Aprendizaje Basado en Instancias (IBL)

Aprendizaje Automático

Ingeniería Informática

Fernando Fernández Rebollo y Daniel Borrajo Millán

Grupo de Planificación y Aprendizaje (PLG)
Departamento de Informática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

27 de febrero de 2009

En Esta Sección:

Definición de IBL

- Método para la aproximación de funciones objetivo de valores continuos o discretos.
- Fases:
  1. Entrenamiento: almacenamiento de todo el conjunto de datos disponibles
  2. Generalización: dado un nuevo dato, se extraen de memoria un conjunto de datos **similares**, que son utilizados para clasificar el nuevo dato
- ¿Cómo definimos el concepto de similitud?
- ¿Coste de clasificación? Todo el cómputo se realiza en tiempo de clasificación $\rightarrow$ Aprendizaje Perezoso (Lazy Learning)

K Nearest Neighbour

- Todas las instancias corresponden con puntos en un espacio de dimensión n ($\mathcal{R}^n$)
- Los vecinos más cercanos de una instancia vienen dados en término de la distancia euclídea:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n}(x_i[r] - x_j[r])^2} \tag{29}$$

- Si los atributos tienen diferentes rangos, se normaliza
- Si los atributos tienen valores simbólicos, ¿se numeran?
- El parámetro k identifica cuántos vecinos se utilizan para la decisión

Función Objetivo

- Dada una instancia x, y sus k vecinos: $x_1, \ldots, x_k$.
- Caso discreto: asignar el valor más común de entre los k más cercanos: $f : \mathcal{R}^n \rightarrow V, V = \{v_1, \ldots, v_s\}$

$$\hat{f}(x) = \arg_{v \in V} \max \sum_{i=1}^{k} \delta(v, f(x_i)) \tag{30}$$

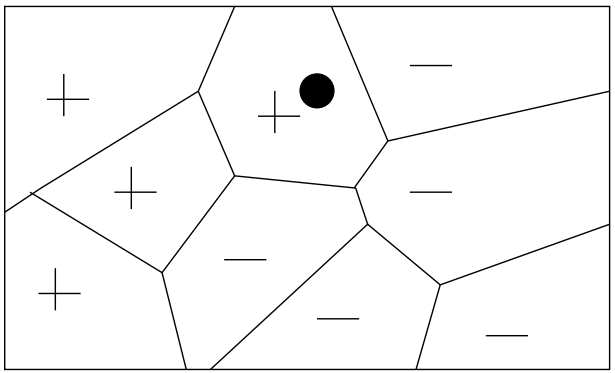
  donde $\delta(a, b) = 1$ si $a = b$, y $\delta(a, b) = 0$ en c.o.c.
- Caso continuo: valor medio de entre los k más cercanos: $f : \mathcal{R}^n \rightarrow \mathcal{R}$

$$\hat{f}(x) = \frac{\sum_{i=1}^{k} f(x_i)}{k} \tag{31}$$

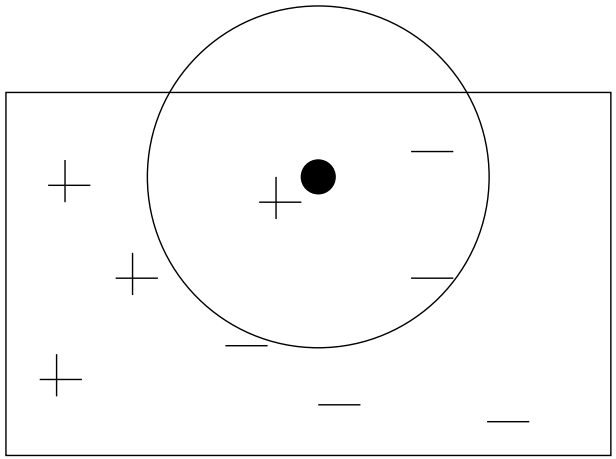
Definición de K

- El parámetro k identifica cuántos vecinos se utilizan para la decisión

(a) 1-NN



(b) 3-NN



- ¿Qué k elegimos? Proceso de prueba y error, dado que depende del problema.
- Las regiones que se forman con 1-NN se denominan regiones de Voronoi

Distance-Weighted K-NN

- Idea: ponderar la importancia que aporta cada vecino al valor de la función objetivo, en función de su distancia.
- Dada una instancia x, y sus k vecinos: $x_1, \ldots, x_k$.
- Caso discreto:
  - Función objetivo:

$$\hat{f}(x) = \arg_{v \in V} \max \sum_{i=1}^{k} w_i \delta(v, f(x_i)) \tag{32}$$

donde $w_i = \frac{1}{d(x,x_i)^2}$

  - Si algunos x_i coinciden exactamente con x, se utiliza la versión no ponderada para los que cumplan dicha condición.

Distance-Weighted K-NN

- Caso continuo:

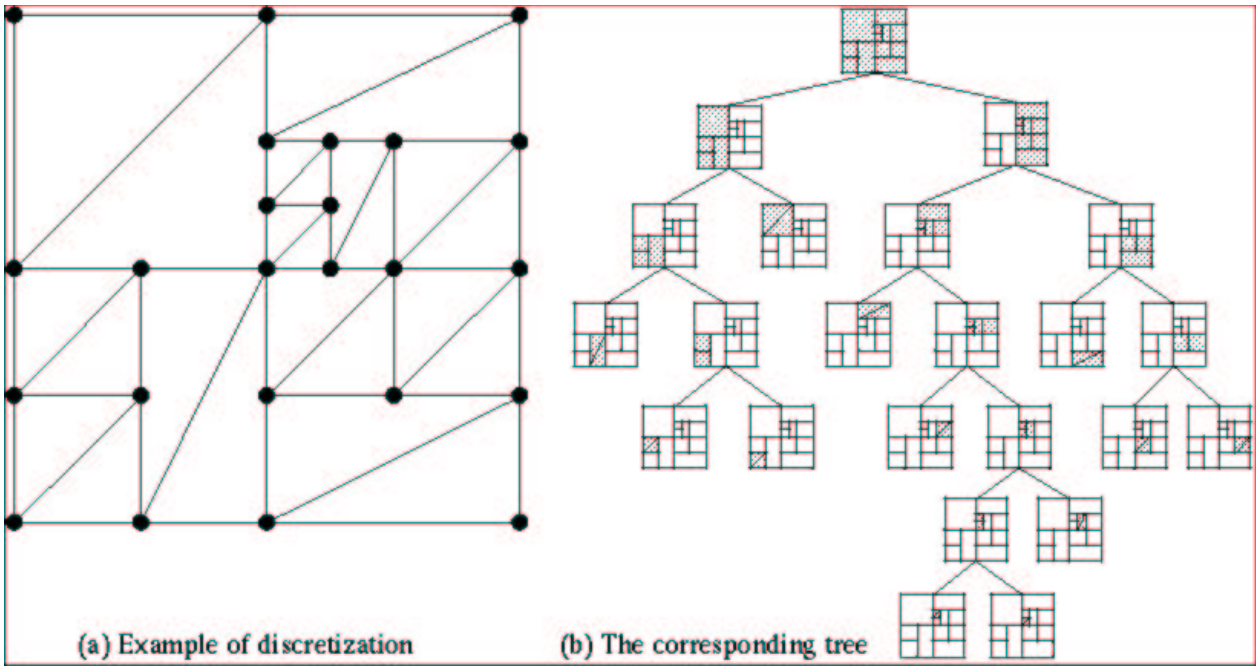
$$\hat{f}(x) = \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i} \tag{33}$$

Problemas con Tamaño Conjunto de Instancias

- Siempre hay problemas cuando el conjunto de instancias es muy grande:
  - Problemas de almacenamiento
  - Problemas de cálculo de vecinos.
- Soluciones:
  - Indexación
  - Selección de instancias
  - Reemplazo de instancias

Indexación

- Árboles KD (Locally Weighted Regression, A. W. Moore)



- Agrupación o clustering: Aprendizaje no supervisado.

Selección de instancias

- Idea: elegir un grupo reducido de instancias (prototipos) (S) que mantengan la misma información que el conjunto total (T)
- Métodos:
  - Incremental:
    - Comenzar con un conjunto S de prototipos vacío
    - Ir añadiendo instancias al conjunto S a partir de las instancias en T, siempre y cuando cumplan un determinado criterio (por ejemplo, cuando al intentar clasificarlo, se clasifica de forma incorrecta).
  - Decremental:
    - Comenzar con un conjunto $S = T$ de prototipos
    - Ir eliminando instancias al conjunto S, siempre y cuando cumplan un determinado criterio (por ejemplo, que al extraerlo del conjunto, se sigue clasificando correctamente).

Reemplazo de instancias

- Idea: calcular prototipos a partir del conjunto de entrenamiento.
- Normalmente 1-NN
- Learning Vector Quantization (LVQ):
  - Comenzar con un conjunto de prototipos $S = \{s_1, \ldots, s_M\}$.
  - Repetir
    - Elegir una nueva instancia, x
    - Obtener el prototipo más cercano de S, $s = \arg_i \text{mín}(d(x, s_i))$
    - Actualizar la posición de s_i:
      $s_i = s_i + \alpha[x - s_i]$ si s_i y x pertenecen a la misma clase
      $s_i = s_i - \alpha[x - s_i]$ si s_i y x pertenecen distinta clase
- Problema de LVQ: Definición del número de prototipos a utilizar

Selección de Características

- Idea: ¿Son todos los atributos o características relevantes para el problema de clasificación?
- Selección de características: determinar qué características son las interesantes, y eliminar las restantes.
- Resuelto en árboles de decisión por propio mecanismo de construcción de los árboles
- Soluciones:
  - Métodos estadísticos
  - Algoritmos genéticos
  - Etc.
- Fases:
  - Selección de características
  - Limpieza de datos
  - K-NN

Ponderación de Características

- Preguntas:
  - ¿Son todos los atributos o características igual de relevantes para el problema de clasificación?
  - ¿Depende esa relevancia de la zona del espacio?
- Ponderación de características: utilizar distancia euclídea ponderada:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} w_r (x_i[r] - x_j[r])^2} \quad (34)$$

- Métodos:
  - Estrategias Evolutivas
  - ...

Locally Weighted Regression

- Basado en la construcción de modelos lineales de forma local en zonas del espacio.
- Ante una nueva consulta, se obtienen los k vecinos
- Se construye un modelo de regresión lineal sobre esos vecinos:

$$\hat{f}(x) = a_0 + a_1 x[1] + \ldots + a_n x[n] \tag{35}$$

- Se devuelve esa salida
- Aplicable con cualquier otro método de aproximación (redes de neuronas).

Case Based Reasoning

- Basado en los mismos principos que KNN
- Representaciones de la información mucho más desarrolladas:
  - Imágenes
  - Documentos
  - Planes
  - Etc.

Referencias

- Tom Mitchell. Machine Learning, capítulo 8, McGraw-Hill 1997
- David Aha. Lazy Learning. Kluwer Academic Publishers, 1997
- Allen Gersho and Robert M. Gray. Vector Quantization and Signal Compression. Kluwer Academic Publishers, 1992
- Teuvo Kohonen. Self-Organizing Maps. Springer, 1995