

## Wk10-2 : k-인접기법(k-Nearest Neighbor) II

### - 최적 k 탐색과 비중 k-인접기법 -

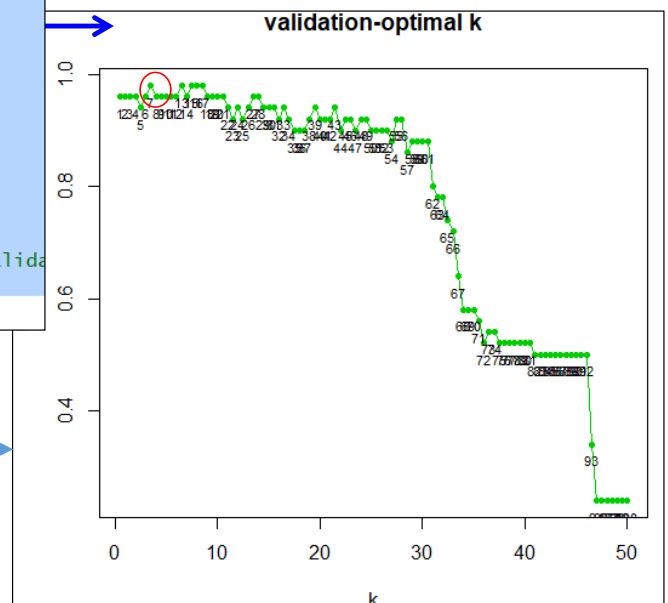
## 6. kNN에서 최적 k 탐색

```
# k selection based in minimum misclassification
accuracy_k <- NULL
for(kk in c(1:nrow(iris.train)))
{
  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))
}
#test_k<-data.frame(k=c(1:nrow(iris.train)),accuracy=accuracy_k)
test_k<-data.frame(k=c(1:nrow(iris.train)/2),accuracy=accuracy_k)

plot(formula=accuracy~k, data=test_k,type="o",pch=20, col=3, main="validation-optimal k")
with(test_k,text(accuracy~k,labels = rownames(test_k),pos=1,cex=0.7))
min(test_k[test_k$accuracy %in% max(accuracy_k),"k"])
```

k=7에서 정확도(.98)가 가장 높음



## 6. kNN에서 최적 k 탐색

### • 최종 kNN모형 (k=7)

```
#k=7 knn
mdl<-knn(train=iris.train,test=iris.test,cl=trainLabels,k=7)
CrossTable(x=testLabels,y=mdl, prop.chisq=FALSE)
```

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

Total Observations in Table: 50

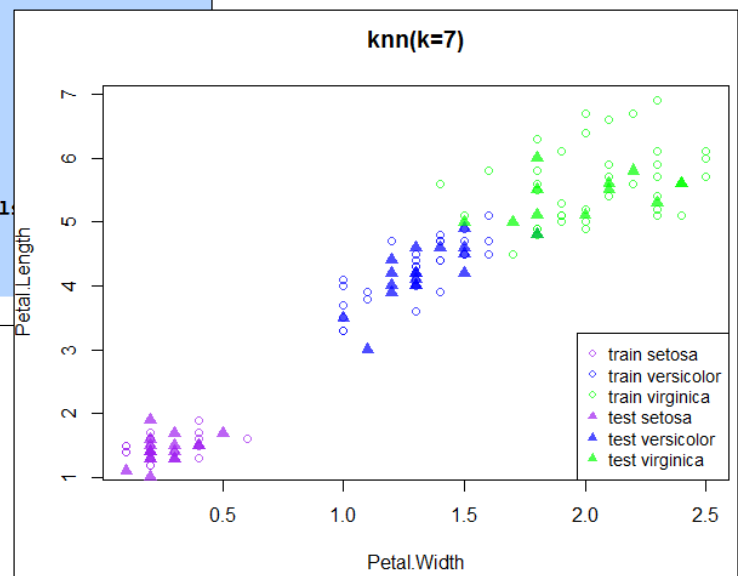
testLabels	mdl	setosa	versicolor	virginica	Row Total
setosa		19	0	0	19
		1.000	0.000	0.000	0.380
		1.000	0.000	0.000	
		0.380	0.000	0.000	
versicolor		0	18	1	19
		0.000	0.947	0.053	0.380
		0.000	1.000	0.077	
		0.000	0.360	0.020	
virginica		0	0	12	12
		0.000	0.000	1.000	0.240
		0.000	0.000	0.923	
		0.000	0.000	0.240	
Column Total		19	18	13	50
		0.380	0.360	0.260	

## 6. kNN(k=7)의 결과-그래픽

```
# with two variables, how classify?
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)[mdl])

legend("bottomright",
      c(paste("train",levels(trainLabels)),paste("test",levels(trainLabels))),
      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.



## 7. Weighted kNN

- 거리에 따라 가중치를 부여하는 두 가지 알고리즘이 존재 (패키지 : kknn)

```
#####  
# weighted KNN  
install.packages("kknn")#weighted value k  
library(kknn)  
help("kknn")
```

kknn {kknn}

R Documentation

### Weighted k-Nearest Neighbor Classifier

**Description**

Performs k-nearest neighbor classification of a test set using a training set. For each row of the test set, the k nearest training set vectors (according to Minkowski distance) are found, and the classification is done via the maximum of summed kernel densities. In addition even ordinal and continuous variables can be predicted.

**Usage**

```
kknn(formula = formula(train), train, test, na.action = na.omit(),  
      k = 7, distance = 2, kernel = "optimal", ykernel = NULL, scale=TRUE,  
      contrasts = c('unordered' = "contr.dummy", ordered = "contr.ordinal"))  
kknn.dist(learn, valid, k = 10, distance = 2)
```

**Arguments**

formula	A formula object.
train	Matrix or data frame of training set cases.
test	Matrix or data frame of test set cases.
learn	Matrix or data frame of training set cases.
distance	Parameter of Minkowski distance.
kernel	Kernel to use. Possible choices are "rectangular" (which is standard unweighted knn), "triangular", "epanechnikov" (or beta(2,2)), "biweight" (or beta(3,3)), "triweight" (or beta(4,4)), "cos", "inv", "gaussian", "rank" and "optimal".

5

## 7. Weighted kNN 결과

- k=5, distance=1

```
#knn(weighted value)  
md2<-kknn(Species~., train=train,test=test,k=5,distance=1,kernel="triangular")  
md2
```

- weighted kNN의 결과를 md2로 저장
- weighted kNN의 결과를 보기 위해서 fitted함수를 이용 (fitted(md2))

```
#knn(weighted value)  
md2<-kknn(Species~., train=train,test=test,k=5,distance=1,kernel="triangular")  
md2  
md2_fit<-fitted(md2)  
md2_fit
```

```
> md2_fit<-fitted(md2)  
> md2_fit  
[1] setosa setosa setosa setosa setosa setosa setosa  
[8] setosa setosa setosa setosa setosa setosa setosa  
[15] setosa setosa setosa setosa setosa versicolor versicolor  
[22] versicolor versicolor versicolor virginica versicolor virginica versicolor  
[29] versicolor versicolor versicolor versicolor versicolor versicolor versicolor  
[36] versicolor versicolor versicolor virginica virginica virginica virginica  
[43] versicolor virginica virginica virginica virginica virginica virginica  
[50] virginica  
Levels: setosa versicolor virginica
```

6

## 7. Weighted kNN 결과

10.2 k-Nearest Neighbor II

```
# accuracy of weighted knn
CrossTable(x=testLabels,y=md2_fit,prop.chisq=FALSE,prop.c=FALSE)
```

- 정확도 : 47/50 -> 94%
- versicolor를 virginica로 오분류(2개)
- virginica를 versicolor로 오분류(1개)

Cell Contents

				N
				N / Row Total
				N / Table Total

Total Observations in Table: 50

testLabels	md2_fit setosa	versicolor	virginica	Row Total
setosa	19 1.000 0.380	0 0.000 0.000	0 0.000 0.000	19 0.380
versicolor	0 0.000 0.000	17 0.895 0.340	2 0.105 0.040	19 0.380
virginica	0 0.000 0.000	1 0.083 0.020	11 0.917 0.220	12 0.240
Column Total	19	18	13	50

## 7. Weighted kNN 결과

10.2 k-Nearest Neighbor II

- k=7, distance=2로 옵션 변경한 결과

```
# weighted knn (k=7, distance=2)
md3<-kkn(Species~., train=train,test=test,k=7,distance=2,kernel="triangular")
md3
# to see results for weighted knn
md3_fit<-fitted(md2)
md3_fit
# accuracy of weighted knn
CrossTable(x=testLabels,y=md3_fit,prop.chisq=FALSE,prop.c=FALSE)
```

- 정확도 : 49/50 -> 98%
- virginica를 versicolor로 오분류(1개)

Total Observations in Table: 50

testLabels	md3_fit setosa	versicolor	virginica	Row Total
setosa	19 1.000 0.380	0 0.000 0.000	0 0.000 0.000	19 0.380
versicolor	0 0.000 0.000	19 1.000 0.380	0 0.000 0.000	19 0.380
virginica	0 0.000 0.000	1 0.083 0.020	11 0.917 0.220	12 0.240
Column Total	19	20	11	50

