

Árbol de Decisión en el Aprendizaje

Resumen:

El aprendizaje de árboles de decisión es uno de los métodos más utilizados y prácticos para la inferencia inductiva. Este capítulo describe una familia de algoritmos de aprendizaje de árboles de decisión, incluyendo algoritmos populares como ID3, ASSISTANT y C4.5. Estos métodos buscan un espacio de hipótesis completamente expresivo y prefieren árboles pequeños sobre árboles grandes.

3.1 Introducción

El aprendizaje de árboles de decisión es un método para aproximar funciones de valor discreto. Los árboles aprendidos pueden ser representados como conjuntos de reglas if-then para mejorar la legibilidad humana. Estos métodos se han aplicado con éxito en tareas como el diagnóstico médico y la evaluación del riesgo crediticio.

3.2 Representación del Árbol de Decisión

Los árboles de decisión clasifican instancias ordenándolas desde la raíz hasta un nodo hoja, que proporciona la clasificación. Cada nodo especifica una prueba de algún atributo y cada rama corresponde a uno de los posibles valores de ese atributo. Un ejemplo típico sería clasificar mañanas de sábado según si son adecuadas para jugar al tenis.

3.3 Problemas Apropriados para el Aprendizaje de Árboles de Decisión

El aprendizaje de árboles de decisión es adecuado para problemas donde:

- Las instancias se representan mediante pares atributo-valor.
- La función objetivo tiene valores de salida discretos.
- Pueden requerirse descripciones disyuntivas.
- Los datos de entrenamiento pueden contener errores.
- Los datos de entrenamiento pueden tener valores de atributo faltantes.

3.4 El Algoritmo Básico de Aprendizaje de Árboles de Decisión

La mayoría de los algoritmos para aprender árboles de decisión utilizan un enfoque de búsqueda codiciosa de arriba hacia abajo. ID3 y C4.5 son ejemplos de estos algoritmos. ID3 selecciona el mejor atributo en cada paso para crecer el árbol.

3.4.1 ¿Qué Atributo es el Mejor Clasificador?

ID3 utiliza la ganancia de información para seleccionar el mejor atributo en cada paso. Esta ganancia mide cuánto se espera que disminuya la entropía al clasificar según ese atributo.

3.4.2 Ejemplo Ilustrativo

Se presenta un conjunto de ejemplos de entrenamiento relacionados con la decisión de jugar al tenis en función de atributos como el clima, la temperatura, la humedad y el viento.

Conclusión:

El aprendizaje de árboles de decisión es una herramienta poderosa en la inferencia inductiva, capaz de manejar datos ruidosos y representar decisiones complejas. La elección del atributo adecuado en cada paso es esencial para la eficacia del árbol resultante.

Notas clase

Árboles de decisión

Los arboles de decision evaluan el conjunto de entrenamiento mediante la union de reglas de decision basado en los atributos de los datos.

Lo que vamos a elegir es un arbol de decision aplicando un algoritmo.

La potencia de los arboles de decision es que pueden implementar cualquier funcion booleana.

Son faciles de entender ademas pq son facilmeente traducibles a las reglas que evaluan los datos.

Hay varios algoritmos que construyen arboles de decision, nosotros vamos a er ID3.

ID3

La idea es elegir un atributo y profundizar en el arbol hasta llegar a una hoja (es decir, una regla que evalua true o false).

Algoritmo:

- creamos una raiz (despues vemos como elegir la raiz)
- si todos los ejemplos para el valor de ese atributo evaluan true o false, entonces creamos una hoja con ese valor
- si no me quedan atributos, etiqueto con el valor mas comun.

- en caso contrario, la raíz pregunta por A , el atributo que mejor clasifica los ejemplos. Para cada valor v_i de A ,
 - genera una rama
 - $Ejemplos_{v_i} = \{ejemplos \text{ en los cuales } A=v_i\}$
 - si $Ejemplos_{v_i}$ es vacío, entonces creo una hoja con el valor más común
 - en caso contrario, creo un subárbol con raíz A y profundizo en el subárbol con los ejemplos $Ejemplos_{v_i}$

¿Cuál es el mejor atributo? El que mejor clasifica los ejemplos. Usamos la noción de entropía para medir la calidad de la clasificación.

En particular la ganancia de información de un atributo, que es la diferencia entre la entropía del conjunto de ejemplos y la entropía de los ejemplos clasificados por el atributo.

Buscamos medir la reducción en la entropía al particionar por el atributo A .

Ganancia es el número de bits que ahorramos si sabemos el valor del atributo A .

Sesgo de ID3

Aca podemos identificar dos tipos de sesgo.

-sesgo preferencial: el algoritmo prefiere ciertas hipótesis sobre otras. En este caso, ID3 prefiere árboles cortos.

-sesgo restrictivo: se maneja con un espacio de hipótesis restringido. En este caso, ID3 considera solo árboles de decisión, pero los árboles de decisión pueden representar cualquier función booleana.

Sobreajuste de ID3

h e H se sobreajusta a un conjunto de entrenamiento si existe h' e H tal que h comete menos errores en el conjunto de entrenamiento que h' , pero comete más errores en el conjunto de prueba.

Se puede dar por:

- ruido en los datos: el árbol aprende a clasificar correctamente ejemplos erróneos, que lo inducen a clasificar mal ejemplos del dominio más general.
- cantidad insuficiente de ejemplos: aparecen regularidades que no se cumplen en el dominio, en nuestro caso, ¿qué tan importante es el horario en un examen?

Se puede corregir:

- Deteniendo el crecimiento del árbol antes de que se sobreajuste.
- Luego de obtenido el árbol, aplicar técnicas de poda.

Evitar sobreajustes

Poda de reglas [C4.5]

- Transformar el arbol en el conjunto de reglas que lo representan.
- Quitamos condiciones de las reglas de forma que estas mejoren su estimacion en el conjunto de validacion
- Ordenamos las reglas de acuerdo a su porcentaje de aciertos en el conjunto de validacion y ante nuevas instancias, aplicamos en orden.

Problema de atributos continuos

Se puede generar dinamicamente intervalos para estos atributos para que luego compitan como atributos discretos. Los arboles de regresion permite manejar los casos en donde la propia funcion objetivo es continua.

Ejemplo con atributos incompletos

En algunos casos los ejemplos tienen atributos sin valores.

Se puede:

- Suplirlos con otros ejemplos que si los tengan.
- Una estrategia es asignar el valor mas comun de ese atributo faltante en todo el conjunto de entrenamiento o en el subconjunto de ese nodo.
- El valor tambien puede asignarse de acuerdo a ciertas probabilidad usualmente estimadas a partir de los otros ejemplos.

Random Forest

Utiliza bagging: se entrena un conjunto de arboles de decision, cada uno con un subconjunto de ejemplos de entrenamiento elegidos al azar.

Se construyen muchos arboles no correlacionados y se someten cortes y decisiones a votacion.

hiperparametros:

- cuantos arboles voy a construir
- cuantos ejemplos voy a usar para entrenar cada arbol
- cuantos atributos voy a usar para entrenar cada arbol

Extra Trees / Ada Boost

Parecido a random forest, pero en vez de elegir el mejor corte, se elige un corte al azar.