svm.R

SANGHOOJEFFREY

Thu Jun 28 03:10:38 2018

```
# 서포트 벡터 머신은 서로 다른 분류에 속한 데이터 간에 간격이 최대가 되는 선을 찾아
 데이터를 분류하는 방법

# 서포트 벡터란?

# 커널 트릭을 이용하여 주어진 데이터를 적절한 고차원으로 옮긴 후 변환된 차원에서 서
포트 벡터머신을 이용해
# 초편면을 찾는 것. 이를 위해선 벡터 간 내적 계산에 있다.
# 이 함수를 이용하면 마치 데이터를 고차원으로 옮긴 듯한 효과를 일으키면서도 데이터
를 고차원으로 옮기는 데 따른
# 계산 비용 증가를 피할 수 있다.

# SVM 모델을 위한 팩키지로 e1071 과 kernlab 등이 있다.
# e1071 은 효율적인 SVM 구현체로 알려진 libsvm 을 R 에서 사용할 수 있도록 만든 팩키
지
# kernlab 은 커널 기반의 기계학습 알고리즘을 R 에서 구현

if(!require(kernlab)) install.packages("kernlab"); library(kernlab)

## Loading required package: kernlab

ksvm.out <-ksvm(Species ~., data=iris)
ksvm.out

## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Gaussian Radial Basis kernel function.
##  Hyperparameter : sigma =  0.812922905871174
##
## Number of Support Vectors : 59
##
```

```
## Objective Function Value : -4.5651 -5.0881 -20.2667
## Training error : 0.026667

predicted1 <- predict(ksvm.out, newdata=iris)

xtabs(~predicted1+iris$Species)

##             iris$Species
## predicted1   setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          2        48

# 기본적인 커널함수로 가우시안 커널을 사용한다 만약 vanilladot(특별한 변환없이 내적 계산)을 지정할 수도 있다.
ksvm.out <-ksvm(Species ~., dat=iris, kernel="vanilladot")

##  Setting default kernel parameters

ksvm.out

## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
##
## Linear (vanilla) kernel function.
##
## Number of Support Vectors : 29
##
## Objective Function Value : -0.9818 -0.322 -17.0644
## Training error : 0.033333

predicted2 <- predict(ksvm.out, newdata=iris)

xtabs(~predicted2+iris$Species)

##             iris$Species
## predicted2   setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         46         1
##   virginica       0          4        49

ksvm.out <-ksvm(Species ~., data=iris, kernel="polydot", kpar=list(degree=3))
ksvm.out

## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 1
```

```
##
## Polynomial kernel function.
##  Hyperparameters : degree =  3  scale =  1  offset =  1
##
## Number of Support Vectors : 22
##
## Objective Function Value : -0.0252 -0.0225 -6.3396
## Training error : 0.013333
```

```
predicted3 <- predict(ksvm.out, newdata=iris)
```

```
xtabs(~predicted3+iris$Species)
```

```
##             iris$Species
## predicted3   setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         49         1
##   virginica       0          1        49
```

```
# SVM 을 잘 사용하려면 파라메터를 잘 찾아야 하고 이를 위해선 교차 검증이 필수적
# e1071 에서는 tune()함수를 사용해 모델을 튜닝할 수 있다.
```

```
if(!require(e1071)) install.packages("e1071"); library(e1071)
```

```
## Loading required package: e1071
```

```
r.tune<-tune.svm(Species~., data = iris, gamma = 2^(-1:1), cost = 2^(2:4)) #
refer to ?tune
attributes(r.tune)
```

```
## $names
## [1] "best.parameters"  "best.performance" "method"
## [4] "nparcomb"         "train.ind"        "sampling"
## [7] "performances"     "best.model"
##
## $class
## [1] "tune"
```

```
r.tune$best.parameters
```

```
##   gamma cost
## 1   0.5    4
```

```
r.tune$best.model
```

```
##
## Call:
## best.svm(x = Species ~ ., data = iris, gamma = 2^(-1:1), cost = 2^(2:4))
##
##
```

```
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  4
##       gamma:  0.5
##
## Number of Support Vectors:  49

# 이 외에도 NaiveBayes, nnet (Neural Network), H2o 등의 팩키지 및 함수가 존재
```