Clasificadores basados en reglas

UCO ONLINE

Clasificadores basados en reglas

Introducción a los clasificadores basados en reglas

Los árboles de decisión pueden convertirse fácilmente en un conjunto de reglas siguiendo cada posible camino desde el nodo raíz hasta cada una de las hojas del árbol. Tomando como ejemplo el árbol creado con C4.5 en la lección anterior (Figura 1), se podrían obtener las reglas que se observan en la Tabla 1.

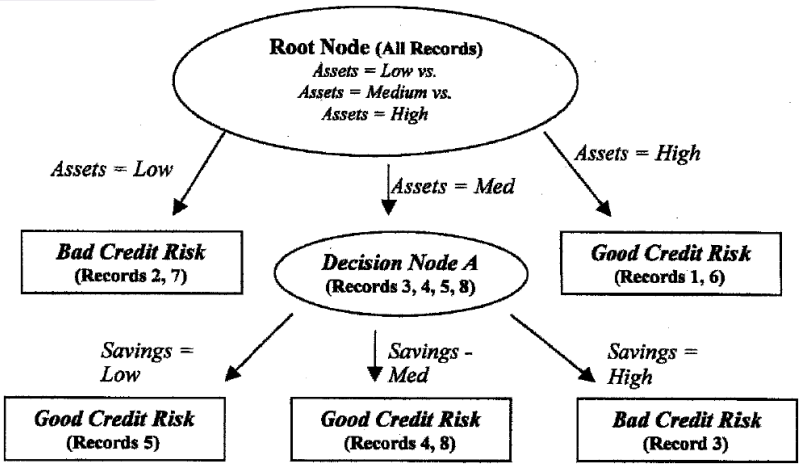

Figura 1. Árbol de decisión creado con C4.5.

Tabla 1. Reglas obtenidas a partir del árbol de decisión anterior.

Antecedent	Consequent	Support	Confidence
If *assets* = *low*	then *bad credit risk.*	$\frac{2}{8}$	1.00
If *assets* = *high*	then *good credit risk.*	$\frac{2}{8}$	1.00
If *assets* = *medium* and *savings* = *low*	then *good credit risk.*	$\frac{1}{8}$	1.00
If *assets* = *medium* and *savings* = *medium*	then *good credit risk.*	$\frac{2}{8}$	1.00
If *assets* = *medium* and *savings* = *high*	then *bad credit risk.*	$\frac{1}{8}$	1.00

Las reglas de decisión se representan en la forma ***SI*** *antecedente,* ***ENTONCES*** *consecuente* (*IF … THEN …*). Para generar un clasificador con dichas reglas, el antecedente coincide con valores específicos para uno o varios atributos, y el consecuente será un valor para la variable objetivo o clase que se encontraba en el nodo hoja.

El soporte (*support*) de cada regla se refiere a la proporción de instancias en el conjunto de entrenamiento que cumplen dichas condiciones (en este caso, que estaban en un nodo hoja específico). Por otro lado, la confianza (*confidence*) indica, qué porcentaje de las instancias que cumplen el antecedente, cumplen también el consecuente. Como en nuestro ejemplo construimos el árbol hasta obtener nodos hoja puros, todas las reglas tienen una confianza de 1 (=100%), pero no tendría por qué ser así utilizando otro método de construcción de la base de reglas.

Dado un conjunto de reglas, y una nueva instancia cualquiera, se recorre la lista de reglas en orden. En el momento en que la nueva instancia cumple con las condiciones del antecedente de una regla, se le asigna la clase que se indica en el consecuente, y no se continúa con la búsqueda. Por lo general (aunque en la Tabla 1 no aparezca) suele incluirse una última regla por defecto, donde si la nueva instancia no cumple con el consecuente de ninguna de las reglas anteriores, se le asigna dicha clase por defecto; por lo general, se asigna la clase mayoritaria.

En este ejemplo, se ha construido el conjunto de reglas para el clasificador a partir de un árbol de decisión, pero también existen algoritmos para inducir dichas reglas directamente. El objetivo final es construir un conjunto de reglas, en principio relativamente pequeño o manejable, que sea capaz de interpretar los datos de entrenamiento y por tanto poder clasificar nuevas instancias de los datos.

Un método principal para la construcción de dichas reglas es: 1) buscar una regla que explique de la mejor manera posible parte de sus instancias de entrenamiento; 2) separar esas instancias; y 3) buscar recursivamente nuevas instancias hasta que no queden instancias sin cubrir por ninguna de las reglas. Algoritmos de inducción de reglas para clasificación difieren de este procedimiento general en que suelen incluir métodos más avanzados para prevenir el sobre-entrenamiento del modelo.

Métodos de construcción de conjuntos de reglas para clasificación

Nótese que, dada la existencia de una asignatura de métodos descriptivos en este mismo máster, donde se explicará en detalle la generación de reglas de asociación a partir de los datos, no se entrará en mayor detalle en cómo generarlas. El objetivo principal de esta lección es conocer la existencia de estos sistemas basados en reglas, comprenderlos, y saber aplicarlos a problemas reales. RIPPER [Coh95] es un algoritmo muy conocido para la inducción de clasificadores basados en reglas. RIPPER genera las reglas a través de aplicar un proceso de *crecimiento* y *podado* (*growing and pruning*) repetidas veces. Durante la fase de crecimiento, las reglas se hacen más restrictivas

para ajustarse a los datos de entrenamiento tanto como sea posible. Durante la fase de podado, las reglas se hacen menos restrictivas para evitar el sobre-entrenamiento. RIPPER es capaz de lidiar con múltiples clases ordenándolas de menor a mayor prevalencia, y tratándolas en orden como si se tratase de un problema de dos clases.

PART [Fun99] infiere las reglas por medio de generar árboles de decisión parciales repetidamente. Cada vez que se construye un árbol, se extrae una única regla de él, por lo que PART no necesita un post-procesado del modelo.

Referencias

[Coh95] Cohen, W. W. (1995). Fast effective rule induction. In Machine learning proceedings 1995 (pp. 115-123).

[Fun99] Fürnkranz, J. (1999). Separate-and-conquer rule learning. Artificial Intelligence Review, 13(1), 3-54.

[Kot07] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), 3-24.

[Lar14] Larose, D. T., & Larose, C. D. (2014). *Discovering knowledge in data: an introduction to data mining* (Vol. 4). John Wiley & Sons.

[Mai10] Maimon, O., & Rokach, L. (Eds.). (2010). Data mining and knowledge discovery handbook, 2nd edition. *Springer*.