k vecinos más próximos

MANADINE - Análisis de Datos 3

4 de diciembre de 2024

Índice

Introducción

kNN

Similaridad y disimilaridad

Ejemplos en R

Bibliografía

Índice

Introducción

kNN

Similaridad y disimilaridad

Ejemplos en R

Bibliografía

Aprendizaje perezoso *vs.* ansioso

Aprendizaje perezoso (*lazy*)

- **Aprendizaje basado en instancias (datos de entrenamiento).**
- **Se almacenan todas las instancias y se usan para clasificar una nueva observación.**
- **Propiedades:**
  - **Poco tiempo de entrenamiento, pero mucho para la predicción.**
  - **Aproximaciones locales a la función objetivo.**
  - **Sencillo de entender y aplicar.**

Aprendizaje perezoso *vs.* ansioso

Aprendizaje ansioso (*eager*)

- **Regresión logística, árboles de decisión, bosques aleatorios, redes neuronales...**
- **Con los datos de entrenamiento se construye un modelo (criterio de clasificación) aplicable a observaciones nuevas.**
- **Ansioso: persigue una única hipótesis que cubra todo el espacio de instancias.**
- **Propiedades:**
  - **Mucho tiempo de entrenamiento, poco para la predicción.**
  - **Aproximaciones más generales a la función objetivo.**
  - **Más complicado de aplicar.**

Índice

Introducción

kNN

Similaridad y disimilaridad

Ejemplos en R

Bibliografía

kNN: los k vecinos más próximos

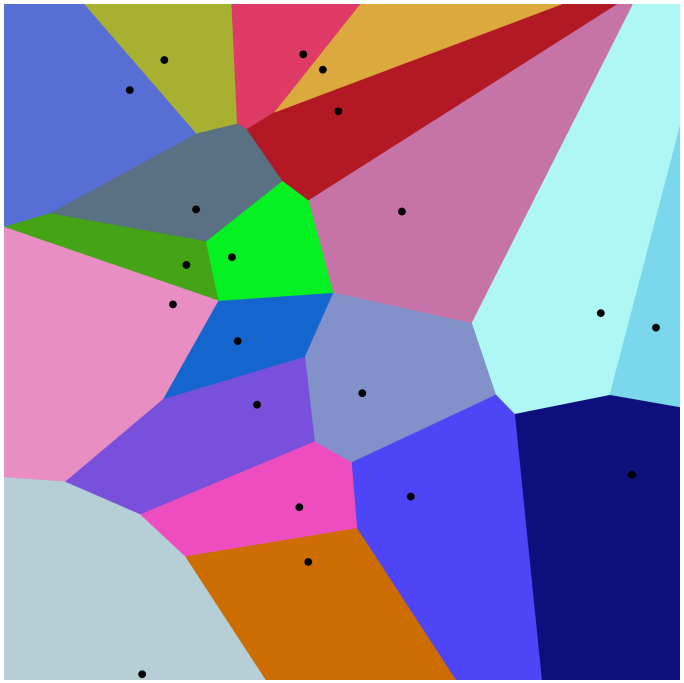
Esquema general

- Se dispone de n observaciones o instancias.
- Cada instancia contiene p variables predictoras $\vec{x}_i = (x_{i1}, ..., x_{ip})$ y el valor y_i de la función objetivo.
- Se calcula la cercanía entre las instancias mediante una distancia $d(\vec{x}_i, \vec{x}_j)$ o una medida de similaridad.
- De las k instancias más cercanas a $\vec{x}_i$, que podemos denotar con índices $i_1, ..., i_k$ (es decir, las instancias $\vec{x}_{i_1}, ..., \vec{x}_{i_k}$), kNN devuelve, según sea la función objetivo,
  - discreta: la moda de $y_{i_1}, ..., y_{i_k}$.
  - continua: el promedio de $y_{i_1}, ..., y_{i_k}$ (media aritmética, media ponderada, mediana...).

kNN: los k vecinos más próximos

Diagrama de Voronói o teselación de Thiessen

Subespacios de decisión inducidos al aplicar kNN con k=1 a un conjunto de instancias.



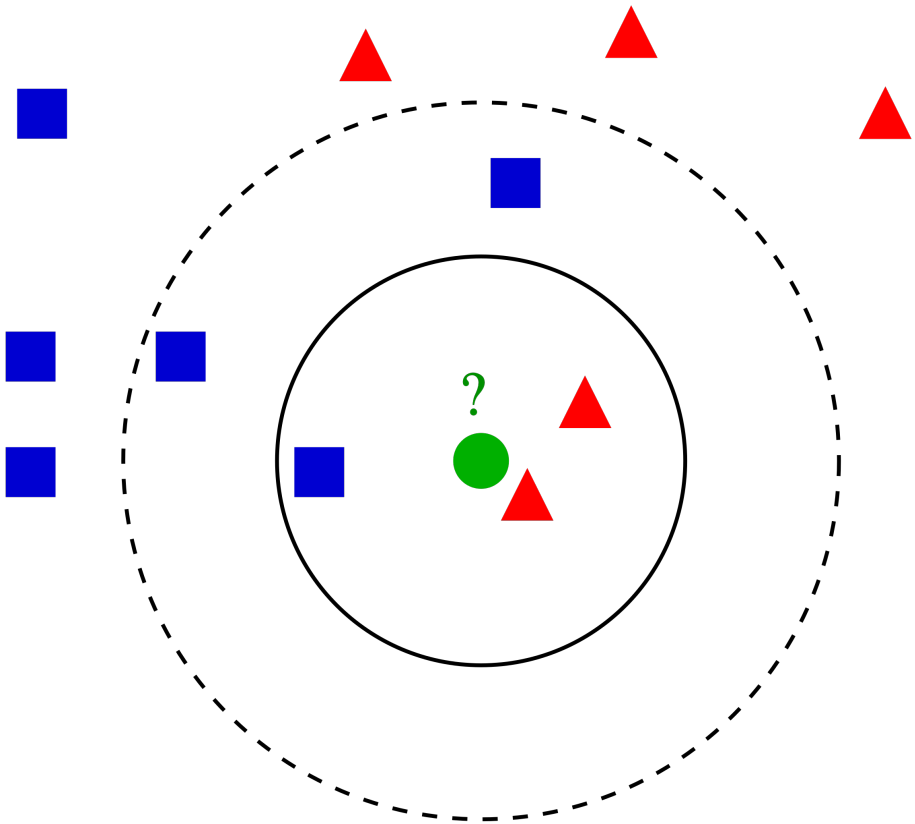
Distancia euclídea

Distancia de Manhattan

Fuente: `https://en.wikipedia.org/wiki/Voronoi_diagram`

kNN: los k vecinos más próximos

Ejemplo



● **clasificado...**

$k = 3 \implies$ **...como** ▲

$k = 5 \implies$ **...como** ■

Fuente:
`https://es.wikipedia.org/wiki/K_vecinos_mÃąs_prÃşximos`

kNN: los k vecinos más próximos

Algoritmo para clasificar

- Considerar instancias de entrenamiento $(\vec{x}_i, y_i)$.
- Elegir el número k de vecinos.
- Dada una nueva instancia, $\vec{x}_q$,
  encontrar los k puntos $\vec{x}_{q_1}$, ..., $\vec{x}_{q_k}$
  más cercanos a $\vec{x}_q$ según la distancia $d(\cdot, \cdot)$.
- Clasificar $\vec{x}_q$ según la moda de y_{q_1}, ..., y_{q_k}.

kNN: los k vecinos más próximos

Algoritmo para regresión

- Considerar instancias de entrenamiento $(\vec{x}_i, y_i)$.
- Elegir el número k de vecinos.
- Dada una nueva instancia, $\vec{x}_q$,
  encontrar los k puntos $\vec{x}_{q_1}$, ..., $\vec{x}_{q_k}$
  más cercanos a $\vec{x}_q$ según la distancia $d(\cdot, \cdot)$.
- La predicción para $\vec{x}_q$ será la media o mediana de y_{q_1}, ..., y_{q_k}.

kNN: los k vecinos más próximos

Regresión con distancia ponderada

- **Ponderar la contribución de cada uno de los k vecinos de acuerdo a su distancia a $\vec{x}_q$.**
- **Peso mayor a los vecinos más próximos,**

$$y_q = \frac{\sum_i w_i y_i}{\sum_i w_i} \qquad w_i = e^{-\frac{d(x_i, x_q)}{2\sigma^2}}$$

kNN: los k vecinos más próximos

Propiedades

- **Robusto a datos con ruido al promediar k valores.**
- **Maldición de la dimensionalidad:**
  la distancia entre vecinos podría estar dominada por atributos irrelevantes.
  Remedios:
  - **Tipificar.**
  - **Eliminar atributos poco relevantes.**

kNN: los k vecinos más próximos

Elección de k

- **k pequeño: Sobreajuste (varianza alta, error pequeño).**
- **k grande: Se tiende a usar puntos irrelevantes (varianza pequeña, mucho error).**
- **Más criterios en** `http://scott.fortmann-roe.com/docs/BiasVariance.html`

Índice

Introducción

kNN

Similaridad y disimilaridad

Ejemplos en R

Bibliografía

Distancias y similaridades

`dist`

La función `dist` de R dispone de varios métodos:

- **euclidean**
- **maximum**
- **manhattan**
- **canberra**
- **binary**
- **minkowski**

`class::knn`

Las funciones `knn` y `knn1` del paquete `class` emplean la distancia euclídea.

Índice

Introducción

kNN

Similaridad y disimilaridad

Ejemplos en R

Bibliografía

Ejemplo 1

iris - enfoque habitual

```
library(class) # para knn
n <- dim(iris)[1]
entrena <- sample(1:n, n*0.75)   # datos de entrenamiento
valida  <- setdiff(1:n, entrena) # datos de validación
vecinos3 <- knn (iris[entrena,-5],
                 iris[valida, -5],
                 iris[entrena, 5],
                 k=3, prob=TRUE)
summary(vecinos3) # frecuencias
table (iris$Species[valida], vecinos3) # matriz de confusión
```

Ejemplo 1 bis

iris - ejemplo de ?knn

```
library(class) # para knn
entrena <- rbind(iris[ 1:25,-5], iris[51: 75,-5], iris[101:125,-5])
valida  <- rbind(iris[26:50,-5], iris[76:100,-5], iris[126:150,-5])
cl <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
vecinos3 <- knn (entrena, valida, cl, k = 3, prob=TRUE)
table (cl, vecinos3)
##    vecinos3
## cl   c  s  v
##   c 23  0  2
##   s  0 25  0
##   v  3  0 22

## proporciones de votos de la clase ganadora:
prob <- attr(vecinos3,"prob")
```

Ejemplo 2

simulación de dos poblaciones normales bidimensionales

```
library(mvtnorm)   # librería para generar datos normal multivariante
library(RcmdrMisc) # librería para numSummary
library(class)     # librería para knn
## Muestra población 1
mu1 <- c(11,12); S1 <- matrix(c(1.5, 0.2, 0.2, 1.5), nrow=2, ncol=2)
n1 <- 100
X1 <- rmvnorm(n1, mu1, S1)
X1 <- cbind(X1, rep(1, n1))
head(X1)
## Muestra población 2
mu2 <- c(12.5,10); S2 <- matrix(c(1.5,0.6,0.6,1.6), nrow=2, ncol=2)
n2 <- 100
X2 <- rmvnorm(n2, mu2, S2)
X2 <- cbind(X2, rep(2, n2))
cor(X2[, 1:2])
X <- rbind(X1, X2) # muestra global
colnames(X) <- c("V1", "V2", "Y")
head(X)
```

Ejemplo 2

simulación de dos poblaciones normales bidimensionales

```
n <- dim(X)[1]
ext_v1 <- c(min(X[,1]), max(X[,1]))  # mínimo y máximo de V1
ext_v2 <- c(min(X[,2]), max(X[,2]))  # mínimo y máximo de V2
## gráfica conjunta de las dos muestras
plot(X[1:n1,1], X[1:n1,2], col="red", xlim=ext_v1, ylim=ext_v2,
     xlab="V1", ylab="V2")
points(X[(n1+1): (n1+n2),1], X[(n1+1):(n1+n2),2], col="blue")
## descripción de los dos grupos
numSummary(X[, 1:2], groups=X[,3])
boxplot(X[,1] ~ X[,3], xlab="Poblaciones", ylab="V1")
boxplot(X[,2] ~ X[,3], xlab="Poblaciones", ylab="V2")
## clasificación usando knn
n <- dim(X)[1]; n
entrena <- sample(1:n, n*0.75); valida <- setdiff(1:n, entrena)
vecinos3 <- knn(X[entrena,-3], X[valida,-3], factor(X[entrena,3]),
                k=3, prob=TRUE)
summary(vecinos3)
table (X[valida,3], vecinos3)
```

Ejemplo 3

Simulación: efecto de la dimensión

```
n <- 1000 ; d <- 1 # dimensiones
g <- rbinom (n, 1, 0.5) * 2 # * diferencia de medias ;
x1 <- rnorm (n, g) # primera variable importa ;
X <- cbind (x1, matrix(rnorm(d*n),n)) # resto, no ;
en <- sample (n, 0.8*n) # datos de entrenamiento ;
aciertos <- function (k)
  mean (diag (table (g[-en],
                     knn (X[en,], X[-en,], g[en], k))))
aciertos(1) ; aciertos(11) ; aciertos(111)
## dif=1 dim=1    -> 68% con k=11
## dif=1 dim=100  -> 60% con k=111
## dif=1 dim=1000 -> 55% con k=111
## dif=2 dim=1    -> 80% con k=11
## dif=2 dim=100  -> 75% con k=111
## dif=2 dim=1000 -> 65% con k=111
```

Ejemplo 4

`MASS::Cushings`

- **El síndrome de Cushing es un desorden hipertensivo asociado con sobresecreción de cortisol por la glándula suprarrenal.**
- **Los datos recogen tasas (mg/24h) de secreción urinaria de dos hormonas esteroideas: tetrahidrocortisona y pregnanetriol.**
- **Type: tipo de síndrome subyacente:**
  - **a** adenoma.
  - **b** hiperplasia bilateral.
  - **c** carcinoma.
  - **u** desconocido.

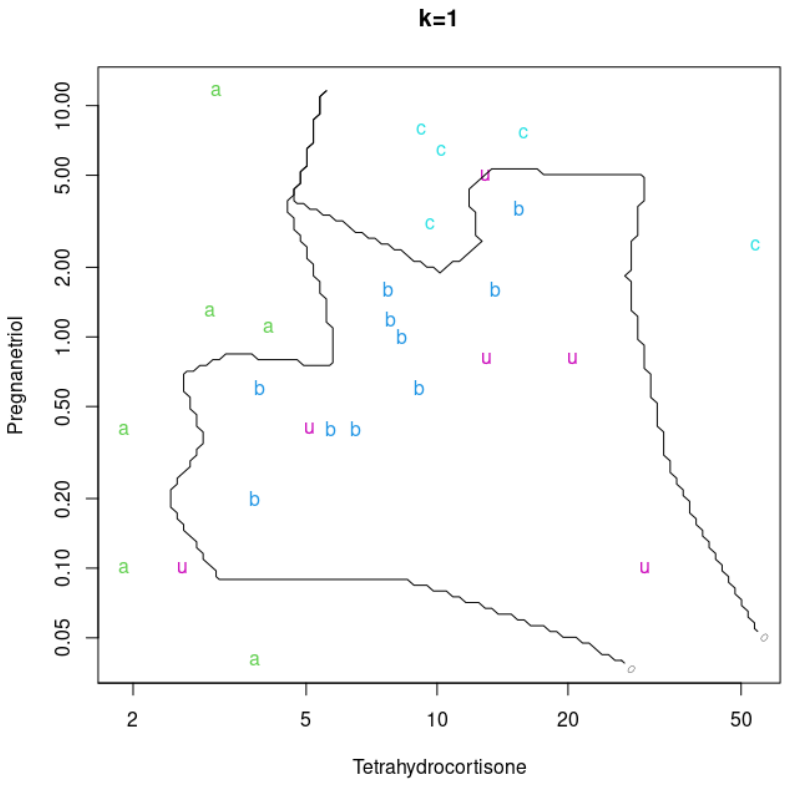
Ejemplo 4

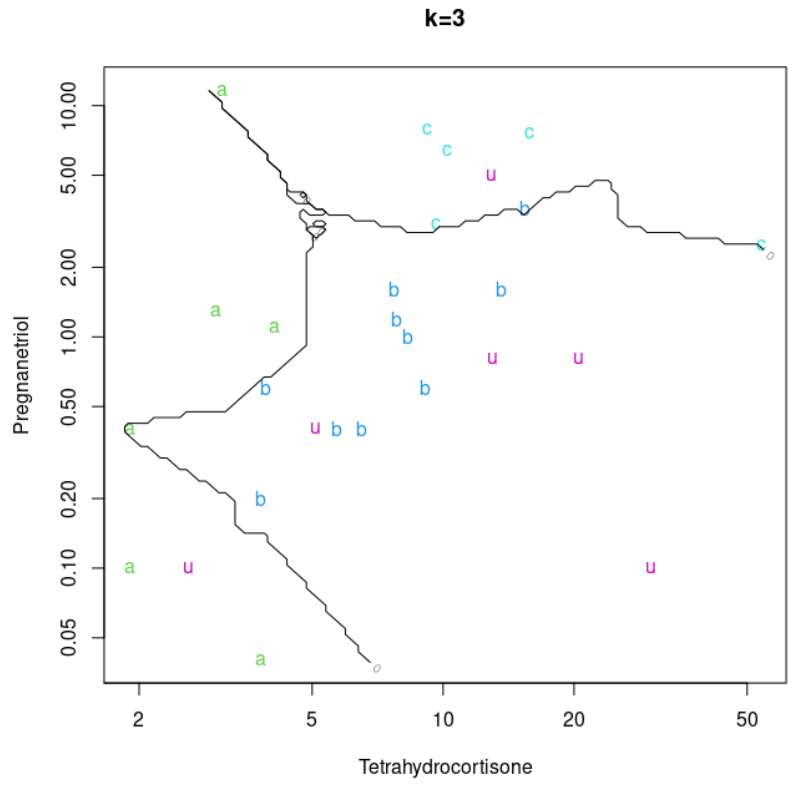
```
## para gráficas del ejemplo; adaptado de
## https://rdrr.io/cran/MASS/src/inst/scripts/ch12.R
library(MASS) # para Cushings
library(nnet) # para class.ind
cushplot <- function(xp, yp, Z, m)
{
  plot(Cushings[,1], Cushings[,2], log="xy", type="n", main=m,
       xlab = "Tetrahydrocortisone", ylab = "Pregnanetriol")
  for(il in 1:4) {
      set <- Cushings$Type==levels(Cushings$Type)[il]
      text(Cushings[set, 1], Cushings[set, 2],
           labels = as.character(Cushings$Type[set]), col = 2 + il) }
  zp <- Z[, 3] - pmax(Z[, 2], Z[, 1])
  contour(exp(xp), exp(yp), matrix(zp, np),
          add = TRUE, levels = 0, labex = 0)
  zp <- Z[, 1] - pmax(Z[, 2], Z[, 3])
  contour(exp(xp), exp(yp), matrix(zp, np),
          add = TRUE, levels = 0, labex = 0)
  invisible()
}
```

Ejemplo 4

```
## hasta el 21 tienen Type conocido
cush <- log(as.matrix(Cushings[, -3]))
tp <- Cushings$Type[1:21]
xp <- seq(0.6, 4.0, length = 100); np <- length(xp)
yp <- seq(-3.25, 2.45, length = 100)
cushT <- expand.grid(Tetrahydrocortisone = xp,
                     Pregnanetriol = yp)
Z <- knn(scale(cush[1:21, ], FALSE, c(3.4, 5.7)),
         scale(cushT, FALSE, c(3.4, 5.7)), tp)
cushplot(xp, yp, class.ind(Z), "k=1")
Z <- knn(scale(cush[1:21, ], FALSE, c(3.4, 5.7)),
         scale(cushT, FALSE, c(3.4, 5.7)), tp, k = 3)
cushplot(xp, yp, class.ind(Z), "k=3")
```

Ejemplo 4


k=1
Pregnanetriol
0.05
0.10
0.20
0.50
1.00
2.00
5.00
10.00
2
5
10
20
50
Tetrahydrocortisone


k=3
Pregnanetriol
0.05
0.10
0.20
0.50
1.00
2.00
5.00
10.00
2
5
10
20
50
Tetrahydrocortisone

Índice

Introducción

kNN

Similaridad y disimilaridad

Ejemplos en R

Bibliografía

Bibliografía

Yizhou Sun (2017)

CS145: INTRODUCTION TO DATA MINING
7: Vector Data: K Nearest Neighbor
https://www.coursehero.com/file/112016329/KNNpdf/

Venables & Ripley (2002)

MODERN APPLIED STATISTICS WITH S
12.3 Non-Parametric Rules
https://link.springer.com/book/
10.1007/978-0-387-21706-2