Un Ambiente de Explotación de Información basado en la Integración de Agrupamiento, Inducción y Ponderación Bayesiana de Reglas

G. Schulz[1], E. Fernández[1,2], H. Merlino[1,2], D. Rodríguez[2], P. Britos[2,1], R. García-Martínez[2,1]
[1]Laboratorio de Sistemas Inteligentes, Facultad de Ingeniería, Universidad de Buenos Aires.
Paseo Colón 850 4to Piso. Ala Sur. (1063) Capital Federal, ARGENTINA.
[2]Centro de Ingeniería de Software Ingeniería del Conocimiento. Escuela de Postgrado. ITBA
25 de Mayo 444 – 6to. Piso. Capital Federal, República Argentina
rgm@itba.edu.ar

Resumen

Actualmente no existe un escenario que integre las funciones de clasificación de instancias, selección y ponderación de reglas, y por lo tanto utilizar a cada una de estas funciones como complemento uno del otro, para lograr una profunda y completa investigación de las características de las poblaciones que se desean estudiar. Esta falencia hace que cada vez que se quiera, por ejemplo, extraer las reglas de producción que dan como consecuencia la clasificación de una población, se necesite primero clasificar a los individuos de una población en un escenario de clasificación, para luego ingresar a estos individuos clasificados en un escenario diferente, capaz de inducir y extraer las reglas. Aquí se propone desarrollar un ambiente capaz de integrar las tres funciones.

Palabras Claves: *inducción de reglas, clasificación automática, elección automática de reglas, integración de inducción y ponderación.*

Abstract

At the moment it does not exist a scenario able to integrate the mechanisms of classification of instances, selection and ponderación of rules, and therefore to use to each one of these mechanisms as complement one of the other, to obtain a deep and complete investigation of the characteristics of the populations that are desired to study. This does that whenever it is wanted, for example, to extract the production rules that gives the classification of a population, is needed first to classify the individuals of the population in a classification scenario, then to enter these individuals classified in a diferent scenario, able to induce and to extract the rules. Here we propose to develop a tool able to integrate the three mechanisms.

1. Introducción

Existen numerosos ambientes que utilizadas en forma exitosa tanto para clasificar a una población de individuos, para inducir reglas inherentes a las características de una población o para ponderar reglas. Sistemas que utilizan a las redes neuronales son un ejemplo de eso, ya que dependiendo de la arquitectura de redes que utilicen, se comportan muy bien como clasificadores de elementos de un dominio; los sistemas que implementan árboles de decisión tales como ID3 [1] o C4.5 [2], por otro lado, son también muy comunes en lo que se refiere a la extracción de reglas de dominios o que utilizan a las redes Bayesianas como modelos de ponderación de reglas.

A continuación se relacionan varios de los software actualmente disponibles en el mercado, junto con una pequeña reseña de las funciones que proveen y de las técnicas utilizadas para brindar esas características:

- **AC^2:** Es un ambiente de data mining diseñado para usuarios conocedores de la materia. *AC*2 tiene un modelado grafico orientado a objetos y librerías en C y C++. Soporta la edición interactiva del árbol que se genera. Se comporta como una librería multiplataforma de funciones de data mining. Provee como funciones: clusterización, clasificación, predicción, segmentación. Utiliza como técnica árboles de decisión [3].
- **AnswerTree:** Es un ambiente de SPSS utilizado para construir árboles de decisión. Como ambiente de data mining apunta perfilar a grupos para la comercialización y las ventas. Utiliza cuatro algoritmos de árboles de decisión. Incluidos están dos algoritmos CHAID, los cuales SPSS ha extendido para manejar categorización nominal, ordinal y variables continuas dependientes. Provee como funciones: Clasificación. Utiliza como técnicas: Árboles de decisión (CHAID, CHAID Exhaustivo, C&RT (variación de CART), QUEST). [4]

- **CART:** Es un ambiente de árbol de decisión que utiliza el algoritmo CART. Para poder manejar la falta de información, los datos son manejados a través de reglas de backup que no siempre asumen que todos los datos de un atributo incierto es el mismo. Se utilizan siete criterios diferentes de splitting (incluyendo el Gini). Debido al uso del motor de traducción de datos, *DBMS/Copy*, se pueden utilizar datos de diferentes tipos de formato (incluyendo Excel, Informix, Lotus, Oracle). Provee como funciones: Clasificación. Utiliza como técnicas: Árboles de decisión (CART). [5], [6].
- **Clementine:** Utiliza iconos descriptivos como interfaz, el usuario crea descripciones de flujos de datos de las funciones que se realizarán. Cada icono representa un paso en el proceso total de minería de datos. Existen incluidos iconos para funciones tales como el acceso a datos, preparación de datos, visualización y modelado. Para asistir a la creación de secuencias, Clementine utiliza Capri. Además puede utilizar grandes conjuntos de datos usando un modelo de cliente/ servidor. Cuando es posible, el servidor convierte peticiones del acceso a los datos en las consultas SQL, que pueden entonces tener acceso a una base de datos emparentada. Provee como funciones: Reglas de asociación, clasificación, clusterización, análisis de factor, pronostico, predicción. Utiliza como técnicas: Apriori, BIRCH, CARMA, árboles de decisión (C5.0, C&RT variación de CART), clusterización Kmeans, redes neuronales (Kohonen, MLP, RBFN), regresión (lineal, logística) inducción de reglas (C5.0, GRI). [7]
- **Elvira:** El programa Elvira está destinado a la edición y evaluación de modelos gráficos probabilistas, concretamente redes Bayesianas y diagramas de influencia. Elvira cuenta con un formato propio para la codificación de los modelos, un lector-intérprete para los modelos codificados, una interfaz gráfica para la construcción de redes, con opciones específicas para modelos canónicos (puertas OR, AND, MAX, etc.), algoritmos exactos y aproximados (estocásticos) de razonamiento tanto para variables discretas como continuas, métodos de explicación del razonamiento, algoritmos de toma de decisiones, aprendizaje de modelos a partir de bases de datos, fusión de redes, etc. [10]
- **Sipina**: Está diseñado especialmente para la inducción de árboles de decisión. Sipina es un software con el cual se puede extraer conocimiento de los datos. Sipina aprende tanto de datos cualitativos como cuantitativos, y produce un grafico enrejado. Los algoritmos que provee Sipina para la generación de árbol de decisión son: SIPINA, ID3, C4.5, CART, Chi2-link, Elisee, QR_MDL y WDTaiqm. [11]
- **Weka:** Contiene y se focaliza en algoritmos de clasificación, regresión, y clusterización de patrones. Weka es un software gratuito y opensource bajo la licencia al público en general del GNU (GLP). Las técnicas que utiliza son: Naïve Bayes, Nearest neighbor, Linear models, OneR, Decision trees, Covering rules, K-means, EM, Cobweb. [8]

2. Problema a resolver

El problema o la falencia de los ambientes anteriormente detallados es que ninguno de ellos logra integrar y complementar las tres funciones en su implementación. Esto hace que cada vez que se quiera, por ejemplo, extraer las reglas que dan como consecuencia la clasificación de una población, se necesite primero clasificar a los individuos de una población en un escenario de clasificación X, para luego ingresar a estos individuos clasificados en un escenario diferente, capaz de inducir y extraer las reglas. Lo mismo ocurriría si se necesita ponderar estas reglas obtenidas. En la Figura 1 se muestra un posible escenario de lo arriba planteado. Allí se observan que son necesarios tres escenarios para poder extraer las reglas inducidas de clasificación.

1) **Escenario de Clasificación:** Recibe como entrada los datos a clasificar. Su función es la de clasificar a esos datos. La salida da como resultado los datos clasificados, en formato A.
2) **Escenario de Transformación** de Datos: Recibe como entrada datos clasificados en un formato A. Su función será la de transformar esos datos que están en formato A al formato B, para que sean entendidos por el escenario 3.
3) **Escenario de Selección de Reglas:** Recibe como entrada los datos clasificados en formato B. Su función es la de inducir las reglas que dieron origen a la clasificación. Su salida son el conjunto de reglas inducidas en formato B.

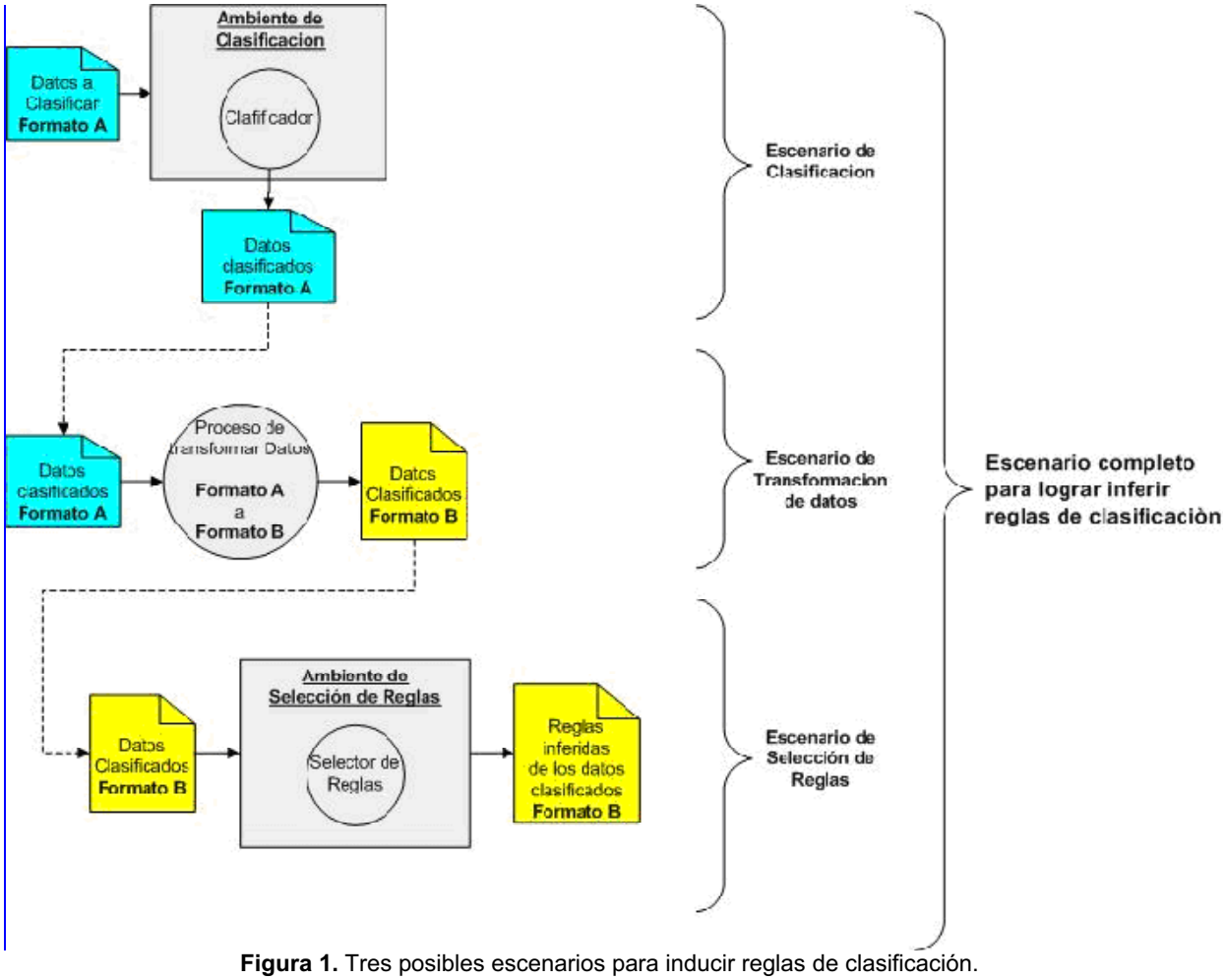


Figura 1. Tres posibles escenarios para inducir reglas de clasificación.

3. Solución

Lo que se plantea con este trabajo es desarrollar un ambiente que provea las funciones de:

- clasificar a una población.
- inducir las reglas que determinan que un determinado individuo pertenezca a cierta clase
- determinar la probabilidad de ocurrencia de una regla (ponderar).

Además esas capacidades, el ambiente va a tener que ser capaz de integrar y complementar a cada una de ellas, logrando que cada una sea el complemento de la otra. Cada uno de las funciones complementará su trabajo con el trabajo de las otras funciones. Así va a ser posible clasificar a una población, luego extraer o inducir las reglas que dieron origen a dicha clasificación para finalmente ponderar dichas reglas, para obtener la probabilidad de ocurrencia de cada una de las reglas antes inducidas. Para lograr implementar cada uno de estos funciones, se van a utilizar redes neuronales denominadas mapas autoorganizados [9] para la clasificación, mediante árboles de decisión como lo son los ID3 [1] se buscará inducir las reglas de clasificación, y se van a utilizar mecanismos probabilísticos, como lo son las Redes Bayesianas, para determinar la probabilidad de ocurrencia de una regla. Para lograr la integración de estas funciones, en la Figura 2 se muestra el flujo de información existente dentro del ambiente. En la figura se observan varios procesos que conforman este flujo. A continuación se detalla cada proceso:

- **Proceso 0. Ingresar población:** El usuario se limita a determinar cual va a ser el archivo que representa a la población que se va a procesar. Una vez que el usuario selecciona el archivo, el ambiente automáticamente intenta buscar dentro del mismo directorio donde se encuentra este archivo población, el archivo esquema que representa el modelado de esta población. Para ello busca un archivo con el mismo nombre que el de la población, pero con extensión xsd en lugar de xml. Si no encuentra al archivo esquema, entonces produce un error informando de la no existencia de este archivo. Si el ambiente encuentra el archivo xsd, entonces se procesan cada uno de los dos archivos, guardándose en

distintas instancias de objetos los datos característicos de cada uno de estos archivos. A cada una de estas instancias llamaremos *Población* y *Modelo* respectivamente.

- **Proceso 1. Validar población:** En esta etapa se realiza la validación del dominio o población que ingresó al ambiente. Para ello lo primero que se hace es verificar que cada uno de los individuos de la población defina los atributos especificados en el *Modelo*, y que los tipos de datos de estos atributos sean válidos de acuerdo a lo que especifica este *Modelo*. Una vez que se comprueba que lo anterior es correcto, se recorre uno a uno los individuos del objeto Población, y dentro de cada individuo se evalúa que cada uno de los atributos que lo caracterizan tenga un valor valido, de acuerdo a lo que se especifica en *Modelo*. En caso de encontrase alguna inconsistencia en los datos de la población, el sistema informará mediante un mensaje el motivo por el cual no se pudo realizar la validación.

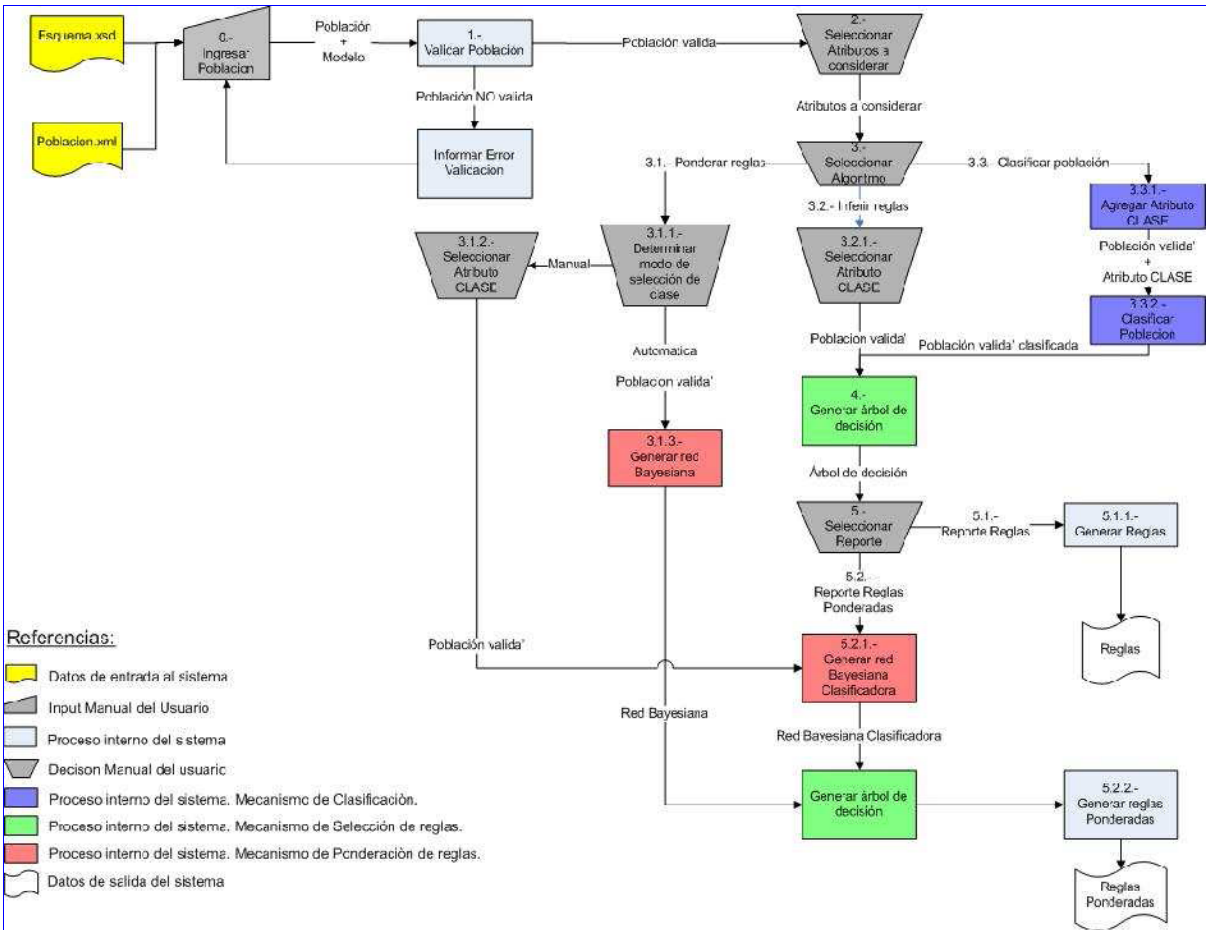


Figura 2. Flujo de procesos dentro del ambiente.

- **Proceso 2. Seleccionar atributos a considerar:** El usuario selecciona, del total de atributos que caracterizan a la población, un subconjunto de estos atributos con los cuales desea que se realice el análisis de la población en estudio. Estos atributos son los que se considerarán de ahora en más en todo el proceso, y determinarán a lo que llamaremos Población valida'. Básicamente esta Población valida' estará compuesta de los mismos individuos que la Población válida, solamente que estos individuos serán determinados por un subconjunto de atributos, y no necesariamente por el total. Supongamos que los atributos Edad, Peso y Altura son los atributos que caracterizan a la población y el usuario elige como atributos a considerar sólo Edad y Altura. De esta manera, lo que llamamos Población valida' serán individuos determinados solamente por estos dos atributos.
- **Proceso 3. Seleccionar algoritmo:** Este es uno de los procesos donde la decisión del usuario es fundamental para la continuación del flujo y procesos del ambiente. Aquí el usuario decide que algoritmo va a utilizar para continuar con el estudio de la población. Los posibles algoritmos a

elegir son los siguientes: Ponderación de reglas, Inducir reglas o Clasificar población.

- **Proceso 3.1. Ponderación de reglas:** Al elegir este algoritmo, el usuario está determinando que el único proceso que necesita realizarle a la población es la ponderación de reglas de decisión que dieron origen a la clasificación de la población. Obviamente al elegir este algoritmo, se presupone que la población en estudio es una población que ingreso al mismo ya clasificada. El usuario va a tener la opción de elegir que atributo es el que determina el atributo clase, o si va a ser la propia del ambiente la encargada de seleccionar este atributo clase. Una vez que se genere la red Bayesiana, para poder determinar cuales son las reglas a inducir, va a ser necesario que el ambiente genere el árbol de decisión. Este punto es transparente al usuario, pero necesario para poder determinar cuales son las reglas.
  - **Proceso 3.1.1. Determinar modo de selección de clase:** El usuario determina si la selección del atributo clase para el procesamiento de la red Bayesiana la deberá hacer automáticamente el sistema, o va a ser el propio usuario el que determinará cual de los atributos que conforman a la población valida' será el atributo clase. Si la selección del atributo clase la deberá hacer el sistema, entonces la red Bayesiana que va a generar el sistema va a ser una red Bayesiana tradicional, y el propio proceso de generación de esta red determinará, como consecuencia de este proceso, cual es este atributo clase. En cambio, si es el usuario quien selecciona qué atributo es el denominado atributo clase, entonces la red Bayesiana que se generará será una red Bayesiana de clasificación.
  - **Proceso 3.1.2. Seleccionar atributo clase:** Si en el proceso 3.1.1 el usuario selecciono que manualmente iba a determinar que atributo sería en atributo clase, en este proceso deberá seleccionar del subconjunto de atributos que caracterizan a la población valida' cual de ellos es el atributo clase. A partir de esta elección, el sistema deberá generar una red Bayesiana de clasificación, cuyo atributo clase es precisamente el atributo seleccionado por el usuario.
  - **Proceso 3.1.3. Generar red bayesiana: El** sistema genera una red bayesiana, valiéndose de la población valida' como datos de entrada para el proceso de entrenamiento y testeo de la red que generará. Será el ambiente, luego de generada la red Bayesiana, el que deberá determinar, según las características de la red que genere, que atributo se determinó como atributo clase.
- **Proceso 3.2. Inducir reglas:** Al elegir este algoritmo, la lectura que debemos hacer es que la población que ha ingresado al ambiente es una población ya clasificada, por lo que la necesidad del usuario recae en lograr información sobre aspectos que no tienen que ver con una clusterización de la población, sino con la de lograr determinar las reglas de decisión que dieron por origen la clasificación de esos individuos.
  - **Proceso 3.2.1. Seleccionar atributo clase:** Como la población que se ha ingresado al ambiente es ya una población clasificada, hay que definirle al ambiente cual de todos los atributos a considerar de la población es el que determina a que clase pertenece cada individuo. Esto lo determina el usuario.
- **Proceso 3.3. Clasificar la población:** Al elegir este algoritmo, lo que está planteando el usuario es una necesidad de clusterizar primero a la población, entendiéndose con esto que la población no tiene determinado ningún atributo que describa a que clase pertenece cada individuo. Esto significa que será el ambiente el encargado de realizar esta tarea, y lo hará mediante un algoritmo que no necesita de ninguna supervisión, por lo que el ambiente asume la total responsabilidad de la tarea de clusterizar a la población.
  - **Proceso 3.3.1. Agregar atributo clase:** En este proceso es el sistema el que agrega un nuevo atributo, denominado CLASE, al conjunto ya existente de atributos característicos de la población. El valor que tome este nuevo atributo es el que determinara a que clase pertenecerá cada uno de los individuos, una vez realizada la clasificación.
  - **Proceso 3.3.2. Clasificar población**: En este proceso se clusteriza a la población, determinándose el valor que tomará, para cada uno de los individuos, el atributo CLASE. La cantidad de clases en la que el ambiente intentará clasificar a los individuos es un valor que el propio usuario del ambiente determinará. La forma con la que se implementa este proceso de clusterización es mediante la utilización redes neuronales denominadas de aprendizaje competitivo y cooperativo. Con este tipo de aprendizaje se pretende

que cuando se presente a la red cierta información de entrada, solo una de las neuronas de salida de la red se active o alcance su valor de respuesta máximo. Es por eso que las neuronas compiten para activarse, quedando finalmente una como neurona ganadora, mientras que el resto quedan anuladas. Los individuos con características similares son clasificados formando parte de la misma categoría y por lo tanto deben activar la misma neurona de salida.

- **Proceso 4. Generar árbol de decisión:** Las instancias del dominio o población con las clases a las que pertenecen son presentadas al ambiente, quien como consecuencia de realizar la tarea de inducción, generará un árbol de decisión.
- **Proceso 5. Seleccionar Reporte:** El usuario simplemente elige que tipo de reporte quiere obtener del ambiente. Puede optar por el Reporte de Reglas, donde el ambiente solamente presentará las reglas que dieron origen a la clasificación, o puede elegir el Reporte de Reglas Ponderadas, donde el ambiente además de presentar las reglas, también determinará la probabilidad de ocurrencia para cada una de esas reglas.
  - **Proceso 5.1. Reporte de Reglas:**
    - **Proceso 5.1.1. Generar Reglas:** El árbol de decisión es recorrido desde la raíz hasta cada una de las hojas, y se generarán las reglas de decisión interpretando o mapeando cada bifurcación del árbol con su respectivo atributo y valor que la bifurcación tome. Las reglas generadas serán del estilo.

```
SI Atributo1 = valor1
   Y Atributo2 = valor2
   Y
   …
   Y AtributoN = valorN
ENTONCES Clase = clase1.
```

  - **Proceso 5.2. Reporte de Reglas Ponderadas:**
    - **Proceso 5.2.1. Generar red Bayesiana clasificadora:** Las instancias del dominio o población clasificadas son presentadas al ambiente. El ambiente utiliza estas instancias como datos de entrenamiento para generar, mediante un algoritmo de entrenamiento supervisado, de una red Bayesiana clasificadora. El tipo de algoritmo a utilizar para el entrenamiento de la red bayesiana es un dato que lo determina el usuario del ambiente.
    - **Proceso 5.2.1. Generar reglas ponderadas:** El árbol de decisión generado a partir de las instancias clasificadas es recorrido por el procesador de reglas, el cual generará las reglas de decisión interpretando o mapeando cada bifurcación del árbol con su respectivo atributo y valor que éste tome en la bifurcación. Para cada una de estas reglas, utilizará a la red bayesiana para poder determinar la probabilidad de ocurrencia de esta regla. Las reglas generadas serán del estilo.

```
SI Atributo1 = valor1
   Y Atributo2 = valor2
   Y
   …
   Y
   AtributoN = valorN
ENTONCES Clase = clase1.
PROBABILIDAD % de
probabilidad de
ocurrencia de la regla
inducida
```

4. Caso de estudio

Se realizó una experimentación con dos bases de datos tomadas como casos de estudio, y se compararon los resultados para cada una de las funciones que provee nuestro ambiente integrado con otros ambientes.

Para realizar esta experimentación, se utilizaron bases de datos obtenidas del *UCI Machine Learning Repository* del Departamento de Información y Ciencias de la Computación de la Universidad de California.

A continuación, en la tabla 1, se resumen las características de las bases de datos utilizadas.

Tabla 1. Bases de datos utilizadas.

Base de datos	Atributos	Instancias	Descripción de la base de datos
Iris	5	150	Contiene 4 atributos numéricos y un atributo nominal que determina la clase de la instancia.
Zoo	18	101	Contiene 16 atributos booleanos y uno numérico que definen diferentes animales. El atributo "tipo" define el atributo clase.

4.1. Clasificador de instancias

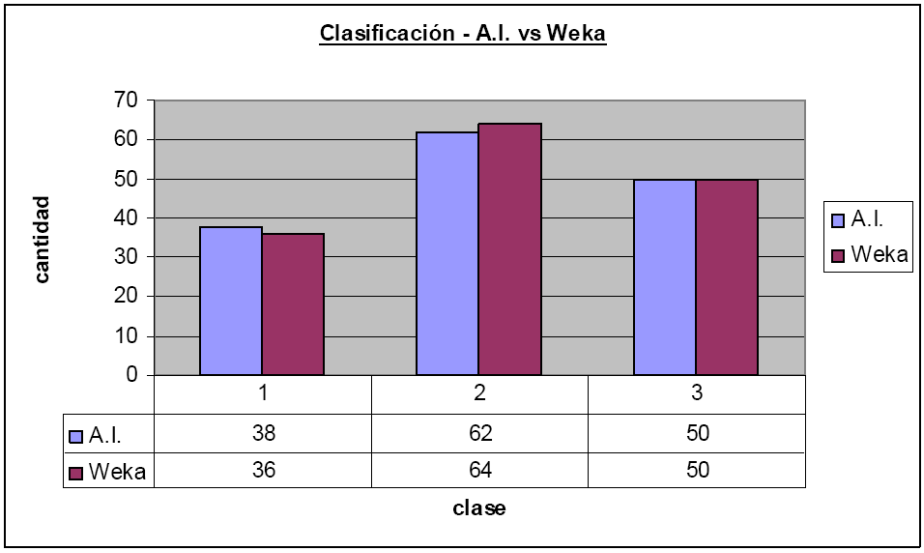
Se realizó la clasificación de instancias de cada una de las poblaciones descriptas en la Tabla 1. Se utilizó el ambiente Weka [8] para realizar la comparación.

A continuación se detalla un resumen de las pruebas realizadas y de los resultados obtenidos para cada base de datos.

- **Iris**

  Se configuraron ambos ambientes para que agrupasen los datos en 3 diferentes grupos.

  En la figura 3 se observa la comparación de la agrupación realizada por ambos ambientes. En la parte interior del gráfico, se detalla la clasificación realizada por nuestro ambiente integrado, mientras que en la parte exterior del grafico se observa la clasificación realizada por el ambiente Weka [8].



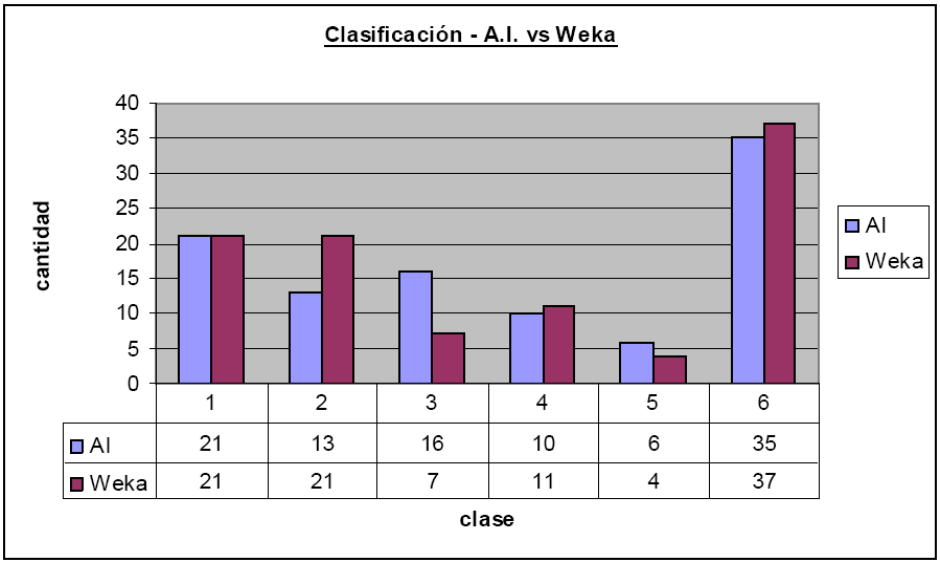
	1	2	3
A.I.	38	62	50
Weka	36	64	50

Figura 3. Clasificación de instancias – Iris.

- **Zoo**

  Se configuraron ambos ambientes para que agrupasen los datos en 6 diferentes grupos.

  En la figura 4 se observa la comparación de la agrupación realizada por ambos ambientes. En la parte interior del gráfico, se detalla la clasificación realizada por nuestro ambiente integrado, mientras que en la parte exterior del grafico se observa la clasificación realizada por el ambiente Weka [8].



	1	2	3	4	5	6
AI	21	13	16	10	6	35
Weka	21	21	7	11	4	37

Figura 4. Clasificación de instancias – Zoo.

4.2. Inducción de reglas

Se utilizó el ambiente Sipina [11] para realizar una comparación con nuestro ambiente integrado,

respecto al funcionamiento del mecanismo de inducción de reglas.

- **Iris**

  Con esta base de datos, nuestro ambiente integrado generó más reglas y reglas más especificas que el ambiente Sipina [11]. Se utilizaron entonces las reglas obtenidas por nuestro ambiente integrado y se obtuvo la confianza de cada de esas reglas en ambos ambientes. En la figura 5 se muestra el gráfico comparativo.

  Debido a que las reglas obtenidas en nuestro ambiente integrado eran reglas más especificas, se observa un mejor comportamiento de este ambiente en cuanto al porcentaje de confianza de cada regla inducida.

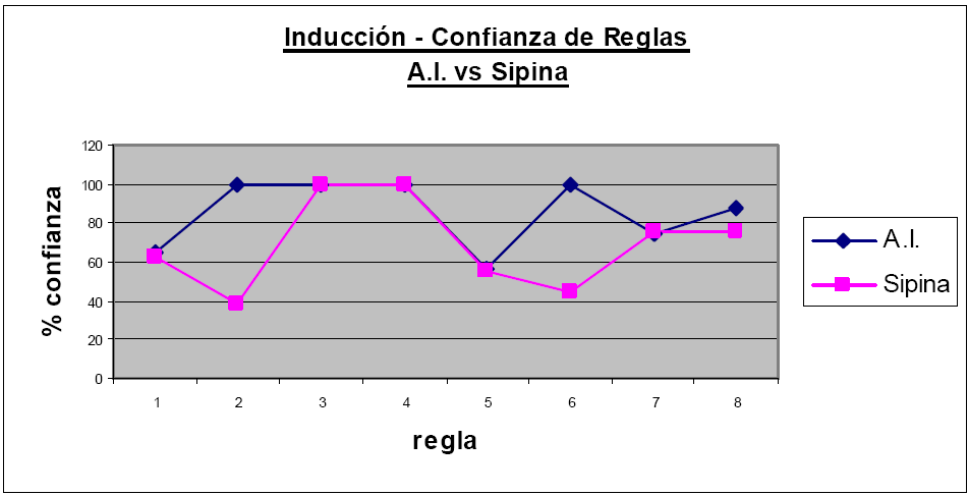


Figura 5. Inducción de reglas – Iris.

- **Zoo**

  Con esta base de datos, nuestro ambiente integrado también generó más reglas de inducción que el ambiente Sipina [11], y al igual que lo ocurrido con la base de datos Iris, las reglas generadas por nuestro ambiente integrado fueron reglas más específicas. Se utilizaron las reglas obtenidas por nuestro ambiente integrado para obtener la confianza de cada una de esas reglas en ambos ambientes. En la figura 6 se observa la relación entre la confianza de cada una de esas reglas en ambos ambientes.

  Nuevamente se observa un mejor comportamiento de nuestro ambiente integrado en cuanto al porcentaje de confianza de cada regla inducida.

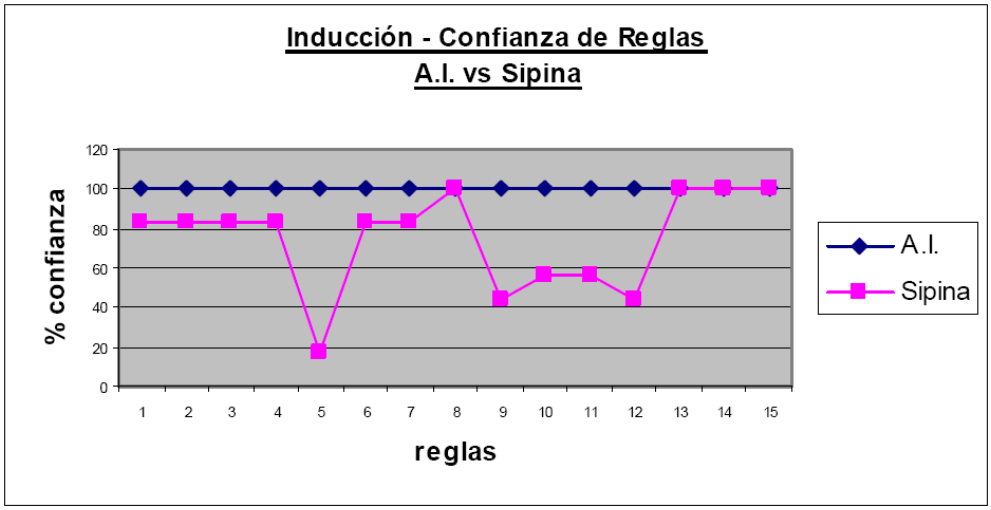


Figura 6. Inducción de reglas – Zoo.

4.3. Ponderación de reglas

Se utilizó el ambiente Elvira [10] para realizar la comparación con nuestro ambiente integrado.

Se tomaron todas las reglas generadas por el árbol de decisión para cada una de las bases de datos en estudio, y se calculó el porcentaje de ocurrencia de cada una de esas reglas en ambos ambientes de comparación.

En las figuras 7 y 8 se muestra gráficamente la comparación entre los resultados de la probabilidad de ocurrencia para cada una de las reglas inducidas por ambos ambientes.

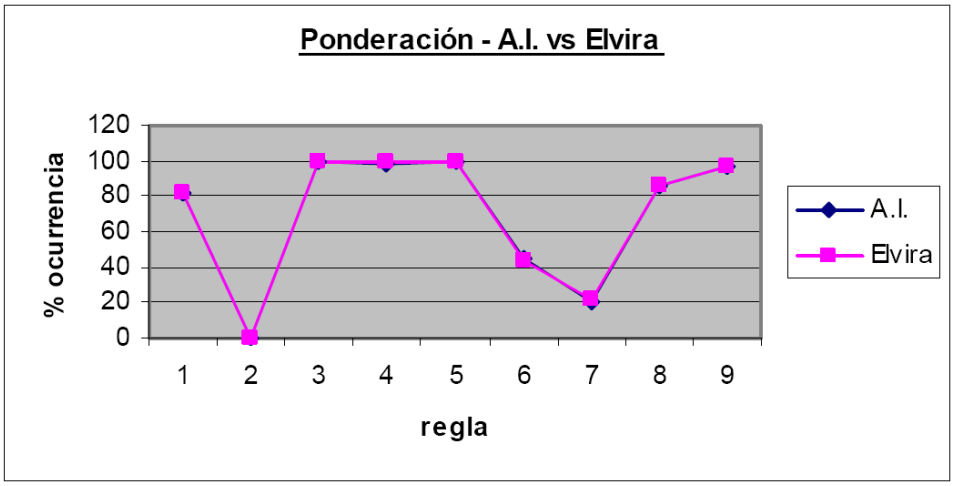


Figura 7. Ponderación de reglas – Iris.

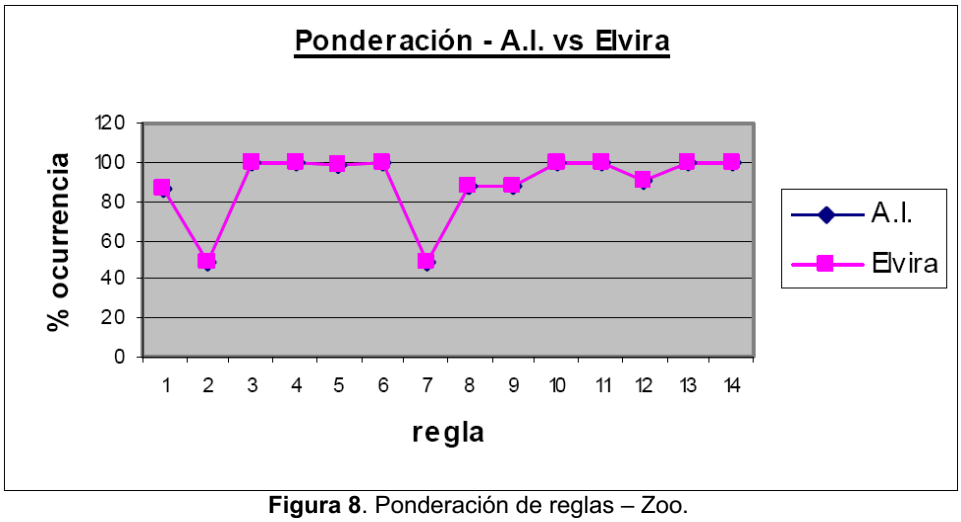


Figura 8. Ponderación de reglas – Zoo.

Los resultados obtenidos para las dos bases de datos comparadas muestran que el comportamiento de los dos ambientes es prácticamente el mismo, obteniéndose un valor muy parecido para la probabilidad de ocurrencia de cada una de las reglas.

5. Conclusiones

En base a los resultados experimentales obtenidos en la clasificación de diversas poblaciones, en la inducción de reglas de producción o en la ponderación de la ocurrencia de éstas reglas, podemos concluir que el ambiente desarrollado se comporta en forma similar a otros ambientes existentes en el mercado. El aspecto más importante de este ambiente es que, en contraposición a los ambientes utilizados para la comparación, presenta en su funcionalidad la integración de las tres funciones, aspecto que los ambientes no tienen. Esta característica hace que el ambiente provea una funcionalidad completa para el estudio de las características de una población de individuos, que de acuerdo a las necesidades del usuario, pueden ser las siguientes:

- Clasificar una población y obtener las reglas de producción que dieron como origen a la clasificación.
- Clasificar una población y obtener la probabilidad de ocurrencia de cada regla de producción.

- Si se tiene una población ya clasificada, obtener las reglas de producción que dan como origen a la clasificación.
- Si se tiene una población ya clasificada, obtener la probabilidad de ocurrencia de cada regla de producción.
- Permite al usuario la elección de los atributos que se quieren considerar, y sólo utilizar esos atributos en el estudio de las características de la población.
- Permite sólo seleccionar aquellas reglas de producción con una probabilidad de ocurrencia mayor a cierto valor deseado.
- Importancia de poder contar con el Standard XML para la representación de la población.

6. Referencias

[1] J. Ross Quinlan. 1986. *Induction of decision trees*. Machine Learning.
[2] J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning.* Machine Learning.
[3] ISoft. 2007. *AC2*. www.alice-soft.com/html/prod_ac2.htm. Vigente al 30/06/2007
[4] SPSS. 2007. *Answer.*/www.spss.com/la/productos/answer-tree/answer.htm. Vigente al 30/06/2007
[5] Salford Systems. 2007. *CART.* www.salfordsystems.com/cart.php. Vigente al 30/06/2007
[6] Breiman L, Friedman J, Olshen R y Stone C. 1984. *Classification and regression trees.* Machine Learning.
[7] SPSS. 2007. *Clementine.* www.spss.com/clementine/. Vigente al 30/06/2007
[8] Ian H. Witten y Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. 2nd Edition. Morgan Kaufmann. San Francisco.
[9] Kohonen, T. 2001. *Self-Organizing Maps*. 3rd Edition. Springer
[10] Proyecto Elvira, Universidad de Granada. Web: http://www.ia. uned.es/~elvira/
[11] University of Lyon. Francia. Sipina. Web: http://eric.univ-lyon2.fr/~ricco/sipina.html.