Desarrollo de Clasificadores basados en Reglas de Asociación

Raudel Hernández León, Jesús A. Carrasco Ochoa,
José Hernández Palancar, J. Fco. Martínez Trinidad

Reporte Técnico No. CCC-10-002
26 de enero de 2010

Coordinación de Ciencias Computacionales
INAOE

Luis Enrique Erro 1
Sta. Ma. Tonantzintla,
72840, Puebla, México.

Desarrollo de Clasificadores basados en Reglas de Asociación

Raudel Hernández león[1,2] Jesús A. Carrasco Ochoa[1] José Hernández Palancar[2]
J. Fco. Martínez Trinidad[1]

[1]Computer Science Department
National Institute of Astrophysics, Optics and Electronics
Luis Enrique Erro # 1, Santa María Tonantzintla, Puebla, 72840, México
[2]Data Mining Departament
Advanced Techonologies Application Center
21812 e 218 y 222, Rpto. Siboney, Playa, La Habana, 12200, Cuba
E-mail: {raudel,airel,fmartine}@inaoep.mx,jpalancar@cenatav.co.cu

Abstract

La clasificación basada en reglas de asociación de clase (CARs) es un tema vigente en el área de minería de datos. Hasta el momento se han desarrollado varias estrategias que incluyen diferentes medidas de calidad para calcular y ordenar el conjunto de CARs, diferentes algoritmos de generación de CARs y diferentes criterios de decisión para asignar una clase en el momento de clasificar. Todas estas estrategias han estado dirigidas a mejorar la eficacia de los clasificadores basados en CARs, no obstante, quedan limitaciones por resolver. Esta propuesta de tesis doctoral aborda el problema de la construcción de clasificadores basados en CARs. Como resultados preliminares, se presenta un nuevo algoritmo eficiente que utiliza la medida Netconf *para calcular y ordenar las reglas. Además se propone un criterio de cubrimiento inexacto que disminuye el número de asignaciones de la clase mayoritaria, influyendo esto directamente en la eficacia del nuevo clasificador.*

1 Introducción

Hoy en día, la mayoría de la información generada se almacena para su posterior consulta y/o procesamiento. Por ejemplo, en las cajas de los supermercados se registra información sobre las compras de los clientes; la evaluación de esta información puede ayudar a idear estrategias de mercado más eficientes. Igualmente, en las redes de computadoras se pueden analizar los datos proporcionados por el protocolo TCP/IP para detectar intrusos o comportamientos anómalos. La cantidad de información almacenada por los sistemas actuales es muy grande para ser analizada manualmente. La Minería de Datos ofrece herramientas para descubrir información implícita en estos grandes conjuntos de datos. Una importante técnica de la Minería de Datos es el descubrimiento o minado de reglas de asociación (ARs por sus siglas en inglés), que permite descubrir relaciones interesantes, llamadas asociaciones, en grandes conjuntos de datos.

Desde finales de los 90 se comenzó a investigar el poder discriminativo de las ARs y éstas se utilizaron para construir clasificadores de alta eficacia [13, 4, 11, 5, 8, 6, 12, 10]. Estos clasificadores se construyen a

partir de un conjunto especial de reglas denominadas Reglas de Asociación de Clase (CARs por sus siglas en inglés). Una CAR es un caso particular de regla de asociación y está compuesta por un conjunto de elementos o ítems (antecedente) y una clase (consecuente). Un ejemplo de CAR es el siguiente:

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

lo que se interpreta cómo:

$$(\text{compro pan}) \wedge (\text{compro leche}) \wedge (\text{compro carne}) \Rightarrow \text{pago en efectivo}$$

donde "$\wedge$" es el operador lógico de conjunción. Esta regla nos dice que el antecedente formado por los ítems "pan", "leche" y "carne" implican el consecuente o clase formado por el ítem "efectivo".

La clasificación con CARs se ha aplicado en diferentes tareas como son: la reducción de fallas en las telecomunicaciones, la detección de redundancia en exámenes médicos [22], la clasificación de imágenes médicas [27], la clasificación de textos [23], la diferenciación de células madres mesenquimales en mamíferos [24] y la predicción de tipos de interacciones proteína-proteína [25].

1.1 Organización del documento

Esta propuesta de tesis está organizada como sigue: la siguiente sección presenta los conceptos básicos del minado de ARs y sus extensiones al minado de CARs, además se presenta el planteamiento del problema. En la sección 3 se describe el trabajo relacionado y se expone la motivación de esta investigación. En la sección 4 se plantea la propuesta de investigación, en la cual se describen los objetivos generales y específicos, la metodología a seguir, las contribuciones esperadas y el calendario de actividades. Los resultados preliminares se presentan en la sección 5 y finalmente, las conclusiones se dan en la sección 6.

2 Conceptos preliminares

En esta sección se presentan algunos conceptos básicos del descubrimiento de ARs y sus extensiones al descubrimiento de CARs. Estos conceptos y definiciones se utilizarán en el resto de esta propuesta de tesis doctoral.

Sea $I = \{i_1, i_2, \ldots, i_n\}$ un conjunto de n ítems y T un conjunto de transacciones. Cada transacción en 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.*

DEFINICIÓN 2.2 *El* soporte *de un conjunto de ítems X, en adelante Sop(X), se define como la fracción de transacciones en T que contienen a X. El* soporte *toma valores en el intervalo* $[0, 1]$.

DEFINICIÓN 2.3 *Sea minSup un umbral previamente establecido, un conjunto de ítems X se denomina frecuente (FI por sus siglas en inglés) si $Sop(X) \geq minSup$.*

DEFINICIÓN 2.4 *Una regla de asociación (AR por sus siglas en inglés) sobre el conjunto de transacciones T 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 $R_2 : X_2 \Rightarrow Y$, se dice que R_1 es más específica que R_2 si $X_2 \subset X_1$.*

Las medidas más usadas en la literatura para evaluar la calidad de una AR son el *soporte* y la *confianza*.

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), se define en función del* soporte *como $\frac{Sop(X \cup Y)}{Sop(X)}$. La confianza toma valores en el intervalo $[0, 1]$.*

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 lexicográfico entre los ítems del conjunto I.

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 Regla de Asociación de Clase (CAR) es una implicación $X \Rightarrow c$ tal que $X \subseteq I$ y $c \in C$. El* soporte *de una regla de asociación de clase $X \Rightarrow c$ es igual a $Sop(X \cup \{c\})$ y la* confianza *es igual a $\frac{Sop(X \cup \{c\})}{Sop(X)}$.*

DEFINICIÓN 2.9 *Una Regla de Asociación de Clase $X \Rightarrow c$ ($X \subseteq I$ y $c \in C$) satisface o cubre a una transacción $t \subseteq I$ si $X \subseteq t$.*

Los clasificadores desarrollados basados en CARs seleccionan, para cada transacción t que se desee clasificar, el subconjunto de CARs que la cubren y con este subconjunto determinan la clase que se asignará a t.

2.1 Planteamiento del problema

Sea 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 1), R un conjunto ordenado de reglas $X \Rightarrow c$ tal que $X \subseteq I$ y $c \in C$, W una función que asigna un peso a cada regla $r \in R$ y D un criterio de decisión que utiliza a R para asignar una clase a cada transacción t que se desee clasificar.

Table 1. Representación general de un conjunto de transacciones

	T_C	Ítems				Clase
Transacciones	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

Dados I, C y T_C, construir un clasificador basado en CARs consiste en calcular R, ordenar R según la función de asignación de peso W y definir el criterio de decisión D. El problema que se plantea en esta propuesta de tesis doctoral es la construcción de clasificadores basados en CARs.

3 Trabajo relacionado

Desde finales de los 90, aprovechando el poder de discriminativo de las ARs, se comienzan a integrar las técnicas de *Classification Rule Mining* (CRM) y *Association Rule Mining* (ARM)[13, 4]. La integración de ambas técnicas consiste en minar un subconjunto especial de reglas de asociación denominadas CARs (ver definición 2.8) y utilizar este subconjunto para construir clasificadores.

Los clasificadores desarrollados desde entonces se dividen en dos grupos: Los clasificadores de dos etapas y los clasificadores integrados.

- **Clasificadores de dos etapas**: Estos clasificadores, en la primera etapa calculan todas las CARs que cumplen con los umbrales de *soporte* y *confianza*[1] establecidos. En la segunda etapa se determina un subconjunto más pequeño de CARs que cubra al conjunto de entrenamiento y con éste se construye el clasificador [13, 4, 8].
- **Clasificadores integrados**: Los clasificadores integrados utilizan diferentes estrategias para generar directamente el conjunto de CARs, construyendo el clasificador en una sola etapa y evitando el costoso proceso de cubrimiento de los clasificadores de dos etapas [7, 9, 12].

Independientemente del tamaño del conjunto de entrenamiento, el número de CARs obtenido puede ser muy grande debido al comportamiento exponencial propio del minado de ARs. Por tanto, la selección y el pesado de las CARs es muy importante en la construcción de estos clasificadores. Para construir un clasificador basado en CARs y posteriormente clasificar nuevas transacciones se siguen tres etapas fundamentales:

1. Calcular el conjunto de CARs dado un conjunto de entrenamiento y un conjunto de umbrales.
2. Evaluar y ordenar el conjunto de CARs teniendo en cuenta alguna(s) medida(s) de calidad.
3. Dada una nueva transacción t y un criterio de decisión, utilizar el conjunto de CARs para asignar una clase a t.

Cualquiera sea la estrategia seguida para calcular el conjunto de CARs (en una o en dos etapas), un clasificador se compone por una lista ordenada de CARs. En la literatura se han reportado cinco criterios principales de ordenamiento de CARs:

a) CSA (*Confianza* - *Soporte* - longitud del Antecedente): El criterio de ordenamiento CSA ordena las CARs primero descendentemente por la *confianza*, en caso de empate ordena descendentemente por el *soporte* y de persistir el empate, ordena ascendentemente por la longitud del antecedente [13, 4].

b) ACS (longitud del Antecedente - *Confianza* - *Soporte*): El criterio de ordenamiento ACS es una variación del criterio de ordenamiento CSA, pero considera primero la longitud del antecedente, seguido de la *confianza* y el *soporte* [11].

c) WRA (del inglés *Weighted Relative Accuracy*): El criterio 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 orden descendente de los pesos asignados [20, 11, 9, 12]. Dada una regla $X \Rightarrow Y$ el valor de WRA se calcula como sigue:

$$WRA(X \Rightarrow Y) = Sop(X)(Conf(X \Rightarrow Y) - Sop(Y))$$

[1] Todos los clasificadores reportados, basados en CARs, usan las medidas de calidad *soporte* y *confianza* para calcular las CARs

d) LAP (del inglés *Laplace Expected Error Estimate*): El criterio de ordenamiento LAP fue introducido por Clark y Boswell [21] y posteriormente se usó en otros clasificadores [7, 9]. Dada una regla $X \Rightarrow Y$ el valor de LAP en función del *soporte* y la *confianza* se define como:

$$LAP(X \Rightarrow Y) = \frac{Sop(X \Rightarrow Y) + 1}{Sop(X) + \mid C \mid}$$

donde C es el conjunto de clases.

e) χ^2 (Chi-Cuadrado): El criterio 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 ordena descendentemente el conjuntos de CARs [4].

En todos los casos, si después de aplicar el criterio de ordenamiento existe empate entre algunas CARs, se mantiene el orden en que éstas fueron generadas. Luego de construido el clasificador, para clasificar una nueva transacción t, se determina el subconjunto de CARs que la cubren (ver definición 2.9) y se utiliza un criterio de decisión para asignar una clase a la transacción t. En los trabajos desarrollados se han reportado tres criterios de decisión para asignar la clase:

1. La Mejor Regla: Se selecciona la primera regla en el orden establecido (mejor regla) que cubra a t y se asigna a t la clase de la regla seleccionada [13].

2. Las Mejores *K* Reglas: 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 y se asigna a t la clase para la que se obtenga mayor promedio [9].

3. Todas las Reglas: Se seleccionan 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 [4].

Para los casos de "Las Mejores *K* Reglas" y "Todas las Reglas", si el criterio de ordenamiento es CSA o ACS se utiliza el valor de la *confianza* como valor de calidad de la regla. Los tres criterios de decisión previamente mencionados tienen limitaciones que pueden afectar la eficacia del clasificador:

- Los algoritmos que siguen el criterio de "La Mejor Regla" apuestan a una sola regla para clasificar y como se menciona en [11], no se puede esperar que una sola regla prediga exactamente la clase de cada transacción que ésta cubra.

- Los algoritmos que siguen el criterio de "Todas las Reglas" corren el riesgo de incluir en la clasificación reglas de bajo interés, es decir, reglas con valores poco significativos de la medida de calidad utilizada para evaluarlos [7].

- Por último, los algoritmos que siguen el criterio de "Las Mejores *K* Reglas" pueden verse afectados si hay desbalance entre el número de reglas por clase que cubran a la transacción que se desee clasificar, por ejemplo, si $k = 5$ y para una clase se tienen sólo 4 reglas que cubren a la transacción, entonces se promedian los valores de calidad de las 4 reglas pudiendo alguna(s) ser de bajo interés.

A continuación se describen los principales clasificadores basados en CARs reportados en la literatura, primero los de dos etapas y después los integrados, además se mencionan las limitaciones de cada uno.

3.1 Clasificadores de dos etapas

Los primeros clasificadores basados en CARs reportados [13, 4] se construyen en dos etapas, primero calculan todas las reglas que cumplen con los umbrales de *soporte* y *confianza* establecidos y en una segunda etapa, determinan un subconjunto más pequeño de reglas que cubra al conjunto de entrenamiento; luego con este subconjunto de reglas construyen el clasificador.

3.1.1 CBA (*Classification Based on Associations*)

El algoritmo CBA, propuesto en el 98 por Bing Liu, fue el primero en integrar las técnicas de ARM y CRM para construir clasificadores [13]. CBA propone un algoritmo llamado CBA-RG para generar el conjunto de CARs y un algoritmo llamado CBA-CB para construir el clasificador.

El algoritmo CBA-RG se basa en el algoritmo de minado de reglas de asociación Apriori [1] y calcula todas las reglas que cumplan con los umbrales de *soporte* y *confianza* establecidos. Si se encuentra más de una regla con igual antecedente CBA-RG selecciona la regla de mayor confianza y en caso de empate selecciona aleatoriamente una de las reglas involucradas.

El algoritmo CBA-CB, en un primer paso, ordena el conjunto de CARs obtenido por CBA-RG siguiendo el criterio CSA, es decir, ordena las reglas descendentemente por la *confianza*, en caso de empate ordena descendentemente por el *soporte* y de persistir el empate, ordena ascendentemente por la longitud del antecedente. Seguidamente, el algoritmo CBA-CB poda el conjunto de CARs, para ello calcula un subconjunto más pequeño de CARs que cubra al conjunto de entrenamiento y elimina el resto de las CARs.

Para clasificar una nueva transacción t, CBA utiliza el criterio de la "Mejor Regla". Esto significa que se selecciona, siguiendo el orden establecido, la mejor regla que cubra a t y asigna a t la clase asociada a la regla seleccionada. Ésta es una de las limitaciones del algoritmo pues no se puede esperar que una sola regla estime correctamente la clase de cada transacción que ésta cubra.

Los experimentos realizados en [13] muestran que, en general, CBA obtiene mejor eficacia que clasificadores basados en árboles de decisión como C4.5 [15].

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

En [4] se propuso el clasificador CMAR, que mejora en varios aspectos al clasificador CBA. Para calcular el conjunto de CARs, CMAR utiliza una extensión del algoritmo de minado de reglas de asociación Fp-growth [2], que es más eficiente que el algoritmo Apriori utilizado en CBA.

Al igual que CBA, CMAR ordena el conjunto de CARs siguiendo el criterio CSA y después calcula un subconjunto más pequeño de CARs que cubra al conjunto de entrenamiento. Adicionalmente, CMAR utiliza otras dos estrategias de poda para reducir aún más el conjunto de CARs. La primera estrategia elimina las reglas más específicas de menor confianza. Dadas dos reglas R_1 y R_2, donde R_2 es más específica que R_1 (ver definición 2.5), CMAR poda R_2 si R_1 tiene mayor *confianza* que R_2. Esta estrategia de poda elimina reglas más específicas pero que a la vez pueden tener valores de calidad más significativos que el valor de calidad de la regla seleccionada. La segunda estrategia selecciona sólo las CARs correlacionadas positivamente, para ello aplica a cada regla $X \Rightarrow c$ el *test* χ^2 y selecciona aquellas cuyos valores de χ^2 sobrepasen cierto umbral.

A diferencia de CBA, que utiliza una sola regla para clasificar, CMAR utiliza todas las reglas que cubren a la transacción t incluyendo reglas de bajo interés, lo que puede afectar la eficacia del clasificador. Si todas las reglas que cubren a t tienen asociada la misma clase, CMAR asigna esta clase a t. En caso contrario, CMAR divide el conjunto de reglas R_t que cubren a t en tantos grupos como clases diferentes haya en

R_t. Para decidir la clase CMAR estima la fortaleza de cada grupo de reglas utilizando la medida *weighted* χ^2 [26], que refleja cuán fuerte es una regla en dependencia de su soporte y la distribución de su clase, finalmente asigna a t la clase del grupo más fuerte.

Los experimentos realizados muestran que CMAR obtiene mejor eficacia que CBA utilizando los mismos conjuntos de datos y los mismos umbrales de *soporte* y *confianza* (1% y 50% respectivamente).

3.1.3 MMAC (*Multi-class, Multi-label Associative Classification*) y MCAR (*Multi-class Classification based on Association Rules*)

En [28], los autores estudiaron el problema de generar reglas de asociación que estimen múltiples clases. Tanto en CBA como en CMAR las reglas obtenidas sólo tienen una clase en el consecuente. CBA es más estricto ya que de las reglas con igual antecedente sólo selecciona la de mayor confianza, CMAR permite obtener reglas con igual antecedente pero en la clasificación asigna una sola clase.

El clasificador propuesto en [28], llamado MMAC, permite generar reglas con igual antecedente que estiman clases diferentes. MMAC calcula el conjunto de CARs que satisface los umbrales de *soporte* y *confianza* utilizando el algoritmo ECLAT de minado de ARs propuesto en [3]. Para ordenar el conjunto de CARs, MMAC utiliza una variación del criterio CSA y considera el siguiente orden: *confianza*, *soporte*, *soporte* del antecedente y longitud del antecedente. Al incluir un nuevo criterio en el ordenamiento reduce la cantidad de empates lo que implica una reducción de las asignaciones aleatorias.

Después de ordenar el conjunto de CARs, MMAC realiza un proceso que los autores llaman "Aprendizaje Recursivo" y obtienen un primer conjunto de reglas, donde cada regla tiene en el consecuente la clase de mayor *soporte*. Con las reglas restantes realizan el mismo proceso, obteniéndose un segundo conjunto de reglas y así sucesivamente hasta que no queden reglas con antecedentes iguales.

Para clasificar una nueva transacción t se selecciona la "Mejor Regla" que cubra a t y se asigna a t la lista de clases de todas las reglas que tienen el mismo antecedente que la regla seleccionada, ordenada decrecientemente por el *soporte* de las clases. Para compararse con CBA se tuvo en cuenta la eficacia considerando sólo la regla con mayor *soporte*. La eficacia obtenida por MMAC supera en la mayoría de los conjuntos de datos la eficacia obtenida por CBA.

Otro clasificador, denominado MCAR, fue presentado en [8] por los mismos autores. La diferencia entre ambos trabajos radica en que MCAR, al igual que CBA y CMAR, genera reglas que estiman una sola clase. Debido a esto, CMAR no necesita realizar el proceso de "Aprendizaje Recursivo" realizado por MMAC.

Tanto MCAR, MMAC, CMAR como CBA realizan un proceso de cubrimiento del conjunto de entrenamiento que es muy costoso en tiempo. Además, todos utilizan la medida de calidad *confianza* (ver limitaciones de la *confianza* en la sección 3.3) para calcular el conjunto de CARs. En la próxima sección se describirán las características de otro grupo de clasificadores que evitan este costoso proceso de cubrimiento.

3.2 Clasificadores integrados

Los clasificadores integrados utilizan diferentes estrategias para generar directamente el conjunto de CARs. De esta forma construyen el clasificador directamente del conjunto de entrenamiento (en un solo paso) y evitan el costoso proceso de cubrimiento de los clasificadores de dos etapas.

3.2.1 PRM (*Predictive Rule Mining*) y CPAR (*Classification based on Predictive Association Rules*)

En [7] se presenta un clasificador llamado PRM y su extensión denominada CPAR, ambos combinan las ventajas de la clasificación basada en CARs y las ventajas de un clasificador tradicional basado en reglas.

PRM, en vez de generar una gran cantidad de reglas candidatas como los clasificadores basados en CARs, utiliza un algoritmo voraz (FOIL [29]) que obtiene las reglas directamente del conjunto de entrenamiento (en un solo paso).

FOIL (*First Order Inductive Learner*) calcula un conjunto de reglas que permiten clasificar cuando existen sólo dos clases (permiten diferenciar ejemplos positivos de ejemplos negativos). En el proceso de generación de las reglas se utiliza una medida denominada *gain* que refleja la ganancia de adicionar un nuevo ítem a una regla [29]. A medida que FOIL va calculando las reglas, va eliminando las transacciones cubiertas por cada regla generada hasta que se cubre todo el conjunto de entrenamiento. En caso de tener más de dos clases, FOIL se aplica a cada clase c tomando las transacciones donde c está presente como ejemplos positivos y las transacciones restantes como ejemplos negativos.

El conjunto de reglas generado por FOIL es muy pequeño. Como una extensión de FOIL se desarrolló el clasificador PRM, el cual después que una transacción es cubierta por una regla, en vez de eliminarla, disminuye su peso. Por tanto, PRM produce más reglas que FOIL y cada transacción del conjunto de entrenamiento es usualmente cubierta por más de una regla.

PRM calcula los valores acumulados de la medida *gain* de cada regla y ordena el conjunto de reglas en orden descendente de estos valores. Para clasificar una nueva transacción t, PRM utiliza el criterio de la "Mejor Regla".

A diferencia de PRM, CPAR utiliza el criterio de ordenamiento LAP (*Laplace expected error estimate*) [21] y además, clasifica utilizando el criterio de las "Mejores *K* Reglas". Los autores argumentan que utilizan K reglas en la clasificación porque no se puede esperar que una simple regla clasifique bien cada transacción que ella cubra, tampoco utilizan todas las reglas porque hay diferente número de reglas por cada clase y se pueden incluir reglas con bajos valores en la clasificación. En los experimentos realizados se muestra que CPAR obtiene en promedio mejores resultados que clasificadores como CBA y CMAR.

En [7], los autores presentan PRM como un clasificador intermedio entre FOIL y CPAR y no realizan experimentos donde se compare su eficacia con otros clasificadores.

3.2.2 TFPC (*Total From Partial Classification*)

Otro clasificador integrado es el TFPC, presentado en [11]. Este clasificador ha sido utilizado en varios trabajos de los mismos autores para evaluar diferentes criterios de ordenamiento [5, 6] y para evaluar la combinación de criterios de ordenamiento [9, 14, 12].

TFPC calcula el conjunto de CARs utilizando una extensión del algoritmo de minado de reglas de asociación Apriori-TFP [30]. El algoritmo Apriori-TFP utiliza dos estructuras arbóreas (P-Trees y T-Trees) para calcular los soportes parciales y totales respectivamente. Ambas estructuras arbóreas se modifican en TFPC para calcular los soportes parciales y totales del conjunto de CARs. En la generación del conjunto de CARs se poda el espacio de búsqueda cada vez que se encuentra una regla que cumpla los umbrales de *soporte* y *confianza* establecidos. Los autores, al utilizar esta estrategia, apuestan a obtener mayor eficacia construyendo el clasificador con reglas más generales de mayor confianza.

Para ordenar el conjunto de CARs, TFPC sigue el criterio CSA y para clasificar una nueva transacción utiliza el criterio de la "Mejor Regla". En los experimentos realizados por los autores, se muestra que TFPC obtiene resultados comparables, en promedio, con los obtenidos por CMAR y CPAR.

Como se puede observar en los clasificadores descritos en esta subsección, al igual que los clasificadores de dos etapas descritos en la subsección anterior, se utiliza la medida de calidad *confianza* para calcular el conjunto de CARs y además, algunos la tienen en cuenta en los criterios de ordenamiento. Muchos autores han estudiado varias medidas para estimar la calidad de las reglas de asociación, a continuación se presentan

las limitaciones de la *confianza* y un resumen del estudio de otras medidas de calidad utilizadas para estimar la calidad de las reglas de asociación.

3.3 Limitaciones de la medida de calidad *Confianza*

Como se mostró en el trabajo relacionado, todos los algoritmos desarrollados para calcular el conjunto de CARs usan la medida de calidad *confianza*. Sin embargo, varios autores han indicado algunas limitaciones que tiene esta medida [32, 33, 16]. En particular, la presencia de ítems con altos valores de *soporte* puede llevar a obtener reglas engañosas. El siguiente ejemplo fue tomado de [32]:

EJEMPLO 3.1 En la base de datos de un censo realizado en 1990, la regla "si sirvió en las fuerzas armadas $\Rightarrow$ no sirvió en Vietnam" tiene una confianza de un 90%. Esta regla sugiere que conociendo que una persona sirvió en las fuerzas armadas podemos pensar que esta persona no sirvió en Vietnam. Sin embargo, el consecuente "no sirvió en Vietnam" tiene un soporte superior al 95%, por tanto la probabilidad de que una persona no haya servido en Vietnam decrece (de 95% a 90%) cuando conocemos que esta persona sirvió en las fuerzas armadas, resultando una asociación negativa. Claramente, esta regla es engañosa.

En [34], los autores sugirieron tres propiedades que toda medida de calidad (ACC) debe satisfacer para separar las reglas buenas de las reglas malas (asignándoles valores altos y bajos respectivamente). Estas propiedades son las siguientes:

PROPIEDAD 3.1 *Si* $Sop(X \Rightarrow Y) = Sop(X)Sop(Y)$ *entonces* $ACC(X \Rightarrow Y) = 0$

Esta propiedad indica que toda medida de calidad debe reflejar la independencia estadística.

PROPIEDAD 3.2 $ACC(X \Rightarrow Y)$ *es monótona creciente con respecto a* $Sop(X \Rightarrow Y)$ *cuando el resto de los parámetros permanece constante.*

La propiedad 3.2 puede interpretarse como sigue: Dado un conjunto de datos D y dos reglas $X \Rightarrow Y$ y $X^{'} \Rightarrow Y^{'}$ tales que $Sop(X) = Sop(X^{'})$ y $Sop(Y) = Sop(Y^{'})$. Si $Sop(X \Rightarrow Y) > Sop(X^{'} \Rightarrow Y^{'})$ entonces $X \Rightarrow Y$ es más fuerte que $X^{'} \Rightarrow Y^{'}$.

PROPIEDAD 3.3 $ACC(X \Rightarrow Y)$ *es monótona decreciente cuando* $Sop(X)$ *(ó* $Sop(Y)$*) crece y el resto de los parámetros permanece constante.*

Una medida de calidad que satisfaga esta propiedad evita obtener reglas engañosas porque su valor no se incrementa al aumentar sólo el *soporte* del antecedente o el *soporte* del consecuente.

Una medida de calidad que satisfaga las propiedades 2 y 3 tiene máximos locales cuando $Sop(X \Rightarrow Y) = Sop(X)$ ó $Sop(X \Rightarrow Y) = Sop(Y)$ y tiene un máximo global cuando $Sop(X \Rightarrow Y) = Sop(X) = Sup(Y)$.

A continuación mostraremos que la medida de calidad *confianza* (ver Eq. 1), utilizada en todos los algoritmos de clasificación basados en CARs, no satisface simultáneamente todas las propiedades:

$$Conf(X \Rightarrow Y) = \frac{Sop(X \cup Y)}{Sop(X)} \tag{1}$$

PROPOSICIÓN 3.1 *La* confianza *no satisface la propiedad 3.1.*

DEMOSTRACIÓN 3.1 *A continuación se muestra un contraejemplo: Dado el conjunto de datos transaccionales mostrado en la tabla 2(A), donde las filas representan las transacciones y las columnas representan los ítems. La tabla 2(B) muestra los* soportes *de los conjuntos de ítems* $\{i_1\}$, $\{i_2\}$ *e* $\{i_1, i_2\}$. *Debido a que* $Sop(\{i_1\})Sop(\{i_2\}) = 0.25 = Sop(\{i_1, i_2\})$, *la* confianza *de* $\{i_1\} \Rightarrow \{i_2\}$ *debe ser* 0. *Sin embargo,* $Conf(\{i_1\} \Rightarrow \{i_2\}) = 0.25/0.5 = 0.5 \neq 0$

Table 2. (A) Conjunto de datos transaccionales y (B) Soporte de algunos conjuntos de ítems en D

A

i_1	i_2	i_3
1	0	0
0	0	1
0	1	1
1	1	1

B

Conj. de ítems	Soporte
$\{i_1\}$	0.5
$\{i_2\}$	0.5
$\{i_1, i_2\}$	0.25

PROPOSICIÓN 3.2 *La* confianza *satisface la propiedad 3.2*

DEMOSTRACIÓN 3.2 *Es evidente de acuerdo a la Eq. (1).*

PROPOSICIÓN 3.3 *La* confianza *satisface la propiedad 3.3 para el* $Sop(X)$

DEMOSTRACIÓN 3.3 *Es evidente de acuerdo a la Eq. (1).*

PROPOSICIÓN 3.4 *La* confianza *no satisface la propiedad 3.3 para el* $Sop(Y)$.

DEMOSTRACIÓN 3.4 *Debido a que* $Sop(Y)$ *no se tiene en cuenta en la Eq. (1) entonces la propiedad 3.3 no se satisface para el* $Sop(Y)$.

En resumen, la confianza no refleja la independencia estadística (propiedad 3.1) ni detecta dependencias negativas. Además, en su definición no considera el *soporte* del consecuente, el cual es muy importante porque es la clase. Por tanto, podemos concluir que la *confianza* no es una buena medida de calidad para separar reglas buenas de reglas malas.

Hasta aquí se han detallado las limitaciones de la *confianza* y se han discutido las propiedades propuestas en [34], las cuales debe satisfacer una medida de calidad para separar las reglas buenas de las reglas malas.

3.4 Estudio de otras medidas de calidad

Además de la *confianza*, en la literatura se han estudiado otras medidas para medir le fuerza de una regla de asociación como son las medidas *Conviction, Interest, Chi-square, Certainty factors* y *Netconf*. En [16] los autores presentaron un estudio de estas medidas (excepto *Netconf*) y concluyeron que sólo las medidas *Interest* y *Certainty factors* satisfacen las propiedades $1-3$ pero ambas tienen otras limitaciones.

La medida *Interest* (Eq. 2) no está acotada [16] por lo que no es fácil comparar los valores de diferentes reglas, las magnitudes de las diferencias entre los valores de *Interest* de diferentes reglas no son fáciles de interpretar y resulta difícil definir un umbral de *Interest*. Además, la medida *Interest* es simétrica y esto casi nunca sucede en la práctica.

$$Int(X \Rightarrow Y) = \frac{Sop(X \Rightarrow Y)}{Sop(X)Sup(Y)} \tag{2}$$

Por otro lado, la medida de calidad *Certainty factors* se define por la ecuación 3

$$CF(X \Rightarrow Y) = \begin{cases} \frac{Conf(X \Rightarrow Y) - Sop(Y)}{1 - Sup(Y)} & \text{si} Conf(X \Rightarrow Y) > Sop(Y) \\ \frac{Conf(X \Rightarrow Y) - Sop(Y)}{Sop(Y)} & \text{si} Conf(X \Rightarrow Y) < Sop(Y) \\ 0 & \text{en otro caso} \end{cases} \tag{3}$$

Valores negativos de *Certainty factors* significan dependencia negativa, valores positivos significan dependencia positiva y 0 significa independencia. Sin embargo, el valor que toma la medida depende del *soporte* del consecuente (la clase en nuestro caso) cuando $Conf(X \Rightarrow Y)$ es cercano a $Sop(Y)$. Para una mejor comprensión veamos el siguiente ejemplo tomado de [19]:

EJEMPLO 3.2 Sean $Sop(X) = 0.5$ y $Sop(Y) = 0.9$. Si $Sop(X \Rightarrow Y) = 0.45$ entonces X y Y son independientes. Además, debido a que:

$$Conf(X \Rightarrow Y) = \frac{Sop(X \Rightarrow Y)}{Sop(X)} = \frac{Sop(X)Sop(Y)}{Sop(X)} = Sop(Y)$$

se tiene que si $Sop(X \Rightarrow Y) = 0.43$, el valor del *Certainty factors* de $X \Rightarrow Y$ es -0.044 por la ecuación 3. Esto significa que existe una pequeña relación negativa entre X y Y. Pero si $Sop(X \Rightarrow Y) = 0.47$, el valor del *Certainty factors* de $X \Rightarrow Y$ es 0.4 por la ecuación 3. Esto significa 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, la medida *Certainty factors* obtiene resultados muy diferentes en cada dirección.

En [19], los autores introdujeron la medida de calidad *Netconf* para estimar la fortaleza de las ARs. Además, los autores mostraron que la medida *Netconf* resuelve las limitaciones de las medidas *confianza*, *Interest* y *Certainty factors*. Sin embargo, los autores no probaron que el *Netconf* satisfacía las propiedades sugeridas en [34]. La demostración de estas propiedades puede verse en la sección de resultdos preliminares.

Table 3. Criterios de Decisión.

Mejor Regla	Mejores K Reglas	Todas las Reglas
Una sola regla no puede predecir exactamente la clase de cada transacción que ésta cubra.	Puede afectarse la eficacia cuando existe gran desbalance entre el número de CARs por clase que cubre a la nueva transacción.	Pueden incluirse reglas de baja calidad entre el conjunto de CARs utilizado para clasificar.

3.5 Motivación

Como puede apreciarse en las subsecciones anteriores, se han desarrollado varias estrategias que incluyen diferentes medidas de calidad para evaluar y ordenar las CARs, diferentes algoritmos de generación de CARs y diferentes criterios de decisión; todas estas estrategias han estado dirigidas a mejorar la eficacia de los clasificadores basados en CARs. No obstante, los algoritmos presentados tienen algunas de las siguientes limitaciones:

1. A pesar de las limitaciones presentadas en [16] (ver subsección 3.3), la *confianza*, es utilizada para calcular y ordenar el conjunto de CARs.
2. Las soluciones propuestas para reducir el número de CARs dejan de generar algunas reglas de buena calidad.
3. Los criterios de decisión existentes tienen limitaciones que pueden afectar la eficacia del clasificador (ver tabla 3).
4. Cuando ninguna CAR cubre a la transacción que se desea clasificar se asigna la clase mayoritaria, lo cual puede afectar la eficacia del clasificador.
5. Cuando hay empate se asigna, de forma aleatoria, una de las clases empatadas, lo cual puede afectar la eficacia del clasificador.

Estas limitaciones motivan a continuar investigando sobre la construcción de clasificadores basados en CARs que tengan una mejor eficacia que los clasificadores existentes basados en CARs proponiendo el uso de medidas de calidad para el cálculo y ordenamiento del conjunto de CARs, que no presenten las desventajas de la *confianza*; proponiendo nuevos criterios decisión, proponiendo criterios de desempate para evitar la asignación aleatoria de clases, etc.

4 Investigación propuesta

En esta sección se presenta la propuesta de investigación compuesta por las preguntas de investigación, los objetivos, la metodología a seguir para alcanzarlos, las contribuciones esperadas y el calendario de actividades.

4.1 Preguntas de investigación

- ¿Es posible proponer una nueva medida de calidad para el cálculo y ordenamiento del conjunto de CARs, que no tenga las limitaciones de la *confianza*?

- ¿Es posible desarrollar un algoritmo eficiente para el cálculo de CARs y una estrategia de poda que permita generar más reglas que obtengan valores significativos de la medida de calidad?
- ¿Es posible proponer una estrategia de cubrimiento que reduzca el número de casos en que ninguna CAR cubra a la transacción que se desea clasificar y así reducir el número de asignaciones de la clase mayoritaria?
- ¿Es posible proponer un criterio de decisión que no tenga los problemas de los criterios de decisión existentes?
- ¿Es posible proponer un criterio de desambiguación que reduzca la cantidad de asignaciones aleatorias cuando hay clases empatadas?

4.2 Objetivo general

El objetivo general de esta propuesta de investigación doctoral es:

Construir un nuevo clasificador basado en CARs a partir de una muestra de entrenamiento, que alcance mayor eficacia que los clasificadores existentes basados en CARs.

Para cumplir este objetivo se definen los siguiente objetivos específicos.

4.3 Objetivos específicos

1. Proponer una nueva medida de calidad para el cálculo y ordenamiento del conjunto de CARs, que no tenga las limitaciones de la *confianza* descritas en la sección 3.3.
2. Diseñar e implementar un algoritmo eficiente para calcular el conjunto de CARs que haga uso de la medida de calidad para CARs del objetivo 1 y además, proponer una estrategia de poda que permita generar más reglas con valores significativos de la medida de calidad que las estrategias de poda existentes.
3. Proponer una estrategia de cubrimiento para reducir los casos en que ninguna CAR cubra a la transacción que se desea clasificar y así reducir el número de asignaciones de la clase mayoritaria.
4. Proponer un nuevo criterio de decisión que resuelva los problemas de los criterios de decisión existentes descritos en la sección 3.
5. Proponer un criterio de desambiguación de clases para reducir la cantidad de asignaciones aleatorias cuando hay clases empatadas.

4.4 Metodología

Para alcanzar los objetivos específicos planteados anteriormente se propone la siguiente metodología:

1. Proponer una nueva medida de calidad para el cálculo y ordenamiento del conjunto de CARs, que no tenga las limitaciones de la *confianza*.
    a) Estudiar las medidas de calidad propuestas en la literatura para ARM y analizar si alguna reduce las limitaciones de la *confianza*. Si el estudio realizado da como resultado alguna medida que no

tenga las limitaciones de la *confianza* entonces se utilizará para el cálculo de las CARs. Inicialmente estudiamos las medidas *Confianza*, *Soporte*, *Conviction*, *Interest*, *Chi-square*, *Certainty factors* y *Netconf*. Por el momento se han obtenido buenos resultados con la medida de calidad *Netconf* propuesta en [19] para estimar la fuerza de una regla de asociación.

b) Independientemente del resultado obtenido en el paso anterior, se propondrá una nueva medida de calidad para el cálculo de CARs que no tenga las limitaciones de la *confianza*.

c) Analizar los criterios de ordenamiento de CARs reportados en la literatura y proponer un nuevo criterio de ordenamiento que haga uso de la(s) medida(s) de calidad resultante(s) en el paso 1a) y 1b). Inicialmente hemos obtenido buenos resultados utilizando la medida *Netconf* para ordenar el conjunto de CARs.

d) Evaluar la combinación de los criterios de ordenamiento existentes con el criterio propuesto en $1c)$. Los trabajos recientes [9, 14, 12] han mostrado que la combinación de criterios de ordenamiento puede mejorar la eficacia de los clasificadores.

2. Diseñar e implementar un algoritmo eficiente para calcular el conjunto de CARs que haga uso de la medida de calidad para CARs del objetivo 1 y además, proponer una estrategia de poda que permita generar más reglas con valores significativos de la medida de calidad que las estrategias de poda existentes.

   a) Adaptar el algoritmo publicado en [18] para calcular el conjunto de CARs haciendo uso de la medida de calidad del objetivo 1. Para ello se crearán sólo las clases de equivalencias [3] que involucren al conjunto de clases predefinido y se modificará la información asociada a cada clase de equivalencia para poder calcular eficientemente, para cada regla, los valores correspondientes de la medida de calidad utilizada.

   b) Proponer una estrategia de poda que permita generar más reglas de buena calidad (según la medida utilizada). En vez de podar el espacio de búsqueda cada vez que se encuentra una regla que satisface los umbrales de *soporte* y *confianza*, se evaluará la siguiente estrategia: cuando se encuentre una regla que satisface el umbral de *Netconf* establecido, se continuarán generando reglas mientras que el valor de *Netconf* no disminuya. Esta misma estrategia se evaluará con la medida que se obtenga en $1b)$.

   c) Evaluar el algoritmo desarrollado en el paso anterior. En la evaluación se medirá como influyen en la eficacia del clasificador la nueva medida de calidad y la nueva estrategia de poda.

3. Proponer una estrategia para reducir los casos en que ninguna CAR cubra a la transacción que se desea clasificar y así reducir el número de asignaciones de la clase mayoritaria.

   a) Considerar el cubrimiento inexacto de las nuevas transacciones que se deseen clasificar. Para ello haremos más flexible el criterio de cubrimiento de una transacción por una regla (ver definición 2.9). Inicialmente probaremos permitir que una regla $\{i_1, i_2, \ldots, i_n\} \Rightarrow c$ cubra a una transacción t si al menos $n-1$ ítems del antecedente de la regla pertenecen a t.

   b) Comparar la eficacia en la clasificación considerando el cubrimiento inexacto contra el cubrimiento exacto.

4. Proponer un nuevo criterio de decisión que resuelva los problemas de los criterios de decisión existentes. Los problemas de cada uno de los criterios de decisión existentes fueron discutidos en la sección 3.

a) Comprobar experimentalmente el análisis hecho por otros autores respecto a los criterios de decisión existentes. Los últimos trabajos utilizan el criterio de las "Mejores K Reglas" por clase como criterio de decisión y descartan los criterios de la "Mejor Regla" y de "Todas las Reglas". No obstante, cuando se utiliza el criterio de las "Mejores K Reglas" y existe gran desbalance en el número de CARs por clase que cubre a una nueva transacción, la eficacia del clasificador se puede afectar.

b) Seleccionar automáticamente un valor de K para el criterio de "Las Mejores K Reglas" posiblemente diferente para cada clase y para cada transacción, de esta forma se puede reducir el efecto del desbalance entre el número de CARs por clase, lo cual puede repercutir directamente en el desbalance del número de CARs por clase que cubre a una nueva transacción. Inicialmente evaluaremos tomar la mejor regla y dado un umbral, tomar todas las reglas cercanas a ella respecto al valor de la medida utilizada.

c) Comparar la eficacia del clasificador aplicando el criterio de decisión propuesto en el paso anterior y aplicando los criterios de decisión existentes.

5. Proponer un criterio de desambiguación de clases para reducir la cantidad de asignaciones aleatorias cuando hay clases empatadas.

   a) Considerar la eliminación de la CAR de menor calidad, mayor calidad o ambas mientras haya empate.

   b) Considerar el uso de un segundo clasificador posiblemente también basado en reglas.

   c) Comparar la eficacia en la clasificación usando los criterios de desambiguación de clases resultantes en los pasos anteriores contra la eficacia que se obtiene al asignar una clase aleatoria.

6. Diseñar e implementar un clasificador basado en CARs, a partir de una muestra de entrenamiento, que utilice las propuestas de los objetivos anteriores y que alcance mayor eficacia que los clasificadores existentes basados en CARs.

   a) Integrar las propuestas hechas en los pasos anteriores para construir un clasificador basado en CARs.

7. Evaluar la eficacia del clasificador obtenido.

   a) Realizar una comparación experimental del clasificador obtenido contra los clasificadores basados en CARs, reportados en la literatura.

   b) Se considerarán en la experimentación los conjuntos de datos del repositorio UCI [31] por ser los comúnmente usados en los trabajos reportados.

4.5 Contribuciones

Con el desarrollo de esta tesis se esperan las siguientes contribuciones:

- Una nueva medida de calidad para el cálculo, minado y ordenamiento del conjunto de CARs que no tenga las limitaciones de la confianza.
- Un algoritmo eficiente para generar el conjunto de CARs.

- Una nueva estrategia de cubrimiento para reducir el número de casos que ninguna CAR cubra a la transacción que se desea clasificar y así reducir el número de asignaciones de la clase mayoritaria.
- Un criterio de decisión que no presente los problemas de los criterios de decisión existentes.
- Un clasificador basado en CARs que alcance mayor eficacia que la alcanzada por los clasificadores existentes basados en CARs.

4.6 Calendario de Actividades

En la figura 1 se muestra el calendario de actividades.

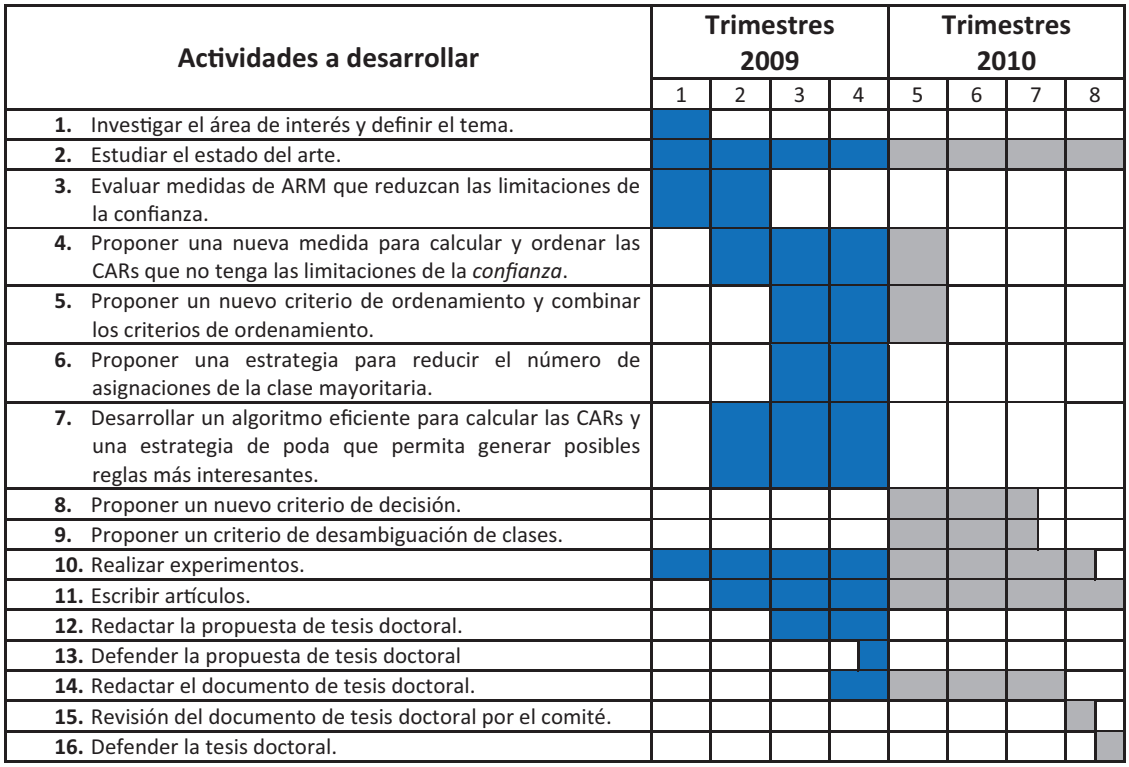

Figure 1. Calendario de Actividades

5 Resultados preliminares

En esta sección se introducen los resultados obtenidos hasta el momento. Como primer resultado se demostró que la medida de calidad *Netconf* satisface las propiedades $1-3$ sugeridas por Shapiro. La medida *Netconf* se define como:

$$Netconf(X \Rightarrow Y) = \frac{Sop(X \Rightarrow Y) - Sop(X)Sop(Y)}{Sop(X)(1 - Sop(X))} \tag{4}$$

Analizando la ecuación (4) y tomando en cuenta que el *soporte* toma valores en el intervalo $[0, 1]$ se tiene que: si $Sop(X \Rightarrow Y) = Sop(X)Sop(Y)$ entonces el numerador de la ecuación 4 se hace 0 y por tanto, se hace 0 el *Netconf* (satisface la propiedad 3.1); si $Sop(X \Rightarrow Y)$ aumenta permaneciendo el resto de los parámetros constantes entonces aumenta el numerador de la ecuación 4 y por tanto aumenta el *Netconf* (satisface la propiedad 3.2); si $Sop(Y)$ aumenta permaneciendo el resto de los parámetros constantes entonces disminuye el numerador de la ecuación 4 y por tanto disminuye el *Netconf* (satisface la propiedad 3.3 para $Sop(Y)$). Sólo nos queda mostrar que el *Netconf* satisface la propiedad 3.3 para $Sop(X)$, para ello probaremos la siguiente proposición.

PROPOSICIÓN 5.1 *La medida de calidad* Netconf *satisface la propiedad 3.3 para* $Sup(X)$.

DEMOSTRACIÓN 5.1 *En la Eq. (4), sean* $Sop(X \Rightarrow Y) = S_{xy}$, $Sop(Y) = S_y$, $Sop(X) = x$ *los soportes de* $X \Rightarrow Y$, Y *y* X *respectivamente, con* S_{xy} *y* S_y *constantes que satisfacen* $0 < S_{xy} \leq S_y < 1$ *y* $x \in (0, 1)$. *Podemos reescribir el miembro derecho de la Eq. (4) en términos de* S_{xy}, S_y, *y* x, *como sigue:*

$$f(x) = \frac{S_{xy} - S_y x}{x(1 - x)} \tag{5}$$

y probar que $f'(x) < 0$, *lo cual implicaría que* $f(x)$ *sería estrictamente decreciente y por tanto la proposición 5.1 sería verdadera. Calculando la primera derivada y reduciendo términos se tiene que:*

$$f'(x) = \frac{-S_y x^2 + 2S_{xy}x - S_{xy}}{x^2(1 - x)^2} \tag{6}$$

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

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

y $x^2(1 - x)^2 > 0$, *por tanto,* $f'(x) < 0$.

Otra propiedad de la medida *Netconf* es que $NetConf(X \Rightarrow Y) \neq NetConf(Y \Rightarrow X)$ para $Sop(X) \neq Sop(Y)$ y $Sop(X) + Sop(Y) \neq 1$, lo cual indica que el *Netconf* es una medida no simétrica. Además, el *Netconf* puede expresar la fortaleza de las asociaciones en ambas direcciones [19]. El *Netconf* toma valores en el intervalo $[-1, 1]($ *si y solo si* $0 < Sop(X) < 1)$ [19], valores positivos del *Netconf* representan una asociación positiva entre el antecedente y el consecuente de la regla; valores negativos del *Netconf* representan una asociación negativa y cero representa independencia.

Debido a que la medida *Netconf* satisface las propiedades $1 - 3$ y no tiene las limitaciones de las medidas *Interest*, *Certainty factors* y *Confianza*, la utilizaremos para calcular y ordenar el conjunto de CARs.

5.1 Clasificador CAR-NF

En esta sección describiremos un nuevo clasificador, llamado CAR-NF, que utiliza un nuevo algoritmo para calcular el conjunto de CARs (CAR-CA, descrito en 5.1.1). CAR-CA utiliza la medida *Netconf* para calcular el conjunto de CARs e introduce una nueva estrategia de poda para obtener más reglas de buena calidad.

5.1.1 Algoritmo CAR-CA

El algoritmo CAR-CA se basa en el algoritmo CA [18] de minado de conjuntos frecuentes, el cual es más eficiente que otros algoritmos eficientes como: Apriori (utilizado en CBA), Fp-growth (utilizado en CMAR), Eclat (utilizado en MMAC y MCAR) y Apriori-TFP (utilizado en TFPC).

CAR-CA permite generar eficientemente un gran número de CARs y evita generar reglas engañosas ya que evalúa y filtra el conjunto de CARs usando la medida de calidad *Netconf*. Por el momento sólo las asociaciones positivas son utilizadas para nosotros.

En [3] para calcular las ARs, los autores proponen particionar el espacio de búsqueda en clases de equivalencias agrupando los conjuntos de ítems de igual tamaño k que tengan un prefijo común de longitud $(k-1)$. Una clase de equivalencia que agrupe conjuntos de ítems de tamaño k se denotará como EC_k. En CAR-CA, consideramos cada clase predefinida $c \in C$ como otro ítem y dividimos el espacio de búsqueda de CARs en clases de equivalencias definidas por la relación de equivalencia siguiente: "Las CARs de tamaño k que compartan el mismo consecuente (la misma clase) y los primeros $k-2$ ítems del antecedente (el cual tiene $k-1$ ítems) pertenecen a la misma clase de equivalencia".

De forma similar al algoritmo CA, para aprovechar las ventajas de las operaciones bit-a-bit, se representa el conjunto de datos como una matriz binaria m x n, siendo m el número de transacciones y n el número de ítems incluyendo las clases. Los valores binarios 1 y 0 denotan la presencia y ausencia respectivamente de un ítem en una transacción. Cada columna asociada a un ítem j puede ser comprimida y representada como un arreglo de enteros I_j, como sigue:

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

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

Los algoritmos previamente desarrollados para el minado de CARs, necesitan un paso de *backtracking* para calcular el *soporte* del antecedente de las reglas. Nuestro algoritmo evita este paso de *backtracking*; para ello, genera iterativamente una lista L_{EC_k} que representa las clases de equivalencias que contienen CARs de longitud k (k-CARs), cuyos elementos tienen el siguiente formato:

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

donde c es el consecuente de las CARs agrupadas, $AntPref_{k-2}$ es el $(k-2)$-itemset común a todos los antecedentes de las CARs agrupadas (prefijo del antecedente), $AntSuff$ es el conjunto de todos los ítems j que extienden al prefijo $AntPref_{k-2}$ (sufijos del antecedente), donde j es lexicográficamente mayor que cada ítem del prefijo del antecedente, y $IA_{AntPref_{k-2}}$ es un arreglo de enteros no nulos que se construye mediante la intersección (usando operaciones AND) de los arreglos I_j, donde j pertenece a $AntPref_{k-2}$. Los arreglos IA almacenan los *soportes* acumulados del prefijo del antecedente de cada clase de equivalencia EC_k. Cuando k aumenta, el número de elementos de IA disminuye porque las operaciones AND generan ceros, y los ceros no se almacenan porque no influyen ni en el *soporte* ni en el *Netconf* de las reglas. El procedimiento para obtener IA es el siguiente: Sean i y j dos ítems,

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

igualmente, sea X un conjunto de ítems y j un ítem

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

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

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

donde $BitCount(b)$ es una función que calcula la cantidad de bits iguales a 1 en b. El *Netconf* (Eq. 4) puede ser fácilmente calculado tomando en cuenta el formato propuesto para agrupar las clases de equivalencias y utilizando las ecuaciones 9, 10 y 11.

Se puede resumir que la eficiencia del algoritmo CAR-CA se basa en dos características principales: el uso eficiente de las clases de equivalencia, introducidas en [3], y el uso eficiente de operaciones bit-a-bit para calcular el *Netconf* de las CARs.

Algoritmos recientes para el minado de CARs [5, 9, 14, 12] podan el espacio de búsqueda cada vez que una regla satisface los umbrales de *soporte* y *confianza* establecidos, esto trae como consecuencia que se puedan perder reglas de buena calidad. CAR-CA, después de encontrar la primera regla que satisface el umbral de *Netconf*, continúa generando reglas hasta que el *Netconf* comience a decrecer, de esta forma permite generar más reglas de buena calidad. En el pseudocódigo diremos que cuando el *Netconf* comience a decrecer se satisface el *Criterio de Poda* (ver línea 7 del Algoritmo 2).

El pseudocódigo del algoritmo CAR-CA se muestra en el Algoritmo 1.

Algoritmo 1: CAR-CA

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

```
1  Answer = ∅
2  C = {Conjunto de clases}
3  L = {1-itemsets}
4  forall c ∈ C do
5  |  ECGen(⟨{c}, NULL, NULL, {L}⟩, L_EC2)
6  |  k = 3
7  |  while L_EC(k-1) ≠ ∅ do
8  |  |  forall ec ∈ L_EC(k-1) do
9  |  |  |  ECGen(ec, L_ECk)
10 |  |  end
11 |  |  Answer = Answer ∪ L_ECk
12 |  |  k = k + 1
13 |  end
14 end
15 return Answer
```

En la línea 3, del algoritmo 1, se calculan los 1-itemsets. En la línea 5, se construyen las clases de equivalencia de tamaño 2 para cada clase c. En las líneas $7-13$, se procesa cada clase de equivalencia de tamaño mayor que 2 utilizando la función *ECGen*.

La función *ECGen* toma como argumento una clase de equivalencia de tamaño $k-1$ y genera un conjunto de clases de equivalencias de tamaño k (ver Algoritmo 2). Las clases de equivalencia generadas por este algoritmo sólo contienen CARs con valores positivos de *Netconf*.

Algoritmo 2: ECGen

Input: Una EC en formato $\langle c, AntPref, IA_{AntPref}, AntSuff \rangle$
Output: El conjunto de clases de equivalencias generado

$Answer = \emptyset$
forall $i \in AntSuff$ **do**
  $AntPref' = AntPref \cup \{i\}$
  $IA_{AntPref'} = IA_{AntPref \cup \{i\} \cup \{c\}}$
  $AntSuff' = \emptyset$
  forall $(i' \in AntSuff)$ *y* (i' *lexicográficamente mayor que* i) **do**
    if $AntPref' \cup \{i'\} \Rightarrow c$ *No satisface el Criterio de Poda* **then**
      $AntSuff' = AntSuff' \cup \{i'\}$
    end
  end
  if $AntSuff' \neq \emptyset$ **then**
    $Answer = Answer \cup \{\langle c, AntPref', IA_{AntPref'}, AntSuff' \rangle\}$
  end
end
return *Answer*

Una vez que el conjunto de CARs es generado, se ordena descendentemente de acuerdo a los valores de *Netconf*. Una CAR con mayor valor de *Netconf* tiene mayor asociación positiva entre su antecedente y su consecuente (la clase).

5.2 Cubrimiento inexacto

En los clasificadores reportados en la literatura, cuando ninguna regla cubre a la transacción que se desea clasificar asignan la clase mayoritaria y esto puede afectar la eficacia del clasificador.

Supongamos que se tiene un umbral de *soporte* igual a 1%, un umbral de *confianza* igual a 50% y un umbral de *Netconf* igual a 35%, en la tabla 4 se muestran 4 reglas y sus respectivos valores de *confianza* y *Netconf*.

Table 4. Ejemplo de reglas y sus valores de Soporte, Confianza y Netconf.

Regla	Soporte	Confianza	Netconf
$\{i_1\} \Rightarrow c$	2%	52%	38%
$\{i_1, i_2\} \Rightarrow c$	1.9%	51%	41%
$\{i_1, i_2, i_3\} \Rightarrow c$	1.7%	53%	43%
$\{i_1, i_2, i_3, i_4\} \Rightarrow c$	1.5%	51%	40%

Los clasificadores que podan el espacio de búsqueda cada vez que encuentran una regla que cumple los umbrales de *soporte* y *confianza*, generarían solamente la primera regla ($\{i_1\} \Rightarrow c$). En caso de tener que clasificar la transacción $\{i_2, i_3\}$ no podrían cubrirla y la asignarían a la clase mayoritaria. La estrategia que sigue el algoritmo CAR-CA permite generar las 3 primeras reglas y no la cuarta porque comienza a decrecer el valor de *Netconf*. No obstante, según la definición 2.9 debido a que $\{i_1, i_2, i_3\} \Rightarrow c$ no cubre a la transacción $\{i_2, i_3\}$, esta se seguiría asignando a la clase mayoritaria.

Como una alternativa, proponemos hacer más flexible la definición 2.9 permitiendo que una regla $\{i_1, i_2, \ldots, i_n\} \Rightarrow c$ cubra a una transacción t si al menos $n-1$ ítems del antecedente de la regla pertenecen a t. Utilizando este criterio de cubrimiento inexacto se reduce el número de asignaciones de la clase mayoritaria lo que repercute directamente en la eficacia del clasificador como se puede ver en los experimentos realizados.

5.3 Clasificación

Para clasificar una nueva transacción, inicialmente decidimos seguir el criterio de decisión de las "Mejores K Reglas" por ser el criterio que mejores resultados ha obtenido en los trabajos reportados. Los algoritmos 3 y 4 muestran el pseudo código de la fase de entrenamiento y de la fase de clasificación respectivamente:

Algoritmo 3: CAR-NF: Fase de entrenamiento

Input: Conjunto de entrenamiento db
Output: Clasificador

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

Algoritmo 4: CAR-NF: Fase de clasificación

Input: Conjunto de $CARs$ ordenadas y una nueva transacción t
Output: Clase asignada

```
1 Answer = ∅
2 BestK = Selecciona_MejoresK(t)
3 Answer = Clasifica(BestK)
4 return Answer
```

5.4 Experimentación

Como en otros trabajos [5, 4, 13, 7], se utilizaron 15 conjuntos de datos del repositorio UCI *Machine Learning Repository* [31] y se utilizó validación cruzada con 10 pliegues (*ten-fold cross-validation*). Para los clasificadores CBA, CMAR, CPAR y TFPC usamos el umbral de *confianza* igual a 50% y el umbral de *soporte* igual a 1%, como sus autores proponen. Para CAR-NF usamos el umbral de *Netconf* igual a 0, así consideramos sólo asociaciones positivas.

Para realizar los experimentos se implementaron dos versiones de CAR-NF. La primera versión, llamada "CAR-NF (1)", usa la medida de calidad *Netconf* pero no aplica la nueva estrategia de poda ni utiliza el cubrimiento inexacto. La segunda versión, llamada "CAR-NF (2)", usa *Netconf*, aplica la nueva estrategia de poda y utiliza el cubrimiento inexacto. El clasificador propuesto, CAR-NF, corresponde a la versión CAR-NF(2).

En la tabla 5, los resultados muestran que ambas versiones, CAR-NF(1) y CAR-NF(2), tienen como promedio mejor eficacia que los clasificadores CBA, CMAR, CPAR, y TFPC. Las implementaciones de

estos clasificadores fueron bajadas del sitio oficial de Frans Coenen (http://www.csc.liv.ac.uk/∼frans). Es importante resaltar que la eficacia obtenida en nuestros experimentos, para cada uno de los clasificadores, es la misma que la reportada en los trabajos previos.

Table 5. Eficacia obtenida por los clasificadores basados en CARS.

Conj. de Datos	CBA	CMAR	CPAR	TFPC	CAR-NF(1)	CAR-NF(2)
adult	70.15	71.23	76.71	76.16	77.38	**78.15**
anneal	**97.93**	83.52	90.22	84.63	93.28	96.76
breast	93.32	85.26	94.88	**95.91**	95.39	95.62
dematology	80.00	**83.92**	80.08	76.30	76.85	77.93
ecoli	**83.17**	78.01	80.59	58.53	81.59	82.65
flare	84.23	84.30	64.75	84.30	84.35	**84.43**
glass	68.30	**75.37**	64.01	64.09	65.01	66.11
iris	94.00	93.33	**95.33**	**95.33**	94.08	94.67
led7	66.56	73.31	71.38	68.71	72.23	**73.88**
nursery	90.12	78.94	78.59	77.75	90.32	**90.82**
pageBlocks	90.94	83.98	**92.54**	89.98	88.79	89.90
penDigits	81.73	83.48	80.39	**87.39**	84.51	85.59
ticTacToe	99.12	99.25	98.64	**99.67**	99.34	99.62
pima	75.03	73.85	74.82	74.36	74.43	**75.19**
wine	53.22	72.24	**88.03**	72.09	83.64	84.74
Promedio	81.85	81.98	82.21	80.35	**84.08**	**85.07**

La tabla 6 muestra las diferencias entre la eficacia de cada clasificador y la eficacia del mejor clasificador en cada conjunto de datos, la última fila de la tabla muestra la diferencia promedio. De la tabla 6, podemos concluir que los resultados de ambas versiones, CAR-NF(1) y CAR-NF(2), están mucho más cercanos al primer lugar que los otros clasificadores. Además, ambas versiones tienen como promedio una diferencia de -2.70 y -1.71 respecto al mejor clasificador. Estas diferencias son bastante pequeñas, el resto de los clasificadores promedian una diferencia al menos 2 veces mayor respecto al mejor clasificador. Para reforzar este resultado asignamos a cada clasificador, como se muestra en la tabla 7, un valor entero entre 1 y 6 de acuerdo a la posición en el ranking basado en la eficacia, y en la última fila de la tabla 7 mostramos el promedio obtenido de los seis clasificadores. Mientras más pequeños sean los valores mejor será el resultado obtenido por el clasificador. En la tabla 7 puede notarse que para ambas versiones de CAR-NF la eficacia obtenida está en promedio entre los 3 primeros lugares.

La tabla 8 muestra el número de CARs (#CARs) y el número de asignaciones de la clase mayoritaria (#ACM) para CAR-NF(1) y CAR-NF(2). Cuando utilizamos el cubrimiento inexacto en CAR-NF(2) obtenemos como promedio un 13.8% más de reglas y se realizan como promedio un 46.9% menos de asignaciones de la clase mayoritaria, lo cual incrementa la eficacia del clasificador.

En general, CAR-NF alcanza buenos resultados mejorando los resultados de otros clasificadores basados en CARs como CBA, CMAR, CPAR y TFPC.

6 Conclusiones

Hasta el momento se desarrolló un algoritmo para calcular el conjunto de CARs utilizando la medida de calidad *Netconf*, que no presenta las limitaciones de la *confianza*, así como una nueva estrategia de poda que

Table 6. Diferencias de la eficacia de cada clasificador respecto al mejor clasificador.

Conj. de Datos	CBA	CMAR	CPAR	TFPC	CAR-NF(1)	CAR-NF(2)
adult	-8.00	-6.92	-1.44	-1.99	-0.77	0.00
anneal	0.00	-14.41	-7.71	-3.30	-4.65	-1.17
breast	-2.59	-10.65	-1.03	0.00	-0.52	-0.29
dematology	-3.92	0.00	-3.84	-7.62	-7.07	-5.99
ecoli	0.00	-5.16	-2.58	-24.64	-1.58	-0.52
flare	-0.20	-0.13	-19.68	-0.13	-0.08	0.00
glass	-7.07	0.00	-11.36	-11.28	-10.36	-9.26
iris	-1.33	-2.00	0.00	0.00	-1.25	-0.66
led7	-7.32	-0.57	-2.50	-5.17	-1.65	0.00
nursery	-0.70	-11.88	-12.23	-13.07	-0.50	0.00
pageBlocks	-1.60	-8.56	0.00	-2.56	-3.75	-2.64
penDigits	-7.00	-3.91	0.00	-5.66	-2.88	-1.80
ticTacToe	-0.55	-0.42	-1.03	0.00	-0.33	-0.05
pima	-0.16	-1.34	-0.37	-0.83	-0.76	0.00
wine	-34.81	-15.79	0.00	-15.94	-4.39	-3.29
Promedio	-5.02	-5.45	-4.25	-6.15	**-2.70**	**-1.71**

Table 7. Ranking basado en la eficacia obtenida en cada conjunto de datos.

Conj. de Datos	CBA	CMAR	CPAR	TFPC	CAR-NF(1)	CAR-NF(2)
adult	6	3	5	4	2	1
anneal	1	4	5	6	3	2
breast	5	4	6	1	3	2
dematology	3	1	2	6	5	4
ecoli	1	5	4	6	3	2
flare	4	2	6	2	5	1
glass	2	1	6	5	4	3
iris	4	6	1	1	5	3
led7	6	2	4	5	3	1
nursery	3	4	5	6	2	1
pageBlocks	2	6	1	3	5	4
penDigits	5	4	6	1	3	2
ticTacToe	5	4	6	1	3	2
pima	2	6	3	4	5	1
wine	6	4	1	5	3	2
Promedio	3.67	3.73	4.07	3.73	**3.60**	**2.07**

permite generar más reglas de buena calidad. Además, se propuso una estrategia de cubrimiento inexacto para reducir los casos en que ninguna CAR cubra a la transacción que se desea clasificar. Estos resultados se integraron y se desarrolló un clasificador basado en CARs que alcanza valores de eficacia superiores a los alcanzados por otros clasificadores del estado del arte.

Teniendo en cuenta estos resultados preliminares, consideramos que los objetivos propuestos pueden ser

Table 8. Número de CARs y número de asignaciones de la clase mayoritaria con y sin cubrimiento inexacto.

	CAR-NF(1)		CAR-NF(2)	
Conj. de Datos	#CARs	#ACM	#CARs	#ACM
adult	23987	359	27215	223
anneal	1041	14	1325	6
breast	438	16	699	4
dematology	5288	24	6142	9
ecoli	466	18	544	7
flare	4071	76	4985	32
glass	3025	16	3721	7
iris	312	11	403	4
led7	6743	104	7560	48
nursery	10894	139	11822	51
pageBlocks	6050	116	7123	42
penDigits	11657	124	12493	53
ticTacToe	694	13	823	5
pima	4790	48	5538	16
wine	364	14	437	5
Promedio	5321.33	72.80	6055.33	34.13

alcanzados en el tiempo planteado y con la calidad deseada, siguiendo la metodología propuesta.

References

[1] Agrawal, R. and Srikant, R.: Fast Algorithms for Mining Association Rules, In Proceedings of the 1994 International Conference on Very Large Data Bases (VLDB'94), Santiago, Chile, pp. 487-499, 1994.

[2] Han, J., Pei, J. and Yin, Y.: Mining Frequent Patterns without Candidate Generation, In Proceedings of the 2000 ACM-SIGMOD International Conference on Management of Data (SIGMOD'2000), Dallas, TX, pp. 1-12, 2000.

[3] Zaki, M., Parthasarathy, S., Ogihara, M. and Li, W.: New Algorithm for fast Discovery of Association Rules, In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD'97), AAAI Press, Menlo, CA, USA, pp. 283-296, 1997.

[4] Li, W., Han, J. and Pei, J.: CMAR: accurate and efficient classification based on multiple class-association rules, In Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, pp. 369-376, 2001.

[5] Coenen, F., Leng, P. and Zhang, Lu.: Threshold Tuning for Improved Classification Association Rule Mining, In Lecture Notes in Artificial Intelligence, Vol. 3518: Advances in Knowledge Discovery and Data Mining - PAKDD'05, pp. 216-225, 2005.

[6] Coenen, F. and Leng, P.: The effect of threshold values on association rule based classification accuracy, In Data Knowl. Eng., Vol. 60, Number 2, pp. 345-360, 2007.

[7] Yin, X. and Han, J.: CPAR: Classification based on Predictive Association Rules, In Proceedings of the SIAM International Conference on Data Mining, 2003.

[8] Thabtah, F., Cowling, P. and Peng, Y.: MCAR: multi-class classification based on association rule, In Computer Systems and Applications, 2005. The 3rd ACS/IEEE International Conference on, pp. 33+, 2005.

[9] Wang, Y.J., Xin, Q. and Coenen, F.: A Novel Rule Weighting Approach in Classification Association Rule Mining, In Data Mining Workshops, International Conference on, pp. 271-276, 2007.

[10] Shidara, Y., Kudo, M. and Nakamura, A.: Classification Based on Consistent Itemset Rules, In Trans. MLDM, Vol. 1, Number 1, pp. 17-30, 2008.

[11] Coenen, F. and Leng, P.: An Evaluation of Approaches to Classification Rule Selection, In ICDM'04, Proceedings of the Fourth IEEE International Conference on Data Mining, pp. 359-362, 2004.

[12] Wang, Y.J., Xin, Q. and Coenen, F.: Hybrid Rule Ordering in Classification Association Rule Mining, In Trans. MLDM, Vol. 1, Number 1, pp. 1-15, 2008.

[13] Liu, B., Hsu, W. and Ma, Y.: Integrating classification and association rule mining, In KDD'98, pp. 80-86, 1998.

[14] Wang, Y.J., Xin, Q. and Coenen, F.: A Novel Rule Ordering Approach in Classification Association Rule Mining, In MLDM '07: Proceedings of the 5th international conference on Machine Learning and Data Mining in Pattern Recognition, Leipzig, Germany, pp. 339-348, 2007.

[15] Quinlan, J.R.: C4.5: programs for machine learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[16] Berzal, F., Blanco, I., Sánchez D. and Vila, M.A.: Measuring the accuracy and interest of association rules: A new framework, In Intell. Data Anal., Vol. 6, Number 3, pp. 221-235, 2002.

[17] Quinlan, J.R. and Cameron-Jones, R.M.: FOIL: A Midterm Report, In Machine Learning: ECML-93, European Conference on Machine Learning, Proceedings, Vol. 667, pp. 3-20, Springer-Verlag, 1993.

[18] Hernández, R., Hernández, J., Carrasco, J.A. and Martínez J.F.: Algorithms for Mining Frequent Itemsets in Static and Dynamic Datasets, To appear in Journal Intelligence Data Analysis, Vol. 14, Number 3, 2009.

[19] Ahn, K.I. and Kim J.Y.: Efficient Mining of Frequent Itemsets and a Measure of Interest for Association Rule Mining, In JIKM, Vol. 3, Number 3, pp. 245-257, 2004.

[20] Lavraĉ, N., Flach, P. and Zupan, B.: Rule Evaluation Measures: A Unifying View, In Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP'99), pp. 174-185, 1999.

[21] Clark, P. and Boswell, R.: Rule Induction with CN2: Some Recent Improvements, In Proceedings of European Working Session on Learning (ESWL'91), Porto, Portugal, pp.151-163, 1991.

[22] Ali, K., Manganaris, S. and Srikant, R.: Partial classification using association rules, In Proceedings of the 3rd International Conference on Knowledge Discovery in Databases and Data Mining, pp.115-118, 1997.

[23] Buddeewong, S. and Kreesuradej, W.: A New Association Rule-Based Text Classifier Algorithm, In Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05), Washington, DC, USA, 2005.

[24] Wang, W., Wang, Y.J., Bañares-Alcántara, R., Cui, Z. and Coenen, F.: Application of Classification Association Rule Mining for Mammalian Mesenchymal Stem Cell Differentiation (ICDM'09), Leipzig, Germany, pp.51-61, 2009.

[25] Park, S.H., Reyes, J.A., Gilbert, D.R., Kim, J.W. and Kim, S.: Prediction of protein-protein interaction types using association rule based classification, In BMC Bioinformatics, Vol. 1, Number 10, 2009.

[26] Li, W.: Classification based on multiple association rules, M.Sc. Thesis, Simon Fraser University, 2001.

[27] Antonie, M., Zaiane, O.R. and Coman, A.: Associative Classifiers for Medical Images, In Lecture Notes in Artificial Intelligence 2797, Mining Multimedia and Complex Data, Springer Verlag, pp. 68-83, 2001.

[28] Thabtha, F., Cowling, P. and Peng, Y.: A New Multi-class, Multi-label Associative Classification Approach, In Proceedings of the 4th International Conference on Data Mining (ICDM'05), Brighton, UK, 2004.

[29] Quinlan, J.R. and Cameron-Jones, R.M.: FOIL: A Midterm Report, In Proceedings of the European Conference on Machine Learning (ECML'93), Vol. 667, pp. 3-20, 1993.

[30] Coenen, F., Leng, P. and Ahmed, S.: Data Structure for Association Rule Mining: T-Trees and P-Trees, In IEEE Transactions on Knowledge and Data Engineering, Vol. 16, Number 6, pp. 774-778, 2004.

[31] Blake, C.L. and Merz, C.J.: UCI Repository of Machine Learning Databases, In http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.

[32] Brin, S., Motwani, R., Ullman, J.D. and Tsur, S.: Dynamic Itemset Counting and Implication Rules for Market Basket Data, In Journal SIGMOD, Vol. 26, Number 2, pp. 255-264, 1997.

[33] Silverstein, C., Brin, S. and Motwani, R.: Beyond Market Baskets: Generalizing Association Rules to Dependence Rules, In Journal Data Mining and Knowledge Discovery, pp. 39-68, 1998.

[34] Piatetsky-Shapiro, G.: Discovery, Analysis, and Presentation of Strong Rules, In Book Knowledge Discovery in Databases, pp. 229-238, 1991.