Árboles de Decisión:
Algoritmos computacionales que utilizan gráficos en forma de flujo.

Son ampliamente utilizados en Inteligencia Artificial para realizar **modelos predictivos**, aunque en muchos casos pueden brindar aspectos descriptivos de relevancia.

Desde una base de datos, se construyen enlaces lógicos que sirven para representar y **categorizar** condiciones que surgen de manera **sucesiva** para la resolución de un problema.

Estos algoritmos crean también reglas, por lo cual se lo suele considerar “primos” de los Algoritmos de Reglas de Asociación.

En algunos casos, se los puede utilizar también para la **selección de variables**.

La idea principal es crear un modelo que prediga el valor de la variable target (label) basado en las variables de entrada.

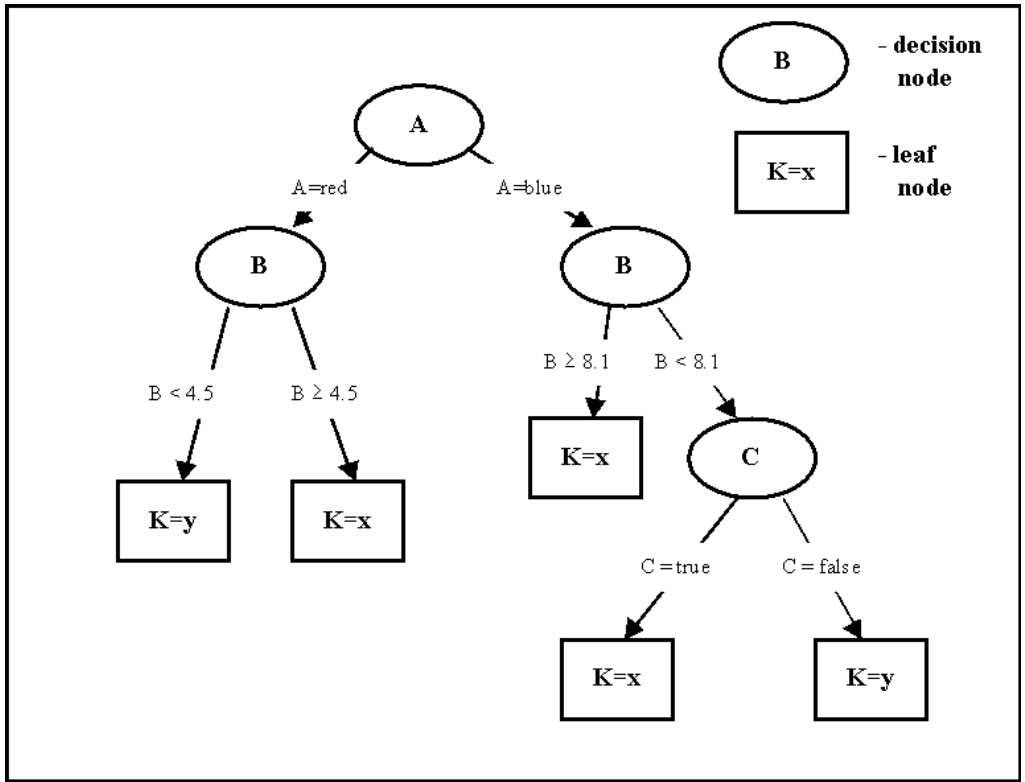
Cada nodo corresponde a una de las variables de entrada. Hay enlaces hijos por cada uno de los posibles valores de las variables de entrada.

Cada rama representa el valor de la variable LABEL de acuerdo al valor de las variables de entrada representadas por el camino desde el tope hasta el final.

La idea principal es crear un modelo que prediga el valor de la variable target (label) basado en las variables de entrada.

Cada nodo corresponde a una de las variables de entrada. Hay enlaces hijos por cada uno de los posibles valores de las variables de entrada.

Cada rama representa el valor de la variable LABEL de acuerdo al valor de las variables de entrada representadas por el camino desde el tope hasta el final.



Tipo de Árboles de Decisión

Clasificación: Cuando se está tratando de predecir una clase. (Variable label)

Regresión: Cuando se está tratando de predecir un número real.

Suele leerse muy seguido estas técnicas como:

CART: **Classification And Regression Tree**

Algoritmos

ID3
C4.5
CART
CHAID
MARS

ID3

1. Calcula la Entropía de cada atributo del dataset
2. Divide el set de datos usando el atributo cuya entropía es mínima (Mayor Ganancia de Información)
3. Realiza un nodo considerando ese atributo.
4. Realiza una nueva recursión usando los atributos aún no seleccionados.
5. Se recomienda utilizarlo con atributos discretos. Dividir atributos continuos suele implicar gran consumo de tiempo y recursos computacionales.
6. ID3 luego evolucionó en el algoritmo C4.5 (libre) que es una implementación del famoso C5 que utilizan plataformas propietarias.
7. En R no hay una buena implementación de ID3. Y en Rapidminer solo soporta atributos nominales. (no soporta atributos numéricos)

Para evitar overfing se prefieren árboles con menos ramas, aunque pierdan en accuracy.

C4.5

1. Son utilizados para clasificación
2. Usa el mismo concepto que ID3 en cuanto a Entropía e Información
3. En Weka se lo conoce como J48 y es un desarrollo muy famoso en Java
4. Con respecto a ID3 mejora:

    1. Maneja mejor los datos continuos
    2. Maneja atributos con datos perdidos
    3. Maneja correctamente el “pruning” luego de que el árbol fue creado

Más datos:
http://en.wikipedia.org/wiki/C4.5_algorithm

En Rapidminer el operador: “DecisionTree” implementa un algoritmo similar a C4.5 y puede manejar tanto datos numéricos como nominales.

En R existe el paquete “C50” que hace una implementación directa del algoritmo C5
También hay una implementación en el paquete “RWeka” del algoritmo J48.

Técnicas

Bagging
Random Forest
Boosted Trees
Rotation Forest

Estas técnica generan varios árboles de manera simultánea.

Random Forest:

El algoritmo Random Forest hace crecer varios árboles de decisión simples.

Para clasificar un nuevo objeto, toma el INPUT de siempre, y se lo pasa a cada uno de un grupo de árboles dentro de lo que denominaremos “bosque”.

Cada árbol va a arrojar una clasificación.

El “bosque” termina eligiendo la clasificación que mejores resultados haya obtenido.

La clave de cómo mejora la clasificación, es la manera aleatoria de cómo genera las muestras individuales.

Se pueden utilizar tanto para Regresión como para Clasificación

Random Forest:

Ventaja de los Random Forest:

- Actualmente es el que la comunidad acepta como el mejor clasificador.
- Funciona de manera eficiente en grandes bases de datos.
- Puede manejar gran cantidad de variables de entrada, sin tener que borrarlas.
- Aún no borrándolas, puede determinar cuales son las variables más importantes en el dataset.
- Maneja bien los datos faltantes. (los estima)

Desventajas:

- Suele producir overfitting en algunas oportunidades.
- Si el set de datos contiene variables categóricas con diferentes números de niveles, Random Forest va a tener una tendencia hacia aquellas variables con mayor número de niveles.
- Se debe tener cuidado con variables que estén correlacionadas.

Palabras clave:

Ensemble Learning

Utilización de múltiples modelos para obtener mejores predicciones.

Boosting

Basado en la pregunta: "¿Puede un conjunto de algoritmos débiles crear un único y fuerte predictor? Predictor Débil: "Apenas mejor que una clasificación random"
En R existe un paquete muy famoso para realizar boosting, llamado: "ada".
(adaptative Boosting)

In boosting, successive trees give extra weight to points incorrectly predicted by earlier predictors

Bagging

(Bootstrap Aggregation)
También se basan en el concepto de mejorar el rendimiento de los modelos predictivos.
Se utilizan para Regresión y Clasificación, generalmente en Árboles de Decisión.
Están muy relacionados con los conceptos de Model Averging.
Paquetes en R: "ipred" y "adabag"
In bagging, successive trees do not depend on earlier trees

Random Forest:

Resumiendo:

Los conceptos anteriormente vistos, tiene como “Filosofía”:

Tomando múltiples muestras aleatorias del set de datos se construyen diversos modelos predictivos individuales, aunque no sean fuertes, cada uno con su predicción.

Las predicciones son secuenciales y basadas en el error cometido en el anterior modelo.

Estas predicciones individuales luego son procesadas, generalmente utilizando algún mecanismo de promedios, esperando conseguir un resultado final mucho mejor que cada una de las predicciones individuales.

Uno suele utilizar estas técnicas, cuando las predicciones con otros algoritmos suelen ser inestables. (varían mucho una de otra tomando diferentes muestras del set de datos.)