Árboles de Clasificación

Abdelmalik Moujahid

Grupo de Inteligencia Computacional
Universidad del País Vasco UPV/EHU
Curso 2014-2015

Índice

Índice

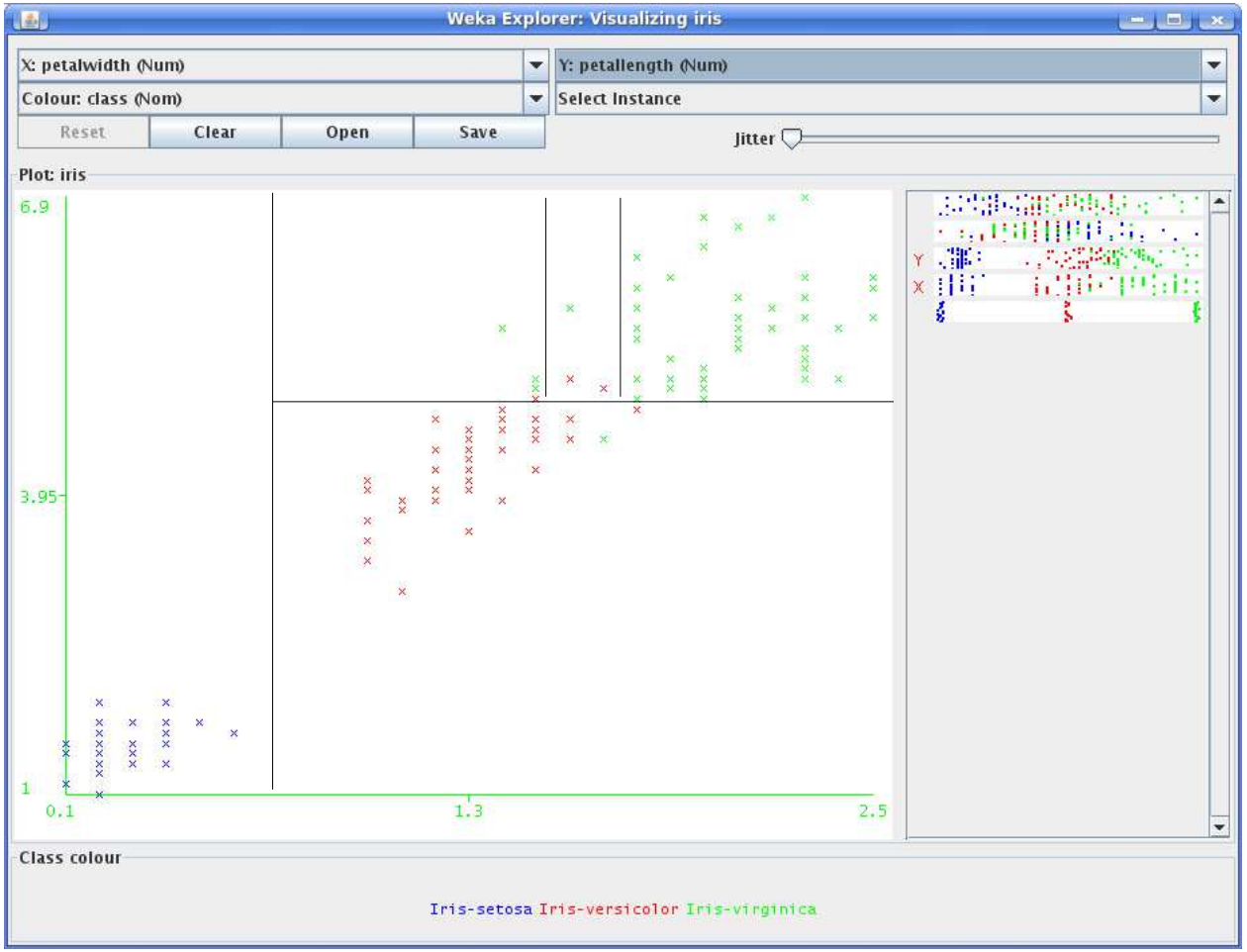
Introducción

- Un árbol de clasificación es un **conjunto de condiciones** organizadas en una estructura jerárquica. La decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen desde el nodo raíz del árbol hasta alguna de sus hojas.
- **Inducir** un árbol de decisión consiste en **dividir** recursivamente el dominio de definición de las variables predictoras en **particiones disjuntas**.
- Una partición es un conjunto de reglas **excluyentes** y **exhaustivas**.

Introducción

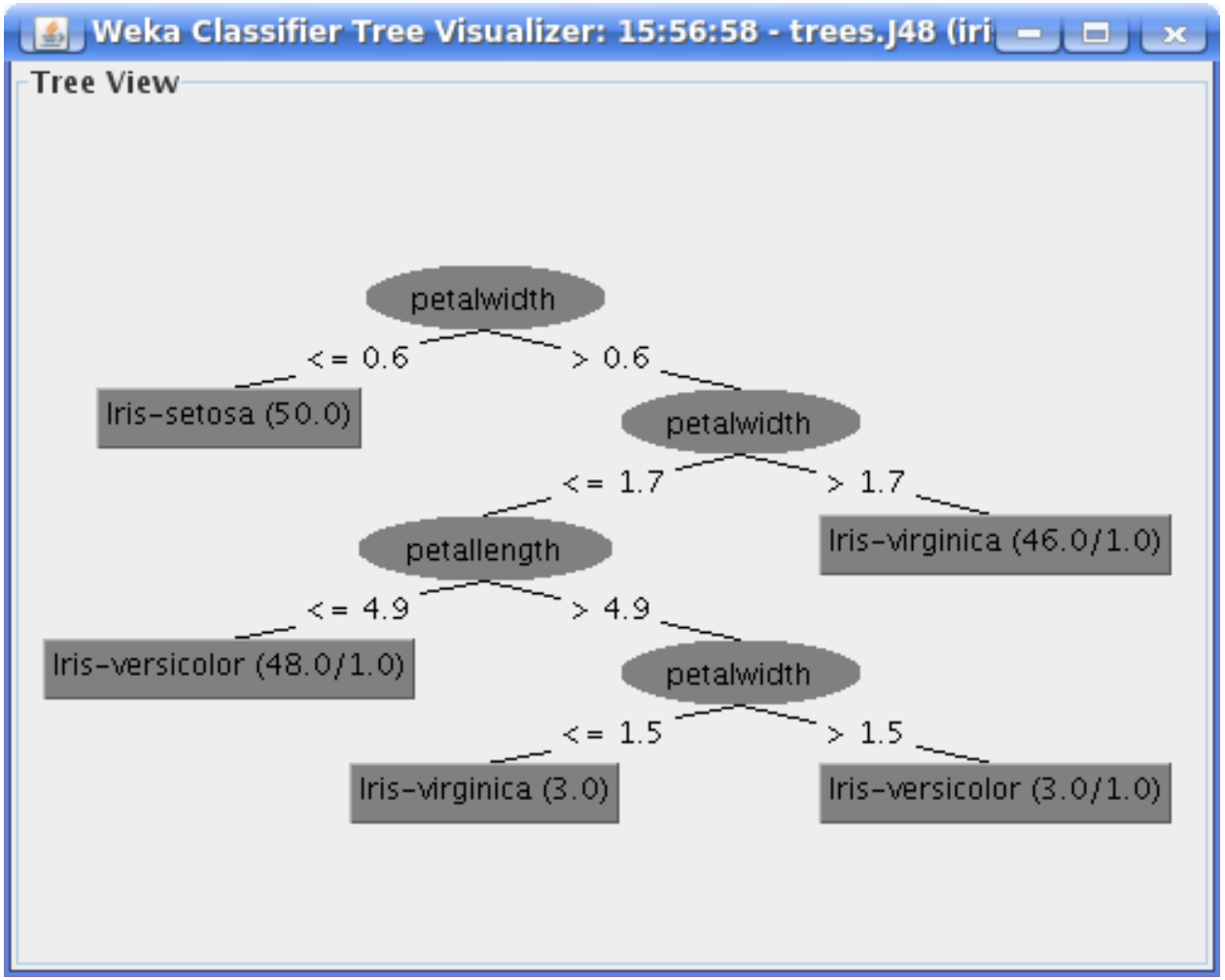
- La tarea de inducción consiste en desarrollar una regla de clasificación que puede determinar la clase de cualquier objeto a partir de los valores que toman sus atributos (training set)
- ¿Los atributos aportan la información suficiente para realizar la tarea de inducción?
- Si los atributos son **adecuados**, siempre es posible construir un árbol de decisión que clasifica correctamente cada uno de los objetos en el conjunto de testeo.
- Generalmente, se pueden encontrar muchos árboles de decisión que clasifican correctamente el conjunto de testeo.

El algoritmo básico



Gráfica de dispersión del dataset iris en el plano (petalwith,petallength), y ilustración de las distintas particiones del conjunto de datos .

El algoritmo básico



Árbol de clasificación correspondiente al ejemplo representado anterior.

Pseudocódigo del algoritmo básico

Input: D conjunto de N patrones etiquetados, cada uno de los cuales está caracterizado por n variables predictoras $X_1, \ldots, X_n$ y la variable clase C
Output: Árbol de clasificación
Begin TDIDT (Top Down Induction of Decision Trees)
if todos los patrones de D pertenecen a la misma clase c
then
resultado de la inducción es un nodo simple (nodo hoja) etiquetado como c
else
begin
1. Seleccionar la variable más informativa X_r con valores $x_r^1, \ldots, x_r^{nr}$
2. Particionar D de acorde con los n_r valores de X_r en $D_1, \ldots, D_{nr}$
3. Construir n_r subárboles $T_1, \ldots, T_{nr}$ para $D_1, \ldots, D_{nr}$
4. Unir X_r y los n_r subárboles $T_1, \ldots, T_{nr}$ con los valores $x_r^1, \ldots, x_r^{nr}$
end
endif
End TDIDT

Árboles de decisión

Los algoritmos de inducción de árboles de decisión se diferencian principalmente según:

- Los criterios de selección de las particiones
- Las estrategias de poda

Índice

El algoritmo ID3

La estrategía básica del algoritmo ID3 (Quinlan, 1986) es iterativa:

1. Se parte de un subconjunto, llamado *window*, elegido aleatoriamente desde el conjunto de entrenamiento
2. Se contruye un árbol de decisión a partir del subconjunto *window*; este árbol clasifica correctamente todos los objetos en *window*.
3. Usando el árbol inducido, se clasifican los demás objetos del conjunto de entrenamiento. Si no se producen errores, el árbol es válido para todo el conjunto de entrenamiento, y entonces finaliza el proceso. Si no, los casos mal clasificados se añaden al subconjunto *window*, y se vuelve al paso 2.

El algoritmo ID3

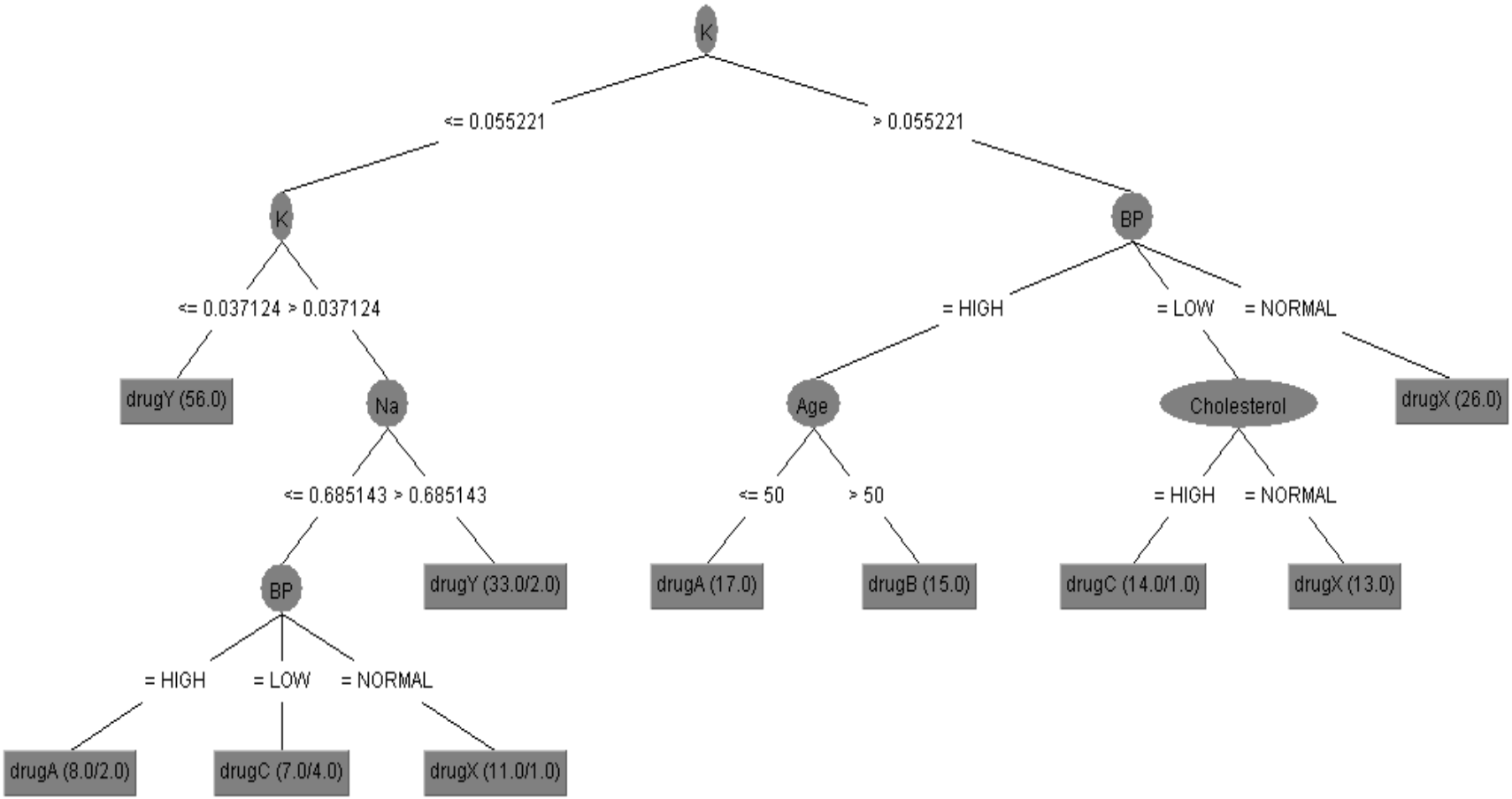
- ID3 selecciona la variable más informativa en base a la cantidad de información mutua: $I(X_i, C) = H(C) - H(C|X_i)$ (*ganancia en información*)
- Matemáticamente se demuestra que este criterio favorece la elección de variables con mayor número de valores
- Selección de variables previa (*preprunning*) basada en un test de independencia entre cada variable predictora X_i y la variable clase C

Índice

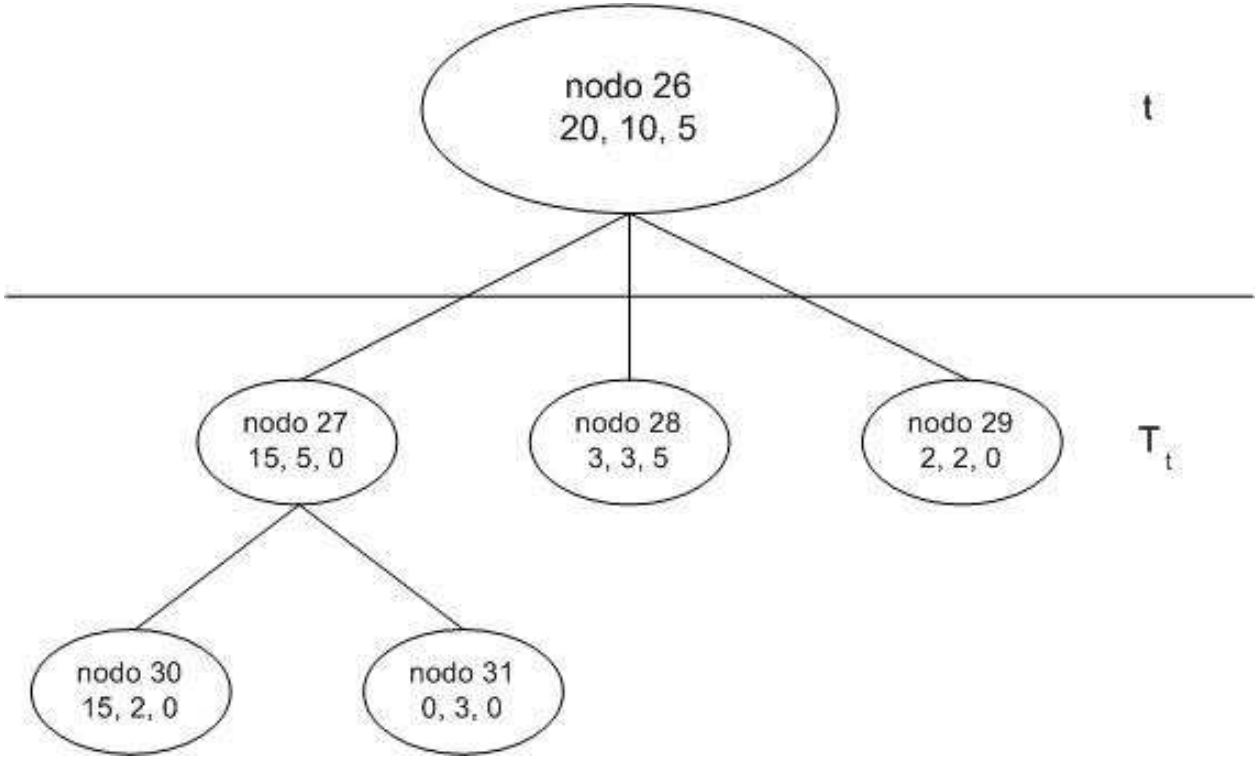
El algoritmo C4.5

- C4.5 (Quinlan, 1993) selecciona la variable más informativa en base al *ratio de ganancia*: $I(X_i, C)/H(X_i)$
- Matemáticamente se demuestra que este criterio evita que se favorezca la elección de variables con mayor número de valores
- Incorporación de una poda del árbol inducido (*postpruning*), basada en un test de hipótesis que trata de responder a la pregunta de si merece la pena expandir o no una determinada rama

Árbol de clasificación obtenido con el algoritmo C45 sobre el dataset Drug1n



El algoritmo C4.5



Ejemplo para el proceso de pos-poda del algoritmo *C*4.5

El algoritmo C4.5

Proceso de poda del árbol

- $N(t) = 35$, ejemplos en el nodo $t = 26$
- $e(t) = 10 + 5 = 15$, ejemplos mal clasificados en el nodo t
- $n'(t) = e(t) + \frac{1}{2} = 15,5$, corrección por continuidad de $e(t)$
- T_t, subárbol a expandir a partir del nodo t
- $h(T_t) = 4$, número de hojas del subárbol T_t
- $n'(T_t) = \sum_{i=1}^{h(T_t)} e(i) + \frac{h(T_t)}{2} = 2 + 0 + 6 + 2 + \frac{4}{2} = 12$, número de errores existentes en las hojas terminales del subárbol T_t
- $S(n'(T_t)) = \sqrt{\frac{n'(T_t)[N(t) - n'(T_t)]}{N(t)}} = \sqrt{\frac{12(35-12)}{35}} \simeq 2,8$, desviación de $n'(T_t)$

El nodo t se expande $\Leftrightarrow n'(T_t) + S(n'(T_t)) < n'(t) \Leftrightarrow 12 + 2,8 < 15,5$
El nodo 26 se expande considerándose los nodos 28, 29, 30 y 31