

Mineria de Datos

Clasificación supervisada: clasificadores k-nn

Dr. Edgar Acuna
Departamento de Matemáticas

Universidad de Puerto Rico- Mayaguez

academic.uprm.edu/eacuna

En el método de los k vecinos mas cercanos ("knn: k nearest neighbors") (Fix y Hodges, 1951) se estima la función de densidad de donde provienen un conjunto de datos.

Aplicado a clasificación supervisada el metodo k -nn se usa para estimar la función de densidad condicional $f(\mathbf{x}/C_j)$, de las predictoras \mathbf{x} por cada clase C_j .

Es un método de clasificación noparamétrico, ya que no se hace ninguna suposición distribucional acerca de las variables predictoras.

Estimacion de densidad univariada por knn

- Sea x_1, x_2, \dots, x_n una muestra con una función densidad desconocida $f(x)$ que se desea estimar y sea t un número real. Recordar que la probabilidad de que x caiga en el intervalo $(t-h, t+h)$, puede ser aproximada por $2hf(t)$, donde f es la función de densidad y h es una constante cercana a cero.
- Por otro lado dicha probabilidad tambien puede ser estimada por k/n , donde k es el número de observaciones que caen en el intervalo $(t-h, t+h)$. O sea $f(t) \cong k/2hn$.
- En estimacion k-nn, k es prefijado y h se determina de acuerdo a el. Más formalmente, sea $d(x,y)=|x-y|$ la distancia usual entre los puntos x e y de la linea recta.

Supongamos que hemos calculados todas las distancias $d(x_i, t) = |x_i - t|$ y que estas son ordenadas de menor a mayor, tal como

$$d_1(t) \leq d_2(t) \leq \dots \leq d_n(t)$$

- Entonces el estimador de la función de densidad f en el punto t según sus k vecinos más cercanos está dado por

$$\hat{f}(t) = \frac{k}{2nd_k(t)}$$

Ejemplo

Estimar usando k-nn la función de densidad correspondiente al siguiente conjunto de datos.

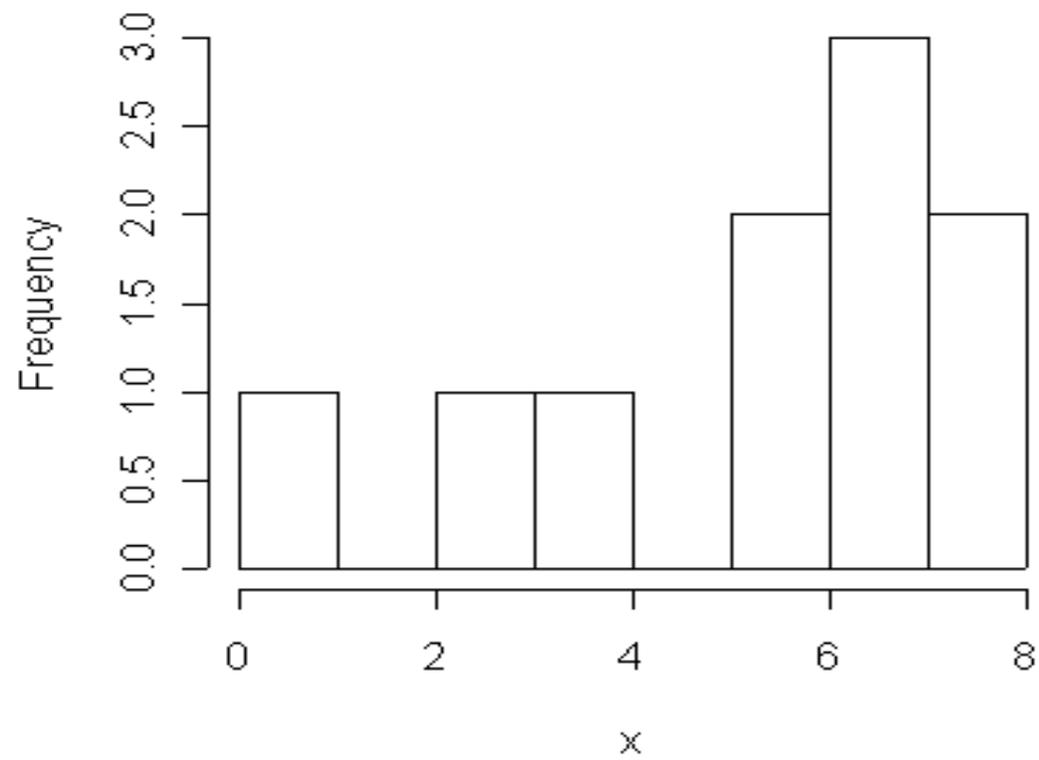
```
7.03860 6.38018 6.95461 2.25521 7.24608 6.62930 3.92734  
0.40701 5.59448 5.05751
```

El histograma (normalizado para que el area sea 1) de los datos se muestra en el siguiente slide.

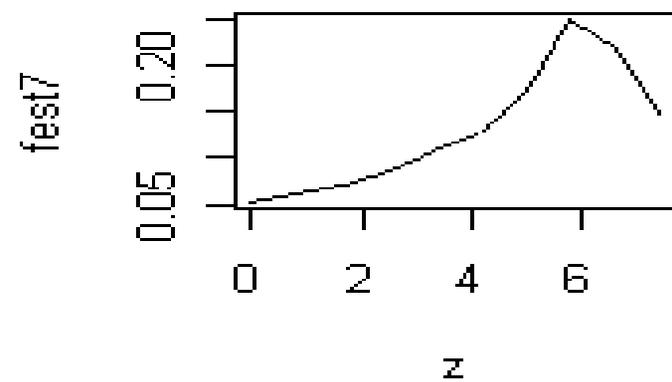
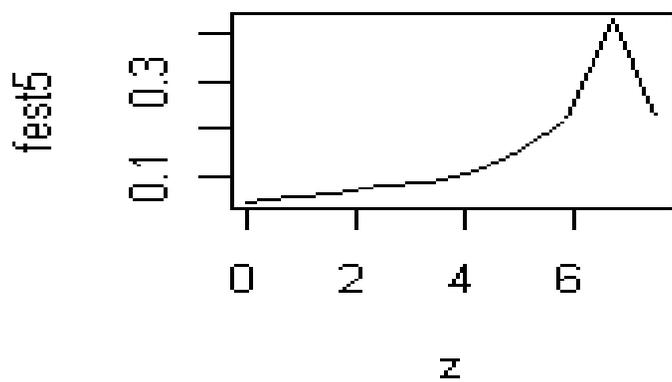
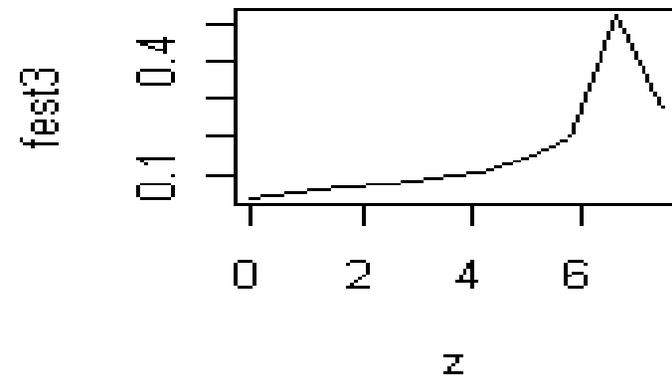
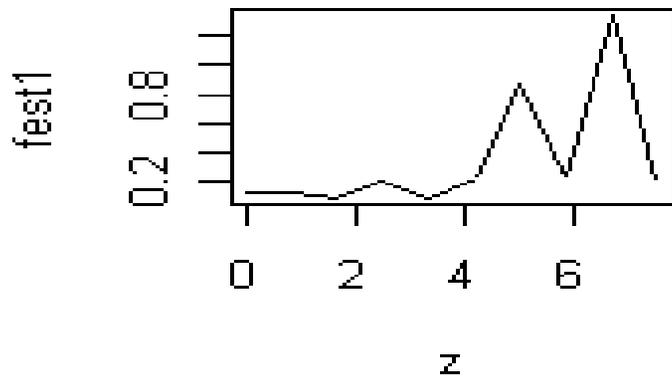
El estimador de la funcion de densidad usando un vecino mas cercano se muestra abajo. Se hace uso de la funcion fdknn en R.

```
>y=c(7.03860,6.38018,6.95461,2.25521,7.24608,6.62930,3.92734,0.40701,5.5  
9448,5.05751)  
> fdknn(y,10,1)  
[1] 0.05000000 2.79624682 0.05656350 0.50938975 0.08448023 0.12811826  
[7] 0.20642959 0.50845160 2.79624682 0.05000000
```

Histogram of x



Estimador k-nn de la funcion de densidad para k=1,3,5,7



Estimacion de densidad multivariada

- El estimado de la función de densidad en el punto d -dimensional \mathbf{x} tiene la forma

$$\hat{f}(\mathbf{x}) = \frac{k}{nv_k(\mathbf{x})}$$

- donde $v_k(\mathbf{x})$ es el volumen de un elipsoide centrado en \mathbf{x} de radio $r_k(\mathbf{x})$, que a su vez es la distancia de \mathbf{x} al k -ésimo punto más cercano.

El clasificador k-nn

Desde el punto de vista de clasificación supervisada el método k-nn es bien simple de aplicar.

En efecto, si las funciones de densidades condicionales $f(\mathbf{x}/C_i)$ de la clase C_i que aparecen en la ecuación

$$P(C_i / \mathbf{x}) = \frac{f(\mathbf{x} / C_i)\pi_i}{f(\mathbf{x})}$$

son estimadas por k-nn. Entonces, para clasificar un objeto, con mediciones dadas por el vector \mathbf{x} , en la clase C_i se debe cumplir que

$$\frac{k_i\pi_i}{n_i\nu_k(\mathbf{x})} > \frac{k_j\pi_j}{n_j\nu_k(\mathbf{x})}$$

para $j \neq i$. Donde k_i y k_j son los k vecinos de \mathbf{x} en las clase C_i y C_j respectivamente.

Asumiendo priors proporcionales a los tamaños de las clases (n_i/n y n_j/n respectivamente) lo anterior es equivalente a:

$$k_i > k_j \text{ para } j \neq i$$

Luego, el procedimiento de clasificación sería así:

- 1) Hallar los k objetos que están a una distancia más cercana al objeto \mathbf{x} , k usualmente es un número impar.
- 2) Si la mayoría de esos k objetos pertenecen a la clase C_i entonces el objeto \mathbf{x} es asignado a ella.
En caso de empate se clasifica al azar.

-
- Hay dos problemas en el método k-nn, la elección de la distancia o métrica y la elección de k.
 - La métrica más elemental que se puede elegir es la euclídeana $d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})'(\mathbf{x} - \mathbf{y})$. Esta métrica sin embargo, puede causar problemas si las variables predictoras han sido medidas en unidades muy distintas entre sí. Algunos prefieren rescalar los datos antes de aplicar el método. Otra distancia bien usada es la distancia Manhattan definida por $d(\mathbf{x}, \mathbf{y}) = |\mathbf{x} - \mathbf{y}|$.
 - Si hay predictoras de distinto tipo, esto es continuas, nominales y ordinales, entonces el uso de la distancia euclídeana nos es un apropiado. Se pueden usar distancia para variables mixtas (Wilson y Martinez, Gower). Algunas veces, las predictoras discretas son continuizadas, pero esto es muy complejo de hacer.

La distancia Gower (I)

- La distancia Gower (Gower, 1971) entre las instancias d -dimensionales x_i y x_j esta definida por

$$\text{dist.Gow}(x_i, x_j) = \frac{\sum_{k=1}^d w_{ijk} d_{ijk}}{\sum_{k=1}^d w_{ijk}}$$

- Si la k -ésima variable es nominal entonces $d_{ijk}=0$ si $x_{ik}=x_{jk}$ y 1 en caso contrario . Los $w_{ijk}=1$
- Si la k -ésima variable es logica o binaria asimetrica entonces $d_{ijk}=0$ si $x_{ik}=x_{jk} = \text{TRUE}$, y 1 en caso contrario . Los w_{ijk} seran 1 en todos los casos excepto cuando $x_{ik}=x_{jk} = \text{FALSE}$ en cuyo caso valen 0.
- Si la k -ésima variable es continua u ordinal entonces $d_{ijk}=|x_{ik}-x_{jk}|/\text{rango}(x_k)$. Los $w_{ijk}=1$. Tambien si x_{ik} o x_{jk} o ambos estan “missing” entonces w_{ijk} vale 0.

La Distancia Gower

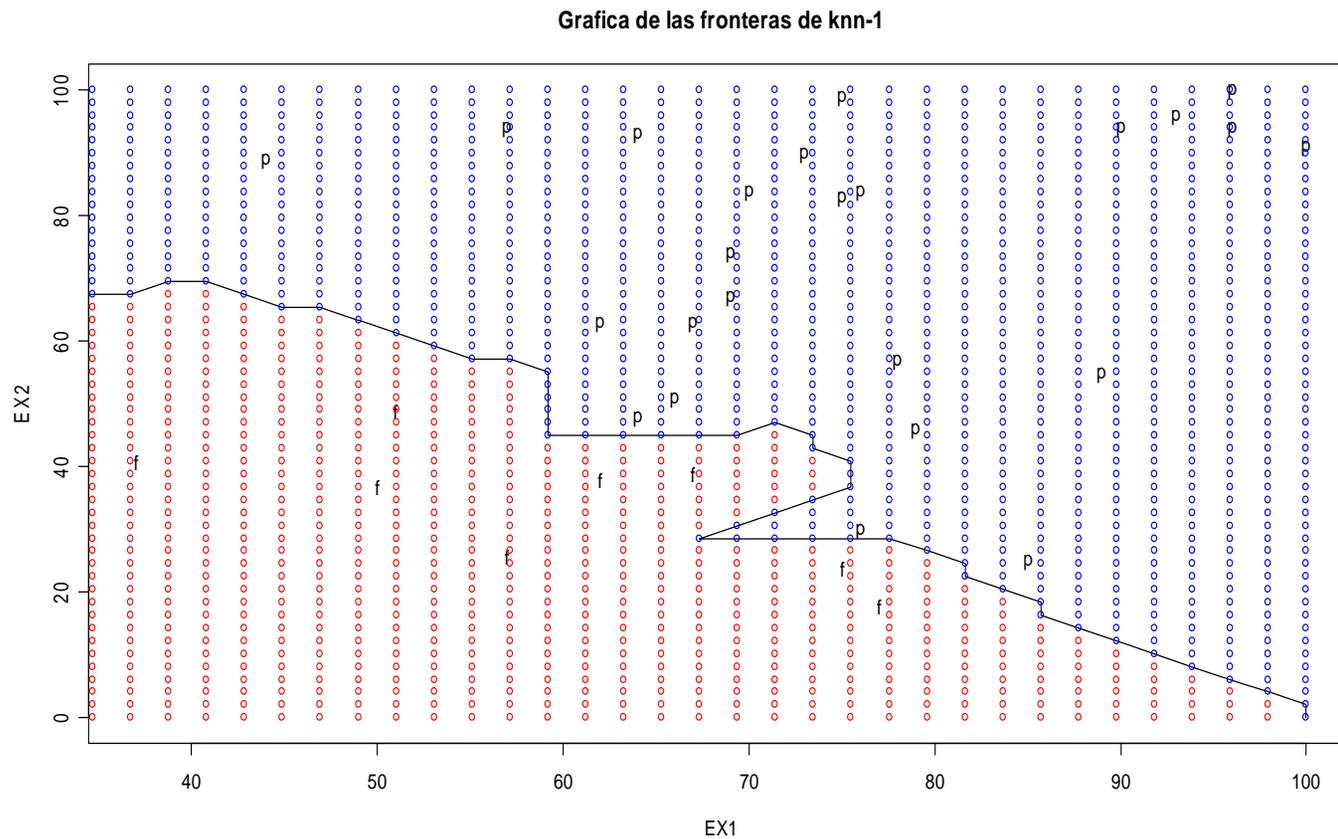
La función `gower.dist` del paquete `StatMatch` de R calcula la distancia gower

`Gower.dist(ejearbol)`

Otra manera de lidiar con variables nominales es usar variables binarias para representarlas. Así si una variable nominal asume k valores distintos se pueden usar $k-1$ variables binarias para representarlas.

-
- Enas y Choi (1996) usando simulación hicieron un estudio para determinar el k óptimo cuando solo hay dos clases presentes y determinaron que si los tamaños muestrales de las dos clases son comparables entonces $k=n^{3/8}$ si habia poca diferencia entre las matrices de covarianzas de los grupos y $k=n^{2/8}$ si habia bastante diferencia entre las matrices de covarianzas.
 - El sesgo del error de clasificación aumenta a medida que k aumenta, en tanto que la varianza disminuye.
 - Se ha demostrado (Cover y Hart, 1967) que la tasa de error del clasificador k -nn es a lo más dos veces la tasa de error óptimo (error del clasificador Bayesiano donde las posteriores son conocidas).

Ejemplo: Frontera de decision



Ejemplo de knn: Bupa

```
> knn(bupa[,1:6],bupa[,1:6],bupa[,7],k=3)
 [1] 2 2 2 2 2 2 1 1 1 1 2 2 2 1 1 2 1 1 2 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 2 1
 [38] 2 2 2 2 2 1 2 1 2 2 2 1 2 1 1 2 2 2 2 2 1 2 1 2 2 2 2 1 1 2 2 2 2 2 1 1 1
 [75] 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 [112] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 1 2 1 1 2 1 2 2
 [149] 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 1 2 1 1 2 1 2 2 1 2 2 1 1 2 2
 [186] 2 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2 2 2 2 2
 [223] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 2 2 1 2 2 1 2 1 1
 [260] 2 1 2 1 2 2 2 2 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 2 2 2 2 2
 [297] 2 1 1 2 1 2 2 2 2 2 2 1 2 2 2 1 2 1 1 1 1 1 2 2 2 2 2 2 1 1 1 1 2 2 2 1 2
 [334] 2 1 2 2 2 1 2 2 1 1 1 2
Levels: 1 2
Error de mala clasificacion por resustitucion
> mean(knn(bupa[,1:6],bupa[,1:6],bupa[,7],k=3)!=bupa[,7])
 [1] 0.1971014
```

Ejemplo (cont)

```
> bupak1=knn(bupa[,-7],bupa[,-7],as.factor(bupa[,7]),k=1)
```

```
> table(bupak1,bupa[,7])
```

```
bupak1  1  2  
  1  145  0  
  2    0 200
```

```
error=0%
```

```
> bupak5=knn(bupa[,-7],bupa[,-7],as.factor(bupa[,7]),k=5)
```

```
> table(bupak5,bupa[,7])
```

```
bupak5  1  2  
  1  94 23  
  2  51 177
```

```
error=21.44%
```

Ejemplo (cont.)

```
># estimacion del error por el metodo de dejar uno afuera
> mean(knn.cv(bupa[,1:6],bupa[,7],k=5)!=bupa[,7])
[1] 0.3391304
> mean(knn.cv(bupa[,1:6],bupa[,7],k=7)!=bupa[,7])
[1] 0.3130435
> mean(knn.cv(bupa[,1:6],bupa[,7],k=9)!=bupa[,7])
[1] 0.3043478
> mean(knn.cv(bupa[,1:6],bupa[,7],k=11)!=bupa[,7])
[1] 0.3188406
># estimacion del error por validacion cruzada
> crossval(bupa,method="knn",kvec=7,rep=10)
[1] 0.3133333
> crossval(bupa,method="knn",kvec=9,rep=10)
[1] 0.3084058
> crossval(bupa,method="knn",kvec=11,rep=10)
[1] 0.3226087
```

Ejemplo de knn: diabetes

```
> diabk3=knn(diabetes[,-9],diabetes[,-9],as.factor(diabetes[,9]),k=3)
> table(diabk3,diabetes[,9])
diabk3  1  2
      1 459 67
      2  41 201
error=14.06%
> diabk5=knn(diabetes[,-9],diabetes[,-9],as.factor(diabetes[,9]),k=5)
> error=mean(diabk5!=diabetes[,9])
>error
19.66
➤#estimacion del error por validacion cruzada
➤crossval(diabetes,method="knn",kvec=3,repert=10)
Loading required package: class
[1] 0.3040365
> crossval(diabetes,method="knn",kvec=5,repert=10)
[1] 0.2901042
```

Otro clasificador noparametrico

- Estimacion de densidad por kernel. Pero su mayor desventaja es que es pesado computacionalmente. El mas usado es el kernel gaussiano.
- Los metodos noparametricos por lo general afrontan el problema de la maldicion de la dimensionalidad. El numero de datos debe crecer casi exponencialmente al numero de la dimension para que la estimacion sea buena.