Práctica 8: Reglas de Asociación – Clasificación

Introducción

En un contexto de clasificación, cuando se utilizan Reglas de Asociación, se las denomina Reglas de Clasificación o Reglas de Asociación de Clase (CARs).

Es una técnica de la Minería de Datos que consiste en, dado un conjunto de instancias de entrenamiento, identificar ciertas características en las instancias para construir reglas que posteriormente se utilicen en la clasificación de nuevas instancias.

Reglas de Asociación de Clase

Dados **I** un conjunto de ítems, **C** un conjunto de clases, $\mathbf{T^c}$ un conjunto de transacciones de la forma $\{i_1, i_2,..., i_n, c\}$ tal que $\forall\ 1 \leq k \leq n\ [i_k \in \mathbf{I} \wedge c \in \mathbf{C}]$; construir un clasificador basado en CARs consiste en:

1) encontrar un conjunto de reglas **R**, de la forma $\mathbf{X} \Rightarrow c$ tal que $\mathbf{X} \subseteq \mathbf{I}$ y $c \in \mathbf{C}$;
2) ordenar el conjunto de reglas **R** y
3) definir un criterio de decisión **D** que utilice a **R** para asignar una clase a cada transacción t que se desee clasificar.

De lo anterior se desprende que $\mathbf{T^c}$ es un conjunto de transacciones etiquetadas (Tabla 1).

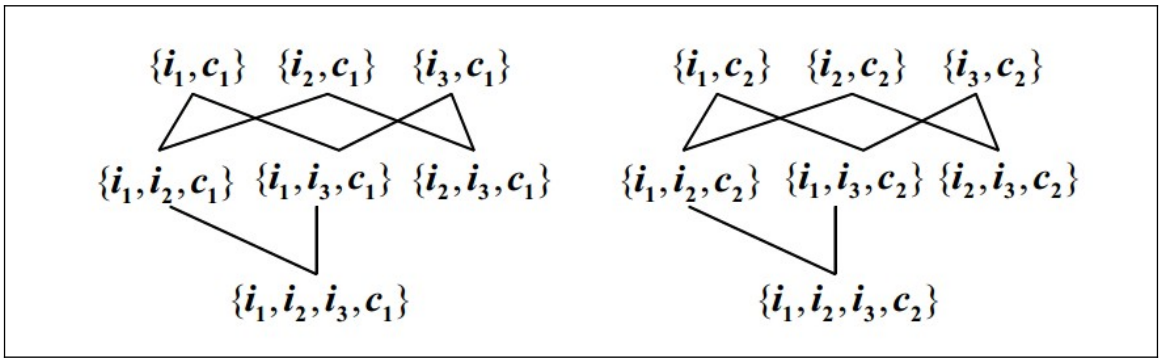
Tabla 1. Ejemplo de transacciones etiquetadas.

T^C	Ítems				Clase
t_1	i_{11}	i_{12}	…	i_{1k}	c_1
t_2	i_{21}	i_{22}	…	i_{2k}	c_2
…			…		
t_n	i_{n1}	i_{n2}	…	i_{nk}	c_n

Búsqueda de CARs

De la misma forma que se buscan itemsets frecuentes, antes de buscar reglas de asociación, las CARs se buscan en un espacio de búsqueda comprendido por el conjunto **I** y por **C**.

Por ejemplo, dado el conjunto de ítems $\mathbf{I = \{i_1, i_2, i_3\}}$ y un conjunto de clases $\mathbf{C = \{c_1, c_2\}}$, el espacio de búsqueda es:



Donde el conjunto **{i_1, i_2, c_1}** representa a la CAR: **{i_1, i_2}** → **c_1**.

El soporte de las CARs disminuye a medida que se avanza a niveles de mayor *k*.

Los algoritmos de minado de CARs utilizan distintas estrategias para recorrer el espacio de búsqueda, que dependiendo de la dirección pueden ser:

Descendentes: desde el primer nivel (k = 2) hasta el nivel donde no se obtiene ninguna CAR.

Ascendentes: desde un nivel aproximado al último donde se generan reglas, hasta el primer nivel.

Además de las estrategias para recorrer el espacio de búsqueda, también hay estrategias para generar las reglas: en amplitud o en profundidad.

Algoritmo CBA (*Classification based on Association*)

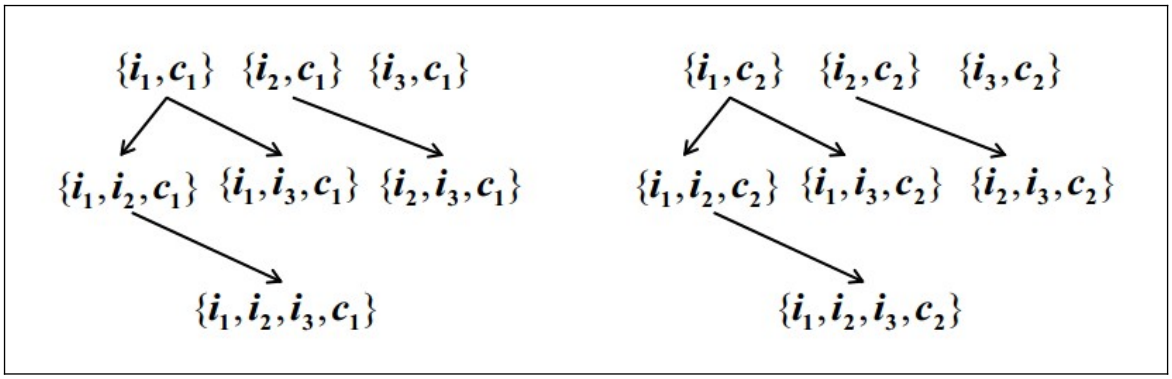
Genera todas las reglas de tamaño k antes de generar las de tamaño k + 1 (amplitud). Utiliza una modificación del algoritmo A priori.

Primero calcula todas las reglas que satisfacen el soporte y confianza mínimos. Luego, selecciona un subconjunto más pequeño de CARs que cubra al conjunto de entrenamiento y con ese subconjunto construye el clasificador.

Si hay una regla con igual antecedente y diferente consecuente, elije la de mayor confianza y elimina las demás.

Algoritmo CMAR (*Classification based on Multiple Association Rules*)

Se generan las reglas por cada rama de la estructura de árbol del espacio de búsqueda. Al terminar con un ítem, continúa con el siguiente:



Utiliza una modificación del algoritmo FP-Growth (más eficiente que Apriori).

Implementación en R

Para implementar reglas de asociación de clase en R se utilizará el paquete **arulesCBA** [2]. En este paquete se implementan tanto el algoritmo CBA, cómo el CMAR. Las funciones son *CBA()* y *CMAR()* respectivamente y los argumentos de las funciones son:

- formula: expresión simbólica del modelo a ajustar. Se utiliza el operador **~** de la forma: "clase **~** ." o "clase **~** var1 **+** var2 **+** …".
- Data: objeto de la clase *transactions*.
- Parameter: lista con los parámetros para el algoritmo A priori (en el caso de CBA).
- Support y confidence: umbrales de soporte y confianza para el algoritmo CMAR.

Actividades

1 – Cargue el archivo “titanic.raw.rdata” utilizado en el ejercicio 2 de la guía 5. Utilizando validación simple en una relación 80%-20%, entrene un clasificador basado en reglas de asociación para clasificar si el pasajero sobrevive o no, según las siguientes características:

a) CBA con soporte mínimo 0,1 y confianza mínima de 0,3. ¿Cuántas CARs encuentra?
b) Repita para soporte = 0,001 y confianza = 0,8. ¿Cuántas CARs encuentra?
c) CMAR con soporte mínimo 0,1 y confianza mínima de 0,3. ¿Cuántas CARs encuentra?
d) Repita para soporte = 0,001 y confianza = 0,8. ¿Cuántas CARs encuentra?
e) Encuentre las predicciones para cada uno de los modelos, muestre las matrices de confusión de cada caso y compare su precisión para elegir el mejor modelo.

2 – Utilice el archivo “Entrenamiento_ECI_2020.csv” de la actividad 2 de la guía 3 para entrenar un modelo de CARs, con validación simple; y clasificar entre las clases “Closed Won” y otros. Utilice las variables categóricas adecuadas del dataset.

a) Encuentre las CARs para un soporte de 0,1 y una confianza de 0,5 utilizando CBA. ¿Cuántas reglas obtiene?
b) Prediga utilizando los datos de test y muestre la matriz de confusión del modelo. ¿Cuál es la precisión del modelo?
c) Compare la precisión del modelo obtenido con CMAR. ¿Cuál clasifica mejor?

Referencias

1. Orallo, J. et all. “Introducción a la Minería de Datos” (2004). Capítulo 9.
2. Paquete arulesCBA. Disponible en: https://cran.r-project.org/web/packages/arulesCBA/arulesCBA.pdf