Algoritmo de K Vecinos Cercanos (KNN)

El algoritmo de K Vecinos Cercanos (KNN) es un método de aprendizaje automático no paramétrico y basado e
Es conocido como un algoritmo 'vago' porque no aprende un modelo explícito, sino que utiliza directamente las

1. Concepto y características

- Modelo no paramétrico: no supone una función fija.
- Basado en instancias: memoriza los datos.
- Supervisado: clasificación y regresión.
- No supervisado: clustering espectral, manifold learning.
- Aplicaciones: detección de intrusos, genética, imágenes, predicción económica, compresión de datos.

2. Funcionamiento

1. Elegir k y una métrica de distancia.
2. Encontrar los k vecinos más cercanos.
3. Asignar etiqueta (clasificación) o promedio (regresión).

3. Métricas de distancia

• Euclídea: sqrt(sum((xi - yi)^2))
• Manhattan: sum(|xi - yi|)
• Chebyshev: max(|xi - yi|)
• Minkowski: (sum(|xi - yi|^p))^(1/p)
• Mahalanobis, Cityblock, L1, L2, Infinity, Seuclídea.
Nota: se recomienda estandarizar datos.

4. Elección de vecinos (k)

- k pequeño: más flexible, alta varianza.
- k grande: más estable, mayor sesgo.
- Usar k impar si hay clases pares.
- Alternativa: vecinos por radio.

5. Ventajas y desventajas

Ventajas: simple, no paramétrico, adaptable, buena precisión, robusto a outliers.
Desventajas: costoso en memoria y cálculo, poco escalable, sensible a la maldición de la dimensionalidad.

6. K vecinos no supervisados

Uso con sklearn.neighbors.NearestNeighbors.
Algoritmos: brute (O(D*N^2)), kd_tree (eficiente D<20), ball_tree (mejor en alta dimensión).

7. K vecinos como clasificador

Clase: KNeighborsClassifier.
En caso de empate, sklearn usa vecinos más cercanos o la primera clase.

8. K vecinos como regresor

Clase: KNeighborsRegressor.
Predicción: promedio de valores de vecinos.

9. Otros métodos

- Nearest Centroid Classifier.

- Nearest Neighbors Transformer.
- Neighborhood Components Analysis (NCA).
- Spectral Clustering.

10. Conclusiones

KNN es intuitivo y poderoso para tareas simples, pero costoso en datasets grandes o de alta dimensionalidad. Es recomendable combinarlo con selección de características o reducción de dimensionalidad.

Fuentes

• Scikit-learn: https://scikit-learn.org/stable/modules/neighbors.html
• Python Machine Learning (Raschka, Mirjalili, 3rd ed.)
• Apuntes del curso original