Tema 8. Árboles de Clasificación

Abdelmalik Moujahid, Iñaki Inza, Pedro Larrañaga
Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad del País Vasco–Euskal Herriko Unibertsitatea

10.1 Introducción

En este tema se va a presentar el paradigma conocido como árbol de clasificación. En el mismo, basándose en un *particionamiento recursivo* del dominio de definición de las variables predictoras, se va a poder representar el conocimiento sobre el problema por medio de una estructura de árbol. El paradigma que se presenta en este tema se conoce también bajo el nombre de *árbol de decisión*.

En la Sección 10.2 se van a presentar las ideas en las que se fundamenta el algoritmo básico de inducción del modelo a partir de datos, para tratar los algoritmos $ID3$ y $C4.5$ en las secciones 10.3 y 10.4 respectivamente.

10.2 El Algoritmo Básico

Veamos a continuación a introducir las ideas fundamentales del denominado algoritmo $TDIDT$ (*Top Down Induction of Decision Trees*) el cual puede ser contemplado como uniformizador de la mayoría de los algoritmos de inducción de árboles de clasificación a partir de un conjunto de datos conteniendo patrones etiqueados.

El pseudocódigo del algoritmo $TDIDT$ puede contemplarse en la Figura 1. La idea

---

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
  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

Figura 1: Pseudocódigo del algoritmo TDIDT

subyacente al algoritmo $TDIDT$ es que mientras que todos los patrones que se correspondan con una determinada rama del árbol de clasificación no pertenezcan a una misma clase, se seleccione la variable que de entre las no seleccionadas en esa

rama sea la más informativa o la más idónea con respecto de un criterio previamente establecido. La elección de esta variable sirve para expandir el árbol en tantas ramas como posibles valores toma dicha variable.

La Figura 2 muestra gráficamente unos patrones caracterizados por 2 variables pre-

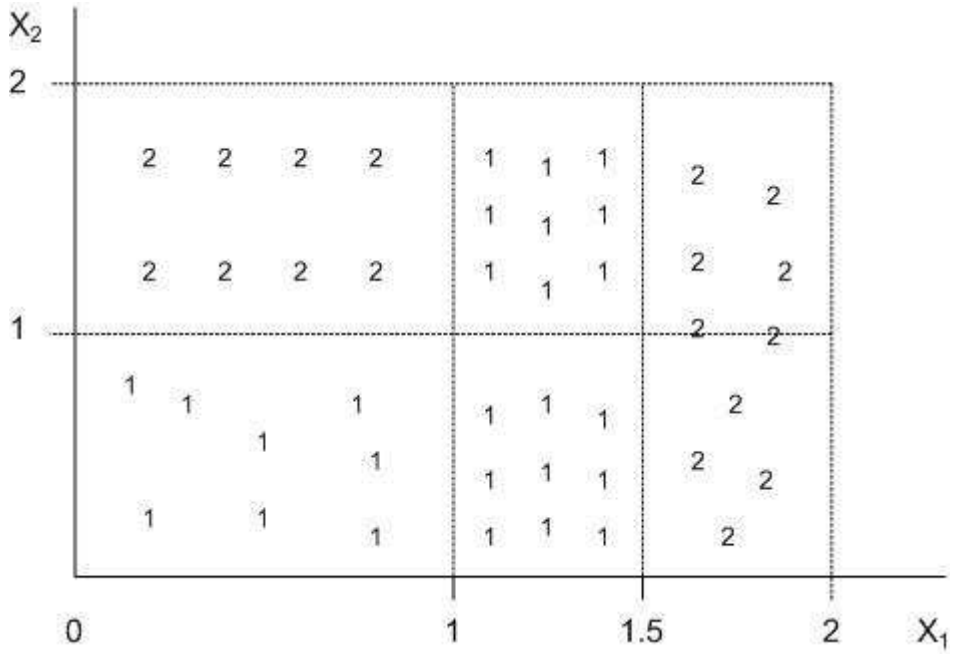

Figura 2: Representación en el plano de distintos patrones caracterizados por 2 variables predictoras X_1 y X_2 y una variable clase C con dos posibles valores

dictoras denotadas por X_1 y X_2 y una variable clase C con dos valores posibles denotados por 1 y 2. Una posible representación del conocimiento subyacente al dominio del ejemplo mostrado en la Figura 2 lo podemos ver por medio del árbol de clasificación de la Figura 3.

Tal y como puede verse en la Figura 3, las variables predictoras se van a representar en el árbol de clasificación insertadas en un círculo, mientras que las hojas del árbol se representan por medio de un rectángulo en el cual se inserta el valor de la variable clase que el árbol de clasificación asigna a aquellos casos que bajan por las correspondientes ramas del árbol de clasificación.

El árbol de clasificación de la Figura 3 tiene para todas las ramas una *profundidad* de 2, siendo este concepto de profundidad el que proporciona una idea de la *complejidad* del árbol de clasificación.

Por otra parte en este ejemplo diríamos que no existe *ruido* en el sentido de que las *hojas* son *puras* al contener tan sólo patrones de un determinado tipo respecto de la variable clase. Nótese que esta es una situación no habitual en un problema de modelización de una situación real.

Finalmente vamos a expresar el árbol de clasificación de la Figura 3 por medio de un conjunto de reglas.

Así las reglas R_1 a R_4 consideradas en su globalidad son equivalentes al árbol de clasificación anterior:

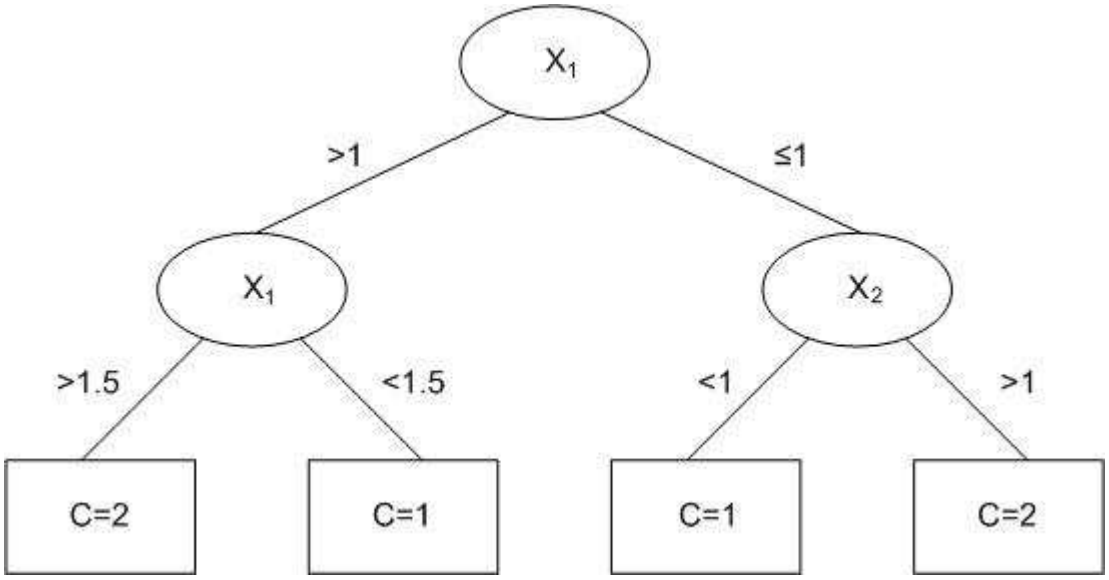

Figura 3: Árbol de clasificación correspondiente al ejemplo representado en la Figura 2

R_1 : If $X_1 > 1{,}5$ then $C = 2$
R_2 : If $1 < X_1 < 1{,}5$ then $C = 1$
R_3 : If $X_1 < 1$ y $X_2 < 1$ then $C = 1$
R_4 : If $X_1 < 1$ y $X_2 > 1$ then $C = 2$

Nótese que no todo conjunto de reglas extiende un árbol de clasificación equivalente al mismo, de ahí que el paradigma de inducción de reglas puede ser considerado como más flexible que el del árbol de clasificación.

10.3 Algoritmo $ID3$

Uno de los algoritmos de inducción de árboles de clasificación más populares es el denominado $ID3$ introducido por Quinlan (1986). En el mismo el criterio escogido para seleccionar la variable más informativa está basado en el concepto de cantidad de información mutua entre dicha variable y la variable clase. La terminología usada en este contexto para denominar a la cantidad de información mutua es la de *ganancia en información* (*information gain*).

Esto es debido a que $I(X_i, C) = H(C) - H(C|X_i)$ y lo que viene a representar dicha cantidad de información mutua entre X_i y C es la reducción en incertidumbre en C debida al conocimiento del valor de la variable X_i.

Matemáticamente se demuestra que este criterio de selección de variables utilizado por el algoritmo $ID3$ no es justo ya que favorece la elección de variables con mayor número de valores.

Además el algoritmo $ID3$ efectúa una selección de variables previa –denominada *preprunning* en este contexto– consistente en efectuar un test de independencia entre cada variable predictora X_i y la variable clase C, de tal manera que para la inducción del árbol de clasificación tan sólo se van a considerar aquellas variables predictoras para las que se rechaza el test de hipótesis de independencia.

10.4 Algoritmo $C4.5$

Quinlan (1993) propone una mejora del algoritmo $ID3$, al que denomina $C4.5$. El algoritmo $C4.5$ se basa en la utilización del criterio *ratio de ganancia (gain ratio)*, definido como $I(X_i, C)/H(X_i)$. De esta manera se consigue evitar que las variables con mayor número de posibles valores salgan beneficiadas en la selección. Además el algoritmo $C4.5$ incorpora una *poda del árbol* de clasificación una vez que éste ha sido inducido. La poda está basada en la aplicación de un test de hipótesis que trata de responder a la pregunta de si merece la pena expandir o no una determinada rama.

Consideremos el subárbol de la Figura 4. En el mismo nos planteamos la pregunta

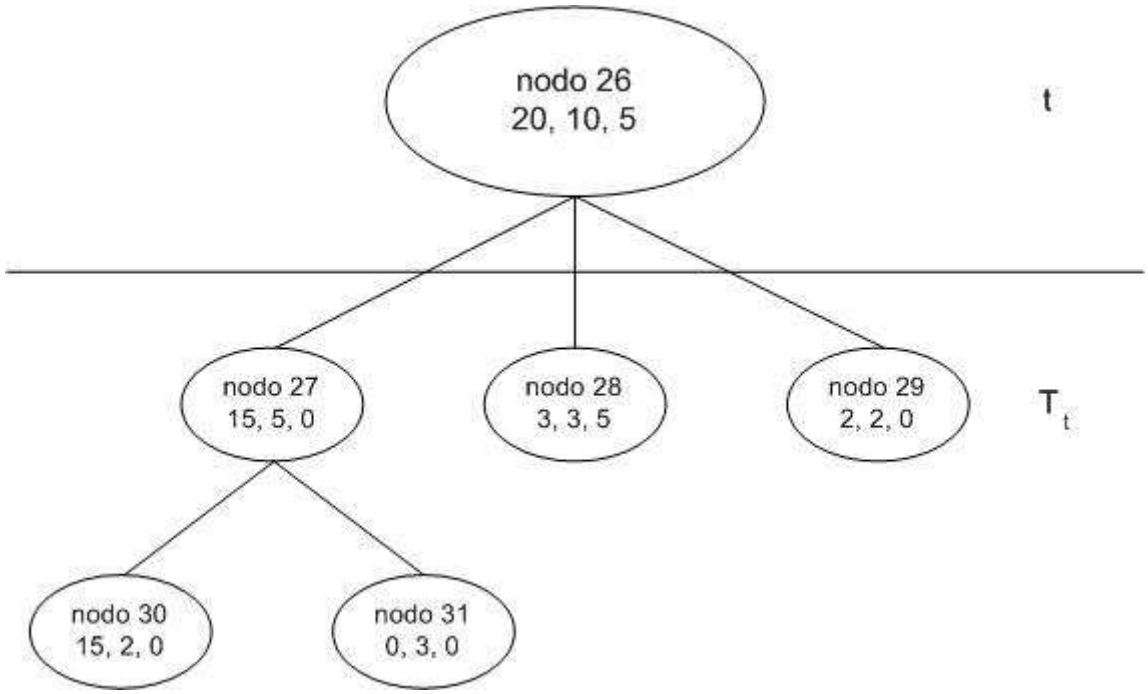

Figura 4: Ejemplo para el proceso de pos-poda del algoritmo $C4.5$

de si es conveniente o no expandir el nodo 26, y considerar como consecuencia de dicha expansión los nodos 28, 29, 30 y 31. Si tomamos la decisión de no expandir el nodo 26, cometeríamos 15 errores, mientras que si expandimos el número de errores se reduce a 10. Si bien en principio la decisión de expandir parece la más adecuada, la pregunta se plantea en el contexto de la utilidad del árbol de clasificación inducido para otras situaciones, es decir, en un contexto de generalización.

Para llevar a cabo el test, vamos a considerar los siguientes términos:

- $N(t)$ número de ejemplos en el nodo t, en el que se está testando la expansión. En el ejemplo, $t = 26$ y $N(t) = 35$.
- $e(t)$ número de ejemplos mal clasificados en el nodo t. En el ejemplo, $e(t) = 10 + 5 = 15$.
- $n'(t)$ corrección por continuidad de $e(t)$, la cual se efectúa sumando $\frac{1}{2}$ a $e(t)$. Es decir, $n'(t) = e(t) + \frac{1}{2}$.

Mientras que los tres términos anteriores $N(t)$, $e(t)$ y $n'(t)$ hacen referencia al nodo t, los siguientes están relacionados con el subárbol T_t que se va a expandir a partir del nodo t.

- $h(T_t)$ denota el número de hojas del subárbol T_t. En el ejemplo, $h(T_t) = 4$.
- $n'(T_t)$ se obtiene a partir del número de errores existentes en las hojas terminales del subárbol T_t, y se define como $n'(T_t) = \sum_{i=1}^{h(T_t)} e(i) + \frac{h(T_t)}{2} = 2+0+6+2+\frac{4}{2} = 12$.
- $S(n'(T_t))$ definido como la desviación de $n'(T_t)$ a partir de la siguiente fórmula:

$$S(n'(T_t)) = \sqrt{\frac{n'(T_t)[N(t) - n'(T_t)]}{N(t)}}$$

  En el ejemplo, $S(n'(T_t)) = \sqrt{\frac{12(35-12)}{35}} \approx 2{,}8$

La decisión acerca de expandir el nodo t, y contemplar el subárbol T_t se toma en base a la siguiente regla:

El nodo t se expande $\Leftrightarrow n'(T_t) + S(n'(T_t)) < n'(t)$.

En el ejemplo, al tener que $n'(T_t) + S(n'(T_t)) = 12 + 2{,}8 < 15{,}5 = n'(t)$, el nodo 26 se expande considerándose los nodos 28, 29, 30 y 31.

Referencias

1. Breiman L., Friedman J.H., Olshen R. A. , Stone P. J. (1984) *Classification and Regression Trees*. Wadsworth International Group.

2. Quinlan J. R. (1986) Induccion of decision trees. *Machine Learning*, 1(1), 81–106.

3. Quinlan (1993) *C4.5: Programs for Machine Learning* Morgan Kaufmann.