Maestría en Inteligencia Artificial Aplicada

k-Vecinos Más Cercanos: *k*NN

(*k-Nearest Neighbors*)

Aprendizaje Automático

Dr. Luis Eduardo Falcón Morales

k Vecino Más Cercano: kNN

- Método muy usado para clasificación y regresión.
- Algoritmo supervisado.
- Método no paramétrico.
- Método de aproximación local.
- Se requiere tener almacenado todo el conjunto de datos de entrenamiento.

lazy learning

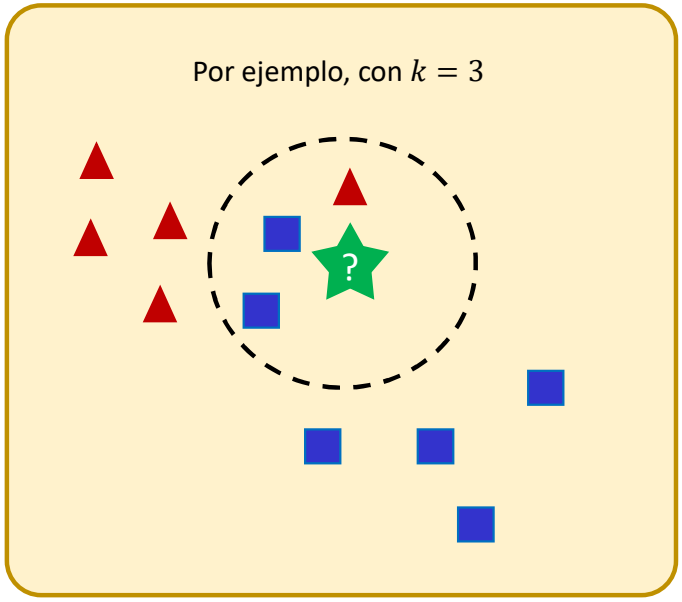
- El algoritmo kNN está dentro del tipo de métodos conocidos como *lazy learning*, es decir, en los que no se lleva a cabo ningún análisis de entrenamiento, sino hasta que se introduce un dato nuevo de prueba.
- Estrictamente un algoritmo de este tipo no está aprendiendo nada.
- Los *lazy learners* también se conocen como aprendizajes basados en instancias o en memoria (*instance-based learning*).

Algoritmo kNN

Conjunto de entrenamiento:

Se tiene un conjunto de n puntos m dimensional $\vec{x}_j = (x_{j1}, x_{j2}, \dots, x_{jm}) \in \mathbb{R}^m$, $j = 1,2, \dots, n$ los cuales pertenecen a Q clases diferentes. Cada punto está etiquetado con la clase a la que pertenece y todos en memoria.

Seleccionar el valor deseado para el entero k: este entero indica que se buscarán los k datos más cercanos a un dato dado en el conjunto de prueba y los cuales se someterán a votación.



Conjunto de prueba:

Sea $\vec{x}_p$ un dato nuevo no etiquetado.

Obtener la clase de mayor frecuencia de entre los k vecinos más cercanos a $\vec{x}_p$ y asignarlo a dicha clase.

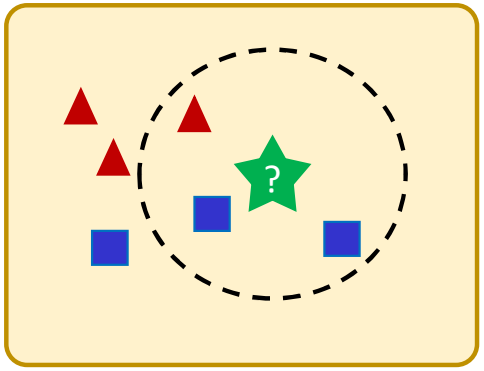
Métricas para kNN

Existen diversas métricas para calcular la distancia a los vecinos más cercanos.

- *Euclidiana*
- *Manhattan* o ℓ_1
- *Chebyshev* o máximo de las diferencias o ℓ_∞
- *Minkowski* o ℓ_p

Distancia *euclidiana* ℓ_2

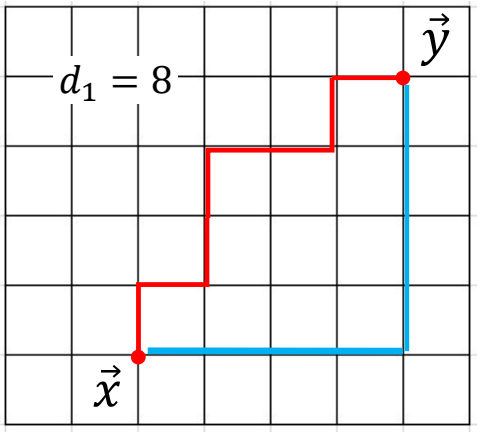
$$d_E(\vec{x},\vec{y}) = \sqrt{\sum_{j=1}^{m}(x_j - y_j)^2}$$

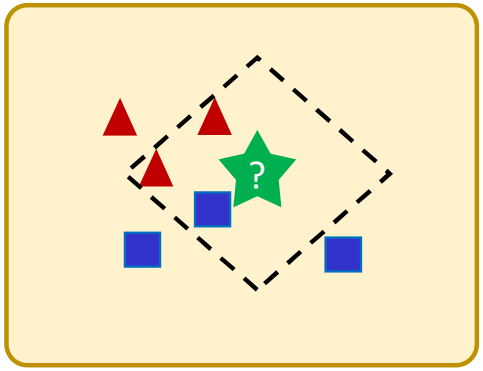


$k = 3$

Distancia *Mahattan* o ℓ_1

$$d_1(\vec{x},\vec{y}) = \sum_{j=1}^{m}|x_j - y_j|$$

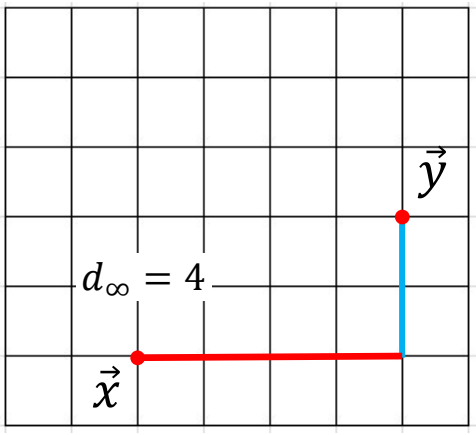


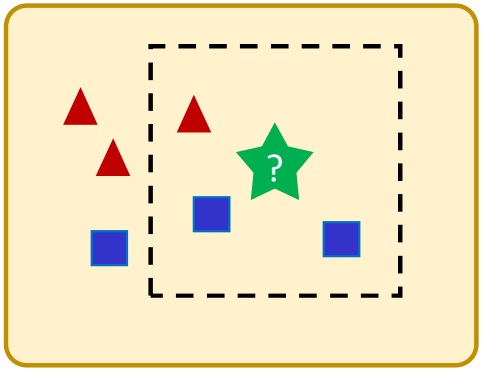


$k = 3$

Distancia *Chebyshev* o máximo o ℓ_∞

$$d_\infty(\vec{x},\vec{y}) = \max_{j\in\{1,2,\ldots,m\}} |x_j - y_j|$$

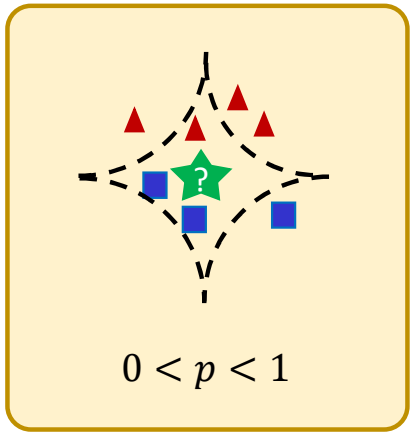




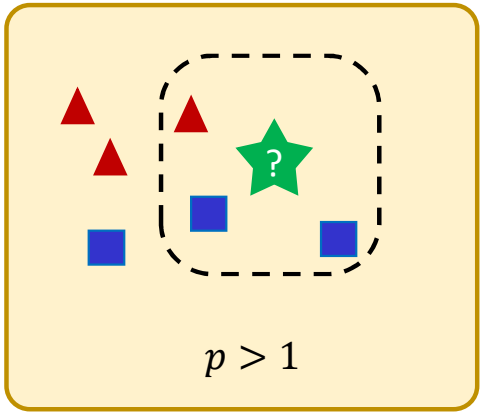
$k = 3$

Distancia *Minkowski* o ℓ_p

$$d_p(\vec{x},\vec{y}) = \left\{\sum_{j=1}^{m} |x_j - y_j|^p\right\}^{1/p}$$



$k = 3$

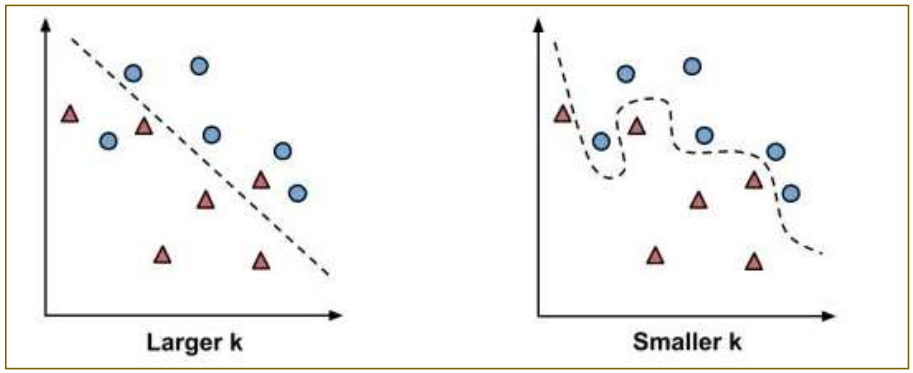


Comentarios adicionales sobre el método kNN:

- El caso $k = 1$ se conoce como el algoritmo del vecino más cercano.
- **kNN para Regresión**. Se puede extender el algoritmo kNN a vectores de la forma $(x_1, x_2, \dots, x_m, y)$ donde $y \in \mathbb{R}$, en cuyo caso a un dato nuevo del conjunto de prueba se le asigna el valor promedio (o por interpolación) de la variable dependiente de los k datos más cercanos.
- Una alternativa del método kNN es considerar las distancias de manera ponderada. Es decir, asignarle un peso mayor a los datos de una clase cuya información describe mejor el comportamiento de la variable dependiente, o bien, darle mayor peso a los datos más cercanos.
- Es muy importante estandarizar todos los datos a una misma escala uniforme antes de calcular todas las distancias, para minimizar el problema del sesgo.
- El método es lento ya que dado un dato nuevo de prueba debe buscar sobre todo el conjunto de datos cuáles son los que están dentro de su vecindad.

Selección del valor de k

Algoritmo kNN



Fuente: Machine Learning with R, 2013
B. Lantz. Packt Publishing.

- En el caso de tener solamente dos clases, seleccionar k entero impar, para evitar la probabilidad de empates en la votación.
- En general el valor de k es arbitrario y existen diversas reglas empíricas para elegirlo. Los valores más comunes son entre 3 y 11, y aplica para 2 o más clases.
- Mientras más pequeño el valor de k, por ejemplo $k = 1$, mayor posibilidad de generar un modelo sobreentrenado (overfitting).
- Una alternativa es realizar varias pruebas para diferentes valores de k y seleccionar aquel valor que realice la mejor clasificación con los datos de prueba.