Universidad Central “Marta Abreu” de Las Villas

Facultad de Matemática, Física y Computación

TRABAJO DE DIPLOMA

Herramienta computacional para hacer inferencias Bayesianas, aplicaciones a Bioinformática

Autores: Anay Rodríguez García

Yasmany Garcia Mondeja

Yasser Díaz Villasuso

Tutores: MSc. María del Carmen Chávez

Dra. Gladys Casas Cardoso

Seminario: Bioinformática

2006

Resumen

La inferencia Bayesiana es el único modo consistente de razonar ante la presencia de incertidumbre, la probabilidad es una medida intuitiva de la incertidumbre. La independencia condicional entre los rasgos de una población simplifica considerablemente los modelos de dependencia. Un modelo que permite disminuir grandemente el espacio probabilístico y a su vez inferir tanto clases como atributos, lo son las redes Bayesianas, es por ello que se implementa una herramienta que desarrolle tales aplicaciones en el campo de la bioinformática.

Este trabajo proporciona un software que permite construir redes Bayesianas y hacer inferencias Bayesianas, con el objetivo de mejorar las posibilidades de predicción de propiedades funcionales o estructurales de secuencias de proteínas a partir de razonamientos probabilísticos, especialmente ante la presencia de información incompleta o con cierta incertidumbre.

Abstract

The Bayesian inference is the only consistent way to reason in the presence of uncertainty, the probability is an intuitive measurement of this uncertainty. Conditional independence between the characteristics of a population simplifies the dependency models considerably. A model that allows the probabilistic space to be diminished greatly and to infer classes as much as attributes, is the Bayesian networks, it's for that reason that a tool that develops such applications in the field of the bioinformatics is implemented.

This work provides software that allows the construction of Bayesian networks and to make Bayesian inferences, with the objective to improve the possibilities of prediction of functional or structural properties of protein sequences from probabilistic reasoning, especially before the presence of incomplete information or with certain uncertainty.

Índice

Introducción 1
CAPÍTULO I. REDES BAYESIANAS. CONCEPTOS 4
1.1 Inferencia Bayesiana 4
1.1.1 ¿Qué es la inferencia Bayesiana? 5
1.2 Redes Probabilísticas 6
1.2.1 Redes Bayesianas 6
1.2.1.1 Construcción de redes Bayesianas 7
1.2.1.2 Limitaciones del algoritmo de construcción de redes Bayesianas 8
1.3 Propagación en redes múltiplemente conexas 9
1.3.1 Propagación en redes Bayesianas utilizando un árbol de unión 11
1.3.1.1 Algoritmo de Propagación utilizando un árbol de unión 11
1.4 Construcción de modelos probabilísticos 12
1.5 Representación de los árboles de decisión 14
1.5.1 Árboles de decisión ID3 14
1.5.1.1 Complejidad 15
1.5.1.2 Fiabilidad 16
1.5.1.3 Capacidades y limitaciones del algoritmo ID3 16
1.6 Criterio de selección de atributos 17
1.6.1 Test de independencia de variables Chi-cuadrado 17
1.6.2 Entropía y ganancia de información 18
1.7 Formato de entrada de datos 19
1.8 Formato de Exportación de Resultados 19
1.8.1 XML (Extensible Markup Language) 20
1.8.1.1 Características del lenguaje XML 20
1.8.2 XMLBIF (eXtensible Markup Language Interchange Format for Bayesian Networks) 20
1.9 Consideraciones finales del capítulo 21
CAPÍTULO II. ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PARA HACER INFERENCIAS BAYESIANAS 22
2.1 Herramientas utilizadas 22
2.1.1 Programación Orientada a Objetos 22
2.2 Alcance de la investigación 23
2.3 Modelación del sistema 24
2.4 Implementación del sistema 26
2.5 Diagrama de transición de estados 31
2.6 Análisis de formato de ficheros para exportar resultados 33
2.6.1 Fichero XMLBIF 33
2.6.2 Fichero XML 33
2.7 Complejidad de algunos algoritmos implementados 36
2.8 Consideraciones finales del capítulo 37
CAPÍTULO III. MANUAL DE USUARIO 38
3.1 Requisitos para la explotación del sistema 38
3.1.2. Requerimientos de hardware 38
3.2 Acceso a la herramienta con el objetivo de hacer inferencias Bayesianas 38
3.2.1 Ficheros de entrada y edición de datos 40
3.2.2 Construcción de la red Bayesiana 41

3.3 Manual para el uso del menú de barras del sistema 45
3.4 Análisis de los resultados 50
Conclusiones 53
Recomendaciones 54
Referencias Bibliográficas 55
Bibliografía 56

Introducción

En Cuba, en los últimos años, es cada vez más creciente la tendencia a la aplicación de las nuevas tecnologías de la información a la vida diaria y en particular se observa un marcado interés en el área de la informatización. El presente trabajo se enmarca dentro de los esfuerzos en este sentido en el área de la bioinformática.

Se pretende mejorar los métodos de inteligencia artificial basados en probabilidades que se utilizan en el análisis de mutaciones de genes, la construcción de árboles filogenéticos y la búsqueda de técnicas más eficientes para el trabajo con redes Bayesianas.

La novedad se garantiza al querer enriquecer los resultados clásicos en este sentido incluyendo en los algoritmos la información que aportan las estructuras algebraicas del código genético con su enfoque desde el punto de vista de la mutabilidad de genes.

Se decide implementar un sistema capaz de permitirle al usuario inferir tanto clases como atributos, tomando datos iniciales de ficheros existentes.

Los resultados esperados son consecuencia de la ejecución de algoritmos de construcción y propagación de redes Bayesianas.

Esta herramienta muestra por pasos: la configuración del árbol de decisión que crea el software; la red Bayesiana; la tabla de probabilidades para cada nodo de la red y el resultado de la inferencia Bayesiana tomando como evidencia los atributos que el usuario seleccione.

Este sistema permite exportar los resultados utilizando formatos de ficheros de amplia utilización, que pueden ser cargados por otras aplicaciones.

Preguntas de investigación:

- ¿Qué métricas son necesarias para la selección de rasgos?
- ¿Cómo se obtienen las relaciones entre los atributos de la aplicación que se analiza?
- ¿Cómo se construye una red Bayesiana a partir de árboles de decisión?
- ¿Es posible utilizar la teoría de redes Bayesianas para lograr estudios de análisis de secuencias en bioinformática?

Objetivo general:

Desarrollar un sistema computacional para lograr el aprendizaje de redes Bayesianas desde datos y realizar inferencias en la misma, utilizando un lenguaje de programación de alto nivel y ejemplos de problemas relacionados con la Bioinformática.

Objetivos específicos:

- Analizar los requerimientos para la presente versión del software.
- Desarrollar el módulo de selección de atributos.
- Implementar el módulo de construcción de árboles de decisión.
- Implementar el módulo de construcción de la red Bayesiana.
- Implementar un algoritmo de inferencia en redes Bayesianas.
- Realizar aplicaciones bioinformáticas para validar la herramienta desarrollada y ofrecer resultados de interés específico en este campo.

El informe queda estructurado en 3 capítulos:

En el capítulo uno se describe un análisis teórico de los diferentes conceptos y algoritmos que son necesarios para la implementación del software como: árboles de decisión, redes Bayesianas, inferencias Bayesianas, algoritmo de propagación de evidencia utilizando árboles de unión. . Se describe información sobre los formatos de archivos que tendrá el sistema como exportación de resultados y los criterios de selección de atributos que inicialmente poseerá la herramienta, se decide utilizar inicialmente Chi-cuadrado y Entropía.

En el capítulo dos se presenta el análisis y diseño del sistema propuesto, se analizan las complejidades de los algoritmos empleados y se explica como se trabaja para mejorar esa complejidad.

El capítulo tres lo constituye el manual de usuario y aplicaciones, en el que se explica el funcionamiento de la herramienta computacional para hacer inferencias Bayesianas y los resultados al aplicar base de datos existentes.

El manual de usuario, donde se explican las utilidades del sistema, brinda la información necesaria para un uso eficiente del software.

Los resultados son utilizados para diagnosticar el posible estado de atributos o clases. En el campo de la Bioinformática se utiliza para mejorar las posibilidades de predicción de propiedades funcionales o estructurales de secuencias de proteínas a partir de razonamientos probabilísticos, especialmente ante la presencia de información incompleta o con cierta incertidumbre.

CAPÍTULO I. REDES BAYESIANAS. CONCEPTOS

Para el desarrollo del sistema fue necesario realizar un proceso de estudio de los algoritmos necesarios para la construcción de redes Bayesianas y la realización de inferencias Bayesianas, se hicieron comparaciones con softwares como: WEKA (Witten and Frank, 2005), ByShell (C. and L.O., 2002), SPSS .

1.1 Inferencia Bayesiana

Los experimentos genómicos generan ingentes cantidades de datos genéticos que plantean problemas de gestión y análisis, lo cual debe implicar la búsqueda de soluciones que deben encontrarse en el campo de la bioinformática mediante la revisión y adaptación de algoritmos y sistemas existentes en el campo de la ciencia de la computación, e incluso el diseño de nuevas aplicaciones. El objetivo que se persigue al aplicar estos algoritmos es la extracción de conocimiento biológico.

Un modelo que permite disminuir considerablemente el espacio probabilístico y a su vez, inferir tanto clases como atributos, es el de la red Bayesiana, es por ello que se hace necesario una herramienta que permita desarrollar tales aplicaciones en el campo de la bioinformática.

La inferencia Bayesiana es el único modo consistente de razonar ante la presencia de incertidumbre, la probabilidad es una medida intuitiva de esta. La independencia condicional entre los rasgos de una población, simplifica considerablemente los modelos de dependencia.

Las redes Bayesianas que son foco de estudio, están basadas en este modelo, por tanto codifican un conjunto de aseveraciones de independencia condicional. La topología de estas

redes, es obtenible por medio de técnicas de segmentación estadística (árboles de decisión con criterios energéticos) o en general técnicas de inducción antecedidas de un proceso de selección de atributos y la propagación de evidencias se puede llevar a cabo de forma exacta, con árboles de unión.(Buntine, 1996, Castillo et al., 1996)

1.1.1 ¿Qué es la inferencia Bayesiana?

La inferencia Bayesiana es un enfoque alternativo para el análisis estadístico de datos que, en buena medida, se contrapone a los métodos que proceden de lo que se ha denominado "estadística frecuentista" y que todos usamos con regularidad. Un elemento cardinal con que predominantemente opera este método alternativo es el manejo subjetivo, no frecuentista, del concepto de probabilidad.

Tanto en el ámbito epidemiológico como en el clínico es bien conocido el teorema de Bayes por su utilidad para la valoración de pruebas diagnósticas, sea para la toma de decisiones clínicas o para evaluar la pertinencia de implementar programas poblacionales de cribado. Por su conducto se puede expresar, por ejemplo, el valor predictivo que cabe atribuir a un resultado positivo de cierta prueba diagnóstica en función de las características intrínsecas de dicha prueba (su sensibilidad y su especificidad) y de la prevalencia de la enfermedad. Así se transforma o modifica la probabilidad a priori, (P [E]), de padecer la enfermedad (representada por su prevalencia) en la probabilidad a posteriori (valor predictivo), una vez observado el resultado (positivo) de la prueba diagnóstica.

El proceso intelectual asociado a la inferencia Bayesiana es mucho más coherente con el pensamiento usual del científico que el que ofrece el paradigma frecuentista. Los procedimientos bayesianos constituyen una tecnología emergente de procesamiento y análisis de información para la que cabe esperar una presencia cada vez más intensa en el campo de la aplicación de la estadística a la investigación clínica, epidemiológica y por qué no en la bioinformática.(Heckerman, 1996, Buntine, 1996) .

1.2 Redes Probabilísticas

En las dos últimas décadas se ha producido un importante desarrollo en el área de los sistemas expertos y, en particular, en los modelos basados en redes probabilísticas (redes Bayesianas y redes de Markov). Esta nueva metodología permite construir de forma eficiente e intuitiva la probabilidad conjunta asociada a un cierto modelo utilizando la potencia de los grafos para definir las relaciones de independencia existente entre las variables de un modelo dado. El grafo resultante define una factorización de la probabilidad conjunta que permite construir esta de forma rápida e intuitiva En este sentido, se analizan las bases teóricas que permiten el desarrollo de modelos de sistemas expertos definidos mediante redes o grafos (las redes de Markov y Bayesianas) y se presentan los métodos exactos y aproximados más importantes para actualizar el conocimiento a la luz de nueva evidencia. (Castillo et al., 1996)

1.2.1 Redes Bayesianas

Las redes Bayesianas están siendo utilizadas cada día más para la representación de conocimiento incierto en sistemas expertos. Las áreas de aplicación (por mencionar algunas) para resolver problemas de la vida diaria son: diagnóstico y pronóstico, visión automática, control de producción, recuperación de información, lenguaje natural, planeación, control, reconocimiento de voz y en casi cualquier área donde tenemos información incompleta e incierta.

Una red Bayesiana es un grafo acíclico dirigido en el que cada nodo representa una variable y cada arco una dependencia probabilística, en la cual se especifica la probabilidad condicional de cada variable dados sus padres. La variable a la que apunta el arco es dependiente (causa-efecto) de la que está en el origen de éste. La topología o estructura de la red nos da información sobre las dependencias probabilísticas entre las variables, pero también sobre las independencias condicionales de una variable (o conjunto de variables) dada otra variable(s). Dichas independencias, simplifican la representación del

conocimiento (menos parámetros) y el razonamiento (propagación de las probabilidades).(Heckerman, 1996, Castillo et al., 1996)

El obtener una red Bayesiana a partir de datos, es un proceso de aprendizaje que se divide en dos etapas: el aprendizaje estructural y el aprendizaje paramétrico. La primera de ellas consiste en obtener la estructura de la red Bayesiana, es decir, las relaciones de dependencia e independencia entre las variables involucradas. La segunda etapa tiene como finalidad obtener las probabilidades a priori y condicionales requeridas a partir de una estructura dada.

Definición 1.1**:** Una Red Bayesiana (RB) o Red Probabilística es un Grafo Acíclico Dirigido (GAD) G = (N, A), con variables aleatorias X y la distribución X | Ψ_X asociada a cada nodo X, siendo Ψ_X el conjunto de antecesores del nodo X. Por tanto, la red define la distribución multivariante:

$$\prod_{X \in N} P(X \mid \Psi X)$$

1.2.1.1 Construcción de redes Bayesianas

Los problemas clásicos a los que nos enfrentamos en la construcción de una red probabilista son tres, a saber:

a) Determinación de la estructura. La determinación de la estructura de una red consiste en encontrar su topología, es decir, las relaciones de dependencia entre las variables relevantes involucradas en un problema dado. La determinación de la topología de una red probabilista es, en muchos de los casos, proporcionada por el experto. Sin embargo, existen otros métodos que no extraen directamente del humano la topología sino de datos estadísticos. Más recientemente se han propuesto métodos que combinan estos dos enfoques. (Chávez et al., 1999)

b) Determinación de los parámetros. En una red probabilista, cada nodo tiene asociada una función de probabilidad (ya sea marginal o condicional). Para cada diferente estructura de una red probabilista, debemos determinar u obtener las distribuciones de probabilidad en cada nodo para esa estructura. (Chávez et al., 1999)

c) Propagación de probabilidades. Si tenemos la topología de una red y las distribuciones de probabilidad para cada nodo, entonces queremos determinar el cambio en estas probabilidades cuando los valores de algunas variables llegan a ser conocidos. El proceso de instanciar estas variables de entrada y propagar sus efectos a través de la red es lo que se llama propagación de probabilidades. (Castillo et al., 1996)

La red Bayesina creada será la unión de familias que conforman árboles de decisión ID3, creados con las dependencias de las variables y todos estos árboles unidos a la clase.

1.2.1.2 Limitaciones del algoritmo de construcción de redes Bayesianas

El algoritmo tiene varias limitantes que deben ser consideradas:

1) La prueba estadística se basa en un nivel de significancia (a=0.01 ó a=0.05 en el caso del test Chi-cuadrado), en cada prueba incondicional. Como se realizan muchas pruebas sucesivamente, el nivel de significancia global es mucho mayor que el nominal. Entonces en cada caso particular y con mayor razón cuando las muestras son pequeñas, debe revisarse el nivel de significancia real que debe emplearse.

2) La dirección de los arcos está determinada exclusivamente por el orden ancestral inducido por la ganancia en información principalmente por cada variable y no indica causalidad u orden temporal.

3) Si la muestra es pequeña los resultados estadísticos son endebles, en el sentido que la distribución de probabilidad de la estadística posiblemente está lejos de parecerse a la de la variable X^2. Por otra parte habrá muchos "huecos" en los datos, es decir, habrá muchas combinaciones de valores de las variables (x, y, z,...) para los cuales no se contará con

casos en la muestra. Esta situación también produce inestabilidad en los estimadores de las probabilidades y de las estadísticas. (Chávez et al., 1999)

1.3 Propagación en redes múltiplemente conexas

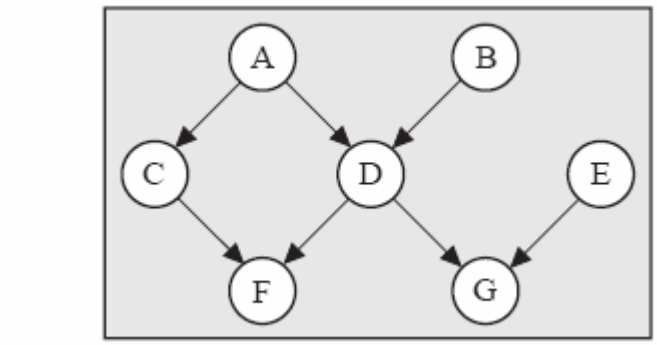


Figura 1.1 *Grafo múltiplemente conexo*

Dos de los métodos de propagación más importantes para este tipo de redes son los denominados métodos de condicionamiento y método de agrupamiento. La idea fundamental del método de propagación por condicionamiento es cortar los múltiples caminos entre los nodos mediante la asignación de valores a un conjunto reducido de variables contenidas en los bucles. De esta forma se tendrá un poliárbol en el cual se podrá aplicar el algoritmo de propagación para poliárboles. Por otra parte, el método de agrupamiento construye representaciones auxiliares, de estructura más simple, uniendo conjuntos de nodos del grafo original (por ejemplo, un árbol de unión). De esta forma se puede obtener un grafo con estructura de poliárbol, en el que pueden aplicarse la propagación de evidencia. (Castillo et al., 1996, Heckerman, 1996, Chávez et al., 1999)

La figura 1.2 muestra las probabilidades iniciales de los nodos cuando no se considera evidencia, y la figura 1.3 muestra las probabilidades actualizadas cuando se considera la evidencia D = 0. A partir de estas figuras, se puede ver que la evidencia no afecta al nodo *E* (la probabilidad marginal inicial coincide con la probabilidad condicionada actualizada). Sin embargo, la evidencia afecta de forma importante a algunos nodos como, por ejemplo,

a los nodos F y G. La estructura de dependencia contenida en el grafo permite determinar qué variables serán afectadas por la evidencia, pero no la magnitud en que esta influencia modifica las probabilidades de los nodos.

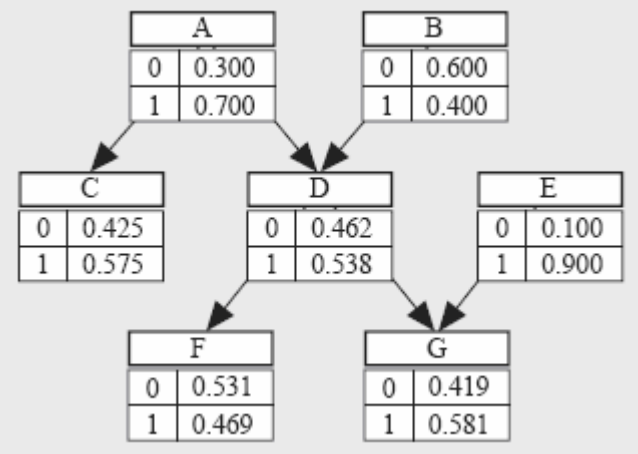


Figura 1.2 *Probabilidades marginales (iniciales) de los nodos*

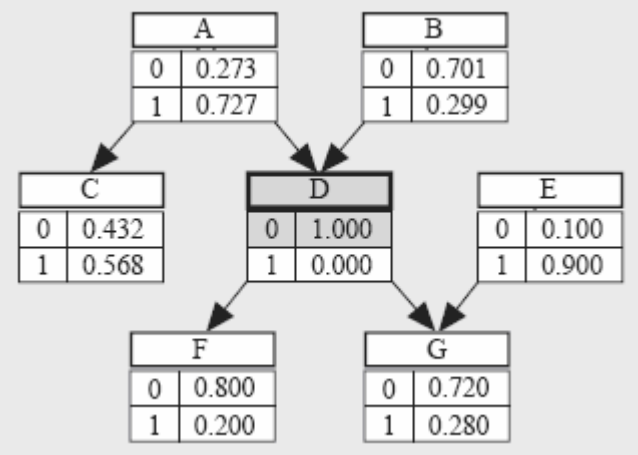


Figura 1.3 *Probabilidades condicionadas (actualizadas), dada la evidencia D = 0*

A pesar de que la complejidad del algoritmo de propagación en poliárboles es lineal en el tamaño de la red, el problema de la propagación de evidencia en redes Bayesianas múltiplemente conexas es un problema NP-complejo(Castillo et al., 1996). En general, tanto el método de condicionamiento, como el de agrupamiento plantean este problema de complejidad. Sin embargo, las características particulares de estos métodos hacen que, en ocasiones, uno de ellos sea más eficiente que el otro en redes con estructura particular. Sin

embargo, en general, ninguno de estos métodos es más eficiente que el otro, sino que son complementarios (Castillo et al., 1996). Este hecho ha motivado la aparición de algunos algoritmos mixtos que combinan las ventajas de ambos métodos.

1.3.1 Propagación en redes Bayesianas utilizando un árbol de unión

Este algoritmo recibe como datos de entrada una red Bayesiana sobre un conjunto de variables y una evidencia, obteniéndose como resultado la probabilidad de cada nodo de la red condicionada a la evidencia P(X|e)(Castillo et al., 1996).

En su implementación se hace inminente la conversión del grafo de la red Bayesiana a un grafo moralizado, triangular, seguidamente se construye un árbol de familias que será utilizado para definir la P (X|e) (Castillo et al., 1996).

El razonamiento probabilístico o propagación de probabilidades consiste en propagar de los efectos de la evidencia a través de la red para conocer la probabilidad a posteriori de las variables. La propagación consiste en darle valores a ciertas variables (evidencia), y obtener la probabilidad posterior de las demás variables dadas las variables conocidas (instanciadas).

1.3.1.1 Algoritmo de Propagación utilizando un árbol de unión

Datos: Una red Bayesiana (D, P) sobre un conjunto de variables X y una evidencia E = e.

Resultados: La función de probabilidad condicionada p (xi|e) para cada nodo Xi distinto del nodo que posee la evidencia.

Pasos del algoritmo:

1. Obtener un árbol de familias del grafo D. Sea C el conjunto de conglomerados resultante.

2. Asignar cada nodo Xi a un sólo conglomerado que contenga a su familia. Sea Ai el conjunto de nodos asignados al conglomerado Ci.

3. Para cada conglomerado Ci definir ψi (ci). Si Ai = φ, entonces definir ψi (ci) = 1.

4. Aplicar el Algoritmo de Propagación en redes de Markov utilizando un árbol de unión a la red de Markov (C, Ψ) y a la evidencia E = e para obtener las funciones de probabilidad condicionada de los nodos (Castillo et al., 1996)**.**.

1.4 Construcción de modelos probabilísticos

La construcción de un modelo probabilístico puede ser realizada en dos etapas:

1. Factorizar la función de probabilidad mediante un producto de funciones de probabilidad condicionada. Esta factorización puede obtenerse de tres formas distintas:
   (a) Utilizando grafos
   (b) Utilizando listas de relaciones de independencia
   (c) A partir de un conjunto de funciones de probabilidad condicionada (Castillo et al., 1996)

2. Estimar los parámetros de cada una de las funciones de probabilidad condicionada resultantes.

Este proceso se ilustra de modo esquemático en la figura 1.4, una línea continua de un rectángulo A hacia un rectángulo B significa que cada miembro de A es también un miembro de B, mientras que una línea discontinua significa que algunos, pero no necesariamente todos, los miembros de A son miembros de B. El camino más simple para definir un modelo probabilístico es comenzar con un grafo que se supone describe la estructura de dependencia e independencia de las variables. A continuación, el grafo puede utilizarse para construir una factorización de la función de probabilidad de las variables. De forma alternativa, también puede comenzarse con una lista de relaciones de independencia y, a partir de ella, obtener una factorización de la función de probabilidad. La factorización obtenida determina los parámetros necesarios para definir el modelo probabilístico. Una vez que estos parámetros han sido definidos, o estimados a partir de un conjunto de datos, la función de probabilidad que define el modelo probabilístico vendrá dada como el producto de las funciones de probabilidad condicionada resultantes.

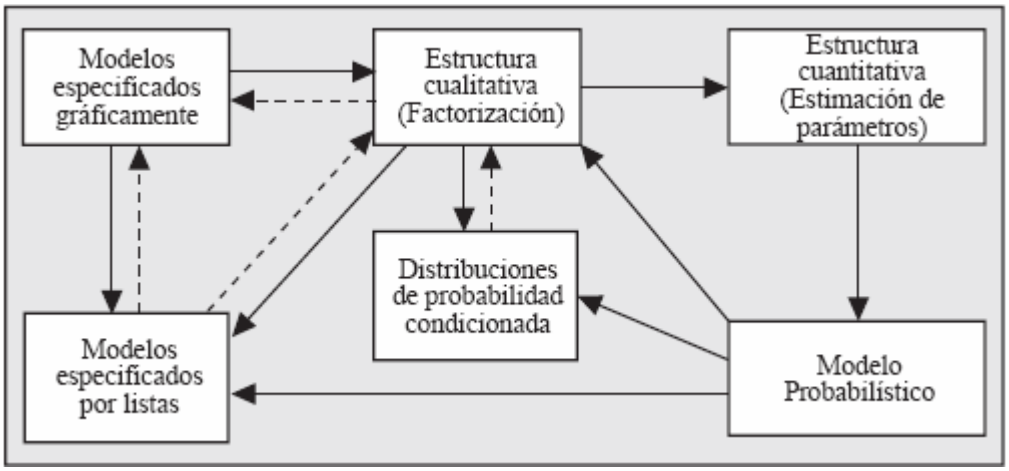


Figura 1.4 *Diagrama mostrando las formas alternativas de definir un modelo probabilístico*

Por otra parte, si se conoce la función de probabilidad que define un modelo probabilístico (que no es el caso habitual en la práctica), se puede seguir el camino inverso y obtener varias factorizaciones distintas (utilizando la regla de la cadena definida en (Castillo et al., 1996)**.** También se puede obtener la lista de independencias correspondiente al modelo, comprobando cuáles de todas las posibles relaciones de independencia de las variables son verificadas por la función de probabilidad. A partir del conjunto de independencias

obtenido, también puede construirse una factorización de la familia paramétrica que contiene a la función de probabilidad dada.(Castillo et al., 1996, Heckerman, 1996, Chávez et al., 1999)

1.5 Representación de los árboles de decisión

La figura 1.5 muestra un árbol de decisión típico. Cada nodo del árbol está conformado por un atributo y puede verse como la pregunta: ¿Qué valor tiene este atributo en el ejemplar a clasificar? Las ramas que salen de los nodos, corresponden a los posibles valores del atributo correspondiente. Un árbol de decisión clasifica a un ejemplar, filtrándolo de manera descendente, hasta encontrar una hoja, que corresponde a la clasificación buscada.(Quinlan, 1986, Hernández, 2004)

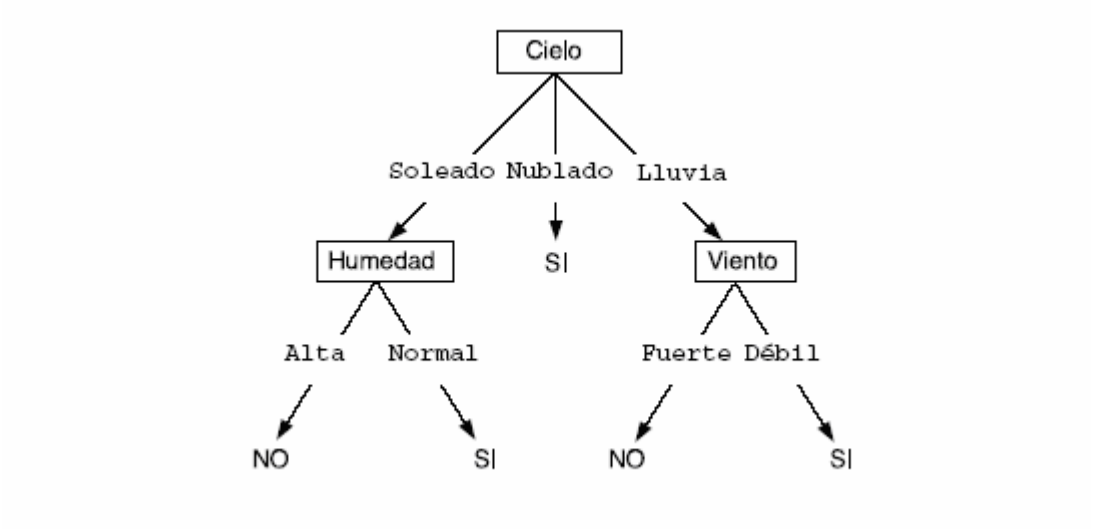


Figura 1.5 *Un árbol de decisión para el concepto" buen día para jugar tenis". Los nodos representan un atributo a ser verificado por el clasificador. Las ramas son los posibles valores para el atributo en cuestión. Los textos en mayúsculas, representan las clases consideradas y los posibles valores del atributo objetivo*

1.5.1 Árboles de decisión ID3

En esta herramienta se desarrolla la implementación de árboles de decisión ID3. El algoritmo realiza una búsqueda en escalada en este espacio que comienza con el conjunto vacío y que avanza recursivamente en la elaboración de una hipótesis que consiste en un árbol de decisión que clasifique adecuadamente los ejemplos analizados.

Para los atributos dados, el espacio de hipótesis de ID3 de todos los árboles de decisión es un espacio completo de funciones discretas finitas. Esto es así gracias a que cualquier función discreta finita puede ser representada por algún árbol de decisión.

En concreto, la búsqueda realizada sigue el principio de divide y vencerás, que en este caso se centra en la división recursiva del árbol en subárboles en los que se busca una mayor homogeneidad en las clases existentes, de tal forma que el proceso se realiza hasta que cada partición contenga ejemplos que pertenezcan a una única clase o hasta que no haya posibilidad de realizar nuevas particiones. La búsqueda es un proceso recursivo basado en una decisión, en la que se determina, en cada momento, cuál es el atributo que origina subárboles más homogéneos; entendiendo por homogeneidad la creación de grupos de ejemplos que pertenezcan a una sola clase.(Quinlan, 1986, Hernández, 2004)

1.5.1.1 Complejidad

Espacio: el espacio ocupado por ID3 es el ocupado por el árbol de decisión. En el peor de los casos, habrá un nodo hoja por cada ejemplo, con un número de nodos intermedios igual al tamaño del espacio de instancias. Esto se produciría, por ejemplo, cuando se le pasara como entrada todos los posibles ejemplos, cada uno con una clase diferente. Como intentaría encontrar una forma de separar todos los ejemplos, esto le llevaría a considerar todas las posibles combinaciones de valores de atributos (tamaño de espacio de instancias). Evidentemente, esto no es un caso real. En el mejor de los casos (por ejemplo, todos los ejemplos pertenecen a la misma clase), habría un único nodo.

Tiempo: crece linealmente con el número de ejemplos de entrenamiento y exponencialmente con el número de atributos.(Quinlan, 1986, Hernández, 2004)

1.5.1.2 Fiabilidad

Si los ejemplos no tienen ruido, ID3 encuentra el árbol de decisión que describe correctamente a todos los ejemplos. Si hay ruido, entonces depende de lo significativos que sean los ejemplos como en el resto de algoritmos inductivos.

1.5.1.3 Capacidades y limitaciones del algoritmo ID3

El espacio de hipótesis de ID3 es completo con respecto a las funciones de valores discretos que pueden definirse a partir de los atributos considerados.
De manera que no existe el riesgo que la función objetivo no se encuentre en el espacio de hipótesis.(Hernández, 2004)

ID3 mantiene sólo una hipótesis mientras explora el espacio de hipótesis posibles. Esto contrasta, por ejemplo, con el algoritmo eliminación de candidatos, que mantiene el conjunto de todas las hipótesis consistentes con el conjunto de entrenamiento.(Hernández, 2004)

El algoritmo básico ID3 no ejecuta vuelta atrás (backtracking) en su búsqueda. Una vez que el algoritmo selecciona un atributo, nunca reconsiderará esta elección. Por lo tanto, es susceptible a los mismos riesgos que los algoritmos estilo ascenso de colina, por ejemplo, caer máximos o mínimos locales. Como veremos, la vuelta atrás puede implementarse con alguna técnica de poda.(Hernández, 2004)

ID3 utiliza todos los ejemplos de entrenamiento en cada paso de su búsqueda guiada por el estadístico ganancia de información. Una ventaja de usar propiedades estadísticas de todos los ejemplos es que la búsqueda es menos sensible al ruido en los datos (Hernández, 2004).

1.6 Criterio de selección de atributos

Los criterios de selección de atributos implementados son: Chi-cuadrado y Entropía. En cada paso al seleccionar un atributo este representará un nodo.

1.6.1 Test de independencia de variables Chi-cuadrado

El procedimiento de realización del test Chi-cuadrado es el siguiente:

1) Se divide el rango de valores que puede tomar la variable aleatoria de la distribución en *K* intervalos adyacentes:

$$[a_0, a_1), [a_1, a_2), \text{K}, [a_{K-1}, a_K)$$

Pueden ser $a_0 = -\infty$ y $a_K = \infty$.

2) Sea N_j el número de valores de los datos que tenemos que pertenecen al intervalo $\lfloor a_{j-1}, a_j \rangle$.

3) Se calcula la probabilidad de que la variable aleatoria de la distribución candidata $F_X(x)$ esté en el intervalo $\lfloor a_{j-1}, a_j \rangle$. Por ejemplo, si se trata de una distribución continua, esa probabilidad sería:

$$p_j = \int_{a_{j-1}}^{a_j} f_X(x) dx$$

Siendo $f_X(x)$ la función densidad de probabilidad de la distribución candidata. También se puede hacer:

$$p_j = F_X(a_j) - F_X(a_{j-1})$$

Nótese que este es un valor teórico, que se calcula de acuerdo a la distribución candidata y a los intervalos fijados.

4) Se forma el siguiente estadístico de prueba:

$$\Delta = \sum_{j=1}^{K} \frac{(N_j - Np_j)^2}{Np_j}$$

Si el ajuste es bueno, Δ tenderá a tomar valores pequeños. Rechazaremos la hipótesis de la distribución candidata si Δ toma valores "demasiado grandes". (Liu, 1995, Castillo et al., 1996)

1.6.2 Entropía y ganancia de información

Una manera de cuantificar la bondad de un atributo en este contexto, consiste en considerar la cantidad de información que proveerá este atributo, tal y como esto es definido en teoría de información (Shannon and Weaver, 1948). Un bit de información es suficiente para determinar el valor de un atributo booleano, por ejemplo, si/no, verdadero/falso, 1/0, etc., sobre el cual no sabemos nada. En general, si los posibles valores del atributo vi, ocurren con probabilidades *P* (vi), entonces el contenido de información, o Entropía (E) de la respuesta actual está dado por:

$$E(P(v_i), \ldots, P(v_n)) = \sum_{i=1}^{n} -P(v_i) \ \log_2 \ P(v_i)$$

Si todos los ejemplos son positivos o negativos, por ejemplo, pertenecen todos a la misma clase, la Entropía sería 0. Una posible interpretación de esto, es considerar la Entropía como una medida de ruido o desorden en los ejemplos.(Liu, 1995, Castillo et al., 1996)

Se define la ganancia de información (GI) como la reducción de la Entropía causada por particionar un conjunto de entrenamiento S, con respecto a un atributo *A*:

$$Ganancia(S, A) = E(S) - \sum_{v \in A} \frac{|S_v|}{|S|} E(S_v)$$

1.7 Formato de entrada de datos

Se utilizan los ficheros tipo WEKA como formato de entrada de datos debido a su amplia utilización mundial. Al elegir este formato se puede hacer comparaciones con salidas obtenidas del conjunto de librerías java, WEKA. Este fichero de entrada contiene un conjunto de ejemplos, atributos: clasificadores y no clasificadores. Para la estructura del fichero WEKA (ver Anexo 2).(Witten and Frank, 2005)

1.8 Formato de Exportación de Resultados

Al crear la red Bayesiana se puede exportar en diferentes formatos estándar, con el objetivo de que los resultados puedan ser usados en otros softwares como datos de entrada y permitir el estudio posterior de estas redes creadas al ser estos formatos muy utilizados.

1.8.1 XML (Extensible Markup Language)

XML es el formato universal para los documentos y los datos estructurados sobre la web, se ha diseñado para la facilidad de la puesta en práctica, y para la interoperabilidad con el SGML (Standard Generalized Markup Language) y el HTML (Hypertext Markup Language), además es una manera simple por la cual representar datos. La llamada metadata (datos que además de proporcionar información se describen así mismos) es útil para que los sistemas que no saben de donde proviene determinada información sepan con que tipos de datos están trabajando.

1.8.1.1 Características del lenguaje XML

Es una arquitectura más abierta y extensible. No se necesita de versiones para que puedan funcionar en futuros navegadores. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en internet o intranet por medio de un validador de documentos, la información se provee en forma estructurada y descriptivamente visual.

1.8.2 XMLBIF (eXtensible Markup Language Interchange Format for Bayesian Networks)

El objetivo del formato XMLBIF[1] es representar los GAD que se pueden asociar a las medidas condicionales de la probabilidad para las variables discretas, con la posibilidad de que las variables de decisión y de utilidad estén presentes en el gráfico, o sea da la posibilidad de representar una red de creencia o red Bayesiana.

[1] http://www-2.cs.cmu.edu/~fgcozman/Research/InterchangeFormat/

El formato acentúa la internacionalización adoptando las pautas de XML para la especificación del documento. XMLBIF es muy similar al HTML y la mayoría de las estructuras usadas en el HTML aparecen en XMLBIF.

1.9 Consideraciones finales del capítulo

En este capítulo han sido mostrados los conceptos y algoritmos que se necesitan dominar para crear una herramienta capaz de crear redes Bayasianas y hacer inferencias Bayesianas, se observan limitaciones, ventajas, de algunos de estos algoritmos.

Este sistema consta de varios módulos, a continuación se describen las utilidades de estos módulos: cargar datos del fichero, construcción de los árboles de decisión, construcción de tablas de probabilidades, construcción de la red Bayesiana, exportar en ficheros la red creada, propagación de evidencia y mostrar los resultados de la inferencia Bayesiana.

CAPÍTULO II. ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PARA HACER INFERENCIAS BAYESIANAS

En este capítulo se desarrolla el análisis y diseño de la herramienta que se propone para lograr el análisis de los datos obtenidos del fichero tipo WEKA, una vez analizados estos datos se construye la red Bayesiana, dejando la red lista para hacer inferencias Bayesianas, dando la posibilidad de exportar los resultados en varios formatos de archivos.(Witten and Frank, 2005)

2.1 Herramientas utilizadas

Se realizó una aplicación sobre Windows para el desarrollo del sistema, utilizando las facilidades que brinda el ambiente de desarrollo Borland Delphi que es una combinación potente de herramientas visuales de disposición, características de desarrollo de aplicaciones y soporte para la edición de código.

Se pretende usar este ambiente de desarrollo debido a que se puede rehusar el código que ya está programado, o sea, el uso de clases de Delphi, para algunas medidas que se deben implementar. Para su posterior publicación en Internet se exportan los resultados en ficheros con formatos (XML, XMLBIF).

- Se utiliza Programación Orientada a Objetos

2.1.1 Programación Orientada a Objetos

La Programación Orientada a Objetos (POO) es una técnica de programación, por cuanto sus conceptos se pueden implementar en los lenguajes de programación tradicionales, para algunos el término está asociado solamente a los lenguajes o herramientas que soportan el

paradigma. Lo cierto es que ha contribuido a que el desarrollo del software evolucione de un enfoque procedural a un enfoque basado en objetos.

En el paradigma de la POO los agentes actuantes son entidades independientes cada uno con su propia estructura interna, que se comunican mutuamente respondiendo o haciendo demandas. Estas entidades, llamadas objetos, están constituidas por propiedades (atributos, datos que definen su estado) y métodos (operaciones aplicadas sobre los datos).

La estructura interna de un objeto no debe ser accedida por otro objeto o programa. Este concepto se conoce como encapsulamiento y es un concepto clave en la POO, que separa la implementación del objeto de su uso.

Los objetos se comunican con otros objetos a través de los mensajes. Los mensajes pueden tener como objetivo modificar o preguntar sobre el valor de un atributo de un objeto, ejecutar acciones específicas, etc. Para cada mensaje existe un método, el cual es un segmento de código que manipula el mensaje.

2.2 Alcance de la investigación

Las redes Bayesianas permiten mejorar las posibilidades de predicción de propiedades funcionales o estructurales de secuencias de proteínas a partir de razonamientos probabilísticos, especialmente ante la presencia de información incompleta o con cierta incertidumbre. Conociendo posiciones en secuencias de proteínas es posible inferir otras posiciones, y una familia dada.

2.3 Modelación del sistema

Con la investigación de la utilidad del sistema para los usuarios se pudo diseñar la herramienta, sus utilidades se reflejan en los casos de uso.

En la modelación de este sistema se utilizó la notación del UML (Unified Modeling Language), que es un lenguaje visual estándar que se utiliza para especificar, visualizar, construir y documentar los diferentes aspectos relativos al desarrollo de un software

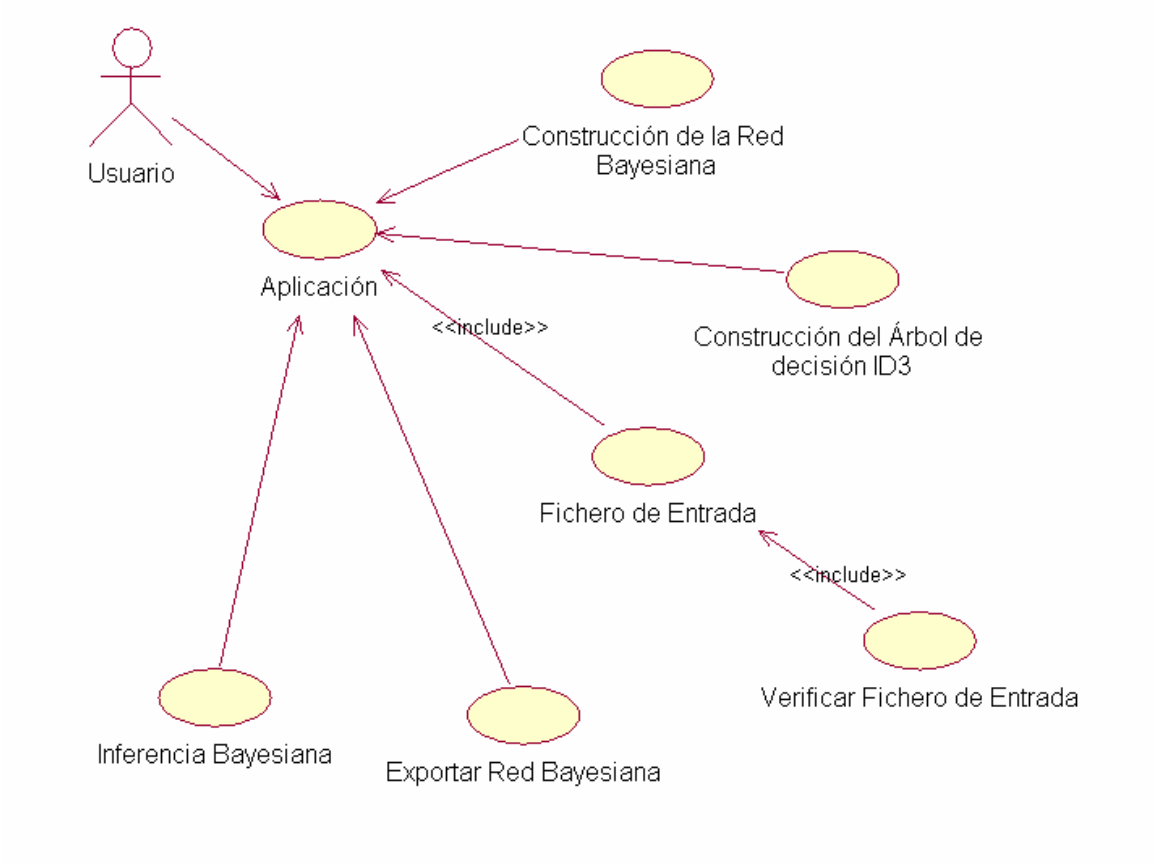


Figura 2.1 *Diagrama de Casos de Uso del sistema*

Una técnica del UML, que permite mejorar la comprensión de los requerimientos del sistema, es la identificación de casos de uso y actores. Los casos de usos son los procesos

que debe llevar a cabo la aplicación en los que toma parte cada uno de los actores (agentes externos). Normalmente un actor estimula al sistema con eventos de entradas o recibe algo de él.

En la herramienta que se presenta, se identifican siete casos de uso y un actor, que es el usuario. En la figura 2.1 se presenta el diagrama de casos de uso del actor usuario, en la figura 2.2 se representan los casos de uso para exportar los resultados.

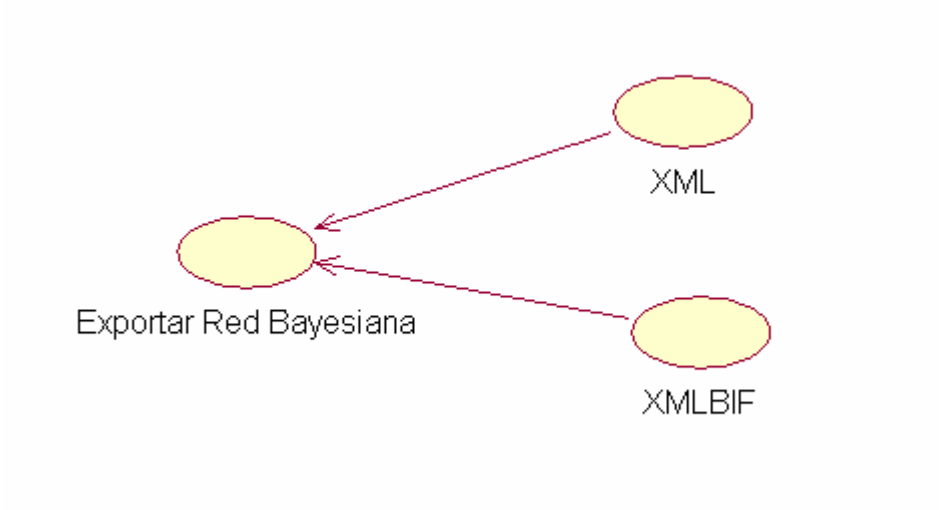


Figura 2.2 *Formatos de exportación de resultados*

A continuación se describe los casos de uso para el autor usuario:

- Aplicación: Caso de uso en el cual se comienza a ejecutar el software implementado.

- Fichero de Entrada: Caso de uso en el cual procedemos a cargar el fichero con los datos a utilizar en la aplicación.

- Verificar Fichero de Entrada: Caso de uso donde se comprueba la estructura del fichero de entrada y luego de esta comprobación se comienzan a utilizar los datos en la aplicación.

- Construcción del Árbol de Decisión (ID3): Caso de uso en el cual se construye el árbol de decisión.

- Construcción de la red Bayesiana: Caso de uso en el cual se construye la red Bayesiana.

- Exportación de resultados: Caso de uso en el cual se exporta la red Bayesiana en varios formatos (XMLBIF, XML).

- Inferencia Bayesiana: Caso de uso donde se hace la inferencia Bayesiana.

2.4 Implementación del sistema

La herramienta de trabajo diseñada prepara la información, de forma tal que al leer de un fichero tipo WEKA, los datos contenidos en este fichero se almacenan en la clase TStatisticDataSet, esta clase contiene tres tipos de atributos FRelation es una cadena que guarda el nombre del fichero; FVariablesDescriptors de tipo TList (lista de objetos), guarda los nombres de los atributos y la clase tomadas del fichero de entrada y sus posibles valores a tomar; FData es una matriz que contiene los casos dados(arreglo de arreglo de enteros). A continuación se muestran otras clases y funciones que son implementadas en el software.

- TStatisticDataSet:

  Esta clase es implementada para que incluya toda la información de los datos del fichero, en ellas se implementan varios procedimientos y funciones que facilitan el trabajo con la información

- Función Chi-Cuadrado:

  Function Chiprob (x: real; n: integer): real;

  Esta función recibe como parámetros un valor de Chi-cuadrado y los grados de libertad y se obtiene como resultado una probabilidad que servirá para definir la dependencia de un atributo de la clase; si la probabilidad es menor a 0.05 entonces diremos que es dependiente.

- Función Entropía:

  Function Entropia: real;

  Esta función devuelve el valor de Entropía de todas las variables, partiendo del conjunto de ejemplos iniciales, podemos decir que a diferencia del algoritmo para calcular el valor de Chi-cuadrado el resultado no es una probabilidad, o sea, es un valor real positivo.

- TDesicionTreeCompiler:

  Esta clase es la encargada de la construcción del árbol de decisión ID3, contiene un atributo FStatisticDataSet que guarda los datos necesarios para la construcción de este árbol, además crea la red Bayesiana partiendo de los árboles de decisión construidos.

  Procedure Calculate_Dependency (const ActiveRecords, ActiveVars: TArrayMarcador; var Dependency: TArrayReal);

  Este procedimiento devuelve una lista de las dependencias de las variables activas con la clase. Entre los parámetros pasados se encuentra ActiveRecords, este arreglo tendrá información de cuáles son los casos (ejemplos) a tener en cuenta al aplicar el

cálculo da la dependencia, ActiveVars será una arreglo que contendrá la información del número de variables a calcular dependencia, o sea, tendrá la información de cuáles variables tomar para el cálculo de la dependencia.

Function Create_Child (const Level: integer; const Father: integer; ActiveRecords, ActiveVars: TArrayMarcador): integer;

Este procedimiento recursivo construye el árbol de decisión, recibe como parámetros Level que define el número de niveles del árbol de decisión; Father, que decide cuál será el nodo padre del árbol; ActiveRecords, lleva el control de que casos tomar en la construcción del árbol, esta función devuelve el índice de la variable en el arreglo por la cual se ramificó en el árbol.

FgraphMat: este atributo es una matriz que contiene los enlaces entre los diferentes nodos de los árboles creados, dicha estructura se va llenando en el procedimiento Create_Child.

- TCompilingDesicionTreeDlg:

  Esta clase es la encargada de hacer la llamada del procedimiento que permite la construcción del árbol de decisión y posee procedimientos encargados de pausar, comenzar o cancelar dicha construcción.

- TCompileDlg:

  Esta es la clase encargada de la configuración que queremos que tome nuestro árbol de decisión, toma los datos de una forma visual: las cantidad de niveles que tendrá el árbol de decisión; permite escoger la clase dentro de todos los atributos que contiene el fichero de entrada, la cantidad mínima de casos (ejemplos) que contendrá un corte, el criterio de selección de atributos que queremos aplicar en la construcción del árbol de decisión.

- TMainForm:

  Esta es la clase visual principal del software carga un fichero existente tipo WEKA, nos da la posibilidad de editar un fichero nuevo de datos, así como nos brinda la opción de salvarlo. Desde esta clase se comienza el proceso de la construcción del árbol de decisión. Esta clase presenta un menú con todas las utilidades del software.

- TBIF:

  Esta clase es la encargada de la construcción de tabla de probabilidades, al ejecutar los procedimientos de esta clase se calculan todas las probabilidades de los nodos, se construye la tabla de probabilidades y se guardan las combinaciones de los valores que toman los nodos en el cálculo de estas probabilidades. Esta clase tiene también como funcionalidad calcular la propagación.

- TPropagationTree:

  Esta clase almacena los procedimientos encargados de hacer la inferencia Bayesiana como son: InicializateProbability, CreateUnionTree, AddEvidence, DeleteEvidences, NumerationPerfect, CreateClique, CerateFamilyTree que son necesarios para ejecutar el algoritmo de propagación.

  A continuación en las figuras 2.3 y 2.4 se muestran los diagramas que relacionan las clases en el sistema implementado.

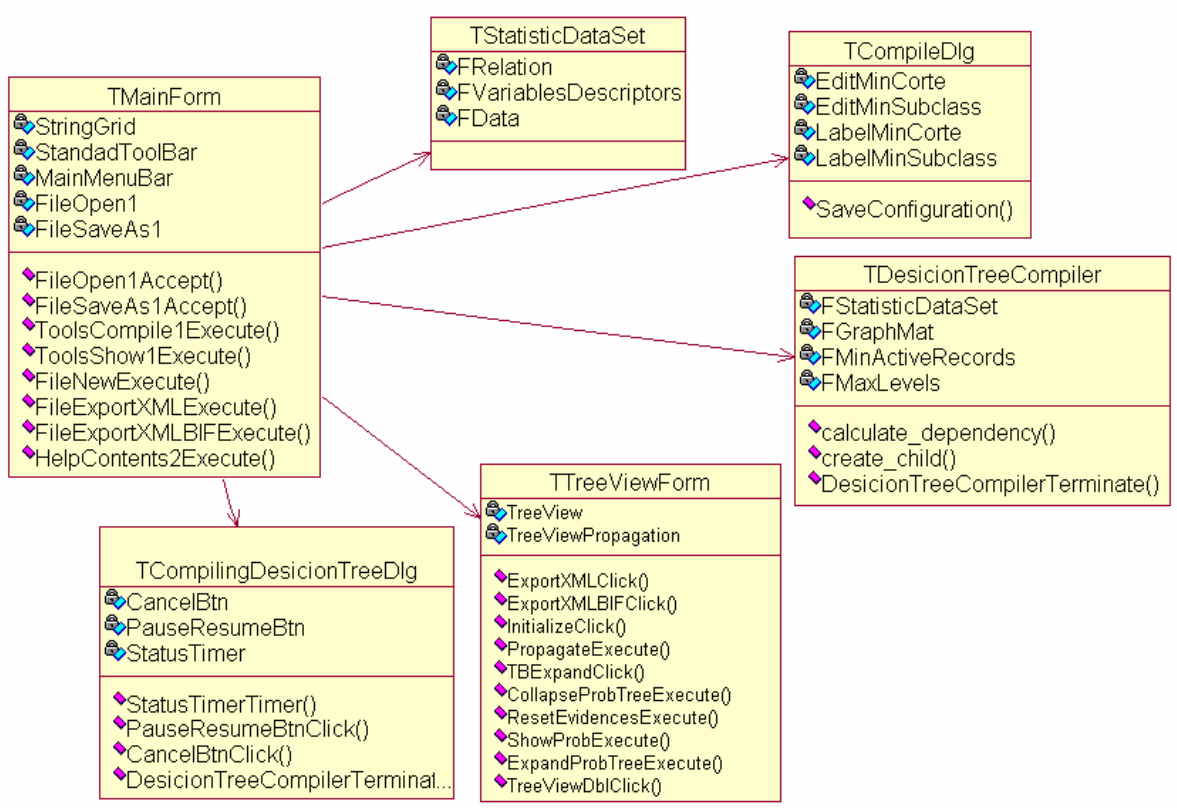


Figura 2.3 *Diagrama principal de relación de clases del sistema*

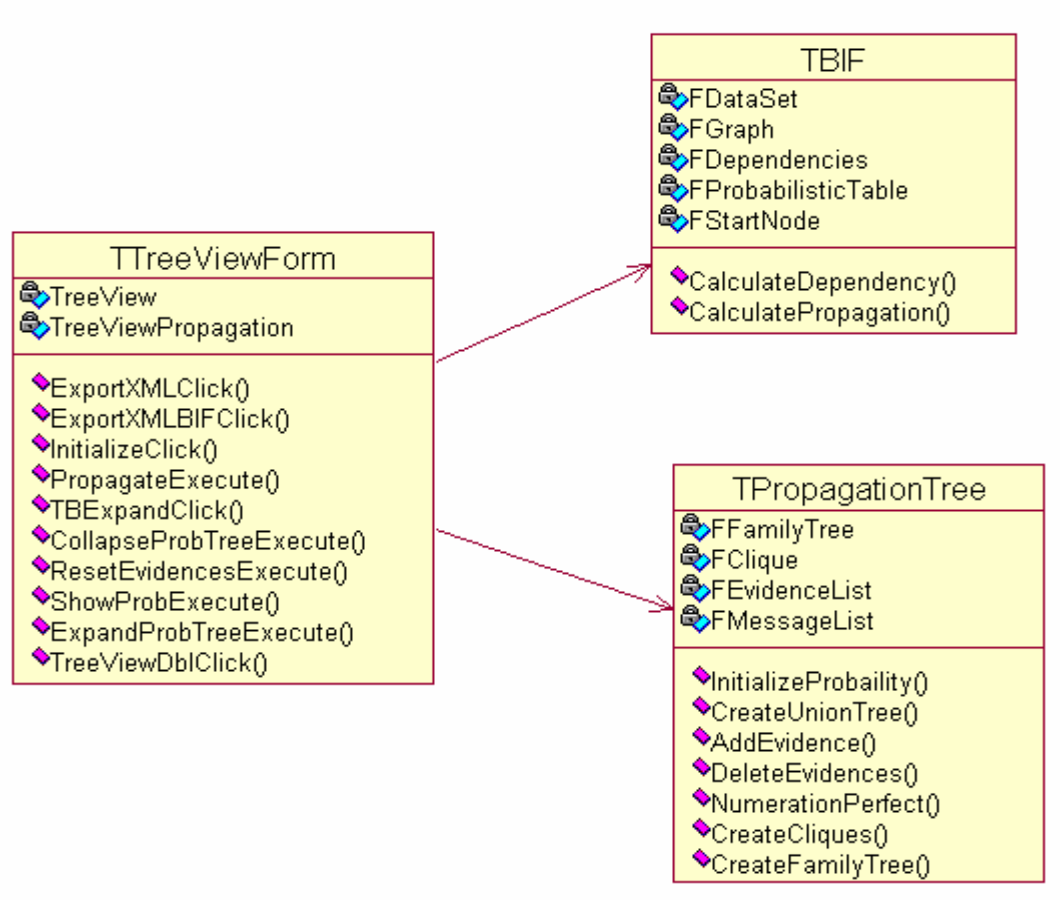


Figura 2.4 *Diagrama de relación de clases*

Con la declaración de clases y la relación entre ellas queda implementada la herramienta computacional para hacer inferencias Bayesianas.

2.5 Diagrama de transición de estados

En el diagrama que se muestra a continuación se puede observar el funcionamiento del software, la relación existente entre las diferentes ventanas que se utilizan en el sistema y la secuencia de acciones a seguir cuando se trabaja con la herramienta implementada, ver (figura 2.5).

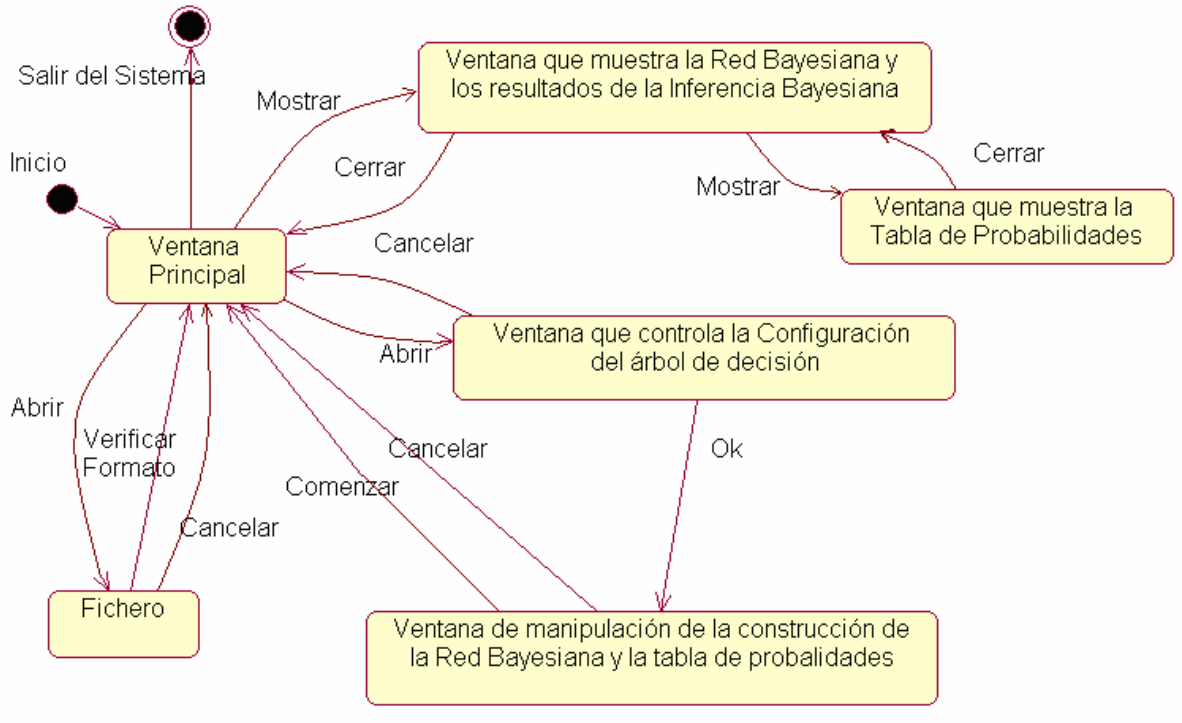


Figura 2.5 *Diagrama de transición de estados*

Se puede observar en la figura 2.5 como al ejecutarse la herramienta se muestra la Ventana Principal del Sistema (VPS), que manipula varias instancias. Cuando se carga un fichero, pueden ocurrir dos acciones: que se cancele la opción de abrir o que se verifique el formato del archivo.

Cuando se ejecuta la ventana que controla la construcción del árbol de decisión pueden existir dos variantes: cancelar que retorna a la VPS o aceptar que muestra la instancia que manipula la construcción de la red Bayesiana y la tabla de probabilidades.

La ventana que manipula la construcción de la red Bayesiana y la tabla de probabilidades cede la posibilidad al usuario de comenzar, pausar y cancelar esa construcción.

Desde la forma visual que muestra la red y los resultados de la inferencia Bayesiana se muestran las probabilidades de los nodos en una nueva ventana, al cerrarse esta forma visual se regresa a la VPS.

2.6 Análisis de formato de ficheros para exportar resultados

Los formatos de exportación de resultados que se utilizan son: XML y XMLBIF. A continuación se describen ejemplos de ambos tipos de archivos.

2.6.1 Fichero XMLBIF

Este formato posee la gran ventaja de agrupar toda la información del grafo de la red Bayesiana: el nombre de las variables que forman los nodos, la tabla de probabilidades, los enlaces a los nodos hijos y los valores posibles que toman las variables. Su uso específico es para almacenar grafos en su estructura. (Ver Anexo1)

En este archivo se define la estructura de un nodo que se llena con toda la información que el posee, por ejemplo:

```
<VARIABLE TYPE="nature">
        <NAME>hear-bark</NAME>
        <OUTCOME>true</OUTCOME>
        <OUTCOME>false</OUTCOME>
        <PROPERTY>position = (154, 241) </PROPERTY>
</VARIABLE>
```

La variable *hear-bark* de tipo *nature* puede tomar dos valores: *true*, *false*, también guarda una posición determinada.

2.6.2 Fichero XML

El formato XML es muy utilizado en la actualidad, aunque es diseñado para guardar cualquier tipo de información, nuestra herramienta exporta la información de la red Bayesiana en un XML.

A continuación se muestra un pequeño fichero XML:

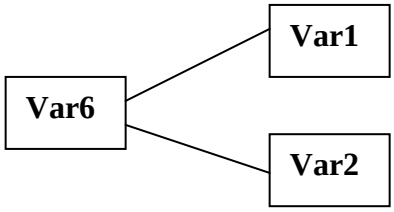
```
<? Xml version="1.0" encoding="UTF-8" standalone="no" ?>

<! Network">
<Network>
 <Node>
    <Name> Var6 </name>           <! Nodo inicial>
    <Node>
          <Name> Var1 </name>
          <Node>
               <Name> Var3 </name>
          </Node>
    </Node>
    <Node>
          <Name> Var2 </name>
          <Node>
               <Name> Var4 </name>
               <Node>
                     <Name> Var5 </name>
               </Node>
          </Node>
    </Node>
 </Node>
</Network>
```

Este formato es similar al XMLBIF, la única diferencia consiste en la forma de agrupar la información del grafo de la red Bayesiana, en este ejemplo solo se guarda información del nombre de los nodos y la lista de hijos que presenta cada nodo, por ejemplo:

```
<Node>
        <Name> Var1 </name>
          <Node>
               <Name> Var3 </name>
          </Node>
</Node>
```

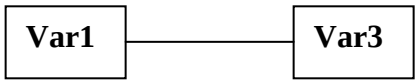
Analizando el fichero que se toma por ejemplo tenemos que la red quedaría con la siguiente estructura:

1) El nodo que tiene como nombre Var6 es el nodo inicial del grafo, tiene enlaces a los nodos que tienen como nombre Var1 y Var2, la siguiente representación.



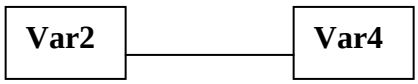
2) El nodo que tiene como nombre Var1 tiene enlace con el nodo que se nombra Var3.

```
<Node>
        <Name> Var1 </name>
           <Node>
                <Name> Var3 </name>
           </Node>
</Node>
```



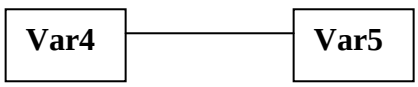
3) El nodo que tiene como nombre Var2 tiene como hijo al nodo Var4.

```
<Node>
         <Name> Var2 </name>
           <Node>
                <Name> Var4 </name>
                <Node>
                      <Name> Var5 </name>
                </Node>
           </Node>
</Node>
```



4) El nodo que tiene como nombre Var4 tiene como hijo al nodo Var5.

```
<Node>
        <Name> Var4 </name>
           <Node>
                <Name> Var5 </name>
           </Node>
 </Node>
```



El grafo resultante sería el que se muestra continuación (ver figura 2.6).

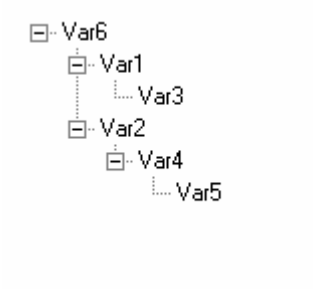


Figura 2.6 *Grafo de la red Bayesiana que guarda el fichero XML*

Este ejemplo no lleva incluida la tabla de probabilidades y el grafo que tiene representado posee solo seis nodos.

2.7 Complejidad de algunos algoritmos implementados

El algoritmo de construcción de las tablas de probabilidades de la red Bayesiana es un problema NP-Completo, porque en el cálculo de las probabilidades se analizan todas las combinaciones de los atributos dependientes de cada nodo. A lo sumo cada nodo tendrá N^m combinaciones, donde m es el número de atributos dependientes y N son los valores posibles que estos pueden tomar, en caso de que todos no tengan la misma cantidad de valores posibles las combinaciones serían mayor igual a N^m, siendo N el menor número de valores posibles de todos los atributos de la base de casos. En la implementación de este algoritmo se utilizan arreglos dinámicos con el objetivo de hacer un uso eficiente de memoria; otra solución que se le da a este problema es a la hora de seleccionar casos, desechando combinaciones que no se encuentran en la base de casos. Es bueno señalar que un uso ineficiente de memoria podría interferir en la eficiencia de este algoritmo.

El algoritmo de propagación de la red Bayesiana es un problema NP-Completo, el procedimiento de envío de mensajes puede generar un número de ciclos que a lo sumo será igual a la sumatoria del producto de la cantidad de nodos por sus vecinos. Esto significa que no existe ningún algoritmo que resuelva este problema en tiempo lineal para redes Bayesianas con cualquier topología.

2.8 Consideraciones finales del capítulo

La herramienta computacional para hacer inferencias Bayesianas fue diseñada para que pueda ser utilizada en futuras investigaciones dentro del área de la Bioinformática con el objetivo de observar el comportamiento de proteínas.

La implementación está dirigida a brindar al usuario toda la información necesaria en el proceso de construcción y de inferencia de la red Bayesiana.

La exportación de resultados en los formatos explicados con anterioridad, permite que los resultados del software puedan utilizarse en otras aplicaciones, al ser estos formatos muy utilizados.

CAPÍTULO III. MANUAL DE USUARIO

Este capítulo está dedicado a describir el manual de usuario del sistema, que sirve de guía para proceder en la interfaz visual y hacer las acciones programadas. Las principales funcionalidades que brinda el sistema son: la construcción de redes Bayesianas y la realización de inferencias Bayesianas.

3.1 Requisitos para la explotación del sistema

Para un correcto uso del sistema es necesario cargar un fichero de datos tipo (WEKA), estos datos serán guardados en estructuras del sistema, todas las acciones de la herramienta dependerán de estos datos.

3.1.2. Requerimientos de hardware

Para que el sistema implementado tenga un buen desempeño se requiere el siguiente hardware:

Requerimiento	Hardware	Memoria RAM (Mb)	Mouse
Mínimo	Celeron	64	Sí
Recomendado	Pentium	1024	Sí

3.2 Acceso a la herramienta con el objetivo de hacer inferencias Bayesianas

Esta herramienta es diseñada y programada para hacer inferencias Bayesianas, las utilidades del software se describen a continuación para que el usuario tenga una visión de cómo utilizarlo.

El sistema cuenta con una barra de herramientas y varios menús encargados del buen funcionamiento del software. Al ejecutar la herramienta implementada aparecen varias opciones deshabilitadas, pues estas utilidades dependen de datos iniciales y no tiene sentido ejecutarlas por separado.

Como muestra la figura 3.1 la barra de herramientas inicialmente presenta dos botones habilitados, estos botones son para crear un fichero nuevo de datos y para abrir un fichero ya existente.

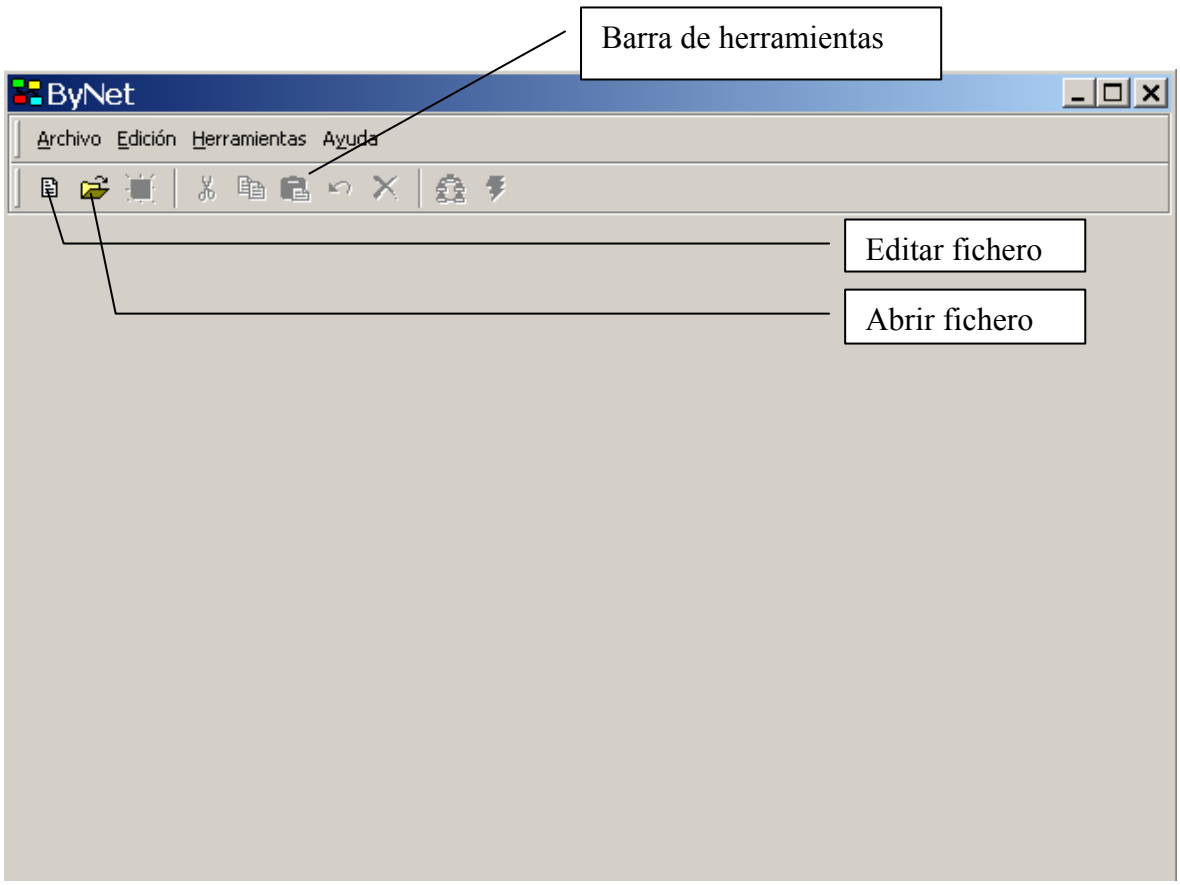


Figura 3.1 *Ventana principal del sistema*

3.2.1 Ficheros de entrada y edición de datos

Al editar un fichero nuevo se muestra la ventana de la figura 3.2, al trabajar con los datos se habilitan en la barra de herramienta los botones que tienen funciones de edición: copiar, pegar, cortar, atrás, borrar. El estado de la barra de herramienta lo podemos apreciar en la figura 3.3

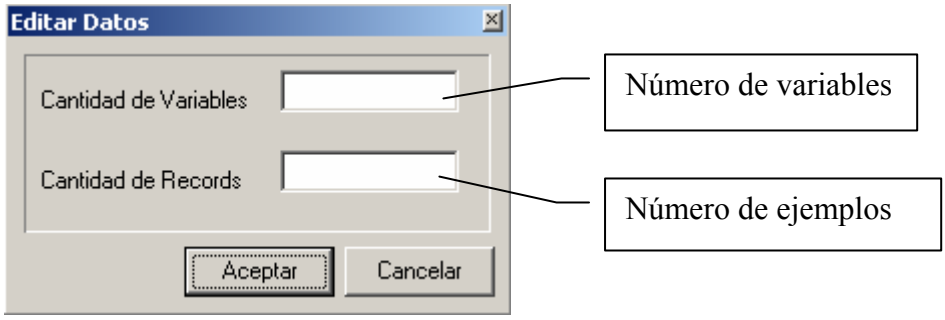


Figura 3.2 *Ventana de configuración de un fichero nuevo*

En la figura 3.2 se hace la configuración del fichero que se quiere crear, especificando el número de variables que tendrá el archivo y la cantidad de ejemplos que presenta.

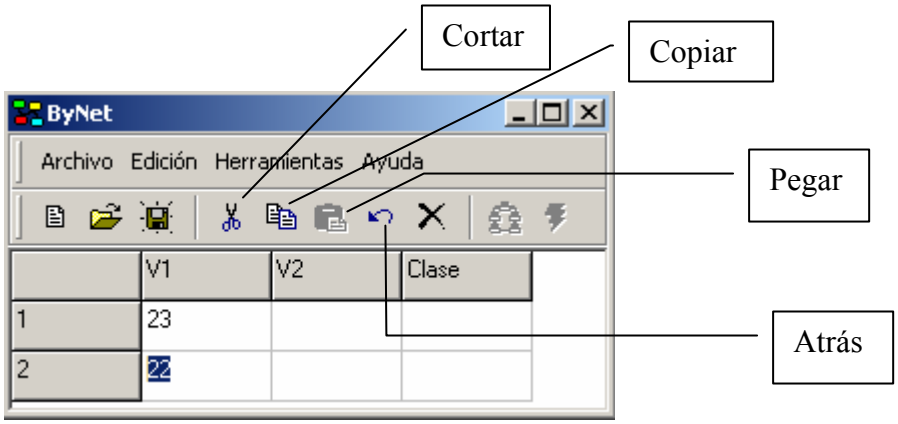
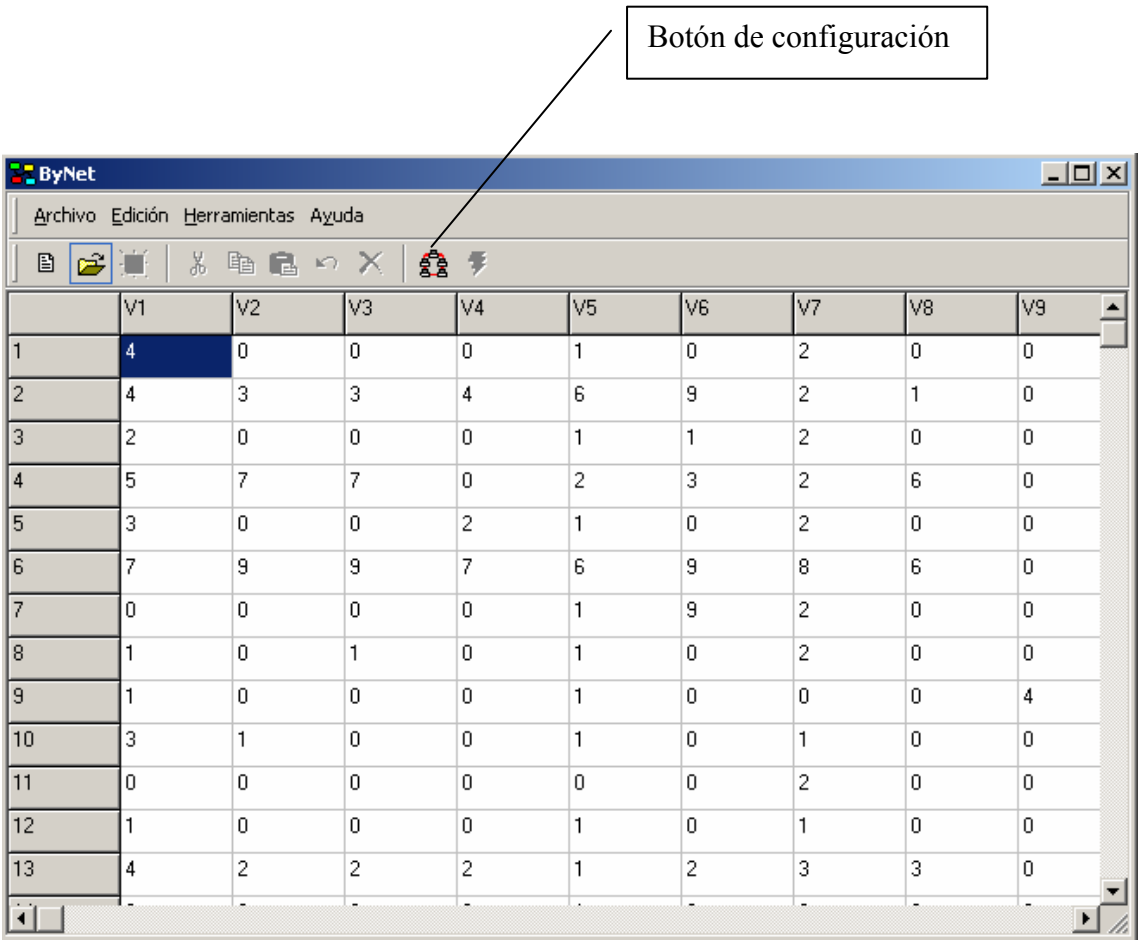


Figura 3.3 *Ventana de edición de datos*

Al abrir un fichero existente (WEKA) se habilita en la barra de herramientas el botón que muestra en una ventana nueva la configuración que tomará el árbol de decisión (ver figura 3.4), el editor muestra los casos de forma indexada, o sea, muestra índices de los valores que las variables toman en el fichero.



	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	4	0	0	0	1	0	2	0	0
2	4	3	3	4	6	9	2	1	0
3	2	0	0	0	1	1	2	0	0
4	5	7	7	0	2	3	2	6	0
5	3	0	0	2	1	0	2	0	0
6	7	9	9	7	6	9	8	6	0
7	0	0	0	0	1	9	2	0	0
8	1	0	1	0	1	0	2	0	0
9	1	0	0	0	1	0	0	0	4
10	3	1	0	0	1	0	1	0	0
11	0	0	0	0	0	0	2	0	0
12	1	0	0	0	1	0	1	0	0
13	4	2	2	2	1	2	3	3	0

Figura 3.4 *Ventana que muestra los datos de un fichero existente*

3.2.2 Construcción de la red Bayesiana

En la configuración del árbol de decisión se toman los parámetros siguientes: criterio de selección de atributos *método (Entropía y Chi-cuadrado); clase*; *máximo número de niveles que tendrá el árbol a construir; cantidad de records; significación en caso de ser Chi-cuadrado* (ver figura 3.5).

Al configurarse el árbol de decisión, se comienza el proceso de construcción de este árbol, mostrándose en una ventana las opciones: *Comenzar*, *Cancela* (ver figura 3.6).

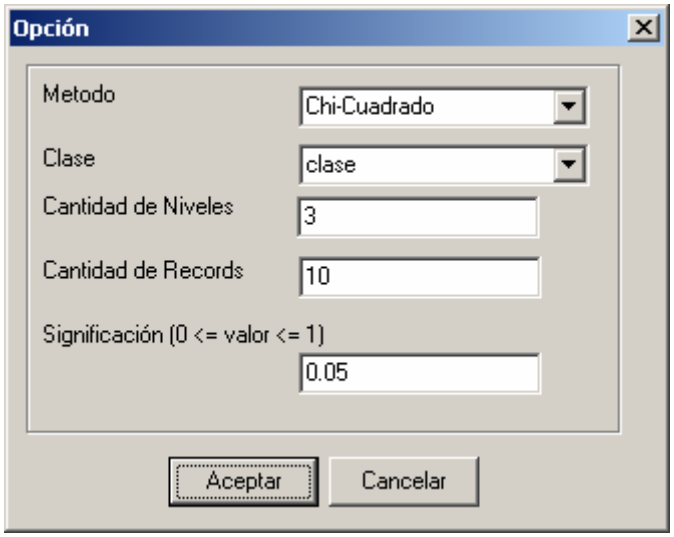


Figura 3.5 *Ventana de configuración del árbol de decisión*

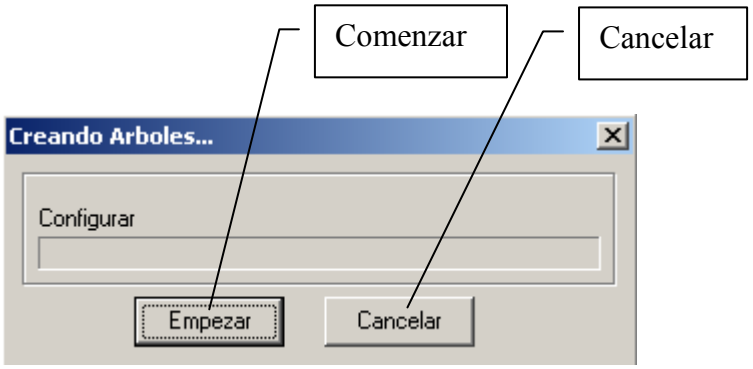


Figura 3.6 *Ventana de manipulación de la construcción del árbol de decisión*

Cuando se crea el árbol de decisión se construye la red Bayesiana, además se comienza a construir la tabla de probabilidades, mostrándose en una barra de progreso el por ciento de cálculos realizados; cuando se termina la construcción de la tabla, se habilita en la barra de herramientas un botón que muestra la red Bayesiana (ver figura 3.7).

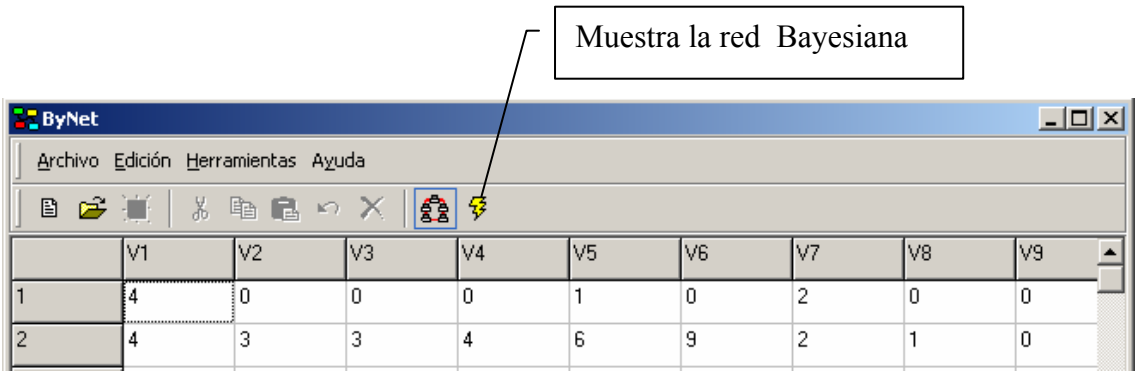


Figura 3.7 *Barra de herramientas con botón habilitado para mostrar red Bayesiana*

En la figura 3.8 se puede percibir como se visualiza la red Bayesiana, esta ventana brinda la opción de salvar la red creada a otro formato (XML, XMLBIF), lo podemos hacer dando clic derecho en la ventana y nos aparece un submenú que brinda estas opciones de exportar, se puede apreciar también en la figura 3.8, al seleccionar uno de estos formatos se especifica el fichero donde se guardará la red.

El signo positivo (+) indica que ese nodo presenta hijos o sea se ramifica, el signo negativo (-) indica que los hijos de ese nodo ya se encuentran visualizados.

Al dar doble clic sobre un nodo de esta red se muestra su respectiva tabla de probabilidad (ver figura 3.9).

En la barra de herramientas de la ventana que aparece en la figura 3.8 aparece habilitado un botón que cambia la configuración de la ventana y habilita los restantes botones de esta barra; estos nuevos botones tendrán la función de inicializar con una probabilidad cada nodo de la red, hacer inferencia Bayesiana dada una evidencia que se toma dando doble clic encima de un nodo de la red, expandir los nodos con sus probabilidades, resumir los nodos. El resultado de esta inferencia se muestra con un valor de probabilidad en cada nodo (ver figura 3.10).

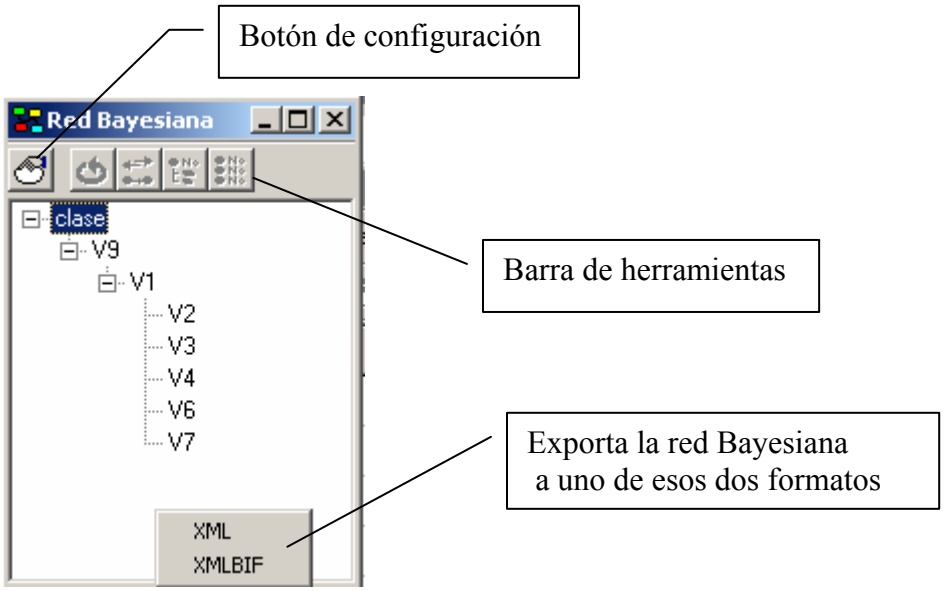


Figura 3.8 *Ventana que muestra la red Bayesiana*

Tabla de Probabilidad

	V5	V6	V4	P(V5/...)
1	0	0	0	0.48
2	1	0	0	0.52
3	0	1	0	0.65
4	1	1	0	0.35
5	0	0	1	0.47
6	1	0	1	0.53
7	0	1	1	0.48
8	1	1	1	0.52

Figura 3.9 *Tabla de probabilidades para el nodo V24*

Al ejecutar el botón de configuración de la ventana que se muestra el la figura 3.8, se divide en dos partes la ventana, una parte seguirá mostrando la red Bayesiana y la otra visualizará los resultados de la inferencia Bayesiana (ver figura 3.10).

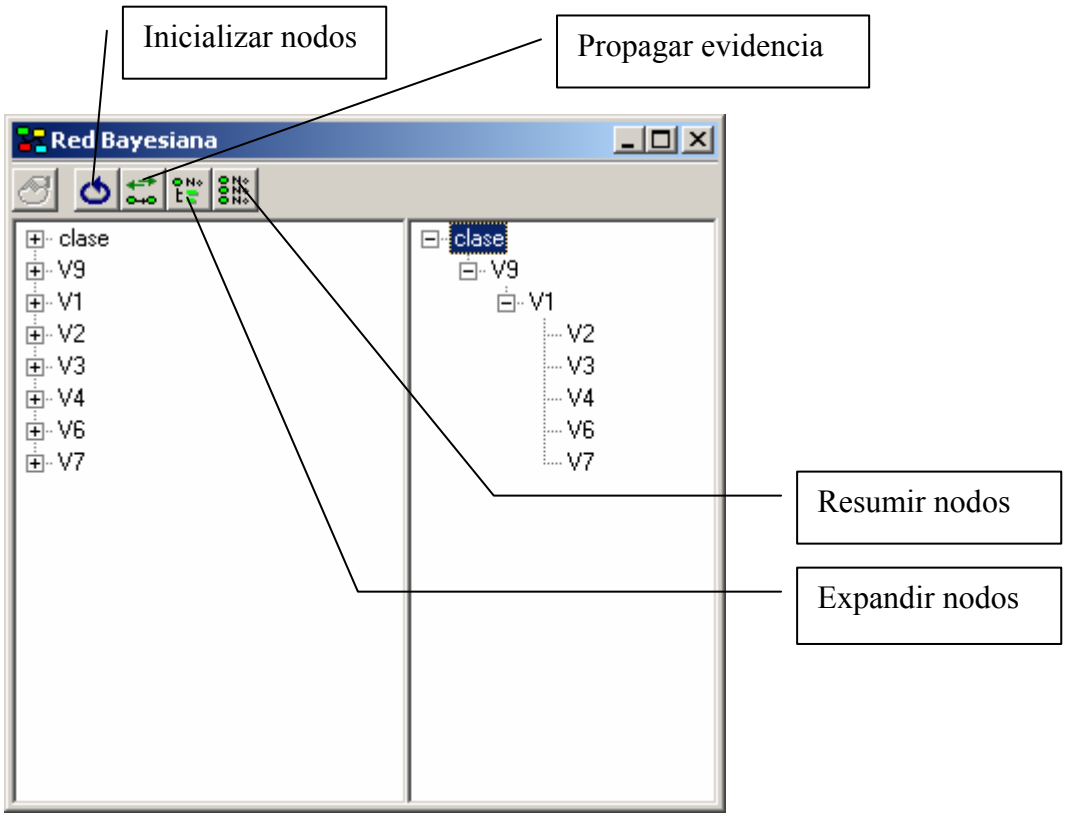


Figura 3.10 Ventana que muestra la propagación de evidencia

3.3 Manual para el uso del menú de barras del sistema

En la ventana principal de la herramienta computacional para hacer inferencias Bayesianas aparece un menú de barras que se descuelgan y brindan diferentes opciones, (ver figura 3.11).

Figura 3.11 *Menú de barras con las opciones* ***Archivo***, ***Edición***, ***Herramientas***, ***Ayuda***

La primera barra del menú es **Archivo**, que se despliega en: **Nuevo**, **Abrir…**, **Guardar…**, **Exportar…**, **Salir** (ver figura 3.12).

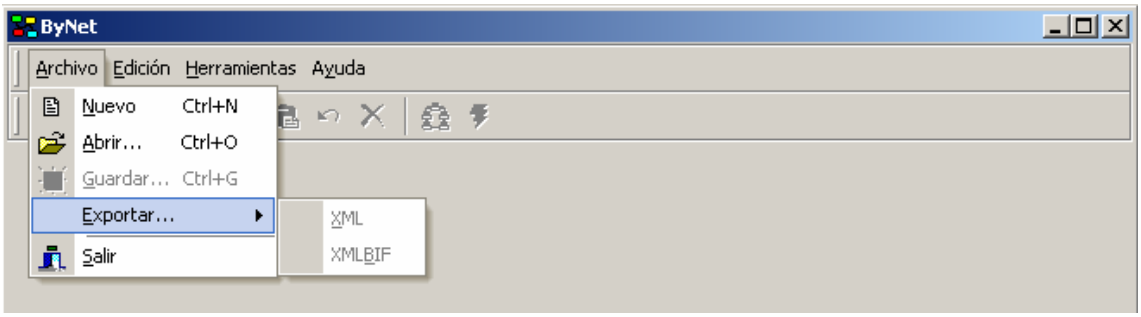


Figura 3.12 *Menú* ***Archivo*** *desplegado*

Estas utilidades son implementadas para el trabajo con ficheros, a continuación se describen sus funciones.

Nuevo: Permite editar un fichero tipo WEKA nuevo, al tomar esta opción nos aparece una ventana donde se especifica el número de variables y el número de casos que tendrá el fichero (ver figura 3.2).

Abrir…: Permite abrir un fichero existente tipo WEKA, visualizándolo en la ventana principal de la herramienta (ver figura 3.4).

Guardar…: Permite salvar un fichero que es editado en esta herramienta, especificándole el camino para guardar el fichero, así como su nombre.

Exportar…: Permite exportar la red creada a los formatos de archivos XML, XMLBIF. Estas opciones aparecen como submenú.

Salir: Salir del sistema.

La segunda barra del menú es **Editar**, que se despliega en: **Cortar**, **Copiar**, **Pegar**, **Seleccionar Todo**, **Deshacer**, **Borrar** (ver figura 3.13).

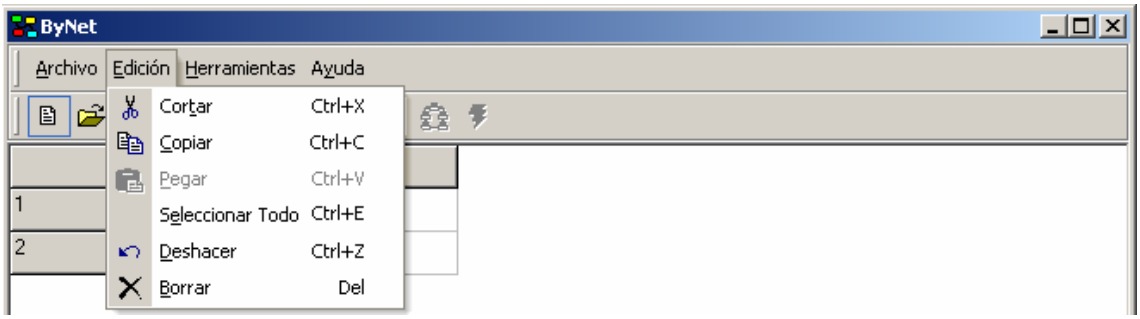


Figura 3.13 *Menú* ***Editar*** *desplegado*

Estas utilidades son implementadas para la edición de ficheros, a continuación se describen sus funciones.

Cortar: Permite cortar un texto seleccionado.

Copiar: Permite copiar un texto seleccionado.

Pegar: Permite pegar un texto copiado.

Seleccionar Todo: Permite seleccionar todo el texto de la celda donde se encuentra el cursor.

Deshacer: Permite ir atrás, o sea deshabilitar una acción reciente, similar a la opción existente en el editor de textos Microsoft Word.

Borrar: Permite borrar un texto seleccionado.

La tercera barra del menú es **Herramientas**, que se despliega en: **Compilar Grafo**, **Ejecutar,** esta opción aparece deshabilitada por defecto hasta que no se ejecute **Compilar Grafo** (ver figura 3.14).

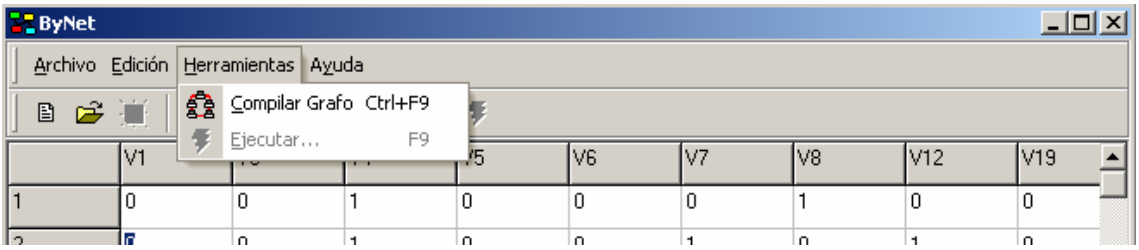


Figura 3.14 *Menú* ***Herramientas*** *desplegado*

Estas utilidades son implementadas para la construcción de la red Bayesiana y para hacer la inferencia Bayesiana, a continuación se amplían esas funciones.

Compilar Grafo: Permite configurar la estructura del árbol de decisión (ver figura 3.6). Seguidamente se construye la red Bayesiana y se calcula la tabla de probabilidades.

Ejecutar: Muestra la red Bayesiana en una ventana, y desde esta misma ventana podemos hacer la inferencia Bayesiana (ver figura 3.10).

La cuarta barra del menú es **Ayuda**, que se despliega en: **Contenido, Acerca de…** **(**ver figura 3.15).

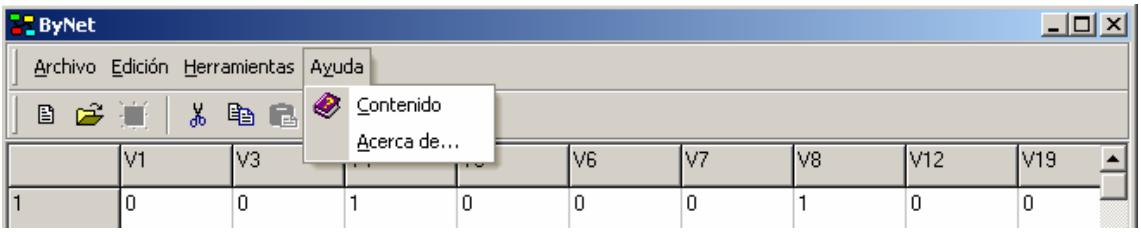


Figura 3.15 *Menú* ***Ayuda*** *desplegado*

Contenido: Esta opción muestra una ventana con información relacionada con el sistema implementado (ver figura 3.15).

Acerca de…: Esta opción muestra al usuario un pequeño manual para el correcto uso del sistema (ver figura 3.16).

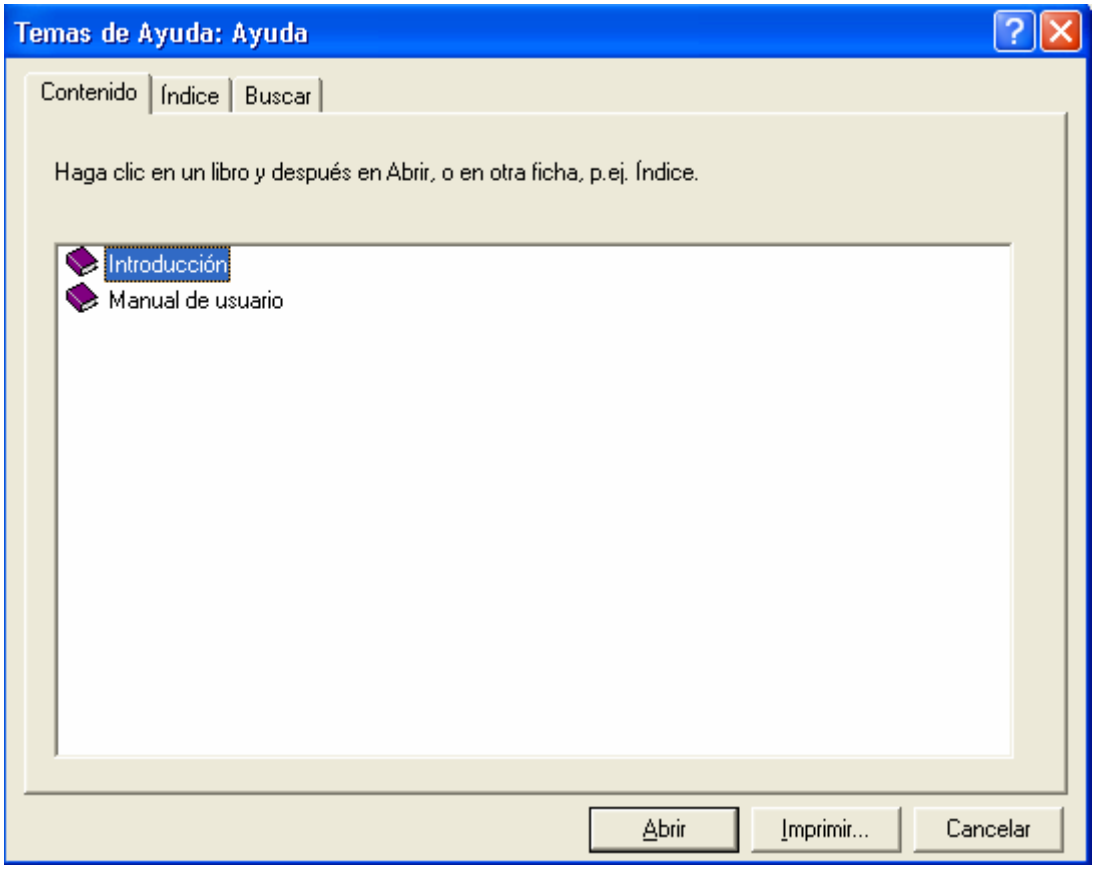


Figura 3.15 *Ventana de ayuda*

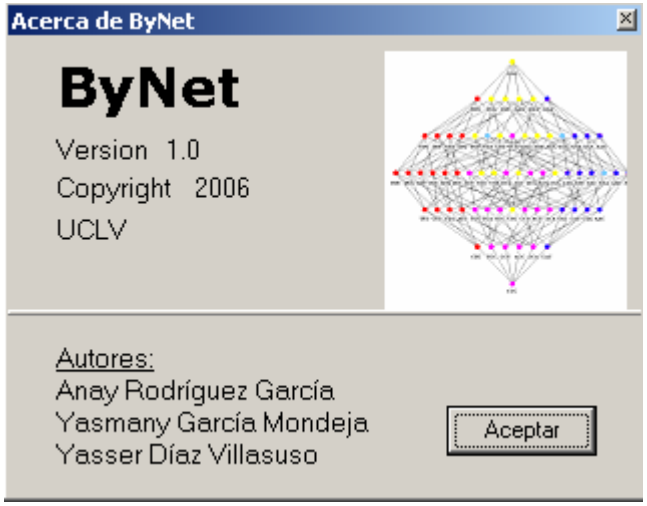


Figura 3.15 *Ventana con información acerca del sistema*

3.4 Análisis de los resultados

Para el análisis de los resultados se construye una red Bayesiana que utiliza datos de secuencias de ADN de mutaciones de la proteína (proteasa) del VIH, pues esta se considera altamente variable. La base de datos ha sido preparada en el grupo de Bioinformática de la UCLV. Para completar el modelo probabilístico que conforma la red se calculan las tablas de probabilidades condicionales.

La red Bayesiana creada será útil para inferir el comportamiento de las posiciones en la secuencia de ADN de una nueva mutación en una familia dada y para pronosticar a que familia deben pertenecer nuevas mutaciones.

La base de datos muestra el estudio para noventa y nueve codones, a su vez cada codón posee tres moléculas básicas del ADN llamadas nucleótidos, se diferencian entre ellas en que cada una posee una base nitrogenada diferente. Cada nucleótido contiene un fosfato, un azúcar (desoxirribosa) y una de las cuatro bases nitrogenadas: Adenina (A), Guanina (G), Timina (T) o Citosina (C). Las variables predictivas son los elementos de la secuencia codificados en forma binaria G↔00; A↔01; T↔10; C↔11 acorde con sus propiedades físico - químicas. Ellos forman un total de 99 * 6 = 594 variables o posiciones diferentes dentro de dicha secuencia.

Analizando el codón número ochenta y dos, al ejecutar el software se puede inferir que si el primer nucleótido presenta la base nitrogenada A o C, entonces el segundo nucleótido presenta como base nitrogenada G o T pues existe 100% de probabilidad de que la segunda posición sea uno, también se comprueba que existe una mayor probabilidad de que la clase de mutaciones sea la tres con un 64.3%. (ver figura 3.16)

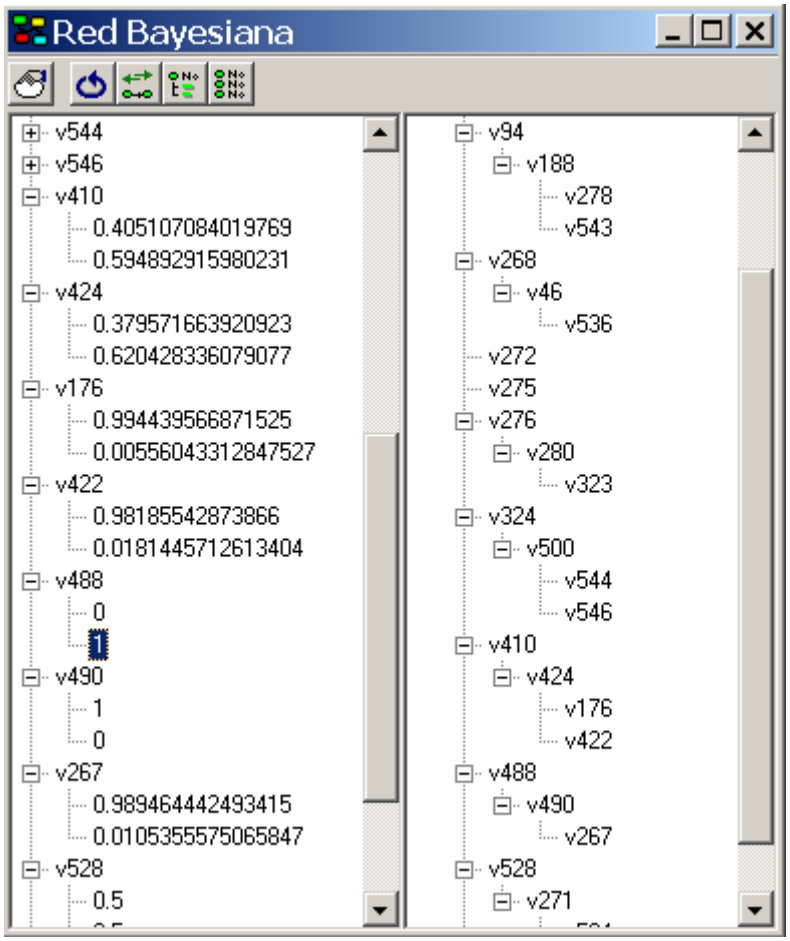


Figura 3.16 *Ventana que muestra resultados de la propagación*

Analizando el décimo codón, partiendo de que la red creada contiene las variables número 55 y 56, estas variables codifican el primer nucleótido del codón que está siendo analizado, si se sabe que la primera posición es cero se infiere que hay un 66.7% de probabilidad de que este nucleótido contenga la base nitrogenada G. Es posible inferir también la clase de mutaciones, en este caso tres en el 100% de los casos. (ver figura 3.17)

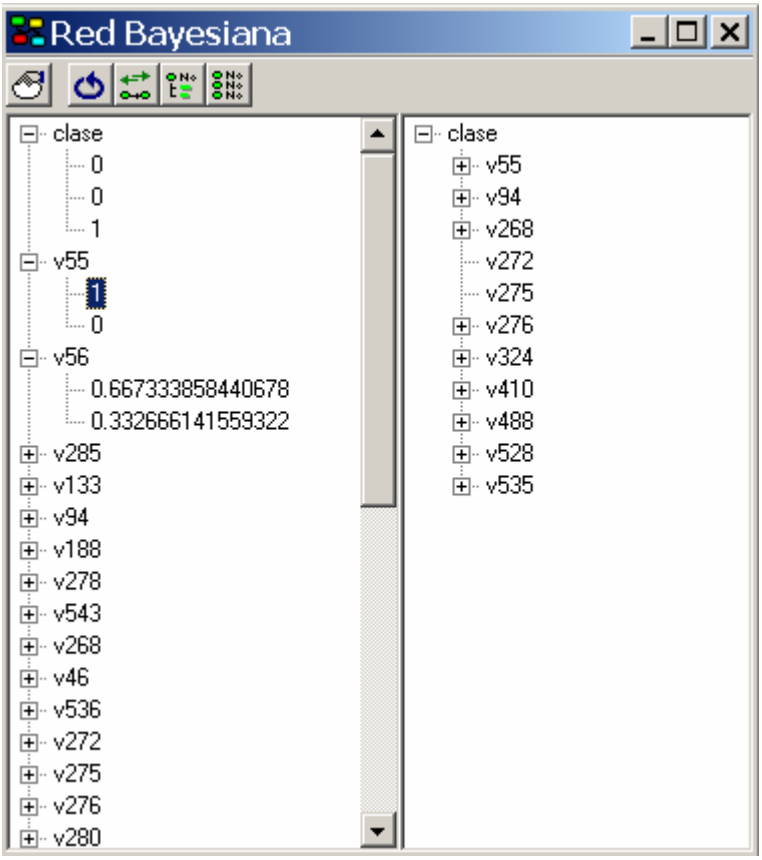


Figura 3.17 *Ventana que muestra resultados de la propagación*

De la misma forma se pueden evidenciar más de una posición dentro del conjunto de nucleótidos.

Conclusiones

Los algoritmos implementados, tanto para la obtención de las tablas de probabilidades como de propagación en árboles de unión, tienen complejidad NP-Completo. Debido a ello fue necesario trabajar con programación dinámica para hacer un uso eficiente de la memoria.

La herramienta desarrollada permite realizar un análisis de selección de atributos utilizando criterios de entropía y el estadístico Chi Cuadrado, aprendizaje en redes Bayesianas desde datos y la implementación del algoritmo de propagación en redes múltiplemente conexas mediante árboles de unión.

Las redes construidas son exportadas a los formatos estándares de archivos XML y XMLBIF, los cuales pueden ser utilizados por otros softwares.

Se realizaron comprobaciones de los resultados con paquetes profesionales como el SPSS y el WEKA.

Se validó el software con una aplicación en Bioinformática, específicamente en el estudio de la proteína proteasa donde los datos se obtuvieron desde secuencias de ADN con mutaciones del VIH.

Recomendaciones

Darle seguimiento a la construcción del software con el objetivo de conformar una herramienta más potente:

- Extender la entrada de datos desde otros formatos de archivos, por ejemplo: XML, XMLBIF.
- Implementar nuevos criterios de selección de atributos, con métodos actuales de Inteligencia Artificial como búsquedas mediante Hormigas, obtención de reductos con la técnica de los Conjuntos aproximados.
- Implementar otros métodos de propagación.
- Seguir ampliando la exportación de resultados a otros formatos, por ejemplo: GXL.
- Añadir al software módulos que permitan realizar experimentos con bases de datos de prueba.
- Realizar una variante del software que use las clases de construcción de la red Bayesiana y de propagación como una extensión del Weka.

Referencias Bibliográficas

BUNTINE, W. (1996) A guide to literature on learning graphical models. *IEEE Transactions and Knowledge Data Engeniering*.

CHÁVEZ, M., C. & RODRÍGUEZ, L. O. (2002) Bayshell, Software para crear redes bayesianas e inferir evidencias en la misma. IN 09358-9358 (Ed.) Registro de Software CENDA.

CASTILLO, E., GUTIÉRREZ, J. M. & S, H. A. (1996) *Sistemas Expertos y Modelos de Redes Probabilísticas*, Springer-Verlag.

CHÁVEZ, M. C., GRAU , R. & GARCÍA , M. (1999) Un método para construir redes bayesianas. Revista de Ingeniería de la Universidad de Antioquia. Medellín, Colombia ed.

HECKERMAN, D. (1996) A Tutorial on Learning With Bayesian Networks. MSR-TR-95-06.

HERNÁNDEZ, A. G. (2004) Aprendizaje Automático: Árboles de Decisión. *Facultad de Física e Inteligencia Artificial.* Xalapa, Universidad Veracruzana.

LIU, H. & SETIONO, R. (1995) Chi2: Feature Selection and Discretization of Numeric Attributes. Proc. 7th IEEE International Conf. on Tools with Artificial Intelligence, Washington D.C.

QUINLAN, J. R. (1986) Induccion of decision trees. *Machine Learning, 1(1), 81-106.*

SHANNON, C. & WEAVER, W. (1948) The mathematical theory of communication.

WITTEN, I. H. & FRANK, E. (2005) *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco.

Bibliografía

ALF-STEINBERGER, C. (1969) The genetic code and error transmission, Proc. Natl. Acad. Sci. USA.

BALAKRISHNAN, J. (2002) Symmetry scheme for amino acid codons. IN E, P. R. (Ed.).

BASHFORD, J. D. (2000) The genetic code as a periodic table, Biosystems.

BASHFORD, J. D., I.TSOHANTJIS & JARVIS, P. D. (1998) A supersymmetric model for the evolution of the genetic code, Proc. Natl. Acad. Sci. USA

BERTMAN, M. O. & JUNGCK, J. R. (1979) Group graph of the genetic code, J. Hered.

BUNTINE, W. (1996) A guide to literature on learning graphical models. IEEE Transactions and Knowledge Data Engeniering.

CASTILLO, E., GUTIÉRREZ, J. M. & S, H. A. (1996) Sistemas Expertos y Modelos de Redes Probabilísticas, Springer-Verlag.

CHAID, S. F. W. (1994) User Manual. SPSS Inc.

CHAID, S. F. W. (1994) User Manual. SPSS Inc.

CHÁVEZ, M., C. & RODRÍGUEZ, L. O. (2002) Bayshell, Software para crear redes bayesianas e inferir evidencias en la misma. IN 09358-9358 (Ed.) Registro de Software CENDA.

CHÁVEZ, M. C., GRAU , R. & GARCÍA , M. (1999) Un método para construir redes bayesianas. Revista de Ingeniería de la Universidad de Antioquia. Medellín, Colombia ed.

CRICK, F. H. C. (1968) The origin of the genetic code, J. Mol. Biol.

FREELAND, S. & HURST, L. (1998) The genetic code is one in a million, J. Mol. Evol.

FRIEDMAN, S. M. & WEINSTEIN, I. B. (1964) Lack of fidelity in the translation of ribopolynucleotides, Proc. Natl. Acad. Sci. USA.

HECKERMAN, D. (1996) A Tutorial on Learning With Bayesian Networks. MSR-TR-95-06.

HERNÁNDEZ, A. G. (2004) Aprendizaje Automático: Árboles de Decisión. Facultad de Física e Inteligencia Artificial. Xalapa, Universidad Veracruzana.

JIMÉNEZ-MONTAÑO, M. A. (1996) The hypercube structure of the genetic code explains conservative and non-conservative amino acid substitutions in vivo and in vitro, Biosystems.

JIMÉNEZ-MONTAÑO, M. A. (1999) Protein Evolution Drives the Evolution of the Genetic Code and Vice Vers, Biosystems

KARASEV, V. A. & STEFANOV, V. E. (2001) Topological Nature of the Genetic Code, J. Theor. Biol.

LIU, H. & SETIONO, R. (1995) Chi2: Feature Selection and Discretization of Numeric Attributes. Proc. 7th IEEE International Conf. on Tools with Artificial Intelligence, Washington D.C.

PARKER, J. (1989). Errors and alternatives in reading the universal genetic code, Microbiol. Rev.

QUINLAN, J. R. (1986) Induccion of decision trees. Machine Learning, 1(1), 81-106.

SÁNCHEZ, R., GRAU, R. & MORGADO, E. (2004) Genetic Code Boolean Algebras. WSEAS Transactions on Biology and Biomedicine

SÁNCHEZ, R., GRAU, R. & MORGADO, E. (2004) The Genetic Code Boolean Lattice, MATCH Commun. Math. Comput. Chem.

SÁNCHEZ, R., MORGADO, E. & GRAU , R. (2005) A Genetic Code Boolean Structure. I. Meaning of Boolean Deductions. Bulletin of Mathematical Biology.

SIEMION, I., SIEMION, P. J. & KRAJEWSKI, K. (1995) Chou-Fasman conformational amino acid parameters and the genetic code, Biosystems.

SWANSON, R. (1984) A unifying concept for the amino acid code, Bull. Math. Biol.

WITTEN, I. H. & FRANK, E. (2005) Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco.

Anexo 1

El siguiente fichero XMLBIF representa una red Bayesiana. En el mismo se definen las variables que formaran parte de los nodos de la red y los valores posibles de cada una, y además se describe la tabla de probabilidades por familia de nodos.

```
<BIF VERSION="3.0">
<NETWORK>
<NAME>Dog-Problem</NAME>

<! -- Variables -->
<VARIABLE TYPE="nature">
        <NAME>light-on</NAME>
        <OUTCOME>true</OUTCOME>
        <OUTCOME>false</OUTCOME>
        <PROPERTY>position = (73, 165) </PROPERTY>
</VARIABLE>

<VARIABLE TYPE="nature">
        <NAME>bowel-problem</NAME>
        <OUTCOME>true</OUTCOME>
        <OUTCOME>false</OUTCOME>
        <PROPERTY>position = (190, 69)</PROPERTY>
</VARIABLE>

<VARIABLE TYPE="nature">
        <NAME>dog-out</NAME>
        <OUTCOME>true</OUTCOME>
        <OUTCOME>false</OUTCOME>
        <PROPERTY>position = (155, 165) </PROPERTY>
</VARIABLE>

<VARIABLE TYPE="nature">
        <NAME>hear-bark</NAME>
        <OUTCOME>true</OUTCOME>
        <OUTCOME>false</OUTCOME>
        <PROPERTY>position = (154, 241) </PROPERTY>
</VARIABLE>

<VARIABLE TYPE="nature">
        <NAME>family-out</NAME>
        <OUTCOME>true</OUTCOME>
        <OUTCOME>false</OUTCOME>
        <PROPERTY>position = (112, 69) </PROPERTY>
</VARIABLE>


<! -- Probability distributions -->

<DEFINITION>
        <FOR>light-on</FOR>
        <GIVEN>family-out</GIVEN>
        <TABLE>0.6 0.4 0.05 0.95 </TABLE>
</DEFINITION>
```

```
<DEFINITION>
        <FOR>bowel-problem</FOR>
        <TABLE>0.01 0.99 </TABLE>
</DEFINITION>

<DEFINITION>
        <FOR>dog-out</FOR>
        <GIVEN>bowel-problem</GIVEN>
        <GIVEN>family-out</GIVEN>
        <TABLE>0.99 0.01 0.97 0.03 0.9 0.1 0.3 0.7 </TABLE>
</DEFINITION>

<DEFINITION>
        <FOR>hear-bark</FOR>
        <GIVEN>dog-out</GIVEN>
        <TABLE>0.7 0.3 0.01 0.99 </TABLE>
</DEFINITION>

<DEFINITION>
        <FOR>family-out</FOR>
        <TABLE>0.15 0.85 </TABLE>
</DEFINITION>


</NETWORK>
</BIF>
```

Anexo 2

El procedimiento para construir un fichero WEKA para ser utilizado en la herramienta ByNet, consta de tres partes fundamentales: nombrar relación, definir atributos y la clase con sus posibles valores, y construir la base de datos o ejemplos. Para cada sección hay una palabra reservada.

Nombrar relación: primero se incluye la palabra *@relation* y a continuación el nombre, por ejemplo:

@relation RED_BAYESIANA

Definir atributos y la clase con sus posibles valores: primero se incluye la palabra *@attribute* seguido el nombre de la variable y por último entre llaves los valores que puede tomar el mismo y así sucesivamente para cada atributo, el último atributo especifica la clase, por ejemplo:

@attribute V1 {0, 1}

@attribute V2 {0, 1}

@attribute V3 {0, 1}

@attribute V4 {0, 1}

@attribute V5 {0, 1}

@attribute V6 {0, 1}

@attribute proteína {0,1, 2}

Los atributos serían V1, V2, V3, V4, V5, V6 que pueden tomar los valores cero y uno respectivamente, mientras la clase proteína puede ser: cero, uno, dos.

Construir la base de datos o ejemplos: primero se coloca la palabra @data, a partir de la próxima línea se comienza a llenar la base de casos, cada línea de la base de casos

tendrá una cantidad de términos igual al número de atributos más la clase, separados por coma, por ejemplo:

@data

0, 0, 1, 0, 0, 0, 1

1, 0, 1, 0, 0, 1, 0

0, 1, 1, 0, 0, 1, 0

0, 0, 1, 0, 0, 0, 1

Cada línea tiene siete términos porque son seis atributos y una clase.

Ahora se muestra un ejemplo de un fichero WEKA completo:

@relation Data1

@attribute V1 {0,1}

@attribute V2 {0,1}

@attribute Class {0,1}

@data

```
0, 0, 0
0, 0, 1
0, 0, 1
0, 1, 0
0, 1, 1
0, 1, 1
1, 0, 0
1, 0, 1
1, 1, 1
1, 1, 0
```