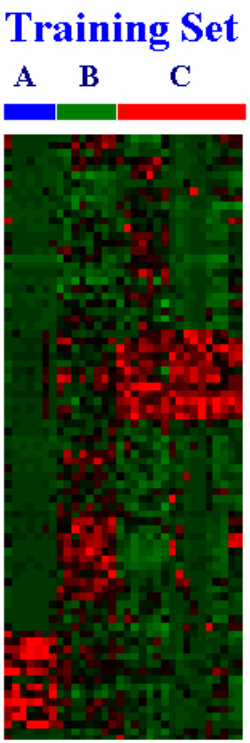
Classification of Genes, Tissues or Samples



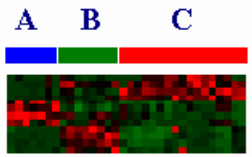
Possible to
1. classification for genes
2. classification for samples (arrays)





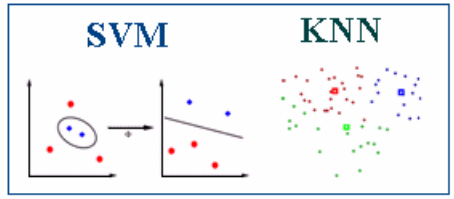


Gene Selection Methods



most predictive 50 genes

Construct

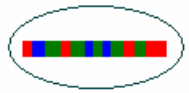


Assign class labels

predicted →

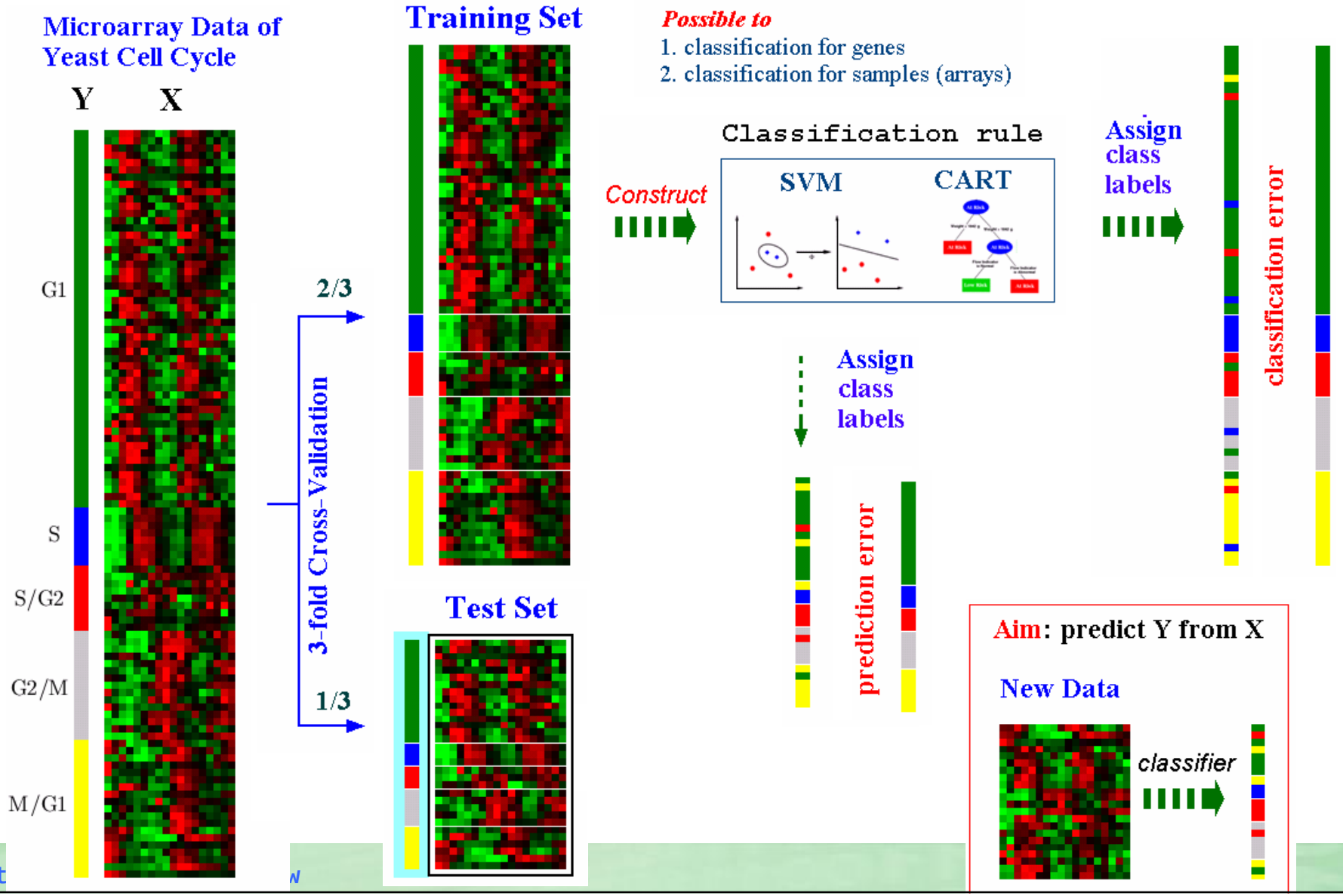
classification error

true →



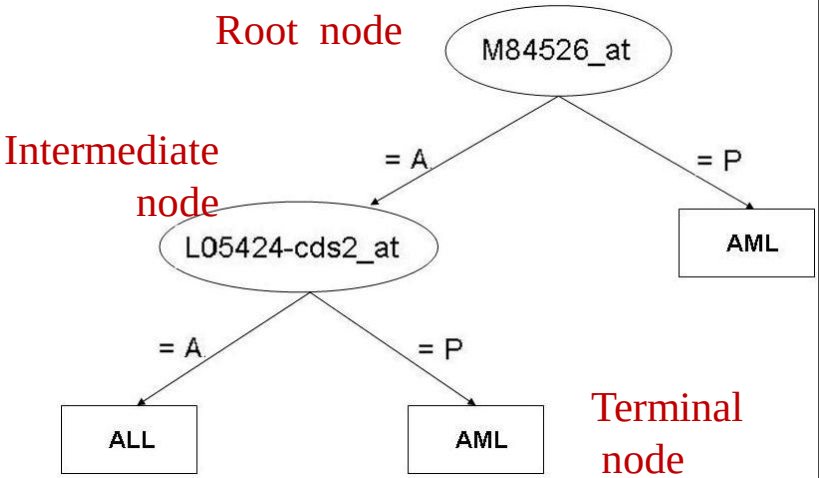
prediction error

n-fold Cross-Validation Error Rates



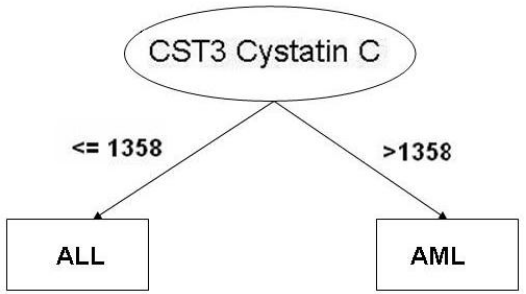
Apply Classification Tree to Microarray Data

- **Growing a tree (classification rule)**: variable selection, split criterion and tree Pruning



The Leukemia Dataset [Golub et al. 1999].

- There are 72 patients (47 ALL ((急性淋巴細胞白血病))and 25 AML(急性骨髓性白血病)).
- Each sample (patient) consists 7219 genes expression values.
- **Training set**: 38 samples (27 ALL and 11 AML)
- **Test set**: 34 samples (20 ALL and 14 AML).
- Nominal values:
  - A: gene is absent or not expressed;
  - P: gene is expressed or present
  - M: the level of the expression is marginal among A and P



Gene ID	Description	References
M84526_at	Human adipsin/complement factor D	[Dobra 2008] [Schachtner et al. 2007]
L05424-cds2_at	Human cell surface glycoprotein CD44	[Screaton et al. 1992] [Krause et al. 2006]
CST3	Cystatin C (amyloid angiopathy and cerebral hemorrhage)	[Tang et al. 2009] [Sun et al. 2004]

OP Netto et al., Applying Decision Trees to Gene Expression Data from DNA Microarrays: A Leukemia Case Study. Technical report.

Apply C5.0 to Iris Data

```
library(C50)
attach(iris)

# setup the training and testing data
id <- sample(1:nrow(iris),
2*floor(nrow(iris)/3))
x.train <- iris[id, 1:4]
y.train <- Species[id]
x.test <- iris[-id, 1:4]
y.test <- Species[-id]

# C5.0 Decision Tree
treeModel <- C5.0(x.train, y.train)
treeModel
summary(treeModel)
```

```
> treeModel

Call:
C5.0.default(x = x.train, y = y.train)

Classification Tree
Number of samples: 100
Number of predictors: 4

Tree size: 7

Non-standard options: attempt to group attributes
```

```
> summary(treeModel)

Call:
C5.0.default(x = x.train, y = y.train)


C5.0 [Release 2.07 GPL Edition]         Mon Jul 06 17:13:47 2015
-------------------------------

Class specified by attribute `outcome'

Read 100 cases (5 attributes) from undefined.data

Decision tree:

Petal.Length <= 1.7: setosa (29)
Petal.Length > 1.7:
:...Petal.Width > 1.7: virginica (33/1)
    Petal.Width <= 1.7:
    :...Petal.Width <= 1.4: versicolor (27)
        Petal.Width > 1.4:
        :...Petal.Length > 5: virginica (2)
            Petal.Length <= 5:
            :...Sepal.Width > 2.6: versicolor (5)
                Sepal.Width <= 2.6:
                :...Sepal.Length <= 6.1: virginica (2)
                    Sepal.Length > 6.1: versicolor (2)


Evaluation on training data (100 cases):

           Decision Tree
         ----------------
         Size      Errors

            7    1( 1.0%)   <<


          (a)   (b)   (c)    <-classified as
         ----  ----  ----
           29                (a): class setosa
                 34     1    (b): class versicolor
                       36    (c): class virginica


        Attribute usage:

        100.00% Petal.Length
         71.00% Petal.Width
          9.00% Sepal.Width
          4.00% Sepal.Length
```

Apply C5.0 to Iris Data

```
# Predition and Accuracy
test.pred <- predict(treeModel, x.test)
(ct <- table(y.test, test.pred))
accuracy <- sum(diag(ct))/sum(ct)
accuracy

names(treeModel)
```

```
> (ct <- table(y.test, test.pred))
            test.pred
y.test       setosa versicolor virginica
  setosa         19          2         0
  versicolor      0         14         1
  virginica       0          1        13
```

```
> accuracy
[1] 0.92
> names(treeModel)
 [1] "names"        "cost"         "costMatrix"   "caseWeights"
 [5] "control"      "trials"       "rbm"          "boostResults"
 [9] "size"         "dims"         "call"         "levels"
[13] "output"       "tree"         "predictors"   "rules"
```

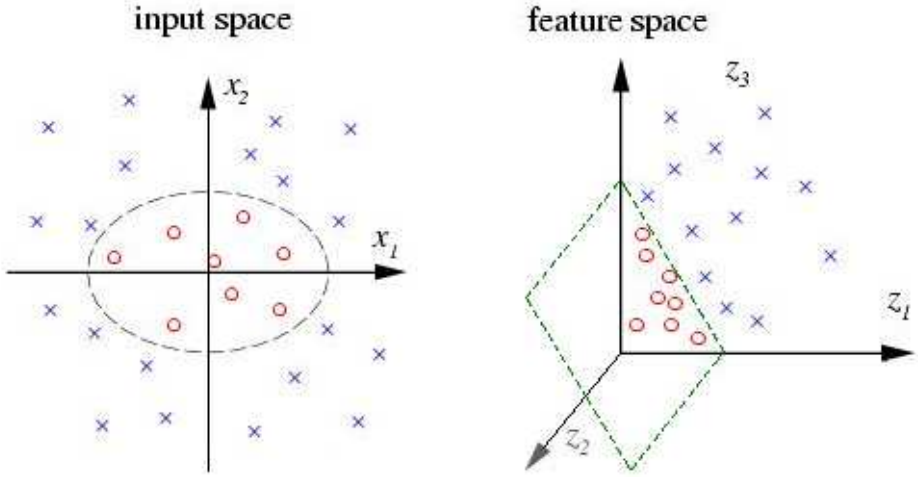
課堂練習: 把分類的結果，呈現在資料的前兩維主成份(PCA)空間上。

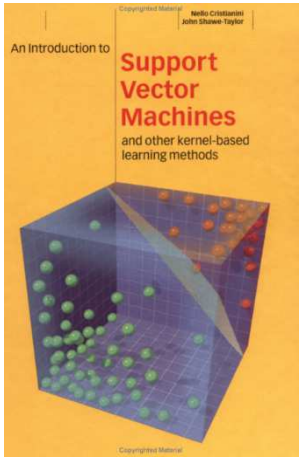
Support Vector Machines (SVMs)

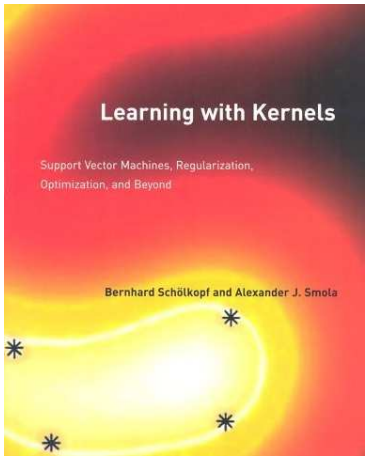
SVMs (Vapnik, 1995) map the data (input space) into high dimensional space (feature space) through a kernel function ϕ and then find a hyperplane $\mathbf{w}$ to separate two groups (binary classification).

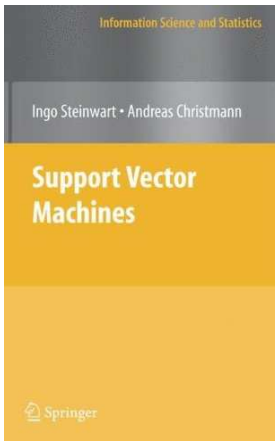
Support Vector Classifiers

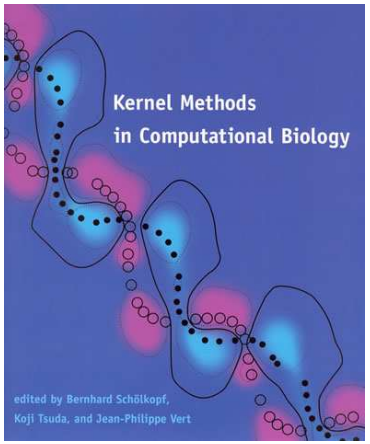
$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$



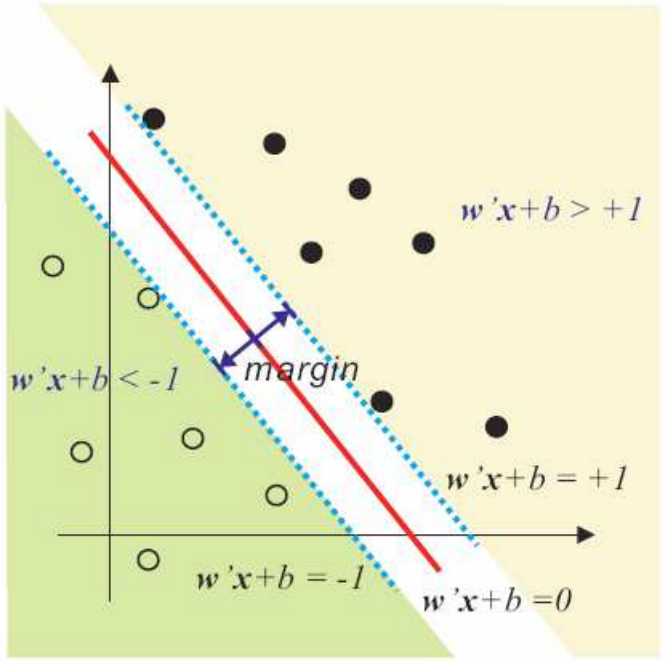


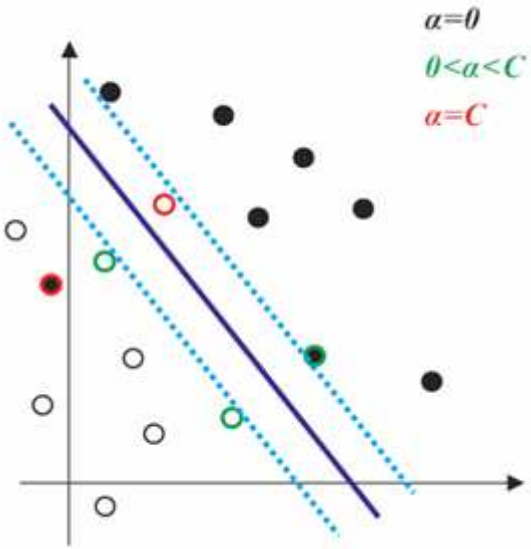






popular in data mining and machine learning





1. A function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ defines two half-spaces where points are classified with large confidence.

2. The distance between the half-spaces is equal to $1/\|\mathbf{w}\|$.

3. $\mathbf{w}^T\mathbf{x}_1 + b_0 = +1$, $\mathbf{w}^T\mathbf{x}_2 + b_0 = -1$,
$\mathbf{w}^T(\mathbf{x}_1 - \mathbf{x}_2) = 2$, $\dfrac{\mathbf{w}^T}{\|\mathbf{w}\|}(\mathbf{x}_1 - \mathbf{x}_2) = \dfrac{2}{\|\mathbf{w}\|}$,

SVM: Optimization Problem

SVMs combine the requirements of

(a) large margin (i.e., small $\|\mathbf{w}\|$), and

(b) few misclassifications or

(b') classifications with little confidence on the training set,

by solving the problem

$$\underset{f(\mathbf{x})=\mathbf{w}^T\mathbf{x}+b}{\arg\min} \{ \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{n} c(f, \mathbf{x}_i, y_i)\},$$

where C controls the tradeoff between the two requirements.

Two approaches for multi-class classification:

- **one-against-others**: The kth SVM model is constructed with all of the samples in the kth class with one group, and all other samples with the other group.
- **one-against-one**: The SVM trained model is constructed by using any two of classes. Therefore, there are total $K(K-1)/2$ classifiers.

Apply SVM to Iris Data

```
# prepare data
library(e1071)
attach(iris)
x <- subset(iris, select = -Species)
y <- Species

# use default setting (?svm)
model <- svm(x, y)
print(model)
summary(model)

# test with train data and report accuracy
pred <- predict(model, x)
table(pred, y)

> sum(diag(table(pred, y)))/length(y)
[1] 0.9733333
```

```
> head(x)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1         3.5          1.4         0.2
2          4.9         3.0          1.4         0.2
3          4.7         3.2          1.3         0.2
4          4.6         3.1          1.5         0.2
5          5.0         3.6          1.4         0.2
6          5.4         3.9          1.7         0.4
> head(y)
[1] setosa setosa setosa setosa setosa setosa
Levels: setosa versicolor virginica
```

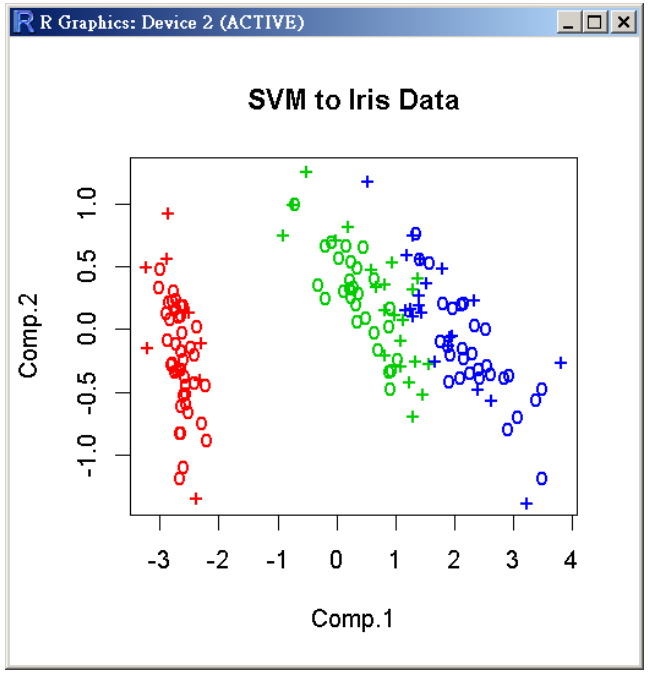
```
> summary(model)

Call:
svm.default(x = x, y = y)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.25

Number of Support Vectors:  51

 ( 8 22 21 )


Number of Classes:  3

Levels:
 setosa versicolor virginica
```

```
> head(pred)
     1      2      3      4      5      6
setosa setosa setosa setosa setosa setosa
Levels: setosa versicolor virginica
```

```
> table(pred, y)
            y
pred         setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          2        48
```

Apply SVM to Iris Data

```
# visualize (classes by color, SV by crosses):
pca <- princomp(iris[,-5], scores=TRUE)
plot(pca$scores[,1:2],
     col = as.integer(iris[,5])+1,
     pch = c("o","+")[1:150 %in% model$index + 1],
     main="SVM to Iris Data")
```




```
> head(model$index)
[1]  9 14 16 21 23 24
> head(1:150 %in% model$index)
[1] FALSE FALSE FALSE FALSE FALSE FALSE
> head(1:150 %in% model$index + 1)
[1] 1 1 1 1 1 1
> head(c("o","+")[1:150 %in% model$index + 1])
[1] "o" "o" "o" "o" "o" "o"
```

課堂練習:

(1) Randomly divide iris data into the training set (2/3) and testing set (1/3) and then apply SVM.

(2) Use the 10-fold CV technique.