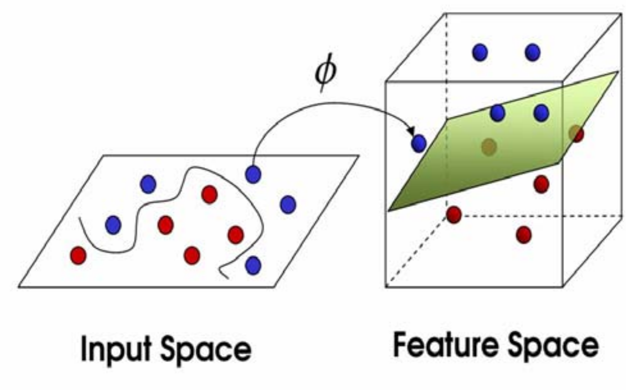
Wk11-2 : 서포트벡터머신 II
(Support Vector Machine)

1. 서포트벡터머신 (kernel 함수)

▪ 커널(kernel)이란?



$$\boxed{f(x) = \Phi(x)^T w + b}$$

커널함수 (kernel function)

- x의 기저함수 (basis function)
- x에 대한 새로운 특징을 추출하는 변환함수

커널함수와 기저함수의 관계: $\boxed{K(x_i, x_j) = \Phi(x_i)'\Phi(x_j)}$

- **radial :** $K(x_i, x_j) = \exp\left(\dfrac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$
- **polynomial :** $K(x_i, x_j) = (x_i'x_j + 1)^r$
- **sigmoid :** $K(x_i, x_j) = \tanh(\kappa x_i'x_j - \delta)$

1. 서포트벡터머신 (kernel 함수)

- 서포트벡터머신을 수행하기 위한 패키지 : e1071
- 오분류율 교차표(confusion matrix) 생성을 위한 패키지 : caret

```
# lec11_2_svm.r
# Classification
# support vector machine using kernel

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

# install package for crosstable
#install.packages("caret")#crosstable
library(caret)

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

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

(e1071 패키지 설치) 라이브러리 설정

(caret 패키지 설치) 라이브러리 설정

1. 서포트벡터머신 (kernel 함수)

- Iris 데이터 (학습데이터와 검증데이터의 분할)

```
# training (100) & test set (50)
set.seed(1000, sample.kind="Rounding")
N=nrow(iris)
tr.idx=sample(1:N, size=N*2/3, replace=FALSE)
# target variable
y=iris[,5]
# split train data and test data
train=iris[tr.idx,]
test=iris[-tr.idx,]
```

데이터분할 (학습데이터 2/3, 검증데이터 1/3)

train (100개의 데이터)
test (50개의 데이터)

2. kernel 함수에 따른 결과비교

• iris 데이터 (학습데이터와 검증데이터의 분할)

```
#svm using kernel
help("svm")
m1<-svm(Species~., data = train)
summary(m1)
m2<-svm(Species~., data = train,kernel="polynomial")
summary(m2)
m3<-svm(Species~., data = train,kernel="sigmoid")
summary(m3)
```

m1-kernel: radial
m2-kernel: polynomial
m3-kernel: sigmoid

help("svm")

`kernel`	the kernel used in training and predicting. You might consider changing some of the f depending on the kernel type. linear: *u'*v* polynomial: *(gamma*u'*v + coef0)^degree* radial basis: *exp(-gamma*|u-v|^2)* sigmoid: *tanh(gamma*u'*v + coef0)*
`degree`	parameter needed for kernel of type `polynomial` (default: 3)
`gamma`	parameter needed for all kernels except `linear` (default: 1/(data dimension))
`coef0`	parameter needed for kernels of type `polynomial` and `sigmoid` (default: 0)

2. kernel 함수에 따른 결과비교

▪ 서포트벡터머신 결과(kernel-radial basis function)

```
> summary(m1)

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


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

Number of Support Vectors:  38

 ( 5 16 17 )


Number of Classes:  3

Levels:
 setosa versicolor virginica
```

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$$

1/(data dimension))

#정확도 측정
pred11<-predict(m1,test)
confusionMatrix(pred11, test$Species)

```
> pred11<-predict(m1,test) # radial basis
> confusionMatrix(pred11, test$Species)
Confusion Matrix and Statistics

예측범주     Reference 실제범주
Prediction   setosa versicolor virginica
  setosa         19          0         0
  versicolor      0         18         1
  virginica       0          1        11

Overall Statistics

               Accuracy : 0.96
                 95% CI : (0.8629, 0.9951)
```

2. kernel 함수에 따른 결과비교

- 서포트벡터머신 결과(kernel-polynomial)

```
> summary(m2)

Call:
svm(formula = Species ~ ., data = train,
    kernel = "polynomial")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  1
     degree:  3
      gamma:  0.25
     coef.0:  0

Number of Support Vectors:  36

 ( 19 13 4 )


Number of Classes:  3

Levels:
 setosa versicolor virginica
```

$K(x_i, x_j) = (x_i' x_j + 1)^r$

```
#정확도 측정
pred12<-predict(m2,test)
confusionMatrix(pred12, test$Species)
```

```
> pred12<-predict(m2,test) # polynomial
> confusionMatrix(pred12, test$Species)
Confusion Matrix and Statistics
```

예측범주 / 실제범주

Prediction \ Reference	setosa	versicolor	virginica
setosa	19	0	0
versicolor	0	19	5
virginica	0	0	7

```
Overall Statistics

               Accuracy : 0.9
                 95% CI : (0.7819, 0.9667)
```


2. kernel 함수에 따른 결과비교

- 서포트벡터머신 결과(kernel-sigmoid)

```
> summary(m3)

Call:
svm(formula = Species ~ ., data = train,
    kernel = "sigmoid")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  sigmoid
       cost:  1
      gamma:  0.25
     coef.0:  0

Number of Support Vectors:  40

 ( 20 15 5 )


Number of Classes:  3

Levels:
 setosa versicolor virginica
```

$K(x_i, x_j) = \tanh(\kappa x_i' x_j - \delta)$

```
#정확도 측정
pred13<-predict(m3,test)
confusionMatrix(pred13, test$Species)
```

```
> pred13<-predict(m3,test) # simoid
> confusionMatrix(pred13, test$Species)
Confusion Matrix and Statistics
```

Prediction \ Reference	setosa	versicolor	virginica
setosa	19	0	0
versicolor	0	15	1
virginica	0	4	11

```
Overall Statistics

               Accuracy : 0.9
                 95% CI : (0.7819, 0.9667)
```

2. kernel 함수에 따른 결과비교

- **서포트벡터머신 결과(kernel-linear)**

```
> summary(m4)

Call:
svm(formula = Species ~ ., data = train,


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

Number of Support Vectors:  23

 ( 2 10 11 )


Number of Classes:  3

Levels:
 setosa versicolor virginica
```

```
#정확도 측정
pred14<-predict(m4,test)
confusionMatrix(pred14, test$Species)
```

```
> pred14<-predict(m4,test) # linear
> confusionMatrix(pred14, test$Species)
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         19          0         0
  versicolor      0         17         0
  virginica       0          2        12

Overall Statistics

               Accuracy : 0.96
                 95% CI : (0.8629, 0.9951)
```