Métodos Basados en Casos y en Vecindad

CC52A - Inteligencia Artificial

Gonzalo Ríos D.

DCC - UChile

Otoño 2010

Definición

Una **distancia** *sobre un conjunto* Ω *es una función* $d{:}\Omega \times \Omega \rightarrow \mathbb{R}$, *tal que:*

- $d(i,j) \geq 0, \ \forall i,j \in \Omega$
- $d(i,i)=0, \ \forall i \in \Omega$
- $d(i,j)=d(j,i), \ \forall i,j \in \Omega$

Una distancia es **métrica** *si cumple la siguiente propiedad:*

- $d(i,j) \leq d(i,k) + d(k,j), \ \forall i,j,k \in \Omega$

Ejemplo

1. *Distancia de Minkowsky:* $d(x,y) = \left(\sum |x_i - y_i|^q\right)^{\frac{1}{q}}, \ \forall q \geq 1$
2. *Distancia Mahalanobis:* $d(x,y) = \sqrt{(x-y)^T \Sigma^{-1} (x-y)}$
3. *Distancia del Coseno:* $d(x,y) = \arccos\left(\frac{x^T y}{\|x\| \|y\|}\right)$

Definición

Una **similaridad** *sobre un conjunto* Ω *es una función* $s{:}\Omega \times \Omega \rightarrow \mathbb{R}$, *tal que*

1. $0 \leq s(i,j) \leq 1, \forall i,j \in \Omega$
2. $1{=}s(i,i) \geq s(i,j),\ \forall i,j \in \Omega$
3. $s(i,j){=}s(j,i),\ \forall i,j \in \Omega$

Proposición

Si s(i,j) es una similaridad, entonces las siguientes funciones son distancias:

1. $d(i,j){=}\sqrt{1-s(i,j)}$
2. $d(i,j){=}1-s(i,j)$
3. $d(i,j){=}\sqrt{1-s^2(i,j)}$

Definición

Una **vecindad** *de un punto x es un conjunto* $V \subseteq \Omega$ *tal que* $x \in V$. *A los puntos* $y \in V\text{-}\{x\}$ *se le llaman vecinos del punto x.*

Definición

Una **bola** *con centro x y radio r es el conjunto* $B(x,r)=\{y \in \Omega | d(x,y) \leq r\}$

Definición

Una **K-vecindad** *de un punto x es un conjunto* $V \subseteq \Omega$ *tal que* $x \in V$ *y* $|V|=K+1$

Podemos notar que una bola es una vecindad que asegura un mínimo de similitud (máximo de distancia) entre el punto y sus vecinos, mientras que una k-vecindad asegura un número fijo de vecinos del punto x.

Un algoritmo de **clasificación** muy usuado es el de Nearest Neighbour, que se basa a clasificar una nueva instancia x en la clase del vecino más cercano del conjunto de entrenamiento. Ejemplo:
Tenemos las variables "Altura" y "Peso", y la variable de la clase es "Colesterol", si tenemos la instancia x=(185,90), entonces calculemos las distancias Euclidiana y de Manhattan

Altura	**Peso**	**Colesterol**	$\mathbf{d_E(x,y)}$	$\mathbf{d_M(x,y)}$
175	99	Si	13.454	19.0
176	78	No	14.213	20.0
174	92	**Si**	11.18	**13.0**
191	98.5	**No**	**10.404**	14.5

Si usamos la distancia Euclidiana, el vecino mas cercano es (191,98.5), por lo que se clasifica como "No". En cambio, si usamos la distancia de Manhattan, el vecino más cercano es (174,92), por lo que se clasifica como "Si".

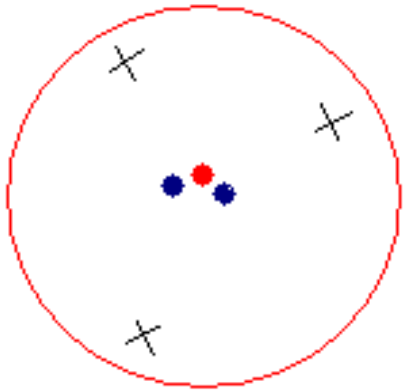
Podemos ver que este algoritmo es muy sensible a la función de distancia tomada y a la presencia de outlayers. Una alternativa más robusta de este algoritmo es el conocido como K-Nearest Neighbour, que asigna la clase más frecuente de entre sus K vecinos más cercanos.
En nuestro ejemplo anterior, si consideramos K=3, entonces:

Altura	**Peso**	**Colesterol**	$\mathbf{d_E(x,y)}$	$\mathbf{d_M(x,y)}$
175	99	**Si**	**13.454**	**19.0**
176	79	No	14.213	20.0
174	92	**Si**	**11.18**	**13.0**
191	98.5	**No**	**10.404**	**14.5**

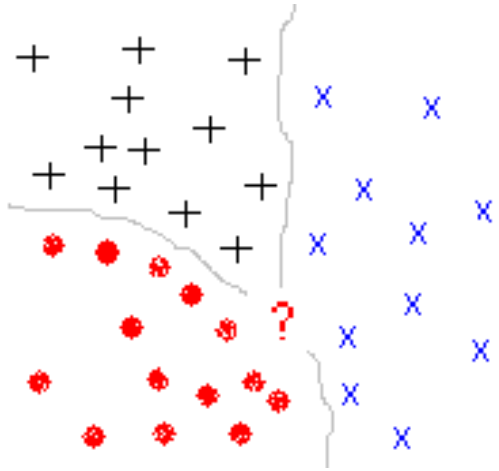
Luego, en ambos casos se clasifica con la clase "Si", ya que obtuvo una frecuencia del 66.7%

Métodos Basados en Vecindades

Problemas del algoritmo K-NN



5-Vecindad



Frontera de Decisión

- El algoritmo K-NN solo toma en cuenta el número de vecinos de cada clase, y descarta la distancia de los vecinos y la geometría de la vecindad.
- Las regiones cercanas a la frontera de decisión son muy dudosas y de poca garantía de que la clasificación sea correcta.
- Costo computacional alto

- **Regla K-NN con rechazo:** Se fija un umbral a priori, y la clasificación sólo se realiza si el número de votos recibida por la clase más votada supera dicho umbral. El umbral recomendado oscila entre K/M y K, siendo M el número de clases. Otra alternativa es la de establecer una mayoría absoluta, es decir, que la clase con mayor votos tenga una diferencia mayor que un cierto umbral con las demás clases.

- **Regla K-NN con rechazo:** Se fija un umbral a priori, y la clasificación sólo se realiza si el número de votos recibida por la clase más votada supera dicho umbral. El umbral recomendado oscila entre K/M y K, siendo M el número de clases. Otra alternativa es la de establecer una mayoría absoluta, es decir, que la clase con mayor votos tenga una diferencia mayor que un cierto umbral con las demás clases.
- **Regla K-NN por distancia media:** A partir de los K-vecinos más próximos, a un nuevo caso a clasificar le es asignada a la clase cuya distancia media es menor dentro de las clases.

Métodos Basados en Vecindades

Variantes del algoritmo K-NN

- **Regla K-NN con rechazo:** Se fija un umbral a priori, y la clasificación sólo se realiza si el número de votos recibida por la clase más votada supera dicho umbral. El umbral recomendado oscila entre K/M y K, siendo M el número de clases. Otra alternativa es la de establecer una mayoría absoluta, es decir, que la clase con mayor votos tenga una diferencia mayor que un cierto umbral con las demás clases.
- **Regla K-NN por distancia media:** A partir de los K-vecinos más próximos, a un nuevo caso a clasificar le es asignada a la clase cuya distancia media es menor dentro de las clases.
- **Clasificador de la distancia mínima**: Primero se selecciona un prototipo o representante para cada clase. Luego, el clasificador asignará un nuevo caso x la clase cuyo representante se encuentre más próximo a ella.

El concepto de Nearest Centroid Neighbourhood se basa en:

- Los k vecinos de x deben estar lo más cerca posible
- El centroide de la vecindad debe estar lo más cerca posible de x

Este tipo de vecindades se denomina **vecindad envolvente**.

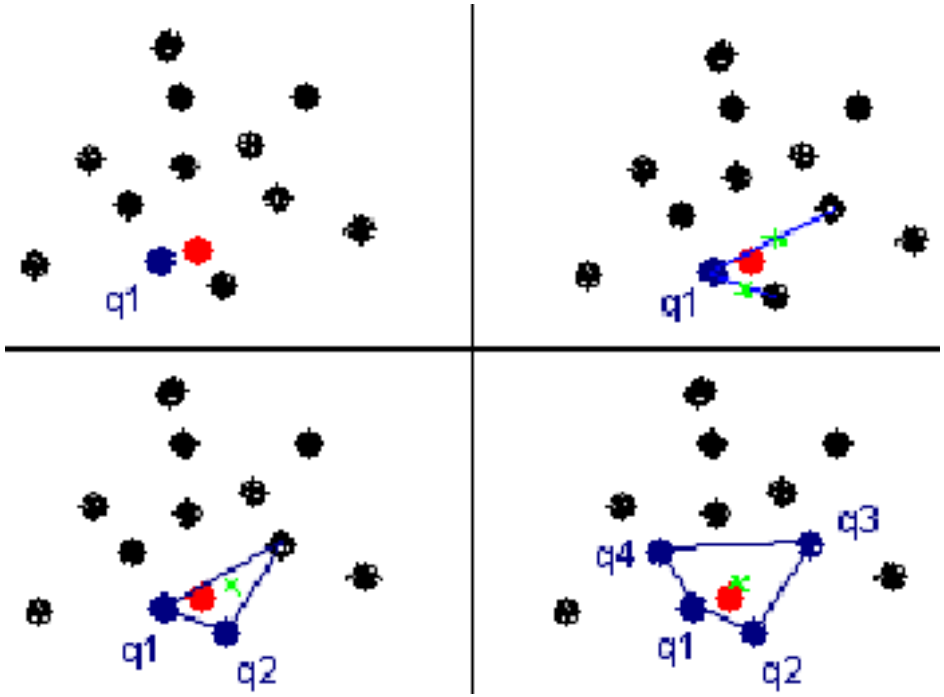
Algoritmo

Vecindad Envolvente

1. *El vecino más cercano a x se denomina* q_1
2. *El i-ésimo vecino* q_i, $i \geq 2$ *es aquel que hace que el centroide del conjunto* $Q_i = \{q_1, ..., q_i\}$ *sea lo más cercano posible a x.*

Métodos Basados en Vecindades

Nearest Centroid Neighbourhood

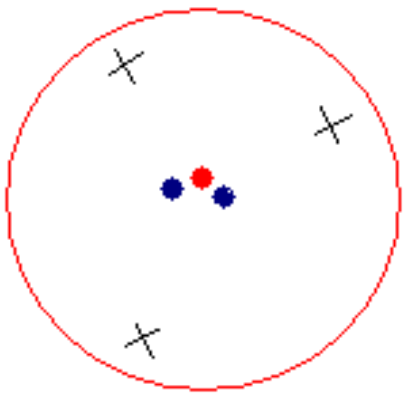


Métodos Basados en Vecindades

Pesado de casos y de atributos

En el algoritmo K-NN, todos los vecinos tenían el mismo peso entre sí, pero una idea bastante razonable sería que los casos que estén más cercanos a x tengan mayor importancia que los que están más lejos. Para esto, se le asigna un peso W_i a cada vecino x_i. Algunas de las opciones son:

- $W_i = \frac{1}{d(x,x_i)}$

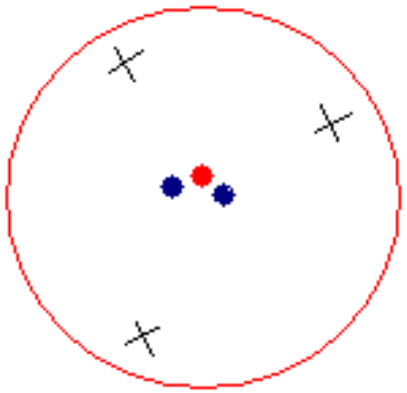


Métodos Basados en Vecindades

Pesado de casos y de atributos

En el algoritmo K-NN, todos los vecinos tenían el mismo peso entre sí, pero una idea bastante razonable sería que los casos que estén más cercanos a x tengan mayor importancia que los que están más lejos. Para esto, se le asigna un peso W_i a cada vecino x_i. Algunas de las opciones son:

- $W_i = \frac{1}{d(x,x_i)}$
- $W_1 \geq W_2 \geq .... \geq W_k$, donde $\{W_i\}_{i=1}^{K}$ están fijos apriori

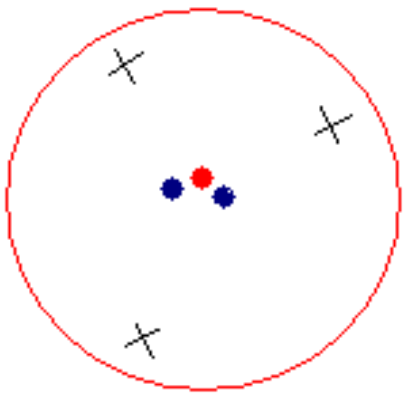


Métodos Basados en Vecindades

Pesado de casos y de atributos

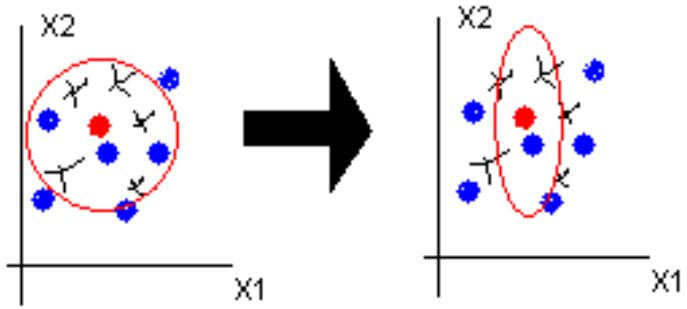
En el algoritmo K-NN, todos los vecinos tenían el mismo peso entre sí, pero una idea bastante razonable sería que los casos que estén más cercanos a x tengan mayor importancia que los que están más lejos. Para esto, se le asigna un peso W_i a cada vecino x_i. Algunas de las opciones son:

- $W_i = \frac{1}{d(x,x_i)}$
- $W_1 \geq W_2 \geq .... \geq W_k$, donde $\{W_i\}_{i=1}^{K}$ están fijos apriori
- W_i=Probabilidad a priori de clase en que pertence el vecino x_i



Un punto muy importante en el algoritmo K-NN es la función de distancia, y la más utilizada en estos casos es d(x,y)=$\sum_{i=1}^{n} W_i(x_i - y_i)^2$, luego la elección de los pesos W_i para cada atributo es muy importante. Una opción de estimar el peso W asociado a la variable X es usar la información mutua entre la variable X y la variable a clasificar C, que es la reducción de incertidumbre de una variable al conocer los valores de la otra.

$$I(X,C) = H(X) - H(X|C) = \sum_{x,\theta} \log p(x,\theta) \frac{p(x,\theta)}{p(x)p(\theta)}$$



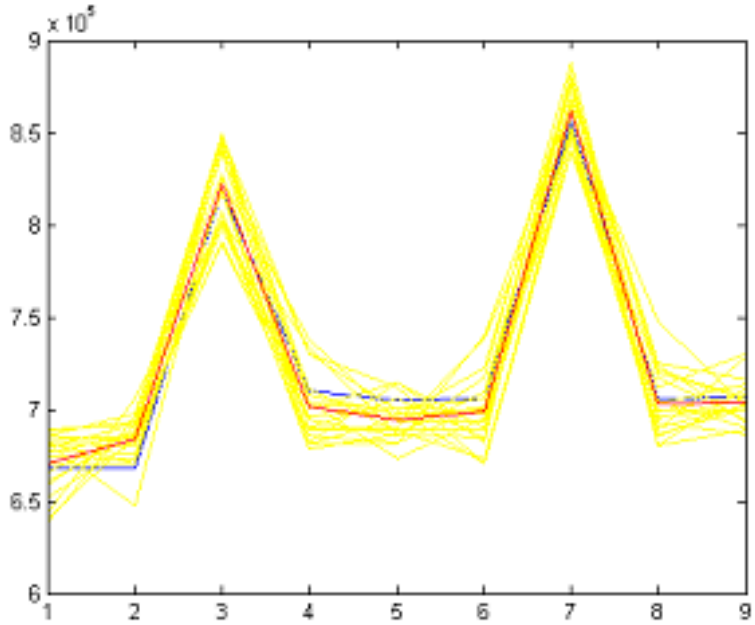
El enfoque principal de los métodos basados en vecindades son para clasificación, pero éste no es el único enfoque. Con una pequeña variación del algoritmo K-NN, éste se puede usar para regresión.
Si tenemos los datos $\{(x_{i1}, ..., x_{in}, y_i)\}_{i=1}^{m}$, donde la variable Y es continua, se puede construir un regresor de la siguiente manera:

Algoritmo

K-NN para Regresión

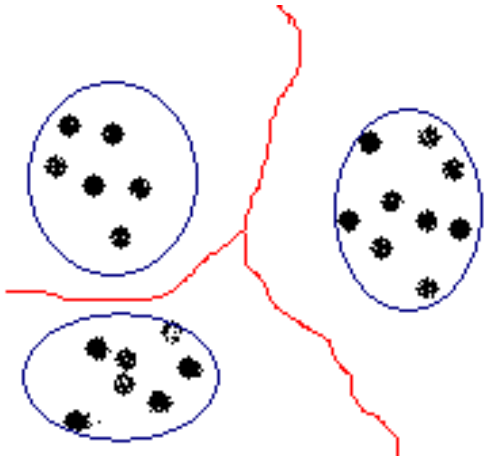
- *Dado un nuevo dato x, calculamos su vecindad* $V_x = \{q_1, ..., q_K\}$
- *Luego, construimos la predicción de Y para x a partir de* $\{y_1, ..., y_K\}$, *con algún estimador:*
  - $y = \frac{1}{K} \sum_{i=1}^{K} y_i$
  - $y = \frac{\sum_{i=1}^{K} s(x,q_i) * y_i}{\sum_{i=1}^{K} s(x,q_i)}$, *donde* $s(x,q_i)$ *es la similitud entre x y* q_i

Todas las variaciones posibles del algoritmo K-NN son aplicables en este caso.



El gráfico anterior muestra la vecindad (amarillo) del dato a predecir (azul) y la predicción (rojo). Este problema es una serie de tiempo, en donde se tiene la historia de ventas de 20.000 empresas a lo largo de 21 meses.

Otro enfoque muy utilizado, usando la noción de vecindades, es el de clustering o agrupamiento. La idea principal es, partiendo de un número apriori de grupos, se construye una partición del espacio, de modo que cada grupo difiera de forma considerable de los otros grupos. A diferencia de la clasificación, en el clustering los grupos se determinan a partir de los datos.



El algoritmo más usado para esta tarea es el de K Medias, o K Means, y se basa en un proceso iterativo de minimizar la distancia entre los datos y el centro del grupo más cercano.

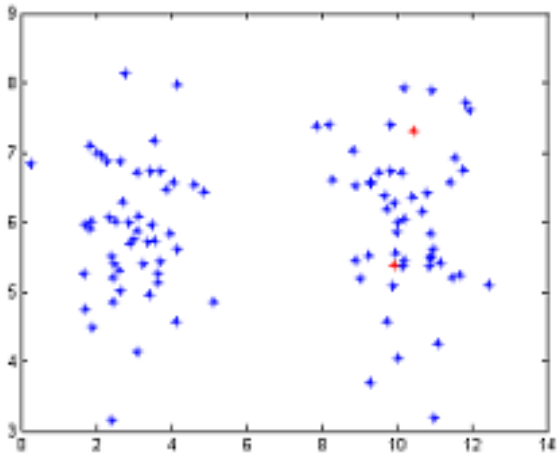
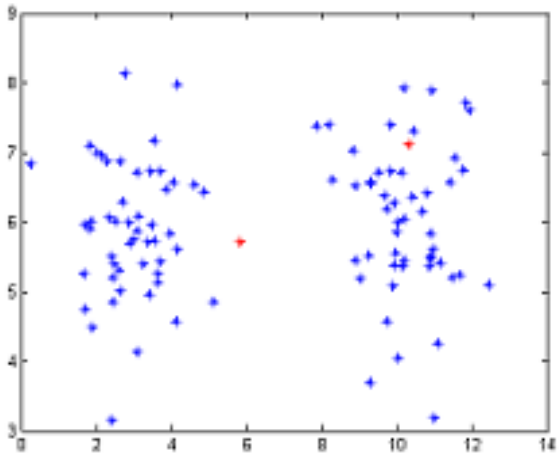
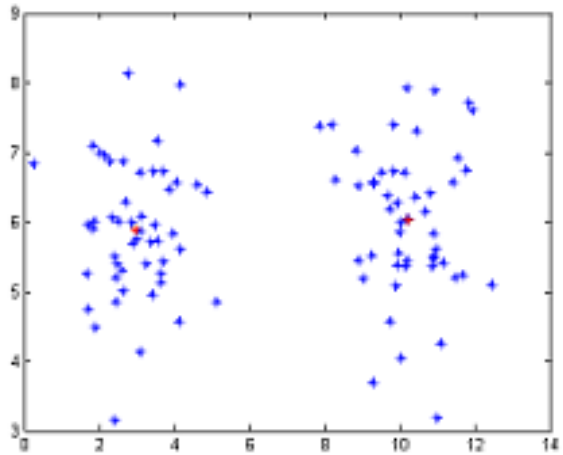
Algoritmo

K Medias

1. *Se inicializa con K prototipos (centros de los grupos)* $A_1, .., A_K$ *distribuidos de forma uniforme o al azar.*
2. *Se calcula para cada instancia* x_i *se calcula el prototipo más próximo, es decir,* $A_g = argmin\{d(x_i, A_j)\}_{j=1}^{K}$
3. *Se desplaza cada uno de los prototipos al centro de masas de su conjunto, es decir,* $A_g = \frac{\sum_{i=1}^{m} x_{g_i}}{m}$
4. *Si el desplazamiento de todos los prototipos es menor a una tolerancia* ε*, parar. De lo contrario, volver a 2.*

Métodos Basados en Vecindades

K Medias para Clustering

El problema principal del algoritmo de K Medias es que, si no se escoge un buen punto inicial, entonces el algoritmo puede converger a un mínimo local, es decir, una solución sub óptima. El problema general a optimizar es:

$$\min J = \sum_{i=1}^{K} \sum_{n=1}^{m} M_{i,n} d^2(x_n - A_i)$$

$$M_{i,n} = \begin{cases} 1 & \text{si} \quad d(x_n - A_i) < d(x_n - A_s),\ \forall s \neq i, s = 1, .., k \\ 0 & \text{en} \quad \text{otro caso} \end{cases}$$