Métodos de Data Mining (Parte IV)

Aprendizaje Basado en Instancias

Jhon Jairo Padilla A., PhD.

Introducción

- Los datos suelen ser almacenados palabra por palabra.
- Se determina si un dato pertenece a una clase si se encuentra en una “zona de cobertura” de esa clase.
- Para determinar si está en la zona de cobertura, se calcula la distancia del dato al punto central de la clase.

Aprendizaje basado en instancias (IBL-Instance Based Learning)

- Hay un conjunto de prueba que se usa para aprender las clases.
- Se usa una función de distancia para determinar cuál miembro del conjunto de entrenamiento está más cerca del dato o instancia que se quiere clasificar.
- Una vez que se ha localizado la instancia de entrenamiento más cercana, su clase se toma como la clase de la instancia de prueba.
- Por tanto, se debe definir la función que determina la distancia. Si los atributos son numéricos esto no tiene problema.

Cálculo de la distancia

- Puede haber varias alternativas, pero la más usada es la distancia euclideana.
- Para k atributos diferentes (a_1, a_2,...a_k) en una medición o instancia, la distancia euclidiana entre el dato 1 y el 2 sería:

$$\sqrt{\left(a_1^{(1)} - a_1^{(2)}\right)^2 + \left(a_2^{(1)} - a_2^{(2)}\right)^2 + \ldots + \left(a_k^{(1)} - a_k^{(2)}\right)^2}.$$

Indica que pertenece al Dato 1

Indica que pertenece al Dato 2

Cálculo de la distancia

- Cuando se calculan las distancias, no es necesario realizar la operación de raíz cuadrada. Se pueden usar las sumas de cuadrados directamente.
- Otra alternativa es la métrica de cuadras de ciudad o Manhattan, donde la diferencia entre valores de atributos no es elevada al cuadrado, sino simplemente sumada (su valor absoluto)
- Otras alternativas toman potencias mayores que el cuadrado. Altas potencias incrementan la influencia de grandes diferencias con respecto a las pequeñas diferencias.
- Generalmente, la distancia euclideana representa un buen compromiso.

Normalización de parámetros

- Diferentes atributos son medidos en diferentes escalas, por lo que si se usa la distancia euclideana directamente, los efectos de algunos atributos podrían ser empequeñecidos por otros que tienen escalas más grandes de medición.
- En consecuencia, usualmente se normalizan los valores de los atributos para que permanezcan entre 0 y 1 con la siguiente expresión:

$$a_i = \frac{v_i - \min v_i}{\max v_i - \min v_i}$$

Normalización de parámetros

- Esta formula implica el uso de atributos numéricos.
- Dicho valor es el que se usa para restar y elevar al cuadrado
- Cuando los atributos son nominales, se asigna 1 a la distancia cuando son diferentes y 0 cuando son iguales. No se requiere normalización entonces.
- Para parámetros perdidos, se toma la máxima distancia posible (1) según sean los valores nominales o numéricos.

Eficiencia de IBL

- Este método es simple y efectivo, pero es muy lento.
- Se requiere encontrar cuál miembro del conjunto de entrenamiento está más cerca a una instancia de prueba. Por tanto, se debe calcular la distancia desde cada miembro del conjunto de entrenamiento y seleccionar la más pequeña.
- Este procedimiento es lineal en el número de instancias de entrenamiento: El tiempo que toma es proporcional al número de instancias en el conjunto de entrenamiento (E) y en el conjunto de prueba (P). ($T \alpha E \times P$)

Vecinos Cercanos (Nearest Neighbors)

- Este método es más eficiente y representa el conjunto de entrenamiento como un árbol, aunque no es tan obvio cómo.
- La estructura que se obtiene es un árbol KD (KD-tree)
- Este es un árbol binario que divide el espacio de entrada con un hiperplano y luego divide cada partición otra vez, de forma recursiva.
- Todas las divisiones son hechas paralelas a uno de los ejes, ya sea vertical u horizontalmente, para el caso bidimensional.
- La estructura de datos es llamada KD-tree porque almacena un conjunto de puntos en un espacio k-dimensional, donde k es el número de atributos.

Cada punto en el conjunto de entrenamiento corresponde a un nodo del árbol, y hasta la mitad son nodos hoja

Ejemplo para K=2

Cada región contiene al menos un punto, o quizás ningún punto.

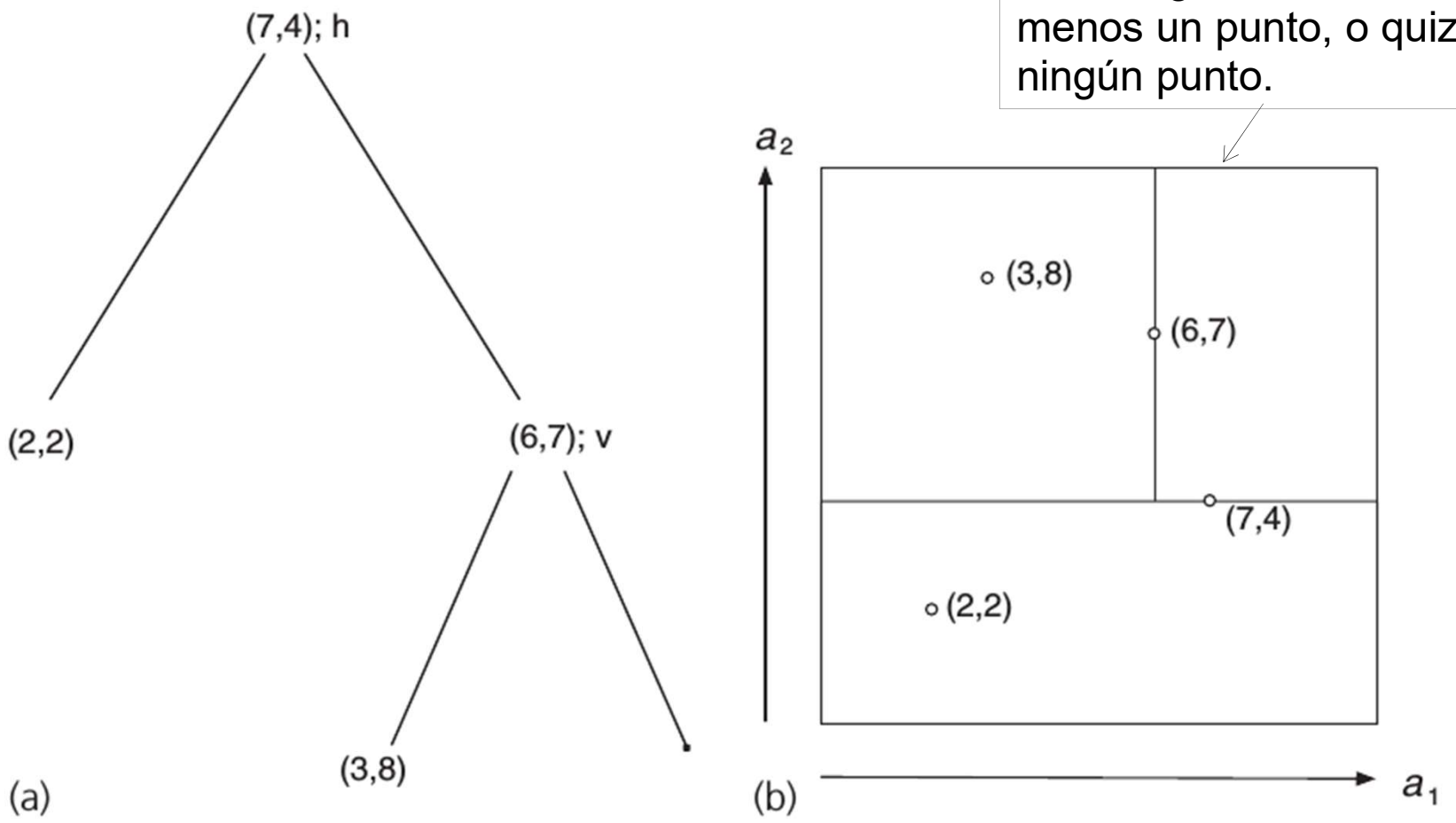
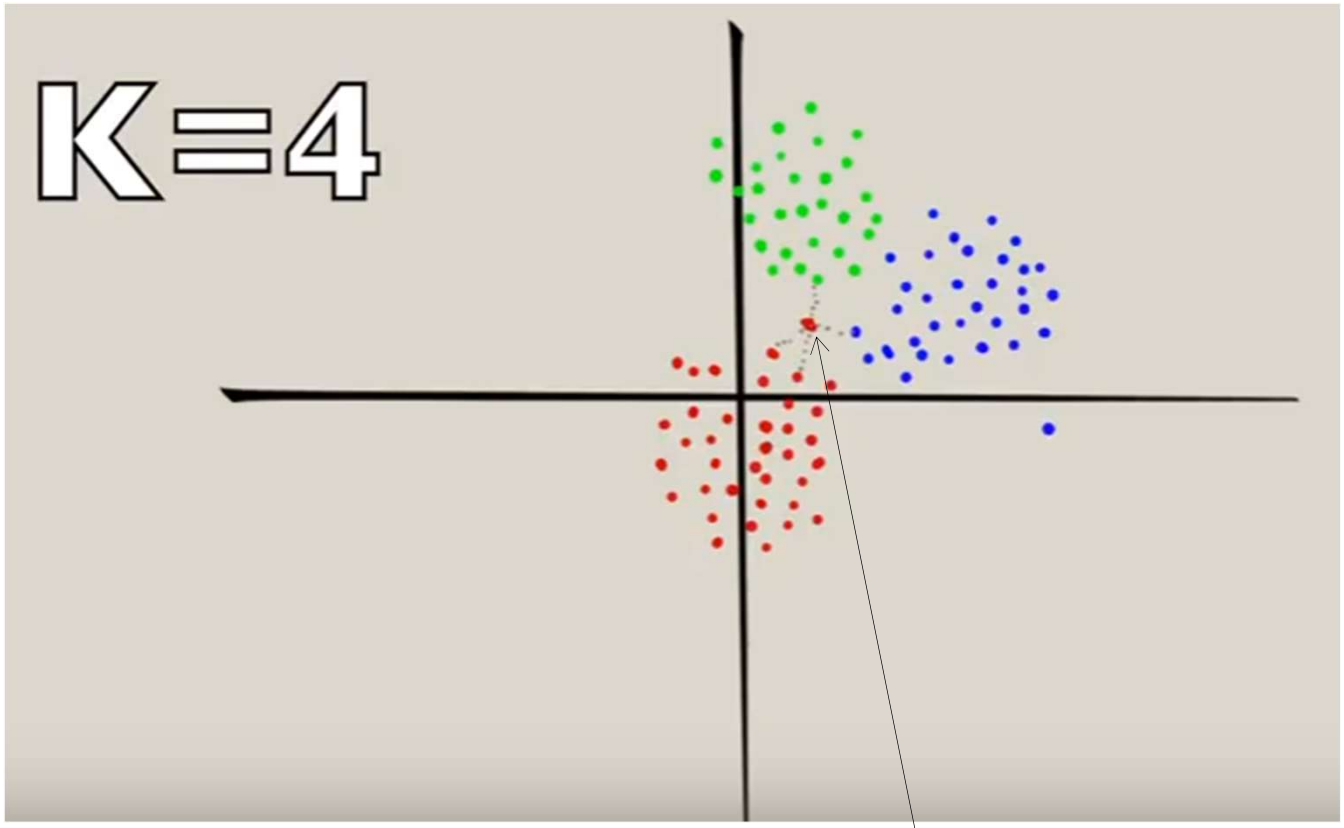


Figure 4.12 A *k*D-tree for four training instances: (a) the tree and (b) instances and splits.

KNN (K vecinos cercanos)



Se escoge la categoría Roja porque hay dos vecinos cercanos de los 4 mas cercanos que estan en rojo, 1 verde y uno azul

Cómo usar el árbol para clasificar el punto objetivo (target)?

- Para ubicar un vecino cercano de un punto objetivo dado, siga el árbol hacia abajo desde su raíz para encontrar la región que contiene el punto objetivo (target)

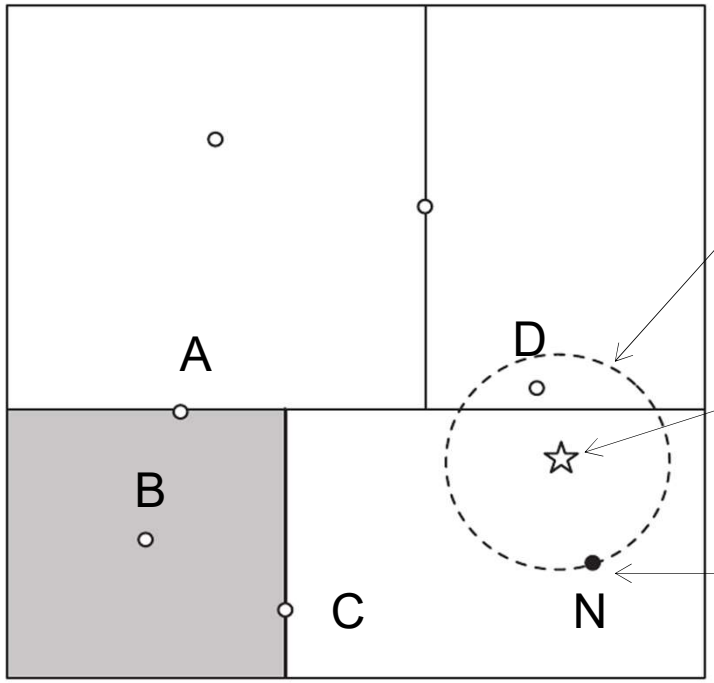


Figure 4.13 Using a *k*D-tree to find the nearest neighbor of the star.

Cualquier vecino cercano debe debe estar cerca, es decir, dentro de del circulo punteado

Target

Dato de entrenamiento elegido (no necesariamente es el más cercano)

Encontrando el vecino cercano al objetivo

- Para determinar si existe un vecino cercano dentro del círculo, primero chequee si es posible para un vecino cercano estar dentro del hermano del nodo.
- El hermano del nodo negro (N) está sombreado en la figura (punto B), y el círculo no lo intersecta, por lo que el hermano no puede contener un vecino cercano.
- Entonces regrese al nodo padre (punto A) que cubre todo sobre la línea horizontal, y chequee el hermano de B.
- En este caso debe explorarse porque el área que él cubre intersecta con el círculo claramente.
- Para explorarlo, encuentre sus hijos (el punto original tiene dos tíos), chequee si ellos intersectan el círculo (el izquierdo (C) no lo hace, pero el derecho (D) sí lo hace), y descienda para ver si este contiene un punto cercano (lo hace)