데이터과학을 위한 R프로그래밍

9주차. k-인접기법과 판별분석

이혜선 교수
포항공과대학교 산업경영공학과

목차

9주차. k-인접기법과 판별분석

9주차

1차시

k-인접기법

POSTECH

데이터마이닝 기법

모형화	특징	내용	적용기법
예측	✠ 타겟변수 값이 주어지는 경우 (supervised learning) ✠ 변수간의 관계	주어진 데이터를 기반으로 모델을 만든 후, y값을 예측 (y=continuous value)	✠ 다중회귀분석 ✠ 주성분 회귀분석 ✠ 부분최소자승법 ✠ 신경망
분류		학습표본을 기반으로 분류규칙을 생성. 분류규칙의 성능을 검증하기 위해 실제범주와 추정된 범주를 비교 (y=0/1 혹은 다범주)	✠ 로지스틱 회귀모형 ✠ 의사결정나무 ✠ 선형판별분석 ✠ 서포트벡터머신
군집	✠ 타겟변수 값이 없는 경우 (unsupervised learning) ✠ 개체간의 관계	주어진 데이터(X변수들)의 속성으로 군집화	✠ 계층형 군집 분석 ✠ K-MEANS
연관규칙		연관성 있는 변수 관계 도출(동시 발생 빈도 분석)	✠ 연관규칙 분석

POSTECH

분류(Classification)

- 분류(Classification) – 지도학습(Supervised Learning). 타겟범주를 알고 있는 데이터로 분류규칙을 생성하고 새로운 데이터를 특정범주에 분류하는 기법
- 군집화(Clustering) – 비지도학습(Unsupervised learning). 독립변수들의 속성을 기반으로 객체들을 그룹화하는 방법

POSTECH

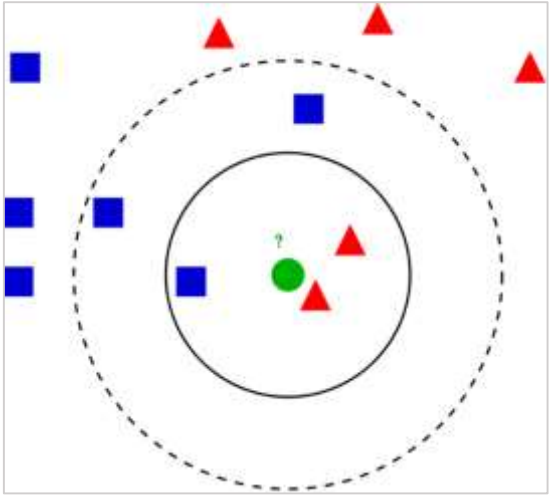
k-인접기법(k-nearest neighbor method)

k개의 가장 가까운 이웃들을 사용해서 분류하는 방법

- k개의 인접객체를 고려할 때, 새로운 객체 ●는 어느 범주로 할당?

만약 k=3로 정하면, 새로운 객체는 ▲의 범주로 분류

만약 k=5로 정하면, 새로운 객체는 ■의 범주로 분류



출처 Wikipedia, the free encyclopedia 참조

POSTECH

k-인접기법(k-nearest neighbor method)

최적 k는?

- k가 너무 크면 데이터 구조를 파악하기 어렵고, 너무 작으면 과적합(overfitting) 위험이 있음
- 교차검증(cross-validation)으로 정확도가 높은 k를 선정
- 장점 : 계산이 단순하고 효율적
- 단점 : 범주형변수에 대해서는 거리를 계산할 수 없음

POSTECH

k-인접기법(k-nearest neighbor method)

kNN 을 수행하기 위한 추가 패키지 설치

```
# lec9_1_knn.R
# Classification
# k-Nearest Neighbor

# packages
install.packages("class")# for knn
install.packages("caret")#for confusion matrix
install.packages("scales")#for graph
library(class)
library(caret)
library(scales)
```

- kNN 수행을 위한 패키지 : “class”
- Confusion matrix를 생성하기 위한 패키지 : “caret”
- 최적 k 등 그래프를 위한 패키지 : “scales”

POSTECH

train/test 데이터 분할(cross-validation)

☑ Iris 데이터(데이터 불러들이기, 학습데이터와 검증데이터의 분할)

```
# read csv file
iris<-read.csv("iris.csv", stringsAsFactors = TRUE)
# head(iris)
str(iris)
attach(iris)
```

데이터 불러들이기

```
# training/ test data : n=150
set.seed(1000)
n <- nrow(iris)
# train set 100, test set 50
tr.idx <- sample.int(n, size = round(2/3* n))
```

데이터분할(학습데이터 100, 검증데이터 50개)

```
# attributes in training and test
iris.train<-iris[tr.idx,-5]
iris.test<-iris[-tr.idx,-5]
```

iris.train(독립변수4개를 포함한 100개의 데이터)
iris.test(독립변수4개를 포함한 50개의 데이터)

trainLabels(학습데이터의 타겟변수)
testLabels(검증데이터의 타겟변수)

POSTECH

kNN의 수행과 결과

kNN함수 : knn(train=학습데이터, test=검증데이터, cl=타겟변수, k=)

```
# knn (5-nearest neighbor)
md1<-knn(train=iris.train,test=iris.test,cl=trainLabels,k=5)
md1
```

```
> md1
 [1] setosa     setosa     setosa
 [4] setosa     setosa     setosa
 [7] setosa     setosa     setosa
[10] setosa     setosa     setosa
[13] setosa     setosa     setosa
[16] setosa     versicolor versicolor
[19] versicolor versicolor versicolor
[22] versicolor versicolor versicolor
[25] versicolor versicolor versicolor
[28] versicolor versicolor versicolor
[31] versicolor versicolor virginica
[34] virginica  virginica  virginica
[37] virginica  virginica  virginica
[40] virginica  virginica  virginica
[43] virginica  versicolor virginica
[46] virginica  virginica  virginica
Levels: setosa versicolor virginica
```

- k=5를 한 kNN의 결과

  md1에는 test 데이터(50개)들을 예측한 결과가 저장되어 있음

POSTECH

kNN의 수행과 결과

- knn의 매뉴얼 : help(knn)

R: k-Nearest Neighbour Classification Find in Topic

k-Nearest Neighbour Classification

Description

k-nearest neighbour classification for test set from training set. For each row of the test set, the `k` nearest (in Euclidean distance) training set vectors are found, and the classification is decided by majority vote, with ties broken at random. If there are ties for the `k`th nearest vector, all candidates are included in the vote.

Usage

```
knn(train, test, cl, k = 1, l = 0, prob = FALSE, use.all = TRUE)
```

Arguments

`train`	matrix or data frame of training set cases.
`test`	matrix or data frame of test set cases. A vector will be interpreted as a row vector for a single case.
`cl`	factor of true classifications of training set
`k`	number of neighbours considered.
`l`	minimum vote for definite decision, otherwise `doubt`. (More precisely, less than `k-l` dissenting votes are allowed, even if `k` is increased by ties.)
`prob`	If this is true, the proportion of the votes for the winning class are returned as attribute `prob`.
`use.all`	controls handling of ties. If true, all distances equal to the `k`th largest are included. If false, a random selection of distances equal to the `k`th is chosen to use exactly `k` neighbours.

POSTECH

kNN(k=5)의 결과 - 정확도

```
# confusion matrix
# accuracy of 5-nearest neighbor classification
confusionMatrix(md1,testLabels)
```

예측값 ← md1

타겟변수의 실제값 → testLabels

```
> confusionMatrix(md1,testLabels)
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         19          0         0
  versicolor      0         17         1
  virginica       0          2        11

Overall Statistics

               Accuracy : 0.94
                 95% CI : (0.8345, 0.9875)
    No Information Rate : 0.38
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9086

 Mcnemar's Test P-Value : NA
```

- virginica를 versicolor로 오분류(1개)
- versicolor를 virginica로 오분류(2개)
- 정확도 : 47/50 → 94%
- 오분류율 : 3/50 → 6%

* md1과 testLabels의 유형이 char(문자), factor(범주) 다를때는 오류가 생김

POSTECH

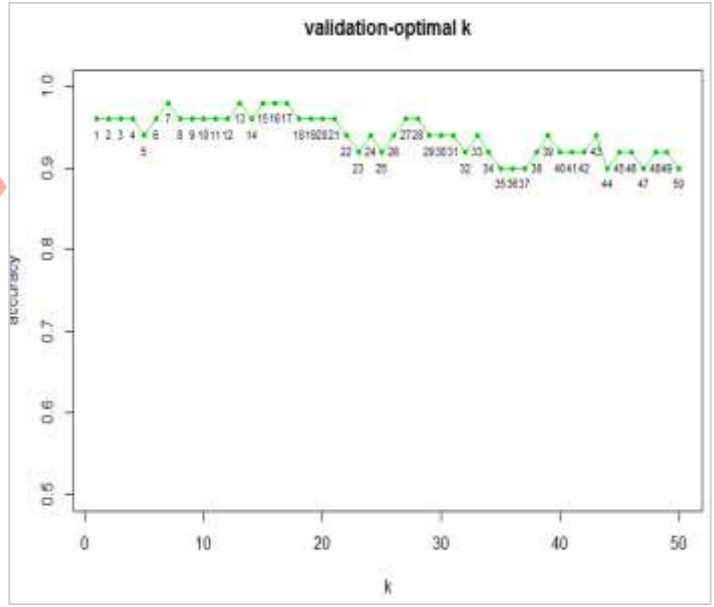
kNN에서 최적 k 탐색

☑ 최적 k의 탐색 : 1 to nrow(train_data)/2(여기서는 1 to 50 까지)

```
# optimal k selection (1 to n/2)
accuracy_k <- NULL
# try k=1 to nrow(train)/2, may use nrow(train)/3(or 4,5) depending the size o
nnum<-nrow(iris.train)/2
for(kk in c(1:nnum))
{
  set.seed(1234)
  knn_k<-knn(train=iris.train,test=iris.test,cl=trainLabels,k=kk)
  accuracy_k<-c(accuracy_k,sum(knn_k==testLabels)/length(testLabels))
}

# plot for k=(1 to n/2) and accuracy
test_k<-data.frame(k=c(1:nnum), accuracy=accuracy_k[c(1:nnum)])
plot(formula=accuracy~k, data=test_k,type="o",ylim=c(0.5,1), pch=20, col=3, ma
with(test_k,text(accuracy~k,labels = k,pos=1,cex=0.7))
```

```
# minimum k for the highest accuracy
min(test_k[test_k$accuracy %in% max(accuracy_k),"k"])
```

k=7에서 정확도(.98)가 가장 높음



POSTECH

kNN에서 최적 k 탐색

최종 kNN모형 (k=7)

```
#k=7 knn
md2<-knn(train=iris.train,test=iris.test,cl=trainLabels,k=7)
confusionMatrix(md2,testLabels)
```

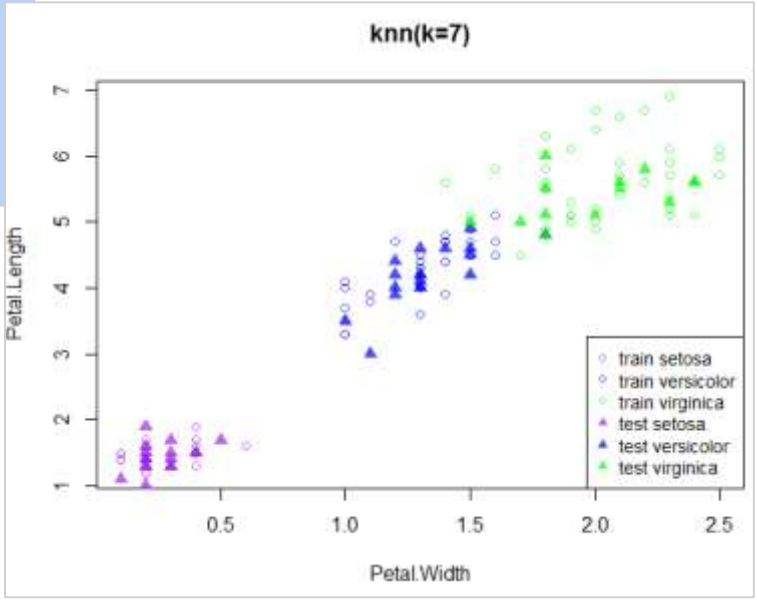
- versicolor를 virginica로 오분류(1개)
- 정확도 : 49/50 → 98%
- 오분류율 : 1/50 → 2%

```
> confusionMatrix(md2,testLabels)
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         19          0         0
  versicolor      0         18         0
  virginica       0          1        12

Overall Statistics

               Accuracy : 0.98
                 95% CI : (0.8935, 0.9995)
    No Information Rate : 0.38
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9695

 Mcnemar's Test P-Value : NA
```

POSTECH

kNN(k=7)의 결과-그래픽

```
# graphic display
plot(formula=Petal.Length ~ Petal.Width,
     data=iris.train,col=alpha(c("purple","blue","green"),0.7)[trainLabels],
     main="knn(k=7)")
points(formula = Petal.Length~Petal.Width,
       data=iris.test,
       pch = 17,
       cex= 1.2,
       col=alpha(c("purple","blue","green"),0.7)[md2]
)
legend("bottomright",
       c(paste("train",levels(trainLabels)),paste("test",levels(testLabels))),
       pch=c(rep(1,3),rep(17,3)),
       col=c(rep(alpha(c("purple","blue","green"),0.7),2)),
       cex=0.9
)
```



- Petal.width와 Petal.length에 산점도를 그려보면 setosa는 잘 분류됨
- virginica와 versicolor는 분류가 잘 되지 않음