Algoritmos de árboles de decisión - Dunham

Pasos básicos:

1. Inducción del árbol de decisión: Construir el árbol
2. Aplicar el árbol a cada tupla de la base de datos para determinar su clase

Propiedades

- Cada nodo interno se etiquetado con un atributo Ai
- Cada arco es etiquetado con un predicado que puede ser aplicado al atributo asociado con el padre.
- Cada nodo hoja es etiquetado con una clase Cj.

Ventajas

- ✓ Fáciles de usar y eficientes
- ✓ Las reglas generadas son fáciles de interpretar y comprender
- ✓ Soportan grandes bases de datos debido a que el tamaño del árbol es independiente del tamaño de la base de datos
- ✓ Cada tupla de la base de datos debe ser filtrada por el árbol
- ✓ El tiempo de procesamiento es proporcional al peso del árbol
- ✓ Los árboles pueden ser construidos por datos con muchos atributos

Desventajas

- × No manejan fácilmente datos continuos
- × Los dominios de los atributos deben ser divididos entre categorías para ser manejados
- × La aproximación usada es que el espacio del dominio sea dividido en regiones rectangulares
- × No todos los problemas de clasificación son de este tipo
- × El problema de clasificación de préstamo simple no puede ser manejado
- × Manejar datos faltantes es difícil porque ramas correctas en el árbol pueden no tomarse
- × Como un árbol de decisión se construye a partir de los datos de entrenamiento, puede ocurrir un sobre entrenamiento (esto se puede solucionar con la poda de árboles).
- × Las correlaciones entre los atributos en la base de datos son ignoradas.

Atributos de división: Atributos en el esquema de la base de datos que se utilizarán para etiquetar nodos en el árbol y alrededor del cual se realizarán las divisiones

Predicados de división: Predicados por los cuales se etiquetan los arcos en el árbol

- ❖ **Elegir atributos de división:** Análisis de los datos en el conjunto de entrenamiento y aporte de los especialistas de datos
- ❖ **Orden de los atributos de división**
- ❖ **Número de divisiones**
- ❖ **Estructura del árbol:** Un árbol equilibrado con la menor cantidad de niveles es deseable. Algunos algoritmos únicamente generan árboles binarios

- **Criterios de detención:** La creación del árbol se detiene cuando los datos de entrenamiento se clasifican perfectamente. Puede ser conveniente detenerse antes para evitar la creación de árboles más grandes. Es concebible que se creen más niveles de los necesarios en un árbol si se sabe que hay distribuciones de datos no representadas en los de entrenamiento.
- **Datos de entrenamiento:**
  Conjunto pequeño -> Árbol insuficientemente específico para trabajar correctamente con datos más generales.
  Conjunto grande -> Árbol sobre entrenado.
- **Poda:** Eliminar comparaciones redundantes o eliminar árboles secundarios para lograr un mejor rendimiento.

Árboles binarios: Tienden a ser más profundos. Los resultados pueden ser peores: se necesitan más comparaciones

Complejidad: O(n log n)

ID3

Hacer preguntas cuyas respuestas brinden la mayor cantidad de información. Elegir atributos de división con la mayor ganancia de información primero. La cantidad de información asociada con un valor de atributo está relacionada con la probabilidad de ocurrencia.
Entropía: Medir la cantidad de incertidumbre o sorpresa o aleatoriedad en un conjunto de datos.

$$H(p_1, p_2, \ldots, p_s) = \sum_{i=1}^{s} (p_i \log(1/p_i))$$

Todos los datos pertenecen a la misma clase -> Entropía es cero.
El objetivo es dividir iterativamente el conjunto de datos dado en subconjuntos donde todos los elementos de cada subconjunto final pertenecen a la misma clase.

ID3 elige el atributo de división con la mayor ganancia de información.
Ganancia: Diferencia entre la cantidad de información necesaria para hacer una clasificación correcta antes de la división frente a la cantidad de información necesaria después de la división.
Las diferencias entre las **entropías del conjunto de datos original** y la suma ponderada de **las entropías de cada uno de los conjuntos** de datos subdivididos.

$$\text{Gain}(D, S) = H(D) - \sum_{i=1}^{s} P(D_i)H(D_i)$$

C4.5

- **Datos faltantes:** Son ignorados al construir el árbol.
  Radio de ganancia – Gain ratio: se calcula mirando solo los otros registros que tienen un valor para ese atributo. Para clasificar un registro con un valor de atributo faltante, el valor para ese elemento puede predecirse en función de lo que se sabe sobre los valores de atributo para los otros registros.
- **Datos continuos:** Dividir los datos en rangos basados en los valores de atributo para ese elemento que se encuentran en la muestra de entrenamiento.

- **Poda:**
  - Un *subárbol se reemplaza* por un nodo hoja si este reemplazo resulta en una tasa de error cercana a la del árbol original
  - *Subárbol de elevación*, reemplaza un subárbol por su árbol más utilizado. Un subárbol se eleva desde su ubicación actual a un nodo más arriba en el árbol.
- **Reglas:** Clasificación vía árboles de decisión o reglas generadas de estos
- **División:** Favorece los atributos con muchas divisiones -> sobre entrenamiento. Un atributo que tenga un valor único para cada tupla en el conjunto de entrenamiento sería el mejor: una tupla (una clase) para cada división.
  Gain Ratio

$$\text{GainRatio}(D, S) = \frac{\text{Gain}(D, S)}{H\left(\frac{|D_1|}{|D|}, \ldots, \frac{|D_s|}{|D|}\right)}$$

  Para fines de división, C4.5 utiliza el GainRatio más grande que garantiza una ganancia de información mayor que el promedio. Esto es para compensar el hecho de que el valor GainRatio está sesgado hacia divisiones donde el tamaño de un subconjunto es cercano al del inicio

C5.0

Boosting:

- Combinar diferentes clasificadores.
- Aumenta el tiempo que lleva ejecutar un clasificador específico, pero mejora la precisión.
- La tasa de error es menos de la mitad que en C4.5.
- No siempre ayuda cuando el conjunto de datos tiene mucho ruido.
- Crea múltiples conjuntos de entrenamiento a partir de uno
- A cada elemento del conjunto se le asigna un peso (importancia para el clasificador)
- A cada clasificador se le asigna un voto, se realiza la votación y la tupla objetivo se asigna a la clase con el mayor número de votos.

CART: Árboles de clasificación y regresión

- Genera un árbol de decisión binario.
- La **entropía** se usa como medida para elegir el mejor atributo y criterio de división.
- Donde se crea un elemento secundario para cada subcategoría, solo se crean dos elementos secundarios.
- La división se realiza alrededor de lo que se determina que es el **mejor punto de división:**

$$\Phi(s/t) = 2P_L P_R \sum_{j=1}^{m} |P(C_j \mid t_L) - P(C_j \mid t_R)|$$

- Obliga a utilizar un orden de los atributos.
- Maneja los datos faltantes simplemente ignorando ese registro al calcular la bondad de una división en ese atributo.
- El árbol deja de crecer cuando ninguna división mejorará el rendimiento.

Técnicas de árboles de decisión escalables

SPRINT (Inducción escalable paralelizable de árboles de decisión) aborda el problema de escalabilidad al garantizar que la técnica CART se pueda aplicar independientemente de la disponibilidad de memoria principal. Se utiliza un **índice de Gini** para encontrar la mejor división.

$$\text{gini}(D) = 1 - \sum p_j^2$$

Bondad de una división:

$$\text{gini}_{\text{split}}(D) = \frac{n_1}{n}(\text{gini}(D_1)) + \frac{n_2}{n}(\text{gini}(D_2))$$

- Se elige la división con el mejor valor de Gini.
- No necesita ordenar los datos por valor de bondad en cada nodo durante el proceso de inducción DT.
- Con datos continuos, el punto de división se elige como el punto medio de cada par de valores consecutivos del conjunto de entrenamiento.

RainForest: elegir un atributo dividido sin necesidad de un conjunto de entrenamiento al mantener metadatos agregados con respecto a los atributos de la base de datos.

Para cada nodo de un árbol de decisión, se utiliza una **tabla llamada grupo de etiquetas de clase de valor de atributo (AVC)**

- Resume para un atributo el recuento de entradas por clase o agrupación de valor de atributo y
- Resume la información necesaria para determinar los atributos de división.

El tamaño de la tabla es proporcional al producto del número de clases, valores de atributos únicos y posibles atributos de división.

Durante la fase de construcción del árbol, se escanean los datos de entrenamiento, se construye el AVC y se elige el mejor atributo de división.

El algoritmo continúa dividiendo los datos de entrenamiento y construyendo el AVC para el siguiente nodo.