Clasificadores k-Nearest Neighbors

A. Moujahid

Grupo de Inteligencia Computacional
Universidad del País Vasco UPV/EHU
Curso 2014-2015

Índice

Índice

Idea básica

Un objeto se clasifica por mayoría de votos de sus vecinos, el objeto se le asigna la clase más común entre sus k vecinos más cercanos (k es un entero positivo, por lo general pequeño).

- Los ejemplos de entrenamiento son vectores definidos en un espacio multidimensinal, cada uno identificado por una etiqueta (clase).
- La fase de entrenamiento consiste simplemente en almacenar los vectores característicos y sus clases correspondientes.
- En la fase de clasificación, un vector no etiquetado es clasificado asignandole la clase más frecuente entre sus k vecinos más próximos.

Medidas de proximidad

Dado un conjunto de datos, X, con N objetos definidos en un espacio d-dimensional.
Una **función de distancia (o disimilitud)** satisface las siguientes condiciones:

1. Simétrica: $D(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_j, \mathbf{x}_i)$
2. Positiva: $D(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \quad \forall \mathbf{x}_i, \mathbf{x}_j$
   Si además satisface las dos siguientes condiciones,
3. $D(\mathbf{x}_i, \mathbf{x}_j) \leq D(\mathbf{x}_i, \mathbf{x}_k) + D(\mathbf{x}_k, \mathbf{x}_j), \quad \forall \mathbf{x}_i, \mathbf{x}_j \quad y \quad \mathbf{x}_k$
4. $D(\mathbf{x}_i, \mathbf{x}_j) = 0$ sii $\mathbf{x}_i = \mathbf{x}_j$

se dice que D es una **métrica**.

Medidas de proximidad

Una **función de similitud** satisface las siguientes condiciones:

1. Simétrica: $S(\mathbf{x}_i, \mathbf{x}_j) = S(\mathbf{x}_j, \mathbf{x}_i)$
2. Positiva: $0 \leq S(\mathbf{x}_i, \mathbf{x}_j) \leq 1 \quad \forall \mathbf{x}_i, \mathbf{x}_j$
   Si además satisface las dos siguientes condiciones,
3. $S(\mathbf{x}_i, \mathbf{x}_j)S(\mathbf{x}_j, \mathbf{x}_k) \leq [S(\mathbf{x}_i, \mathbf{x}_j) + S(\mathbf{x}_j, \mathbf{x}_k)]S(\mathbf{x}_i, \mathbf{x}_k)$,
   $\forall \mathbf{x}_i, \mathbf{x}_j \quad y \quad \mathbf{x}_k$
4. $S(\mathbf{x}_i, \mathbf{x}_j) = 1$ sii $\mathbf{x}_i = \mathbf{x}_j$

se dice que S es una **métrica**.

Medidas de disimilitud

1. Distancia Euclidea (L_2): $D(\mathbf{x}_i, \mathbf{x}_j) = \left[\sum_{l=1}^{d} \left|\mathbf{x}_{il} - \mathbf{x}_{jl}\right|^2\right]^{\frac{1}{2}}$
2. Distancia de Minkowski (L_p): $D(\mathbf{x}_i, \mathbf{x}_j) = \left[\sum_{l=1}^{d} \left|\mathbf{x}_{il} - \mathbf{x}_{jl}\right|^p\right]^{\frac{1}{p}}$
3. Distancia de Manhattan (L_1): $D(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{d} \left|\mathbf{x}_{il} - \mathbf{x}_{jl}\right|$
4. Distancia de Chebyshev (L_∞): $D(\mathbf{x}_i, \mathbf{x}_j) = max_{1 \leqslant l \leqslant d} \left|\mathbf{x}_{il} - \mathbf{x}_{jl}\right|$
5. Distancia de Mahalanobis: $D(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T S^{-1} (\mathbf{x}_i - \mathbf{x}_j)}$

- $L_\infty < L_2 < L_1$
- S *es la matriz de covarianza entre variables.*

Medidas de similitud

1. Similitud Coseno: $S(\mathbf{x}_i, \mathbf{x}_j) = cos\alpha = \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}$
2. Coeficiente de correlación de Pearson: $r(i,j) = \frac{C(i,j)}{\sqrt{C(i,i)C(j,j)}}$

- *Disimilitud basada en la similitud coseno:*
  $D(\mathbf{x}_i, \mathbf{x}_j) = 1 - S(\mathbf{x}_i, \mathbf{x}_j)$
- *Disimilitud basada en el coeficiente de Pearson:*
  $D(\mathbf{x}_i, \mathbf{x}_j) = (1 - r(i,j))/2.$

Índice

K-*Nearest Neighbour*

Características

- Un nuevo caso se va a clasificar en la clase más frecuente a la que pertenecen sus K vecinos más cercanos
- Idea muy simple e intuitiva
- Fácil implementación
- No hay modelo explícito (Procesamiento retartado o *Lazy*)
- Coste computacional muy alto
- Requiere el almacenamiento de todos los casos clasificados previamente

Notación

	X_1	...	X_i	...	X_n	C	C_M
1	x_1^1	...	x_i^1	...	x_n^1	c^1	c_M^1
...	...	...	...	...	...	...	...
j	x_1^j	...	x_i^j	...	x_n^j	c^j	c_M^j
...	...	...	...	...	...	...	...
N	x_1^N	...	x_i^N	...	x_n^N	c^N	c_M^N

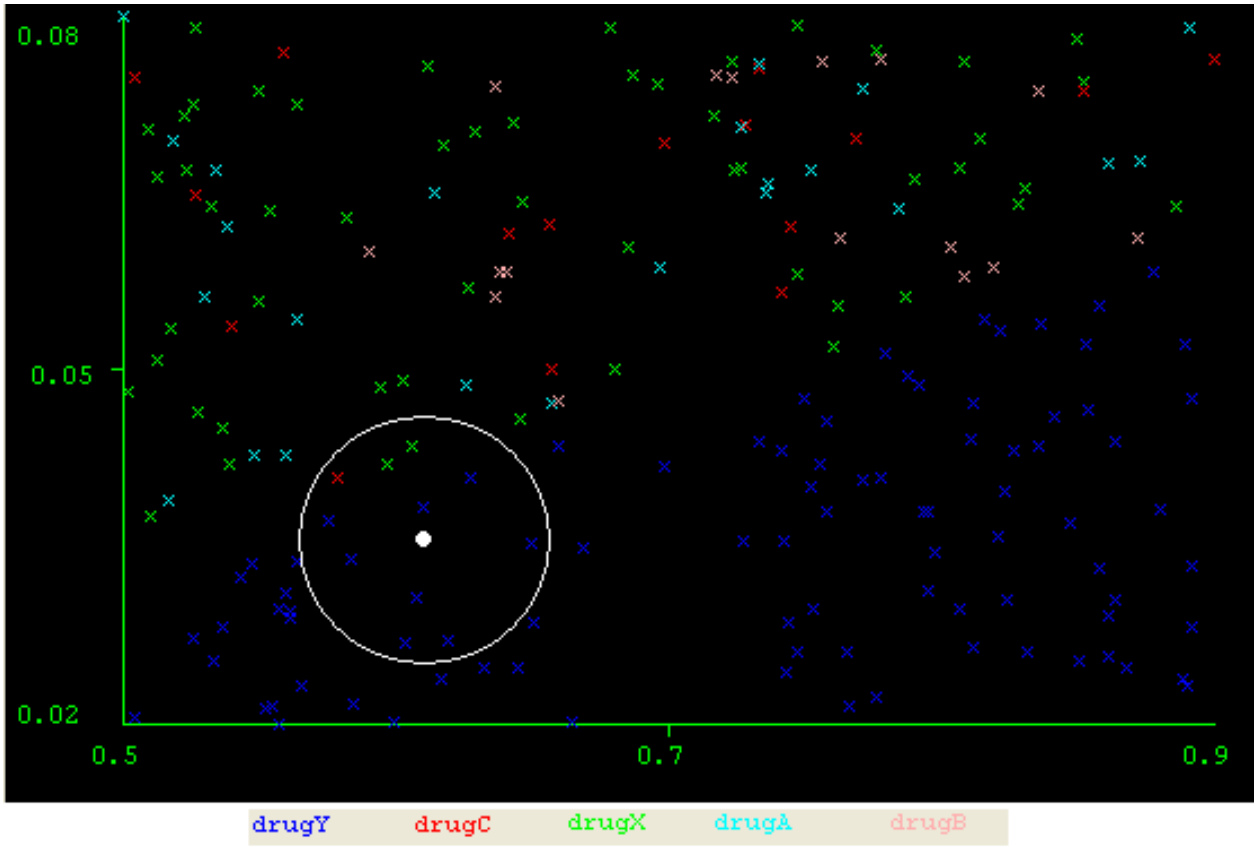
Problema de clasificación supervisada.

El algoritmo K-NN básico

Pseudocódigo para el clasificador K-NN

COMIENZO
 Entrada: $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_N, c_N)\}$
 $\mathbf{x} = (x_1, \ldots, x_n)$ nuevo caso a clasificar
 PARA todo objeto ya clasificado (x_i, c_i)
 calcular $d_i = d(\mathbf{x}_i, \mathbf{x})$
 Ordenar $d_i (i = 1, \ldots, N)$ en orden ascendente
 Quedarnos con los K casos $D_{\mathbf{x}}^K$ ya clasificados más cercanos a $\mathbf{x}$
 Asignar a $\mathbf{x}$ la clase más frecuente en $D_{\mathbf{x}}^K$
FIN

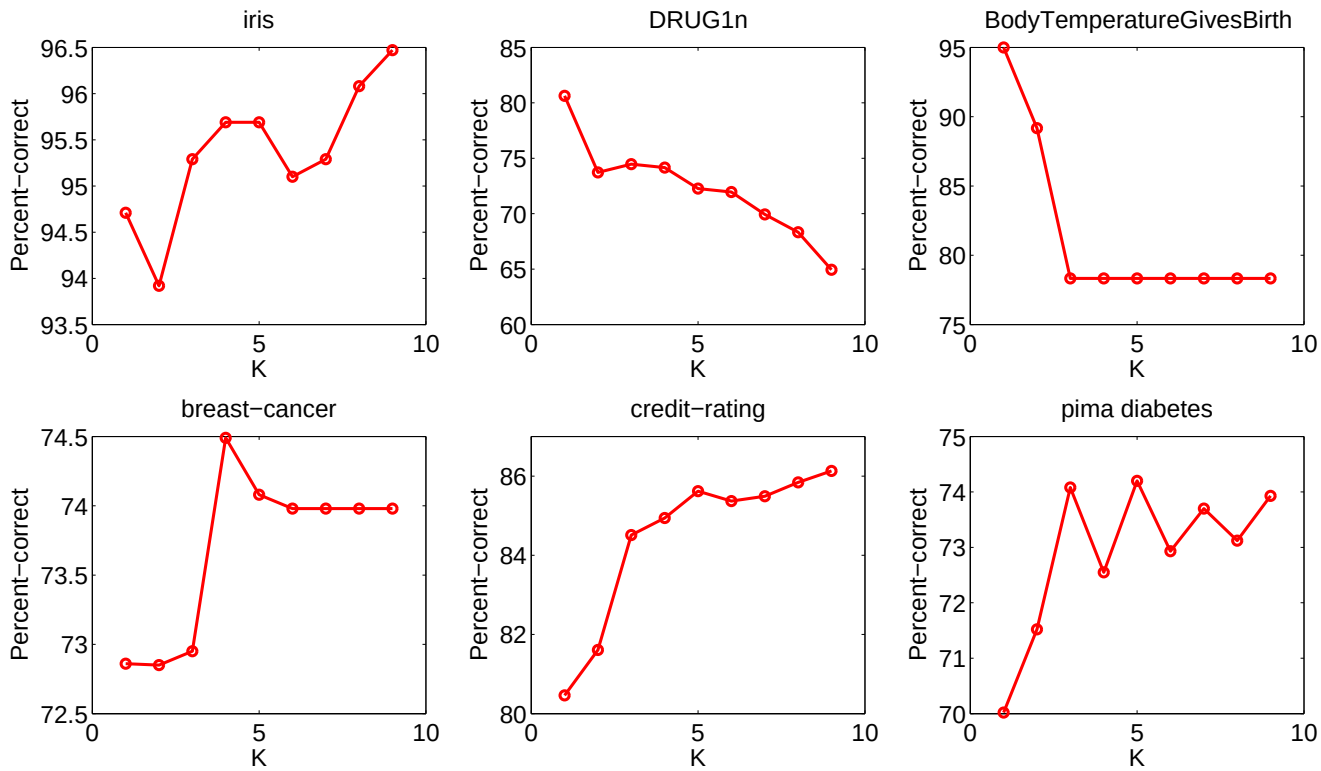
El algoritmo K-NN básico



Ejemplo de aplicación del algoritmo K-NN básico sobre el dataset Drug1n con $K = 11$. Distribución de los casos en el plan dado por (Na,K)

El algoritmo K-NN básico

Ejemplo de la no monotocidad del porcentaje de bien clasificados en función de K



Índice

Variantes del clasificador K-NN básico

Variantes del clasificador K-NN

- K-NN con rechazo
- K-NN con distancia media
- K-NN con distancia mínima
- K-NN con pesado de vecinos
- K-NN con pesado de variables

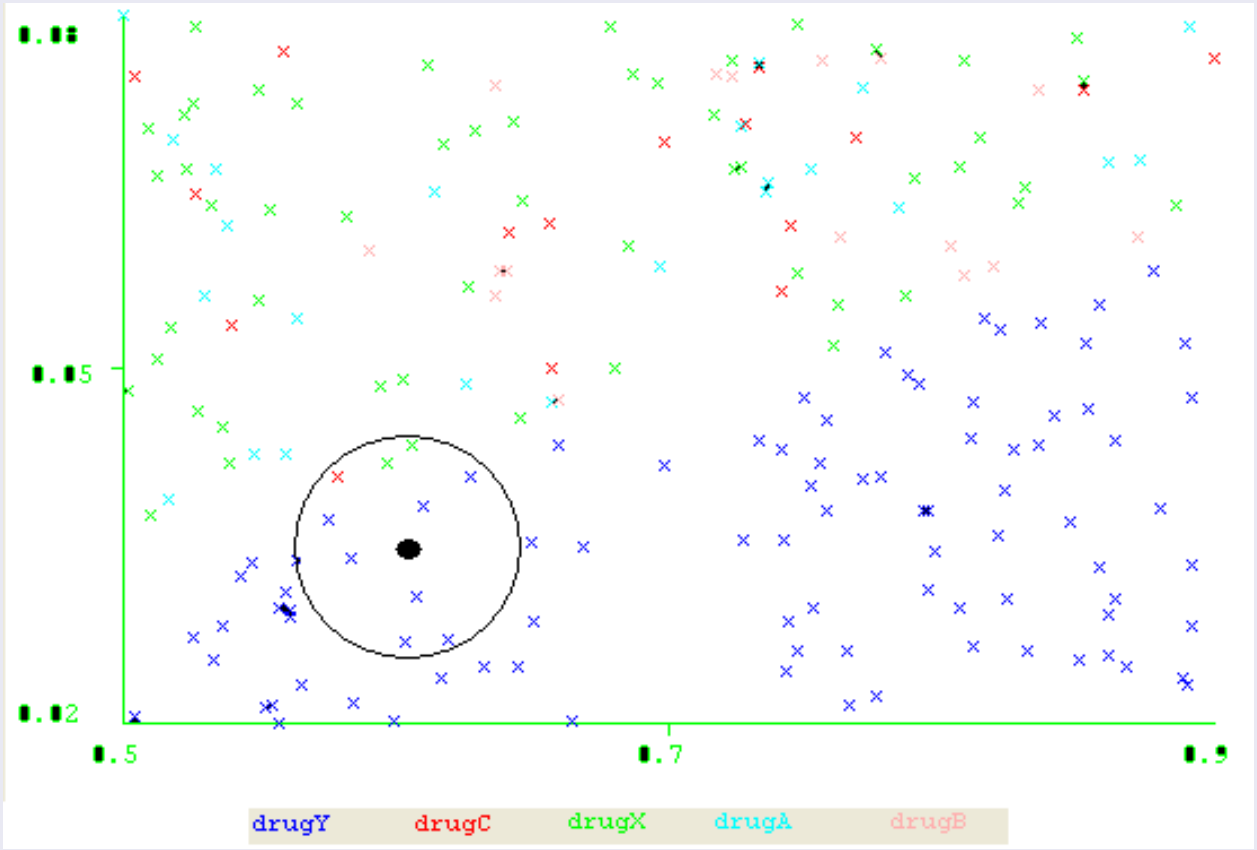
Variantes del clasificador K-NN básico

K-NN con rechazo

- Para clasificar un caso exigo ciertas garantías
- Si no las tengo puedo dejar el caso sin clasificar
  - Umbral prefijado (por ejemplo, $\frac{K}{m}$, donde m es el número de clases)
  - Mayoría absoluta

Variantes del clasificador K-NN básico

K-NN con distancia media



Ejemplo de ilustración del K-NN con distancia media sobre el dataset Drug1n con $K = 11$.

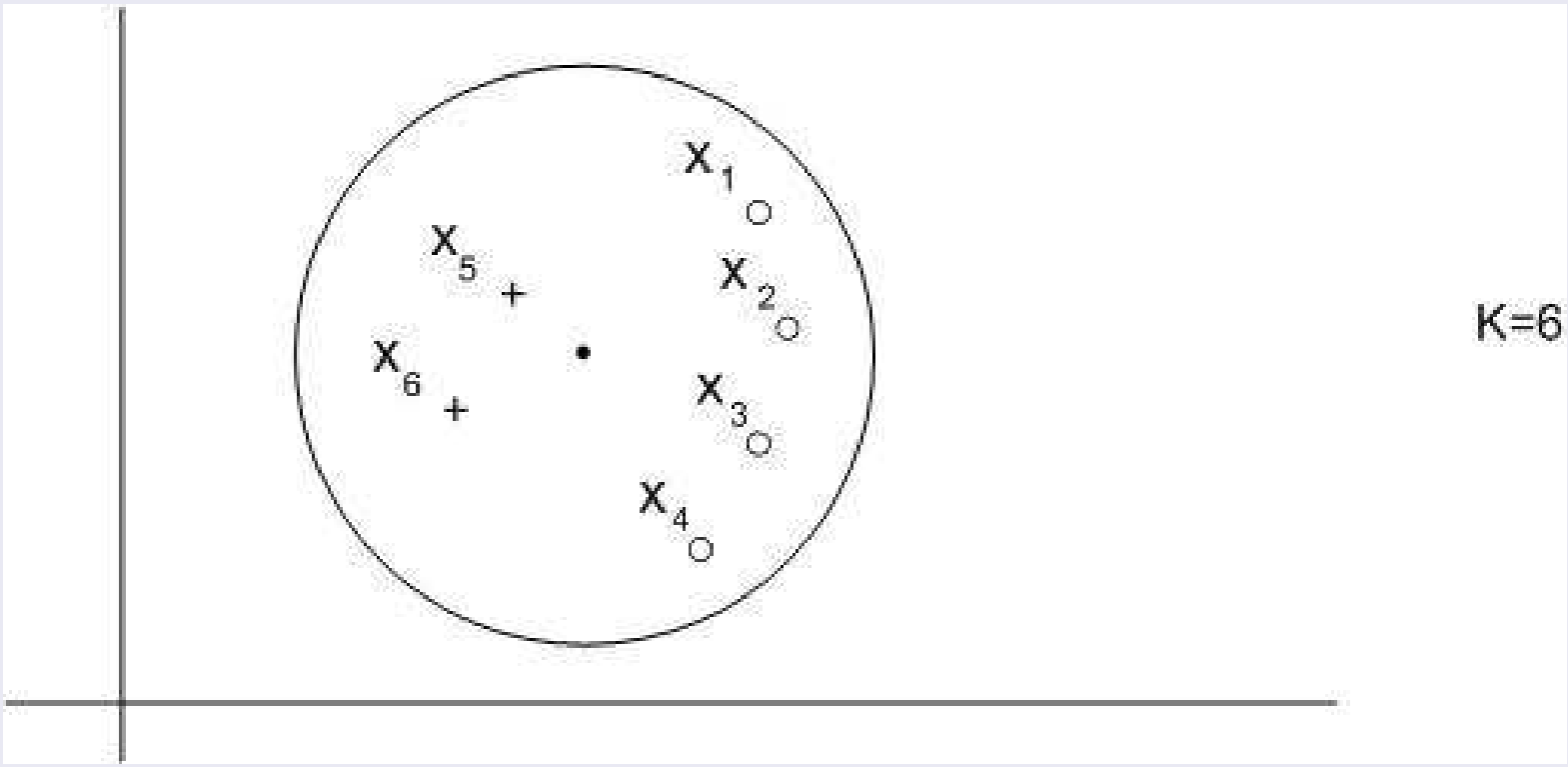
Variantes del clasificador K-NN básico

K-NN con distancia mínima

- Seleccionar un caso por clase (ej. el más cercano al baricentro de la clase)
- Reducción de la dimensión del fichero almacenado de N a m
- Ejecutar un 1-NN a dicho fichero reducido
- Efectividad condicionada a la homogeneidad dentro de las clases. A mayor homogeneidad más efectivo

Variantes del clasificador K-NN básico

K-NN con pesado de vecinos



Ejemplo de ilustración del K-NN con pesado de casos seleccionados

Variantes del clasificador K-NN básico

K-NN con pesado de vecinos

	$d(\mathbf{x}_i, \mathbf{x})$	w_i
$\mathbf{x}_1$	2	0,5
$\mathbf{x}_2$	2	0,5
$\mathbf{x}_3$	2	0,5
$\mathbf{x}_4$	2	0,5
$\mathbf{x}_5$	0,7	$1/0,7$
$\mathbf{x}_6$	0,8	$1/0,8$

Peso a asignar a cada uno de los 6 objetos seleccionados

Variantes del clasificador K-NN básico

K-NN con pesado de variables

- Mismo peso a todas las variables:

$$d(\mathbf{x}, \mathbf{x}_r) = \sum_{j=1}^{n} (x_j - x_{rj})^2$$

- Distinto peso a cada variable:

$$d(\mathbf{x}, \mathbf{x}_r) = \sum_{j=1}^{n} w_j (x_j - x_{rj})^2$$

- Determinar w_j a partir de $I(X_j, C)$ la cantidad de infomación mutua entre X_j y C

Variantes del clasificador K-NN básico

Ejemplo

Dado el siguiente conjunto de entrenamiento, determinar, usando herramientas de la teoría de la información, cuales de las variables es más relevante dada la clase.

X_1	X_2	C
0	0	1
0	0	1
0	0	1
1	0	1
1	0	1
1	1	1
0	1	0
0	1	0
0	1	0
1	1	0
1	1	0
1	0	0

Índice

Selección de prototipos

Aproximaciones

- Edición de Wilson
- Condensación de Hart

Selección de prototipos

Edición de Wilson

- Someter a prueba a cada uno de los elementos del fichero de casos inicial
- Para cada caso se compara su clase verdadera con la que propone un clasificador K-NN obtenido con todos los casos excepto el mismo
- Si ambas clases no coincidan, el caso es eliminado
- Edición de Wilson repetitiva parando el procedimiento cuando en 2 selecciones sucesivas no se produzcan cambios

Selección de prototipos

Condensación de Hart (1968)

- Para cada caso, y siguiendo el orden en el que se encuentran almacenados los casos en el fichero, se construye un clasificador K-NN con tan sólo los casos anteriores al caso en cuestión
- Si el caso tiene un valor de la clase distinto al que le asignará el clasificador K-NN, el caso es seleccionado
- Si por el contrario la clase verdadera del caso coincide con la propuesta por el clasificador K-NN, el caso no se selecciona
- El método es dependiente del orden en que se encuentren almacenados los casos en el fichero

Selección de prototipos

Pseudocódigo del algoritmo de condensación de Hart

1. Se parte de un subconjunto S que contiene la primera instancia del conjunto de entrenamiento. El resto de las instancias forman el subconjunto T
2. Clasificar las instancias de T usando S. Cada instancia mal clasificada se transfiere de T a S
3. Volver al paso 2 hasta que ninguna transferencia ocurra