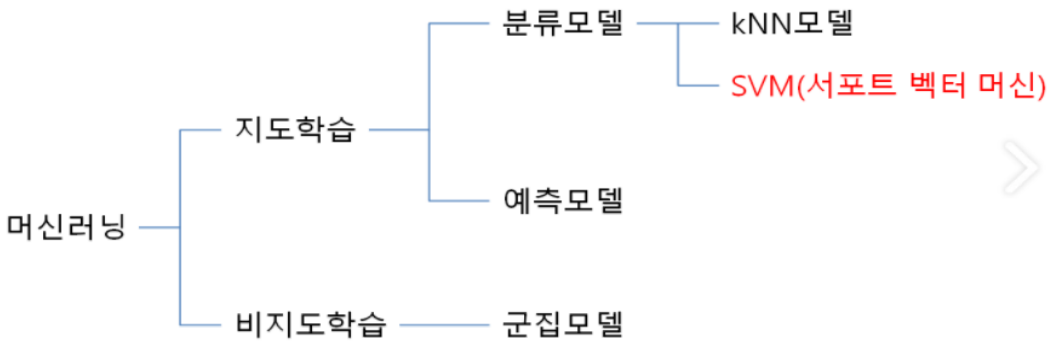
	단위별 학습내용 (Week11)
wk11-1	서포트벡터머신 I
wk11-2	서포트벡터머신 II
wk11-3	서포트벡터머신 III

Wk11-1 : 서포트벡터머신 I
(Support Vector Machine)

1. 서포트벡터머신 (Support Vector Machine)

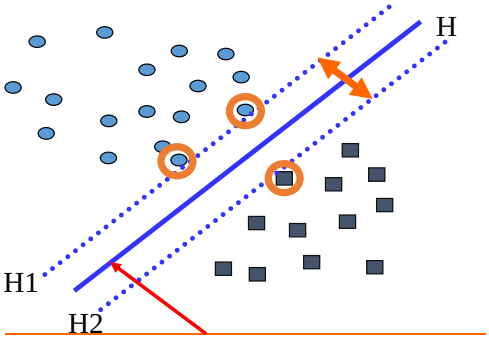


장점
- ✓ 정확도가 상대적으로 좋음
- ✓ 다양한 데이터 (연속형, 범주형)를 다룰 수 있음

단점
- ✓ 해석상의 어려움이 있다
- ✓ 데이터가 많을 때 속도와 메모리가 많이 듬

1. 서포트벡터머신 (Support Vector Machine)

▪ 선형 SVM



<분리 가능한 경우의 하이퍼플레인>

H1: w' x + b = 1
H2: w' x + b = -1 } H와 평행인 두 하이퍼플레인 (단, H1과 H2사이에 객체 X)

각각의 범주에서 H(분리 하이퍼플레인)와 가장 가까운 객체를 지닌 평면

H1와 H2간의 거리 (margin): 2/||w||

이 거리를 최대로 하는 분리 하이퍼플레인을 찾자!

$$\text{Max } \frac{2}{w'w}$$

Subject to

$$w'x_i + b \geq 1 \quad for\ y_i = 1 \quad (i=1,\ldots,N)$$
$$w'x_i + b \leq -1 \quad for\ y_i = -1 \quad (i=1,\ldots,N)$$

최적화 문제 변환

$$\text{Min } \frac{w'w}{2}$$

Subject to

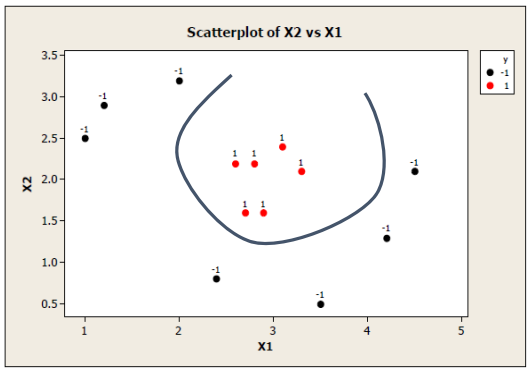
$$y_i(w'x_i + b) \geq 1 \quad (i=1,\ldots,N)$$

학습 표본 객체: N개

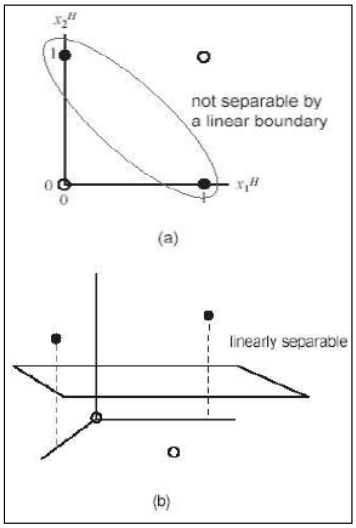
xi : p개의 변수로 이루어진 i번째 객체 벡터
yi : 대응하는 i 분류된 범주
(두 가지의 범주를 갖는다고 가정, Yi=1 or –1)

1. 서포트벡터머신 (Support Vector Machine)

▪비선형 SVM



<비선형 하이퍼플레인 도출>



<고차원(커널) 공간으로의 변환>

- 대부분의 패턴은 선형적으로 분리 불가능
  - 비선형 패턴의 입력공간을 선형 패턴의 'feature space'로 변환
  - Kernel method로 비선형 경계면 도출

2. iris 데이터 설명

• iris 데이터 (iris.csv)

input변수(독립변수)　　output변수(종속변수, 타겟변수)

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3	1.4	0.1	setosa

타겟변수(y) : setosa, versicolor, virginica

Iris setosa　Iris versicolor　Iris virginica

3. 서포트벡터머신 패키지와 함수

- 서포트벡터머신을 수행하기 위한 패키지 : e1071
- 서포트벡터머신 함수 : svm

```
# lec11_1_svm.r
# Classification
# support vector machine (e1071)

# install package for support vector machine
install.packages("e1071")
library (e1071)
#help(svm)

# set working directory
setwd("D:/tempstore/moocr/wk11")

# read data
iris<-read.csv("iris.csv")
attach(iris)
```

e1071 패키지 설치, 라이브러리 설정

데이터 불러오기

4. 서포트벡터머신 결과

- 서포트벡터머신 함수 : svm(y변수~x변수, data=)
- iris 데이터의 서포트벡터머신 결과 (전체 데이터를 사용한 결과)

```
## classification
# 1. use all data
m1<- svm(Species ~., data = iris)
summary(m1)
```

svm의 결과 요약

```
> summary(m1)

Call:
svm(formula = Species ~ ., data = iris)


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
```

svm에서 주어지는 옵션(default)

kernel=radial basis function,
gamma=1/(# of dimension) (1/4=0.25)

4. 서포트벡터머신 결과

• svm모델에 적용한 예측범주와 실제범주 비교(전체 데이터를 사용한 결과)

```
# classify all data using svm result (m1)
# first 4 variables as attribute variables
x<-iris[, -5]
pred <- predict(m1, x)

# Check accuracy (compare predicted class(pred) and true class(y))
# y <- Species or y<-iris[,5]
y<-iris[,5]
table(pred, y)
```

x<-iris[, -5]
iris데이터에서 타겟값인 5번째 열을 제외한 데이터, 즉 4개의 독립변수들만 있는 데이터

pred<-predict(m1, x)
svm모델 m1을 적용하여 예측된 범주값을 pred로 저장

```
> y<-iris[,5]
> table(pred, y)
            y
pred         setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          2        48
```

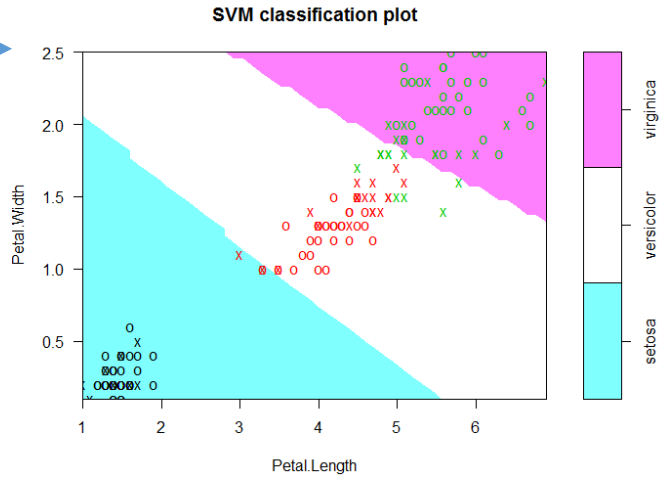
오분류율 :
(2+2)/150=0.0266 (2.66%)

4. 서포트벡터머신 결과 - 시각화

• iris 데이터의 서포트벡터머신 결과 (전체 데이터를 사용한 결과)

```
# visualize classes by color
plot(m1, iris,  Petal.Width~Petal.Length, slice=list(Sepal.Width=3,
```

svm결과를 그림으로 시각화



4개 변수 중 petal.width와 petal.length가 중요변수