RNPS No. 2143
ISSN 2072-6260
Versión Digital

SERIE GRIS

REPORTE TÉCNICO
Minería de Datos

CAR-IC y CAR-NF: nuevos clasificadores basados en CARs

Raudel Hernández León y
José Hernández Palancar

RT_018 **agosto 2013**

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

RNPS No. 2143
ISSN 2072-6260
Versión Digital

REPORTE TÉCNICO
Minería de Datos

SERIE GRIS

CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

CAR-IC y CAR-NF: nuevos clasificadores basados en CARs

Raudel Hernández León y
José Hernández Palancar

RT_018 **agosto 2013**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

Tabla de contenido

Lista de figuras

CAR-IC y CAR-NF: Nuevos clasificadores basados en CARs

Raudel Hernández León y José Hernández Palancar

Dpto. Minería de Datos. Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV).
La Habana, Cuba.
{rhernandez,jpalancar}@cenatav.co.cu

RT_018, Serie Gris, CENATAV
Aceptado: 30 de octubre de 2012

Resumen. La *clasificación* basada en 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. La clasificación basada en CARs se ha utilizado en diferentes tareas como: la clasificación y segmentación de textos, el etiquetado automático de imágenes, entre otras. No obstante, los principales clasificadores desarrollados, basados en CARs, presentan varias limitaciones.
En el marco de este reporte técnico se introduce un algoritmo para calcular el conjunto de reglas, CAR-CA, el cual introduce una nueva estrategia de poda que permite obtener reglas específicas, en lugar de reglas generales, con altos valores de la medida de calidad. Además, se introducen dos clasificadores basados en CARs, CAR-IC y CAR-NF, que utilizan una nueva estrategia de ordenamiento basada en el tamaño de las reglas, un nuevo criterio de cubrimiento que considera el cubrimiento inexacto en ausencia de reglas que cubran completamente a la nueva instancia, y un nuevo criterio de decisión para asignar una clase a la nueva instancia. Como un resultado adicional, se demuestra cual es el mínimo valor de la medida de calidad que evita la ambigüedad al momento de clasificar y ambos clasificadores utilizan este valor. En el caso específico del clasificador CAR-NF se introduce el uso de la medida de calidad Netconf para calcular las reglas. Los experimentos realizados muestran que los clasificadores propuestos son superiores en calidad a los clasificadores más exitosos basados en CARs.

Palabras clave: clasificación supervisada, reglas de asociación de clase, medidas de calidad.

Abstract. Classification based on Class Association Rules (CARs) or associative classification is a data mining technique that consists of, given a training instance set, finding certain characteristics in the instances in order to build rules that are subsequently used for classifying unseen instances. Associative classification has been used in different tasks, for example: text classification, text segmentation, and automatic image annotation, among others. However, associative classification methods still have some weaknesses.
In this technical report we propose an algorithm called CAR-CA, which introduces a new pruning strategy that allows to obtain rules with high values of the quality measure. Besides, we introduce two classifiers based on CARs, CAR-IC and CAR-NF, both use a new way for ordering the set of CARs based on the rule size, a new covering criterion that considers the inexact coverage when any rule covers the new instance, and a new strategy for deciding the class of a new instance. Additionally, these classifiers use as threshold for the quality measure, the minimum value that avoids ambiguity at the classification stage. In particular, the CAR-NF classifier introduces the use of the Netconf measure to compute the set of CARs. The experimental results show that the proposed CARs based classifiers CAR-IC and CAR-NF have better performance than the main successful classifiers based on CARs.

Keywords: supervised classification, class association rules, quality measures.

1. Introducción

Actualmente, la mayor parte de la información generada por los sistemas informáticos se almacena para su posterior consulta y/o procesamiento. No obstante, en muchas ocasiones el volumen de información almacenado es muy grande para ser analizado por un humano. La Minería de Datos ofrece diversas técnicas que permiten atacar este problema y descubrir información implícita en grandes conjuntos de datos.

Una de las técnicas más estudiadas de la Minería de Datos es la clasificación supervisada o simplemente "clasificación". La clasificación consiste en identificar características esenciales de diferentes clases de instancias a partir de un conjunto de entrenamiento y posteriormente, utilizar estas características para determinar la clase de nuevas instancias.

Entre los enfoques de clasificación más utilizados en el estado del arte se encuentran los probabilísticos [22,24], los basados en inducción de reglas [20,39], los árboles de decisión [38], las máquinas de soporte vectorial [23], los híbridos, que combinan reglas de asociación y listas de decisión [11], las redes neuronales [33] y las reglas de asociación de clase [34]. Particularmente, los clasificadores basados en reglas de asociación de clase (CAR, por sus siglas en inglés) son preferidos por muchos especialistas debido a su alta interpretabilidad [32,49,53,54], aspecto que hace que tanto el proceso de clasificación como los resultados sean más fáciles de entender. Además, su alta interpretabilidad permite a los especialistas modificar las reglas con base en su experiencia y así mejorar la eficacia del clasificador.

Además de los clasificadores basados en CARs, los árboles de decisión también generan reglas comprensibles. Para construir un clasificador utilizando árboles de decisión se sigue una estrategia voraz seleccionando en cada momento la característica que mejor separa las clases. Sin embargo, esta estrategia voraz puede podar reglas interesantes. En [47], los autores probaron que las reglas obtenidas de los árboles de decisión son un subconjunto de las reglas generadas por los clasificadores basados en CARs, asumiendo un umbral relativamente bajo de concurrencia de los elementos que componen la regla.

Una CAR es un caso particular de regla de asociación (AR, por sus siglas en inglés) que está compuesta por un conjunto de elementos o ítems (antecedente) y una clase (consecuente). Un ejemplo de CAR es el siguiente:

$$\text{pan, leche, carne} \Rightarrow \text{efectivo} ,$$

lo que se interpreta cómo:

$$(\text{compró pan}) \wedge (\text{compró leche}) \wedge (\text{compró carne}) \Rightarrow (\text{pagó en efectivo}) ,$$

donde "$\wedge$" es el operador lógico de conjunción. Esta regla nos dice que el antecedente formado por los ítems "pan", "leche" y "carne" implica al consecuente (la clase) formado por el ítem "efectivo". Dado un conjunto de transacciones de compras realizadas en un mercado, puede resultar interesante una CAR que ocurra muchas veces o una CAR donde la probabilidad del consecuente dado el antecedente sea alta; la frecuencia con que ocurre una CAR en un conjunto de transacciones es conocido como el Soporte de la CAR y la probabilidad de que esté presente el consecuente, dado que está presente el antecedente, se conoce como la Confianza de la CAR.

Los clasificadores basados en CARs se han utilizado en diferentes tareas como: la reducción de fallas en las telecomunicaciones y la detección de redundancia en exámenes médicos [5], la

clasificación de imágenes médicas [6,36,40,41], la clasificación de secuencias de genes [29], la clasificación de textos [13], la diferenciación de células madres mesenquimales en mamíferos [50] y la predicción de tipos de interacciones proteína-proteína [35], entre otros.

1.1. Problemática actual

En general, un clasificador basado en CARs está compuesto por un conjunto ordenado de CARs y un criterio de decisión. Para clasificar una nueva transacción t, se determina el conjunto de CARs que satisfacen o cubren a t y se utiliza el criterio de decisión para determinar la clase a la que pertenece t.

Varios trabajos [32,46,54] han proporcionado evidencias de que los clasificadores basados en CARs son competitivos con los clasificadores probabilísticos [22], con los basados en árboles de decisión [38] y con los basados en inducción de reglas [20,39]. No obstante, los principales clasificadores desarrollados, basados en CARs, presentan las siguientes limitaciones:

- Para generar las CARs se utilizan las medidas de calidad Soporte y Confianza, las cuales representan la frecuencia de la regla y la confianza que se tiene en la misma. No obstante, ambas medidas presentan varias limitaciones. La selección de un umbral de Soporte muy alto puede generar CARs que contengan información demasiado evidente, pero dejar de generar CARs interesantes. Por el contrario, la selección de un umbral de Soporte muy bajo puede generar un gran volumen de CARs, las cuales pueden ser de poca relevancia para el análisis de los datos o inducir a criterios falsos como resultado de dicho análisis. Entre las limitaciones de la Confianza se encuentran las siguientes: (1) no detecta independencia estadística entre el antecedente y el consecuente de la regla, (2) no detecta dependencias negativas; lo que trae como consecuencia que se generen reglas engañosas y (3) no considera en su definición al consecuente, en nuestro caso, la clase. Adicionalmente a las limitaciones mencionadas, en ningún trabajo se fundamentan los umbrales de Soporte y Confianza utilizados para evaluar la calidad de las CARs.
- Los trabajos recientes, en el afán de reducir el volumen de CARs generadas, podan el espacio de CARs cada vez que encuentran una CAR que satisface ciertas restricciones. Utilizando esta heurística se obtienen CARs demasiado generales (pequeñas) dejándose de obtener posibles CARs un poco más específicas (grandes) que pudieran ser de mayor interés.
- Al momento de clasificar nuevas transacciones, una CAR cubre a una transacción si todos los ítems de su antecedente están contenidos en la transacción. Debido a esto, sucede con frecuencia que ninguna CAR cubre a la nueva transacción y el clasificador asigna a la nueva instancia la clase mayoritaria o se abstiene. Un gran número de asignaciones de la clase mayoritaria o abstenciones puede afectar la eficacia del clasificador.
- Los criterios de decisión existentes ("La Mejor Regla", "Las Mejores K Reglas", "Todas las Reglas"), para determinar la clase que se asignará a una nueva transacción, tienen las siguientes limitaciones:
  a) Si se utiliza el criterio "La Mejor Regla" se está apostando a que una sola regla puede predecir correctamente la clase de cada transacción que ésta cubra, lo cual es poco probable.
  b) Si se utiliza el criterio "Las Mejores K Reglas" puede afectarse la eficacia cuando 1) existe desbalance en el número de CARs con altos valores de la medida de calidad, por clase, que cubren a la nueva transacción; o cuando 2) la mayoría de las mejores K reglas se obtuvieron a partir del mismo ítem, dando lugar a cierta redundancia.

c) Si se utiliza el criterio "Todas las Reglas" pueden incluirse reglas con bajos valores de la medida de calidad en el conjunto de reglas utilizado para clasificar.

Como se acaba de mencionar, independientemente de cuánto se ha avanzado en el desarrollo de clasificadores basados en CARs, aún quedan aspectos que resolver. Los diferentes resultados que se presentan en este reporte técnico van dirigidos a resolver estas deficiencias.

1.2. Organización del reporte

El resto del documento está organizado de la siguiente forma:

En la Sección 2 se presenta el marco teórico necesario para abordar el problema de la clasificación basada en CARs. Se incluyen los conceptos preliminares, las medidas de calidad utilizadas para calcular las CARs, las estrategias de ordenamiento de CARs y los criterios de decisión existentes.

En la Sección 3 se introduce el algoritmo CAR-CA, desarrollado para calcular un conjunto de CARs a partir de un conjunto de entrenamiento.

En la Sección 4 se introducen la nueva estrategia de ordenamiento y los nuevos criterios de cubrimiento y decisión. Adicionalmente, se introducen los nuevos clasificadores y se presentan los experimentos realizados para evaluarlos.

Finalmente, en la Sección 5 se presentan las conclusiones y el trabajo futuro.

2. Marco teórico

Es importante aclarar que este reporte tiene como precedente un estado del arte [27] donde se describen los diferentes clasificadores reportados, basados en CARs. Debido a lo anterior, en esta sección sólo se presenta el marco teórico necesario para abordar el problema de la clasificación basada en CARs. En la Subsección 2.1 se define formalmente el problema de la construcción de clasificadores basados en CARs y se dan los conceptos necesarios para entender el resto del documento. En la Subsección 2.2 se realiza un análisis crítico de las medidas de calidad de ARs utilizadas para calcular las CARs y en las Subsecciones 2.3 y 2.4 se analizan las estrategias de ordenamiento de CARs y los criterios de decisión, utilizados para elegir la clase que se asigna a la nueva instancia, reportados en la literatura.

2.1. Conceptos preliminares

El problema de la construcción de clasificadores basados en CARs se define formalmente como sigue:

Dados I un conjunto de ítems, C un conjunto de clases, T^C un conjunto de transacciones de la forma $\{i_1, i_2, ..., i_n, c\}$ tal que $\forall_{1 \leq k \leq n}[i_k \in I \wedge c \in C]$ (ver Tabla 2.1); construir un clasificador basado en CARs consiste en: 1) encontrar un conjunto de reglas R, de la forma $X \Rightarrow c$ tal que $X \subseteq I$ y $c \in C$; 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.

Al igual que el minado de conjuntos frecuentes de ítems, paso previo al minado de ARs [3,25,55], el cálculo de todas las CARs es una tarea que consume muchos recursos debido a su complejidad exponencial [3]. Por lo general, el cálculo de las CARs se corresponde con un recorrido

Tabla 1. Representación general de un conjunto de transacciones.

Transacciones

T^C	Ítems				Clase
t_1	i_{11}	i_{12}	...	i_{1k_1}	c_1
t_2	i_{21}	i_{22}	...	i_{2k_2}	c_2
...			...		
t_n	i_{n1}	i_{n2}	...	i_{nk_n}	c_n

por el retículo[1] formado por las CARs. Para comprender mejor lo anterior, supóngase un conjunto de tres ítems $I = \{i_1, i_2, i_3\}$ y dos clases $C = \{c_1, c_2\}$. En la Figura 1 se observa el retículo (espacio de búsqueda) de las CARs que se pueden formar con los ítems del conjunto I y las clases del conjunto C. El primer nivel del retículo está compuesto por las CARs de tamaño dos; noten que en este nivel, con el objetivo de ser más intuitivos, se repitieron los ítems de tamaño uno y se intercalaron las clases entre ellos. También buscando simplicidad se omitió el conjunto vacío y se representó cada CAR como un conjunto con los ítems que forman el antecedente al principio y la clase como último elemento, *e.g.* el conjunto $\{i_1, i_2, c_1\}$ representa a la CAR $\{i_1, i_2\} \Rightarrow c_1$.

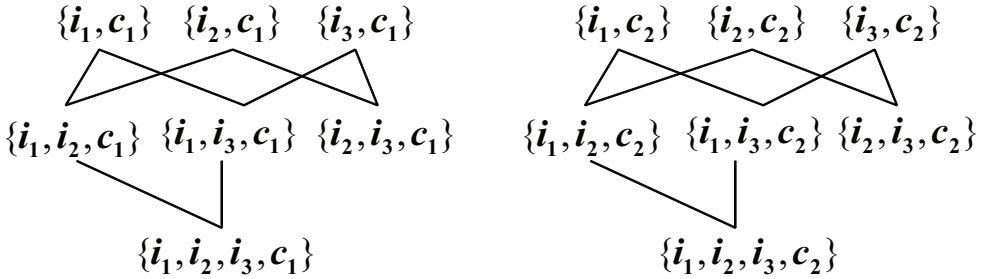


Fig. 1. Retículo de las CARs que se forman con tres ítems y dos clases.

Es fácil comprobar que los Soportes de las CARs (ver Def. 2.2 y 2.6) disminuyen al recorrer el espacio de búsqueda desde el primer nivel hasta el último, verificándose la clausura descendente del Soporte[2] al igual que sucede en el retículo que forman los conjuntos de ítems [3]. Los algoritmos de minado de CARs definen diferentes estrategias para recorrer el espacio de búsqueda. Estas estrategias pueden clasificarse atendiendo a la dirección del recorrido en:

- Descendentes: Si el recorrido se realiza desde el primer nivel (CARs de tamaño 2) hacia el último nivel, deteniéndose en el nivel donde no se genere ninguna CAR.
- Ascendentes: Si el recorrido se realiza en el sentido opuesto, desde un nivel aproximado a los últimos niveles donde se generen CARs (variaciones del algoritmo Eclat [55] de minado de ARs) hacia el primer nivel.

Al mismo tiempo, dentro de estas estrategias se pueden generar las CARs de dos formas:

1. En amplitud (*Breadth-First Search*): Se generan todas las CARs de tamaño k antes de generar las CARs de tamaño $k + 1$. Un clasificador que sigue esta estrategia es el CBA, que utiliza para calcular las CARs una modificación del algoritmo *Apriori* de minado de ARs.

[1] Un retículo, red o *lattice* es un conjunto parcialmente ordenado en el cual todo subconjunto finito no vacío tiene un supremo y un ínfimo.

[2] La clausura descendente del Soporte plantea que todo subconjunto de un conjunto frecuente de ítems es frecuente o lo que es igual, todo superconjunto de un conjunto no frecuente de ítems es no frecuente.

2. En profundidad (*Depth-First Search*): Se generan las CARs por cada rama de la estructura arbórea derivada del retículo (ver Fig. 2), es decir, por cada clase, se toma como antecedente el primer ítem y se extiende, tanto como sea posible, con los ítems lexicográficamente mayores a los ya incluidos, generando todos los superconjuntos frecuentes y retrocediendo (*backtracking*) cuando no se pueda extender más. Después de terminar con un ítem se toma el siguiente y se realiza el mismo proceso. Algunos de los clasificadores que siguen esta estrategia son CMAR y TFPC, los cuales utilizan modificaciones de los algoritmos de minado de ARs *Fp-growth* y Apriori-TFP respectivamente.

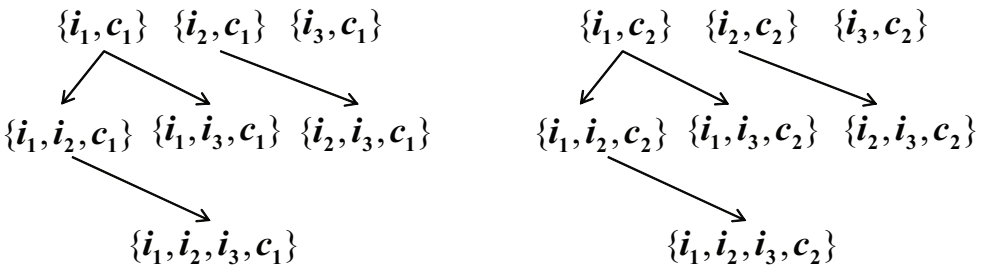


Fig. 2. Estructura arbórea derivada del retículo de la Figura 1.

Si se observa la estructura arbórea de la Figura 2, se puede apreciar que a partir del segundo nivel la CAR que etiqueta a cada nodo (CAR asociada al nodo) es el resultado de extender, con un nuevo ítem, el antecedente de la CAR asociada al nodo padre. En el caso del primer nivel, cada CAR se genera con un ítem y una clase.

En la estructura arbórea de la Figura 2 se puede observar que las CARs asociadas a los hijos de un nodo coinciden con la CAR asociada al nodo padre en todos los ítems del antecedente con excepción del último ítem. Adicionalmente, se puede notar que para cada clase c, el subárbol cuyo nodo raíz tiene asociada la CAR con el antecedente formado solo por el primer ítem abarca la mitad del espacio de búsqueda derivado de c; el subárbol cuyo nodo raíz tiene asociada la CAR con el antecedente formado solo por el segundo ítem abarca la cuarta parte del espacio de búsqueda derivado de c, y así sucesivamente; sobre esta característica de la estructura arbórea regresaremos más adelante.

Luego de formalizar el problema de la construcción de clasificadores basados en CARs y describir los recorridos del espacio de búsqueda reportados en la literatura, se presentarán los conceptos básicos del minado de ARs y sus extensiones al minado de CARs. Estos conceptos son necesarios para una mejor comprensión de este reporte y se utilizarán en el resto del documento.

Sean $I = \{i_1, i_2, \ldots, i_n\}$ un conjunto de n ítems y T un conjunto de transacciones, donde cada transacción $t \in T$ está formada por un conjunto de ítems X tal que $X \subseteq I$.

Definición 2.1. *El tamaño de un conjunto de ítems está dado por su cardinalidad; un conjunto de ítems de cardinalidad k se denomina k-itemset.*

Es importante aclarar que cuando se haga referencia a un conjunto de ítems X se estará hablando de un subconjunto de I y se supondrá, sin pérdida de generalidad, que existe un orden entre los ítems del conjunto I.

Definición 2.2. *El Soporte de un conjunto de ítems X, en adelante Sop(X), es la fracción de transacciones en T que contienen a X y se define como $\frac{|T_X|}{|T|}$, donde T_X es el conjunto de transacciones en T que contienen a X. El Soporte toma valores en el intervalo* $[0, 1]$.

Aunque el Soporte toma valores en el intervalo cerrado $[0, 1]$, en algunas demostraciones durante este reporte se trabajará con el intervalo abierto $(0, 1)$ ya que un conjunto de ítems con Soporte igual a 0 no está presente en ninguna transacción y un conjunto de ítems con Soporte igual a 1 está presente en todas las transacciones. En ambos casos, las reglas generadas por estos conjuntos de ítems no son útiles para clasificar.

Definición 2.3. *Sea* minSop *el umbral de Soporte previamente establecido, un conjunto de ítems X se denomina frecuente si $Sop(X) \geq minSop$.*

Definición 2.4. *Una AR sobre el conjunto de ítems I es una implicación $X \Rightarrow Y$ tal que $X \subset I$, $Y \subset I$ y $X \cap Y = \emptyset$.*

Definición 2.5. *Dadas dos reglas $R_1 : X_1 \Rightarrow Y$ y $R_2 : X_2 \Rightarrow Y$, R_1 es más específica que R_2 (R_2 es más general que R_1) si $X_2 \subset X_1$.*

Definición 2.6. *El Soporte de una regla de asociación $X \Rightarrow Y$ es igual a $Sop(X \cup Y)$.*

Definición 2.7. *La Confianza de una regla de asociación $X \Rightarrow Y$, en adelante Conf($X \Rightarrow Y$), es la probabilidad de encontrar a Y en las transacciones que contienen a X y se define, en función del Soporte, como $\frac{Sop(X \cup Y)}{Sop(X)}$. La Confianza toma valores en el intervalo $[0, 1]$.*

Para extender las definiciones anteriores al problema de clasificación basada en CARs, además del conjunto I, se tiene un conjunto de clases C y un conjunto de transacciones etiquetadas T^C (conjunto de entrenamiento). Las transacciones del conjunto T^C están formadas por un conjunto de ítems X y una clase $c \in C$. Esta extensión no afecta las definiciones de Soporte y Confianza enunciadas previamente.

Definición 2.8. *Una CAR es una implicación $X \Rightarrow c$ tal que $X \subseteq I$ y $c \in C$. El Soporte de una CAR $X \Rightarrow c$ es igual a $Sop(X \cup \{c\})$ y la Confianza es igual a $\frac{Sop(X \cup \{c\})}{Sop(X)}$.*

Definición 2.9. *El tamaño de una CAR $X \Rightarrow c$ está dado por su cardinalidad,* i.e. *$\mid X \mid + 1$; una CAR de cardinalidad k se denomina k-CAR.*

Como se mencionó anteriormente, para cada nueva transacción t que se desee clasificar se selecciona el subconjunto de CARs que la cubren y con este subconjunto se determina la clase que se asigna a t. El criterio de cubrimiento utilizado en los trabajos reportados exige que todos los ítems del antecedente de la CAR estén contenidos en t (ver Def. 2.10).

Definición 2.10. *Una CAR $X \Rightarrow c$ satisface o cubre totalmente (de manera exacta) a una transacción t si $X \subseteq t$.*

Este criterio de cubrimiento será referenciado en el resto de este reporte como "cubrimiento exacto".

2.2. Medidas de calidad

Los principales algoritmos de minado de ARs utilizan el Soporte y la Confianza para evaluar la calidad de las ARs [3,18,25,26,55]. No obstante, en la literatura se han reportado diferentes medidas de calidad como alternativa al Soporte y la Confianza (*Lift*, *Conviction* y *Certainty*

Factor). En [10], los autores presentaron un análisis de estas medidas señalando las limitaciones de cada una.

La medida *Lift* (Eq. 1) mide la dependencia entre el antecedente y el consecuente de la regla. El valor 1 indica independencia y los valores mayores que 1 indican una correlación o dependencia positiva entre el antecedente y el consecuente. La medida *Lift* tiene la limitación de ser no acotada, por tanto, las diferencias de los valores de *Lift* no resultan significativas siendo difícil definir un umbral para esta medida. Además, esta medida es simétrica (Eq. 1), lo cual casi no sucede en situaciones prácticas, puesto que el hecho de que un conjunto de atributos esté presente, con cierta frecuencia, en una clase no significa que sean los únicos atributos de ésta, ni siquiera que sean los más predominantes, de modo que no siempre es posible afirmar que dicha clase implique dicho conjunto de atributos.

$$Lift(X \Rightarrow Y) = \frac{Sop(X \Rightarrow Y)}{Sop(X)Sup(Y)} = \frac{Sop(X \cup Y)}{Sop(X)Sop(Y)} = \frac{Sop(Y \Rightarrow X)}{Sop(Y)Sop(X)} = Lift(Y \Rightarrow X) \quad . \tag{1}$$

La medida *Conviction* (Eq. 2) es similar a la medida *Lift* pero mide la dependencia entre X y $\neg Y$, donde $\neg Y$ significa ausencia de Y y su Soporte se define como: $Sop(\neg Y) = 1 - Sop(Y)$. El valor 1 indica independencia mientras que los valores mayores que 1 indican una dependencia negativa entre el antecedente X y el consecuente $\neg Y$, lo que implica una dependencia positiva entre X y Y.

$$Conv(X \Rightarrow Y) = \frac{Sop(X)Sop(\neg Y)}{Sop(X \Rightarrow \neg Y)} = \frac{Sop(X)Sop(\neg Y)}{Sop(X) - Sop(X \Rightarrow Y)} \quad . \tag{2}$$

La medida de calidad *Certainty Factor* (Eq. 3) se define en dependencia de si $Conf(X \Rightarrow Y)$ es menor, mayor o igual que $Sop(Y)$. Valores negativos del *Certainty Factor* indican dependencia negativa entre el antecedente y el consecuente, valores positivos indican dependencia positiva y 0 indica independencia. Sin embargo, el valor del *Certainty Factor* de una regla $X \Rightarrow Y$ depende del $Sop(Y)$ cuando éste es cercano a $Conf(X \Rightarrow Y)$ y ambos son cercanos a 1.

$$CF(X \Rightarrow Y) = \begin{cases} \frac{Conf(X \Rightarrow Y) - Sop(Y)}{1 - Sop(Y)} & si\ Conf(X \Rightarrow Y) > Sop(Y) \quad , \\ \frac{Conf(X \Rightarrow Y) - Sop(Y)}{Sop(Y)} & si\ Conf(X \Rightarrow Y) < Sop(Y) \quad , \\ 0 & si\ Conf(X \Rightarrow Y) = Sop(Y) \quad . \end{cases} \tag{3}$$

Para una mejor comprensión, veamos el siguiente ejemplo tomado de [4]:

Ejemplo 2.1. Supongamos que $Sop(X) = 0.5$ y $Sop(Y) = 0.9$. Si $Sop(X \Rightarrow Y) = 0.45$ entonces X y Y son independientes según el Certainty Factor ya que:

$$\begin{gathered} Conf(X \Rightarrow Y) = \frac{Sop(X \Rightarrow Y)}{Sop(X)} = \frac{0.45}{0.5} = \frac{0.5 * 0.9}{0.5} = 0.9 = Sop(Y) \quad , \\ \therefore CF(X \Rightarrow Y) = 0 \quad . \end{gathered} \tag{4}$$

Si $Sop(X \Rightarrow Y) = 0.43$ entonces $CF(X \Rightarrow Y) = -0.044$ por Eq. 3, esto significa que existe una ligera dependencia negativa entre X y Y. Por otro lado, si $Sop(X \Rightarrow Y) = 0.47$ entonces $CF(X \Rightarrow Y) = 0.4$ por Eq. 3, esto muestra que X y Y tienen una alta dependencia positiva. La diferencia entre 0.43 y 0.45 es igual a la diferencia entre 0.45 y 0.47, sin embargo, el Certainty Factor obtiene valores muy diferentes.

En los algoritmos de minado de CARs, de manera similar a los algoritmos de minado de ARs, el Soporte y la Confianza han sido las medidas más utilizadas para evaluar la calidad de las CARs. No obstante, se han propuesto otras medidas de calidad que han dado buenos resultados.

En [7] y [48], los autores propusieron dos medidas denominadas *Complement Class Support* (CCS) y *Class Correlation Ratio* (CCR) que permiten obtener CARs correlacionadas positivamente, *i.e.* CARs donde exista una dependencia positiva entre el antecedente y el consecuente. El uso de estas medidas de calidad mostró buenos resultados para conjuntos de datos no balanceados, *i.e.* conjuntos de datos con desbalance en la cantidad de transacciones de cada clase. En [9] se presentó un estudio donde se evaluó el uso de las medidas *Conviction*, *Lift*, χ^2 (Chi-Cuadrado), Confianza, *Laplace*, *Mutual Information*, *Cosine*, *Jaccard* y *ϕ-Coefficient* en la construcción de clasificadores basados en CARs. Los experimentos realizados sobre 17 conjuntos de datos del repositorio UCI [8] mostraron los mejores resultados para las medidas *Conviction*, Confianza y *Laplace*.

Aunque el Soporte y la Confianza son las medidas de calidad más utilizadas en el minado de CARs, no son necesariamente las medidas ideales. Ambas medidas se han criticado por muchos autores [1,2,10,12,44,45]. La selección de un umbral de Soporte muy alto puede generar CARs que contengan conocimiento demasiado evidente y a su vez dejar de generar CARs interesantes. Por el contrario, la selección de un umbral de Soporte muy bajo puede generar un gran volumen de CARs, las cuales pueden ser redundantes o introducir ruido. Por tanto, el Soporte no es una medida de calidad apropiada para calcular las CARs y resulta difícil determinar un umbral adecuado.

De manera similar al Soporte, la Confianza presenta varias limitaciones, *e.g.* (1) no detecta independencia estadística (ver a continuación Ej. 2.2), (2) no detecta dependencias negativas lo que trae como consecuencia que se generen reglas engañosas (ver a continuación Ej. 2.3) y (3) no considera en su definición al consecuente, en nuestro caso, la clase.

Ejemplo 2.2. Considérese el conjunto de transacciones mostrado en la Tabla 2(a), donde las filas representan las transacciones y las columnas representan los ítems. La Tabla 2(b) muestra el Soporte de los conjuntos de ítems $\{i_1\}$, $\{i_2\}$ y $\{i_1, i_2\}$. Debido a que $Sop(\{i_1\})Sop(\{i_2\}) = 0.25 = Sop(\{i_1, i_2\})$, se tiene que $\{i_1\}$ y $\{i_2\}$ son estadísticamente independientes y la Confianza no permite detectar este hecho, dado que $Conf(i_1 \Rightarrow i_2) = 0.25/0.5 = 0.5 \neq 0$

Tabla 2. (a) Conjunto de transacciones D y (b) Soportes de algunos conjuntos de ítems en D.

(a)

T_{id}	i_1	i_2	i_3
t_1	1	0	0
t_2	0	0	1
t_3	1	1	1
t_4	0	1	1

(b)

Conjunto de ítems	Soporte
$\{i_1\}$	0.50
$\{i_2\}$	0.50
$\{i_1, i_2\}$	0.25

Ejemplo 2.3. Supongamos que se tiene un conjunto de datos T tal que $Sop(X) = 0.5$, $Sop(c) = 0.7$, $Sop(X \Rightarrow c) = 0.3$, $Conf(X \Rightarrow c) = 0.6$ y umbral de Confianza $minConf = 0.5$. Con estos valores, se puede pensar que $X \Rightarrow c$ es una CAR interesante pero no es así. La clase c se encuentra en el 70 % de las transacciones, por tanto, asignar siempre c es más confiable que

utilizar la CAR $X \Rightarrow c$, la cual tiene una Confianza del 60 %. *En este caso $X \Rightarrow c$ es una regla engañosa.*

Como puede apreciarse, son varias las medidas utilizadas para evaluar la calidad de las reglas, tanto en el minado de ARs como en el minado de CARs. En ambas áreas, los principales trabajos optan por utilizar el Soporte y la Confianza a pesar de las limitaciones que presentan.

2.3. Estrategias de ordenamiento

Para construir un clasificador basado en CARs, deben ordenarse las CARs generadas llevando las de mayor interés a las primeras posiciones del orden. En la literatura se han reportado cinco estrategias fundamentales de ordenamiento:

a) CSA (Confianza - Soporte - longitud del Antecedente): La estrategia de ordenamiento CSA combina la Confianza, el Soporte y la longitud del antecedente (cantidad de ítems que lo forman). CSA ordena las CARs en forma descendente de acuerdo con la Confianza. Las CARs que tengan valores iguales de Confianza se ordenan en forma descendente de acuerdo con el Soporte, y en caso de empate, se ordenan en forma ascendente de acuerdo con la longitud del antecedente [32,34].
b) ACS (longitud del Antecedente - Confianza - Soporte): La estrategia de ordenamiento ACS es una variante de la estrategia CSA, pero considera primero la longitud del antecedente, seguida de la Confianza y el Soporte [17].
c) WRA (del inglés *Weighted Relative Accuracy*): La estrategia de ordenamiento WRA, asigna a cada CAR un peso calculado en función del Soporte y la Confianza y después ordena el conjunto de CARs en forma descendente de acuerdo con los pesos asignados [17,31,52,53]. Dada una regla $X \Rightarrow c$ el valor de WRA se calcula como sigue:

$$WRA(X \Rightarrow c) = Sop(X)(Conf(X \Rightarrow c) - Sop(c)) \quad . \tag{5}$$

d) LAP (del inglés *LAPlace expected error estimate*): La estrategia de ordenamiento LAP fue introducida en [15] y posteriormente se usó en otros clasificadores [52,54]. Al igual que en la estrategia WRA, en LAP se asigna a cada CAR un peso y después se ordena en forma descendente de acuerdo con los pesos asignados. Dada una regla $X \Rightarrow c$ el valor de LAP se define como:

$$LAP(X \Rightarrow c) = \frac{Sop(X \Rightarrow c) + 1}{Sop(X) + \mid C \mid} \quad , \tag{6}$$

donde C es el conjunto predefinido de clases.
e) χ^2 (Chi-Cuadrado): La estrategia de ordenamiento χ^2 es una técnica bien conocida en estadística que se utiliza para determinar si dos variables, en nuestro caso ítems, son independientes o no. Luego de calcular el valor de χ^2 para cada CAR, también en función del Soporte y la Confianza, se ordenan las CARs en forma descendente de acuerdo con los valores de χ^2 [32].

En todas las estrategias anteriores, en caso de persistir el empate entre algunas CARs, prevalece el orden en que éstas fueron generadas. Es importante notar que estas estrategias se definen en función del Soporte y/o la Confianza, por lo que pudieran tener las mismas limitaciones de estas medidas.

2.4. Criterios de decisión

Luego de construido el clasificador, para clasificar una nueva transacción t se determina el subconjunto de CARs que la cubren de manera exacta (ver Def. 2.10) y se utiliza un criterio de decisión para asignar una clase a t. En la literatura se reportan tres criterios de decisión:

1. La Mejor Regla: En este criterio de decisión se selecciona la primera regla en el orden establecido (la mejor regla) que cubra a t y se asigna a t la clase de la regla seleccionada [34].
2. Las Mejores K Reglas: En este criterio de decisión se seleccionan, por cada clase, las primeras K reglas en el orden establecido que cubran a t, se promedian los valores de calidad de las reglas en cada clase y se asigna a t la clase para la que se obtenga mayor promedio [52].
3. Todas las Reglas: En este criterio de decisión se seleccionan, para cada clase, todas las reglas que cubran a t, se promedian los valores de calidad de las reglas en cada clase y se asigna a t la clase para la que se obtenga mayor promedio [32].

Si la estrategia de ordenamiento es CSA o ACS, para el caso "Las Mejores K Reglas" y "Todas las Reglas", se utiliza el valor de Confianza para promediar. Los tres criterios de decisión previamente mencionados tienen limitaciones que pueden afectar la eficacia del clasificador:

- Los clasificadores que siguen el criterio "La Mejor Regla" apuestan a una sola regla para clasificar y como se menciona en [17], no se puede esperar que una sola regla prediga exactamente la clase de cada transacción que ésta cubra.
- Los clasificadores que siguen el criterio "Las Mejores K Reglas" pueden verse afectados si: 1) hay desbalance en el número de reglas con altos valores de la medida de calidad, por cada clase, que cubren a la transacción que se desee clasificar, o 2) si la mayoría de las mejores K reglas se obtuvieron a partir del mismo ítem, lo cual trae consigo cierta redundancia.
- Los clasificadores que siguen el criterio "Todas las Reglas" pueden incluir reglas de baja calidad para clasificar [54].

3. Algoritmo CAR-CA para calcular un conjunto de CARs

Como se reporta en [27], varios han sido los algoritmos de minado de ARs (Apriori, Fp-growth, Eclat y Apriori-TFP) que se han modificado para calcular CARs. Debido a que en este reporte se utiliza una medida de calidad diferente a la utilizada por estos algoritmos (ver Subsecc. 4.5) y se introduce una nueva estrategia para podar el espacio de búsqueda de las CARs (ver Subsecc. 3.1), también diferente a la estrategia de poda empleada por estos algoritmos, es necesario proponer un nuevo algoritmo para calcular las CARs que además de utilizar la nueva medida de calidad e implementar la nueva estrategia de poda, sea competitivo con respecto a la eficiencia y el consumo de memoria con los algoritmos previamente mencionados.

En esta sección se introduce el algoritmo CAR-CA (del inglés *Class Association Rules using Compressed Arrays*), desarrollado para calcular un conjunto de CARs a partir de un conjunto de entrenamiento. CAR-CA es una modificación del algoritmo CA [26] de minado de ARs. De acuerdo con los experimentos mostrados en [26], CA es más eficiente[3] que los algoritmos Apriori (utilizado en CBA), Fp-growth (utilizado en CMAR), Eclat (utilizado en MCAR) y Apriori-TFP

[3] En [26] se realizaron experimentos con conjuntos de datos grandes, densos y dispersos; adicionalmente se hicieron pruebas de escalamiento del algoritmo CA.

(utilizado en TFPC). Es importante resaltar que en la construcción de clasificadores basados en CARs el mayor costo computacional lo tiene el algoritmo utilizado para calcular las CARs. El costo de calcular las CARs es igual al costo de calcular los conjuntos frecuentes de ítems (las clases se consideran como un ítem más) pues las CARs se obtienen a partir de los conjuntos frecuentes de ítems de forma inmediata, *e.g.* el conjunto $\{i_1, i_2, c_1\}$ da lugar a la CAR $\{i_1, i_2\} \Rightarrow c_1$.

CAR-CA aprovecha las ventajas del algoritmo CA respecto al cálculo de los Soportes de los conjuntos de ítems e introduce una estrategia de poda que permite obtener CARs específicas (ver Def. 2.5) con altos valores de la medida de calidad.

En las subsecciones siguientes se exponen las características del algoritmo CAR-CA (ver Subsecc. 3.1) y se describe el algoritmo (ver Subsecc. 3.2).

3.1. Características del algoritmo CAR-CA

Un factor determinante para lograr un cálculo eficiente del Soporte de las CARs es la forma en que se representen los datos. Conceptualmente, un conjunto de transacciones es una matriz de dos dimensiones donde las filas representan las transacciones y las columnas representan los ítems. En la literatura se han reportado cuatro formas de representar esta matriz [43]:

- Lista Horizontal de Ítems (LHI): Se construye por cada transacción una lista con los ítems presentes en ella.
- Vector Horizontal de Ítems (VHI): Se construye por cada transacción un vector binario donde un 1 en la posición i-ésima denota la presencia del i-ésimo ítem en la transacción y un 0 denota la ausencia.
- Lista Vertical de Identificadores de Transacciones (LVTid): Se construye por cada ítem una lista con los identificadores de las transacciones (Tids) donde aparece el mismo.
- Vector Vertical de Identificadores de Transacciones (VVTid): Se construye por cada ítem un vector binario donde un 1 en la posición i-ésima denota la presencia del ítem en la i-ésima transacción y un 0 denota la ausencia.

Con el objetivo de aprovechar las ventajas de las operaciones *bit-a-bit*, en CAR-CA se utiliza una representación VVTid considerando cada clase como un ítem más. Para ello, se representa el conjunto de datos como una matriz binaria de m x n, donde m es el número de transacciones y n es el número de ítems (incluyendo las clases). Cada vector binario asociado a un ítem j se comprime y representa como un arreglo de enteros I_j de la siguiente forma:

$$I_j = \{W_{1,j}, W_{2,j}, \ldots, W_{q,j}\},\ q = \lceil m/32 \rceil \quad , \tag{7}$$

donde cada entero del arreglo I_j representa 32 transacciones (en una arquitectura de 32 *bits*).

En la Figura 3 se muestra un ejemplo de la representación utilizada para el caso de una arquitectura de 32 *bits*. Este ejemplo contiene 64 transacciones, cuatro ítems y dos clases. En la Sección a) se muestra un conjunto de transacciones, cada una formada por un conjunto de ítems y una clase. En la Sección b) se representa la presencia de un ítem en una transacción con un 1 y la ausencia con un 0. Finalmente, en la Sección c), se toman las secuencias de 32 valores consecutivos de 1s y 0s, verticalmente por cada ítem, y se convierten en el número entero correspondiente a la secuencia de *bits* con esos valores. Por ejemplo, las primeras 32 transacciones del ítem 1 se convierten en el entero $W_{1,1}$, las siguientes 32 transacciones se convierten en el entero $W_{2,1}$, etc.

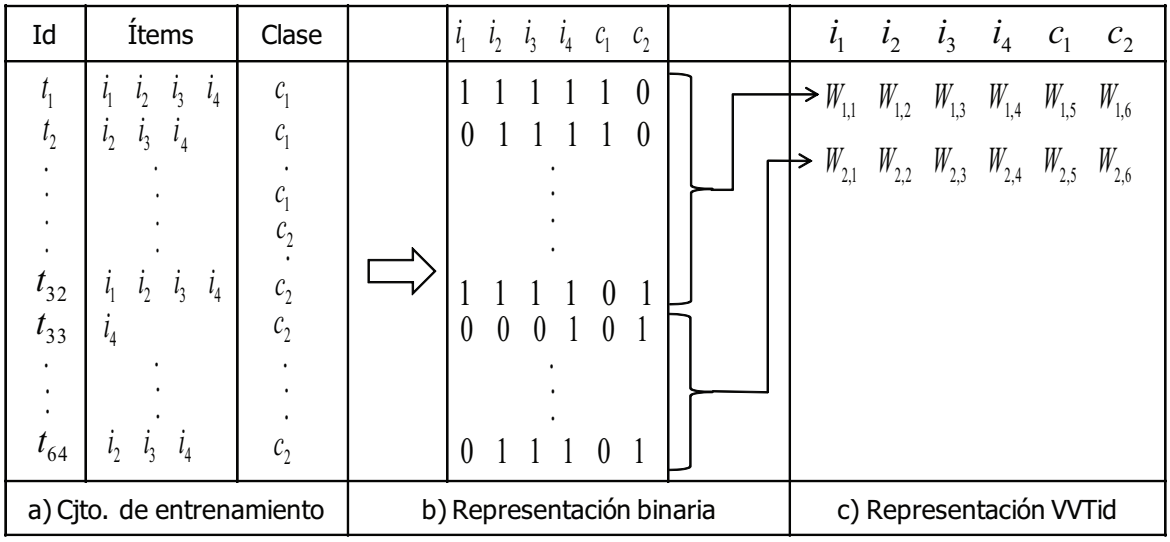


Fig. 3. Ejemplo de la representación de un conjunto con 64 transacciones en una arquitectura de 32 *bits*.

Otro factor importante, en el proceso de generación de las CARs, es la estrategia que se utilice para recorrer el retículo o espacio de búsqueda (ver Fig. 1 en la Subsecc. 2.1). Al igual que en [55], CAR-CA divide el espacio de búsqueda en clases de equivalencia. En [55], para calcular los conjuntos frecuentes de ítems, los autores proponen agrupar en una misma clase de equivalencia a los conjuntos de ítems de tamaño k que coincidan en los primeros $(k-1)$ ítems (prefijo de longitud $(k-1)$). En CAR-CA, para estructurar el espacio de búsqueda, proponemos la siguiente relación de equivalencia:

Definición 3.1. *Sea U el conjunto de todas las CARs, diremos que $r_1 \in U$ está relacionada con $r_2 \in U$ ($r_1 \, R \, r_2$) si se cumple que:*

- *r_1 y r_2 tienen el mismo tamaño (k).*
- *r_1 y r_2 tienen el mismo consecuente (la misma clase).*
- *r_1 y r_2 coinciden en los primeros $k-2$ ítems del antecedente, el cual tiene $k-1$ ítems.*

Es trivial comprobar que la relación propuesta es reflexiva, simétrica y transitiva, por lo que es una relación de equivalencia. Las relaciones de equivalencia definen clases de equivalencia. En la Figura 4 se muestra gráficamente la estructuración en clases de equivalencia definida por la relación de equivalencia propuesta.

En general, la clase de equivalencia de un elemento a, según una relación R sobre un conjunto Q, es el subconjunto de todos los elementos $q \in Q$ tales que $q \, R \, a$. El conjunto de todas las clases de equivalencia definidas sobre Q según la relación R, forma una partición de ese conjunto, es decir, son un conjunto de subconjuntos disjuntos y la unión de todos ellos es el conjunto Q. Por tanto, al estructurar el espacio de búsqueda de CARs en clases de equivalencia, se pueden calcular por separado las CARs agrupadas en cada clase de equivalencia.

Como se vio en la Subsección 2.1, del retículo formado por las CARs se deriva una estructura arbórea donde cada clase es raíz de un árbol y determina un subespacio de búsqueda asociado a ella. Además, para cada clase c, el subárbol cuyo nodo raíz tiene asociada la CAR con el

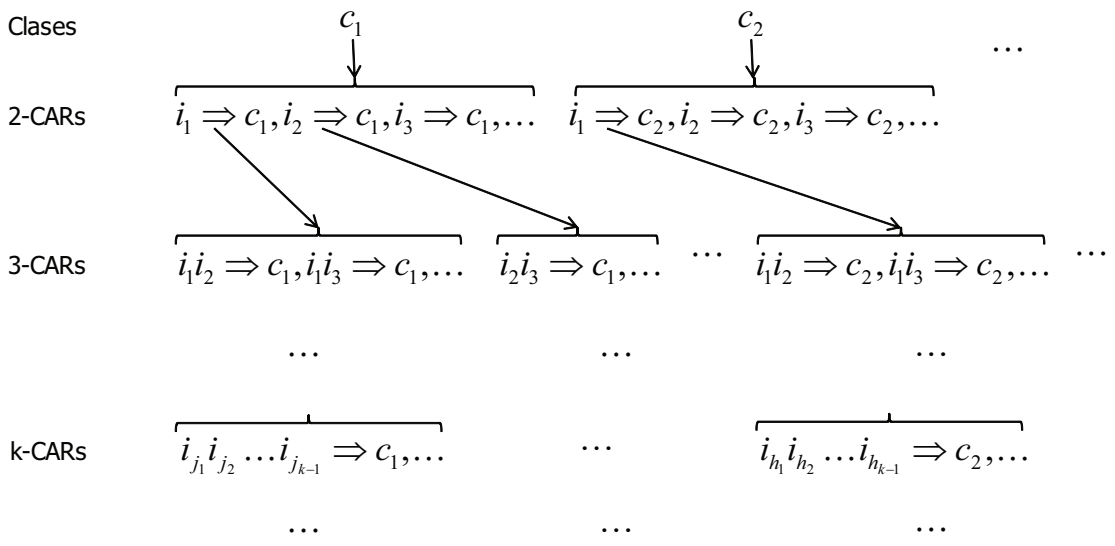


Fig. 4. Espacio de búsqueda de CARs estructurado en clases de equivalencia.

antecedente formado solo por el primer ítem abarca la mitad del espacio de búsqueda derivado de c; el subárbol cuyo nodo raíz tiene asociada la CAR con el antecedente formado solo por el segundo ítem abarca la cuarta parte del espacio de búsqueda derivado de c, y así sucesivamente. Lo anterior implica que: si el primer ítem i tiene un Soporte muy alto, entonces la cantidad de CARs candidatas que contengan a i será muy grande y por consiguiente, la cantidad de clases de equivalencia generadas en los próximos niveles, también será grande. Para eliminar ese problema el algoritmo CAR-CA ordena los ítems frecuentes de tamaño 1 en orden ascendente de su Soporte, de esta forma cada subárbol minimiza su altura y el algoritmo procesa más rápido cada clase de equivalencia.

Independientemente de la estrategia seguida para recorrer el espacio de búsqueda, pueden generarse millones de CARs [3,28,55]. Para afrontar este problema, los trabajos recientes [19,51,52,53] podan el espacio de búsqueda cada vez que encuentran una CAR que satisface el umbral de la medida de calidad (ACC[4]) que utilizan. Esta estrategia de poda tiene las siguientes limitaciones:

- Muchas ramas o caminos del espacio de búsqueda pueden ser recorridas en vano debido a que nunca se satisfaga el umbral de ACC.
- Si se obtiene una CAR $X \Rightarrow c$ entonces se poda y no se buscan más CARs, por tanto, no pueden obtenerse CARs de la forma $X' \Rightarrow c$ con $X \subset X'$, de esta forma no se consideran CARs específicas (grandes) con mayor valor de ACC, las cuales podrían ser útiles para clasificar.

En el algoritmo CAR-CA, se introduce una nueva estrategia de poda que permite obtener CARs específicas con altos valores de ACC. Esta estrategia de poda se aplica en dos situaciones diferentes:

1. Si una CAR candidata $X \Rightarrow c$ no satisface el umbral de ACC entonces no se sigue extendiendo, de esta forma se evita recorrer en vano algunas partes del espacio de búsqueda, *i.e.* se poda el

[4] Los clasificadores desarrollados, basados en CARs, utilizan medidas de calidad basadas en el soporte. Sin embargo, los resultados obtenidos en este reporte pueden aplicarse para cualquier medida de calidad.

espacio de búsqueda evitando generar CARs candidatas a partir de CARs que no satisfacen el umbral de ACC.
2. Si una CAR candidata $X \Rightarrow c$ satisface el umbral de ACC, entonces se continúa extendiendo, mientras $ACC(X \cup \{i\} \Rightarrow c)$ sea mayor o igual que $ACC(X \Rightarrow c)$, de esta forma se permite obtener CARs más específicas con altos valores de ACC.

Como se verá en la Sección 4, utilizar CARs específicas con altos valores de ACC tiene un impacto positivo en la eficacia de los clasificadores. Adicionalmente, la combinación de la representación VVTid del conjunto de datos y la estructuración del espacio de búsqueda en clases de equivalencia reduce el consumo de memoria y acelera el cálculo del Soporte durante el cálculo de las CARs.

3.2. Algoritmo CAR-CA

Para calcular el conjunto de CARs, el algoritmo CAR-CA sigue una estrategia de recorrido en amplitud por cada clase de equivalencia. CAR-CA genera iterativamente una lista L_{EC_k} que representa las clases de equivalencia que agrupan CARs de tamaño k (k-CARs), los elementos de L_{EC_k} tienen el siguiente formato:

$$\langle c, AntPref_{k-2}, IA_{AntPref_{k-2}}, AntSuff \rangle \quad , \tag{8}$$

donde:

- c es el consecuente de las CARs agrupadas.
- $AntPref_{k-2}$ es el $(k-2)$-itemset común a los antecedentes de las clases agrupadas (prefijo del antecedente).
- $AntSuff$ es el conjunto de los ítems j que pueden extender al prefijo $AntPref_{k-2}$ (sufijos del antecedente), donde j es lexicográficamente mayor que todos los ítems de $AntPref_{k-2}$.
- $IA_{AntPref_{k-2}}$ es un arreglo de pares $(value, id)$, con $value > 0$ y $1 \leq id \leq q$ ($q = \lceil m/32 \rceil$, m es la cantidad de transacciones), que se construye mediante intersecciones de los arreglos I_j, con $j \in AntPref_{k-2}$, utilizando operaciones de AND.

Los arreglos IA almacenan el Soporte del prefijo del antecedente de cada clase de equivalencia EC_k, el cual se utiliza para calcular el Soporte del antecedente de cada CAR en EC_k. Si k es grande, el número de elementos de IA es pequeño porque las operaciones AND generan muchos ceros, los cuales no se almacenan porque no influyen en el cómputo del Soporte. El procedimiento para obtener IA es el siguiente: Sean i y j dos ítems, se tiene que:

$$IA_{\{i\}\cup\{j\}} = \{(W_{k,i} \,\&\, W_{k,j}, k) \mid (W_{k,i} \,\&\, W_{k,j}) > 0, k \in [1, q]\} \quad , \tag{9}$$

análogamente, sean X un conjunto de ítems y j un ítem, se tiene que:

$$IA_{X\cup\{j\}} = \{(b \,\&\, W_{k,j}, k) \mid (b, k) \in IA_X, (b \,\&\, W_{k,j}) > 0, k \in [1, q]\} \quad , \tag{10}$$

Para calcular el Soporte de un conjunto de ítems X asociado con un arreglo de enteros IA_X, se utiliza la expresión (11):

$$Sop(X) = \sum_{(b,k)\in IA_X} BitCount(b) \quad , \tag{11}$$

donde $BitCount(b)$ es una función que calcula la cantidad de *bits* iguales a 1 que tiene la representación binaria del entero b.

Para ilustrar el proceso de minado de CARs seguido por el algoritmo CAR-CA, supongamos que se tiene la clase de equivalencia $E = \langle c_1, \{i_1, i_2\}, IA_{i_1 i_2}, \{i_3, i_4, \ldots\}\rangle$, la cual agrupa CARs de tamaño 4 ($E \in EC_4$); las CARs $\{i_1, i_2, i_3\} \Rightarrow c_1$ y $\{i_1, i_2, i_4\} \Rightarrow c_1$ son ejemplos de CARs agrupadas en E. Para mayor claridad, se asume una arquitectura de 4 *bits* y se muestran en la Figura 5 los arreglos de enteros I_{c_1}, I_{i_3} y I_{i_4} para 16 transacciones (cuatro bloques de cuatro transacciones cada uno). Adicionalmente, se muestran los pares $(value, id)$ del arreglo de enteros $IA_{i_1 i_2}$ que almacena el Soporte del prefijo del antecedente $\{i_1, i_2\}$ de la clase de equivalencia E (ver $IA_{i_1 i_2}$ en la Fig. 5).

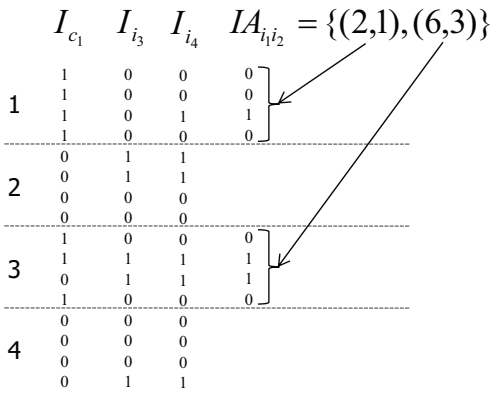
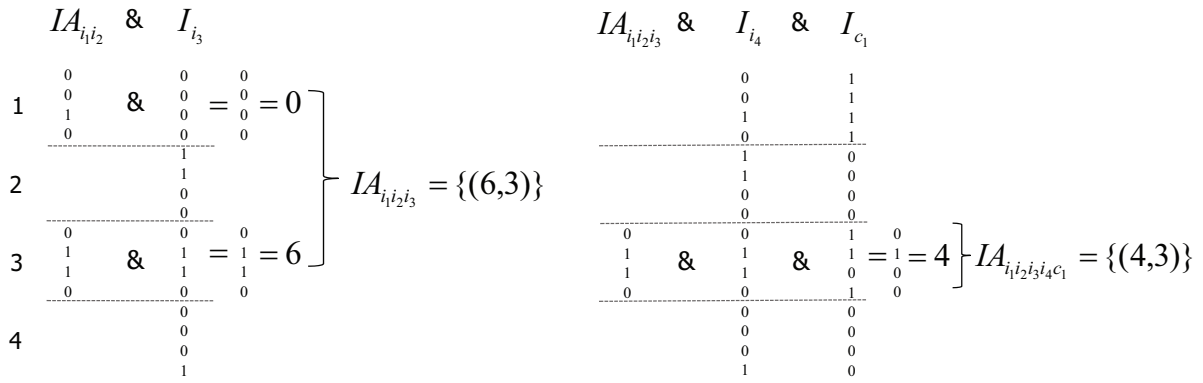


Fig. 5. Representación binaria en una arquitectura de 4 *bits*.



(a) Calculando $IA_{i_1 i_2 i_3}$. (b) Calculando $IA_{i_1 i_2 i_3 i_4 c_1}$.

Fig. 6. Obtención de las clases de equivalencia de EC_5 que se generan a partir de E.

En la Figura 6, se muestran los pasos para construir las clases de equivalencia de EC_5 que se generan a partir de la clase de equivalencia E. En el primer paso, mostrado en la Figura 6(a), se obtiene el arreglo $IA_{i_1 i_2 i_3}$ intersectando el arreglo I_{i_3} con los valores almacenados en $IA_{i_1 i_2}$, solo se intersectan los bloques con igual *id*. El arreglo $IA_{i_1 i_2 i_3}$ almacena el Soporte de $\{i_1, i_2, i_3\}$, el cual puede calcularse como $BitCount(6) = 2$ (Ec. 11). En el segundo paso, mostrado en la Figura 6(b), se obtiene el arreglo $IA_{i_1 i_2 i_3 i_4 c_1}$ de forma análoga y se calcula el Soporte de la CAR

$\{i_1, i_2, i_3, i_4\} \Rightarrow c_1$ como $BitCount(4) = 1$. Si $ACC(\{i_1, i_2, i_3, i_4\} \Rightarrow c_1)$ es mayor o igual que el umbral de ACC entonces se construiría la clase de equivalencia $\langle c_1, \{i_1, i_2, i_3\}, IA_{i_1 i_2 i_3}, \{i_4, \ldots\}\rangle$.

El uso de arreglos de enteros pudiera parecer poco eficiente respecto al consumo de memoria, sin embargo, es importante hacer notar que el número de elementos de los arreglos de enteros decrece rápidamente debido a la gran cantidad de ceros que generan las operaciones AND y como se mencionó, los ceros no son almacenados por nuestro algoritmo.

Algorithm 1: CAR-CA

Input: Conjunto de entrenamiento en representación VVTid
Output: Conjunto de CARs

```
1  Answer = ∅
2  C = {Conjunto de clases predefinidas}
3  L = {1-itemsets}
4  forall c ∈ C do
5      ECGen(⟨{c}, ∅, NULL, {L}⟩, L_EC2 = ∅)
6      Answer = Answer ∪ L_EC2
7      k = 3
8      L_ECk = ∅, L_ECk+1 = ∅, ...
9      while L_ECk−1 ≠ ∅ do
10         forall ec ∈ L_ECk−1 do
11             ECGen(ec, L_ECk)        // ec está en formato < ... >
12         end
13         Answer = Answer ∪ L_ECk
14         k = k + 1
15     end
16 end
17 return Answer
```

En el algoritmo 1 se muestra el pseudocódigo de CAR-CA. En la línea 3 del algoritmo 1 se calculan los 1-itemsets. En la línea 5, se construyen, para cada clase c, las clases de equivalencia de tamaño 2. En las líneas $7-15$, se procesa cada clase de equivalencia de tamaño 2 utilizando la función *ECGen*. La función *ECGen* recibe como argumento el umbral de ACC (α), una clase de equivalencia de tamaño $k-1$ y el conjunto de clases de equivalencia de tamaño k (*ecSet*); como resultado, *ECGen* actualiza el conjunto de clases de equivalencia *ecSet* (ver Alg. 2). Las clases de equivalencia generadas por *ECGen* solo contienen CARs con ACC mayor o igual que α.

Como se describió en la Subsección 3.1, en la literatura se han reportado cuatro formas de representar la matriz de datos, dos horizontales (*LHI* y *VHI*) y dos verticales (*LVTid* y *VVTid*). Todos los autores coinciden en las ventajas del almacenamiento vertical sobre el horizontal, esta ventaja se debe principalmente a que el almacenamiento vertical permite calcular los Soportes de los conjuntos de ítems mediante intersecciones de listas o arreglos de enteros, según sea el caso.

Tomar una decisión, referente al consumo de memoria, entre las representaciones *LVTid* y *VVTid* no es una tarea trivial. En [14], los autores analizaron estos dos formatos y concluyeron experimentalmente, que la eficiencia en memoria depende de la densidad de los datos. Particularmente, en arquitecturas de 32 *bits* el formato *LVTid* resulta más costoso en términos de memoria si el Soporte de los conjuntos de ítems es mayor que 1/32 o cercano al 3 % del total de transacciones. En este formato, se necesita un entero para representar la presencia de un ítem contra un simple *bit* en el formato *VVTid*.

En el algoritmo *CAR-CA* se necesita, por cada clase de equivalencia, un par de enteros por cada 32 transacciones, uno para el valor del bloque de 32 transacciones y el otro para el identificador

Algorithm 2: ECGen

Input: Un umbral α
Una EC en formato $\langle c, AntPref, IA_{AntPref}, AntSuff\rangle$
Un conjunto de clases de equivalencia $ecSet$
Output: El conjunto actualizado de clases de equivalencia $ecSet$

```
1  forall i ∈ AntSuff do
2      AntPref' = AntPref ∪ {i}
3      IA_AntPref' = IA_AntPref ∪ {i} ∪ {c}
4      AntSuff' = ∅
5      forall (i' ∈ AntSuff) ∧ (i' mayor lexicográficamente que i) do
6          V = ACC(AntPref' ⇒ c)
7          V' = ACC({AntPref' ∪ {i'}} ⇒ c)
8          if V' > α y V' >= V then
9              AntSuff' = AntSuff' ∪ {i'}
10         end
11     end
12     if AntSuff' ≠ ∅ then
13         ecSet = ecSet ∪ {⟨c, AntPref', IA_AntPref', AntSuff'⟩}
14     end
15 end
16 return ecSet
```

del bloque (en el arreglo IA solo se almacenan los enteros mayores que cero). El consumo de memoria de este formato es mayor que el del formato $VVTid$ si la cantidad de bloques diferentes de 0 es mayor que el 50 % del total de bloques.

Considerando un conjunto de datos con m transacciones en una arquitectura de 32 bits y un umbral de Soporte igual a $minSop$, el formato $VVTid$ requiere $\lceil m/8\rceil$ bytes de memoria mientras que *CAR-CA*, en el peor de los casos cuando el conjunto de datos es disperso, requiere $8{*}\text{mín}\{m{*}minSop, \lceil m/32\rceil\}$ bytes. A medida que el umbral de Soporte $minSop$ sea más pequeño, el algoritmo *CAR-CA* será más eficiente con respecto al consumo de memoria.

4. Clasificadores propuestos

Como se mencionó en la Subsección 1.1, un clasificador basado en CARs está compuesto por un conjunto ordenado de CARs y un criterio de decisión. Al momento de clasificar una nueva transacción t, se determina el conjunto de CARs que cubren a t y se utiliza el criterio de decisión para asignar una clase a t.

En esta sección se introducen dos clasificadores basados en CARs; el primero, denominado CAR-IC (ver Subsecc. 4.4), utiliza el Soporte y la Confianza como medidas de calidad para calcular las CARs, el segundo, denominado CAR-NF (ver Subsecc. 4.5), utiliza para calcular las CARs el Netconf. En las primeras subsecciones de la presente sección se introducen las estrategias de ordenamiento (ver Subsecc. 4.1) y los criterios de cubrimiento y decisión (ver Subsecc. 4.2 y 4.3) utilizados por ambos clasificadores.

4.1. Estrategia propuesta de ordenamiento de CARs

Una vez calculadas las CARs, antes de proceder a la etapa de clasificación, estas reglas se deben ordenar. En este reporte se introduce una estrategia de ordenamiento que es consecuente con la estrategia de poda propuesta en la Subsección 3.1, la cual favorece a las CARs específicas con altos valores de ACC.

La idea intuitiva detrás de esta estrategia de ordenamiento es que las CARs más específicas deben ser preferidas antes que las CARs más generales, ya que las primeras contienen más ítems de la transacción que se desea clasificar. En caso de empate, deben ser preferidas las CARs con mayor valor de ACC (las más interesantes). Por ejemplo, en la Tabla 3(a), se tiene un clasificador compuesto por tres CARs que están ordenadas, considerando primero, las CARs más generales. Dada la transacción $\{i_1, i_2, i_3, i_4, i_5, i_6\}$ y utilizando el criterio de decisión de "La Mejor Regla", se clasificaría la transacción como perteneciente a la clase c_1. No obstante, intuitivamente, las clases c_2 y c_3 tienen mayor probabilidad de ser la clase correcta ya que las CARs $\{i_1, i_2, i_3\} \Rightarrow c_2$ y $\{i_4, i_5, i_6\} \Rightarrow c_3$ contienen tres de los seis ítems de la transacción, mientras que la CAR $\{i_1\} \Rightarrow c_1$ solo contiene al ítem i_1.

Tabla 3. Ejemplo de estrategias de ordenamiento.

(a) CARs más generales primero

#	CAR	ACC
1	$\{i_1\} \Rightarrow c_1$	0.75
2	$\{i_4, i_5, i_6\} \Rightarrow c_3$	0.75
3	$\{i_1, i_2, i_3\} \Rightarrow c_2$	0.70

(b) CARs más específicas primero

#	CAR	ACC
1	$\{i_4, i_5, i_6\} \Rightarrow c_3$	0.75
2	$\{i_1, i_2, i_3\} \Rightarrow c_2$	0.70
3	$\{i_1\} \Rightarrow c_1$	0.75

En la Tabla 3(b), se muestran las mismas tres CARs ordenadas, considerando primero, las CARs más específicas y en caso de empate, considerando primero, las de mayor valor de ACC. En este caso, dada la transacción $\{i_1, i_2, i_3, i_4, i_5, i_6\}$, se asignaría la clase c_3 porque la CAR $\{i_4, i_5, i_6\} \Rightarrow c_3$ tiene el mismo tamaño que la CAR $\{i_1, i_2, i_3\} \Rightarrow c_2$, pero la primera tiene mayor valor de ACC.

La estrategia de ordenamiento propuesta consiste en ordenar el conjunto de CARs en forma descendente de acuerdo con el tamaño de las CARs (las más grandes primero) y en caso de empate, ordenar en forma descendente de acuerdo con los valores de ACC (las de mayor valor primero). De persistir el empate, se mantiene el orden en que fueron generadas las CARs.

4.2. Criterio de cubrimiento propuesto

Para clasificar una nueva transacción t, se selecciona el conjunto de CARs que cubre a t y se emplea un criterio de decisión para asignar una clase a t. El criterio de cubrimiento utilizado en los trabajos reportados en la literatura, el cual denominamos "cubrimiento exacto" (ver Def. 2.10), exige que todos los ítems del antecedente de la CAR estén contenidos en t. Considérese el conjunto de CARs de la Tabla 4, si se utilizaran estas reglas para clasificar las transacciones $\{i_2, i_3\}$ y $\{i_2, i_3, i_4\}$, entonces se asignaría la clase por defecto (o el clasificador se abstendría), ya que no se cubriría ninguna de estas transacciones, utilizando el criterio de cubrimiento exacto, por alguna CAR de la Tabla 4.

Tabla 4. Ejemplo de un conjunto de CARs.

CAR	ACC
$\{i_1\} \Rightarrow c$	0.52
$\{i_1, i_2\} \Rightarrow c$	0.52
$\{i_1, i_2, i_3\} \Rightarrow c$	0.54
$\{i_1, i_2, i_3, i_4\} \Rightarrow c$	0.56
$\{i_5\} \Rightarrow c$	0.51
$\{i_5, i_6\} \Rightarrow c$	0.53
$\{i_5, i_6, i_7\} \Rightarrow c$	0.53

Para aliviar este problema y con el objetivo de reducir el número de transacciones no cubiertas, en esta sección se propone un nuevo criterio de cubrimiento para los casos en que ninguna CAR cubre a la transacción t de manera exacta:

Definición 4.1. *Una CAR $X \Rightarrow c$, con $|X| = n \geq 2$, cubre parcialmente (de manera inexacta) a la transacción t si existe un conjunto de ítems $X' \subseteq t$ tal que $X' \subset X$ y $|X'| = n - 1$.*

Utilizando este criterio de cubrimiento parcial o inexacto, las transacciones $\{i_2, i_3\}$ y $\{i_2, i_3, i_4\}$ son cubiertas por las CARs $\{i_1, i_2, i_3\} \Rightarrow c$ y $\{i_1, i_2, i_3, i_4\} \Rightarrow c$, respectivamente. Este criterio de cubrimiento permite reducir el número de transacciones no cubiertas, lo cual puede repercutir directamente en la eficacia del clasificador.

Es importante resaltar que debido a la estrategia seguida por el algoritmo CAR-CA (ver Secc. 3) para generar las CARs, el hecho de que ninguna regla cubra a la transacción $\{i_2, i_3\}$ de manera exacta significa que a partir de los ítems $\{i_2\}$ y $\{i_3\}$ no se generó ninguna CAR interesante, pues en ese caso las reglas $\{i_2\} \Rightarrow c$ y $\{i_3\} \Rightarrow c$ ($c \in C$) cubrirían a la transacción $\{i_2, i_3\}$ de manera exacta. Siguiendo este análisis y dado que ninguna regla cubre a t de manera exacta, para que una regla r cubra t de manera inexacta todos los ítems de r con excepción del primero tienen que pertenecer a t. Como se puede observar, la Definición 4.1 se planteó de forma general y no sólo considerando la exclusión del primer ítem, lo cual la hace útil cuando se utilicen otras estrategias para generar las CARs diferentes a la utilizada por el algoritmo CAR-CA.

Por otro lado, en la Definición 4.1 se exige que al menos $n-1$ ítems del antecedente de la regla (de un total de n ítems) pertenezcan a la transacción que se desea clasificar. Hacer más flexible esta exigencia, *i.e.* permitir el cubrimiento inexacto si menos de $n-1$ ítems del antecedente de la regla pertenecen a la transacción, pudiera traer como consecuencia que la mayoría de las CARs de tamaño mayor o igual a $k+1$ (suponiendo que se permitió el cubrimiento inexacto con sólo $n-k$ ítems) cubran a la transacción, siendo el resultado de la clasificación similar al que se obtiene si no se considera cubrimiento alguno y se seleccionan siempre, por cada clase, todas las reglas de tamaño mayor o igual que $k+1$. Este análisis se comprobó experimentalmente para $k = 2$ y $k = 3$.

4.3. Criterio de decisión propuesto

Como se mencionó en la Subsección 2.4, se han reportado tres criterios de decisión para asignar una clase al momento de clasificar. Estos criterios de decisión tienen limitaciones que pueden afectar la eficacia del clasificador (ver Subsecc. 2.4).

De los tres criterios de decisión, el criterio "Las Mejores K Reglas" ha sido el que mejores resultados ha brindado [52]. Sin embargo, utilizar este criterio de decisión cuando la mayoría de las mejores K reglas se obtuvieron extendiendo el mismo ítem o cuando existe desbalance en el número de CARs con altos valores de ACC, por cada clase, que cubren a la nueva transacción, puede afectar la eficacia del clasificador (ver Ej. 4.1 y 4.2 respectivamente).

Tabla 5. Ejemplo de dos conjuntos de reglas ((a) y (b)) y el resultado de ordenarlas utilizando la estrategia propuesta en la Subsección 4.1((c) y (d)).

(a)

#	CAR	ACC
1	$\{i_1\} \Rightarrow c_1$	0.91
2	$\{i_1, i_2\} \Rightarrow c_1$	0.96
3	$\{i_1, i_2, i_3\} \Rightarrow c_1$	0.96
4	$\{i_1, i_2, i_3, i_4\} \Rightarrow c_1$	0.96
5	$\{i_1, i_3\} \Rightarrow c_1$	0.92
6	$\{i_2\} \Rightarrow c_1$	0.84
7	$\{i_2, i_3\} \Rightarrow c_1$	0.84
8	$\{i_2, i_3, i_4\} \Rightarrow c_1$	0.85
9	$\{i_3\} \Rightarrow c_1$	0.81
10	$\{i_3, i_4\} \Rightarrow c_1$	0.83

(b)

#	CAR	ACC
1	$\{i_2\} \Rightarrow c_2$	0.86
2	$\{i_2, i_3\} \Rightarrow c_2$	0.87
3	$\{i_2, i_3, i_4\} \Rightarrow c_2$	0.93
4	$\{i_2, i_4\} \Rightarrow c_2$	0.85
5	$\{i_2, i_4, i_5\} \Rightarrow c_2$	0.96
6	$\{i_3\} \Rightarrow c_2$	0.88
7	$\{i_3, i_5\} \Rightarrow c_2$	0.88
8	$\{i_3, i_5, i_6\} \Rightarrow c_2$	0.90
9	$\{i_4\} \Rightarrow c_2$	0.87
10	$\{i_4, i_5\} \Rightarrow c_2$	0.89

(c)

#	CAR	ACC
1	$\{i_1, i_2, i_3, i_4\} \Rightarrow c_1$	0.96
2	$\{i_1, i_2, i_3\} \Rightarrow c_1$	0.96
3	$\{i_2, i_3, i_4\} \Rightarrow c_1$	0.85
4	$\{i_1, i_2\} \Rightarrow c_1$	0.96
5	$\{i_1, i_3\} \Rightarrow c_1$	0.92
6	$\{i_2, i_3\} \Rightarrow c_1$	0.84
7	$\{i_3, i_4\} \Rightarrow c_1$	0.83
8	$\{i_1\} \Rightarrow c_1$	0.91
9	$\{i_2\} \Rightarrow c_1$	0.84
10	$\{i_3\} \Rightarrow c_1$	0.81

(d)

#	CAR	ACC
1	$\{i_2, i_4, i_5\} \Rightarrow c_2$	0.96
2	$\{i_2, i_3, i_4\} \Rightarrow c_2$	0.93
3	$\{i_3, i_5, i_6\} \Rightarrow c_2$	0.90
4	$\{i_4, i_5\} \Rightarrow c_2$	0.89
5	$\{i_3, i_5\} \Rightarrow c_2$	0.88
6	$\{i_2, i_3\} \Rightarrow c_2$	0.87
7	$\{i_2, i_4\} \Rightarrow c_2$	0.85
8	$\{i_3\} \Rightarrow c_2$	0.88
9	$\{i_4\} \Rightarrow c_2$	0.87
10	$\{i_2\} \Rightarrow c_2$	0.86

Ejemplo 4.1. Supóngase que se tienen los dos conjuntos de CARs mostrados en las Tablas 5(a) y 5(b). Supóngase además, que se desea clasificar la transacción $\{i_1, i_2, i_3, i_4, i_5, i_6\}$ *utilizando el criterio "Las Mejores K Reglas" con* $K = 5$ *(valor comúnmente utilizado en los trabajos reportados). Primeramente, se ordenan las CARs con la estrategia de ordenamiento propuesta en la Subsección 4.1(ver Tablas 5(c) y 5(d)) y se seleccionan, para cada clase, las primeras cinco reglas del orden establecido que cubren a la transacción* $\{i_1, i_2, i_3, i_4, i_5, i_6\}$*. Los promedios de los valores de ACC de las primeras cinco reglas, delimitadas con una línea en las Tablas 5(c) y 5(d), son 0.93 y 0.91 respectivamente; por tanto, se asignaría la clase* c_1*. Sin embargo, los antecedentes de las CARs seleccionadas de la clase* c_1 *son todos subconjuntos de la regla* 1*, lo cual se debe a que casi todas estas reglas se obtuvieron a partir de extensiones de* $\{i_1\} \Rightarrow c_1$*.*

Ejemplo 4.2. Supóngase que se tienen los dos conjuntos de CARs mostrados en las Tablas 6(a) y 6(b). Supóngase además, que se desea clasificar la transacción $\{i_1, i_2, i_3, i_4\}$ *utilizando el criterio "Las Mejores K Reglas" con* $K = 5$*. Primeramente, se ordenan las CARs con la estrategia de*

Tabla 6. Ejemplo de dos conjuntos de reglas ((a) y (b)) y el resultado de ordenarlas con la estrategia propuesta en la Subsección 4.1((c) y (d)).

(a)

#	CAR	ACC
1	$\{i_1\} \Rightarrow c_1$	0.80
2	$\{i_1, i_2\} \Rightarrow c_1$	0.82
3	$\{i_1, i_2, i_3\} \Rightarrow c_1$	0.95
4	$\{i_2\} \Rightarrow c_1$	0.83
5	$\{i_2, i_3\} \Rightarrow c_1$	0.94
6	$\{i_3\} \Rightarrow c_1$	0.84
7	$\{i_3, i_4\} \Rightarrow c_1$	0.96

(b)

#	CAR	ACC
1	$\{i_1\} \Rightarrow c_2$	0.80
2	$\{i_2\} \Rightarrow c_2$	0.83
3	$\{i_2, i_3\} \Rightarrow c_2$	0.92
4	$\{i_2, i_3, i_4\} \Rightarrow c_2$	0.92
5	$\{i_3\} \Rightarrow c_2$	0.91
6	$\{i_3, i_4\} \Rightarrow c_2$	0.92
7	$\{i_4\} \Rightarrow c_2$	0.91

(c)

#	CAR	ACC
1	$\{i_1, i_2, i_3\} \Rightarrow c_1$	0.95
2	$\{i_3, i_4\} \Rightarrow c_1$	0.96
3	$\{i_2, i_3\} \Rightarrow c_1$	0.94
4	$\{i_1, i_2\} \Rightarrow c_1$	0.82
5	$\{i_3\} \Rightarrow c_1$	0.84
6	$\{i_2\} \Rightarrow c_1$	0.83
7	$\{i_1\} \Rightarrow c_1$	0.80

(d)

#	CAR	ACC
1	$\{i_2, i_3, i_4\} \Rightarrow c_2$	0.92
2	$\{i_2, i_3\} \Rightarrow c_2$	0.92
3	$\{i_3, i_4\} \Rightarrow c_2$	0.92
4	$\{i_3\} \Rightarrow c_2$	0.91
5	$\{i_4\} \Rightarrow c_2$	0.91
6	$\{i_2\} \Rightarrow c_2$	0.83
7	$\{i_1\} \Rightarrow c_2$	0.80

ordenamiento propuesta en la Subsección 4.1(ver Tablas 6(c) y 6(d)) y se seleccionan, para cada clase, las primeras 5 *reglas del orden establecido que cubren a la transacción* $\{i_1, i_2, i_3, i_4\}$*. Los promedios de los valores de ACC de las primeras* 5 *reglas, delimitadas con una línea en las Tablas 6(c) y 6(d), son 0.90 y 0.92 respectivamente; por tanto, se asignaría la clase* c_2*. Sin embargo, note que si se consideraran solo las tres primeras CARs de cada clase se obtendrían como promedio 0.95 y 0.92 respectivamente y se asignaría la clase* c_1*, esta situación se presenta porque la clase* c_2 *tiene más reglas con ACC mayor que 0.90 que la clase* c_1*. Esto sucede porque el criterio "Las mejores K Reglas" no toma en cuenta el desbalance en el número de CARs con altos valores de ACC, por cada clase, que cubren a la nueva transacción.*

Con el objetivo de resolver los problemas de los criterios de decisión existentes, en este reporte se propone un nuevo criterio de decisión, denominado DK (del inglés "Dynamic K") el cual es consecuente con las estrategias de poda y ordenamiento propuestas en las Subsecciones 3.1 y 4.1, respectivamente. La estrategia de poda propuesta da mayor importancia a las CARs específicas con altos valores de ACC mientras que la estrategia de ordenamiento, considera primero las CARs de mayor tamaño y en caso de empate, las de mayor valor de ACC.

En el criterio de decisión DK, se seleccionan, por cada clase, las CARs maximales que cubren a la transacción t que se desea clasificar, *i.e.* las CARs de mayor tamaño de cada rama del espacio de búsqueda. Esto se debe a que si la CAR $\{i_1, i_2, i_3\} \Rightarrow c$ cubre a t, entonces también cubren a t las CARs $\{i_1, i_2\} \Rightarrow c$ y $\{i_1\} \Rightarrow c$, todas generadas en la misma rama del espacio de búsqueda; pero de acuerdo con la estrategia de poda propuesta, el valor de ACC de $\{i_1, i_2, i_3\} \Rightarrow c$ es mayor o igual que el valor de ACC de las CARs $\{i_1, i_2\} \Rightarrow c$ y $\{i_1\} \Rightarrow c$. Por tanto, para evitar la redundancia y permitir la inclusión de más ítems diferentes en los antecedentes de las CARs, se seleccionan las CARs maximales que cubren a t. Por ejemplo, en las Tablas 7 y 8 se muestran las reglas de los Ejemplos 4.1 y 4.2 que cubren a las transacciones $\{i_1, i_2, i_3, i_4, i_5, i_6\}$ y $\{i_1, i_2, i_3, i_4\}$,

respectivamente; las cuáles están ordenadas con la estrategia de ordenamiento propuesta en la Subsección 4.1.

Tabla 7. Reglas maximales del Ejemplo 4.1 que cubren a la transacción $\{i_1, i_2, i_3, i_4, i_5, i_6\}$.

(a)

#	CAR	ACC
1	$\{i_1, i_2, i_3, i_4\} \Rightarrow c_1$	0.96
2	$\{i_2, i_3, i_4\} \Rightarrow c_1$	0.85
3	$\{i_1, i_3\} \Rightarrow c_1$	0.92
4	$\{i_3, i_4\} \Rightarrow c_1$	0.83

(b)

#	CAR	ACC
1	$\{i_2, i_4, i_5\} \Rightarrow c_2$	0.96
2	$\{i_2, i_3, i_4\} \Rightarrow c_2$	0.93
3	$\{i_3, i_5, i_6\} \Rightarrow c_2$	0.90
4	$\{i_4, i_5\} \Rightarrow c_2$	0.89

Tabla 8. Reglas maximales del Ejemplo 4.2 que cubren a la transacción $\{i_1, i_2, i_3, i_4\}$.

(a)

#	CAR	ACC
1	$\{i_1, i_2, i_3\} \Rightarrow c_1$	0.95
2	$\{i_3, i_4\} \Rightarrow c_1$	0.96
3	$\{i_2, i_3\} \Rightarrow c_1$	0.94

(b)

#	CAR	ACC
1	$\{i_2, i_3, i_4\} \Rightarrow c_2$	0.92
2	$\{i_3, i_4\} \Rightarrow c_2$	0.92
3	$\{i_4\} \Rightarrow c_2$	0.91
4	$\{i_1\} \Rightarrow c_2$	0.80

Luego de seleccionar las CARs maximales de cada clase que cubran a la transacción t, la idea intuitiva del nuevo criterio de decisión es asignar una clase c tal que todas las CARs seleccionadas con consecuente c, que cubran a t (supóngase que son K), tengan mayor promedio de ACC que las primeras K CARs seleccionadas de las clases restantes. De las clases que satisfagan la condición anterior se prefiere la clase que tenga menos CARs seleccionadas que cubran a t, ya que el tamaño de las CARs disminuye al aumentar K debido a que las CARs se ordenan en forma descendente con respecto al tamaño.

Teniendo en cuenta el análisis anterior se puede formalizar el criterio de decisión DK como sigue: Dado un conjunto de clases $C = \{c_1, c_2, \ldots, c_m\}$ y una nueva transacción t, para cada clase $c_j \in C$ se seleccionan, según el orden establecido, las CARs maximales que cubren a t. Sean $N = \{N_1, N_2, \ldots, N_m\}$ los subconjuntos de CARs maximales para cada clase del conjunto C, denotemos al promedio de los valores de calidad de las primeras K CARs de un subconjunto N_j como $Promedio(N_j, K)$. Para asignar una clase a t se determina el subconjunto $N_j \in N$ de menor cardinalidad tal que $\forall_{N_i \in N,\, |N_i| \geq |N_j|}[Promedio(N_j, |N_j|) \geq Promedio(N_i, |N_j|)]$ y se asigna la clase asociada a N_j. Si se aplicara el criterio de decisión DK a los Ejemplos 4.1 y 4.2 para clasificar las transacciones $\{i_1, i_2, i_3, i_4, i_5, i_6\}$ y $\{i_1, i_2, i_3, i_4\}$ respectivamente, se asignaría la clase c_2 en el primer ejemplo (el promedio de las cuatro primeras CARs de la clase c_2 es 0.92 y el promedio de las cuatro primeras CARs de la clase c_1 es 0.89) y se asignaría la clase c_1 en el segundo ejemplo (el promedio de las tres primeras CARs de la clase c_1 es 0.95 y el promedio de las tres primeras CARs de la clase c_2 es 0.92).

El criterio de decisión DK no tiene los problemas de los tres criterios existentes ya que: (1) desde un inicio selecciona las CARs maximales y de mayor valor de calidad, que cubren a la nueva transacción; de esta forma no se incluyen CARs de baja calidad para clasificar ni se seleccionan varias reglas de la misma rama del espacio de búsqueda, (2) el resultado no se afecta por el desbalance de CARs de buena calidad que cubren a la nueva transacción porque para cada

transacción a clasificar se calcula el promedio de la misma cantidad de CARs en cada clase, (3) no favorece a una clase por tener una CAR de buena calidad y otras de no tan buena calidad porque para determinar la clase, DK considera todas las CARs de buena calidad que cubren a la nueva transacción y no solo la mejor, de esta forma no cae en el error de asumir siempre que la mejor regla va a clasificar bien a todas las transacciones que cubra.

4.4. CAR-IC

En esta subsección se introduce el primer clasificador basado en CARs de los dos propuestos en este reporte; el mismo se denomina CAR-IC (del inglés *Classification based on Association Rules using Inexact Coverage*).

Como se ha mencionado en trabajos anteriores [17,32,34], los umbrales de Soporte y Confianza utilizados para calcular las CARs se deben determinar con mucho cuidado. Si se utilizan umbrales de Soporte muy pequeños (cercanos a 0) se puede generar un gran volumen de CARs, mientras que si se utilizan umbrales muy altos (cercanos a 1) se pueden dejar de generar muchas CARs interesantes. En los trabajos reportados no se fundamentan los umbrales utilizados para el Soporte y la Confianza, los mismos se definen experimentalmente. En este reporte se propone utilizar como umbral para la Confianza el mínimo valor que evita la ambigüedad al momento de clasificar. Dos CARs se consideran ambiguas si tienen el mismo antecedente implicando diferentes clases. En el caso del Soporte, se propone utilizar como umbral el valor 0.01 al igual que en los otros clasificadores basados en CARs.

Para determinar el mínimo valor de Confianza que evita la ambigüedad se introducen tres proposiciones. La Proposición 4.1 plantea que la suma de los valores de Confianza de todas las CARs con igual antecedente es 1. Luego, la Proposición 4.2 garantiza que, de todas las CARs con igual antecedente, solo una puede tener un valor de Confianza mayor que 0.5. Por último, la Proposición 4.3 garantiza que 0.5 es el mínimo valor de Confianza que evita la ambigüedad al momento de clasificar.

Proposición 4.1. *Sean X un conjunto de ítems y $C = \{c_1, c_2, \ldots, c_m\}$ el conjunto de clases predefinidas, se satisface la siguiente igualdad:*

$$\sum_{i=1}^{m} Conf(X \Rightarrow c_i) = 1 \quad . \tag{12}$$

Demostración. De la definición de Confianza de una CAR (ver Def. 2.8) se tiene que:

$$\begin{aligned} \sum_{i=1}^{m} Conf(X \Rightarrow c_i) &= \sum_{i=1}^{m} \frac{Sop(X \Rightarrow c_i)}{Sop(X)} \quad , \\ &= \frac{\sum_{i=1}^{m} Sop(X \Rightarrow c_i)}{Sop(X)} \quad . \end{aligned} \tag{13}$$

De las Definiciones 2.2 y 2.6, se tiene que:

$$\begin{aligned}\sum_{i=1}^{m} Sop(X \Rightarrow c_i) &= \sum_{i=1}^{m} Sop(X \cup \{c_i\}) \quad , \\ &= \sum_{i=1}^{m} \frac{|D_{X\cup\{c_i\}}|}{|D|} \quad , \\ &= \frac{|D_X|}{|D|} = Sop(X) \quad .\end{aligned} \tag{14}$$

Debido a que cada transacción tiene una y solo una clase se cumple que $\sum_{i=1}^{m} |D_{X\cup\{c_i\}}| = |D_X|$. Por tanto, sustituyendo (14) en (13)

$$\sum_{i=1}^{m} Conf(X \Rightarrow c_i) = \frac{Sop(X)}{Sop(X)} = 1 \quad . \tag{15}$$

□

Proposición 4.2. *Sean X un conjunto de ítems y $C = \{c_1, c_2, \ldots, c_m\}$ el conjunto de clases predefinidas, a lo sumo una CAR $X \Rightarrow c_k$ ($c_k \in C$) tiene un valor de Confianza mayor que 0.5.*

Demostración. De la Definición 2.7, se tiene que la $Conf(X \Rightarrow c)$ toma valores en el intervalo $[0, 1]$ ya que la Confianza es la probabilidad de encontrar al consecuente c en las transacciones que contienen al antecedente X. Además, de la Proposición 4.1 se tiene que la suma de los valores de Confianza de todas las CARs con igual antecedente es 1. Por tanto, no se puede tener más de una CAR con iguales antecedentes y valores de Confianza mayores que 0.5 pues en este caso la suma de sus valores de Confianza sería mayor que 1. □

Con base en la Proposición 4.2, si se selecciona un umbral de Confianza igual a 0.5 no se pueden obtener dos CARs con igual antecedente implicando clases diferentes, por tanto, no existe ambigüedad al momento de clasificar. No obstante, la Proposición 4.2 no garantiza que 0.5 es el mínimo valor de Confianza que evita la ambigüedad; debido a esto, se demuestra la siguiente proposición:

Proposición 4.3. *Sean X un conjunto de ítems, $C = \{c_1, c_2, \ldots, c_m\}$ el conjunto de clases predefinidas y D un conjunto de datos, el mínimo valor de Confianza que garantiza evitar la ambigüedad al momento de clasificar, en las CARs que tienen a X como antecedente, es 0.5.*

Demostración. Para demostrar la proposición anterior es suficiente encontrar un conjunto de datos donde cualquier valor de Confianza menor que 0.5 no evite la ambigüedad. Supongan que se tiene un conjunto de datos D con solo dos clases c_1 y c_2 tales que $|D_{\{c_1\}}| = |D_{\{c_2\}}|$ y que el conjunto de ítems X está presente en todas las transacciones de D (ver Tabla 9).

En el conjunto de datos D se cumple que las CARs $X \Rightarrow c_1$ y $X \Rightarrow c_2$ tienen valor de Confianza igual a 0.5 (ver Ec. 16), por tanto, para todo umbral de Confianza $\alpha < 0.5$ ambas CARs son candidatas a ser seleccionadas para clasificar una transacción que sea cubierta por X, existiendo ambigüedad al momento de clasificar.

$$\begin{aligned}Conf(X \Rightarrow c_1) &= Conf(X \Rightarrow c_2) \quad , \\ &= \frac{n}{2 * n} = 0.5 > \alpha \quad .\end{aligned} \tag{16}$$

Tabla 9. Conjunto de transacciones utilizado en la demostración de la Proposición 4.3.

Transacciones	Conjuntos de ítems	Clases
t_1	$\cdots X \cdots$	c_1
t_2	$\cdots X \cdots$	c_1
$\cdots$	$\cdots$	$\cdots$
t_n	$\cdots X \cdots$	c_1
t_{n+1}	$\cdots X \cdots$	c_2
t_{n+2}	$\cdots X \cdots$	c_2
$\cdots$	$\cdots$	$\cdots$
t_{2n}	$\cdots X \cdots$	c_2

□

Considerando el análisis anterior, se propone utilizar en CAR-IC un umbral de Confianza igual a 0.5 para calcular las CARs. Adicionalmente, como umbral de Soporte se propone utilizar el valor 0.01 al igual que en los otros clasificadores basados en CARs, de esta forma todos tienen el mismo espacio de CARs candidatas y la comparación es justa.

Es importante resaltar que los clasificadores propuestos en este reporte, así como todos los clasificadores basados en CARs, tienen las limitaciones de que pudieran no obtener reglas cuando ninguna supera los umbrales definidos, para alguna clase.

A continuación se describe el clasificador CAR-IC, el cual tiene dos etapas, la de entrenamiento y la de clasificación. En la etapa de entrenamiento (ver Alg. 3), CAR-IC utiliza el algoritmo CAR-CA, descrito en la Sección 3, y las medidas de calidad Soporte y Confianza para calcular el conjunto de CARs. Una vez calculadas las reglas, éstas se ordenan con la estrategia de ordenamiento propuesta en la Subsección 4.1 (algoritmo *Ordena_CARs*), es decir, las reglas se ordenan en forma descendente de acuerdo con sus tamaños y en caso de empate se ordenan en forma descendente de acuerdo con sus valores de Confianza. De persistir el empate se mantiene el orden en que se generaron las CARs.

Algorithm 3: CAR-IC (etapa de entrenamiento)

Input: conjunto de entrenamiento D
Output: conjunto ordenado de $CARs$

```
1 Answer = ∅
2 CARs = CAR-CA(D)
3 Answer = Ordena_CARs(CARs)
4 return Answer
```

Luego, en la etapa de clasificación (ver Alg. 4), para clasificar una nueva transacción t se seleccionan, por cada clase, las CARs maximales que cubren a t. Para ello el algoritmo *Maximales* selecciona del conjunto ordenado de CARs, comenzando por las de mayor tamaño, las CARs que cubren a t cuyos antecedentes no estén completamente contenidos en alguna regla maximal ya seleccionada. Es importante resaltar que, para determinar si una CAR cubre o no a la transacción t, CAR-IC utiliza primero el criterio de cubrimiento exacto y en caso de que ninguna CAR cubra a t, de manera exacta, utiliza el criterio de cubrimiento inexacto propuesto en la Subsección 4.2; adicionalmente, a diferencia de los trabajos reportados que asignan la clase mayoritaria cuando ninguna CAR cubre a la transacción t, el clasificador CAR-IC se abstiene y cuenta como mal clasificadas las transacciones que no son cubiertas por alguna CAR. A las CARs maximales

seleccionadas se les aplica el criterio de decisión DK (algoritmo *Dynamic_K*), descrito en la Subsección 4.3, para calcular un valor de K para cada clase y determinar la clase que se asignará a t de acuerdo con el promedio de los valores de Confianza de las K CARs de cada clase (algoritmo $Clasifica$).

Algorithm 4: CAR-IC (etapa de clasificación)

Input: conjunto ordenado de $CARs$, nueva transacción t
Output: clase asignada

```
1 Answer = ∅
2 Max = Maximales(t)
3 DK = Dynamic_K(Max)
4 Answer = Clasifica(DK)
5 return Answer
```

En caso de existir empate en los promedios de los valores de Confianza, se asigna la clase de mayor Soporte entre las clases involucradas en el empate, de forma similar a los otros clasificadores evaluados.

Para evaluar el clasificador CAR-IC se realizaron varios experimentos sobre 20 conjuntos de datos[5] del repositorio UCI [8]. Los atributos de estos conjuntos de datos fueron discretizados y normalizados por el autor de [16] utilizando la herramienta LUCS-KDD DN. En la Tabla 10 se muestran las características principales de estos conjuntos de datos como son: número de transacciones o instancias, número de ítems diferentes y número de clases.

Tabla 10. Conjuntos de datos utilizados en los experimentos.

BD	# instancias	# ítems	# clases
adult	48842	97	2
anneal	898	73	6
breast	699	20	2
connect4	67557	129	3
dermatology	366	49	6
ecoli	336	34	8
flare	1389	39	9
glass	214	48	7
heart	303	52	5
hepatitis	155	56	2
horseColic	368	85	2
ionosphere	351	157	2
iris	150	19	3
led7	3200	24	10
letRecog	20000	106	26
mushroom	8124	90	2
pageBlocks	5473	46	5
penDigits	10992	89	10
pima	768	38	2
waveform	5000	101	3

Para realizar los experimentos se utilizó validación cruzada con 10 particiones [30,42]; es importante aclarar que se utilizaron las mismas particiones para todos los clasificadores.

[5] Estos 20 conjuntos de datos son los conjuntos de datos comúnmente utilizados en los trabajos donde se introducen los principales clasificadores basados en CARs, reportados en la literatura.

Los experimentos se realizaron en una computadora con procesador Intel Core 2 Duo de 1.86 GHz, 1 GB DDR2 de memoria RAM y con sistema operativo Windows XP SP2. Los algoritmos propuestos en este reporte se programaron en el lenguaje de programación ANSI C. Los códigos fuente de los clasificadores CBA, CMAR, CPAR y TFPC se obtuvieron de la página web del Dr. Frans Coenen (http://www.csc.liv.ac.uk/∼frans), el código fuente del clasificador DDPMine fue proporcionado por uno de sus autores, el Dr. Hong Cheng, y en el caso del clasificador HARMONY, se utilizaron los valores de eficacia reportados en [49]. La eficacia depende del número de transacciones clasificadas correctamente y se calcula como $\frac{T_{CC}}{T_T}$, donde T_{CC} es el número de transacciones correctamente clasificadas y T_T es el total de transacciones clasificadas.

Tabla 11. Comparación de eficacia de CAR-IC y los principales clasificadores basados en CARs.

BD	CBA	CMAR	CPAR	TFPC	HARMONY	DDPMine	CAR-IC
adult	**84.21**	79.72	77.24	80.79	81.90	82.82	82.85
anneal	94.65	89.09	**94.99**	88.28	91.51	90.86	93.26
breast	**94.09**	88.84	92.95	89.98	92.42	86.53	90.46
connect4	66.67	64.83	65.15	65.83	**68.05**	67.80	57.24
dermatology	80.00	82.92	80.08	76.30	62.22	63.42	**83.93**
ecoli	**83.17**	77.01	80.59	58.53	63.60	64.25	82.16
flare	84.23	83.30	64.75	84.30	75.02	77.10	**86.45**
glass	68.30	**74.37**	64.10	64.09	49.80	53.61	71.12
heart	**57.33**	55.36	55.03	51.42	56.46	57.19	56.48
hepatitis	57.83	81.16	74.34	81.16	83.16	82.29	**84.62**
horseColic	79.24	80.06	81.57	79.06	82.53	81.07	**84.54**
ionosphere	31.64	89.61	89.76	86.05	92.03	**93.25**	86.24
iris	94.00	92.33	94.70	95.33	93.32	94.03	**97.91**
led7	66.56	72.31	71.38	68.71	**74.56**	73.98	73.02
letRecog	28.64	26.25	28.13	27.57	**76.81**	76.12	75.23
mushroom	46.73	**100.00**	98.52	99.03	99.94	**100.00**	98.54
pageBlocks	90.94	87.98	92.54	89.98	91.60	**93.24**	92.59
penDigits	87.39	82.48	80.39	81.73	96.23	**97.87**	82.78
pima	75.03	72.85	74.82	74.36	72.34	75.22	**76.01**
waveform	77.58	72.22	70.66	66.74	80.46	**83.83**	75.06
Promedio	72.41	77.63	76.58	75.46	79.20	79.72	**81.52**

En un primer experimento, mostrado en la Tabla 11, se compara la eficacia de CAR-IC con la eficacia de los principales clasificadores basados en CARs. Cada valor de la tabla es el promedio de los 10 valores de eficacia obtenidos como resultado de evaluar cada una de las 10 particiones generadas por la validación cruzada. Este experimento muestra que CAR-IC supera en el promedio de eficacia (ver última fila de la Tabla 11) a los principales clasificadores reportados basados en CARs. El clasificador CAR-IC es superior al clasificador DDPMine, el cual ocupa el segundo lugar, en 1.8 puntos porcentuales.

No obstante, el promedio de la eficacia puede resultar engañoso. Como estudio alternativo, se hizo un ordenamiento de los clasificadores por posición de acuerdo con los valores de eficacia obtenidos por cada clasificador en cada conjunto de datos. Para ello, se sustituyeron los valores de eficacia en cada fila de la Tabla 11 por un entero entre 1 y 7 de acuerdo con la posición que ocupa cada valor de eficacia entre los restantes valores de la fila. En la última fila de la Tabla 12 se muestra el resultado de promediar los valores de la posición de cada clasificador en los 20 conjuntos de datos. Se puede observar que CAR-IC obtiene los mejores resultados quedando en promedio entre las dos o tres primeras posiciones (2.75). En la segunda posición se encuentra el

Tabla 12. Ranking de posición basado en la eficacia obtenida en cada conjunto de datos.

BD	CBA	CMAR	CPAR	TFPC	HARMONY	DDPMine	CAR-IC
adult	1	6	7	5	4	3	2
anneal	2	6	1	7	4	5	3
breast	1	6	2	5	3	7	4
connect4	3	6	5	4	1	2	7
dermatology	4	2	3	5	7	6	1
ecoli	1	4	3	7	6	5	2
flare	3	4	7	2	6	5	1
glass	3	1	4	5	7	6	2
heart	1	5	6	7	4	2	3
hepatitis	7	5	6	5	2	3	1
horseColic	6	5	3	7	2	4	1
ionosphere	7	4	3	6	2	1	5
iris	5	7	3	2	6	4	1
led7	7	4	5	6	1	2	3
letRecog	4	7	5	6	1	2	3
mushroom	7	1	6	4	3	1	5
pageBlocks	5	7	3	6	4	1	2
penDigits	3	5	7	6	2	1	4
pima	3	6	4	5	7	2	1
waveform	3	5	6	7	2	1	4
Promedio	3.80	4.80	4.45	5.35	3.70	3.15	2.75

clasificador DDPMine quedando en promedio entre las tres o cuatro primeras posiciones (3.15). Resultan interesantes los casos de los clasificadores CBA y CMAR; ya que CBA es el peor en promedio de eficacia, sin embargo, es el cuarto en el promedio de posición; esto se debe a que solo en cinco de los 20 conjuntos de datos ocupa las últimas posiciones (lugares seis o siete). Por el contrario, CMAR es cuarto en el promedio de eficacia pero ocupa las últimas posiciones en ocho de los 20 conjuntos de datos, resultando quinto en el promedio de posición.

En la primera columna de la Tabla 13 se muestra la eficacia del clasificador CAR-IC al aplicar solo la estrategia de ordenamiento propuesta; los valores de la segunda columna son consecuencia de incorporar el cubrimiento inexacto y por último, los valores de la tercera columna provienen de incorporar el nuevo criterio de decisión. La última fila de la Tabla 13 refleja el aumento en el promedio de eficacia de 1.02 puntos porcentuales al incluir el cubrimiento inexacto y 0.92 al adicionar a este último el nuevo criterio de decisión.

Adicionalmente, se realizó un experimento (ver Tabla 14) donde se muestra el porcentaje de abstenciones de CAR-IC utilizando y sin utilizar el criterio de cubrimiento inexacto. Utilizar el cubrimiento inexacto disminuyó casi a la mitad la cantidad de abstenciones (pasando de 2.89 % a 1.54 %) y produjo un aumento del promedio de eficacia en 1.02 puntos porcentuales.

4.5. CAR-NF

El segundo clasificador basado en CARs propuesto en este reporte, denominado CAR-NF (del inglés *Classification based on Association Rules using Netconf measure*), introduce el uso del Netconf para calcular y ordenar el conjunto de CARs. El Netconf resuelve las limitaciones de otras medidas de calidad, mencionadas en la Subsección 2.2, propuestas para evaluar reglas de asociación y reglas de asociación de clase.

Tabla 13. Impacto de cada aporte en la eficacia de CAR-IC.

BD	CAR-IC(-CI-KD)	CAR-IC(-KD)	CAR-IC
adult	82.11	82.61	82.85
anneal	91.80	92.73	93.26
breast	84.42	90.03	90.46
connect4	56.02	56.02	57.24
dermatology	78.43	80.16	83.93
ecoli	82.06	82.06	82.16
flare	85.98	85.98	86.45
glass	68.12	68.95	71.12
heart	53.21	54.35	56.48
hepatitis	84.56	84.62	84.62
horseColic	82.47	82.47	84.54
ionosphere	84.07	86.10	86.24
iris	96.06	96.67	97.91
led7	72.71	73.02	73.02
letRecog	73.14	73.14	75.23
mushroom	98.54	98.54	98.54
pageBlocks	91.82	92.26	92.59
penDigits	77.86	81.93	82.78
pima	75.23	76.01	76.01
waveform	73.06	74.39	75.06
Promedio	79.58	80.60	81.52

Tabla 14. % de abstenciones y eficacia de CAR-IC con y sin cubrimiento inexacto.

BD	%Abst. (-CI)	Acc.	%Abst. (CI)	Acc.
adult	0.85	82.11	0.33	82.61
anneal	1.11	91.80	0.12	92.73
breast	7.63	84.42	1.59	90.03
connect4	0.83	56.02	0.80	56.02
dermatology	5.46	78.43	2.73	80.16
ecoli	1.65	82.06	1.32	82.06
flare	1.04	85.98	1.04	85.98
glass	3.63	68.12	2.08	68.95
heart	5.50	53.21	4.03	54.35
hepatitis	1.43	84.56	0.72	84.62
horseColic	2.42	82.47	1.81	82.47
ionosphere	6.65	84.07	3.80	86.10
iris	1.48	96.06	0.74	96.67
led7	2.29	72.71	1.42	73.02
letRecog	1.45	73.14	1.28	73.14
mushroom	0.25	98.54	0.22	98.54
pageBlocks	1.08	91.82	0.53	92.26
penDigits	6.40	77.86	2.14	81.93
pima	4.63	75.23	3.62	76.01
waveform	1.96	73.06	0.51	74.39
Promedio	2.89	79.58	1.54	80.60

Como se ha mencionado en las secciones anteriores, los principales clasificadores basados en CARs utilizan el Soporte y la Confianza como medidas de calidad para calcular el conjunto de CARs. No obstante, varios autores han señalado un conjunto de limitaciones del Soporte y la

Confianza [10,12,44,45]. En particular, la presencia de ítems con altos valores de Soporte pueden llevar a la obtención de CARs engañosas (ver Ej. 2.3) ya que los ítems con Soporte muy alto están presentes en muchas transacciones y pueden ser predichos por muchos conjuntos de ítems [10].

La generación de reglas engañosas se pone de manifiesto cuando se procesan conjuntos de datos correspondientes a censos poblacionales, donde muchos ítems son muy propensos a ocurrir con y sin la presencia de otros ítems (*e.g.* el conjunto de datos empleado en [12], el cual cuenta con 2166 ítems, la mayoría de los cuales tienen Soporte superior al 95 %).

En [37], los autores sugieren varias propiedades que debería cumplir una buena medida de calidad (ACC) para evaluar las reglas de asociación de clase. Estas propiedades son las siguientes:

Propiedad 4.1. Si $Sop(X \Rightarrow Y) = Sop(X)Sop(Y)$ entonces $ACC(X \Rightarrow Y) = 0$

Esta propiedad establece que toda buena medida de calidad debe reflejar la independencia estadística [10].

Propiedad 4.2. $ACC(X \Rightarrow Y)$ es monótona creciente con respecto a $Sop(X \Rightarrow Y)$ cuando el resto de los parámetros permanece constante.

La Propiedad 4.2 se puede interpretar como sigue: Supóngase un conjunto de datos D y dos reglas $X \Rightarrow Y$ y $X^{'} \Rightarrow Y^{'}$ tal que $Sop(X) = Sop(X^{'})$ y $Sop(Y) = Sop(Y^{'})$. Si la fracción de transacciones en D que contienen a $X \cup Y$ ($Sop(X \Rightarrow Y)$) es mayor que la fracción de transacciones en D que contienen $X^{'} \cup Y^{'}$ ($Sop(X^{'} \Rightarrow Y^{'})$) entonces $ACC(X \Rightarrow Y) > ACC(X^{'} \Rightarrow Y^{'})$ (lo cual significa que $X \Rightarrow Y$ es mejor que $X^{'} \Rightarrow Y^{'}$).

Propiedad 4.3. $ACC(X \Rightarrow Y)$ es monótona decreciente cuando $Sop(X)$ (o $Sop(Y)$) crece y el resto de los parámetros permanece constante.

Una ACC que satisfaga la Propiedad 4.3 evita obtener reglas engañosas porque su valor no aumenta por solo aumentar el Soporte del consecuente (o del antecedente).

Una ACC que satisfaga las Propiedades 2 y 3 tiene máximos locales cuando $Sop(X \Rightarrow Y) = Sop(X)$ o $Sop(X \Rightarrow Y) = Sop(Y)$ y tiene un máximo global cuando $Sop(X \Rightarrow Y) = Sop(X) = Sop(Y)$.

A continuación, se demuestra que la medida de calidad Confianza (ver Ec. 17), la cual ha sido utilizada en los principales clasificadores basados en CARs, no satisface simultáneamente las tres propiedades anteriores:

$$Conf(X \Rightarrow Y) = \frac{Sop(X \Rightarrow Y)}{Sop(X)} \quad . \tag{17}$$

Proposición 4.4. *La Confianza no satisface la Propiedad 4.1.*

Demostración. Es suficiente tomar como contraejemplo el Ejemplo 2.2 de la Subsección 2.2. □

Proposición 4.5. *La Confianza satisface la Propiedad 4.2*

Demostración. En la ecuación 17, si se mantiene constante el denominador, entonces la Confianza aumenta al incrementarse el valor del numerador ($Sop(X \Rightarrow Y)$). □

Proposición 4.6. *La Confianza satisface la Propiedad 4.3 para* $Sop(X)$

Demostración. En la ecuación 17, si se mantiene constante el numerador, entonces la Confianza disminuye al incrementarse el valor del denominador ($Sop(X)$).

Proposición 4.7. *La Confianza no satisface la Propiedad 4.3 para* $Sop(Y)$.

Demostración. Debido a que $Sop(Y)$ no se considera en la definición de la medida de calidad Confianza, la Propiedad 4.3 no se satisface para $Sop(Y)$. □

En resumen, la Confianza no refleja la independencia estadística (Propiedad 4.1) ni detecta dependencias negativas entre el antecedente y el consecuente, así como no considera al consecuente en su definición. Por tanto, de acuerdo con [37], la Confianza no es una buena medida de calidad para evaluar las CARs.

Como se mencionó en la Subsección 2.2, en la literatura se han reportado diferentes medidas como alternativa al Soporte y la Confianza (*Lift*, *Conviction* y *Certainty Factor*). En [10], los autores presentaron un análisis de estas medidas señalando las limitaciones de cada una (ver Secc. 2.2 para más detalles).

Posteriormente, en [4], se propuso una medida, llamada Netconf para estimar la fortaleza de las reglas de asociación. Esta medida, definida en la ecuación 18, tiene entre sus principales ventajas que detecta las reglas engañosas generadas con la Confianza. Por ejemplo, supóngase que $Sop(X) = 0.4$, $Sop(Y) = 0.8$ y $Sop(X \Rightarrow Y) = 0.3$, por tanto $Sop(\neg X) = 1 - Sop(X) = 0.6$ y $Sop(\neg X \Rightarrow Y) = Sop(Y) - Sop(X \Rightarrow Y) = 0.5$ (ver Tabla 15). Si se calcula $Conf(X \Rightarrow Y)$ se obtiene 0.75 (un alto valor de Confianza) pero Y está presente en el 80 % de las transacciones, por tanto considerar la regla $X \Rightarrow Y$ para clasificar es menos eficaz que asignar la clase aleatoriamente; claramente, $X \Rightarrow Y$ es una regla engañosa [10]. En este ejemplo, $Netconf(X \Rightarrow Y)$ = -0.083 mostrando una dependencia negativa entre el antecedente y el consecuente. Si se analiza la regla $\neg X \Rightarrow Y$ entonces $Conf(\neg X \Rightarrow Y) = 0.83 > 0.8 = Sop(Y)$, esto significa que la regla $\neg X \Rightarrow Y$ es de mejor calidad, según la Confianza, que la regla $X \Rightarrow Y$, pero dependiendo del umbral es posible que se obtengan ambas reglas, con lo cual se apoyaría que todas las transacciones se clasifiquen en Y, ya sea que tengan a X o no. Adicionalmente, el Netconf de la regla $\neg X \Rightarrow Y$ es 0.083 mostrando una dependencia positiva entre el antecedente y el consecuente. Con lo cual, usando Netconf solo la regla $\neg X \Rightarrow Y$ (que es la mejor) sería generada.

Tabla 15. Diferentes formas en que dos conjuntos de ítems pueden aparecer en un conjunto de transacciones.

Transacción		Soporte
¬X	¬Y	0.1
¬X	Y	0.5
X	Y	0.3
X	¬Y	0.1

$$Netconf(X \Rightarrow Y) = \frac{Sop(X \Rightarrow Y) - Sop(X)Sop(Y)}{Sop(X)(1 - Sop(X))} \quad . \tag{18}$$

En [4], los autores mostraron que el Netconf resuelve las limitaciones de las medidas estudiadas en [10]. Entre las principales propiedades del Netconf probaron que:

- El Netconf refleja la independencia estadística, por tanto $Netconf(X \Rightarrow Y) = 0 \Leftrightarrow Sop(X \Rightarrow Y) = Sop(X)Sop(Y)$.

- $Netconf(X \Rightarrow Y) \neq Netconf(Y \Rightarrow X)$ para $Sop(X) \neq Sop(Y)$, esto significa que el Netconf no es una medida simétrica, por lo que indica la fortaleza de la implicación en ambas direcciones.
- $Netconf(X \Rightarrow Y)$ toma valores en el intervalo [-1,1].
- Valores positivos del Netconf representan dependencias positivas, valores negativos representan dependencias negativas y el valor cero representa independencia.

No obstante, los autores no probaron que el Netconf satisface todas las Propiedades 4.1-4.3, sugeridas por [37]. Debido a esto y a que el Netconf no ha sido utilizado anteriormente en tareas de clasificación, se propone en este reporte utilizarlo para construir clasificadores basados en CARs.

Primeramente, se demuestra que el Netconf satisface las Propiedades 4.1-4.3 sugeridas en [37] y posteriormente, se muestra cuáles son los valores mínimos de Netconf para calcular el conjunto de CARs que evitan la ambigüedad al momento de clasificar.

De acuerdo con la ecuación 18 y considerando que el Soporte toma valores en el intervalo $[0, 1]$, es fácil comprobar que el Netconf satisface las Propiedades 4.1 y 4.2, y además, satisface la Propiedad 4.3 para $Sop(Y)$.

Para mostrar que el Netconf satisface completamente la Propiedad 4.3, se demuestra la siguiente proposición.

Proposición 4.8. *El Netconf satisface la Propiedad 4.3 para* $Sop(X)$.

Demostración. Sean $Sop(X \Rightarrow Y) = S_{xy}$, $Sop(Y) = S_y$ y $Sop(X) = S_x$; S_{xy} y S_y satisfacen las desigualdades $0 < S_{xy} \leq S_y < 1$ y $S_x \in (0, 1)$. Suponiendo que S_{xy} y S_y son constantes, se puede reescribir el miembro derecho de la ecuación (18) en función de S_{xy}, S_y, y S_x, como sigue:

$$f(S_x) = \frac{S_{xy} - S_y S_x}{S_x(1 - S_x)} \quad . \tag{19}$$

Si se prueba que $f'(S_x) < 0$, entonces $f(S_x)$ es estrictamente decreciente y por consiguiente, la Proposición 4.8 es verdadera. Calculando la primera derivada y reduciendo términos semejantes se tiene que:

$$f'(S_x) = \frac{-S_y S_x^2 + 2S_{xy}S_x - S_{xy}}{S_x^2(1 - S_x)^2} \quad . \tag{20}$$

Debido a que $0 < S_{xy} \leq S_y < 1$ se tiene que:

$$-S_y S_x^2 + 2S_{xy}S_x - S_{xy} \leq -S_{xy}S_x^2 + 2S_{xy}S_x - S_{xy} = -S_{xy}(S_x - 1)^2 < 0 \ ,$$

por tanto, $f'(S_x) < 0$ ya que $S_x^2(1 - S_x)^2 > 0$. □

Con la demostración de la Proposición 4.8, se ha mostrado que el Netconf satisface las Propiedades 4.1-4.3. Como se mencionó anteriormente, una medida de calidad que satisfaga la Propiedad 4.3 evita obtener reglas engañosas. Por ejemplo, si se regresa al Ejemplo 2.3 y se evalúa el Netconf para $Sop(X) = 0.5$, $Sop(Y) = 0.7$ y $Sop(X \Rightarrow Y) = 0.3$, se obtiene el valor -0.2, lo cual significa que existe una dependencia negativa entre X y Y y consecuentemente, $X \Rightarrow Y$ no es una buena regla para clasificar.

Al igual que en la Subsección 4.4, donde se determina el mínimo valor de Confianza que evita la ambigüedad al momento de clasificar, en esta sección se determina el valor correspondiente para el Netconf.

Para determinar este valor de Netconf se introducen dos proposiciones, la primera proposición garantiza que, de todas las CARs con igual antecedente, solo una puede tener un valor de Netconf mayor que 0.5. Por su parte, la segunda proposición garantiza que cuando se tienen solo dos clases, 0 es el mínimo valor de Netconf que evita la ambigüedad al momento de clasificar, mientras que si se tienen más de dos clases el mínimo valor de Netconf que evita la ambigüedad al momento de clasificar es 0.5

Proposición 4.9. *Sean X un conjunto de ítems y $C = \{c_1, c_2, \ldots, c_m\}$ el conjunto de clases predefinidas, a lo sumo una CAR $X \Rightarrow c_k$ ($c_k \in C$) tiene un valor de Netconf mayor que 0.5.*

Demostración. Supóngase que existen dos CARs $X \Rightarrow c_{k_1}$ y $X \Rightarrow c_{k_2}$ con $c_{k_1}, c_{k_2} \in C$ tales que

$$Netconf(X \Rightarrow c_{k_1}) > 0.5 \quad , Netconf(X \Rightarrow c_{k_2}) > 0.5 \quad . \tag{21}$$

sumando estas desigualdades se obtiene

$$Netconf(X \Rightarrow c_{k_1}) + Netconf(X \Rightarrow c_{k_2}) > 1 \quad . \tag{22}$$

De las ecuaciones 2.2 y 2.6, definidas en la Subsección 2.1, se tiene que $Sop(c_{k_1}) \geq Sop(X \Rightarrow c_{k_1})$, $Sop(c_{k_2}) \geq Sop(X \Rightarrow c_{k_2})$, $Sop(X) \geq Sop(X \Rightarrow c_{k_1}) + Sop(X \Rightarrow c_{k_2})$ y $Sop(X) \in [0, 1]$, por tanto se satisfacen las siguientes desigualdades:

$$\begin{array}{rl} \frac{Sop(c_{k_1}) - Sop(X \Rightarrow c_{k_1})}{1 - Sop(X)} \geq & 0 \quad , \\ \frac{Sop(c_{k_2}) - Sop(X \Rightarrow c_{k_2})}{1 - Sop(X)} \geq & 0 \quad , \\ 1 \geq & \frac{Sop(X \Rightarrow c_{k_1}) + Sop(X \Rightarrow c_{k_2})}{Sop(X)} \quad . \end{array} \tag{23}$$

Debido a que las tres desigualdades de la ecuación 23 tienen la misma dirección, se pueden sumar obteniéndose

$$1 + \frac{Sop(c_{k_1}) - Sop(X \Rightarrow c_{k_1})}{1 - Sop(X)} + \frac{Sop(c_{k_2}) - Sop(X \Rightarrow c_{k_2})}{1 - Sop(X)} \geq \frac{Sop(X \Rightarrow c_{k_1})}{Sop(X)} + \frac{Sop(X \Rightarrow c_{k_2})}{Sop(X)} \quad , \tag{24}$$

y moviendo algunos términos al lado derecho se tiene

$$1 \geq \frac{Sop(X \Rightarrow c_{k_1})}{Sop(X)} - \frac{Sop(c_{k_1}) - Sop(X \Rightarrow c_{k_1})}{1 - Sop(X)} + \frac{Sop(X \Rightarrow c_{k_2})}{Sop(X)} - \frac{Sop(c_{k_2}) - Sop(X \Rightarrow c_{k_2})}{1 - Sop(X)} \quad . \tag{25}$$

Luego, trabajando con los dos primeros términos del lado derecho de la desigualdad (25)

$$\begin{array}{l} \frac{Sop(X \Rightarrow c_{k_1})}{Sop(X)} - \frac{Sop(c_{k_1}) - Sop(X \Rightarrow c_{k_1})}{1 - Sop(X)} = \\ = \frac{Sop(X \Rightarrow c_{k_1}) - Sop(X \Rightarrow c_{k_1}) Sop(X) - Sop(c_{k_1}) Sop(X) + Sop(X \Rightarrow c_{k_1}) Sop(X)}{Sop(X)(1 - Sop(X))} \\ = \frac{Sop(X \Rightarrow c_{k_1}) - Sop(c_{k_1}) Sop(X)}{Sop(X)(1 - Sop(X))} \\ = Netconf(X \Rightarrow c_{k_1}) \qquad (ver \quad Ec. \quad 18) \end{array}$$

Análogamente, $\frac{Sop(X \Rightarrow c_{k_2})}{Sop(X)} - \frac{Sop(c_{k_2}) - Sop(X \Rightarrow c_{k_2})}{1 - Sop(X)} = Netconf(X \Rightarrow c_{k_2})$ y sustituyendo en la desigualdad (25) se obtiene

$$1 \geq Netconf(X \Rightarrow c_{k_1}) + Netconf(X \Rightarrow c_{k_2}) \quad , \tag{26}$$

lo cual contradice (22). □

De acuerdo con la Proposición 4.9, si para cada conjunto de ítems X se selecciona el umbral de Netconf 0.5, se obtiene a lo sumo una CAR con antecedente X y Netconf mayor que 0.5, de esta forma se evita la ambigüedad en el momento de clasificar. Es importante notar que un valor de Netconf mayor que 0.5 puede ser considerado como un valor alto ya que el Netconf toma valores en $[-1, 1]$, siendo la dependencia entre el antecedente y el consecuente más positiva cuando el Netconf es cercano a 1. En CAR-NF, se desea calcular tantas CARs como sea posible mientras se evite la ambigüedad en el momento de clasificar.

La Proposición 4.9 garantiza que usar 0.5 como umbral de Netconf evita la ambigüedad al momento de clasificar, sin embargo no garantiza que 0.5 es el mínimo valor de Netconf que evita la ambigüedad. Por tanto, se demostró la siguiente proposición:

Proposición 4.10. *Sean X un conjunto de ítems y $C = \{c_1, c_2, \ldots, c_m\}$ el conjunto de clases predefinidas:*

a) Si $|C| > 2$ entonces 0.5 es el mínimo valor de Netconf que evita la ambigüedad, para todo conjunto de datos, en las CARs que tienen a X como antecedente.
b) Si $|C| = 2$ entonces 0 es el mínimo valor de Netconf que evita la ambigüedad, para todo conjunto de datos, en las CARs que tienen a X como antecedente.

Demostración.

a) Para el caso en que se tienen más de dos clases, se utiliza el siguiente contraejemplo. Supóngase que se tiene un conjunto de transacciones D con m clases $\{c_1, c_2, ..., c_m\}$ tales que $|D_{\{c_1\}}| = |D_{\{c_2\}}|$ y $|D_{\{c_3\}}| = ... = |D_{\{c_m\}}| = 1$ y que el conjunto de ítems X está presente solo en las transacciones de las clases c_1 y c_2 (ver Tabla 16).

Tabla 16. Conjunto de transacciones utilizado en la demostración de la Proposición 4.10.

Transacciones	Conjuntos de ítems	Clases
t_1	$\cdots X \cdots$	c_1
t_2	$\cdots X \cdots$	c_1
$\cdots$	$\cdots$	$\cdots$
t_n	$\cdots X \cdots$	c_1
t_{n+1}	$\cdots X \cdots$	c_2
t_{n+2}	$\cdots X \cdots$	c_2
$\cdots$	$\cdots$	$\cdots$
t_{2n}	$\cdots X \cdots$	c_2
t_{2n+1}	$\cdots Y \cdots$	c_3
$\cdots$	$\cdots$	$\cdots$
t_{2n+m-2}	$\cdots Y \cdots$	c_m

En el conjunto de datos D, para todo valor de Netconf $\alpha < 0.5$ se cumple que las CARs $X \Rightarrow c_1$ y $X \Rightarrow c_2$ tienen valores de Netconf mayor que α (ver Ec. 27).

$$\begin{aligned} Netconf(X \Rightarrow c_1) &= Netconf(X \Rightarrow c_2) \\ &= \frac{\frac{n}{2n+m-2} - \frac{2n}{2n+m-2}\frac{n}{2n+m-2}}{\frac{2n}{2n+m-2}(1 - \frac{2n}{2n+m-2})} \\ &= \frac{\frac{n}{2n+m-2}(1 - \frac{2n}{2n+m-2})}{\frac{2n}{2n+m-2}(1 - \frac{2n}{2n+m-2})} = 0.5 > \alpha \quad . \end{aligned} \tag{27}$$

Luego, considerando lo planteado por la Proposición 4.9, se concluye que para más de dos clases, 0.5 es el mínimo valor de Netconf que evita la ambigüedad, para todo conjunto de datos, en las CARs que tienen a X como antecedente.

b) Para el caso en que se tienen solo dos clases, primeramente se prueba que se satisface la siguiente ecuación

$$Netconf(X \Rightarrow c_1) + Netconf(X \Rightarrow c_2) = 0 \quad . \tag{28}$$

De acuerdo con la ecuación 18,

$$\begin{aligned} \sum_{i=1}^{2} Netconf(X \Rightarrow c_i) &= \frac{\sum_{i=1}^{2} Sop(X \Rightarrow c_i) - \sum_{i=1}^{2} Sop(X)Sop(\{c_i\})}{Sop(X)(1 - Sop(X))} \\ &= \frac{\sum_{i=1}^{2} Sop(X \Rightarrow c_i) - Sop(X)\sum_{i=1}^{2} Sop(\{c_i\})}{Sop(X)(1 - Sop(X))} \quad . \end{aligned} \tag{29}$$

De la ecuación 14, definida en la Subsección 4.4, se tiene que $\sum_{i=1}^{2} Sop(X \Rightarrow c_i) = Sop(X)$ y dado que cada transacción tiene una y solo una clase se cumple que

$$\begin{aligned} Sop(\{c_1\}) + Sop(\{c_2\}) &= \frac{|D_{\{c_1\}}| + |D_{\{c_2\}}|}{|D|} \\ &= 1 \quad . \end{aligned} \tag{30}$$

Luego, sustituyendo en 29, se obtiene

$$\sum_{i=1}^{2} Netconf(X \Rightarrow c_i) = \frac{Sop(X) - Sop(X) * 1}{Sop(X)(1 - Sop(X))} = 0 \quad . \tag{31}$$

Como se tienen dos clases c_1 y c_2, los valores de Netconf de las CARs $X \Rightarrow c_1$ y $X \Rightarrow c_2$ son ambos iguales a 0 o uno es positivo y el otro negativo. Si ambos valores son iguales a 0 entonces existe independencia estadística entre X y c_1 y entre X y c_2. Por tanto, en ambos casos, si se toma el valor 0 como umbral de Netconf a lo sumo una de las CARs puede tener Netconf mayor que 0. Si se toma un umbral de Netconf $\alpha < 0$, se tienen conjuntos de transacciones como el mostrado en la Tabla 17 donde los valores de Netconf de las CARs $X \Rightarrow c_1$ y $X \Rightarrow c_2$ son iguales a 0 (ver Ec. 32) y por tanto, mayores que α.

$$Netconf(X \Rightarrow c_1) = Netconf(X \Rightarrow c_2) = \frac{\frac{n}{4n} - \frac{2n}{4n}\frac{2n}{4n}}{\frac{2n}{4n}(1 - \frac{2n}{4n})} = 0 \quad . \tag{32}$$

Luego, se puede concluir que si se tienen solo dos clases, 0 es el mínimo valor de Netconf que evita la ambigüedad, para todo conjunto de datos, en las CARs que tienen a X como antecedente.

□

Tomando en cuenta la proposición anterior, en este reporte se propone utilizar, para calcular las CARs en el clasificador CAR-NF, un umbral de Netconf igual a 0 cuando se tienen solo dos clases y un umbral de Netconf igual a 0.5 cuando existen más de dos clases.

Tabla 17. Conjunto de transacciones con solo dos clases, donde ambas CARs ($X \Rightarrow c_1$ y $X \Rightarrow c_2$) tienen Netconf igual a 0.

Transacciones	Conjuntos de ítems	Clases
t_1	$\cdots X \cdots$	c_1
$\cdots$	$\cdots$	$\cdots$
t_n	$\cdots X \cdots$	c_1
t_{n+1}	$\cdots Y \cdots$	c_1
$\cdots$	$\cdots$	$\cdots$
t_{2n}	$\cdots Y \cdots$	c_1
t_{2n+1}	$\cdots X \cdots$	c_2
$\cdots$	$\cdots$	$\cdots$
t_{3n}	$\cdots X \cdots$	c_2
t_{3n+1}	$\cdots Y \cdots$	c_2
$\cdots$	$\cdots$	$\cdots$
t_{4n}	$\cdots Y \cdots$	c_2

A continuación se describe el clasificador CAR-NF, que al igual que el clasificador CAR-IC tiene dos etapas, la de entrenamiento y la de clasificación. En la etapa de entrenamiento (ver Alg. 5), CAR-NF utiliza el algoritmo CAR-CA, descrito en la Sección 3, y el Netconf como medida de calidad para calcular el conjunto de CARs. Una vez calculadas las reglas, entonces se ordenan con la estrategia de ordenamiento propuesta en la Subsección 4.1 (algoritmo $Ordena\_CARs$), es decir, las reglas se ordenan en forma descendente de acuerdo con sus tamaños y en caso de empate se ordenan en forma descendente de acuerdo con sus valores de Netconf. De persistir el empate se mantiene el orden en que se generaron las CARs.

Algorithm 5: CAR-NF (fase de entrenamiento)

Input: conjunto de entrenamiento D
Output: conjunto ordenado de $CARs$

1 $Answer = \emptyset$
2 $CARs =$ `CAR-CA(`D`)`
3 $Answer =$ `Ordena_CARs(`$CARs$`)`
4 **return** $Answer$

Luego, en la etapa de clasificación (ver Alg. 6), para clasificar una nueva transacción t se seleccionan, por cada clase, las CARs maximales de cada rama del espacio de búsqueda que cubren a t. Para ello el algoritmo $Maximales$ selecciona del conjunto ordenado de CARs, comenzando por las de mayor tamaño, las CARs que cubren a t cuyos antecedentes no estén completamente contenidos en alguna regla maximal ya seleccionada. Es válido resaltar que para determinar si una CAR cubre o no a la transacción t, CAR-NF utiliza primero el criterio de cubrimiento exacto y en caso de que ninguna CAR cubra a t de manera exacta, entonces utiliza el criterio de cubrimiento inexacto propuesto en la Subsección 4.2; adicionalmente, al igual que el clasificador CAR-IC, el clasificador CAR-NF se abstiene y cuenta como mal clasificadas las transacciones que no son cubiertas por alguna CAR. A las CARs maximales seleccionadas se les aplica el criterio de decisión DK (algoritmo $Dynamic\_K$), descrito en la Subsección 4.3, para calcular un valor de K para cada clase y determinar la clase que se asignará a t de acuerdo con el promedio de los valores de Netconf de las K CARs de cada clase (algoritmo $Clasifica$).

Algorithm 6: CAR-NF (fase de clasificación)

Input: conjunto ordenado de $CARs$, nueva transacción t
Output: clase asignada

```
1 Answer = ∅
2 Max = Maximales(t)
3 DK = Dynamic_K(Max)
4 Answer = Clasifica(DK)
5 return Answer
```

En caso de existir empate en los promedios de los valores de Netconf, se asigna la clase de mayor Soporte entre las clases involucradas en el empate, de forma similar al clasificador CAR-IC.

Es válido resaltar que los pseudocódigos presentados en esta sección no difieren de los pseudocódigos mostrados en la Subsección 4.4 ya que las estrategias de poda y ordenamiento, así como el criterio de decisión son independientes de la medida de calidad utilizada.

Para evaluar el clasificador CAR-NF se realizaron experimentos similares a los empleados en la evaluación del clasificador CAR-IC y adicionalmente, se incluyen pruebas de significancia estadística. De igual forma, los conjuntos de datos utilizados en estos experimentos, así como la técnica de discretización/normalización empleada y las características de la computadora utilizada coinciden con lo descrito en la Subsección 4.4.

Tabla 18. Comparación de eficacia de CAR-NF y los principales clasificadores basados en CARs.

BD	CBA	CMAR	CPAR	TFPC	HARMONY	DDPMine	CAR-IC	CAR-NF
adult	84.21	79.72	77.24	80.79	81.90	82.82	82.85	**87.33**
anneal	94.65	89.09	94.99	88.28	91.51	90.86	93.26	**96.42**
breast	**94.09**	88.84	92.95	89.98	92.42	86.53	90.46	87.65
connect4	66.67	64.83	65.15	65.83	**68.05**	67.80	57.24	67.09
dermatology	80.00	82.92	80.08	76.30	62.22	63.42	**83.93**	80.39
ecoli	83.17	77.01	80.59	58.53	63.60	64.25	82.16	**86.92**
flare	84.23	83.30	64.75	84.30	75.02	77.10	86.45	**88.58**
glass	68.30	**74.37**	64.10	64.09	49.80	53.61	71.12	72.13
heart	57.33	55.36	55.03	51.42	56.46	57.19	56.48	**61.92**
hepatitis	57.83	81.16	74.34	81.16	83.16	82.29	84.62	**87.60**
horseColic	79.24	80.06	81.57	79.06	82.53	81.07	84.54	**86.41**
ionosphere	31.64	89.61	89.76	86.05	92.03	**93.25**	86.24	86.93
iris	94.00	92.33	94.70	95.33	93.32	94.03	**97.91**	97.72
led7	66.56	72.31	71.38	68.71	74.56	73.98	73.02	**78.18**
letRecog	28.64	26.25	28.13	27.57	**76.81**	76.12	75.23	75.70
mushroom	46.73	**100.00**	98.52	99.03	99.94	**100.00**	98.54	99.52
pageBlocks	90.94	87.98	92.54	89.98	91.60	93.24	92.59	**97.81**
penDigits	87.39	82.48	80.39	81.73	96.23	**97.87**	82.78	84.03
pima	75.03	72.85	74.82	74.36	72.34	75.22	76.01	**79.67**
waveform	77.58	72.22	70.66	66.74	80.46	**83.83**	75.06	79.07
Promedio	72.41	77.63	76.58	75.46	79.20	79.72	81.52	**84.05**

En un primer experimento, mostrado en la Tabla 18, se compara la eficacia de CAR-NF con la eficacia de los principales clasificadores basados en CARs y se incluyen los resultados alcanzados por el clasificador CAR-IC. Cada valor de la tabla es el promedio de los 10 valores de eficacia obtenidos como resultado de evaluar cada una de las 10 particiones. Este experimento muestra que CAR-NF supera en el promedio de eficacia a los principales clasificadores reportados basados

en CARs incluyendo entre estos al clasificador CAR-IC, también propuesto en este reporte. El clasificador CAR-NF supera al segundo lugar (CAR-IC) en más de 2.5 puntos porcentuales y además, CAR-NF queda entre los dos primeros lugares en 15 de los 20 conjuntos de datos mientras CAR-IC lo logra en siete de los 20.

En la Tabla 19, se observa que CAR-NF también obtiene los mejores resultados en el promedio de posición de acuerdo con la eficacia obtenida por cada clasificador en cada conjunto de datos, quedando en la segunda posición. El clasificador CAR-IC ocupa la segunda posición en promedio, quedando entre las tres o cuatro primeras posiciones (3.15).

Tabla 19. Ranking basado en la eficacia obtenida en cada conjunto de datos.

BD	CBA	CMAR	CPAR	TFPC	HARMONY	DDPMine	CAR-IC	CAR-NF
adult	2	7	8	6	5	4	3	1
anneal	3	7	2	8	5	6	4	1
breast	1	6	2	5	3	8	4	7
connect4	4	7	6	5	1	2	8	3
dermatology	5	2	4	6	8	7	1	3
ecoli	2	5	4	8	7	6	3	1
flare	4	5	8	3	7	6	2	1
glass	4	1	5	6	8	7	3	2
heart	2	6	7	8	5	3	4	1
hepatitis	8	6	7	6	3	4	2	1
horseColic	7	6	4	8	3	5	2	1
ionosphere	8	4	3	7	2	1	6	5
iris	6	8	4	3	7	5	1	2
led7	8	5	6	7	2	3	4	1
letRecog	5	8	6	7	1	2	4	3
mushroom	8	1	7	5	3	1	6	4
pageBlocks	6	8	4	7	5	2	3	1
penDigits	3	6	8	7	2	1	5	4
pima	4	7	5	6	8	3	2	1
waveform	4	6	7	8	2	1	5	3
Promedio	4.70	5.55	5.35	6.30	4.35	3.85	3.60	2.30

En la primera columna de la Tabla 20 se muestra la eficacia del clasificador CAR-NF luego de utilizar solo la estrategia de ordenamiento propuesta; los valores de la segunda columna son consecuencia de incluir el cubrimiento inexacto; la tercera columna muestra el resultado de incorporar el nuevo criterio de decisión y finalmente, la última columna muestra los valores de eficacia si se elige para clasificar la clase mayoritaria en vez de abstenerse[6], cuando no hay CARs que cubran a la nueva transacción. La tercera fila de la Tabla 20 refleja un aumento de 0.92 puntos porcentuales en el promedio de eficacia del clasificador CAR-NF, al incluir el cubrimiento inexacto y un aumento de 2.40 al adicionar el criterio de decisión.

En experimento similar al realizado con el clasificador CAR-IC, en la Tabla 21 se muestra el por ciento de abstenciones utilizando y sin utilizar el criterio de cubrimiento inexacto, así como los respectivos impactos en la eficacia del clasificador. Utilizar el cubrimiento inexacto disminuyó casi a la mitad la cantidad de abstenciones (pasando de 3.23 % a 1.97 %) y produjo un aumento del promedio de eficacia en 0.92 puntos porcentuales.

[6] Es válido aclarar que la última columna de la tabla 20 se adiciona para mostrar cuánto pudiera beneficiar el uso de la clase mayoritaria, en vez de abstenerse como se prefiere en este reporte técnico.

Tabla 20. Impacto de cada aporte en la eficacia de CAR-NF.

BD	CAR-NF(-CI-KD)	CAR-NF(-KD)	CAR-NF	CAR-NF(clase mayoritaria)
adult	83.42	84.50	87.33	88.84
anneal	93.43	95.38	96.42	96.56
breast	85.26	85.43	87.65	88.24
connect4	62.18	62.18	67.09	67.96
dermatology	78.78	79.66	80.39	80.44
ecoli	82.36	84.01	86.92	86.95
flare	86.31	86.45	88.58	88.67
glass	67.89	68.92	72.13	72.21
heart	56.79	57.34	61.92	61.99
hepatitis	85.87	87.02	87.60	88.50
horseColic	83.25	83.56	86.41	86.98
ionosphere	84.34	86.02	86.93	87.65
iris	96.67	96.67	97.72	98.02
led7	74.53	75.88	78.18	78.21
letRecog	71.14	73.42	75.70	75.77
mushroom	99.52	99.52	99.52	99.73
pageBlocks	92.44	94.93	97.81	98.03
penDigits	78.04	78.32	84.03	84.18
pima	77.65	78.53	79.67	80.23
waveform	74.68	75.22	79.07	79.61
Promedio	80.73	81.65	84.05	84.44

Tabla 21. % de abstenciones y eficacia de CAR-NF con y sin cubrimiento inexacto.

BD	%Abst. (-CI)	Acc.	%Abst. (CI)	Acc.
adult	1.55	83.42	0.44	84.50
anneal	3.46	93.43	0.99	95.38
breast	6.84	85.26	5.56	85.43
connect4	0.86	62.18	0.85	62.18
dermatology	5.16	78.78	3.34	79.66
ecoli	1.98	82.36	0.33	84.01
flare	1.20	86.31	1.04	86.45
glass	4.15	67.89	2.60	68.92
heart	5.13	56.79	3.67	57.34
hepatitis	2.87	85.87	0.72	87.02
horseColic	2.72	83.25	1.81	83.56
ionosphere	5.70	84.34	3.80	86.02
iris	1.48	96.67	0.74	96.67
led7	2.19	74.53	0.49	75.88
letRecog	3.01	71.14	0.56	73.42
mushroom	0.27	99.52	0.23	99.52
pageBlocks	3.19	92.44	2.50	94.93
penDigits	6.19	78.04	5.70	78.32
pima	4.63	77.65	3.04	78.53
waveform	1.96	74.68	0.96	75.22
Promedio	3.23	80.73	1.97	81.65

En los trabajos donde se propusieron los clasificadores CBA, CMAR, CPAR entre otros, no se describen las técnicas de discretización/normalización utilizada. Debido a esto, decidimos mostrar una comparación de los valores de eficacia de los clasificadores propuestos en este reporte

(CAR-IC y CAR-NF) con los mejores valores de eficacia reportados para los demás clasificadores, independientemente de la técnica de discretización/normalización utilizada por estos.

Tabla 22. Mejores valores reportados por cada clasificador, independientemente de la técnica de discretización/normalización utilizada.

BD	CBA-R	CMAR-R	CPAR-R	TFPC-R	Harmony-R	DDPMine-R	CAR-IC	CAR-NF
adult	84.20	80.10	76.70	80.80	81.90	82.82	82.85	87.33
anneal	97.90	97.30	98.40	88.30	91.51	90.86	93.26	96.42
ecoli	83.17	77.01	80.59	58.53	63.60	64.25	82.16	86.92
flare	84.20	84.30	64.75	84.30	75.02	77.10	86.45	88.58
glass	73.90	70.10	74.40	64.50	49.80	53.61	71.12	72.13
heart	81.90	82.20	82.60	51.40	56.46	57.19	56.48	61.92
hepatitis	81.80	80.50	79.40	81.20	83.16	82.29	84.62	87.60
horseColic	82.10	82.60	84.20	79.10	82.53	81.07	84.54	86.41
iris	94.70	94.00	94.70	95.30	93.32	94.03	97.91	97.72
led7	71.90	72.50	73.60	57.3	74.56	73.98	73.02	78.18
letRecog	28.64	25.50	28.13	26.40	76.81	76.12	75.23	75.70
mushroom	46.70	100.00	98.52	99.00	99.94	100.00	98.54	99.52
pageBlocks	90.90	90.00	92.54	90.00	91.60	93.24	92.59	97.81
pima	72.90	75.10	73.80	74.40	72.34	75.22	76.01	79.67
waveform	80.00	83.20	80.90	74.40	80.46	83.83	75.06	79.07
Promedio	76.99	79.63	78.88	73.66	78.20	79.04	81.99	85.00

En la Tabla 22 se puede observar que los clasificadores propuestos superan al resto de los clasificadores evaluados independientemente de la técnica de discretización/normalización utilizada. El clasificador CAR-IC, segundo lugar, supera al tercer lugar, CMAR, por más de dos puntos porcentuales; y el clasificador CAR-NF supera al clasificador CAR-IC por más de tres puntos porcentuales. En este experimento se utilizaron solo 15 de los 20 conjuntos de datos porque para los otros cinco no se encontraron valores reportados en la literatura revisada.

Para determinar si las diferencias de eficacia obtenidas son significantes estadísticamente, se realizaron pruebas de t-Student con una sola cola y 95 % de certeza [21]. Cada celda de la Tabla 23 muestra el número de veces que el clasificador de la fila gana/pierde con respecto al clasificador de la columna en los 20 conjuntos de datos. Se considera empate cuando el resultado de la prueba t-Student es mayor que el 5 % (0.05). Una información detallada acerca de este test, así como una implementación del mismo se pueden encontrar en el sitio http://faculty.vassar.edu/lowry/webtext.html.

Tabla 23. Comparación dos a dos de los clasificadores evaluados. Cada celda muestra el número de veces que el clasificador de la fila gana/pierde con respecto al clasificador de la columna en los 20 conjuntos de datos.

	CBA	CMAR	CPAR	TFPC	Harmony	DDPMine	CAR-IC	CAR-NF
CBA		11/7	8/5	10/5	8/8	7/9	6/12	2/16
CMAR	7/11		8/8	8/5	5/12	5/12	4/14	5/15
CPAR	5/8	8/8		10/3	6/11	5/11	4/14	2/17
TFPC	5/10	5/8	3/10		5/14	4/13	1/16	2/16
HARMONY	8/8	12/5	11/6	14/5		2/7	8/10	6/13
DDPMine	9/7	12/5	11/5	13/4	7/2		6/9	5/13
CAR-IC	12/6	14/4	14/4	16/1	10/8	9/6		2/16
CAR-NF	16/2	15/5	17/2	16/2	13/6	13/5	16/2	

Las pruebas de significancia estadísticas muestran los mejores resultados para el clasificador CAR-NF, el cual siempre ganó en al menos 13 de los 20 conjuntos de datos. En segundo lugar quedó el clasificador CAR-IC, que superó a los demás clasificadores con la excepción del clasificador CAR-NF.

5. Conclusiones y trabajo futuro

El desarrollo de clasificadores basados en CARs continúa siendo objeto de interés debido a sus diferentes aplicaciones y fundamentalmente, debido a su interpretabilidad, característica que podría permitir a los especialistas modificar las reglas con base en su experiencia y así mejorar los resultados.

Los clasificadores basados en CARs, previos a este reporte, presentan varias limitaciones que pueden afectar sus eficacias. Estas limitaciones están relacionadas con: (a) el uso, para calcular las CARs, de la medida de calidad Confianza, la cual como se mostró en la Subsección 2.2 tiene varias deficiencias; (b) el empleo de estrategias de poda y ordenamiento que prefieren las reglas generales en lugar de las reglas específicas como se mostró en la Subsección 4.1 y (c) el uso de los criterios de decisión "La Mejor Regla", "Las Mejores K Reglas" y "Todas las Reglas" que en algunos casos pueden afectar la eficacia del clasificador como se mostró en la Subsección 4.3.

En este reporte inicialmente se propuso el algoritmo CAR-CA para calcular el conjunto de CARs, el cual introduce una nueva estrategia de poda que permite obtener reglas más específicas con altos valores de la medida de calidad. Para ordenar el conjunto de reglas se propuso una estrategia de ordenamiento, que da prioridad a las CARs de mayor cantidad de ítems (más específicas) y en caso de empate da prioridad a las CARs de mayor valor de la medida de calidad. Tanto la estrategia de poda como la estrategia de ordenamiento propuestas son independientes de la medida de calidad que se utilice para calcular las reglas.

Posteriormente, se propusieron dos clasificadores basados en CARs, CAR-IC y CAR-NF, que utilizan al algoritmo CAR-CA para calcular las CARs y solucionan las deficiencias (a), (b) y (c). En ambos clasificadores se introdujeron (en la etapa de clasificación) nuevos criterios de cubrimiento y decisión. Con base en los resultados experimentales se puede concluir que el criterio de cubrimiento propuesto permite disminuir la cantidad de asignaciones de la clase mayoritaria o abstenciones, lo que impacta en la eficacia de clasificación. También con base en los resultados experimentales se puede concluir que el criterio de decisión propuesto resuelve los problemas de los tres criterios de decisión reportados en la literatura para la clasificación con CARs.

La diferencia entre ambos clasificadores radica en la medida de calidad utilizada para calcular las CARs. El clasificador CAR-IC utiliza la medida de calidad Confianza para calcular las reglas mientras que CAR-NF propone el uso del Netconf para esta tarea. El Netconf resuelve las deficiencias de la Confianza para calcular CARs. Entre las principales deficiencias resueltas se encuentran que el Netconf (1) refleja la independencia estadística, (2) considera al consecuente de la clase en su definición, (3) no es una medida simétrica y (4) no genera reglas engañosas. Tanto para la Confianza como para el Netconf, se realizó un estudio de sus valores y se determinaron umbrales adecuados para evitar la ambigüedad al momento de clasificar. El clasificador CAR-NF obtuvo los mejores resultados de todos los clasificadores evaluados, incluido el clasificador CAR-IC. Sin embargo, en algunos conjuntos de datos los resultados de CAR-IC superaron a los resultados de CAR-NF, esto se debe a que pocas reglas alcanzaron el umbral de Netconf, mientras que utilizando la Confianza se obtuvieron suficientes reglas para lograr mejor clasificación.

A partir de los experimentos realizados se puede concluir que cada una de las aportaciones (nueva estrategia de ordenamiento, nuevos criterios de cubrimiento y decisión) por separado ayuda a mejorar la eficacia del clasificador y que todas las aportaciones juntas permiten superar en calidad de clasificación, con una diferencia significativa, a los otros clasificadores basados en CARS.

Luego del estudio realizado de los valores de las medidas Confianza y Netconf se puede concluir que los umbrales utilizados en los clasificadores CAR-IC (Confianza = 0.5) y CAR-NF (Netconf = 0 si se tienen solo dos clases y Netconf = 0.5 en caso contrario) evitan la ambigüedad al momento de clasificar.

Finalmente, con base en los experimentos, se puede concluir que los clasificadores CAR-IC y CAR-NF son mejores opciones para enfrentar el problema de la clasificación basada en CARs que los clasificadores basados en CARs existentes en el estado del arte.

Los clasificadores propuestos en este reporte utilizan reglas que consideran a una sola clase en el consecuente. Sin embargo, en trabajos recientes, se ha abordado el problema de calcular reglas que consideren en el consecuente más de una clase y de esta forma asignar una lista de clases, con cierto orden, al momento de clasificar una nueva transacción.

Como trabajo futuro inmediato se pretende extender los resultados obtenidos para abordar este problema. Como primer paso se deben utilizar umbrales de Confianza o Netconf menores a los empleados en este reporte con el objetivo de obtener varias reglas con igual antecedente, las cuales se pudieran integrar posteriormente en una sola regla con varias clases, en cierto orden, en el consecuente. Como trabajo futuro a mediano o largo plazo, se pretende calcular reglas negativas para clasificar ($\{i_1\} \Rightarrow \neg c$); las reglas negativas pueden ser de gran utilidad para descartar clases o para desempatar.

Referencias bibliográficas

1. L. M. Adamo. *Data mining for association rules and sequential patterns: sequential and parallel algorithms.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
2. C. C. Aggarwal and P. S. Yu. A new framework for itemset generation. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (PODS'98)*, pages 18–24, Seattle, Washington, USA, 1998. ACM.
3. R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*, pages 487–499, Santiago, Chile, 1994.
4. K. I. Ahn and J. Y. Kim. Efficient Mining of Frequent Itemsets and a Measure of Interest for Association Rule Mining. *Information and knowledge Management*, 3(3):245–257, 2004.
5. K. Ali, S. Manganaris, and R. Srikant. Partial classification using association rules. In *Proceedings of the 3rd International Conference on Knowledge Discovery in Databases and Data Mining (KDD'97)*, pages 115–118, 1997.
6. M. Antonie, O. R. Zaiane, and A. Coman. Associative Classifiers for Medical Images. *Lecture Notes in Artificial Intelligence, Mining Multimedia and Complex Data*, 2797:680–83, 2001.
7. B. Arunasalam and S. Chawla. CCCS: a top-down associative classifier for imbalanced class distribution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'06)*, pages 517–522, New York, NY, USA, 2006. ACM.
8. A. Asuncion and D. J. Newman. UCI Machine Learning Repository. *In http://www.ics.uci.edu/∼mlearn/MLRepository.html*, 2007.
9. J. P. Azevedo and A. M. Jorge. Comparing Rule Measures for Predictive Association Rules. In *Proceedings of the 18th European conference on Machine Learning (ECML'07)*, pages 510–517, Berlin, Heidelberg, 2007. Springer-Verlag.
10. F. Berzal, I. Blanco, D. Sánchez, and M. A. Vila. Measuring the accuracy and interest of association rules: A new framework. *Intelligent Data Analysis*, 6(3):221–235, 2002.

11. F. Berzal, J. C. Cubero, D. Sánchez, and J. M. Serrano. ART: A Hybrid Classification Model. *Machine Learning*, 54(1):67–92, 2004.
12. S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic Itemset Counting and Implication Rules for Market Basket Data. *SIGMOD Rec.*, 26(2):255–264, 1997.
13. S. Buddeewong and W. Kreesuradej. A New Association Rule-Based Text Classifier Algorithm. In *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, pages 684–685, Washington, DC, USA, 2005. IEEE Computer Society.
14. D. Burdick, M. Calimlim, and J. Gehrke. MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases. In *Proceedings of the International Conference on Data Engineering (ICDE'01)*, Heidelberg, Germany, 2001.
15. P. Clark and R. Boswell. Rule Induction with CN2: Some Recent Improvements. In *Proceedings of European Working Session on Learning (ESWL'91)*, pages 151–163, Porto, Portugal, 1991.
16. F. Coenen. The LUCS-KDD discretised/normalised ARM and CARM Data Library. Department of Computer Science, The University of Liverpool, UK. *In http://www.csc.liv.ac.uk/∼frans/KDD/Software/LUCS-KDD-DN*, 2003.
17. F. Coenen and P. Leng. An Evaluation of Approaches to Classification Rule Selection. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 359–362, Washington, DC, USA, 2004.
18. F. Coenen, P. Leng, and S. Ahmed. Data Structures for Association Rule Mining: T-trees and P-trees. *IEEE Transactions on Knowledge and Data Engineering*, 16:774–778, 2004.
19. F. Coenen, P. Leng, and L. Zhang. Threshold Tuning for Improved Classification Association Rule Mining. In *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05)*, pages 216–225, 2005.
20. W. Cohen. Fast Effective Rule Induction. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML'95)*, pages 115–123. Morgan Kaufmann, 1995.
21. T. G. Dietterich. Aproximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.
22. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Willey & Sons, 1973.
23. S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management (CIKM'98)*, pages 148–155, Bethesda, Maryland, USA, 1998.
24. N. Friedman, D. Geiger, and M.Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.
25. J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In *Proceedings of the 2000 ACM-SIGMOD International Conference on Management of Data (SIGMOD'2000)*, pages 1–12, Dallas, TX, 2000.
26. R. Hernández, J. Hernández, J. A. Carrasco, and J. Fco. Martínez. Algorithms for Mining Frequent Itemsets in Static and Dynamic Datasets. *Intelligent Data Analysis*, 14(3):419–435, 2010.
27. R. Hernández, J. A. Carrasco, J. Hernández, and J. Fco. Martínez. Desarrollo de Clasificadores basados en Reglas de Asociación. *Reporte Técnico No. CCC-10-002, Coordinación de Ciencias Computacionales, INAOE*, 2010.
28. J. Holt and M. S. Chung. Multipass algorithms for mining association rules in text databases. *Knowledge and Information Systems*, 3(2):168–183, 2001.
29. K. Kianmehr and R. Alhajj. CARSVM: A class association rule-based classification framework and its application to gene expression data. *Artificial Intelligence in Medicine*, pages 7–25, 2008.
30. R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of International Joint Conference on AI*, pages 1137–1145, 1995.
31. N. Lavraĉ, P. Flach, and B. Zupan. Rule Evaluation Measures: A Unifying View. In *Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP'99)*, pages 174–185. Springer-Verlag, 1999.
32. W. Li, J. Han, and J. Pei. CMAR: accurate and efficient classification based on multiple class-association rules. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'01)*, pages 369–376, 2001.
33. R. P. Lippmann. Pattern Classification using neural networks. *IEEE Communications Magazine*, pages 47–64, 1989.
34. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 80–86, New York, NY, USA, 1998.
35. S. H. Park, J. A. Reyes, D. R. Gilbert, J. W. Kim, and S. Kim. Prediction of protein-protein interaction types using association rule based classification. *BMC Bioinformatics*, 10(1), 2009.
36. K. Perumal and R. Bhaskaran. Supervised Classification Performance of Multispectral Images. *Journal of Computing*, 2(2), 2010.

37. G. Piatetsky-Shapiro. *Discovery, Analysis, and Presentation of Strong Rules*. AAAI/MIT Press, Cambridge, MA, 1991.
38. J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
39. J. R. Quinlan and R. M. Cameron-Jones. FOIL: A midterm report. In *Proceedings of the European Conference on Machine Learning (ECML'93)*, pages 3–20. Springer-Verlag, 1993.
40. P. Rajendran and M. Madheswaran. An Improved Image Mining Technique For Brain Tumour Classification Using Efficient Classifier. *International journal of Computer Science and Information Security*, 6(3), 2009.
41. P. Rajendran and M. Madheswaran. Hybrid Medical Image Classification Using Association Rule Mining with Decision Tree Algorithm. *Journal of Computing*, 2(1), 2010.
42. S. Salzberg. On comparing classifiers: pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1(3):317–328, 1997.
43. P. Shenoy, J. Haritsa, S. Sudarshan, G. Bhalotia, M. Bawa, and D. Shah. Turbo-charging Vertical Mining of Large Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Dallas, USA, 2000.
44. C. Silverstein, S. Brin, and R. Motwani. Beyond Market Baskets: Generalizing Association Rules to Dependence Rules. *Data Mining and Knowledge Discovery*, pages 39–68, 1998.
45. M. Steinbach and V. Kumar. Generalizing the notion of confidence. *Knowledge and Information Systems*, 12(3):279–299, 2007.
46. F. A. Thabtah, P. Cowling, and Y. Peng. MMAC: A New Multi-Class, Multi-Label Associative Classification Approach. *Data Mining, IEEE International Conference on*, 0:217–224, 2004.
47. A. Veloso, W. Meira Jr., and M. J. Zaki. Lazy Associative Classification. In *Proceedings of the Sixth International Conference on Data Mining (ICDM'06)*, pages 645–654, Washington, DC, USA, 2006. IEEE Computer Society.
48. F. Verhein and S. Chawla. Using Significant, Positively Associated and Relatively Class Correlated Rules for Associative Classification of Imbalanced Datasets. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, Washington, DC, USA*, pages 679–684, Washington, DC, USA, 2007. IEEE Computer Society.
49. J. Wang and G. Karypis. On Mining Instance-Centric Classification Rules. *IEEE Transactions on Knowledge and Data Engineering*, 18(11):1497–1511, 2006.
50. W. Wang, Y. J. Wang, nares-Alcántara R. Ba Z. Cui, and F. Coenen. Application of Classification Association Rule Mining for Mammalian Mesenchymal Stem Cell Differentiation. In *Proceedings of the 9th Industrial Conference on Advances in Data Mining. Applications and Theoretical Aspects (ICDM'09)*, pages 51–61, Berlin, Heidelberg, 2009. Springer-Verlag.
51. Y. J. Wang, Q. Xin, and F. Coenen. A Novel Rule Ordering Approach in Classification Association Rule Mining. In *Proceedings of the 5th international conference on Machine Learning and Data Mining in Pattern Recognition (MLDM'07)*, pages 339–348, Leipzig, Germany, 2007.
52. Y. J. Wang, Q. Xin, and F. Coenen. A Novel Rule Weighting Approach in Classification Association Rule Mining. *Data Mining Workshops, International Conference on*, 0:271–276, 2007.
53. Y. J. Wang, Q. Xin, and F. Coenen. Hybrid Rule Ordering in Classification Association Rule Mining. *Trans. MLDM*, 1(1):1–15, 2008.
54. X. Yin and J. Han. CPAR: Classification based on Predictive Association Rules. In *Proceddings of the SIAM International Conference on Data Mining*, San Francisco, CA, 2003.
55. M. Zaky, S. Parthasarathy, M. Ogihara, and W. Li. New Algorithm for fast Discovery of Association Rules. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD'97)*, pages 283–296, Menlo, CA, 1997. AAAI Press.

RT_018, agosto 2013
Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona
Diseño de Portada: Di. Alejandro Pérez Abraham
RNPS No. 2143
ISSN 2072-6260
Indicaciones para los Autores:
Seguir la plantilla que aparece en www.cenatav.co.cu
C E N A T A V
7ma. A No. 21406 e/214 y 216, Rpto. Siboney, Playa;
La Habana. Cuba. C.P. 12200
Impreso en Cuba