SVM 실습

원중호

서울대학교 통계학과

2019년 6월 21일

목차

Introduction

Support vector classifier

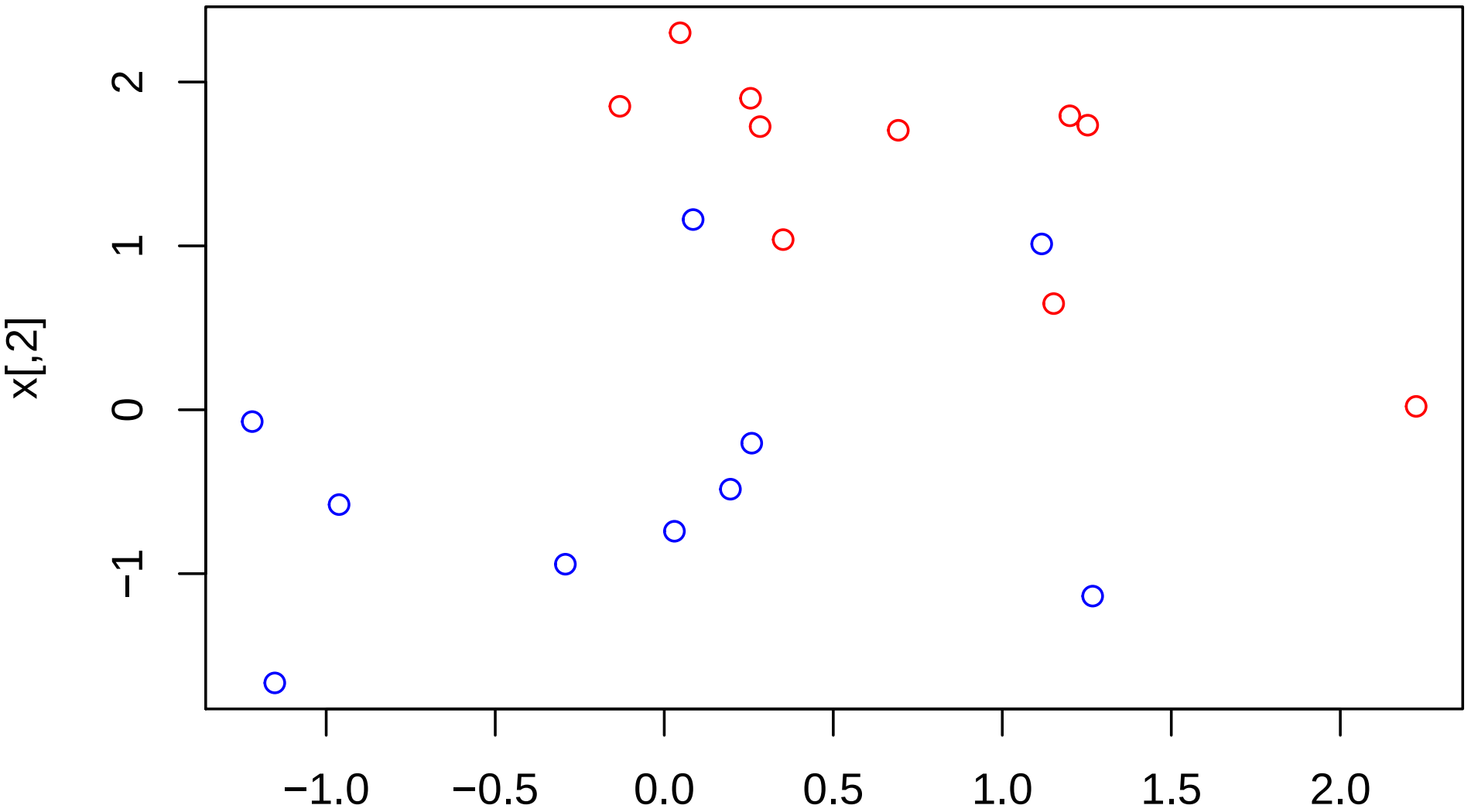
- 20개의 2차원 자료를 생성해서 2차원 설명변수 x를 구성.
  - 첫 10개에는 $y = 1$을 나머지는 $y = -1$을 배정.
  - $y = 1$인 설명변수는 일괄적으로 1을 더함.

```
set.seed(3)
x=matrix(rnorm(20*2), ncol=2)
y=c(rep(-1,10), rep(1,10))
x[y==1,]=x[y==1,] + 1
dat=data.frame(x=x, y=as.factor(y))
```

Support vector classifier

- 그림은 다음과 같다. y값에 따라 색깔이 달리 표시되게 하였다.

```
plot(x, col=(3-y))
```



Support vector classifier

- 지지벡터분류기의 적합 : e1071 패키지 안의 svm 함수를 이용.

```
library(e1071)
svmfit=svm(y~., data=dat, kernel="linear", cost=10,scale=FALSE)
```

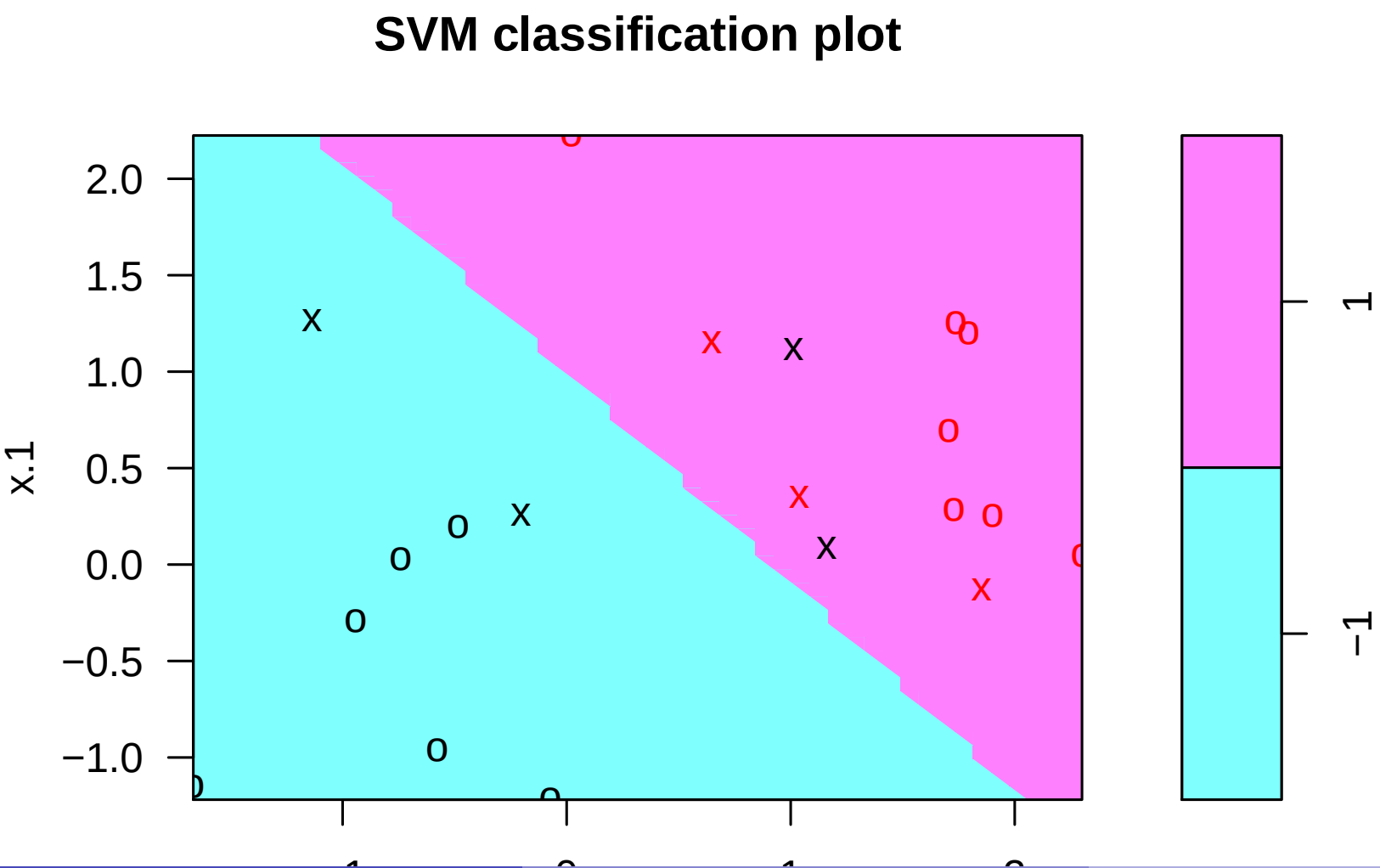
svm 함수의 옵션

- kernel = "linear" : 결정경계가 선형.
- cost : 예산에 관계되는 파라미터. 이 값이 커지면 예산이 작아지고, 이 값이 작아지면 예산이 커진다.
- scale = FALSE : 설명변수를 표준화하지 않는다는 뜻. (문제에 따라서는 표준화하는 것이 좋을 때도 많음.)

Support vector classifier

- x축 : x_2, y축 :x_1, 자료의 색깔은 y 값을 표시
  - x 표시 : 지지벡터, o 표시 : 그 외의 벡터

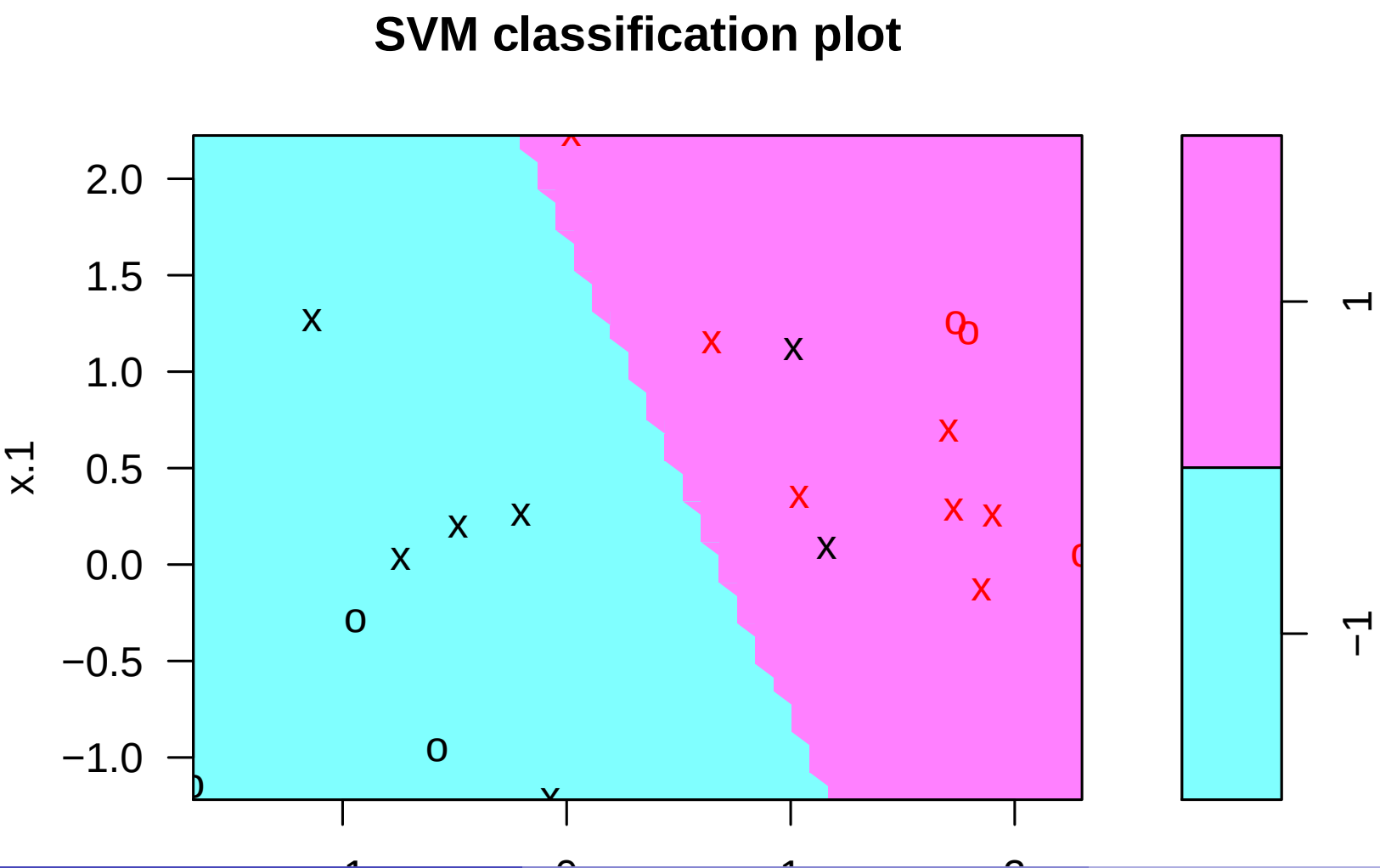
```
plot(svmfit, dat); svmfit$index
```



Support vector classifier

- 큰 예산 (적은 cost=0.1 값) : 여백을 크게 해서 지지벡터의 수를 늘림

```
svmfit=svm(y~., data=dat, kernel="linear", cost=0.1,scale=FALSE)
plot(svmfit, dat); svmfit$index
```



Support vector classifier

- 교차검증
  - 교차검증은 `tune` 함수를 이용한다.
  - `range`는 교차검증할 때 비교할 `cost`의 값들을 지정.

```
set.seed(1)
tune.out=tune(svm,y~.,data=dat,
              kernel="linear",
              ranges=list(cost=c(0.001, 0.01, 0.1, 1,5,10,100)))
```

Support vector classifier

```
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##   0.1
##
## - best performance: 0.1
##
## - Detailed performance results:
##    cost error dispersion
## 1 1e-03  0.65  0.4743416
## 2 1e-02  0.65  0.4743416
## 3 1e-01  0.10  0.2108185
## 4 1e+00  0.15  0.2415229
## 5 5e+00  0.10  0.2108185
```

Support vector classifier

```
bestmod=tune.out$best.model
summary(bestmod)
```

```
##
## Call:
## best.tune(method = svm, train.x = y ~ ., data = dat, ranges = lis
##     0.01, 0.1, 1, 5, 10, 100)), kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##       gamma:  0.5
##
## Number of Support Vectors:  14
##
##  ( 7 7 )
##
##
```

Support vector classifier

- 예측

```
xtest=matrix(rnorm(20*2), ncol=2)
ytest=sample(c(-1,1), 20, rep=TRUE)
xtest[ytest==1,]=xtest[ytest==1,] + 1
testdat=data.frame(x=xtest, y=as.factor(ytest))
ypred=predict(bestmod,testdat)
table(predict=ypred, truth=testdat$y)
```

```
##         truth
## predict -1  1
##      -1 10  1
##      1   1  8
```

- 20개 중 18개를 맞추고 2개를 틀렸다.

Support vector classifier

- cost = 0.01일 때의 예측결과이다.

```
svmfit=svm(y~., data=dat, kernel="linear", cost=.01,scale=FALSE)
ypred=predict(svmfit,testdat)
table(predict=ypred, truth=testdat$y)
```

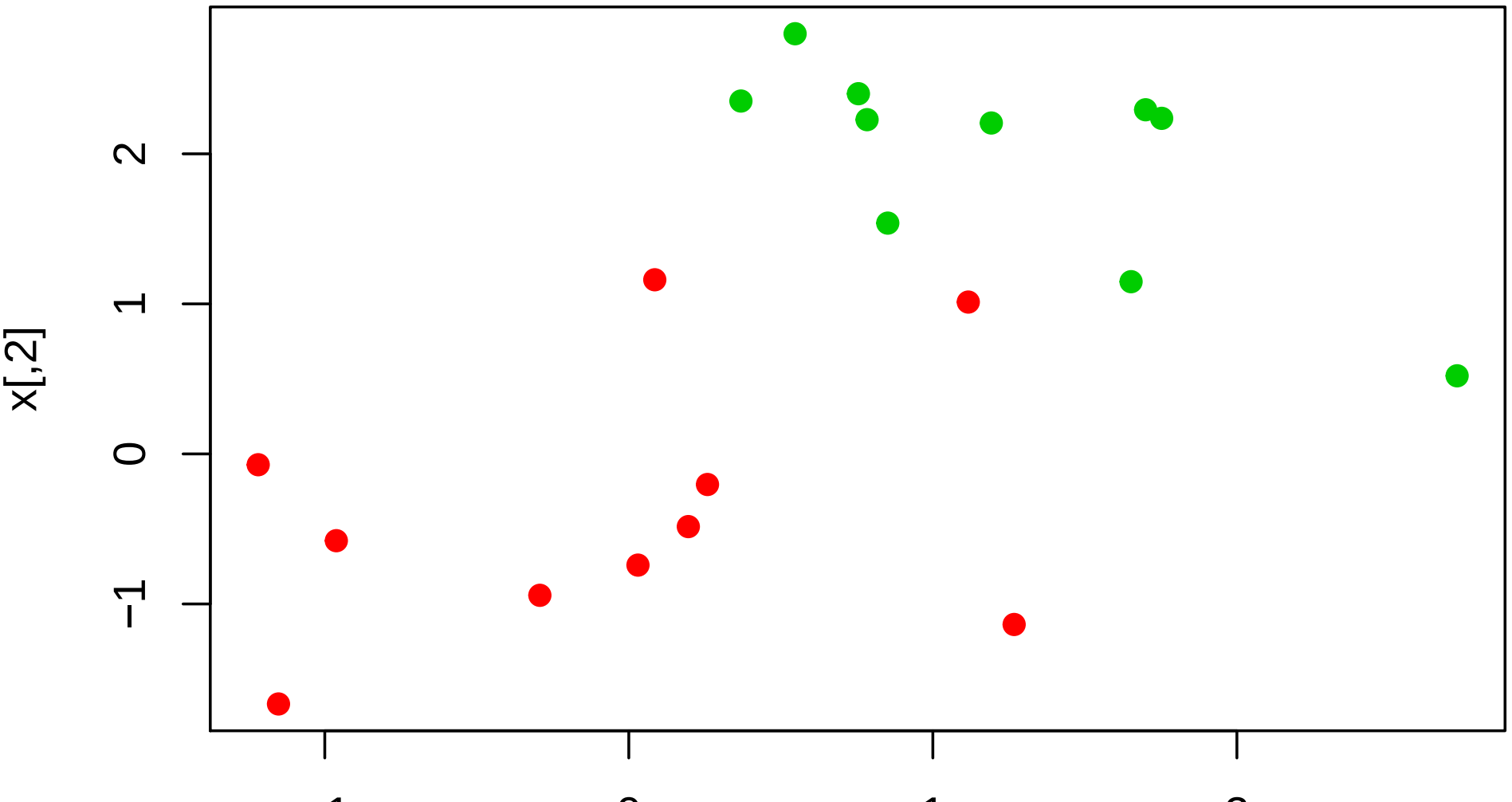
```
##         truth
## predict -1 1
##      -1  7 0
##      1   4 9
```

- 20개 중 16개를 맞추었다.

Support vector classifier

- 자료가 완전히 분리되는 경우

```
x[y==1,]=x[y==1,]+0.5
plot(x, col=(y+5)/2, pch=19)
```



Support vector classifier

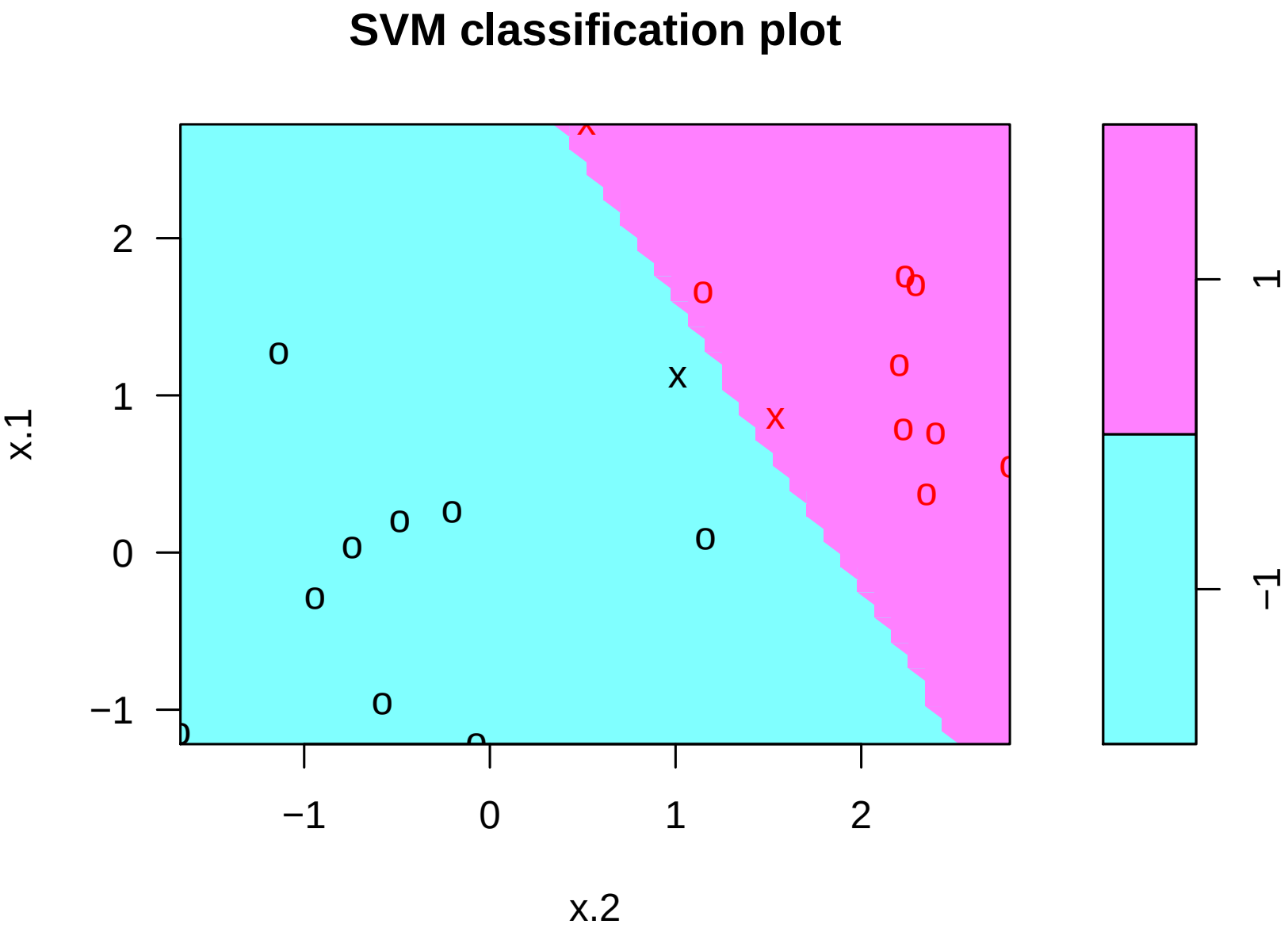
- 예산을 작게하면(cost를 크게하면) 여백이 작게 되어서 분리초평면을 구하게 된다.

```
dat=data.frame(x=x, y=as.factor(y))
svmfit=svm(y~., data=dat, kernel="linear", cost=1e5)
summary(svmfit)
```

```
##
## Call:
## svm(formula = y ~ ., data = dat, kernel = "linear", cost = 1e+05)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1e+05
##       gamma:  0.5
##
## Number of Support Vectors:  3
##
```

Support vector classifier

```
plot(svmfit, dat)
```



Support vector classifier

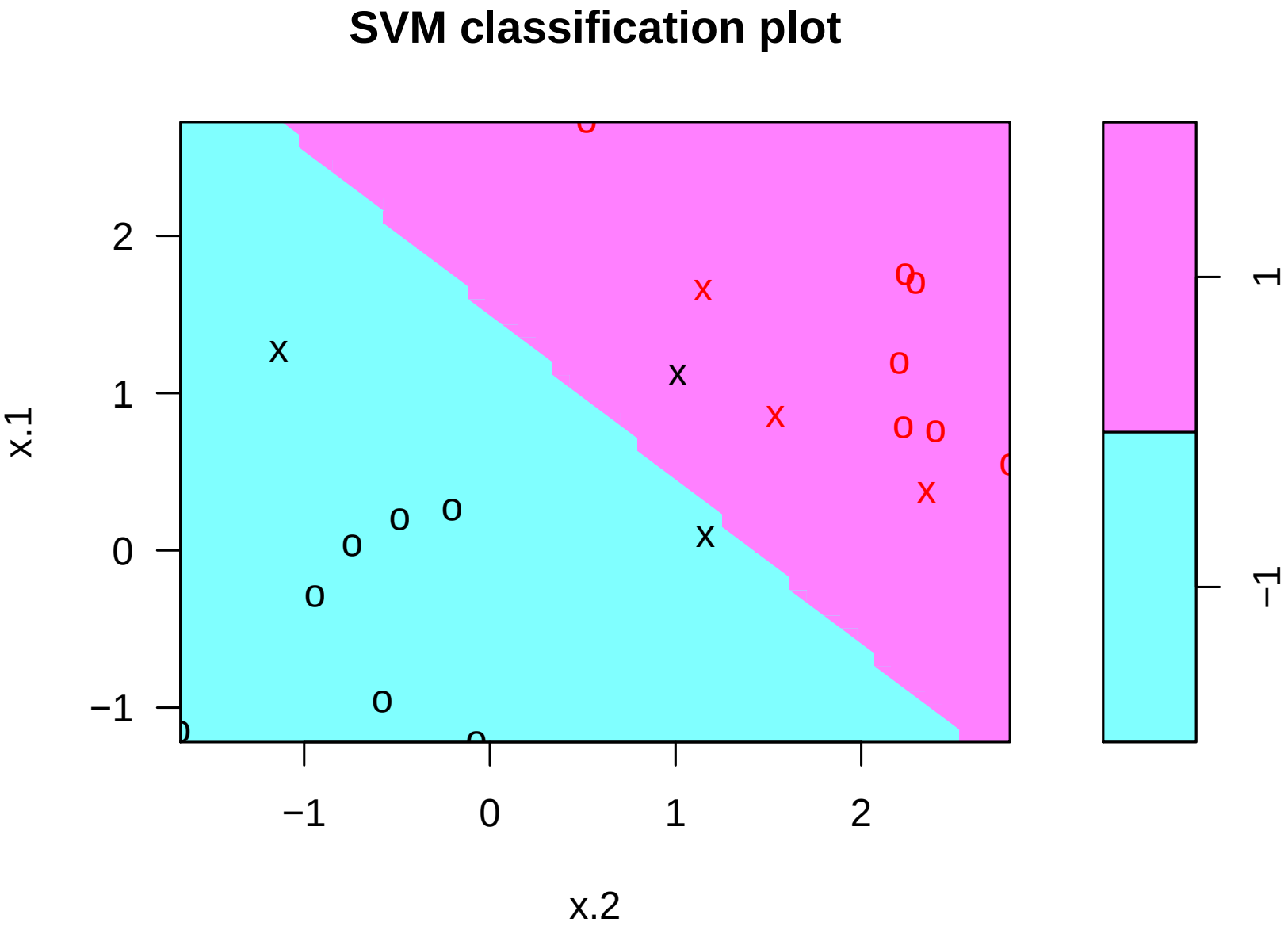
- 예산을 크게하면(cost를 작게하면) 여백이 크게 되어서 오분류하는 관측치가 생기게 된다.

```
svmfit=svm(y~., data=dat, kernel="linear", cost=1)
summary(svmfit)
```

```
##
## Call:
## svm(formula = y ~ ., data = dat, kernel = "linear", cost = 1)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##       gamma:  0.5
##
## Number of Support Vectors:  6
##
##  ( 3 3 )
```

Support vector classifier

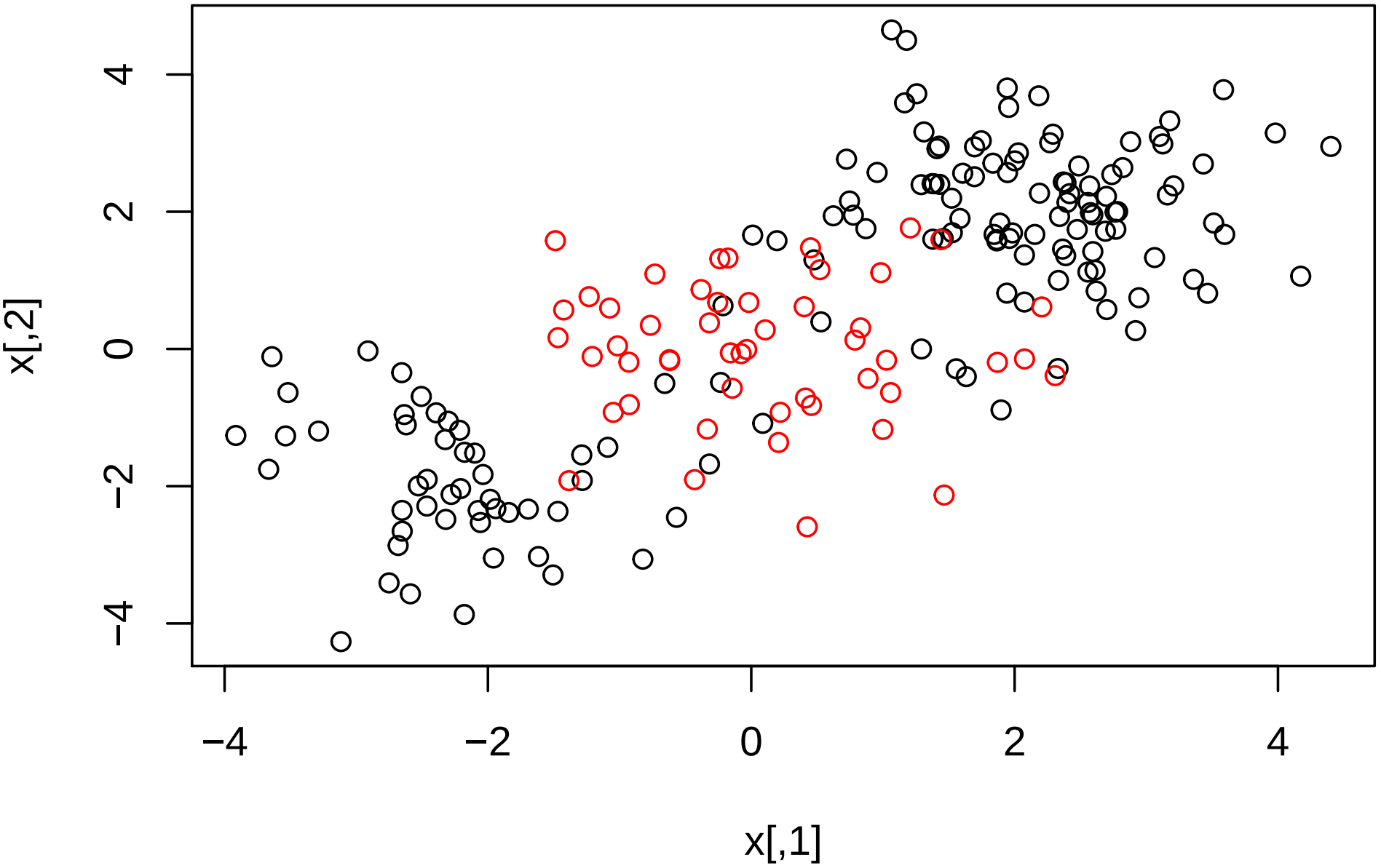
```
plot(svmfit,dat)
```



Support vector machine

```
set.seed(1)
x=matrix(rnorm(200*2), ncol=2)
x[1:100,]=x[1:100,]+2
x[101:150,]=x[101:150,]-2
y=c(rep(1,150),rep(2,50))
dat=data.frame(x=x,y=as.factor(y)); plot(x, col=y)
```

Support vector machine

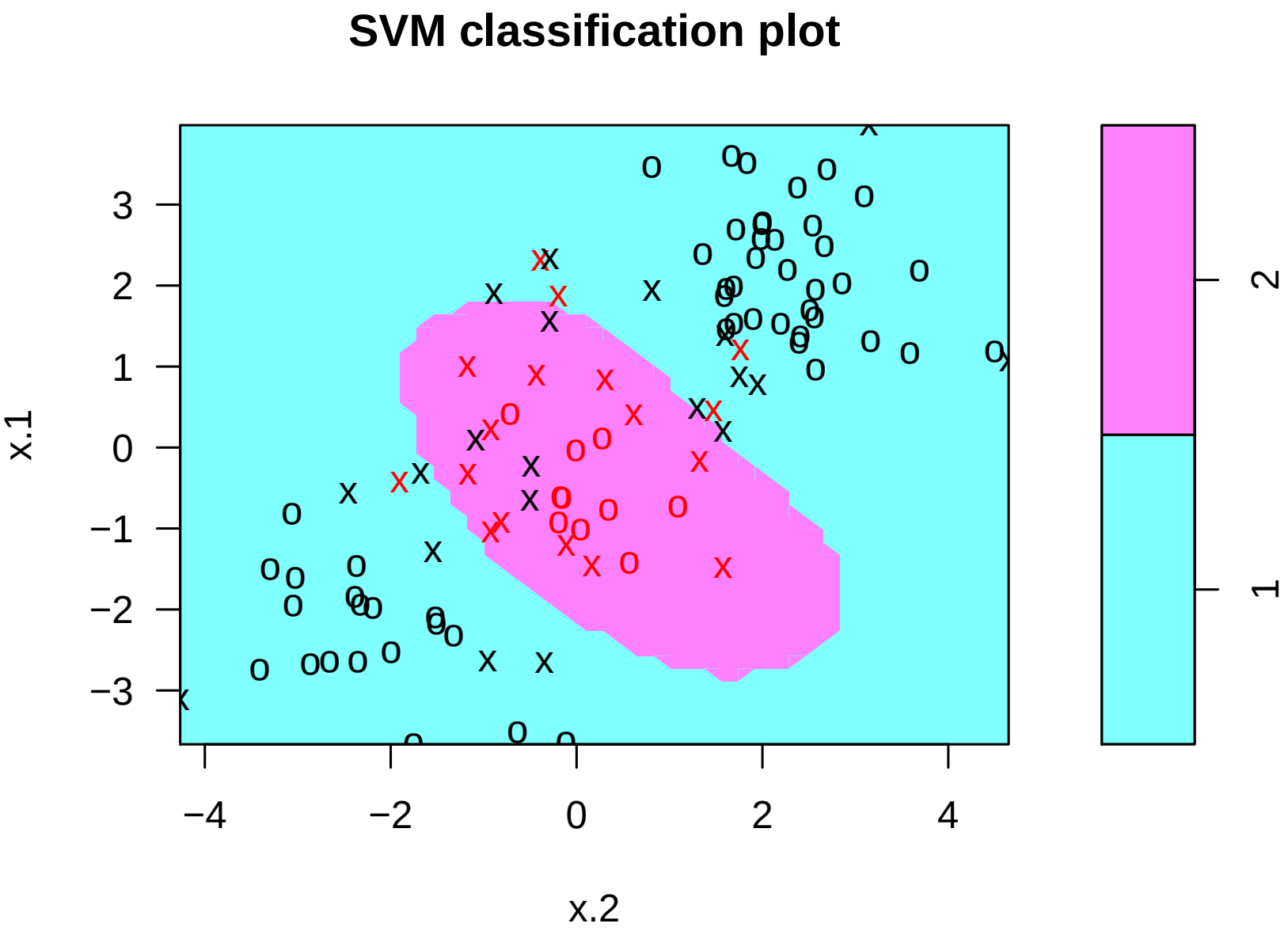


Support vector machine

- 지지벡터기계의 적합
  - svm 함수를 이용, 커널만 바꾸면 된다.

```
train=sample(200,100)
svmfit=svm(y~., data=dat[train,], kernel="radial", gamma=1, cost=1)
plot(svmfit, dat[train,])
```

Support vector machine



Support vector machine

```
summary(svmfit)
```

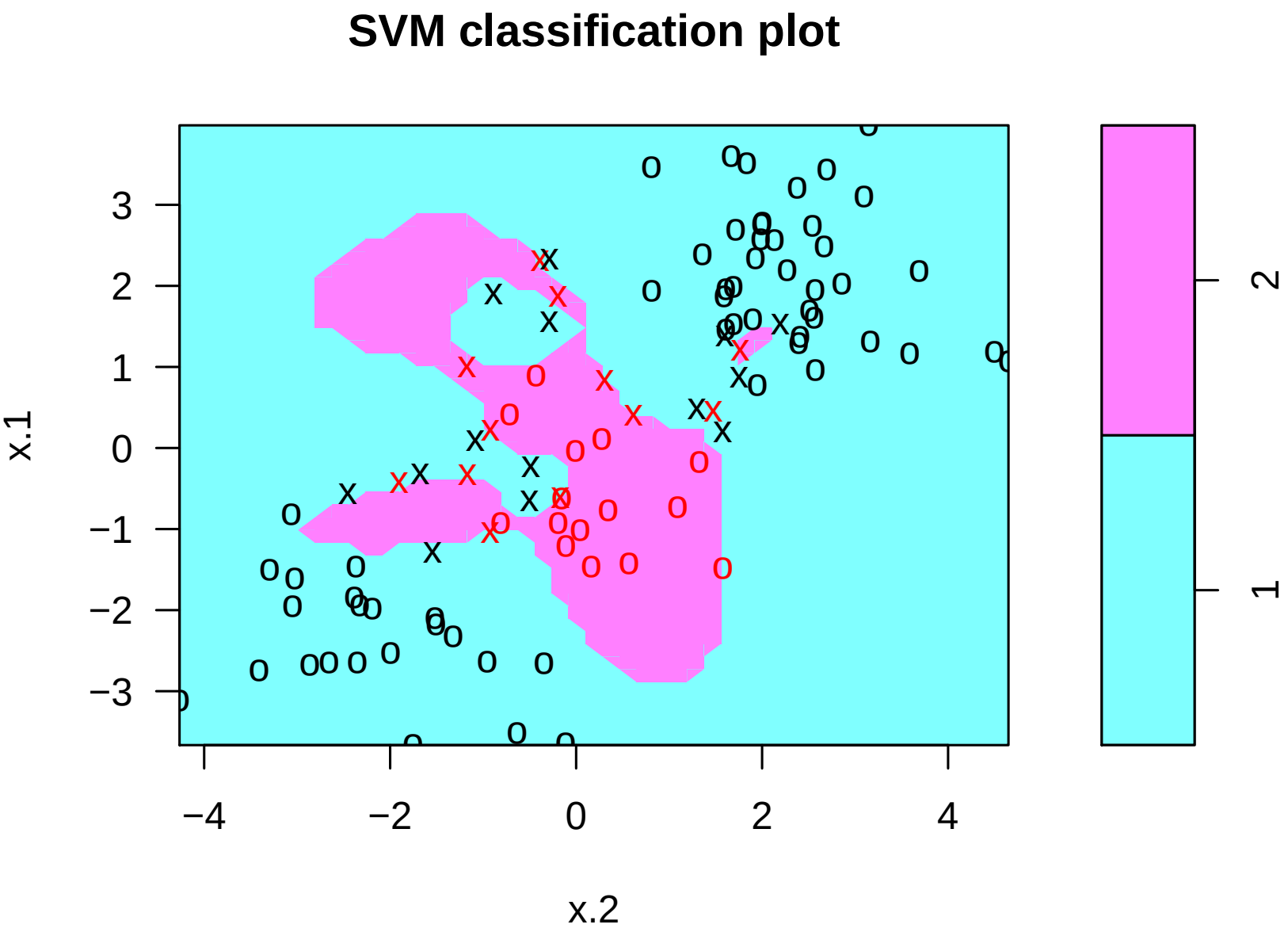
```
##
## Call:
## svm(formula = y ~ ., data = dat[train, ], kernel = "radial",
##     gamma = 1, cost = 1)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  1
##
## Number of Support Vectors:  37
##
##  ( 17 20 )
##
##
## Number of Classes:  2
```

Support vector machine

- cost의 값을 크게 하고 다시 적합하였다.

```
svmfit=svm(y~., data=dat[train,], kernel="radial",gamma=1,cost=1e5)
plot(svmfit,dat[train,])
```

Support vector machine



Support vector machine

- 비용과 감마의 값을 교차검증을 이용해서 결정한다.

```
set.seed(1)
tune.out=tune(svm, y~., data=dat[train,], kernel="radial",
              ranges=list(cost=c(0.1,1,10,100,1000),summary(tune.out
table(true=dat[-train,"y"],
      pred=predict(tune.out$best.model,newx=dat[-train,]))
```

```
##      pred
## true  1  2
##    1 54 23
##    2 17  6
```

Reference

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 6). New York: springer.