

Algoritmos Seleccionados

Decision Stump

Un Decision Stump (tronco de decisión) es un árbol de decisión que utiliza un solo atributo para la división en el caso de los atributos discretos, esto suele significar que el árbol sólo consta de un único nodo interior (es decir, la raíz sólo tiene hojas como nodos sucesores). Si el atributo es numérico, el árbol puede ser más complejo. Los nodos hoja de un árbol de decisión contienen el nombre de clase, mientras que un nodo que no es hoja es un nodo de decisión. El nodo de decisión es una prueba de atributo con cada rama (a otro árbol de decisión) siendo un valor posible del atributo. Este operador se puede aplicar tanto en conjuntos de datos nominales como numéricos. [3]

REPTree

Rep Tree es un algoritmo de aprendizaje que construye un árbol de decisión usando reducción de ganancia/varianza de la información y poda basada en la reducción del error. Debido a que este algoritmo se encuentra optimizado para ser rápido, ordena los valores numéricos sólo una vez. Los valores perdidos, se manejan dividiendo las instancias existentes en segmentos. Los parámetros que pueden ser establecidos son: número mínimo de instancias en cada hoja, la profundidad máxima del árbol y la cantidad de datos usada para la poda. [2]

Hoeffding Tree

El algoritmo del árbol de Hoeffding es un método de aprendizaje del árbol de decisión para la clasificación de datos de flujo porque es capaz de aprender de flujos de datos masivos. Los árboles de Hoeffding aprovechan el hecho de que una pequeña muestra a menudo puede ser suficiente para elegir un atributo de división óptimo.[4]

Características

- ✓ A diferencia de árboles de decisiones incrementales es que tiene sólidas garantías de rendimiento,
- ✓ El algoritmo toma como entrada una secuencia de ejemplos de entrenamiento, S , descrita por los atributos A , y el parámetro de precisión.
- ✓ Las únicas estadísticas que deben mantenerse en el algoritmo del árbol de Hoeffding son los recuentos n_{ijk} para el valor v_j del atributo A_i con etiqueta de clase k .

J48

La clasificación es el proceso de construir un modelo de clases a partir de un conjunto de registros que contienen etiquetas de clase. El algoritmo de árbol de decisión sirve para averiguar el comportamiento del vector de atributos para una serie de instancias. También sobre la base de las instancias de entrenamiento se encuentran las clases para las instancias recién generadas. Las características adicionales de J48 son la contabilización de los valores perdidos, la poda de los árboles de decisión, los rangos de valores de atributos continuos, la derivación de reglas. [1]

RandomTree

El operador de árbol aleatorio funciona exactamente igual que el operador de árbol de decisión con una excepción: para cada división solo está disponible un subconjunto aleatorio de atributos. Se recomienda que estudie la documentación del operador Árbol de decisión para obtener una comprensión básica de los árboles de decisión.

Este operador aprende árboles de decisión a partir de datos nominales y numéricos. Los árboles de decisión son poderosos métodos de clasificación que pueden entenderse fácilmente.

La representación de los datos como árbol tiene la ventaja en comparación con otros enfoques de ser significativa y fácil de interpretar. El objetivo es crear un modelo de clasificación que prediga el valor de la etiqueta en función de varios atributos de entrada de ExampleSet. Cada nodo interior del árbol corresponde a uno de los atributos de entrada. El número de aristas de un nodo interior es igual al número de valores posibles del atributo de entrada correspondiente. Cada nodo de hoja representa un valor de la etiqueta dados los valores de los atributos de entrada representados por la ruta desde la raíz hasta la hoja.[5]

Tabla Comparativa entre los Algoritmos de Clasificación por Clase de Vehículo

Algoritmos	Tiempo necesario para construir el modelo	Instancias clasificadas incorrectamente	Instancias clasificadas incorrectamente	Error absoluto relativo
Decision Stump	0.09 seconds	5894 (79.8104 %)	5894 (79.8104 %)	68.908 %
REPTree	0.41 seconds	675(9.1401 %)	675(9.1401 %)	10.9292 %
Hoeffding Tree	0.23 seconds	1961(26.5538 %)	1961(26.5538 %)	68.908 %
J48	0.16 seconds	670 (9.0724 %)	670 (9.0724 %)	10.6816 %
Random Tree	0.05 seconds	729(9.8714 %)	729(9.8714 %)	11.7208 %

Bibliografía

[1]. [J48] IHYA, R. y Namir, A. (2019). *J48 Algorithms of machine learning for predicting user's the acceptance of an E-orientation Systems*.

[2] Bae, B. P. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. *Expert Syst. Appl*, 42(6), 2928–2934.

[3] RapidMiner GmbH, "Decision stump - RapidMiner documentation", Rapidminer.com.
[En línea]. Disponible en:
https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/decision_stump.html. [Consultado: 08-feb-2022].

[4] "¿Qué es el algoritmo del árbol de Hoeffding?" <https://www.tutorialspoint.com/what-is-hoeffding-tree-algorithm> (accessed Feb. 07, 2022).

[5] Random Tree - RapidMiner Documentation. (n.d.). Retrieved February 7, 2022, from https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/random_tree.html