

Aplicaciones de Minería de Datos I

Lectura 4: Árboles de Decisión



Tabla de contenidos

Árboles de Decisión

- Descripción de la tarea de inducción

- Algoritmo básico de aprendizaje de árboles de decisión:ID3

- Espacio de búsqueda y bias inductivo

Sobre-ajuste y bajo-ajuste

Referencias



Árboles de Decisión (1/5)

¿Qué es un algoritmo de árbol de decisión?

Es un algoritmo que trabaja a base de diagramas, que determinará el curso de una acción o de mostrar una probabilidad estadística. *Cada rama del árbol representa una posible decisión, salida o reacción.*



Árboles de Decisión (2/5)

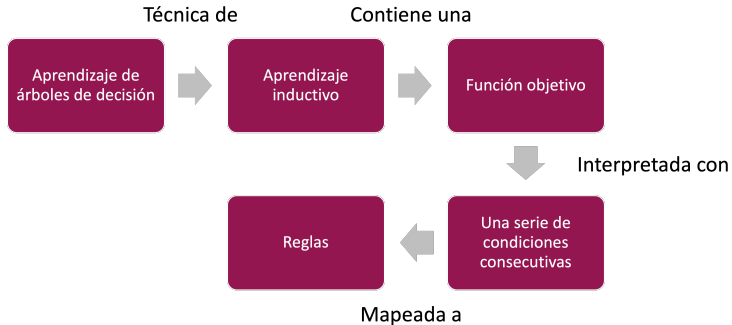


Figura 1: Estructura del árbol de decisión.

Árboles de Decisión (3/5)

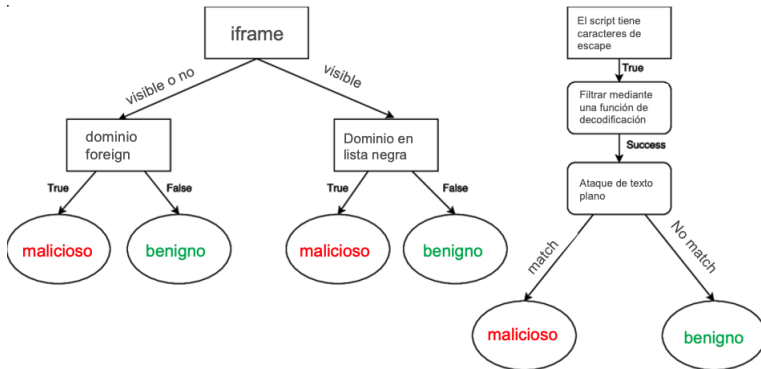


Figura 2: Ejemplo de árbol de decisión para el aprendizaje del concepto malware en sitios web. Se pretende clasificar instancias con atributos relativos a diversos factores de transacciones con el fin de decidir si realmente corresponde a software malicioso o no.

Árboles de Decisión (4/5)

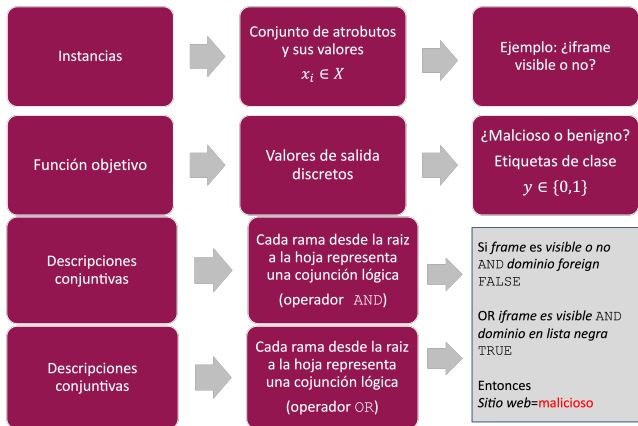


Figura 3: ¿Cuándo se considera adecuado el aprendizaje mediante árboles de decisión?

Árboles de Decisión (5/5)

Ventajas	Desventajas
Fáciles de comprender y traducir a reglas	Los atributos de salida deben ser categorías
Trabajan con conjuntos de datos tanto numéricos como nominales	No se permiten múltiples atributos de salida
Trabajan con datos multidimensionales	Los árboles construidos a partir de datos numéricos pueden resultar muy complejos
No requieren conocimiento en un dominio dado ni establecer parámetros	

Cuadro 1: Ventajas y desventajas de los árboles de disección.



Descripción de la tarea de inducción (1/7)

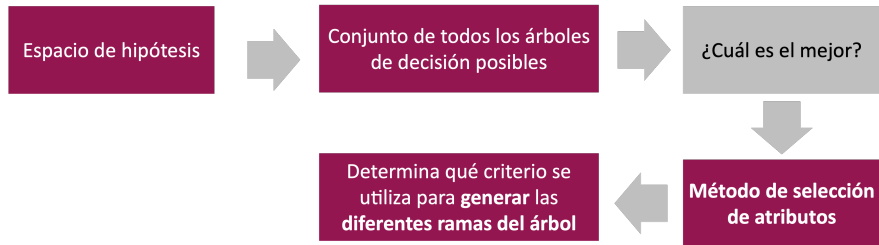


Figura 4: Etapas de la de la tarea de inducción.

Descripción de la tarea de inducción (2/7)

¿Qué es un método de selección de atributos?

Es un método que determina qué criterio se utiliza para generar las diferentes ramas del árbol, que van determinando la clasificación en las diferentes clases.



Descripción de la tarea de inducción (3/7)

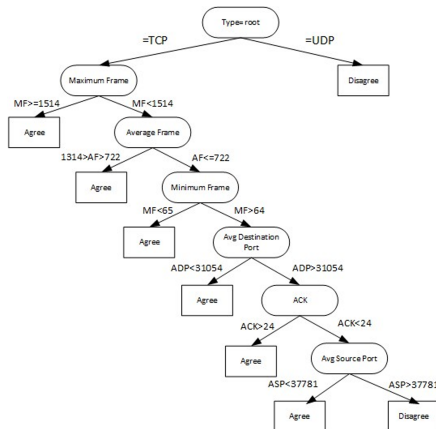
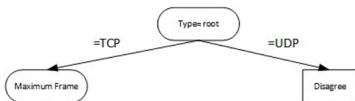


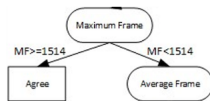
Figura 5: Árbol de decisión diseñado para detectar muestras de malware en dispositivos móviles, mediante análisis de paquetes de red [Afi et al., 2016].

Descripción de la tarea de inducción (4/7)

El criterio **dependerá del tipo de dato que sea el atributo**. Por ejemplo, si el tipo de dato del atributo es discreto, se crea una rama para cada valor conocido del atributo.



División categórica



División numérica con puntos de separación

Figura 6: El método de selección de los atributos. Especifica una heurística para seleccionar el atributo que mejor discrimina los ejemplos para una clase.

Descripción de la tarea de inducción (5/7)

```
PROCEDIMIENTO Inducir_Arbol (Ejemplos E, Lista_Atributos,
Método_Selección_Atributos)
COMIENZO
P1 Crear un nodo N;
P2 SI todos los elementos de E pertenecen a la misma clase, C
  ENTONCES retornar N como nodo hoja etiquetado con la clase C,
P3 SINO SI la lista de atributos (Lista_Atributos) está vacía
  ENTONCES retornar N como nodo hoja etiquetado con la clase más numerosa
  en los ejemplos
P4 SINO aplicar Método_Selección_Atributos(E, Lista_Atributos) para
  seleccionar el atributo A que mejor particiona E
P5 Borrar Atributo A de la lista de Atributos Lista_Atributos
P6 Etiquetar N con el atributo seleccionado
P7 PARA CADA valor V de A
    Siendo Ev el subconjunto de elementos en E con valor
    V en el atributo A.
P8 SI Ev está vacío
    ENTONCES unir al nodo N una hoja etiquetada con la
    clase mayoritaria en E.
P9 SINO unir al nodo N el nodo retornado de
    Inducir_Arbol (Ev, Lista_Atributos,
    Método_Selección_Atributos)
FIN PARA CADA
FIN SI-SINO
FIN
```

Figura 7: Algoritmo básico de construcción de árboles de decisión.



Descripción de la tarea de inducción (6/7)

Condiciones del algoritmo:

- ▶ Todos los ejemplos de E_v pertenecen a la misma clase, con lo cual el **nodo se convierte en un nodo hoja**
- ▶ No hay más atributos para dividir ejemplos
 - ▶ Se puede convertir el nodo en una hoja, y etiquetarlo con la clase mayoritaria en los ejemplos de E_v (esto se denomina **votación mayoritaria**)
- ▶ Si no hay ejemplos para una rama, la partición E_v está vacía
 - ▶ Se crea un nodo hoja con la clase mayoritaria en E



Descripción de la tarea de inducción (7/7)

¿Qué es **divide-y-vencerás** en árboles de decisión?

El algoritmo básico aquí descrito es del tipo **divide-y-vencerás**, construido sin retroceder en ningún caso para volver a reconsiderar una decisión tomada en un paso previo.

Esto último relativo a siempre avanzar hacia adelante es denominado método codicioso (greedy en inglés)



Algoritmo básico de aprendizaje de árboles de decisión: ID3 (1/5)

¿Qué es el algoritmo ID3?

Es un algoritmo que utiliza la **ganancia de información** para seleccionar en cada paso según se va generando el árbol aquel atributo que mejor distribuye los ejemplos de acuerdo a su clasificación objetivo

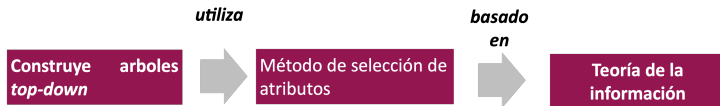


Figura 8: Estructura del algoritmo ID3.



Algoritmo básico de aprendizaje de árboles de decisión: ID3 (2/5)

¿Cómo se mide la mejor distribución de ejemplos o se selecciona aquel atributo cuyo conocimiento aporta mayor información?

*ID3 utiliza conceptos que forman parte de la teoría de la información: utiliza la **ganancia de información** que a su vez mide la reducción esperada de **entropía**.*



Algoritmo básico de aprendizaje de árboles de decisión: ID3 (3/5)

¿Por qué es importante la entropía?

La entropía caracteriza la **heterogeneidad** de un conjunto de ejemplos. Cuando una clase C puede tomar n valores, la entropía del conjunto de ejemplos E respecto a la clase C se define como:

$$\mathbf{Entropia}(E) = \sum_{i=1}^n -p_i \log_2 p_i \quad (1)$$

,siendo p_i la proporción de ejemplos de E que pertenecen a la clase i .



Algoritmo básico de aprendizaje de árboles de decisión: ID3 (4/5)

*Dado que la entropía mide la **heterogeneidad** de un conjunto de ejemplos, la **ganancia de información** utiliza la entropía para **medir la efectividad de un atributo** para clasificar ejemplos. Específicamente mide la reducción de entropía cuando se distribuyen los ejemplos de acuerdo a un atributo concreto.*



Algoritmo básico de aprendizaje de árboles de decisión: ID3 (5/5)

Definición de Ganancia de la Información

Siendo un atributo A con V_a posibles valores y un conjunto de ejemplos E , la fórmula para calcular la ganancia de información viene dada por la siguiente expresión:

$$\mathbf{Ganancia}(A, B) = \mathbf{Entropia}(E) - \sum_{v \in V_a} \frac{|E_v|}{E} \mathbf{Entropia}(E_v) \quad (2)$$

, donde E_v es el subconjunto de ejemplos para los que el atributo A toma el valor v dentro de los posibles valores de v (especificados en V_a)



Espacio de búsqueda y bias inductivo (1/5)

En el problema de construcción de árboles de decisión, el espacio de hipótesis o posibles soluciones es el conjunto de todos los posibles árboles de decisión.



Espacio de búsqueda y bias inductivo (2/5)

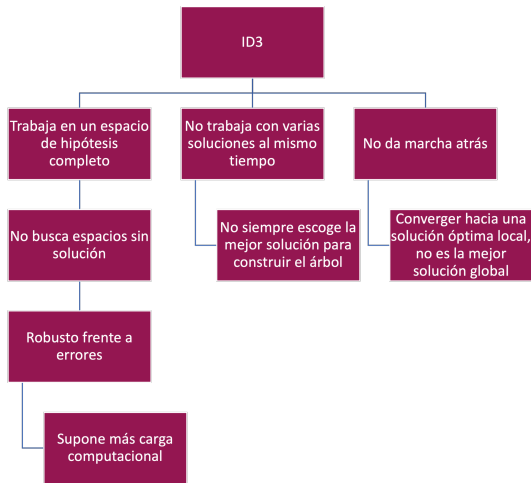


Figura 9: Consideraciones del algoritmo ID3.

Espacio de búsqueda y bias inductivo (3/5)



Figura 10: Consideraciones del algoritmo ID3.

Espacio de búsqueda y bias inductivo (4/5)

¿En qué se basa ID3 para generalizar el árbol de decisión, esto es, considerar que el árbol clasificará correctamente instancias no utilizadas en la etapa de aprendizaje?

ID3 realiza una búsqueda en escalada, **guiada por la medida de ganancia de información**, desde árboles más sencillos a árboles más complejos, buscando aquel que clasifica correctamente los datos de entrenamiento.



Espacio de búsqueda y bias inductivo (5/5)

¿En qué se basa ID3 para generalizar el árbol de decisión, esto es, considerar que el árbol clasificará correctamente instancias no utilizadas en la etapa de aprendizaje?

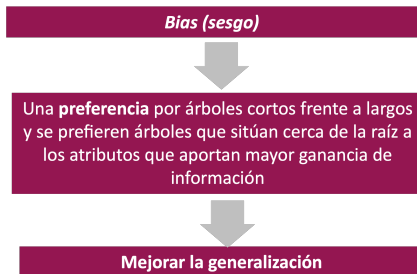


Figura 11:

Sobre-ajuste y bajo-ajuste (1/6)

¿Qué es el sobre-ajuste?

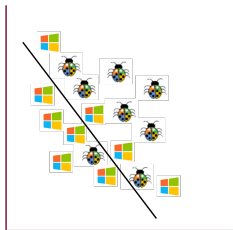
El sobre-ajuste es un comportamiento indeseable del aprendizaje automático que se produce cuando el modelo ofrece predicciones precisas para los datos de entrenamiento, pero no para los datos nuevos (de prueba).

¿Qué es el bajo-ajuste?

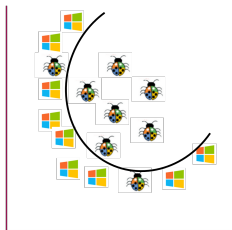
El bajo-ajuste es un escenario en el que un modelo es incapaz de capturar con precisión la relación entre las variables de entrada y de salida, generando una alta tasa de error tanto en el conjunto de entrenamiento como en los datos no vistos (de prueba).



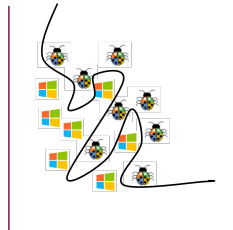
Sobre-ajuste y bajo-ajuste (2/6)



Bajo-ajuste



Ajuste deseado



Sobre-ajuste

Figura 12: Ejemplo de sobre ajuste.

Sobre-ajuste y bajo-ajuste (3/6)

Pospoda	Prepoda
<p>Podar el árbol una vez generado.</p> <ul style="list-style-type: none">• Se pueden llegar a tener en cuenta combinaciones de atributos antes de realizar la poda.• Existen ocasiones en que dos o más atributos combinados aportan bastante información en la clasificación mientras que los atributos por sí solos no.	<p>Limitar el crecimiento del árbol</p> <ul style="list-style-type: none">• Tiene la ventaja de que ahora los costes de procesamiento debidos a generar nodos y ramas que posteriormente serían podados

Figura 13: Estrategias para evitar el sobre-ajuste.

Sobre-ajuste y bajo-ajuste (4/6)

Técnicas para estimar el sobreajuste

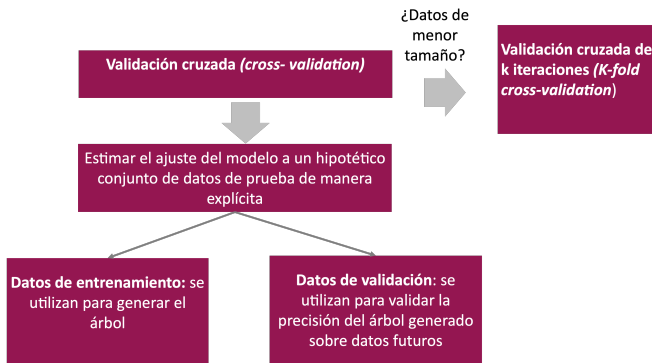


Figura 14: Técnicas para estimar el sobre-ajuste.

Sobre-ajuste y bajo-ajuste (5/6)

¿Qué es la validación cruzada (*cross-validation*)?

Es un método estadístico para evaluar y comparar algoritmos de aprendizaje dividiendo los datos en dos segmentos: uno utilizado para aprender o entrenar un modelo y otro utilizado para validar el modelo.

¿Qué es la validación cruzada de k -iteraciones (k -fold *cross-validation*)?

Consiste en dividir el conjunto de datos en un número k iteraciones y se utiliza para evaluar la capacidad del modelo cuando se le proporcionan nuevos datos. k se refiere al número de grupos en que se divide la muestra de datos.



Sobre-ajuste y bajo-ajuste (6/6)

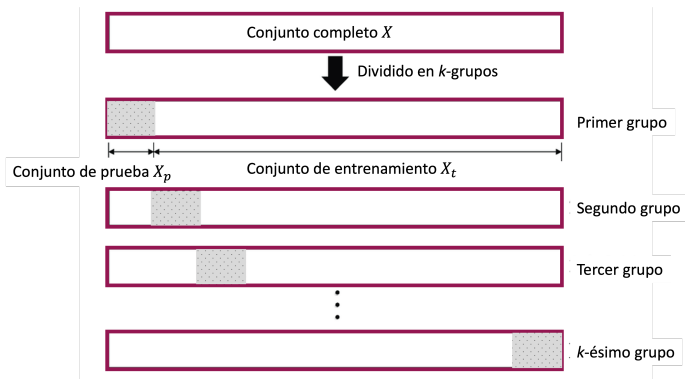





Figura 15: Ejemplo de la técnica de *k-fold cross-validation*.

Referencias

-  Lin, C. T., Wang, N. J., Xiao, H., & Eckert, C. (2015). Feature Selection and Extraction for Malware Classification. J. Inf. Sci. Eng., 31(3), 965-992.
-  Layton, R. (2015). Learning data mining with python. Packt Publishing Ltd.
-  Afifi, F., Anuar, N. B., Shamshirband, S., & Choo, K. K. R. (2016). DyHAP: Dynamic hybrid ANFIS-PSO approach for predicting mobile malware. PloS one, 11(9), e0162627.

