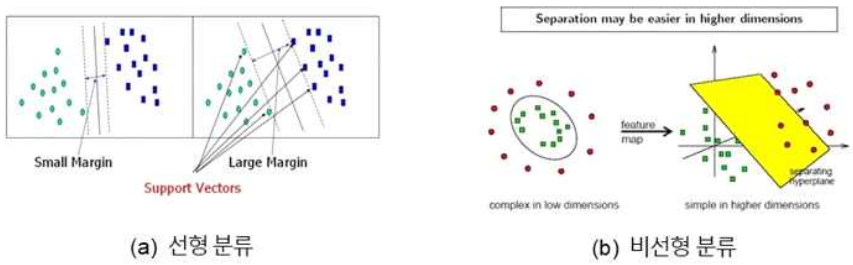
9장 서포트벡터머신(Support Vector Machines)

9.1 서론
서포트벡터머신(Support Vector Machines, 이하 SVM)모형은 고차원 또는 무한 차원의 공간에서 초평면을 찾아 이를 이용하여 분류와 회귀를 수행
SVM은 지도학습기법으로 비-중첩 분할을 제공하며 모든 속성을 활용하는 전역적분류 모형이다.
SVM은 최대 마진을 가지는 선형판별에 기초초하여 속성들 간의 의존성은 고려하지 않은 방법이다.

9.2 기초개념
자료를 잘 분리하는 초평면은 마진(초평면으로부터 가장 가까운 훈련용 자료까지의 거리)이 가장 큰 경우이며
마진이 가장 큰 초평면을 분류기로 사용할 때, 새로운 자료에 대한 오분류가 가장 낮아진다.
SVM모형은 선형분류뿐 아니라, 커널 트릭이라 불리는 다차원 공간상으로의 맵핑 기법을 사용하여 비선형분류도 효율적으로 수행한다.

9.2 기초 개념



(a) 선형 분류 (b) 비선형 분류

[그림9.1] SVM을 이용한 분류

9.3 SVM 알고리즘
d-차원 공간상에 n개의 점으로 구성된 분류 자료를 D={(x, y), i=1,2,3,...}이라고 할때,
y는 단지 두 개의 값(-1, +1)만을 취하는 분류 변수라고 하자.
아래의 초평면 h(x)는 d-차원에서 선형판별함수를 제공하며, 원래의 공간을 2개의 반-공간으로 나눈다.

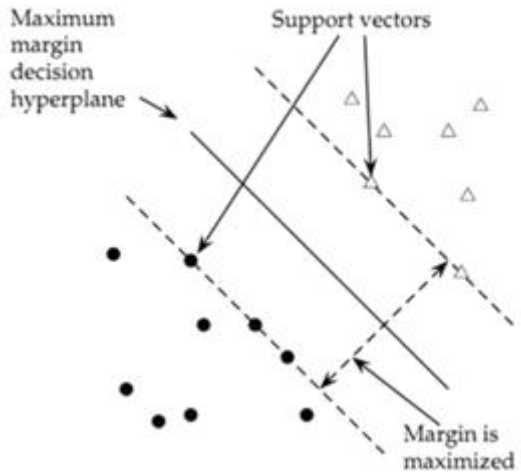
$$h(x) = w^T x + b = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b$$

위 식에서 w는 d-차원의 가중치 벡터이고, b는 편향 상수이다.
초평면 상의 점들은 h(x)=0이다.
즉, 초평면은 $w^t x = -b$ 를 만족하는 모든 점으로 정의된다.
만약 데이터 셋이 선형으로 분리가 가능한 경우. 분리 초평면은 h(x) < 0인 모든 점들은 -1의 군집으로
h(x) > 0인 모든 점들은 +1의 군집으로 분류되도록 구해질 수 있다.
이 경우 h(x)는 모든 자료에 대해 군집을 예측해주는 선형분류기 또는 선형판별의 임무를 수행한다.
더욱이 가중치 벡터 w는 초평면에 직교하므로, 그 방향에 수직인 방향을 제공하는 반면,
편향 b는 d-차원 공간에서 초평면의 오프셋을 고정한다.

분리초평면 h(x)=0이 주어질때, 각 자료점 x와 초평면 간의 거리는 다음과 같이 가능하다.

$$\delta_i = \frac{y_i h(x_i)}{\| w \|}$$

선형분류기의 마진은 모든 n개의 점에서 분리 초평면까지의 최소 거리로 정의된다.
이때 최소 거리를 가지는 모든 점(벡터 x)들을 선형분류기에 대한 서포트벡터라고 한다.
달리 말하자면, 서포트 벡터는 분리 초평면의 마진 상에 정확히 위치한다는 점이다.



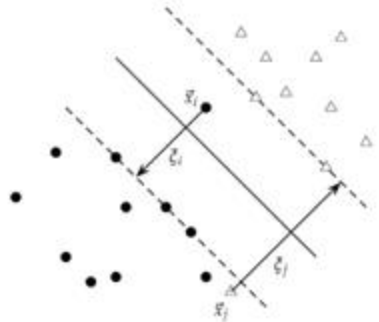
초평면의 정준 표현으로, y*를 가지는 각 서포트벡터에 대해 $y_i^* h(x_i^*)$ = 1이 된다. (= 마진이 1)
마찬가지로 서포트 벡터가 아닌 모든 점에 대해 $y_i h(x_i)$ > 1 이 된다. (= 모든 점들은 마진보다 크니까)
그 이유는 정의에 의해 이 점은 서포트 벡터보다 초평면으로부터 더 멀리 떨어져 있기 때문이다.
따라서 모든 자료 $x_i \in D$에 대해 $y_i h(x_i) \geq 1$ 이다

SVM의 기본 생각은 최대 마진을 가지는 초평면을 선택하는 것이다. 즉, 최적의 정준 초평면을 찾는 것이다.
이를 위해 모든 가능한 분리초평면 가운데 최대의 마진을 제공하는 가중치 벡터 w와 편향 b를 찾아야 한다.
즉, 그 초평면은 1/abs(w)을 최대로 하는 것이다.
이문제는 다음과 같이 선형제약 하에서 abs(w)를 최소화하는 문제와 동등하다.

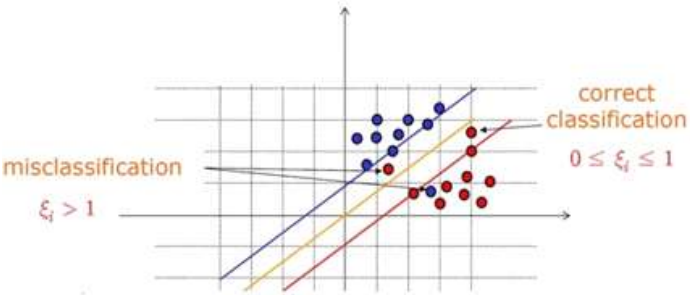
목적함수 : $\min \frac{\| w \|^2}{2}$

선형제약 : $y_i h(x_i) \geq 1, \forall x_i \in D$

위에 최적화 문제는 라그랑지 승수법을 이용하여 해결될 수 있다.
SVM은 군집들이 다소간 중복되어 완벽한 분리가 불가능하여 선형적으로 분리가 되지 않는 점들도,
각 점 $x_i \in D$에 대해 여유변수 ξ_i를 도입하여 다룰 수 있다.



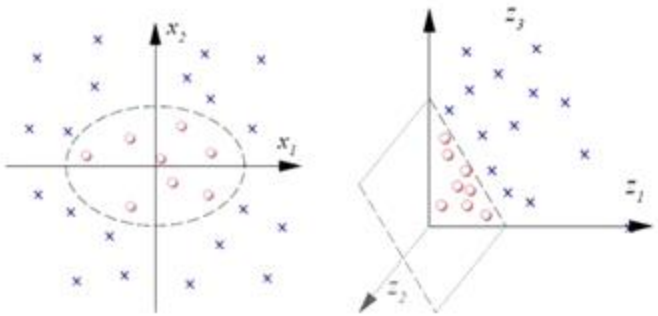
만약 $0 \le \xi_i \le 1$이면 그 점은 여전히 정확히 분류되고, 만약 $\xi > 1$이면 그 점은 오분류가 된다. (마진 1개까지 거리가 1)
즉, 이상치로 분류되는 몇 개의 점을 제외하고 초평면을 만들 수 있지만, 결국엔 이 초평면으로 각 요소가 분리되어야 한다.



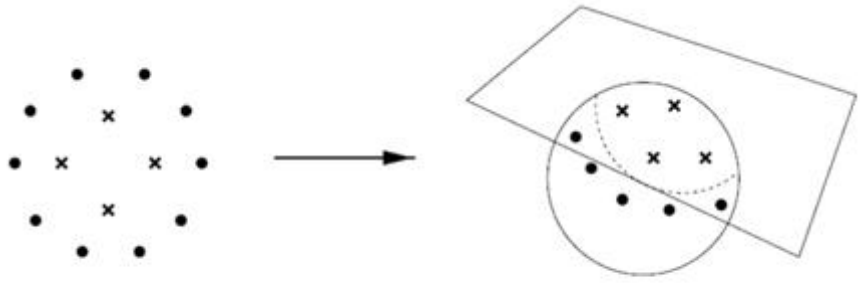
따라서 분류의 목적은 최대 마진을 가지며 동시에 여유변수의 합을 최소로 하는 초평면(w와 b)를 찾는 것이다.
이 문제 역시 라그랑지 승수법을 통해 해결할 수 있다.

SVM은 비선형결정경계를 가지는 문제도 해결할 수 있다.
핵심 아이디어는 원래의 d차원 공간을 d'차원 (d' > d)으로 맵핑하여 그 점들이 선형적으로 분리 가능하도록 하는 것이다.
원 데이터 셋 D = $\{(x_i, y_i), i = 1,2,...,n\}$와 변환함수 Φ가 주어질 때,
새로운 데이터 셋은 변환공간에서 $D_\phi = \{(\Phi(x_i), y_i), i = 1,2,\dots,n\}$ 으로 구해진다.
다음의 그림은 다항 맵핑을 사용한 예를 보여준다.

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



이제 선형결정면을 d'차원에서 구한 후, 이를 다시 원래의 d차원 공간상의 비선형면으로 맵핑한다.
이때 변환된 공간(d'차원)에서의 선형결정면은 $f(x) = w \bullet \Phi(x) + b$ 이며, 이 값의 부호를 통해 분류를 수행하게 된다.



(a) 맵핑 전 (b) 맵핑 후

이 과정에서 w와 b를 구하기 위해 맵핑된 결과 $\Phi(x)$는 별도로 계산될 필요가 없다.
그 이유는 w가 다음 같이 $w = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$ for some variables α 으로 표현될 수 있음이 밝혀져 있어
w를 직접 최적화하는 대신 α를 최적화할 수 있기 때문이다.
이때 판별규칙은 $f(x) = \sum_{i=1}^{n} \alpha_i \Phi(x_i) \bullet \Phi(x) + b$이 되며, $K(x_i, x) = \Phi(x_i) \bullet \Phi(x)$를 커널 함수라고 한다
따라서 변환된 공간에서 요구되는 유일한 연산은 내적으로, 이는 x_i와 x간의 커널함수(K) 와 함께 정의된다.
실제 Φ의 선택에 따라 맵핑된 공간의 차원이 무한 차원까지 커질 수 있어 Φ를 직접 이용하거나 맵핑된 자료의 내적 계산에 어려움이 있다.

이를 극복하는 방법은 다음과 같다. 두 점 p와 q에 대해 맵핑을 적용한 후, 그들의 내적을 계산하면 다음과 같다. ($\Phi(p) \bullet \Phi(q) = (p \bullet q)^2$)
위에 결과로부터 맵핑된 점들 간의 내적은 원 자료의 내적을 계산한 뒤 제곱을 취해 구할 수 있다.
따라서 Φ함수의 적용 없이 내적 $\Phi(p) \bullet \Phi(q)$ 을 계산할 수 있다.
맵핑 공간에서의 내적과 동등한 함수를 커널 함수하고 하고, 이를 K라고 표현한다.

SVM에서 통상적으로 사용되는 커널은 다음과 같다.
가우시안과 라플라스 RBF, 베셀 커널은 자료에 관한 사전정보가 없을 때 사용되는 일반적 목적의 커널이다.
가우시안 커널 : $K(x_i, x_j) = e^{-\frac{\| x_i - x_j \|^2}{2\sigma^2}}$, σ : 퍼진 정도 또는 표준편차
가우시안 RBF 커널 : $K(x_i, x_j) = e^{-\gamma \| x_i - x_j \|^2}, \gamma \geq 0$
라플라스 RBF 커널 : $K(x_i, x_j) = e^{-\gamma \| x_i - x_j \|}, \gamma \geq 0$
제1종 베셀함수 커널 : $K(x_i, x_j) = \left(\frac{Bessel_{\nu+1}^{n}(\sigma \| x_i - x_j \|)}{(\| x_i - x_j \|)^{-n(\nu+1)}} \right)$
역탄젠트 커널 : $K(x_i, x_j) = \tanh(x_i^T x_j + offset)$

선형 커널은 문서분류 등에서 자주 발생하는 대량의 희박 자료벡터를 다룰 때 유용하다.
선형(linear) 커널 : $K(x_i, x_j) = x_i^T x_j$

다항 커널은 이미지 처리에 자주 사용되며,
다항 커널 : $K(x_i, x_j) = (x_i^T x_j + 1)^q$, q=다항식의 차수

시그모이드 커널은 신경망에 대한 프록시로 주로 사용된다.
시그모이드 커널 : $K(x_i, x_j) = \tanh(\alpha x_i^T x_j + offset)$

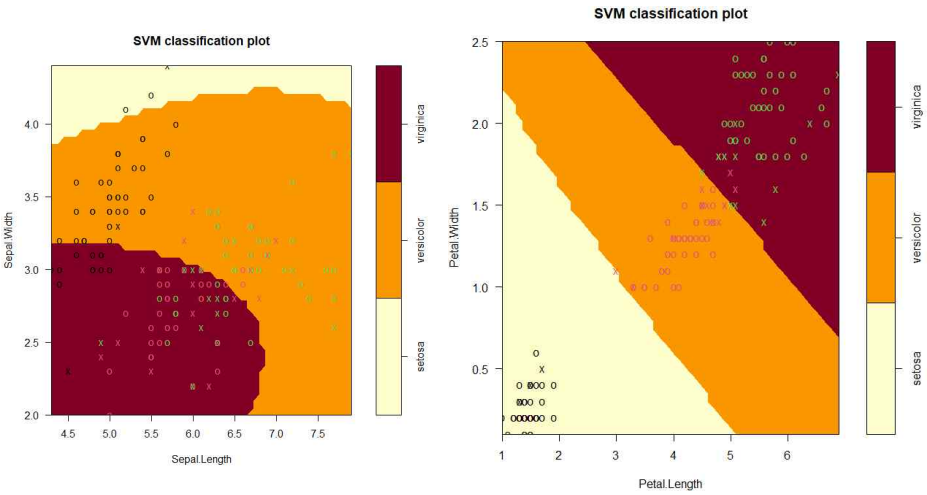
스플라인과 ANOVA RBF 커널은 전형적으로 회귀 문제에 잘 수행된다.
ANOVA Radial bssis 커널 : $K(x_i, x_j) = \left(\sum_{k=1}^{n} e^{-\sigma (x_i^k - x_j^k)^2} \right)^d$
일차원의 선형스플라인 커널 : $K(x_i, x_j) = 1 + x_i x_j \min(x_i, x_j) - \frac{x_i + x_j}{2} \min(x_i, x_j)^2 + \frac{\min(x_i, x_j)^3}{3}$

SVM을 수행하는 R 패키지는에는 {e1071}, {kernlab}, {klaR}, {svmpath}, {shogun} 등이 있다.

예제 1 : {e1071}svm() 함수를 이용한 SVM 분류를 수행
데이터 불러오기
```
library("e1071")
data(iris)
```

{e1071}svm() 함수를 사용하여 SVM 분류를 수행
type : svm()의 수행 방법을 정한다.디폴트 값으로 "C-classification", "eps-regression" 있고, 다른 값들로도 설정가능하다.
kernel : 훈련과 예측에 사용되는 커널로, "radial"옵션은 가우시안 RBF를 의미. 커널 종류가 결과의 정확도에 큰 영향을 주지 않는다.
degree : 다항 커널이 사용될 경우의 모수(차수) 이다.
gamma : 선형을 제외한 모든 커널에 요구되는 모수로, 디폴트는 1/(데이터 차원) 이다.
coef0 : 다항 또는 시그모이드 커널에 요구되는 모수로, 디폴트는 0이다.
cost : 제약 위배의 비용으로, 디폴트는 1이다.
cross : k-중첩 교차타당도에서 k값을 지정한다.
```
svm.e1071 <- svm(Species ~ . , data = iris, type = "C-classification", kernel = "radial", cost = 10, gamma = 0.1)
summary(svm.e1071) # 결과 요약
# 결과
# Parameters:
#    SVM-Type:  C-classification
# SVM-Kernel:  radial
# cost:  10
#
# Number of Support Vectors:  32
## ( 3 16 13 )
```

Plot함수를 이용하여 그 결과에 대한 시각화할 수 있다.
나머지 변수의 지정된 값에서의 단면(2차원 그림)으로, x: 서포트벡터, o: 자료점 이다.
```
plot(svm.e1071, iris, Petal.Width ~ Petal.Length, slice = list(Sepal.Width = 3, Sepal.Length = 4))
plot(svm.e1071, iris, Sepal.Width ~ Sepal.Length, slice = list(Petal.Width = 2.5, Petal.Length = 3))
```



predict() 함수를 이용하여 새로운 데이터에 대해 분류를 진행한다.
setosa, virginica는 50개 모두 잘 분류 되었으며, versicolor은 50개 중 47개가 잘 분류 되었다.
```
pred <- predict(svm.e1071, iris, decision.values = TRUE)
acc <- table(pred, iris$Species) # table로 형변환
acc
# 출력
# pred          setosa versicolor virginica
# setosa          50          0         0
# versicolor       0         47         0
# virginica        0          3        50
```

classAgreement() 함수를 통해 모형의 정확도를 확인할 수 있다.
```
classAgreement(acc)
# 출력
# $diag [1] 0.98 -> 1 - (3/150) = 0.98
# $kappa [1] 0.97 ->
# $rand [1] 0.9739597
# $crand [1] 0.941045
```

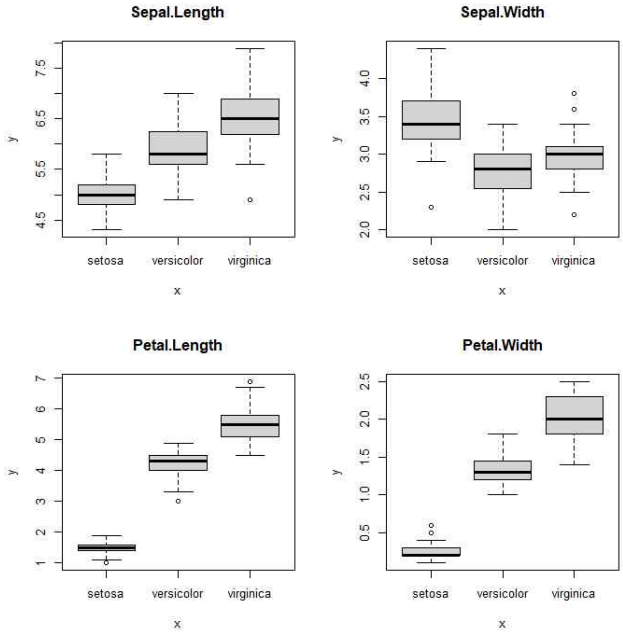
tune.svm()함수는 제공된 모수 영역에서 격자 탐색을 사용하여 통계적 방법의 초모수를 조율할 수 있다.
# 이 함수는 최적의 모수를 제공해주며, 동시에 여러 모수값에 대한 검정에 대한 자세한 결과를 제공해준다.
# 6 * 2 = 12개 조합에서 모수 조율이 이루어진다.
tuned <- tune.svm(Species~., data = iris, gamma = 10^(-6:-1), cost = 10^(1:2))
summary(tuned)

```
# 출력
# - best parameters:
#    gamma cost  -> 적합이 가장 좋은 모수는 gamma가 0.01, cost가 10
#    0.01    10
# - best performance: 0.04    -> 적합이 가장 좋은 초모수는 에러가 0.04일 때를 말한다.
# - Detailed performance results:
#       gamma cost      error dispersion
# 4   1e-03    10 0.12000000 0.07568616
# 5   1e-02    10 0.04000000 0.04661373 -> 에러가 가장 낮은 것중 처음인 경우로 적합
# 6   1e-01    10 0.04666667 0.05488484
# 7   1e-06   100 0.78000000 0.06324555
# 8   1e-05   100 0.65333333 0.14673399
# 9   1e-04   100 0.12000000 0.07568616
# 10 1e-03   100 0.04000000 0.04661373
# 11 1e-02   100 0.04000000 0.04661373
# 12 1e-01   100 0.05333333 0.04216370
```

예제 2 : {kernlab}ksvm()함수를 이용하여 svm 분류를 수행한다.
데이터 불러오기
```
library("kernlab")
data(iris)
```

{kernlab}ksvm()함수를 이용하여 svm 분류를 수행
```
svm.kernlab <- ksvm(Species ~ ., data=iris,          type="C-bsvc", kernel="rbfdot", kpar=list(sigma=0.1), C=10, prob.model=TRUE)
svm.kernlab
# 결론 : Support Vector Machine object of class "ksvm"
#
# SV type: C-bsvc  (classification)
# parameter : cost C = 10
#
# Gaussian Radial Basis kernel function.
# Hyperparameter : sigma =  0.1
#
# Number of Support Vectors : 32
#
# Objective Function Value : -5.8442 -3.0652 -136.9786
# Training error : 0.02
# Probability model included.
```

plot() 함수를 이용하여 분류된 결과에 대한 각 변수별 분포를 상자 그림의 형태로 나타낼 수 있다.
```
fit <- fitted(svm.kernlab) # svm 분류를 한 데이터를 적합한다.
par(mfrow=c(2,2))
plot(fit, iris[,1], main="Sepal.Length")
plot(fit, iris[,2], main="Sepal.Width")
plot(fit, iris[,3], main="Petal.Length")
plot(fit, iris[,4], main="Petal.Width")
par(mfrow=c(1,1))
```



predict()함수를 통해 새로운 자료에 대한 분류(예측)을 수행
```
# 여기서는 모형 구축에 사용된 자료를 재사용하여 분류를 수행하였다.
# 그 결과 setosa와 viginica는 50개 모두,
# versicolor는 50개 중 47개가 제대로 분류되었다.
head(predict(svm.kernlab, iris, type= "probabilities")) # 각 집단에 속할 확률
# 결과
# setosa   versicolor    virginica
# [1,] 0.9829708 0.009665580 0.007363588
# [2,] 0.9776135 0.014055228 0.008331223
# [3,] 0.9847721 0.008004313 0.007223543
# [4,] 0.9811618 0.010493190 0.008345026
# [5,] 0.9840873 0.008694829 0.007217860
# [6,] 0.9705965 0.019267779 0.010135686
head(predict(svm.kernlab, iris, type = "decision"))
# 결과
# [,1]       [,2]       [,3]
# [1,] -1.398258 -1.185073 -4.223584
# [2,] -1.275065 -1.148625 -4.225240
# [3,] -1.460398 -1.191025 -3.886884
# [4,] -1.371812 -1.149203 -3.782832
# [5,] -1.433083 -1.191048 -3.933006
# [6,] -1.171098 -1.091091 -3.839142
head(predict(svm.kernlab, iris, type = "response"))
# 결과
# [1] setosa setosa setosa setosa setosa setosa
# Levels: setosa versicolor virginica
table(predict(svm.kernlab, iris), iris[,5])
# 결과
# setosa          50          0          0
# versicolor       0         47          0
# virginica        0          3         50
```

예제 3 : {e1071}의 svm()함수를 이용하여 서포트벡터회귀(SVR)를 수행
분석용 자료 생성
x <-c(1:20)
y <- c(3,4,8,4,6,9,8,12,15,26,35,40,45,54,49,59,60,62,63,68)
data<-data.frame(x, y)

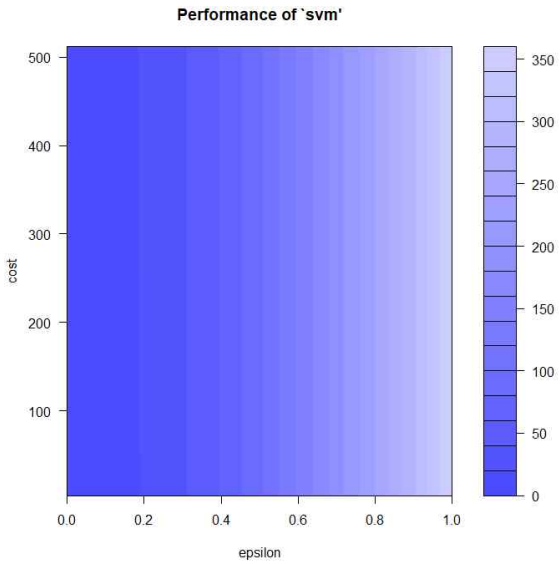
서포트벡터회귀(SVR)와 성능을 비교하기 위해 기본 회귀(lm)을 먼저 적합한다.
RMSE = 5.70
plot(data, pch=16) # 그림에 표시
model <- lm(y ~ x, data) # lm를 통해 단순회귀 적합
abline(model) # 적합된 회귀직선을 그린다.

기본회귀직선의 RMSE
lm.error <- model$residuals # 실제값과 회귀값 사이의 오차를 저장
lmRMSE <- sqrt(mean(lm.error^2)) # 오차제곱평균(MSE)의 루트를 씌워서 RMSE를 구한다.
lmRMSE
결과 : [1] 5.703778

svm()함수를 통해 서포트벡터회귀(SVR)를 수행한다.
# 기본 회귀적합보다 감소된 RMSE = 3.16의 결과를 얻을 수 있었다.
model <- svm(y ~ x , data) # 서포트벡터회귀(SVR)를 수행
pred.y <- predict(model, data) # 데이터에 대해 회귀를 적합한다.
points(data$x, pred.y, col = "red", pch=4) # 적합된 값들을 그래프에 X표시로 맵핑한다.

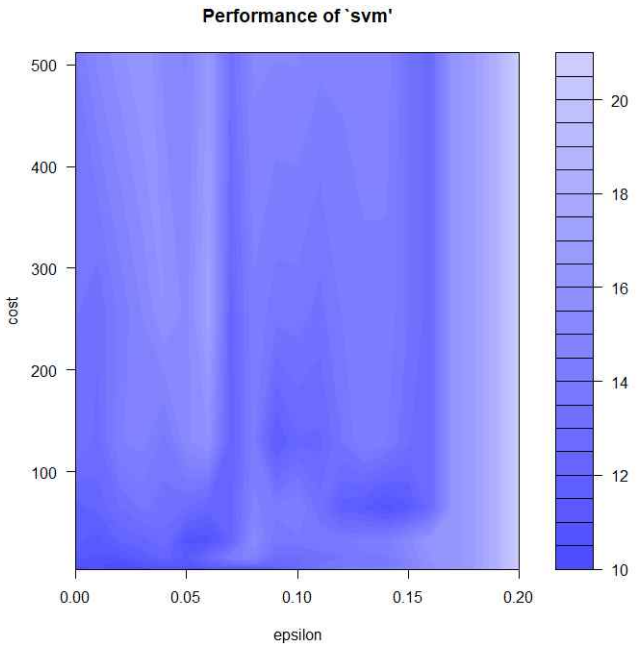
서포트벡터회귀(SVR)직선의 RMSE
error <- data$y - pred.y # 실제값과 회귀값 사이의 오차를 저장
svmRMSE <- sqrt(mean(error^2)) # 오차제곱평균(MSE)의 루트를 씌워서 RMSE를 구한다.
svmRMSE
결과 : [1] 3.157061

초모수 최적화 또는 모형선택
# SVR의 성능을 높이기 위해 모델의 모수들을 최적화할 필요가 있다.
# 과적합을 막기위해 cost 모수를 변경할 수 있으며, 이렇게 모수의 선택과정을 초모수 최적화 또는 모형선택이라고 한다.
# 적용된 svm() 함수는 디폴트로 e회귀가 적용되었으며, e값은 디폴트로 0.1이 사용되었다.
# 모수 e와 cost의 다양한 값에 대해, 격자 탐색을 수행하여 최적의 모수를 찾는다.
# e * cost = 11*8 = 88개의 모형에 대해 훈련이 수행되었고,
# 최적의 모수로 e = 0, cost = 4이 선정되었다.
# tune()함수는 각모형에 대한 MSE 값을 제공한다.(제곱근을 취해 RMSE 값을 구할 수 있다.)
tuneResult <- tune(svm, y ~ x, data = data, ranges = list(epsilon = seq(0,1,0.1), cost = 2^(2:9)))
tuneResult
결과 : Parameter tuning of 'svm':
#
- sampling method: 10-fold cross validation
- best parameters:
epsilon cost
0 128
#
- best performance: 7.843891



plot()함수를 통해 RMSE를 시각화, 색이 짙어질수록
# RMSE가 0에 가까우므로 더 나은 모형임을 뜻한다.
# cost는 잘 확인이 안되지만, e값은 확실히 작아질수록 좋은 모형이 되고 있다.
plot(tuneResult)

보다 정교한 모수 조율을 위해 보다 좁은 영역에서 격자탐색을 실시한다.
# 위 그림에서 cost 모수의 영향은 크지 않으므로 그 값을 그대로 유지하였다. 총 21*8=168개의 모델이 적합되었다.
tuneResult <- tune(svm, y ~ x, data=data, ranges = list(epsilon=seq(0,0.2,0.01), cost=2^(2:9)))
tuneResult # 최적의 모델로 e = 0.02, cost = 4가 선정되었다.
결과 : Parameter tuning of 'svm':
#
- sampling method: 10-fold cross validation
- best parameters:
epsilon cost
0.02   4
- best performance: 10.05218



plot()함수를 통해 RMSE를 시각화, 색이 짙어질수록
# RMSE가 0에 가까우므로 더 나은 모형임을 뜻한다.
# 해당 그림에서 가장 짙은 곳인 e = 0.08, cost = 4가
# 가장 좋은 모형임을 뜻한다.
plot(tuneResult)

새로운 데이터에 예측을 수행
# 최적의 모형은 분석자가 아닌 컴퓨터가 수행을 해주며, 최적의 모형을 통해 예측을 수행할 수 있다.
tunedModel <- tuneResult$best.model # 최적의 모형 찾기
# Parameters:
# SVM-Type: eps-regression
# SVM-Kernel: radial
# cost: 4
# gamma: 1
# epsilon: 0.02
Number of Support Vectors: 18

predict()함수를 이용하여 예측을 수행
tpred.y <- predict(tunedModel, data) # 새로운 데이터에 최적의 모형을 이용하여 예측을 수행

최적의 모수를 적용한 서포트벡터회귀(SVR)직선의 RMSE
error <- data$y - tpred.y # 예측 오차를 저장
tsvmRMSE <- sqrt(mean(error^2)) # 평균제곱오차에 루트를 씌운 RMSE를 구한다.
tsvmRMSE # RMSE = 2.101462 -> 성능이 크게 향상되었다.
결과 : [1] 2.252056

그림으로 표현
모수에 대한 조율 전과 후의 SVR을 적합한 결과를 그림으로 그려보면 다음과 같다.
그림에서 푸른색(진한선)이 가장 원 데이터의 추세를 잘 표현하고 있다.
plot(data, pch=16) # 검은색 점 = 원래 데이터
points(data$x, pred.y, col = "red", pch=4, type="b") # 빨간 X = 모수 조율 전 SVR을 적합한 회귀선
points(data$x, tpred.y, col = "blue", pch=4, type="b") # 파란 X = 모수 조율 후 SVR을 적합한 회귀선