UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS

FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN

DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

MODELOS DE REDES BAYESIANAS EN EL ESTUDIO DE SECUENCIAS GENÓMICAS Y OTROS PROBLEMAS BIOMÉDICOS

Monografía

MSc. María del Carmen Chávez Cárdenas

Dr. Ricardo Grau Ábalo

Dra. Gladys Casas Cardoso

Santa Clara, 2008

"Vivir es el arte de derivar conclusiones suficientes de premisas insuficientes"

S. Butler

SÍNTESIS

Este trabajo está relacionado con problemas de análisis de regiones genómicas codificantes para proteínas utilizando un tipo de modelo gráfico-probabilístico: las redes bayesianas. Las posibilidades del uso de las redes bayesianas se fortalece si se realiza el aprendizaje de las mejores estructuras y parámetros. En el trabajo se presentan tres nuevos algoritmos para el aprendizaje estructural desde datos. Dos de estos algoritmos obtienen la estructura de dependencias basándose en la detección de interacciones al estilo del algoritmo CHAID (*Chi-square Automatic Interaction Detector*).

El tercero de estos algoritmos se basa en un método de optimización bioinspirado, concretamente la optimización basada en enjambres de partículas (*Particle Swarm Optimization*, PSO) para contribuir a la reducción de atributos.

En la validación de estos algoritmos se han utilizado 18 archivos de datos del repositorio de aprendizaje automatizado, así como otros enfoques alternativos para el aprendizaje de la estructura de redes bayesianas, reportados anteriormente; cuyos resultados demuestran la validez de los modelos propuestos. Además se desarrollaron tres aplicaciones que responden a problemas reales de distintas áreas.

Los dos primeros problemas pertenecen al área de la Bioinformática, la primera aplicación es sobre la predicción de interacciones de proteínas y la segunda sobre predicción de sitios de *splicing* en regiones genómicas codificantes para proteínas. Para concluir se presenta una aplicación sobre un tema médico bien conocido: el diagnóstico de la hipertensión arterial.

ABSTRACT

The current thesis is concerned with the analysis of coding regions for proteins by using a type of graph-probabilistic model: Bayesian networks. The capabilities of the Bayesian networks are significantly enhanced as long as the best structures and parameters are properly learned. This study puts forward three new algorithms for structural learning from data. Two of them become cognizant about the dependency structure owing to the detection of the interactions like in the CHAID (*Chi-square Automatic Interaction Detection*) algorithm.

The third one of these approaches is anchored on a bio-inspired optimization method, i.e. the optimization driven by swarms of particles (*Particle Swarm Optimization, PSO*) to help reduce attributes.

Eighteen widely used data repositories from University of California at Irvine have been employed in the validation of the aforementioned algorithms, besides considering other alternative models previously reported in literature. The results attained demonstrate the feasibility of the proposed methods. In addition, three applications that respond to real problems in different fields were developed.

The first two problems lie under the umbrella of bioinformatics; the former is concerned with the prediction of protein interactions whereas the latter has to do with splicing sites forecasting. Last but not least, an application addressing the well-known problem of hypertension diagnosis is introduced.

Acrónimos

UCLV: Universidad Central "Marta Abreu" de Las Villas

CEI: Centro de Estudios de Informática

INIVIT: Instituto de Investigaciones de Viandas Tropicales

IA: Inteligencia Artificial, del inglés *Artificial Intelligence*

IBP: Instituto de Biotecnología de las Plantas

RB: Redes Bayesianas, del inglés *Bayesian Networks*

ADN: Ácido Desoxirribonucleico

ARN: Ácido Ribonucleico

ML: Aprendizaje automático o computarizado, del inglés *Machine Learning*

GDA: Grafo Dirigido Acíclico, del inglés *Directed Acyclic Graph*

BLAST: Herramienta de búsqueda de regiones similares entre secuencias biológicas, del inglés *Basic Local Alignment Search Tool*

FASTA: Sistema para comparar nucleótidos o proteínas, del inglés *FAST-All*

CHAID: Detector automático de interacciones Chi-cuadrado, del inglés *Chi-square Automatic Interaction Detector*

PSO: optimización basada en enjambres de partículas, del inglés *Particle Swarm Optimization*

Weka: plataforma de aprendizaje automatizado, implementada en Java por la Universidad de Waikato en Nueva Zelanda, del inglés *Waikato Environment for Knowledge Analysis*

HTA: HiperTensión Arterial

DPC: Distribución de Probabilidad Conjunta

IMC: Información Mutua Condicional

AIC: Criterio de Información de Akaike, del inglés *Akaike Information Criterion*

MDL: longitud de descripción mínima, del inglés *Minimal Description Length*

MNB: Modelo Naïve Bayes o MBN: Modelo Bayesiano Naïve o CNB: Clasificador Naïve Bayes

TAN: Naïve Bayes aumentado a árbol, del inglés *Tree Augmented Naïve Bayes*

kDB: clasificador bayesiano con *k* dependencias, del inglés *k Dependence Bayesian classifier*

PC: Constructor eficiente, del inglés *Power Constructor*

VP: verdaderos positivos, del inglés *true positive* (TP)

rVP: razón de VP, del inglés *true positive rate*

FP: falsos positivos, del inglés *false positive* (FP)

rFP: razón de FP, del inglés *false positive rate*

VN: verdaderos negativos, del inglés *true negative* (TN)

rVN: razón de VN, del inglés *true negative rate*

FN: falsos negativos, del inglés *false negative* (FN)

rFN: razón de FN, del inglés *false negative rate*

ROC: Curva de operación del receptor, del inglés *Receiving Operation Curve*

UCI: Universidad de California Irvine

UCIML: Bases de datos del repositorio de aprendizaje automático, del inglés UCI *Repository of Machine-Learning Databases*

EDAs: algoritmos de estimación de distribuciones, del inglés *Estimation of distribution algorithms*

GO: genes ontólogos, del ingles *Gene Ontology*

AUC: área bajo la curva ROC, del inglés, *Area under the Receiving Operation Characteristic Curve*

AD: Árbol de Decisión, del inglés *Decision Tree*

LOO-CV: validación cruzada dejando uno fuera, del inglés *Leave one out crossvalidation*

TABLA DE CONTENIDOS

INTRODUCCIÓN *1*

1. LAS REDES BAYESIANAS Y LA BIOINFORMÁTICA *11*

1.1 Redes Bayesianas **11**

1.1.1 Aprendizaje en Redes Bayesianas 13

1.1.1.1 Aprendizaje estructural de Redes Bayesianas 13

1.1.1.2 Aprendizaje paramétrico de Redes Bayesianas 21

1.1.2 Propagación en Redes Bayesianas 24

1.1.2.1 Propagación en árboles de unión 25

1.1.2.2 Algoritmo de propagación mediante la eliminación de variables 28

1.1.3 Las Redes Bayesianas como clasificadores 29

1.1.3.1 Necesidad de la reduccion de atributos en algunos casos 31

1.1.3.2 Optimización de enjambres de partículas 32

1.1.3.3 Evaluación de las Redes Bayesianas como clasificadores 34

1.1.4 Productos de *software* para Redes Bayesianas 36

1.2 Aplicaciones de las Redes Bayesianas en Bioinformática **37**

1.2.1 Estudio de secuencias genómicas 38

1.2.2 Problemas bioinformáticos que se resuelven mediante Redes Bayesianas 39

1.3 Consideraciones finales del capítulo **41**

2. NUEVOS ALGORITMOS DE APRENDIZAJE ESTRUCTURAL DE REDES BAYESIANAS *42*

2.1 Aprendizaje Estructural de Redes Bayesianas basado en técnicas estadísticas **42**

2.1.1 Aprendizaje Estructural de Redes Bayesianas basado en árboles de decisión obtenidos con el algoritmo CHAID 44

2.1.1.1 Fundamentos generales del Algoritmo 44

2.1.1.2 Algoritmo ByNet 47

2.1.1.3 Algunas consideraciones sobre el algoritmo ByNet 48

2.1.2 Aprendizaje Estructural de Redes Bayesianas basado en el algoritmo CHAID 49

2.1.2.1 Fundamentos generales del Algoritmo 49

2.1.2.2 Algoritmo BayesChaid 50

2.1.2.3 Algunas consideraciones sobre el algoritmo BayesChaid 51

2.2 Aprendizaje Estructural de Redes Bayesianas basado en técnicas de Inteligencia Artificial **52**

2.2.1 Técnicas de IA usadas en el manejo de información en Redes Bayesianas 52

2.2.2 Fundamentos generales del Algoritmo 53

2.2.3 Algoritmo BayesPSO 57

2.2.4 Algunas consideraciones sobre el algoritmo BayesPSO 58

2.3 Análisis del comportamiento de los algoritmos **59**

2.3.1 Análisis de la complejidad temporal 59

2.3.1.1 Análisis de la Complejidad temporal del Algoritmo ByNet 60

2.3.1.2 Análisis de la Complejidad temporal del Algoritmo BayesChaid 60

2.3.1.3 Análisis de la Complejidad temporal del Algoritmo BayesPSO 61

2.3.1.4 Comparación de los algoritmos 61

2.3.2 Ejemplo de aplicaciones para validar los resultados ... 62
2.4 Conclusiones parciales del capítulo ... 67
3. APLICACIONES DE LOS ALGORITMOS PROPUESTOS ... 68
3.1 Sobre la implementación de los algoritmos ... 68
3.2 Planteamiento del problema sobre predicción de interacciones de proteínas ... 71
3.2.1 Análisis de los datos ... 72
3.2.2 Rasgos del problema ... 73
3.2.3 Discusión de los resultados ... 73
3.2.4 Validación mediante el uso del modelo obtenido con el algoritmo ByNet ... 75
3.2.5 Mejorando el balance de verdaderos positivos y negativos ... 78
3.3 Planteamiento del problema sobre localización de *splice sites* ... 80
3.3.1 Análisis de datos ... 82
3.3.2 Rasgos del problema ... 82
3.3.3 Discusión de los resultados ... 83
3.3.4 Validación mediante el uso del modelo obtenido con el algoritmo *ByNet* ... 84
3.3.5 Mejorando la predicción de verdaderos splice sites ... 87
3.4 Predicción de la Hipertensión arterial ... 88
3.4.1 Análisis de los datos ... 90
3.4.2 Discusión de los resultados ... 90
3.4.3 Validación mediante el uso del modelo obtenido con el algoritmo *BayesChaid* ... 91
3.4.4 Mejorando la predicción de falsos sanos ... 91
Conclusiones parciales del capítulo ... 92
CONCLUSIONES Y RECOMENDACIONES ... 94
REFERENCIAS BIBLIOGRÁFICAS ... 96
Producción Científica del autor sobre el tema de la tesis ... 106
ANEXOS ... 109
Anexo 1. Conceptos básicos ... 109
Anexo 2. Comparación de paquetes de *software* de Modelos Gráficos: RB ... 115
Anexo 3. Clasificación de *Software* de Redes Bayesianas y Clasificadores Bayesianos en propietario y libre ... 118
Anexo 4. Técnicas y Herramientas de Genómica y Proteómica (Gibas y Per 2001) ... 120
Anexo 5. La Prueba Chi-cuadrado y la técnica de CHAID ... 122
Anexo 6. Características de las bases de datos del repositorio de la UCIML utilizadas para validar los algoritmos de aprendizaje estructural de Redes Bayesianas ... 126
Anexo 7. Red Bayesiana de clasificación de *donors* con el algoritmo ByNet. Ejemplos de propagación de evidencias con el *software* ELVIRA. ... 127
Anexo 8. Red Bayesiana de clasificación de *acceptors* con el algoritmo ByNet. Ejemplos de propagación de evidencias con el *software* ELVIRA. ... 129
Anexo 9. Red Bayesiana de diagnóstico de la HTA con el algoritmo BayesChaid. Ejemplos de propagación de evidencias con el *software* ELVIRA ... 131
Anexo 10. Diagrama de relación de Clases ... 133
Anexo 11. Sintaxis de los ficheros de datos para Weka y comandos para ejecutar Weka *parallel* ... 134

INTRODUCCIÓN

La secuenciación de genomas ha generado un amplio catálogo de miles de millones de secuencias de pares de bases nucleotídicas de ADN (Ácido desoxirribonucleico), o moléculas esenciales de la vida. Una de las dificultades que se afronta en los estudios biológicos actuales proviene, paradójicamente, de esta enorme cantidad de datos. Se conocen las secuencias (nucleotídicas o de aminoácidos para los cuales ellas codifican) de más de un millón y medio de proteínas, las de más de cien genomas (ver anexo 1 de conceptos básicos), la estructura tridimensional de más de 20 mil proteínas, etc. Gracias a los experimentos de matrices de ADN o micro arreglos (*micro arrays*) se sabe cuándo y cómo se expresan muchos genes; también se dispone de muchos datos que indican qué proteínas interactúan entre sí. Además, todo el conocimiento científico acumulado a lo largo de las últimas décadas se encuentra disperso en más de 12 millones de artículos (Galperin 2007).

La disponibilidad de genomas completos de muchas especies, además del humano, el volumen de información ubicado actualmente en las bases de datos públicas, por ejemplo la base de datos GenBank (Benson et al. 2005), y los ambiciosos proyectos masivos de estudios sobre la interacción entre proteínas, han generado un cambio de paradigma en las investigaciones biológicas: de una estrategia de extraer el máximo de información a partir de unos pocos datos, se ha pasado a la necesidad de obtener la información esencial a partir de grandes volúmenes de datos. Para sólo poner un ejemplo, cuando se secuencia un genoma se tiene poco más que una larga serie de letras (bases nucleotídicas) (Dopazo y Valencia 2002) que constituyen realmente instrucciones y datos complicados. Para avanzar en la compresión de la información que encierran estos libros de instrucciones se deben encontrar los genes y predecir su función y esto está lejos de ser resuelto para cualquiera de los genomas ya secuenciados.

Se ha dado un avance en el planteamiento de la estructura-función en los genes así como la interrelación entre ellos y a su vez, su relación, por ejemplo, con procesos metabólicos normales así como con enfermedades asociadas a factores hereditarios o transformaciones genéticas. Estos descubrimientos conllevan el manejo de una cantidad elevada de datos,

imposibles de procesar de forma manual y que exigen de aplicaciones informáticas especializadas. Por tal motivo son muy importantes los avances en el orden computacional que se aplican al procesamiento de los datos para convertirlos en información esencial. Por ejemplo, los 34 000 genes humanos (la cifra es aproximada) pueden dar lugar a varios cientos de miles de proteínas y funciones, cifra que se multiplica gracias al multiuso de secciones codificantes, facilitado por el evento conocido como "*splicing*" o corte de intrones[1] y además, a las modificaciones postraduccionales que pueden sufrir las proteínas.

El enfoque clásico, que consistía en conocer una determinada función y buscar el gen responsable, se transformó y creó un nuevo escenario donde se dispone de un importante número de genes desconocidos a los que es necesario asignar una función. Este nuevo momento dio lugar al desarrollo de la Bioinformática (Christos y Valencia 2003).

Existe consenso acerca de la necesidad de la revisión y adaptación de algoritmos y sistemas existentes en el campo de la Ciencia de la Computación con estos objetivos, e incluso, el diseño de nuevos algoritmos e implementaciones.

Antecedentes

Los estudios bioinformáticos que se desarrollan en el mundo tienen mucho de experimental, de uso de métodos de prueba y error, de abuso de hipótesis “ad-hoc”, además de ser inmensamente costosos por los materiales y la información que requieren, tanto para la experimentación biológica como para el procesamiento computacional.

En el año 2002 se crea el Grupo de Bioinformática en la Universidad Central “Marta Abreu” de Las Villas (UCLV) con objetivos específicos, que emprenden el estudio, desde el punto de vista matemático puro y estadístico, de estructuras algebraicas en el código genético con pretensiones de ayudar a predecir estructura, funciones, evolución o mutaciones en general. Estas investigaciones básicas obtienen un rápido éxito.

Una vez que estos resultados se han obtenido y publicado (Sánchez y Grau 2005), (Sánchez et al. 2004) , se hace necesario buscar nuevas herramientas computacionales que junto a

[1] intrones: segmentos no codificantes para proteínas que forman parte de los genes de organismos superiores y que se intercalan con los exones, o zonas codificantes en un gen.

estas representaciones algebraicas permitan perfeccionar el análisis de secuencias. Los enfoques de aprendizaje automático o *Machine Learning* (ML), por ejemplo las Redes Neuronales, los Modelos Ocultos de Markov, las Máquinas con Vectores Soporte, las Redes Bayesianas (RB), etc., se ajustan idealmente para dominios caracterizados por la presencia de grandes volúmenes de datos, modelos "ruidosos", y la ausencia de teorías generales que permitan hacer análisis determinísticos o incluso estadísticos.

La idea fundamental que se persigue es descubrir conocimiento o aprender automáticamente desde los datos, a través de un proceso de inferencia o modelo de adaptación. Una arquitectura unificada dentro de los métodos de aprendizaje automático es el enfoque probabilístico bayesiano para la modelación e inferencia (Baldi y Soren 2001).

Las RB son una técnica de Inteligencia Artificial (IA) que ha mostrado resultados relevantes frente a este tipo de datos. Ellas constituyen una representación del conocimiento que tiene en cuenta las relaciones entre las variables[2] y hacen una selección de las más importantes por su propia caracterización, a la vez que permiten hacer inferencias sobre las mismas y en particular pueden ser usadas para tareas de clasificación. Esencialmente, una RB es un grafo dirigido acíclico (GDA) y una distribución de probabilidad para cada nodo del grafo (Buntine 1996), (Castillo et al. 1997), (Heckerman 1996), (Charles River Analytics 2004).

La definición de una RB supone siempre dos tareas. La primera es determinar la estructura de relaciones de dependencia entre las variables "independientes"[3], digamos por ejemplo, las posiciones de una secuencia, en relación a una variable "dependiente". La segunda tarea es obtener la distribución de probabilidades (parámetros) que permitirá hacer inferencias. Entre estas dos tareas, la primera es esencial por ser realmente la más difícil y es imprescindible para poder realizar la segunda. Así, las posibilidades del uso de las RB se fortalece si es posible realizar el aprendizaje de las mejores estructuras y parámetros,

[2] Indistintamente se utilizan los términos variables, atributos o rasgos para referirnos a las variables predictoras en los problemas que se tratan, y cuando se habla de la variable dependiente se refiere como variable dependiente o clase.

[3] En las RB, el carácter "dependiente" o "independiente" de las variables es intercambiable. Aquí se utilizan estos términos por analogía con los de otras técnicas de pronóstico pero se anotan por esa razón entre comillas. Las llamadas variables independientes, son las predictivas de la variable dependiente u objetivo, pero no son independientes entre sí.

especialmente si se logra optimizar el aprendizaje estructural acorde con el dominio del campo de aplicación, en este trabajo la Bioinformática y en particular el análisis de secuencias genómicas. Se requiere además la implementación de estas nuevas técnicas de aprendizaje y de inferencia en productos de *software*, preferiblemente en plataformas de *software* libre para facilitar la divulgación y uso por la comunidad científica.

Las RB se han utilizado en Biología e incluso en Bioinformática (Wilkinson 2007), pero se usan técnicas muy generales de aprendizaje que tal vez no tienen en cuenta la información esencial de los datos biológicos o de las secuencias genómicas (Liu y Logvinenco 2003). Este es el campo de estudio. A continuación se detalla esta situación problémica.

Situación problémica

La genómica y la proteómica, generan continuamente grandes cantidades de datos que plantean problemas de gestión y análisis, lo cual enfrenta a la Bioinformática el reto de encontrar nuevas soluciones que permitan el procesamiento eficiente de dicha información. Los especialistas confrontan no solo el problema técnico que presenta el manejo de grandes volúmenes de datos, sino la búsqueda de nuevos algoritmos con los que se pueda extraer nuevo conocimiento desde datos ruidosos o sujetos a errores.

Las herramientas bioinformáticas clásicas más usadas en el contexto del análisis de secuencias incluyen métodos de búsqueda de secuencias similares e inducción de propiedades a partir de la similaridad. Los programas BLAST (*Basic Local Alignment Search Tool*)[4] y FASTA (*FAST-All*, (EBI 1999))[5] son muy conocidos; también el alineamiento múltiple (CLUSTAL es un algoritmo clásico para esta tarea), la definición de regiones conservadas con posible significado funcional, y el uso de estas regiones para buscar nuevas secuencias, así como métodos filogenéticos en aras de reconstruir relaciones evolutivas entre las secuencias (Cohen 2004). Esencialmente, ellas son herramientas de aprendizaje no supervisado o supervisado. Sin embargo, algunas de estas herramientas bioinformáticas, por ejemplo las de alineamiento están diseñadas para trabajar con una

[4] BLAST se utiliza para buscar regiones similares entre secuencias biológicas.
[5] FASTA permite hacer una comparación rápida de proteínas o nucleótidos.

cantidad relativamente pequeña de secuencias o de clases objetivo, y se limita así el procedimiento clásico para saber más sobre una secuencia que consiste, básicamente, en alinear ésta con otras disponibles en bases de datos, cuyas características o funciones son conocidas y "buscar" información sobre la misma a partir de similaridades con un grupo reducido de secuencias conocidas.

Un nuevo reto del análisis de secuencias biológicas está en la manipulación de mucha información, que además, puede contener incertidumbre. Usualmente los especialistas de bioinformática afrontan así la realización de dos tareas principales: clasificar los datos en grupos y después, investigar qué información tienen en común los miembros de cada grupo, que los distinguen del resto de los otros grupos. La ejecución de estas tareas se basa esencialmente en la aplicación de técnicas de agrupamiento y de la aplicación posterior de otros métodos que permitan extraer información característica de un grupo de elementos. Es dentro de la segunda tarea que se pretende utilizar las RB. Los métodos para extraer la información en la segunda fase pueden incluir cualquier técnica de aprendizaje supervisado; pero la extracción de conocimiento en el análisis de secuencias genómicas o datos biológicos no siempre constituye un problema de regresión o clasificación. Dada la incertidumbre presente en estos datos, resulta apropiada la aplicación de métodos bayesianos, por las ventajas que ofrece sobre las técnicas estadísticas y bioinformáticas convencionales (Silva y Muñoz 2000).

Las RB aventajan a métodos tradicionales de clasificación en dos aspectos esenciales:

1. Permiten realizar inferencias en presencia de información o evidencias incompletas.
2. Las inferencias pueden ser no solo sobre la "clase o variable dependiente" sino sobre cualquiera de las variables "independientes" cuya información se desconozca a partir de evidencias de otras variables.

Estos dos aspectos son típicos en los problemas actuales de análisis de secuencias. Por ejemplo, a partir de una base de datos de mutaciones de un virus con niveles conocidos de resistencia antiviral ante determinada droga, puede ser interesante el clásico problema de clasificación de la resistencia de una nueva mutación, aun cuando no se tengan disponibles todos los datos de ésta. Puede también ser interesante, a partir de cierto nivel de resistencia

deseado, conocer información probabilística sobre determinadas posiciones de esa secuencia, necesaria para obtener un determinado nivel de resistencia, así como combinaciones de las distintas interrogantes que se puedan presentar. Todos estos problemas se pueden resolver si se hacen diferentes inferencias con una RB única que tenga una buena estructura y una vez que se definan los parámetros asociados a la misma.

También en otras aplicaciones biológicas y médicas se presentan problemas similares. Por ejemplo, en el diagnóstico probabilístico diferenciado de una determinada enfermedad, a partir de una base de casos con información sobre riesgos y casos nuevos con información incompleta, o la investigación de la necesidad probabilística de un riesgo difícil de explorar ante casos con diagnóstico conocido.

Debido a las bondades que presentan las RB surge la idea de trabajar con este tipo de técnica; aunque esto no necesariamente alivia la solución de los problemas, y mucho menos la solución combinada con técnicas de la IA, si la estructura de la red exige el cálculo de un gran número de probabilidades condicionales o parámetros, como es usual. Se plantea entonces el problema de simplificar la estructura de la red con el apoyo de otros modelos gráficos probabilísticos o de optimización, así como en información concreta del dominio de aplicación, para en definitiva aliviar el cálculo de probabilidades, facilitar inferencias y reducir complejidad computacional.

Hay otras insuficiencias en el estado del arte actual de algunas aplicaciones computacionales. Por ejemplo, la plataforma inteligente para aprendizaje *Weka* (*Waikato Environment for Knowledge Analysis*) (Witten y Frank 2005), que es libre y de código abierto, tiene incorporadas muchas técnicas estadísticas o de IA y brinda la posibilidad de experimentar con el conjunto de ellas para investigar con cuáles se obtienen mejores resultados. Pero las RB que incluye hasta ahora usan sólo los métodos clásicos de aprendizaje y apenas permiten resolver tareas de clasificación, no así de inferencia inversa como las mencionadas anteriormente.

Además, en el campo de la aplicación al análisis de secuencias genómicas, existen muchos problemas abiertos, los cuales han sido abordados por diferentes técnicas, en particular, de clasificación, con resultados que aún no satisfacen las expectativas de los especialistas en

ciencias biológicas y que sugieren la aplicación de nuevos métodos con el propósito de alcanzar mejores desempeños en las predicciones. Entre los ejemplos de tales problemas se encuentran la localización de los sitios de *splicing*, la detección de interacciones de proteínas, la predicción de actividad antiviral y otros que serán abordados en la presente tesis con la aplicación de los métodos propuestos en la misma.

La comunidad bioinformática actual ha llegado al consenso de que ninguna técnica por separado dará una solución definitiva a varios de estos problemas, producto de las indeterminaciones propias de los procesos biológicos y la presencia de muchos ruidos o ausencia de información y ello reclama de los "*ensembles*" o "multiclasificadores". Es ello otra justificación para la búsqueda, casi interminable, de nuevos algoritmos que, desde una óptica diferente, puedan aportar elementos extras a la solución de tales problemas en conjunción con otros algoritmos o modelos. En este sentido, el presente trabajo contribuye a la detección de interacciones esenciales entre variables supuestamente predictivas para abordar tales problemas.

Consecuentemente se plantea el siguiente:

Objetivo general

Desarrollar e implementar nuevos algoritmos de aprendizaje estructural de RB a partir de la combinación de métodos clásicos con otros modelos gráficos como los árboles de decisión y los algoritmos de optimización bioinspirados, que simplifiquen la red, que tengan resultados con eficiencia similar o superior a las RB clásicas y otras técnicas en problemas de clasificación de carácter biológico, y capaces de ser utilizados efectivamente en el análisis de secuencias genómicas para extraer información múltiple y adicional de las mismas.

Este objetivo general se desglosa en los siguientes objetivos específicos:

- Desarrollar nuevos algoritmos de aprendizaje estructural de RB que conduzcan a redes relativamente simples, en las cuales se minimicen las relaciones esenciales de dependencia entre las variables, con eficiencia similar o superior a las ya existentes, y particularmente aplicables en estudios bioinformáticos y biomédicos.

- Realizar la implementación computacional de los métodos propuestos en plataformas de *software* libre, de modo que se facilite su utilización práctica por la comunidad científica internacional, y a su vez poder compararlos con otros modelos clásicos de RB u otras técnicas de aprendizaje.
- Ilustrar cómo los modelos desarrollados pueden contribuir a la solución de problemas reales y aun abiertos de Bioinformática, relacionados con el análisis de secuencias genómicas, e ilustrar su generalidad con las posibilidades de aplicación también en otros problemas de diagnóstico médico.

Para el cumplimiento de estos objetivos se trazaron las siguientes

Tareas de investigación

1. Confección del marco teórico relacionado con la teoría de las RB y las experiencias reportadas de aplicación a la Bioinformática. Revisión de la teoría relacionada con los modelos que se pretenden combinar.
2. Desarrollar y formalizar nuevos algoritmos de aprendizaje estructural de RB basados en:
   a. Integración de árboles de decisión obtenidos con el algoritmo de detección de interacciones basado en Chi-cuadrado (CHAID)
   b. Detección de interacciones esenciales, perfeccionado el algoritmo de búsqueda de las mismas
   c. Algoritmos de optimización bioinspirados, concretamente la optimización basada en enjambres de partículas, para contribuir a la reducción de atributos.
3. Implementar y evaluar los tres algoritmos elaborados sobre la plataforma *Weka* y realizar la validación cruzada en forma paralela para así facilitar la evaluación de los algoritmos en problemas bioinformáticos.
4. Mostrar y evaluar los resultados de la aplicación en problemas tales como:
   a. Detección de interacciones entre proteínas
   b. Localización de genes a través de la predicción de *splice sites*
   c. Diagnóstico médico de la hipertensión arterial (HTA)

Novedad Científica

La novedad científica y el consecuente valor teórico del presente trabajo se resumen en los siguientes puntos:

1. Se desarrollan y formalizan tres nuevos algoritmos de aprendizaje estructural de RB combinando técnicas clásicas con árboles de decisión, técnicas novedosas de detección de interacciones y el algoritmo de optimización inspirado en bandadas de pájaros que permiten reducir la estructura de dependencias y los cálculos consecuentes.
2. Se muestran nuevos enfoques para afrontar problemas aún no resueltos cabalmente en Bioinformática, relacionados con la localización de genes a través de sitios de *splicing* y la detección de interacciones entre proteínas. Se ilustra además la generalidad de los enfoques en un problema de diagnóstico médico relacionado con el diagnóstico de la HTA.

La novedad está avalada por las publicaciones que se describen al final de la tesis.

Valor práctico

Desde este punto de vista, el valor del trabajo radica en la disponibilidad de la implementación de los algoritmos, como extensiones al *software* libre *Weka*, con distribución de datos para la ejecución de las validaciones cruzadas para ejecutar en un cluster de computadoras o en máquinas conectadas en red. Ello facilita:

1. El uso libre por la comunidad científica a nivel nacional e internacional con posibilidades de comparar con otros algoritmos implementados también sobre *Weka*, tanto en aplicaciones bioinformáticas como en otras áreas.
2. El abordar en particular problemas que requieren inferencias en múltiples direcciones además de clasificación (los modelos de RB implementados sobre *Weka* solo trabajan como clasificadores).
3. El aportar nuevos algoritmos que resulten ser eficientes en la solución de problemas planteados en la bioinformática actual.

Después de la revisión de la literatura y el desarrollo consecuente del marco teórico, se formuló la siguiente hipótesis de investigación:

Hipótesis de investigación

Los modelos gráfico probabilísticos, específicamente los árboles de decisión y las técnicas de detección de interacciones, así como la optimización inspirada en bandadas de partículas permiten definir nuevos algoritmos de aprendizaje estructural de RB que resultan más simples y que pueden ser utilizadas con eficiencia similar o superior a los ya existentes, en diversos problemas biológicos, específicamente en el análisis de regiones genómicas codificantes para proteínas y en problemas Biomédicos.

Estructura de la tesis

El trabajo se presenta esencialmente en tres capítulos a partir de la presente Introducción. El Capítulo 1 se dedica a la elaboración del marco teórico desde el punto de vista de las tendencias actuales en el desarrollo y evaluación de las RB. Se muestran algunas aplicaciones interesantes de estas técnicas, especialmente en el campo de la Bioinformática. En el Capítulo 2 se propone y formalizan matemáticamente los tres nuevos algoritmos de aprendizaje de la estructura de RB. Se realiza una validación de los algoritmos, para lo que se utilizan 18 archivos de datos de la UCIML (UCI *Repository of Machine-Learning Databases*) y un análisis de la complejidad temporal de los mismos. El Capítulo 3 está dedicado a mostrar el comportamiento de los nuevos algoritmos en dos problemas bioinformáticos y en la predicción de HTA como ejemplo de aplicación en otras ramas. Se formulan finalmente las conclusiones y recomendaciones, se detallan las referencias bibliográficas y se incluyen algunos anexos para mostrar detalles complementarios.

1. LAS REDES BAYESIANAS Y LA BIOINFORMÁTICA

El presente capítulo se dedica a sustentar teóricamente el tema de la tesis, por lo que se analizan aquellos enfoques y antecedentes relacionados con las RB y su aplicación, por ejemplo a problemas bioinformáticos de análisis de secuencias. Se provee un marco de referencia para interpretar las soluciones a los problemas desarrollados, y se exponen los conceptos relacionados con la Bioinformática en función de las aplicaciones que se desarrollan. Se analizan los problemas actuales existentes en esta temática relacionados con la obtención de RB desde datos y la posible aplicación en problemas de la vida real.

1.1 Redes Bayesianas

Las redes probabilistas son representaciones gráficas de las variables y de las relaciones entre las variables que caracterizan un problema (Wiltaker 1990). Las RB son un tipo muy popular de redes probabilísticas (Charles River Analytics 2004), que proveen información sobre las relaciones de dependencia e independencia condicional existentes entre las variables. La inclusión de las relaciones de independencia en la propia estructura de la red, hace de las RB una buena herramienta para representar conocimiento de forma compacta pues se reduce el número de parámetros necesarios. Estas relaciones simplifican la representación de la función de probabilidad conjunta como el producto de las funciones de probabilidad condicional de cada variable.

Al representar una distribución de probabilidad, las RB tienen una semántica clara, lo que permite procesarlas para hacer diagnóstico, aprendizaje, explicación, e inferencias (Heckerman 1996). Según la interpretación, pueden representar causalidad y se refieren como redes causales (Spirtes 1993), (Pearl 1993), pero no necesariamente tienen que representar relaciones de causalidad, sino de correlación (Grau et al. 2004).

Según (Stuart y Norvig 1996), (Stuart y Norvig 2003), una RB se define como un grafo que cumple lo siguiente:

- Los nodos de la red son variables aleatorias que se denotan con la letra X o con subíndices X_1, X_2, ...X_n. En principio estas variables pueden representar rasgos o atributos, pero puede ocurrir también que un rasgo original tenga que ser

descompuesto en varias variables aleatorias. Por ejemplo, si el rasgo tiene múltiples valores puede desearse trabajar con variables aleatorias dicotómicas, una por cada valor del rasgo original

- Cada par de nodos se conecta entre sí mediante arcos dirigidos. El significado de un enlace que va del nodo *X* al nodo *Y* es el de que *X* ejerce una influencia directa sobre *Y*. En términos de probabilidades esto significa que hay una dependencia condicional de *Y* respecto a *X*, esto es que la probabilidad de *Y* es diferente de la probabilidad de *Y* dado *X*.
- Por cada nodo hay una tabla de probabilidad condicional que sirve para cuantificar los efectos de los padres sobre el nodo. Los padres de un nodo son aquellos nodos cuyos arcos apuntan hacia éste.
- El grafo no tiene ciclos dirigidos (por lo tanto es un GDA). Esto significa que no se presentan ambigüedades en el encadenamiento de probabilidades condicionales por el hecho de influencias directas cíclicas.

Vista la RB como el grafo junto con las tablas de probabilidad condicional, se puede interpretar como una representación bien aproximada de la función de distribución de probabilidad conjunta (DPC)[6] de la variable clase y de todos los rasgos predictores. La red en sí codifica un conjunto de aseveraciones de independencia condicional. Las tablas de probabilidades condicionales completan la caracterización de la distribución conjunta.

El grafo es importante para construir la red en sí. Los valores que aparecen en las tablas de probabilidad condicional son imprescindibles en el procedimiento de inferencia. Esta representación es a lo que algunos autores llaman *I–mapa minimal* de la distribución conjunta (Castillo et al. 1997), (García 1990).

Formalmente esta representación de la DPC define un modelo de RB, como un par (*G, P*), donde *G* es un GDA, $P = \{p(X_1|\tau_1),\ p(X_2|\tau_2),\ \ldots,\ p(X_n|\tau_n)\}$ es un conjunto de *n* distribuciones de probabilidad condicionales, una por cada variable X_i (nodos del grafo), y

[6] Ver definición en anexo 1 de conceptos básicos.

τ_i es el conjunto de padres del nodo X_i en G. El conjunto P define la DPC asociada, como muestra la expresión:

$$p(X) = \prod_{i=1}^{n} p(X_i|\tau_i) \qquad X = (X_1, X_2, ..., X_n) \tag{1.1}$$

1.1.1 Aprendizaje en Redes Bayesianas

¿Por qué es de interés el aprendizaje de redes probabilísticas o RB?. El interés de la IA en el aprendizaje de RB, se debe a la asociación entre la incertidumbre y el aprendizaje automático (Buntine 1994; Buntine 1995). El problema del aprendizaje bayesiano puede describirse informalmente así: dado un conjunto de entrenamiento $D = \{d_1, d_2, ..., d_n\}$ de instancias del problema, encuéntrese la RB que se ajuste mejor a D. Típicamente, este problema se divide en dos aspectos:

- Aprendizaje estructural: obtener la estructura de la RB, es decir, las relaciones de dependencia e independencia condicional entre las variables involucradas.
- Aprendizaje paramétrico: dada una estructura de RB, obtener las probabilidades a priori y condicionales requeridas.

Las técnicas de aprendizaje estructural dependen del tipo de estructura de red: árboles, poli árboles y redes múltiplemente conexas. Otra alternativa es combinar conocimiento subjetivo del experto con aprendizaje. Para ello se parte de la estructura dada por el experto, la cual se valida y mejora utilizando datos estadísticos. Resulta evidente que la calidad de una red obtenida de esta manera depende mucho del conocimiento sobre el dominio de aplicación de los encuestados. Esta última opción no siempre es aplicable en problemas bioinformáticos.

1.1.1.1 Aprendizaje estructural de Redes Bayesianas

Uno de los modelos más simples, y que por su facilidad de utilización se ha convertido en un estándar con el cual comparar las bondades de los diferentes métodos, es el denominado modelo bayesiano Naïve (MBN) o Naïve-Bayes (Duda y Hart 1973). Su denominación proviene de la hipótesis de que las variables predictivas son condicionalmente independientes dada la variable a clasificar y con esto ya queda definida una estructura, por

lo que sólo se tienen que aprender las probabilidades de los valores de los atributos dada la clase. Pero es obvio que esta suposición de independencia es demasiado fuerte.

Una forma de mejorar la estructura de un MBN se logra si se añaden arcos entre los nodos o atributos que tengan cierta dependencia. Se han realizado varias generalizaciones del MBN (Friedman y Goldszmidt 1996), (Larrañaga 2000). Otra forma es realizando operaciones locales hasta que no mejore la predicción:

- eliminar un rasgo o atributo,
- unir dos atributos en una nueva variable combinada,
- introducir un nuevo atributo que haga que dos atributos dependientes sean independientes (nodo oculto).

Se pueden ir probando cada una de las opciones anteriores midiendo la dependencia de los atributos dada la clase o información mutua condicional (IMC) según la expresión:

$$IMC(X_i, X_j \mid C) = \sum_{X_i, X_j} P(X_i, X_j \mid C) \frac{\log(P(X_i, X_j \mid C)}{P(X_i \mid C)\, P(X_j \mid C)} \qquad (1.2)$$

Algoritmo para árboles

El método para aprendizaje estructural de árboles se basa en el algoritmo desarrollado por (Chow y Liu 1968) para aproximar una distribución de probabilidad por un producto de probabilidades de segundo orden.

Se trata de un problema de optimización para obtener la estructura en forma de árbol que más se aproxime a la distribución "real". Para ello se utiliza una medida de la diferencia de información entre la distribución real (P) y la aproximada (P^*) usando la expresión:

$$I(P, P^*) = \sum_x P(X) \log(P(X) / P^*(X)) \qquad (1.3)$$

El objetivo es minimizar I. Para ello se puede definir esta diferencia en función de la información mutua entre pares de variables, de la siguiente forma:

$$I(X_i, X_j) = \sum_x P(X_i, X_j) \frac{\log(P(X_i, X_j))}{P(X_i)\,P(X_j)} \qquad (1.4)$$

Se puede demostrar (Chow y Liu 1968) que la diferencia de información es una función del de la suma de las informaciones mutuas (pesos) de todos los pares de variables que constituyen el árbol con signo cambiado. Por ello se necesita encontrar el árbol con mayor peso. Basado en lo anterior, el algoritmo para determinar el árbol bayesiano óptimo a partir de datos es el siguiente:

1. *Calcular la información mutua entre todos los pares de variables (n(n-1)/2).*
2. *Ordenar las informaciones mutuas de mayor a menor.*
3. *Seleccionar la rama de mayor valor como árbol inicial.*
4. *Agregar la siguiente rama mientras no forme un ciclo; si lo forma, desechar.*
5. *Repetir (4) hasta que se cubran todas las variables (n-1 ramas).*

El algoritmo no provee la direccionalidad de los arcos, por lo que ésta se puede asignar en forma arbitraria o utilizando una semántica externa (experto). Además el algoritmo realiza una búsqueda incompleta, por lo que se puede producir el problema de sobre ajuste, para lo cual se sugiere en la literatura el uso del estadístico Chi-cuadrado.

El algoritmo 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 de ascensión de colinas, por ejemplo, caer en máximos o mínimos locales (Hernández 2004).

Algoritmo para poli árboles

En (Rebane y Pearl 1988) se extiende el algoritmo de Chow y Liu (Chow y Liu 1968) para poli árboles. Para ello se parte del esqueleto (estructura sin direcciones) obtenido con el algoritmo anterior y se determina la dirección de los arcos utilizando pruebas de dependencia entre tripletas de variables. De esta forma se obtiene una red bayesiana en forma de poli árbol (ver anexo 1 sobre conceptos básicos). El algoritmo de (Rebane y Pearl 1988) se basa en probar las relaciones de dependencia entre todas las tripletas de variables en el esqueleto. Dadas tres variables, existen tres casos posibles:

1. Arcos divergentes: $X \leftarrow Y \rightarrow Z$
2. Arcos secuenciales: $X \rightarrow Y \rightarrow Z$
3. Arcos convergentes: $X \rightarrow Y \leftarrow Z$

Los primeros dos casos son indistinguibles, pero el tercero es diferente, ya que las dos variables "padre" son marginalmente independientes. Entonces el algoritmo consiste en:

1. *Obtener el esqueleto utilizando el algoritmo de Chow y Liu.*
2. *Recorrer la red hasta encontrar una tripleta de nodos que sean convergentes (tercer caso) o nodo "multipadre".*
3. *A partir de un nodo "multipadre" determinar las direcciones de los arcos utilizando la prueba de tripletas hasta donde sea posible (base causal).*
4. *Repetir 2-3 hasta que ya no se puedan descubrir más direcciones.*
5. *Si quedan arcos sin direccionar, utilizar semántica externa para obtener su dirección.*

El algoritmo está restringido a poli árboles y no garantiza obtener todas las direcciones. Desde el punto de vista práctico, un problema es que generalmente no se obtiene independencia absoluta (información mutua cero), por lo que habría que considerar una cota empírica que "aproxima" la independencia.

Algoritmos para redes múltiplemente conexas

Existen dos clases de métodos para el aprendizaje genérico de RB, que incluyen redes múltiplemente conexas. Éstos son:

1. Métodos basados en medidas de ajuste y búsqueda.
2. Métodos basados en pruebas de independencia o basados en restricciones.

Los métodos del tipo uno, tienen dos aspectos principales: una medida para evaluar que tan buena es cada estructura respecto a los datos y un método de búsqueda que genere diferentes estructuras hasta encontrar la óptima, de acuerdo a la medida seleccionada.

Esta es la ventaja esencial de estos métodos respecto a los basados en restricciones, pues estos últimos encuentran un único modelo basado en la información categórica de la independencia condicional entre las variables.

Entre los algoritmos de búsqueda, los clásicos son dos algoritmos de búsqueda golosa (en inglés *greedy search*): *K*2 (Cooper y Herskovits 1992), que opera en el espacio de los GDA que son compatibles con un orden dado y el algoritmo *B* de (Buntine 1994), el cual no tiene en cuenta el orden de las variables. A continuación se describen ambos algoritmos.

Algoritmo K2

Entrada: Un conjunto de variables ordenadas: $\{X_1, \ldots, X_N\}$.

Salida: Una estructura de RB *G*

Etapa de Iniciación:

Para $i=1$ ***hasta*** N ***hacer***

$\Pi_i = \phi$ *// El conjunto de padres de la variable i es vacío*

Etapa Iterativa:

Para $i=1$ ***hasta*** N ***hacer***

Repetir

// Π_i *conjunto de padres de la variable i y* $q_i(\Pi_i)$ *contribución de la variable* X_i *con conjunto de padres* Π_i

Seleccionar el nodo $Y \in \{X_1, \ldots, X_{i-1}\} \setminus \Pi_i$ *que maximiza* $g = q_i(\Pi_i \cap \{Y\})$

$\delta \leftarrow g - q_i(\Pi_i)$

sí $\delta > 0$ ***entonces***

$\Pi_i \leftarrow \Pi_i \cap \{Y\}$

hasta que $\delta \leq 0$ *ó* $\Pi_i = \{X_1, \ldots, X_{i-1}\}$

Algoritmo B

Entrada: Un conjunto de variables $\{X_1, \ldots, X_N\}$.

Salida: Una estructura de RB G

Etapa de Iniciación:

Para $i=1$ *hasta* N

$\Pi_i = \phi$

Para $i=1$ ***hasta*** N *y* ***Para*** $j=1$ ***hasta*** N ***hacer***

sí $i \neq j$ ***entonces***

$A[i, j] = m_i(X_j) \,.\, m_i(\phi)$

en otro caso

$A[i, j] = -\infty$ *//no permitir* $X_i = X_j$

Etapa Iterativa:

repetir

Seleccionar los i, j, que maximizan A [i, j]

sí $A[i, j] > 0$ ***entonces***

$\Pi_i \leftarrow \Pi_i \cap \{X_j\}$

*// Ascen$_i$ son los ascendientes de la variable X_i y Desc$_i$ son los descendientes del nodo X_i (*ver anexo 1*)*

para $X_a \in Ascen_i$, $X_b \in Desc_i$ ***hacer***

$A[a, b]=-\infty$ *//no permitir ciclos*

para $k = 1$ ***hasta*** N ***hacer***

sí $A[i, k] > -\infty$ ***entonces***

$A[i, k] = m_i(\Pi_i \cap \{X_k\}) \,.\, m_i(\Pi_i)$

hasta que $A[i, j] \leq 0$ *o* $A[i, j] = -\infty, \ \forall i, j$

Los algoritmos K2 y B, no garantizan encontrar la solución óptima. Por otra parte, es cierto que el algoritmo K2 es uno de los más rápidos para aprendizaje en RB y puede utilizarse para problemas supervisados y no supervisados, pero depende del orden que se establece entre las variables. No siempre es posible obtener el orden, por ejemplo las posiciones de las secuencias genómicas no son intercambiables y no es fácil establecer a priori un orden total de importancia (Acid y De Campos 2003), (Kjærulff y Madsen 2008).

El algoritmo B al igual que el algoritmo K2 inicializa el conjunto de padres de un nodo como vacío. En el caso del algoritmo K2 la no existencia de ciclos lo garantiza el orden preestablecido en las variables; el algoritmo B por su parte, chequea en cada paso que no se formen ciclos. Ambos algoritmos verifican además que al añadir un nuevo arco, este mejore la medida de calidad que se emplea. El algoritmo K2 termina cuando al añadir un padre a una variable no incrementa la medida de calidad que se utiliza y ya no quedan más variables. Este algoritmo no garantiza obtener la red con mayor valor de probabilidad.

La complejidad computacional reportada para ambos algoritmos está basada en la complejidad por nodo y las veces que se aplica la métrica de calidad. Concretamente, si para un nodo la complejidad es del orden $O(M.K.T)$, y su calidad se calcula a lo sumo en el orden $O(N^2.K)$ veces, la complejidad del peor caso en ambos algoritmos es de orden $O(N^2.K^2.M.T)$, y como $K \leq N$ se obtiene prácticamente una complejidad máxima $O(N^4.M.T)$ (Neapolitan 1990), (Bouckaert 1995).

En las expresiones anteriores, los parámetros considerados son:

M: número de casos en la base de ejemplos,

K: cantidad máxima de padres,

N: número de variables o rasgos del problema *y*

T: cantidad máxima de valores posibles para las variables.

Existen varias medidas para evaluar las RB obtenidas. En (Bouckaert 1995) se hace un análisis de cada una; las más utilizadas son:

- *Medida bayesiana*: estima la probabilidad de la estructura dado los datos (verosimilitud) la cual se trata de maximizar.

Considerando variables discretas y que los datos son independientes, para cada estructura de red se puede calcular la frecuencia de los datos originales correctamente predichos por dicha estructura y comparar estas ocurrencias.

- *Longitud de descripción mínima* (en inglés *Minimal Description Length, MDL*): estima la longitud (tamaño en bits) requerida para representar la probabilidad conjunta con cierta estructura, la cual se compone de dos partes: representación de la estructura y representación del error de la estructura respecto a los datos.

La medida *MDL* hace un compromiso entre la exactitud y la complejidad del modelo. La exactitud se estima midiendo la información mutua entre los atributos y la clase; y la complejidad contando el número de parámetros.

La exactitud se puede estimar en base al "peso" de cada nodo, en forma análoga a los pesos en el método de aprendizaje de árboles. En este caso el peso de cada nodo se estima en base a la información mutua con sus padres, el peso (exactitud) total está dado por la suma de los pesos de cada nodo (Bouckaert 1995), (Morales 2006).

A diferencia del enfoque basado en una medida global, el enfoque basado en pruebas de independencia usa medidas de dependencia local entre subconjuntos de variables.

Entre los algoritmos más populares para el aprendizaje de RB basado en pruebas de independencias se encuentra el algoritmo *PC* (*Power Constructor, Constructor eficiente*) de (Spirtes y Meek 1995). Estas pruebas de independencia resultan costosas y es obvio que se convierte en un problema sobre todo cuando se analizan secuencias largas, por ejemplo de una proteína mediana, para no citar un genoma completo. A continuación se describe esqueléticamente el algoritmo.

Algoritmo PC

Entrada: Un conjunto de variables $\{X_1, \ldots, X_N\}$.

Salida: Una estructura de red bayesiana G

Paso 1. Realizar pruebas de independencia condicional entre cada par de variables.

Paso 2. Identificar el esqueleto o grafo no dirigido en función de la independencia o no entre las variables

Paso 3. Identificar los arcos convergentes

Paso 4. Identificar los arcos secuenciales y los arcos divergentes

En general ninguno de los algoritmos para obtener la estructura de RB es considerado mejor, se debe decidir cual es más conveniente usar. Esto depende del problema que se quiere resolver. Si las variables del problema están fuertemente correlacionadas, no es posible usar MNB, pues supone independencia entre todas las variables. Tampoco se debe usar la arquitectura TAN, pues solo se tiene en cuenta la dependencia de las variables predictivas y la clase, y a lo sumo la relación entre dos variables predictivas, o sea se puede tener hasta dos padres, incluida la variable clase.

Estos modelos son simples, pero la suposición de independencia que se hace los limita. Aunque hay muchas referencias de utilización en la bibliografía del modelo MNB, es solo por su simplicidad.

Los algoritmos K2 y B, también tienen limitaciones, pues son algoritmos de búsqueda golosa con todas las limitaciones de la misma. El algoritmo K2 además está limitado por el orden que se debe establecer entre las variables. El algoritmo B está limitado también por las pruebas para la no existencia de ciclos.

El uso del algoritmo PC está limitado por la cantidad de pruebas de independencia, sobre todo cuando se consideran dominios de aplicación con muchas variables (por ejemplo más de cien).

1.1.1.2 Aprendizaje paramétrico de Redes Bayesianas

El aprendizaje paramétrico consiste en encontrar los parámetros asociados a una estructura dada de una RB. Dichos parámetros consisten en las probabilidades *a priori* de los nodos raíz y las probabilidades condicionales de las demás variables, dados sus padres.

Para dominios de aplicación en los que existe conocimiento *a priori* es posible en principio, determinar las probabilidades a partir de expertos. Para poder brindar esta información, por ejemplo en dominios médicos, estos especialistas deberían ser capaces de brindar con "bastante certeza" los valores de todas las probabilidades condicionales que exige la estructura de la red. Sin embargo cuando esa estructura es medianamente compleja, la tarea

de determinar las probabilidades de cada uno de los nodos a partir de conocimiento a priori se hace difícil incluso para buenos expertos. En dominios bioinformáticos esta posibilidad difícilmente se tiene, por la complejidad intrínseca de los problemas biológicos involucrados. En cualquier caso, si se dispone de una base de datos, una opción más prometedora o al menos tranquilizadora, es extraer el conocimiento desde los datos.

Si en la fase de aprendizaje se conocen datos con todas las variables, es fácil obtener las probabilidades requeridas. Las probabilidades previas corresponden a las marginales de los nodos raíz, y las condicionales se obtienen de las conjuntas de cada nodo con su(s) padre(s).

El aprendizaje paramétrico depende de las características de los datos que se utilicen, si se tiene en cuenta que estos pueden ser completos o no. En la tesis se trabaja fundamentalmente con datos de aplicaciones Bioinformáticas, en los cuales no siempre hay conocimiento a priori sobre la información de que se dispone (precisamente es lo que se busca), y por tanto es necesario aprender desde los datos.

Aprendizaje paramétrico con datos completos

El aprendizaje de los parámetros es simple cuando todas las variables son completamente observables en el conjunto de entrenamiento. El método más común es el llamado estimador de máxima verosimilitud, que consiste esencialmente en estimar las probabilidades deseadas a partir de la frecuencia de los valores de los datos de entrenamiento.

La calidad de estas estimaciones dependerá de que exista un número suficiente de datos en la muestra. Cuando esto no es posible se puede cuantificar la incertidumbre existente representándola mediante una distribución de probabilidad *a priori*, para así considerarla explícitamente en la definición de las probabilidades. Habitualmente se emplean distribuciones Beta (Saucier 2000) en el caso de variables binarias, y distribuciones Dirichlet (Neapolitan 1990) para variables multivaluadas. Esta aproximación es además útil cuando se cuenta con el apoyo de expertos en el dominio de aplicación para concretar los valores de los parámetros de las distribuciones.

Si existen variables de tipo continuo la estrategia más habitual es discretizarlas antes de construir el modelo estructural. Existen algunos modelos de RB con variables continuas, pero están limitados a variables gaussianas relacionadas linealmente (Kenley 1986). La mayoría de los modelos ya establecidos suponen variables discretas.

Aprendizaje paramétrico con datos incompletos

Aparecen mayores dificultades cuando los datos de entrenamiento no están completos. En este sentido pueden plantearse dos tipos de información incompleta:

- *Valores faltantes*: Faltan algunos valores de uno o varias variables en algunos ejemplos.
- *Nodo oculto*: Faltan todos los valores de una variable.

El primer caso es más sencillo, y existen varias alternativas para tratarlos, entre ellas:

1. *Eliminar los ejemplos con valores ausentes.*
2. *Considerar un nuevo valor adicional para la variable: 'desconocido'.*
3. *Considerar el valor más probable a partir de los datos de la misma en las demás instancias.*
4. *Considerar el valor más probable en base a las demás variables (supone cierto estudio de correlación).*

Las dos primeras opciones son habituales en problemas de aprendizaje, y válidas siempre y cuando se cuente con un número elevado de datos completos. La tercera opción viene a ignorar las posibles dependencias de la variable con las demás, cuando ya se cuenta con la estructura que las describe en el grafo; no suele proporcionar los mejores resultados (Stuart y Norvig 1996). La cuarta técnica se sirve de la red ya conocida para inferir los valores desconocidos. Primero se rellenan las tablas de parámetros usando todos los ejemplos completos. Después, para cada instancia incompleta, se asignan los valores conocidos a las variables correspondientes en la red y se propaga su efecto para obtener las probabilidades *a posteriori* de las variables no observadas. Entonces se toma como valor observado el más probable y se actualizan todas las probabilidades del modelo antes de procesar la siguiente instancia incompleta.

La aparición de nodos ocultos requiere un tratamiento más complejo que no es objetivo de esta investigación abordar. Para un mejor estudio de este tema consultar (Stuart y Norvig 1996).

1.1.2 Propagación en Redes Bayesianas

Para los distintos sistemas de inferencia probabilística, el objetivo principal es el cálculo de la distribución de probabilidad posterior de un conjunto de variables de consulta, en base a determinadas variables de evidencia. En (Castillo et al. 1997), se hace referencia a distintos algoritmos para la *propagación de evidencias*.

La clasificación de estos algoritmos se basa en la forma en que se usa la red, o sea, si se trabaja directamente con la red obtenida: algoritmo de inversión de arcos (Schachter 1990) y algoritmo de eliminación de variables (Shenoy 1992) o con estructuras auxiliares para propiciar el paso de mensajes: propagación en *árboles de unión* (*junction trees*) (Pearl 1988), (Castillo et al. 1997), (Jensen 2001), (Schachter 1994), (Baldi y Soren 2001), (El-Hay 2001), (Jensen y Nielsen 2007),), (Kjærulff y Madsen 2008) y propagación perezosa (*lazy propagation*) (Shenoy 1992), (Madsen y Jensen 1999).

En el presente trabajo, se describen dos algoritmos de propagación exacta: el primero mediante *árboles de unión* (Pearl 1988), (Castillo et al. 1997), (Jensen 2001), (Schachter 1994), (Baldi y Soren 2001), (El-Hay 2001), (Jensen y Nielsen 2007),), (Kjærulff y Madsen 2008), (El-Hay 2001) y el segundo basado en la eliminación de variables (Shenoy 1992). El primer algoritmo se puede utilizar para propagación de evidencias en redes múltiplemente conexas, transformando la estructura inicial en un árbol de familias (ver anexo 1). Se escoge este algoritmo teniendo en cuenta que según (El-Hay 2001) el algoritmo de propagación en árboles de unión es uno de los más populares, de hecho es uno de los más utilizados en las herramientas de RB revisadas, por ejemplo *HUGIN*. El algoritmo de eliminación de variables se escoge basado en las experiencias de los miembros del proyecto *ELVIRA*[7], que manifiestan que es el algoritmo menos complejo para esta tarea.

[7] Curso de Modelos Gráficos impartido por profesor Manuel Gómez, Departamento de Ciencias de la Computación e IA, Universidad de Granada, en el doctorado curricular sobre *Soft computing*, desarrollado en la UCLV.

La propagación de evidencias, para el caso exacto, se realiza si se tiene en cuenta un conjunto de variables *evidenciales* $E \subseteq D$ con valor evidencial $E = e$ (*e representa uno de los posibles valores de la variable de evidencia*) y el conjunto de variables no evidenciales $D \mid E$ para las cuales se calculan las probabilidades condicionales $P(X_i \mid e)$. Una forma de calcular $P(X_i \mid e)$ es mediante la fórmula de Bayes, pero el método resulta ineficiente desde el punto de vista computacional debido al elevado número de combinaciones de valores que involucra. Si se tienen en cuenta las estructuras de independencias entre las variables se reduce el número de cálculos considerablemente.

1.1.2.1 Propagación en árboles de unión

El *método de agrupamiento*, inicialmente desarrollado por (Lauritzen y Spiegelhalter 1988), se basa en la construcción de subconjuntos de nodos (aglomerados, conglomerados o cliques) que capturan las estructuras locales del modelo probabilístico asociado al grafo. De esta forma, el proceso de propagación de evidencia puede ser acometido calculando probabilidades locales (que dependen de un número reducido de variables), evitando así calcular probabilidades globales (que dependen de todas las variables). Los *conglomerados* de un grafo son los subconjuntos que representan sus estructuras locales. A continuación, el algoritmo de agrupamiento calcula los conglomerados del grafo, luego obtiene las funciones de probabilidad condicionada de cada conglomerado, calculando de forma iterativa varias funciones de probabilidad locales. Por último, se obtiene la función de probabilidad condicionada de cada nodo *marginalizando* la función de probabilidad de cualquier conglomerado en el que esté contenido. Algunas modificaciones de este método utilizan una representación gráfica de la *cadena de conglomerados* para propagar la evidencia de forma más eficiente (Castillo et al. 1997).

En el presente trabajo se utiliza un algoritmo que realiza una modificación del algoritmo de agrupamiento, que se basa en el envío de *mensajes* en un árbol de unión construido a partir

de una cadena de conglomerados. De igual forma que el método de agrupamiento, este *método de propagación en árboles de unión* es válido para *redes de Markov*[8] y RB.

Factorización de la DPC por potencias

Sean C_1 C_2,..., C_m subconjuntos de un conjunto de variables $V = \{X_1, X_2..., X_N\}$. Si la *DPC* de $X_1, X_2..., X_N$ puede ser escrita como un producto de m funciones no negativas φ_i, (i = 1, 2, ..., m), esto es,

$$P(x_1,\ldots,x_N)=\prod_{i=1}^{m}\varphi_i(c_i) \tag{1.5}$$

donde c_i es la realización de C_i, las funciones φ_i se llaman factores potenciales o funciones potenciales de la DPC.

Para RB, la representación potencial se construye a través del grafo no dirigido obtenido moralizando[9] y triangulando (ver anexo 1) el grafo original. Esta representación potencial se obtiene asignando la función de probabilidad condicionada, $P(x_i \mid \varphi_i)$, de cada nodo X_i a la función potencial de un conglomerado que contenga a la familia del nodo (Castillo et al. 1997).

Por tanto, para describir el método de propagación mediante árboles de unión se supone que se tiene un grafo no dirigido triangulado con conglomerados $C = \{C_1, C_2\ldots, C_m\}$ y una representación potencial $\vartheta = \{\varphi_1(c_1),\ldots, \varphi_m(c_m)\}$. Este método utiliza un árbol de unión del grafo para propagar la evidencia.

Absorción de la evidencia

Si se modifican las funciones potenciales de los conglomerados que contienen nodos evidenciales en la forma siguiente: para cada conglomerado C_i que contenga algún nodo evidencial, la nueva función potencial $\varphi_i^*(c_i)$ se define como:

---

[8] Una red de Markov es un modelo de dependencia asociado a un grafo no dirigido, o sea un par (G, φ), donde G es un grafo no dirigido y φ es un conjunto de funciones potenciales definidas en los conglomerados asociados al grafo no dirigido G.

[9] Moralizar un grafo es la operación que consiste en adicionar aristas entre nodos que tienen un hijo común, esto es "casar" a los padres de cada nodo.

$$\varphi_i^*(c_i) = \begin{cases} 0, & \text{si algún valor en } c_i \text{ es inconsistente con e,} \\ \varphi_i(c_i), & \text{en otro caso} \end{cases} \tag{1.6}$$

Para el resto de los conglomerados no es necesario realizar ningún cambio. Luego se tiene:

$$P(x \mid e) \propto \prod_{i=1}^{m} \varphi_i^*(c_i) \tag{1.7}$$

El algoritmo de propagación exacta mediante la unión de árboles puede enunciarse de la siguiente forma:

Algoritmo de propagación exacta mediante árboles de unión

Entrada: Una RB (G, P) con variables aleatorias X y la distribución $X \mid \varphi_X$ asociada a cada nodo X y un conjunto de nodos de evidencia $E \subseteq G$ con valor evidencial $E = e$.

Salida: Dependencias de probabilidad condicional $P(X_i \mid e)$ para cada nodo no evidencial X_i.

Pasos iniciales:

1. *Obtener el árbol de familias para el GDA G. Sea* $C = \{C_1, C_2, \ldots, C_m\}$ *el conjunto de conglomerados resultantes.*
2. *Asignar cada nodo* X_i *en G a uno y sólo un conglomerado que contenga* X_i*. Sea* A_i *el conjunto de nodos asignados al conglomerado* C_i*.*
3. *Para cada conglomerado* C_i *en C definir:*

$$\varphi_i(C_i) = \prod_{x_i \in A_i} P(X_i \mid \pi_i) \tag{1.8}$$

Si $A_i = \varnothing$*, entonces definir* $\varphi_i(C_i) = 1$.

4. *Absorber la evidencia* $E = e$ *en las funciones potenciales* $\vartheta = \{\varphi_1(C_1), \varphi_1(C_2) \ldots, \varphi_m(C_m)\}$.

Pasos iterativos

5. *Para* $i = 1, 2, \ldots, m$ *hacer: Para cada vecino* B_j *del conglomerado* C_i*, si* C_i *ya recibió los mensajes de todos sus vecinos, calcular y enviar el mensaje* $M_{ij}(S_{ij})$ *a* B_j*, donde:*

$$M_{ij}(s_{ij}) = \sum_{c_i \setminus s_{ij}} \varphi_i(C_i) \prod_{k \neq j} M_{ki}(s_{ki}) \tag{1.9}$$

6. *Repetir el paso 5 hasta que no pueda ser calculado ningún nuevo mensaje.*

7. *Calcular la DPC de cada conglomerado C_i usando:*

$$P(c_i) = \varphi_i(C_i) \prod_{k} M_{ki}(s_{ik}) \tag{1.10}$$

Para cada nodo no evidencial X_j en la red calcular $P(X_j \mid e)$ usando:

$$P(X_j \mid e) = \sum_{c_k \setminus x_j} P(C_k) \tag{1.11}$$

donde C_k es el conglomerado menor que contiene a X_j.

1.1.2.2 Algoritmo de propagación mediante la eliminación de variables

Entrada: Una RB (*G, P*) en la que:

T: conjunto de factores potenciales iniciales. Inicialmente: distribuciones asociadas a las variables de la red,

X: variables de la red,

H: variables objetivo, aquellas sobre las que versa la consulta,

E: conjunto de variables evidenciales, variables observadas,

Y: variables a eliminar de la red: $Y = X \setminus \{H, E\}$ (siempre y cuando no estén d-separadas de *H* dado *E* ni sean sumideros), ver anexo 1 sobre conceptos básicos.

Salida: Dependencias de probabilidad condicional $P(X_i \mid e)$ para cada nodo no evidencial X_i.

A continuación se describen los pasos del algoritmo basado en la eliminación de variables.

1. Para cada variable $Z \in Y$

   - Sea φ_Z el conjunto de factores potenciales en *T* que contienen a la variable *Z*

- Sea ϕ_Z el potencial obtenido de la combinación de todos los potenciales en φ_Z
- Marginalizar ϕ_Z para eliminar Z : $\phi = \sum_Z \phi_Z$
- Actualizar el conjunto de potenciales: $T = (T - \phi_Z) \cap \phi$

2. Aquí solo quedarán potenciales definidos sobre las variables de interés H

1.1.3 Las Redes Bayesianas como clasificadores

Un problema de clasificación se define así: dado un conjunto de entrenamiento T con un conjunto de clases C, encontrar una descripción tal que se pueda encontrar la clase C_j de un objeto t_i sin hacer uso de T. Estrictamente, según (Aytug 2000) encontrar la función de membresía: $f: T \rightarrow C$ tal que la probabilidad $P(f(t_i) = c_j)$ sea aproximadamente 1, $t_i \varepsilon T$, $c_j \varepsilon C$.

Una RB puede ser utilizada como un clasificador en el caso particular en el que el nodo no evidenciado y a inferir es precisamente el que representa la variable clase o variable dependiente; en este caso se habla de un clasificador bayesiano. Los clasificadores bayesianos minimizan el costo del error en la clasificación usando la siguiente función:

$$\gamma(x) = \arg\min_k \sum_{c=1}^{ro} \cos t(k,c) p(c \mid x_1, x_2, ..., x_N) \qquad (1.12)$$

donde *cost* (*k, c*) denota el costo de una mala clasificación según cierto criterio. En el caso de una función para dos clases, el clasificador asigna la clase a posteriori más probable para una instancia dada, o sea:

$$\gamma(X) = \arg\max_C p(C \mid X_1, X_2, ..., X_N) = \arg\max_C p(C) p(X_1, X_2, ..., X_N \mid C) \qquad (1.13)$$

En dependencia de la forma en que se aproxime $p(X_1, X_2, ..., X_N \mid C)$ se obtienen diferentes clasificadores (Larrañaga et al. 2005).

Cuando las RB se usan como clasificadores, se está en presencia de problemas de clasificación supervisada, pues la clase forma parte del conjunto de entrenamiento, o sea se conoce para cada objeto o ejemplo, la clase a la que pertenece (Larrañaga 2000). En este caso el problema se formula del siguiente modo: Sea C_j la variable clase, y $\{X_1, X_2, ..., X_N\}$

vector de rasgos que describen cada caso; $p(C_j \mid \{X_1, X_2, \ldots, X_N\})$ probabilidad de que un objeto con las características $\{X_1, X_2, \ldots, X_N\}$ pertenezca a la clase C_j. Se trata de encontrar la clase C_j^* verificando que $p(C_j^* \mid \{X_1, X_2, \ldots, X_N\}) = max_j \; p(C_j \mid \{X_1, X_2, \ldots, X_N\})$.

Para el clasificador Naïve Bayes (CNB) (Duda y Hart 1973) la probabilidad de que el *i-ésimo* ejemplo pertenezca a la clase *j-ésima* puede calcularse aplicando el teorema de Bayes. El clasificador CNB no obtiene resultados favorables en los casos de dominios en los que las variables estén correlacionadas.

En (Friedman et al. 1997a) se presenta el método CNB aumentado a árbol, conocido en la literatura como algoritmo *TAN* (*Tree Augmented Naïve Bayes*), con este modelo se obtienen mejores resultados que con CNB, a la vez que mantiene la simplicidad computacional de éste. El modelo *TAN* es una red bayesiana en la que la variable a clasificar no tiene padres, mientras que el conjunto de variables padres de cada una de las variables predictoras, X_i, contiene necesariamente a la variable a clasificar, y a lo sumo otra variable. El algoritmo propuesto por (Friedman et al. 1997a) es una adaptación del algoritmo de (Chow y Liu 1968) y calcula la información mutua entre cada par de variables dada la clase. El mismo garantiza que la estructura obtenida tiene asociada la máxima verosimilitud entre todas las estructuras *TAN* posibles. El modelo *TAN* está limitado por el número de padres de las variables predictivas. En la página 275 del libro de Jensen se puede obtener la descripción detallada del mismo, (Jensen y Nielsen 2007). El *k Dependence Bayesian classifier* (*kDB*, *clasificador bayesiano con k dependencias)* (Sahami 1996) evita esta restricción pues una variable predictiva puede tener hasta *k* padres además de la clase. La complejidad de estos algoritmos es $O(N^2)$ para la arquitectura *TAN* y $O(N^2 (M.C.T^2 + K))$ para la arquitectura *kDB,* donde *N* es el número de variables del problema, *C* la cantidad de clases, *T* número máximo de valores descritos que un rasgo puede tener y *M* el número de casos de la base de ejemplos.

En (Pazani 1996) se presenta un modelo en el que se reduce el número de probabilidades a considerar, pues las variables no relevantes para el problema se consideran como en el modelo CNB y las condicionalmente dependientes dada la clase, se unen en una sola; para ello proponen usar dos algoritmos voraces, que se basan en la teoría de modelización hacia delante y hacia atrás (Larrañaga 2000).

1.1.3.1 Necesidad de la reduccion de atributos en algunos casos

Uno de los problemas de todo proceso de aprendizaje es escoger los atributos para describir los datos. Frecuentemente se dispone de más rasgos de los que son necesarios para aprender, y muchos algoritmos de aprendizaje tienen problemas cuando hay bastantes atributos irrelevantes. Por ello hacen falta técnicas que permitan reconocer atributos no necesarios. Existen dos aproximaciones para la selección de atributos:

- Transformación del conjunto de datos a un espacio de menor número de dimensiones (técnicas no supervisadas de reducción de dimensiones: Análisis de componentes principales (Dillon y Goldstein 1984), (Escofier y Pages 1992), (Lebart 1998), escalado multidimensional (Peña 2002), proyección aleatoria (Ruiz 2006).

- Obtención del subconjunto de atributos más adecuado para la predicción (técnicas supervisadas: técnicas de filtrado (*Filters*) o técnicas de envoltura (*Wrappers*), propuestas por (John et al. 1994).

  Técnicas de filtrado: Suponen que se tiene una medida de evaluación de cada atributo que permite obtener su relevancia respecto al objetivo, establecen un orden para los atributos midiendo su relevancia en la predicción de la clase separadamente (computacionalmente barato) y a partir del orden se decide cuantos atributos eliminar. Son ejemplos: Entropía (ID3), prueba Chi-cuadrado (por ejemplo CHAID), *relief* (creencia) (Larrañaga et al. 2003), (Lanzi 2006), (John et al. 1994).

  Técnicas de envoltura: Evalúan subconjuntos de atributos hasta encontrar el más adecuado (computacionalmente caro), para la evaluación utilizan un algoritmo de aprendizaje y no se puede hacer una búsqueda exhaustiva. Entre estas técnicas se encuentran *Hill-climbing* (ascensión de colinas), *simulated annealing* (recocido simulado), *best first (primero el mejor)*, algoritmos genéticos, etc. En general en estas técnicas hay dos estrategias: *Forward selection* (selección hacia delante), *backward elimination* (eliminación hacia atrás) (Larrañaga et al. 2003) (Lanzi 2006) (John et al. 1994), (Ruiz 2006).

Los problemas de bioinformática se caracterizan por un gran número de rasgos, por lo que en la mayoría de los casos se hace necesario la reducción de dimensionalidad (Saeys 2004).

Además, el alto costo en tiempo y recursos que han mostrado los algoritmos exactos de búsqueda, ha conllevado al auge y desarrollo de heurísticas y metaheurísticas cuyo uso ha arrojado resultados muy alentadores. Pueden citarse por ejemplo los algoritmos que usan heurísticas aleatorias bioinspiradas como método de búsqueda en la selección de rasgos, entre ellos, los algoritmos genéticos (Li et al. 2004), la optimización basada en enjambres de partículas (Kennedy 1997), (Kennedy y Eberhart 1995b),(Kennedy y Eberhart 1995a), (Kennedy y Spears 1998). y las colonias de hormigas (Dorigo y Stützle 2002), (Dorigo y Stützle 2004), (Dorigo et al. 2006), (Dorigo 2007).

Dentro de los algoritmos bioinspirados usados para la selección de rasgos, la Inteligencia de Enjambres (*Swarm Intelligence*, *SI*) ha sido objeto de estudio, investigación y de mucha aplicación por su simplicidad y robustez.

1.1.3.2 Optimización de enjambres de partículas

La metaheurística PSO, fue desarrollada por Kennedy y Eberhart (Kennedy y Eberhart 1995b), (Kennedy y Eberhart 1995a) y está inspirada en el comportamiento social observado en grupos de individuos tales como bandadas de pájaros, enjambres de insectos o bancos de peces. Un enjambre se define como una colección estructurada de organismos (agentes) que interactúan. La inteligencia no está en los individuos sino en la acción de todo el colectivo. Tal comportamiento social se basa en la transmisión del éxito de cada individuo a los demás del grupo, lo cual resulta en un proceso "sinergético" que permite a los individuos satisfacer de la mejor manera posible sus necesidades más inmediatas, tales como la localización de alimentos o de un lugar de cobijo. Cada organismo (partícula) se trata como un punto en un espacio *N* dimensional el cual ajusta su propio "vuelo" de acuerdo a su propia experiencia y la experiencia del resto de la banda. La banda (*swarm*) "vuela" por el espacio de búsqueda localizando regiones o partículas prometedoras (Kennedy y Spears 1998).

La metaheurística PSO ha mostrado ser muy eficiente para resolver problemas de optimización de un sólo objetivo con rápidas tasas de convergencia (Kennedy et al. 2001). Dado un espacio de decisión *N*-dimensional, cada partícula *i* del enjambre conoce su posición actual $X_i = \{X_{i1}, X_{i2}, ..., X_{iN}\}$, la velocidad $V_i = \{V_{i1}, V_{i2}, ..., V_{iN}\}$ con la cual ha llegado a dicha posición, así como la mejor posición $X_{iBest} = \{X_{iBest1}, X_{iBest2}, ..., X_{iBestN}\}$ en la

que se ha encontrado, denominada "mejor personal". Además, todas las partículas conocen la mejor posición encontrada dentro del enjambre X_{gBest} denominada "mejor global".

Si se supone el uso de la información proveniente del mejor global, en cada iteración *t* del algoritmo PSO, cada componente *j* de la velocidad y la posición de cada partícula *i* del enjambre se actualiza conforme a:

$$V_i = wV_i + c_1\, rand\left(X_{iBest_i} - X_i\right) + c_2\, Rand\left(X_{gBest} - X_i\right) \tag{1.14}$$

donde *w* es el parámetro de inercia que regula el impacto de las velocidades anteriores en la nueva velocidad de la partícula. Típicamente a *w* se asigna un valor fijo, por ejemplo 0.8 y en otros casos se le asigna un valor inicial entre 1 y 1.5 que se hace decrecer durante la ejecución del algoritmo o funciones que garantizan esto, como por ejemplo *w* = 0.5 + *rand*()/2. O sea, se proponen pesos de inercia altos al principio y se reducen con el número de iteraciones. Si *w*=0 la velocidad de la partícula se determina por las mejores posiciones ya sea su mejor posición o la mejor posición global alcanzada por todas las partículas.

Sí *w* toma un valor grande, significa que las partículas deben cambiar su velocidad instantáneamente y moverse lejos de la mejor posición según su conocimiento, o sea se favorece la exploración global (*global search*), si *w* es pequeño, la razón en la cual la partícula debe cambiar su velocidad es baja, es decir la inercia sugiere continuar el camino original, aún cuando se conozca el mejor estado (*fitness*), se favorece la exploración local (*local search*).

El valor c_1 es el parámetro cognitivo que indica la influencia máxima de la mejor experiencia individual de la partícula en su nueva velocidad y c_2 es el parámetro social que indica la influencia máxima de la información social en el nuevo valor de velocidad de la partícula; mientras que, *rand*() y *Rand*() son funciones que retornan un número aleatorio en el intervalo [0, 1], mediante el cual se determina la influencia real de las informaciones individual y social en la nueva velocidad para la partícula.

La selección de estos parámetros tiene impacto en la velocidad de convergencia y la velocidad del algoritmo para encontrar el óptimo. En (Chávez et al. 2007c), se tomaron por ejemplo los valores $c_1 = c_2 = 2$, pero en realidad se recomienda en el trabajo que c_1 y c_2 no

tomen necesariamente el mismo valor sino, que se generen aleatoriamente con distribución uniforme en el intervalo [0, 2]. En (Beielstein et al. 2002) se recomienda que la suma de estos valores sea menor o igual a 4. El trabajo de Beielstein et al. resulta interesante pues hace un análisis de los parámetros del algoritmo *PSO* mediante técnicas de diseños experimentales (Mahamed et al. 2005). Para obtener una mayor información acerca de la influencia de estos parámetros en la efectividad del algoritmo *PSO* ver (Beielstein et al. 2002; Kennedy et al. 2001; Shi y Eberhart 1998).

En el presente trabajo se utiliza PSO binario como algoritmo de búsqueda de la estructura simplificada, por selección de rasgos, de una RB.

1.1.3.3 Evaluación de las Redes Bayesianas como clasificadores

Cuando las RB se utilizan como un modelo clasificador, se hace necesario evaluar su desempeño, al igual que se realiza la evaluación en cualquier problema de clasificación supervisada. Para ello se utilizan criterios[10] tales como: porciento de clasificaciones correctas, diferentes medidas del error, el índice de Kappa (Brender et al. 1994), medida *F* (Van Rijsbergen 1979), y funcionales de calidad y error (Ruiz-Shulcloper 2000), (Donald et al. 1994). La capacidad del modelo para representar confiablemente el sistema real, se relaciona esencialmente con la precisión (Daalen 1992), no existe un modelo clasificador mejor que otro de manera general; para cada problema nuevo es necesario determinar con cuál se pueden obtener mejores resultados, y es por esto que han surgido varias medidas para evaluar la clasificación y comparar los modelos empleados para un problema determinado. Las medidas más conocidas para evaluar la clasificación están basadas en la matriz de confusión que se obtiene cuando se prueba el clasificador en el conjunto de datos del entrenamiento o en un conjunto de datos externos que no intervienen en el aprendizaje.

En la Tabla 1.1 se muestra la matriz de confusión de un problema de dos clases, donde *Pos/pos* es la clase positiva y *Neg/neg* la clase negativa.

[10] Indistintamente se utilizan los términos criterio o medida para hacer referencia a los aspectos cuantitativos o cualitativos a considerar en la evaluación.

Tabla 1.1. Matriz de confusión de un problema de dos clases

Matriz de Confusión		*Clase verdadera*	
		Pos	*Neg*
Clase Predicha	*pos*	*VP*	*FP*
	neg	*FN*	*VN*
Total columna		*P*	*N*

En la Tabla 1.1 las siglas *VP* y *VN* representan los elementos bien clasificados de la clase positiva y negativa respectivamente y *FP* y *FN* identifican los elementos negativos y positivos mal clasificados respectivamente. Basados en estas medidas, se calcula el error, la exactitud, la razón de *VP* (*rVP*) o sensibilidad, la razón de *FP* (*rFP*), la precisión y la especificidad, el coeficiente de correlación de *Matthews* (*mcc*), que se muestran en las expresiones de la Tabla 1.2.

Tabla 1.2. Medidas de evaluación estándar

Nombre	***Medida***
Exactitud	$\frac{VP+VN}{P+N}$
rVP o sensibilidad	$\frac{VP}{P}$
rVN o especificidad	$\frac{VN}{N}$
rFP	$\frac{FP}{N}$
rFN	$\frac{FN}{P}$
Precisión	$\frac{VP}{VP+FP}$
Medida F	$\frac{2}{\frac{1}{precision}+\frac{1}{sensibilidad}}$
Correlación de *Matthews* (mcc)	$\frac{VP*VN-FP*FN}{\sqrt{(VP+FN)(VN+FP)(VP+FP)(VN+FN)}}$

Otra forma de evaluar el rendimiento de un clasificador es por las curvas ROC (*Receiver Operator Curve, Curva de operación del receptor*) (Fawcett 2004). En esta curva se representa el valor de razón de VP contra la razón de FP, mediante la variación del umbral de decisión. Se denomina umbral de decisión a aquel que decide si una instancia *x*, a partir del vector de salida del clasificador, pertenece o no a cada una de las clases. Usualmente, en el caso de dos clases se toma como umbral por defecto 0.5; pero esto no es siempre lo más conveniente. Se usa el área bajo esta curva, denominada AUC (*Area Under the Curve, área bajo la curva* ROC) como un indicador de la calidad del clasificador. En tanto dicha área esté más cercana a la unidad, el comportamiento del clasificador está más cercano al clasificador perfecto (aquel que lograría 100% de VP con un 0% de FP).

En (Larrañaga et al. 2005) se hace una comparación de diferentes paradigmas de clasificación supervisada en Bioinformática: bayesianos, estadísticos, inductivos y de IA. Resulta interesante el uso de las curvas ROC para la comparación, así como análisis de la razón de error basado en la matriz de confusión (Fawcett 2004). Además se describe como estimar la razón de error cuando se usa el modelo para clasificar nuevas instancias. En (Friedman y Goldszmidt 1996) se exponen en detalle los clasificadores bayesianos explicados anteriormente y se hace una comparación de ellos con los criterios de medida descritos y utilizando bases de datos disponibles en el sitio *Web* de la Universidad de California Irvine: UCIML para el aprendizaje automático (Asuncion y Newman 2007).

1.1.4 Productos de *software* para Redes Bayesianas

Múltiples productos de *software* se han creado para el uso de RB. Los primeros resultaron costosos, pues fueron el resultado de grandes proyectos de investigación como por ejemplo: *Netica*, *Elvira*[11], *Hugin* (Madsen et al. 2005). A *Netica* y *Hugin* se puede acceder gratuitamente a una versión demostrativa de los mismos pero como se consideran *software* propietario o comercial, esta versiones de prueba, no permiten utilizar todas sus capacidades (Ver anexos 2 y 3). Por ejemplo en el caso de *Hugin* se puede trabajar en este

[11] *Elvira* es fruto de un proyecto de investigación, en el que participan investigadores de varias universidades españolas y de otros centros. Está destinado a la edición y evaluación de *modelos gráficos probabilistas*, concretamente RB y diagramas de influencia. Está escrito y compilado en Java, lo cual permite que funcione en diferentes plataformas y sistemas operativos (MS-DOS/Windows, Linux, Solaris, etc.)

modo demo hasta con 20 variables solamente, lo que resulta evidentemente insuficiente para resolver problemas de análisis de secuencias aunque fuese con el objetivo de comparar resultados con otras herramientas. El *software* *Elvira* tiene características diferentes, de hecho se han realizado programas que permiten hacer la conversión de datos entre *Weka* y *Elvira.* La tendencia en la actualidad es la de realizar herramientas con código libre (*open source*) y hacer extensiones al mismo con los algoritmos que se proponen.

En (Murphy 2005) se hace una comparación detallada de 46 productos de *software* de RB, la que se resume en el apéndice 2. En (KDnuggets 2008) se hace una descripción de otros productos discretizados en *software* propietario y *software* libre, ver resumen en anexo 3. En el trabajo de (Doldán 2007) se comentan los productos de *software* del anexo 2 y otras direcciones de Internet con herramientas sobre RB. En (Charles River Analytics 2004) se describe el *software BNet*: familia de herramientas para construir y usar redes de creencia (se utiliza como ayuda en la predicción y visualización del pronóstico del tiempo). Incluye dos productos:

- *BNet.Builder*: Para crear RB, entrar información y obtener resultados.
- *BNet.EngineKit*: Para incluir la tecnología de las RB en nuestras aplicaciones.

En el contexto de esta investigación se utiliza el *software* libre *Weka* al que se hace extensiones para probar los algoritmos que se presentan. Se utiliza además una versión del *Weka* que permite una cierta paralelización en un cluster de computadoras real, como el que existe en la UCLV o un cluster virtual, todo lo cual ayuda a reducir la complejidad computacional en tiempo de procesamiento.

1.2 Aplicaciones de las Redes Bayesianas en Bioinformática

Como se ha argumentado, las RB constituyen un formalismo muy atractivo de representación del conocimiento con incertidumbre, resultado de la sinergia entre métodos probabilísticos-estadísticos de análisis de datos y técnicas de IA. Ellas se han aplicado con éxito en muy diversos campos, para modelar la incertidumbre en sistemas expertos, para resolver problemas de clasificación, predicción, inferencia, sistemas de toma de decisiones, entre otros. La Bioinformática no se exceptúa como campo de aplicación. Siempre que surge la necesidad de extraer información desde datos, en presencia de incertidumbre, datos

ruidosos o sujetos a errores, los métodos bayesianos son ampliamente utilizados por las ventajas que ofrece sobre las técnicas estadísticas convencionales (Jeroen et al. 2008), (Silva y Muñoz 2000).

También se ha comentado que el desarrollo alcanzado por las Ciencias Biológicas ha permitido la acumulación de mucha información experimental disponible en grandes bases de datos. La secuenciación del ADN (Consortium 2004), (Benson et al. 2005), produjo un crecimiento exponencial de las descripciones lineales de proteínas y moléculas de ADN y ARN (Ácido ribonucleico) y planteó los problemas informáticos de interés biológico: el almacenamiento y manejo eficiente de la información y la extracción de información útil para en última instancia, comprender las relaciones entre los genes, las proteínas, la funcionabilidad, la vida y la salud. La Bioinformática constituye el campo de conocimientos multidisciplinario entre la biología, la informática y la matemática que debe abordar este problema. En ella surge en particular, la necesidad de desarrollar nuevos algoritmos para el tratamiento de problemas de análisis de secuencias.

1.2.1 Estudio de secuencias genómicas

Los algoritmos de aprendizaje automático son ideales para dominios caracterizados por la presencia de gran cantidad de datos, patrones ruidosos y la ausencia de teorías generales determinísticas. La idea fundamental de estos algoritmos es aprender automáticamente la "teoría" a partir de los datos, a través de un proceso de inferencia o inducción, modelación o aprendizaje desde ejemplos, aunque la inducción sea incompleta, y por tanto condicionada a una probabilidad, según criterios bayesianos.

En (Larrañaga et al. 2005) se describen los principales dominios biológicos donde son necesarias las técnicas de aprendizaje automático. En dicho documento se hace una división en seis dominios fundamentales: genómica, proteómica, micro-arreglos (antes citados como matrices de ADN o *micro arrays*), sistemas biológicos, evolución y minería de texto. El resto de las aplicaciones se agrupan en "otras". Todos estos dominios tienen problemas en los que se hace necesario el estudio de secuencias biológicas. La genómica es considerada uno de los dominios más importantes pues como se ha descrito anteriormente, la cantidad de secuencias identificadas se incrementa notablemente en esta época. El análisis de

secuencias genómicas persigue fundamentalmente la búsqueda de genes, y de sus regiones regulatorias. De igual modo, el dominio de la proteómica resulta de interés en la actualidad. De hecho en la presente tesis se analizan dos problemas fundamentales, uno en el campo de la genómica: localización de sitios de *splicing* o corte de intrones, y otro en el dominio de la proteómica para la predicción de interacciones de proteínas.

En Internet se cuenta con herramientas para el análisis de secuencias, algunas de las que se describen en el anexo 4, extraído del libro: (Gibas y Per 2001). Además en el artículo de (Gilbert 2004) se hace una descripción de los principales productos de *software* de Bioinformática libres en Internet. Este último documento es además, de acceso libre.

1.2.2 Problemas bioinformáticos que se resuelven mediante Redes Bayesianas

Las RB son valiosas siempre que sea necesario extraer información desde datos sujetos a incertidumbre, subjetividad, cualquier tipo de error o ruidosos. Por tanto, no resulta ninguna sorpresa que las RB se apliquen ampliamente en la actualidad a los campos de la genética, la genómica, sistemas biológicos, etc. donde este tipo de datos complejos es una norma. En el trabajo se mencionan sólo algunos ejemplos, pues la literatura en este campo también crece notablemente con el número de aplicaciones que se realizan (Liu y Logvinenco 2003), (Wilkinson 2007).

En (Pe'er et al. 2001) se presenta una RB de interacciones entre genes (interacciones de causalidad, mediación, activación, e inhibición). El método se aplica a expresión de datos de mutantes de levadura (*Saccharomyces Cerevisiae*) y se descubren una variedad de estructuras metabólicas, señales y caminos regulatorios. En (Friedman 2004) se discute otro problema aplicado a la bioinformática usando un modelo probabilístico.

Las interacciones entre proteínas son importantes para muchos procesos biológicos, identificarlas resulta vital para comprender la maquinaria de la célula. Las RB han sido ampliamente utilizadas con este objetivo; en (Wu et al. 2006) se hace uso de esta teoría para redes de interacciones de proteínas en hongos utilizando solamente anotaciones de genes ontólogos (*Gene Ontology*, GO). El nivel más alto de confianza obtenido para la clasificación de verdaderas interacciones es de un 78 %. En (Jansen et al. 2003) se realizó una aplicación similar utilizando otros rasgos desde datos genómicos. Resultados en

arabidopsis se pueden ver en (Cui et al. 2007). Otras investigaciones de interacciones de proteínas se describen en los trabajos (Long et al. 2005), (Lu et al. 2005) y (Qi et al. 2006). En humanos hay resultados muy interesantes con RB en (Scott y Barton 2007). En (Asthana et al. 2007) se hace uso de redes probabilistas para predicción de interacciones de proteínas utilizando, para la propagación, algoritmos previamente usados para redes de comunicación y en (Troyanskaya et al. 2003) se usan las RB para predicción de función de genes desde distintas fuentes de datos en la levadura (*Saccharomyces cerevisiae).*

Otra aplicación bien interesante en este campo es la localización de genes en un genoma completo, o en una larga secuencia genómica, lo cual fue considerado durante varios años como el problema principal de la Bioinformática. Contribuye de manera importante a su solución, la identificación de sitios de *splicing*, que separan zonas codificantes y no codificantes. Este es un buen ejemplo de un problema abierto en Bioinformática (Saeys 2004).

El hecho de que el genoma de determinada especie esté completamente o casi completamente secuenciado significa apenas que se conoce la secuencia de bases de ADN que lo conforman, pero ello está lejos de implicar que se sabe el rol de todas sus partes, incluso la localización de subsecuencias donde aparece o puede aparecer un gen, y mucho menos su funcionalidad. En países como Estados Unidos de América, se da la situación extrema, por demás sin ningún tipo de ética, que se patenta la información apenas aproximada de una subsecuencia que probablemente “contiene un gen".

La localización *in sílico* de los genes se aborda desde varios puntos de vista. Se conoce en primer lugar que todas las secuencias que representan un gen comienzan con un codón de inicio y finalizan con uno de los tres codones de terminación, pero la presencia de tales codones no siempre indica el inicio y el final del gen. Si a ello se une la posible existencia de hasta seis marcos diferentes de lectura[12], así como la presencia de zonas amplias no

[12] En una secuencia de ADN, las tripletas codificantes (codones) pueden estar alternadas e incluso mezcladas con secuencias no codificantes. Por tanto, al leer una secuencia de codones, aparecen tres marcos de lectura. Si además, se tiene en cuenta que pueden aparecer producto de la doble hélice en sentido contrario, se habla de 6 marcos posibles de lectura.

codificantes y usualmente más largas que los genes mismos, se comprende la dificultad del problema.

Se ha intentado abordar la localización de los genes a través de otras subsecuencias que están relacionadas con la estructura primaria de los mismos o su expresión, en particular, *promotors*, *motifs* o los sitios de *splicing* (*splice sites*). Se ha abordado el problema desde diversas técnicas de clasificación de Estadística y de IA.

En general se han logrado buenos resultados, pero la supremacía de estas combinaciones en lugares que no son verdaderos *splice sites* hace que, aunque los por cientos de clasificación sean buenos, se comete un gran error en la predicción de falsos negativos. Otros autores han centrado sus esfuerzos precisamente en reducir los falsos negativos en la clasificación, y han logrado muy buenos resultados si se hace una buena selección de rasgos (Saeys 2004)

Con esta mejora se logra superar los índices de clasificación sin dañar el rendimiento del sistema, recuérdese además que una pequeña cantidad de parámetros permite evitar problemas como el sobre ajuste u *overfitting (Cai et al. 2000)*.

1.3 Consideraciones finales del capítulo

En el presente capítulo se presentaron los fundamentos teóricos relacionados con el concepto de RB, se analizaron los problemas actuales de estos sistemas concernientes con el aprendizaje de la estructura y los parámetros de las mismas.

Se mostró la posible utilización de las RB como modelos de clasificación y más generalmente, para la inferencia tanto de variables predictoras como de la variable objetivo o dependiente. Además se plantearon ejemplos de aplicación de estos modelos en el campo de la Bioinformática y se mostró un resumen de los productos de *software* más utilizados para el trabajo con RB.

A partir del análisis realizado, se plantea la necesidad de buscar nuevos algoritmos de búsqueda de la topología de RB desde datos, implementar en plataformas de *software* libre los modelos propuestos para ser usados por la comunidad científica sin restricciones y aplicar esta teoría en el análisis de secuencias genómicas y en dominios biomédicos.

En el próximo capítulo se proponen tres métodos para la primera tarea.

2. NUEVOS ALGORITMOS DE APRENDIZAJE ESTRUCTURAL DE REDES BAYESIANAS

Como se ha mencionado, la definición de una RB supone siempre dos tareas. La primera es caracterizar la estructura de dependencias entre las variables predictoras y la segunda es la "determinación" de la distribución de probabilidades (parámetros) que permitirá hacer inferencias. Ambas tareas son muy importantes, pero la primera es esencial por ser la más complicada y además por ser imprescindible para poder realizar la segunda (Friedman et al. 1997b).

Así, las posibilidades del uso de las RB se amplían si es posible realizar el aprendizaje de las mejores estructuras y parámetros. Ello es especialmente útil si se logra mejorar el aprendizaje estructural acorde con el dominio del campo de aplicación, en este caso la Bioinformática, en el análisis de regiones genómicas codificantes para proteínas, o en un domino de diagnóstico médico.

Los enfoques propuestos hasta el momento para la primera tarea: (Neapolitan 1990),(Heckerman 1997), (Acid y De Campos 2003), (Acid et al. 2005), (Bouckaert 2007), (Kjærulff y Madsen 2008), demuestran insatisfacción aún con las soluciones.

A continuación se proponen tres nuevos algoritmos que han sido publicados en detalles en diversas formas y resumidos en (Chávez et al. 2008d). Dos de ellos están basados en pruebas de independencia según el estadístico Chi-cuadrado y el tercero está basado en medidas de ajuste y búsqueda.

2.1 Aprendizaje Estructural de Redes Bayesianas basado en técnicas estadísticas

Los dos algoritmos que se describen en los siguientes epígrafes permiten obtener la estructura de una RB desde datos. En ambos casos se utiliza la prueba Chi-cuadrado (ver prueba Chi-cuadrado en anexo 5) para buscar las variables más significativamente relacionadas con la variable dependiente (Silva 1997).

En el primer algoritmo se construyen árboles de decisión basados en la técnica CHAID (ver algoritmo de obtención de árboles de decisión CHAID en el anexo 5); el segundo parte de esta misma idea, pero se obtienen dependencias entre las variables, no mediante los árboles de decisión completos, sino que se busca selectivamente a lo ancho y en profundidad en el árbol de todas las interacciones posibles.

Técnicas estadísticas usadas en el manejo de información en Redes Bayesianas

Por su propia semántica, las RB se basan en la teoría de las probabilidades, lo que las hace un formalismo fuerte. Ellas procuran buscar una versión simplificada de la DPC (Ver anexo 1 sobre conceptos básicos) de un conjunto de variables supuestamente relacionadas, a diferencia de otros métodos estadísticos clásicos que se condicionan *a priori* o *posteriori* de una distribución de probabilidad, tal es el caso, por ejemplo, del análisis discriminante y la regresión.

Los métodos estadísticos tradicionales no resultan del todo adecuados para el análisis y modelado de datos de sistemas biológicos. Ocurre con frecuencia que una similitud o una significación estadística no se corresponde necesariamente con una similitud o significación biológica (entendida la significación como probabilidad de la diferencia). Se requieren nuevos modelos, así como la integración de diferentes paradigmas para abordar los problemas actuales.

Se han desarrollado modelos de RB multinomiales y *gaussianas* (Castillo et al. 1997). En el trabajo se usan los primeros, propios de variables discretas, lo que exige que si se trata de un dominio con variables continuas sería imprescindible discretizar estas variables previamente con la posible pérdida de información. Aunque los problemas de pérdida de información por discretización son discutibles, particularmente en el área biológica donde los números aparentemente reales tienen realmente un carácter ordinal. Con independencia de este problema, las RB con variables discretas son ampliamente utilizadas en la actualidad, lo cual se ha descrito a grandes rasgos en la presente tesis.

2.1.1 Aprendizaje Estructural de Redes Bayesianas basado en árboles de decisión obtenidos con el algoritmo CHAID

En este epígrafe se describe el algoritmo ByNet, el cual obtiene árboles de decisión al estilo CHAID (Ver algoritmo de obtención de árboles de decisión CHAID en el anexo 5).

El usuario es quién decide cuántos árboles obtener. Una vez que se construye el primer árbol se descarta el conjunto de variables que forman parte de él, lo cual contribuye a la reducción de variables. Este algoritmo tiene la limitante de que puede obtenerse un modelo asimétrico.

En una RB un modelo asimétrico significa que un nodo en la red tiene muchos padres. Ello trae consigo que se necesita cubrir todas las combinaciones de valores entre la variable asociada al nodo y las variables de los nodos que apuntan a ella. Esto no siempre se logra, incluso en dominios bioinformáticos caracterizados por grandes volúmenes de datos.

2.1.1.1 Fundamentos generales del Algoritmo

Para obtener la topología de la RB se aplica la técnica CHAID, más que segmentar la población; en este caso se usa para:

- Conocer cuáles, entre decenas de variables pueden ser eliminadas.
- Comprender el orden de importancia de los rasgos desde el punto de vista estadístico.
    - en las investigaciones epidemiológicas: para comprender el orden de los factores de riesgo en la caracterización de una enfermedad
    - en estudios de secuencias: conocer las posiciones más importantes para el análisis que se hace
- Entender cómo interactúan los rasgos unos con otros
    - en las investigaciones epidemiológicas: para entender cómo ciertos factores de riesgo se relacionan con otros
    - en estudios de secuencias: saber cómo interactúan estas posiciones

El CHAID permite obtener un árbol en forma automática con las características mencionadas.

Los parámetros que utiliza el algoritmo ByNet para acotar la búsqueda y obtener determinada estructura de RB se describen a continuación:

- Cota máxima de la probabilidad del estadístico Chi-cuadrado que será aceptado por el método como una posible interacción. El dominio de dicha probabilidad es el intervalo de números reales entre cero y uno. A medida que este valor se aproxima a cero, el algoritmo es más exigente para declarar una interacción y consecuentemente, se observará una disminución en la cantidad de interacciones aceptadas por el método y disminuirá la cantidad de arcos y de nodos en la red. De esta forma, no sólo constituye un método de aprendizaje automatizado sino que además obtiene una selección de atributos (*ChiSquareMaxSignificance*). En caso de que el valor observado sea inferior a 0.05, las variables seleccionadas para formar parte de la red serán estadísticamente significativas.

- Mínima cantidad de casos que debe tener una población para que el método considere su posible subdivisión. Esta es una cota necesaria para lograr cierto nivel de fiabilidad ante los test Chi-cuadrado. Debe tratarse de acuerdo al tamaño de la población, la distribución de la clase y los posibles grados de libertad de las tablas de contingencias aspirantes (*MinCountOfInstancesToSplit*).

- Cota sobre la máxima cantidad de arcos que pudiese tener el camino más largo dentro de la topología generada. Su mayor influencia se sienta en la complejidad de la red a obtener. Tiene amplia repercusión en el espacio de búsqueda cuando las aplicaciones tienen muchos casos y rasgos (*MaxDepth*).

  Se sugiere que los niveles de profundidad en los árboles sean pequeños, a lo sumo tres. Esto está relacionado con que las relaciones entre variables a gran distancia en la red no son fuertes, y tienden a complejizar la estructura de la misma.

- Cantidad máxima de árboles de decisión que deben obtenerse, la variable clase depende de cada uno de los nodos origen de dichos árboles (*m_nMaxNrOfTrees*). Se sugiere que el mayor valor por defecto sea a lo sumo 10. Las pruebas que se han realizado han mostrado que este valor se considera suficientemente grande, sobre todo en el caso que las variables predictivas puedan tomar varios valores. El algoritmo puede llevar a

estructuras de red con una distribución asimétrica para la variable clase; pues como se mencionó con anterioridad, a pesar de que las bases de datos tengan muchas instancias, no siempre se tiene un cubrimiento del dominio para todas las combinaciones que se presentan entre las variables que son raíz de los distintos árboles y la variable clase.

La estructura de RB se obtiene si se establece un enlace desde los nodos que representan a las variables más significativas en los árboles creados hasta el nodo asociado a la variable dependiente o clase, esto significa que se establece un arco dirigido de cada variable más significativa en cada uno de los árboles hacia la variable clase. El algoritmo no descarta la participación de los expertos pues posibilita obtener distintos árboles si se cambian parámetros o se hace interactivamente con el usuario, lo que permite que se tenga en cuenta la valoración del especialista en el campo de aplicación a la hora de seleccionar la topología más adecuada.

Pudiera pensarse que la ventaja de experiencia anterior es poco aplicable en problemas de Bioinformática, donde se pretende casi siempre descubrir conocimiento. Sin embargo, la valoración del especialista en bioinformática no solo se basa en la experiencia anterior adquirida en su rama, sino, además, en los resultados obtenidos por otras áreas de las ciencias biológicas como la paleontología, la filogenia y la genómica comparativa, etc., por solo citar algunas. Además, se debe tener presente que en ocasiones una significación estadística no necesariamente coincide con una significación biológica, y se hace necesario introducir en el modelo, variables con significado biológico, el cual puede ser inferido a partir del conocimiento establecido en cualquiera de las áreas de investigación antes mencionadas.

Como es posible obtener más de un árbol usando este método, el número de árboles a generar se considera un parámetro y para evitar ciclos se eliminan sucesivamente las variables incluidas (Chávez et al. 1999), (Chávez et al. 2005), (Grau et al. 2006), (Chávez et al. 2007a), (Grau et al. 2007b).

El aporte del algoritmo *ByNet* está en la forma en que se construye la RB a partir de árboles de decisión que se crean mediante la técnica CHAID, estos árboles obtienen las relaciones

fundamentales entre las variables desde el punto de vista estadístico (Chávez et al. 2008a), (Grau et al. 2007b).

2.1.1.2 Algoritmo ByNet

A continuación se describe el Algoritmo ByNet que implementa la elaboración de la estructura de la red utilizando los árboles de decisión.

Entrada: Un conjunto de variables $\{X_1, \ldots, X_N, C\}$

Salida: Una estructura de RB G

Paso 1. Inicializar:

- La red G como un grafo vacío
- *ListaVariablesSign* = []; // Lista que almacena las variables significativas (Chi-cuadrado)
- *ListaVariablesPosibles* = $[X_1, X_2, \ldots, X_N]$;
- *m_nMaxNrOfTrees* es seleccionado por el usuario (Por defecto se toma valor 10).

Paso 2. Para i =1 hasta *N* hacer

Prob [*i*] = *ChiCuadrado* [X_i ; *C*]

Sí Prob [*i*] < 0.05 entonces *Adicionar* (X_i , *ListaVariablesSign*)

Paso 3. *Ordenar* (*ListaVariablesSign*)

Paso 4. Mientras (*ListaVariablesSign* $\neq \phi$) y (*m_nMaxNrOfTrees* ≥ 0) hacer

raíz = *Primero* (*ListaVariablesSign*);

a = TREECHAID (*raíz* , *ListaVariablesPosibles*);

m_nMaxNrOfTrees = m_nMaxNrOfTrees-1;

Borrar las variables que forman el árbol almacenado en *a* de *ListaVariablesSign*

Borrar las variables que forman el árbol almacenado en *a* de *ListaVariablesPosibles*

Paso 5. *Retornar* (Red G)

En el *Paso* 1 de inicialización, G representa la estructura de RB, en *ListaVariablesSign* se almacenan las variables predictivas que resultan significativas acorde a la prueba Chi-cuadrado y *ListaVariablesPosibles* almacena todas las variables predictivas del problema.

En el *Paso* 2 se llama a una función, a la que denominamos *ChiCuadrado* que calcula la significación estadística mediante la prueba Chi-cuadrado, entre las variables predictivas y la variable clase.

En el *Paso* 3 se ordenan ascendentemente las variables predictivas acorde al valor de probabilidad obtenido en el paso dos.

En el *Paso* 4 se selecciona la primera variable en la lista. TREECHAID es una función que obtiene los árboles al estilo de la técnica CHAID (como se describió en el Anexo 5), pero el nodo raíz se pasa como parámetro a la función, además de la lista de variables posibles. En el *Paso* 5 se devuelve la RB en G.

Entre el *Paso* 3 y el *Paso* 4, es posible dar la posibilidad de interactuar con especialistas del dominio de aplicación, de modo que no se tengan en cuenta sólo la dependencia estadística entre las variables, sino que se puedan introducir en el modelo en otro orden. Esto significa que se puede forzar el orden de entrada de las variables, teniendo en cuenta la opinión del experto.

2.1.1.3 Algunas consideraciones sobre el algoritmo ByNet

El algoritmo ByNet parte de la construcción sucesiva de árboles de decisión utilizando la técnica CHAID. El primer árbol obtenido romperá por la variable más significativa acorde al test Chi-cuadrado. Las siguientes variables que formen parte de ese árbol estarán en niveles inferiores, lo que significa que su posición dentro de la red estará más alejada de la clase. Como que se construye un árbol para cada variable que esté significativamente asociada a la clase y que no haya estado presente en árboles anteriores, la variable por la que rompe el último de ellos, puede tener menor importancia que otras ya incorporadas a otros árboles, y sin embargo, ella tendrá una dependencia directa de la clase.

Esta forma de proceder se realiza para evitar ciclos, pero de hecho trae consigo que el algoritmo ByNet no siempre ofrezca buenos resultados cuando la correlación entre las variables predictivas es muy elevada. En ese caso, los primeros árboles de decisión

recogerán la información más importante contenida en los datos, mientras que los últimos ofrecerán una información mucho menor. La red conformada con la unión de todos los árboles no hace distinción entre unos y otros.

Por otra parte, originalmente la elección de cuántos árboles de decisión crear, es o bien puramente estadística, basada en la significación del Chi-cuadrado o se puede hacer que pertenezca por entero al usuario. Pudiera pensarse en el empleo de alguna heurística que lo ayudara a tomar una decisión en este sentido, pero ello hace más complejo el proceso de obtención de la estructura de la red.

Basados en las limitaciones aquí mencionadas, surgió la idea del siguiente algoritmo, también para el aprendizaje estructural en RB.

2.1.2 Aprendizaje Estructural de Redes Bayesianas basado en el algoritmo CHAID

El algoritmo BayesChaid se basa, como su nombre lo indica, en ideas de la técnica de CHAID con adaptaciones. Estas consisten esencialmente en hacer la búsqueda de las dependencias entre las variables no mediante los árboles de decisión completos, sino que busca a lo ancho y en profundidad en el árbol de interacciones posibles (Chávez et al. 2008b).

2.1.2.1 Fundamentos generales del Algoritmo

Para realizar la búsqueda de la estructura de la red, ésta se acota por un conjunto de parámetros que impone el usuario y que fueron explicados previamente en el epígrafe 2.1.1, pues algunos coinciden con los del algoritmo ByNet.

Para comprender la idea del algoritmo BayesChaid, se decidió utilizar en su cuerpo una función booleana "**Terminar**" que, dadas dos variables predictivas que se pasan como parámetros, devuelve "verdadero" en caso de que se cumpla una de las condiciones de terminación. Ellas son:

- Mínima cantidad de casos que debe tener una población para que el método considere su posible subdivisión (*MinCountOfInstancesToSplit*). Condición que se usa en el algoritmo ByNet, pues es una cota necesaria para lograr cierto nivel de fiabilidad ante los test Chi-cuadrado.

- Cota sobre la máxima cantidad de arcos que pudiese tener el camino más largo dentro de la topología generada (*MaxDepth*). En el algoritmo ByNet se usa como niveles de los árboles que se obtienen, y en el algoritmo BayesChaid se utiliza para evitar caminos largos. Esto influye en la complejidad de los algoritmos de propagación.
- En este algoritmo se incluye un nuevo parámetro, *MaxNrOfParents*, que es la cantidad de padres que podrán tener los nodos de la red a generar. Esto influye de forma especial en las tablas de probabilidades condicionadas generadas para la red.

El algoritmo por pasos se describe a continuación.

2.1.2.2 Algoritmo BayesChaid

Entrada: Un conjunto de variables $\{X_1, \ldots, X_N, C\}$

Salida: Una estructura de RB G

Paso 1. Inicializar:

- La red G como un grafo vacío
- *ListaVariablesSign* = []; // Lista que almacena las variables significativas (Chi-cuadrado)
- *ListaVariablesPosibles* = $[X_1, X_2, \ldots, X_N]$;

Paso 2. Para i =1 hasta N hacer

Prob $[i]$ = *ChiCuadrado* $[X_i ; C]$

Sí Prob $[i] < 0.05$ entonces *Adicionar* (X_i , *ListaVariablesSign*)

Paso 3. Para cada $X_i \in$ *ListaVariablesSign* hacer

- Crear un enlace desde la variable C hasta X_i

Paso 4. *Ordenar (ListaVariablesSign)*

Paso 5. Para i =1 hasta $|ListaVariablesSign|-1$ hacer

Para j =i+1 hasta $|ListaVariablesSign|$ hacer

- Si No (*Terminar* (X_i, X_j)) entonces
  - *Pr* $[i, j]$ = *ChiCuadrado* $[X_i ; X_j]$

- *Sí (Pr [i, j] < 0.05) entonces*

  Crear un enlace desde X_i a X_j

Paso 6. *Retornar* (Red G)

En el *Paso* 1 de inicialización, al igual que en el algoritmo explicado anteriormente, G representa la estructura de RB; en *ListaVariablesSign* se almacenan las variables predictivas que resultan significativas acorde a la prueba Chi-cuadadro y *ListaVariablesPosibles* almacena todas las variables predictivas del problema.

En el Paso 2 se llama a una función, a la que denominamos *ChiCuadrado* que calcula la significación estadística mediante la prueba Chi-cuadrado, entre las variables predictivas y la variable clase.

En el *Paso* 3 se crea un enlace desde el nodo correspondiente a la variable clase a cada uno de los nodos asociados a las variables significativas en *ListaVariablesSign*.

En el *Paso* 4 se ordena ascendentemente la *ListaVariablesSign* según la significación estadística.

En el *Paso* 5 |*ListaVariablesSign*| representa la cardinalidad de la lista de variables significativas y *Terminar* (X_i, X_j) es una función que, como se explicó con anterioridad, chequea las condiciones de terminación del algoritmo para X_i y X_j. En el *Paso* 6 se devuelve la RB en G.

Entre el *Paso* 4, es posible cambiar el orden de las variables según criterio de expertos, lo que permite introducir en el modelo de RB variables con significado biológico.

2.1.2.3 Algunas consideraciones sobre el algoritmo BayesChaid

El algoritmo BayesChaid se basa también en el criterio de la prueba Chi-cuadrado para obtener la estructura de la red. Debido a que no construye árboles de decisión, este método no se ve afectado por la presencia de variables predictivas altamente correlacionadas. En su primer paso funciona de manera similar al algoritmo Naïve Bayes, pero con un proceso incluido de selección de rasgos (según el criterio del Chi-cuadrado). De esta forma se garantiza que las variables más relacionadas con la clase, se encuentren en relación directa

con ella. El algoritmo es capaz de "modelar" además, las relaciones existentes entre otras variables predictivas.

Obsérvese que ni en este caso ni en el anterior se tienen en cuenta las limitaciones de la prueba Chi-cuadrado, ver anexo 5. Ello no constituye un problema, pues la prueba estadística en este caso se utiliza sólo como criterio de selección de un nodo en la RB.

2.2 Aprendizaje Estructural de Redes Bayesianas basado en técnicas de Inteligencia Artificial

En el capítulo I se ha explicado que el aprendizaje de la estructura de una red bayesiana se convierte en un problema de optimización combinatoria, consistente en la búsqueda de la mejor red de todas las posibles en un espacio en el que intervienen *N* atributos para identificar los objetos del dominio de aplicación. Para el caso de redes múltiplemente conexas se trata de un problema de tipo *NP* (Cooper 1990), (Chickering 1996). Debido a esto, es que surge la necesidad de utilizar algoritmos que hacen uso de heurísticas para facilitar la búsqueda de la RB que representen satisfactoriamente el problema.

2.2.1 Técnicas de IA usadas en el manejo de información en Redes Bayesianas

La IA ha jugado un papel importante como fuente inagotable de técnicas, métodos, modelos y algoritmos tanto para el análisis de datos biológicos como para el modelado y simulación de sistemas biológicos. Técnicas tales como redes neuronales artificiales, algoritmos evolutivos, autómatas celulares, RB y modelos ocultos de Markov, resultan ser enfoques ideales para dominios que se caracterizan por una explosión de datos y muy poca teoría, como es el caso de la Bioinformática.

En la actualidad los modelos bioinspirados se muestran eficientes en la solución de problemas prácticos, y en particular se pretende utilizar la técnica PSO en la búsqueda de la estructura de una RB. Este método muestra similaridades con otras técnicas de la computación evolutiva, como los AG (Davis 1991), pero no usa operadores de mutación y cruce, y tiene pocos parámetros a ajustar por lo que resulta más fácil de implementar (Beielstein et al. 2002), (Mahamed et al. 2005).

Una alternativa de los AG son los algoritmos de estimación de distribuciones (*Estimation of distribution algorithms*, EDAs), utilizados en Cuba fundamentalmente por (Ochoa et al. 2000), (Ochoa et al. 2003), (Caballero 2007) y (Piñero 2005) y un resumen de la aplicación en Bioinformática en (Armañanzas et al. 2008). En la tesis de (Caballero 2007) se muestran algunas deficiencias relacionadas con este método, y en la tesis de (Piñero 2005) se utilizan los EDAs en la optimización de reglas borrosas.

Aprendizaje estructural de Redes Bayesianas con PSO, combinación con la reducción de atributos

PSO ha sido exitosamente utilizado en la resolución de problemas de optimización con variables continuas. En este caso se selecciona el algoritmo propuesto por (Wang et al. 2006) de selección de atributos para usarlo en el aprendizaje de RB, pero considerando que se trata de un PSO binario (Correa et al. 2007). En (Chávez et al. 2007c) se muestra que si se realiza una selección de atributos previa, con el propio algoritmo PSO propuesto por (Wang et al. 2006), el aprendizaje de RB tiene mejor eficiencia que las que se obtienen con todos los rasgos de la base de casos, específicamente en problemas con muchas variables (digamos, más de 100).

2.2.2 Fundamentos generales del Algoritmo

Como se ha explicado anteriormente, la búsqueda de la estructura de la red puede formularse como un problema de optimización en el espacio de las posibles redes Ω, en otras palabras, determinar $X_{ópt} \in \Omega$, $H(X_{ópt}) \geq H(X_i), \ \forall \ X_i \in \Omega$.

La función objetivo *H* es una métrica de calidad de las descritas en el capítulo cuatro de la tesis de Bouckaert (Bouckaert 1995) para el caso de búsqueda local se utiliza cualquiera de las métricas implementadas en Weka, o medidas que miden la exactitud en el caso de una búsqueda global cuando se trabajan con validaciones cruzadas de los datos según implementaciones en el ambiente *Weka* (Witten y Frank 2005) u otras implementadas como parte de este trabajo.

Entre las métricas de calidad local ya implementadas en Weka están:

a. La métrica de entropía según expresión

$$H(Bs, D) = -N\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}\frac{N_{ijk}}{N}\log\frac{N_{ijk}}{Nij} \qquad (2.1)$$

donde

D: base de datos

Bs: estructura de red

r_i: cardinalidad del rasgo x_i, $i = 1, ..., n$

q_i: cardinalidad del conjunto de padres de x_i en *Bs,* se puede calcular como el producto de las cardinalidades de los nodos padres de x_i

N_{ij}: número de casos en D para los que los padres de x_i toman su *j-ésimo* valor, $j = 1, ..., q_i$

N_{ijk}: número de casos en D para los que los padres de x_i toman su *j-ésimo* valor, y para los casos que x_i toma su *k-ésimo valor,* $k = 1, ..., r_i$

El número de parámetros de la métrica es: $K = \sum_{i=1}^{n}(r_i - 1).q_i$

b. La métrica AIC (*Akaike Information Criterion*, Criterio de Información de Akaike), según la expresión

$$Q_{AIC}\ (B_S, D) = H(B_S, D) + K \qquad (2.2)$$

c. La métrica MDL (*Minimum Description Length*, longitud de descripción mínima), según expresión

$$Q_{AIC}\ (B_S, D) = H(B_S, D) + \frac{K}{2}\log N \qquad (2.3)$$

d. La métrica Bayesiana según

$$Q_{Bayes}(B_S, D) = P(B_S)\prod_{i=0}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij}+N_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(N'_{ij}+N_{ijk})}{\Gamma(N'_{ijk})} \qquad (2.4)$$

donde

$P(B_S)$ es una red a priori, en la implementación de Weka no se tiene en cuenta.

$\Gamma(.)$ es la función gamma

N'_{ij} y N'_{ijk} representan selección de cantidades a priori, limitadas por: $N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk}$

Si $N'_{ij}=1$, o lo que es lo mismo, $N'_{ij}=r_i$ se obtiene la métrica K2.

e. La métrica K2

$$Q_{K2}(B_S, D) = P(B_S)\prod_{i=0}^{n}\prod_{j=1}^{q_i}\frac{(r_i-1)!}{(r_i-1+N_{ij})!}\prod_{k=1}^{r_i}N_{ijk}! \qquad (2.5)$$

Si $N_{ijk} = 1/r_i.q_i$, o lo que es lo mismo, $N'_{ij} = 1/q_i$ se obtiene la métrica BDe

Es posible utilizar otras medidas propuestas en la literatura en el caso de conjuntos de datos con clases desbalanceadas como *fitness*, (Beielstein et al. 2002), (Ye 2003), (Eitrich et al. 2007), (Guo y Viktor 2007), las que se han añadido como métricas de calidad global en el trabajo.

Las métricas de calidad global que se han implementado, se han utilizado frecuentemente para bases de datos desbalanceadas, se basan en los resultados del modelo clasificador, es por ello que en su mayoría se utilizan medidas para evaluar clasificadores en problemas de clasificación supervisada, a partir de la matriz de confusión que se obtiene cuando se prueba el clasificador en un conjunto de datos que no intervienen en el entrenamiento.

Entre las métricas de calidad global que se han implementado en Weka como parte de este trabajo se tienen:

a. La métrica G-*means*, según expresión

$$g = \sqrt{sensibilidad * especificidad} \qquad (2.6)$$

b. La métrica de la sensibilidad relativa se representa en la expresión

$$RS = \frac{sensibilidad}{especificidad} \qquad (2.7)$$

c. La métrica GM en la expresión

$$GM = \sqrt{rVP + rVN} \qquad (2.8)$$

d. El coeficiente de correlación de *Matthews* en la expresión

$$mcc = \frac{VP * VN - FP * FN}{\sqrt{(VP + FN)(VN + FP)(VP + FP)(VN + FN)}} \tag{2.9}$$

e. Otra métrica de efectividad se muestra en la expresión

$$\xi_\beta = \frac{1 - (\beta^2 + 1) * precisión * sensibilidad}{\beta^2 * precisión + sensibilidad} \tag{2.10}$$

En la expresión 2.10 si $\beta = 1$ se obtiene la media armónica entre sensibilidad y precisión, si $\beta = 0$, $\xi_\beta = 1 - precisión$ y si $\beta = \infty$, $\xi_\beta = 1 - sensibilidad$.

En la modelación del problema de búsqueda a partir del algoritmo PSO se define cada partícula como una RB, la cual se representa como una matriz de adyacencia *B*. Se puede pensar que el espacio de búsqueda Ω tiene cardinalidad 2^{n^2}; de hecho se puede trabajar con dicho espacio, pero habría que chequear que no existan ciclos. La eliminación de ciclos se puede lograr, por ejemplo, eliminando de forma aleatoria arcos que formen parte de ciclos existentes (Chávez et al. 2007c).

Se conoce que un grafo dirigido representa un ordenamiento topológico, si y solo sí este no presenta ciclos, es posible a partir de una permutación formar un grafo acíclico dirigido. Partiendo de esto en (Chávez et al. 2007c), (Chávez et al. 2008c), (Chávez et al. 2008a) se propone una forma de generar el espacio de búsqueda garantizando que no existan ciclos.

Para comprender la idea del algoritmo BayesPSO, se decidió utilizar en su cuerpo una función booleana “Terminación” que devuelve “verdadero” en caso de que se cumpla una de las condiciones de terminación. Ellas son:

- Cantidad de generaciones que van a interactuar las partículas o iteraciones del algoritmo (*CantGeneraciones*),
- El algoritmo puede terminar si en dos iteraciones sucesivas no se mejora el resultado de la métrica de calidad que evalúa la RB.

2.2.3 Algoritmo BayesPSO

El algoritmo que se propone es PSO binario, por lo que el algoritmo PSO original (Wang et al. 2006), (Ferat et al. 2007) se adapta, de modo que la actualización de las partículas se realiza como propone (Correa et al. 2007).

Entrada: Un conjunto de variables $\{X_1, \ldots, X_N, C\}$

Salida: Una estructura de RB G

Paso 1. *Inicializar*

- X_i es una partícula, en este caso una RB representada como una matriz de adyacencia (matriz del espacio Ω),
- $\{X_1, X_2, \ldots, X_N\}$ es una *b*andada (conjunto de partículas),
- $\{V_1, V_2, \ldots\}$ son *v*elocidades (matrices del espacio Ω asociadas a cada partícula que indican su movimiento),
- $\{X_{pBest1}, X_{pBest2}, \ldots, X_{pBestN}\}$ son los mejores puntos del espacio localizados por cada partícula,
- X_{gBest} es el mejor punto localizado por la bandada.

Paso 2. *Repetir*

Paso 3. *Actualizar la velocidad.* Las velocidades de todas las partículas *(*1,2,…, *cantPart*) se actualizan acorde a la expresión (1.14) que se detalló en el epígrafe 1.1.3.2 del capítulo I. y que repetimos en 2.11.

$$V_i = wV_i + c_1\, rand\left(X_{iBest_i} - X_i\right) + c_2\, Rand\left(X_{gBest} - X_i\right) \tag{2.11}$$

Según la expresión (2.12) la velocidad se convierte al intervalo [0, 1].

$$S\left(V_i\right) = \frac{1}{1 + e^{-V_i}} \tag{2.12}$$

Paso 4. *Actualizar la posición.* Para lograr actualizar las partículas se debe añadir la velocidad a cada partícula según expresión (2.13).

$$X_i(t+1) = X_i(t) + V_i \tag{2.13}$$

Paso 5. *Actualizar la memoria*. Actualizar X_{pBest} *y* X_{gBest} acorde a los resultados de la función objetivo o *fitness* en dicha iteración y el objetivo que se persigue ya sea minimizar o maximizar la métrica de calidad empleada.

Paso 6. *Hasta Terminación* ()

Paso 7. *Retornar* (X_{gBest} en G)

En el *Paso* 1 se asigna aleatoriamente valores a la población de X_i (la generación de las redes acíclicas se logra a partir de una permutación aleatoria π de (1, 2,…, *n*) con distribución uniforme, dicha red se representa como una matriz de adyacencia), de esta forma se propone en (Chávez et al. 2007c), V_i y X_{pBesti} de cada partícula se inicializan como copia de X_i y X_{gBest} se inicializa con la mejor partícula de acuerdo a la métrica.

En el *Paso* 3, *cantPart* es la cantidad de partículas que van a existir en cada generación.

El algoritmo se repite del *Paso* 2 al *Paso* 6 mientras no se cumpla la condición de terminación.

En el *Paso* 5 se puede seleccionar como función *fitness* una de las implementadas en Weka, u otras que se le han añadido al mismo como se explicó previamente.

2.2.4 Algunas consideraciones sobre el algoritmo BayesPSO

El algoritmo BayesPSO difiere de manera notable con respecto a los otros dos. Es más complejo desde el punto de vista espacial, pues parte de estructuras de redes seleccionadas al azar y después de un proceso iterativo, se llega a la estructura final de la red.

El algoritmo garantiza una buena solución. Se señala como desventaja que es necesario mantener almacenadas en cada paso, las matrices triangulares superiores relacionadas con cada partícula (RB), la matriz que almacena la mejor red obtenida hasta el momento por cada una de ellas, y además la correspondiente a la mejor RB global, que es la candidata a ser la solución final en cada paso y que será la solución definitiva en el último paso.

No obstante a estas deficiencias, cabe señalar que, con los ejemplos que se han corrido y que se mostrarán en epígrafes posteriores de esta tesis, no han existido nunca dificultades

de memoria. Más importante es el análisis de la complejidad computacional desde el punto de vista temporal que se hará en el siguiente epígrafe.

2.3 Análisis del comportamiento de los algoritmos

El análisis teórico de los algoritmos que se proponen para el aprendizaje estructural de RB se hace mediante el orden de complejidad temporal de cada uno, y a su vez se utiliza un fichero de datos extraído del *UCIML* (Asuncion y Newman 2007) para el estudio de los resultados que muestran dichos algoritmos si se utiliza la RB como clasificador junto a otros clasificadores. Además, se hace una comparación con otras 18 bases de la *UCIML*.

2.3.1 Análisis de la complejidad temporal

Para realizar el análisis de la complejidad temporal debemos tener en cuenta las siguientes variables:

$A \rightarrow$ número de árboles a crear en el algoritmo ByNet,

$M \rightarrow$ número de casos en la base de datos[13],

$N \rightarrow$ número de atributos del problema,

$T \rightarrow$ cantidad máxima de valores diferentes que pueden tomar los atributos,

$K \rightarrow$ número máximo de padres por nodos de la RB,

$N' \rightarrow$ variables significativas según la prueba Chi-cuadrado y que forman la *ListaVariablesSign*,

$V \rightarrow$ niveles o profundidad en los árboles para el algoritmo ByNet o camino mas largo en la RB y que relacionamos con la variable *MaxDepth*,

$P \rightarrow$ cantidad de partículas en el algoritmo BayesPSO,

$I \rightarrow$ cantidad de generaciones o iteraciones en el algoritmo BayesPSO, y

$C \rightarrow$ número de clases diferentes.

[13] Es equivalente a número de instancias o tamaño máximo de la población.

2.3.1.1 Análisis de la Complejidad temporal del Algoritmo ByNet

El análisis de la complejidad temporal se hace sobre la base de la descripción por pasos del algoritmo descrito previamente en el epígrafe 2.1.1.2:

La complejidad temporal del *Paso* 1 de inicialización es un O (*N*), pues el tamaño de la lista de variables posibles coincide con la cantidad de atributos del problema.

La complejidad temporal del *Paso* 2 es un O(*M* * *N*) pues hay que recorrer todos los atributos y la función que determina la probabilidad del estadístico Chi-cuadrado tiene complejidad O(*M*). El *Paso* 3 que ordena la lista de variables significativas tiene orden de complejidad temporal O(*N'LogN*'), pues *N'* es la cantidad de elementos de esta lista.

En el *Paso* 4 se entra en un ciclo donde se construyen los árboles; se sabe que la complejidad para construir los árboles es a lo sumo O($N * T^2$) (Quinlan 1986), (Quinlan 1993).

En el peor caso se construye un árbol completo en el cual cada camino en el árbol prueba cada variable de la lista de variables significativas, se asumen N' atributos y T valores diferentes que pueden tomar los atributos.

En cada nivel V, en el árbol, se debe examinar los T-V valores que quedan para cada atributo en ese nivel para calcular la probabilidad aplicando Chi-cuadrado, de donde se obtiene:

$$\sum_{V=1}^{T} V * N' = O(N' * T^2) \tag{2.14}$$

Si se llegan a construir todos los árboles, resulta finalmente una complejidad temporal máxima de: O($A * M * N' * T^2$), el cual puede ser aún menor, si se hace el análisis como (Ruiz 2006) que reporta un orden de complejidad temporal medio para árboles: O($M*N*log_2M$).

2.3.1.2 Análisis de la Complejidad temporal del Algoritmo BayesChaid

El análisis de la complejidad temporal del algoritmo BayesChaid se hace sobre la base de la descripción por pasos del algoritmo descrito previamente en el epígrafe 2.1.2.2.

La complejidad temporal del *Paso* 1 de inicialización es un O (*N*), pues el tamaño de la lista de variables posibles coincide con la cantidad de atributos del problema.

La complejidad temporal del *Paso* 2 es O (*M* * *N*) pues hay que recorrer todos los atributos y determinar la probabilidad del estadístico Chi-cuadrado.

Para el *Paso* 3 la complejidad es un O (*N´*).

En el *Paso* 4 se ordena la lista de variables significativas cuyo orden de complejidad temporal es O (*N'LogN*') donde *N'* es la cantidad de elementos de esta lista.

En el *Paso* 5 se evalúa la función que determina la probabilidad del estadístico Chi-cuadrado dentro de dos ciclos anidados, obteniéndose un O ($N'^2 * M$), que coincide con el orden de complejidad temporal del algoritmo BayesChaid.

2.3.1.3 Análisis de la Complejidad temporal del Algoritmo BayesPSO

Para el análisis de la complejidad temporal del algoritmo BayesPSO, descrito en el epígrafe 2.2.3 se debe conocer el orden de la métrica que mide la calidad de la red que se obtiene. La complejidad temporal es un O ($I * P * max(N^2, O(métrica))$).

La complejidad de las métricas se reportan en el capítulo cuatro de la tesis de Bouckaert (Bouckaert 1995), en el peor caso es un O(*M.K.T*).

Hasta aquí se demuestra que los tres algoritmos tienen una complejidad temporal polinomial si se hace una selección adecuada de los parámetros.

2.3.1.4 Comparación de los algoritmos

Si se toma el valor que pueden tomar los parámetros que se utilizan para realizar el análisis de complejidad temporal de los algoritmos, es posible hacer una comparación entre los tres algoritmos descritos en este capítulo, cuyo orden de complejidad temporal se muestra en la Tabla 2.1.

Si se parte de la definición de RB hay algunos parámetros a los que se le asignan valores suficientemente pequeños, de modo que la RB que se obtiene sea lo menos compleja posible, sin descuidar la calidad de los resultados que ofrece la misma.

Por ejemplo, en el algoritmo ByNet el parámetro que mide la cantidad de árboles a construir es importante, como ya se explicó previamente, pues para valores pequeños se logra evitar obtener modelos asimétricos, este parámetros está estrechamente relacionado con el número de padres en los algoritmos BayesChaid y BayesPSO.

Los valores que pueden tomar las variables resultan también significativos en el análisis de la complejidad.

Si se pretende establecer un orden en cuanto a la complejidad temporal de los algoritmos, el de menor complejidad resulta ByNet, le sigue BayesChaid y por último BayesPSO.

Tabla 2.1. Resultado del análisis de la complejidad temporal de los algoritmos propuestos en el trabajo.

Algoritmo	***Complejidad Temporal***
ByNet	$O(A * M * N * T^2)$
BayesChaid	$O(N'^2 * M)$
BayesPSO	$O(I * P * max(N^2, O(métrica)))$

2.3.2 Ejemplo de aplicaciones para validar los resultados

Para mostrar los resultados de los algoritmos se seleccionó originalmente una base de datos de las donadas en el *UCIML* (Asuncion y Newman 2007): la base de datos para reconocimiento de *Promoters* en secuencias de la *E. Colic*. La base de datos se crea por (Harley y Reynolds 1987). Esta base de datos se ha utilizado en la evaluación de algoritmos de aprendizaje automatizado (Towell et al. 1990).

Está formada por 106 casos y 58 atributos (incluida la clase), de ellos 57 rasgos predictores se corresponden con posiciones de pares de bases de secuencias nucleotídicas, representadas por A, G, T y C (ver anexo 1) y la variable clase por (*positivos* o *negativos*).

Se utilizaron los algoritmos que se proponen en la tesis, y según se aprecia en la Tabla 2.2 los tres algoritmos muestran resultados similares, el algoritmo BayesPSO hace mejor clasificación de los casos positivos con razón 0.906. Atendiendo al área bajo la curva ROC el mejor resultado es para el algoritmo BayesChaid con valor 0.942, pero si se utiliza el coeficiente de correlación de *Matthews* el mayor valor lo obtiene el algoritmo ByNet con valor 0.7.

Tabla 2.2. Resultados de la evaluación realizada con los algoritmos propuestos en el trabajo, en la base de datos para reconocimiento de *Promoters* en secuencias de la *E. Colic*.

Algoritmo	***Exactitud***	***rVP***	***rVN***	***Área bajo la curva ROC***	***mcc***
ByNet	84.90	0.887	0.811	0.905	**0.700**
BayesChaid	84.90	0.849	0.849	**0.942**	0.698
BayesPSO	83.96	**0.906**	0.774	0.896	0.685

Se hicieron además pruebas con 18 bases de datos del repositorio *UCIML* (Asuncion y Newman 2007).

En el anexo 6 se puede ver las características de las bases de datos utilizadas. Estas son diversas, 11 de ellas tienen rasgos discretos, 3 tienen rasgos continuos y 4 presentan combinación de ellos; 10 de ellas tienen 2 clases y el resto tiene, entre 3 y 19. La cantidad de casos varía también en las bases de datos, desde bases con 24 casos hasta bases con 3196 casos. De esta manera podemos ver la comparación para bases internacionales de propósito general y en particular el comportamiento de estos métodos ante bases bioinformáticas.

En las Tablas 2.3 y 2.4 se muestran los resultados de la exactitud y área bajo la curva *ROC*, para cada uno de los algoritmos propuestos en el trabajo, descritos en los epígrafes 2.1-2.3 de este capítulo.

La comparación en cuanto a exactitud con las 18 bases de datos de la UCIML se hace entre los algoritmos propuestos: ByNet, BayesChaid y BayesPSO y tres clasificadores bayesianos: RB K2, RB TAN y CBN.

Se seleccionó la prueba no paramétrica de Friedman (Siegel 1987) para analizar si hay diferencias significativas entre los resultados obtenidos por los métodos utilizados.

Como se observa en la Tabla 2.5, la significación en cuanto a exactitud es 0.199, mayor que 0.05, por lo que no se evidencian diferencias significativas entre los clasificadores.

Tabla 2.3. Resultados de la ejecución con bases de datos de la UCIML para exactitud

Bases		***ByNet***	***BayesChaid***	***BayesPSO***	**RB K2**	**RB TAN**	**CBN**
1	promoters	84.90	84.90	83.96	83.96	82.08	90.57
2	mammographic	81.89	83.14	82.62	82.41	81.27	83.25
3	Lung-cancer	78.13	75.00	78.13	71.88	65.63	75.00
4	hepatitis	85.16	86.45	83.87	84.52	85.16	85.16
5	e colic	67.26	85.12	84.52	85.12	84.82	85.42
6	crx	85.49	86.36	85.92	86.21	85.63	78.23
7	breast-cancer-w	90.78	97.51	97.36	97.36	95.31	97.36
8	contac-lenses	87.50	83.33	83.33	70.83	66.67	70.83
9	hayes-roth-m	57.58	81.06	74.24	72.73	67.42	80.30
10	kr-vs-kp	75.47	93.77	93.77	88.30	92.24	88.33
11	Monk 1	72.58	70.97	99.19	79.03	95.97	77.42
12	vote	94.33	91.67	92.67	91.33	93.67	89.67
13	Balance-scale	63.52	92.16	93.92	92.16	92.96	92.16
14	tic-tac-toe	70.25	72.96	72.65	76.62	76.83	69.62
15	iris	95.33	94.00	95.33	94.00	94.67	94.00
16	labor	84.21	87.72	89.47	91.22	89.47	89.47
17	segment-challenge	62.13	91.80	94.80	95.73	95.73	92.60
18	soybean	68.08	93.11	89.16	94.58	94.58	92.97

Tabla 2.4. Resultados de ejecución de los algoritmos con bases UCIML para el área bajo la curva ROC.

Bases		***ByNet***	***BayesChaid***	***BayesPSO***	**RB K2**	**RB TAN**	**CBN**
1	promoters	0.905	0.942	0.896	0.910	0.909	**0.967**
2	mammographic	0.831	0.898	**0.902**	0.898	0.899	0.901
3	Lung-cancer	**0.785**	0.778	0.768	0.725	0.725	0.768
4	hepatitis	0.751	0.886	0.870	0.910	0.894	**0.912**
5	e colic	0.693	**0.932**	0.926	0.930	0.920	0.930
6	crx	0.888	0.917	**0.928**	0.926	0.919	0.894
7	breast-cancer-w	0.971	0.992	0.992	0.993	0.984	0.994
8	contac-lenses	0.884	0.882	0.882	**0.900**	0.894	**0.900**
9	hayes-roth-m	0.648	**0.959**	0.932	0.934	0.916	0.948
10	kr-vs-kp	0.836	**0.984**	0.964	0.954	0.982	0.954
11	Monk 1	0.683	0.752	**1.000**	0.874	0.998	0.788
12	vote	0.952	**0.982**	**0.982**	0.980	0.984	0.967
13	Balance-scale	0.657	0.924	0.946	0.924	**0.968**	0.924
14	tic-tac-toe	0.731	0.783	0.790	0.831	**0.822**	0.744
15	iris	0.967	0.987	0.985	0.984	**0.988**	**0.988**
16	labor	0.964	**0.968**	0.951	0.966	0.949	0.965
17	segment-challenge	0.910	0.992	**0.997**	0.996	**0.997**	0.993
18	soybean	0.958	**0.998**	0.995	**0.998**	**0.998**	**0.998**

Tabla 2.5. Resultados de la prueba de Friedman si se comparan los algoritmos en cuanto a la exactitud en las 18 bases de datos de la UCIML.

N			18
Chi-cuadrado			7.392
gl			5
Sig. Asintót.			0.193
Sig. Monte Carlo	Sig.		0.199
	Intervalo de Confianza de 99%	Límite inferior	0.188
		Límite superior	0.209

En la Tabla 2.6 se observan los rangos promedios de esta prueba para la exactitud, el mejor resultado se obtiene con el algoritmo BayesChaid con rango promedio 4.06 y le sigue en rango promedio BayesPSO con valor 3.94.

Tabla 2.6. Rangos promedios obtenidos con prueba de Friedman para la exactitud.

Algoritmo	***Rango promedio***
ByNet	2.58
BayesChaid	4.06
BayesPSO	3.94
RB K2	3.58
RB TAN	3.44
CBN	3.39

Para los valores *de las áreas bajo la curva ROC* se verifican diferencias entre los métodos empleados. Para analizar cuáles son los métodos que difieren, se realizan pruebas no paramétricas de Wilcoxon (Siegel 1987).

En la Tabla 2.7 se muestran los resultados de la prueba de Friedman en la comparación entre los tres algoritmos propuestos y los tres clasificadores bayesianos escogidos para la comparación según el área bajo la curva ROC. Se aprecia que la significación es de 0.00 valor menor que 0.05, por lo que se rechaza la hipótesis de que el comportamiento de los clasificadores en similar para esta medida de validación (área bajo la curva ROC).

Para ver entre que clasificadores hay diferencias, se aplica la prueba de rangos de Wilcoxon (Siegel 1987). Estos resultados se muestran en la Tabla 2.8.

Tabla 2.7. Resultados de la prueba de Friedman si se comparan los algoritmos en cuanto al área bajo la curva ROC en las 18 bases de datos de la UCIML.

N			18
Chi-cuadrado			21.777
gl			5
Sig. Asintót.			.001
Sig. Monte Carlo	Sig.		0.00
	Intervalo de Confianza de 99%	Límite inferior	0.00
		Límite superior	0.001

Las pruebas de Wilcoxon evidencian que el único algoritmo que tiene diferencias significativas con los demás es ByNet en cuanto a las áreas bajos las curvas ROC

Tabla 2.8 Resultados de la prueba de rangos de Wilcoxon basada en las áreas bajo la curva ROC en las 18 bases de datos de la UCIML entre los seis clasificadores.

Algoritmos	***Z***	***Sig. asintót. bilateral***
BayesChaid - ByNet	-2.069	0.039
BayesPSO - ByNet	-2.534	0.011
RB K2 - *ByNet*	-2.201	0.028
RB TAN - *ByNet*	-1.870	0.062
CBN - *ByNet*	-2.01	0.044
BayesPSO - BayeChaid	-.682	0.496
K2 - *BayeChaid*	-.057	0.955
RB TAN - *BayeChaid*	-.071	0.943
CBN - *BayeChaid*	-.362	0.717
RB K2 - *BayesPSO*	-.305	0.760
RB TAN - *BayesPSO*	-.213	0.831
CBN - *BayesPSO*	-.142	0.887

Los resultados de la comparación evidencian que los métodos BayesChaid y BayesPSO muestran resultados en cuanto a eficiencia mejor a *K2*, *TAN* y *Naïve Bayes*, pero ByNet difiere del resto cuando se evalúan los resultados con las áreas bajo las curvas ROC.

El algoritmo ByNet no ofrece buenos resultados como clasificador en todas las aplicaciones, pero tiene valor teórico ya que permite obtener sub-grupos de relaciones entre posiciones distantes en secuencias genómicas para inferir en cualquier posición de la misma.

2.4 Conclusiones parciales del capítulo

En este capítulo se proponen tres nuevos algoritmos para la obtención de la estructura de la RB, dos basados en criterios estadísticos (prueba de independencia Chi-cuadrado) y uno basado en medidas de ajuste y búsqueda.

Se demuestra que la complejidad temporal de los tres algoritmos es polinomial. Se logra establecer un orden en cuanto a la complejidad temporal de cada uno, siendo el algoritmo ByNet el de mejor resultado, seguido del algoritmo BayesChaid y por último BayesPSO.

El algoritmo ByNet es el más simple de los tres, pero las relaciones estadísticas de independencia no quedan correctamente reflejadas en la red lo que influye en la calidad de los resultados cuando se hacen inferencias sobre la misma. Este algoritmo se presenta en el trabajo pues constituye un antecedente a los que se desarrollan posteriormente: BayesChaid y BayesPSO, el primero de estos dos ofrece buenos resultados como clasificador. Si se escoge un número de niveles pequeño, las redes que obtiene son bastantes simples pues tiene incluida la selección de atributos basada en la significación estadística, resultado que es muy fácil de extender a significación biológica, si se tiene conocimiento sobre el domino del problema.

El algoritmo BayesPSO, es el de mayor complejidad desde el punto de vista de la representación de las redes, y también desde el punto de vista temporal pues depende del número de atributos en el conjunto de datos y de la métrica que se escoja. Además se obtienen redes múltiplemente conexas más complejas, esto influye directamente en la complejidad en la propagación de evidencias. Pero se ha demostrado que si el problema no tiene demasiados atributos (digamos, menos de 100) o se hace una selección previa de estos, el movimiento sobre el espacio de búsqueda garantiza obtener las mejores propiedades sobre la red resultante.

Se comparan los métodos con otros clasificadores bayesianos reportados en la literatura y se demuestra estadísticamente que los algoritmos propuestos muestran un desempeño similar a los clásicos para esta tarea.

Los algoritmos BayesChaid y BayesPSO tienen eficiencia similar o mejoran los resultados de los demás clasificadores con los que se compararon.

3. APLICACIONES DE LOS ALGORITMOS PROPUESTOS

En este capítulo se describen las implementaciones computacionales realizadas y se presentan dos aplicaciones Bioinformáticas: una sobre predicción de interacciones de proteínas y otra sobre predicción de sitios de *splicing* o búsqueda de genes. Además se muestra otra aplicación real sobre diagnóstico de la HTA, lo que ilustra la factibilidad de usar los algoritmos desarrollados en otras áreas.

3.1 Sobre la implementación de los algoritmos

Como se explicó en el epígrafe 1.1.4 se cuenta en el mercado internacional con numerosos productos de *software* para el aprendizaje, edición y ejecución de RB. En múltiples casos estos sistemas tienen un alto precio debido esencialmente a los beneficios que les reportan a las organizaciones que los utilizan. Esta es una de las causas por lo que se hace necesario que las investigaciones lleven conjuntamente desarrollos de productos de *software* para los modelos que se proponen.

En los inicios de esta investigación, se propuso el primer algoritmo explicado en el epígrafe 2.1.1. Las aplicaciones se dedicaron, fundamentalmente a problemas de diagnóstico médico, y para el aprendizaje de RB se usaron productos de *software* tradicionales como: *Mathematica, SPSS, Microsoft Excel*, etc. Para hacer inferencias con estas redes se hizo un primer desarrollo de *software* denominado ByShell (Chávez y Rodríguez 2002), después se desarrolló un sistema que incluyó el aprendizaje y la propagación, pero que aún no estaba totalmente validado (Rodríguez et al. 2006). Estos *software* se desarrollaron en Borlan Delphi, y esta plataforma no es de código libre.

Si se tiene en cuenta que, la implementación de nuevos algoritmos como extensión de la plataforma de aprendizaje automatizado *Weka*, ha mostrado ventajas tales como (Morell et al. 2006):

- Se simplifica la implementación pues solo hay que redefinir métodos ya existentes en *Weka* o crear otros nuevos con las funcionalidades específicas del algoritmo a adicionar.

- Se facilita la validación de un nuevo modelo. No es necesario preocuparse por implementar las variantes de validación, ni las medidas de desempeño a emplear; se pueden utilizar las ya existentes en la herramienta. Es posible hacer ejecuciones en lotes y en varias terminales, sin esfuerzos adicionales de programación utilizando el modo Experimentador.
- Se propicia y facilita la comparación del nuevo algoritmo con otros ya reportados en la literatura e implementados en la herramienta, facilitando el análisis de la factibilidad de este último, algo que sería más costoso en tiempo si se hubiera implementado como un modelo aislado.
- Facilidades para el pre-procesamiento de los datos, y el hecho de que los filtros estén separados de los algoritmos facilita la implementación y el reuso.
- Se propicia el uso y divulgación de los nuevos modelos implementados. El hecho de queden incorporados a *Weka* los hace disponibles para la comunidad de científicos y usuarios de este campo.
- El tiempo de desarrollo del prototipo de un *software* a la medida, utilizando un algoritmo implementado en *Weka*, se disminuye a partir de reutilizar su código.

Para fortalecer esta plataforma, al lograr una contribución en este campo, se pueden incluir los tres algoritmos definidos en el capítulo dos.

El *Weka* está desarrollado en Java, y está estructurado de forma tal que se hace muy sencillo hacer cambios en el código. Por las características del trabajo que se está desarrollando, sólo se hará referencia a lo concerniente a la adición de los algoritmos de aprendizaje estructural propuestos como nuevos modelos de clasificación.

En este sistema existe una clase abstracta que implementa los métodos que debe usar cualquier clasificador, y que es denominada *weka.classifiers.Classifier*. Si se desea implementar un nuevo modelo de clasificación se deben redefinir el método *buildClassifier*(), y al menos uno entre los métodos *clasifyInstance*() y *distributionForInstance*().

Si la adición a *Weka* es un nuevo clasificador bayesiano, el cual forma parte del paquete "*weka.classifier.BayesNet*", el clasificador *BayesNet* usa diferentes tipos de algoritmos para el aprendizaje automatizado de la RB a usar en el proceso de clasificación, a los que denomina algoritmos de búsqueda, cuya tarea principal es definir la estructura de la RB.

La clase que contenga la codificación del nuevo algoritmo debe quedar ubicada en este nombre de espacio: "*weka.classifier.bayes.net.search*", en los paquetes ya definidos: *ci*, *fixed*, *global* y *local*, o crear uno nuevo si el algoritmo no se ajusta a ninguno de los ya existentes. Los algoritmos ByNet y BayesChaid se ubicaron en un nuevo paquete que nombramos: *deterministic*, y el algoritmo BayesPSO en el paquete *global* de los ya definidos en *Weka.* Las relaciones que se establecen entre los paquetes y clases se pueden ver en el diseño del diagrama de relación de clases, que se muestra en el anexo 10, la cual se elaboró con un lenguaje de modelo unificado: Paradigma Visual para UML VisualParadigm (Hernandis 2005), (Headquarters 2007).

Los ficheros de datos se deben definir de uno de los dos tipos siguientes:

- ARFF (*Attribute-Relation File Format*)
- CSV (delimitado por comas)

En el anexo 11 se especifica la sintaxis de estos ficheros.

Para utilizar la herramienta *Weka* previamente se debe instalar la maquina virtual de java, y *Weka* se ejecuta simplemente ejecutando el fichero *weka.jar*, o desde la línea de comandos: *java -jar weka.jar* o por ejemplo C:\WINDOWS\system32\java.exe -Xmx290m -jar " D:\mchavez\Weka\weka.jar"

En el ejemplo la ejecución permite aumentar la memoria virtual con -Xmx290m a partir 290MB como memoria mínima, además se índica el camino en el que está instalado el java y el camino al fichero *weka.jar*.

Cuando se ejecuta *Weka* se tiene una ventana selectora de interfaces, la que se utiliza para realizar pruebas: *Explorer* y para hacer experimentos: *Experimenter*.

Se le puede indicar a *Weka* por línea de comandos la ejecución de un clasificador específico, como los implementados para crear RB, se deben incluir todos los parámetros necesarios: parámetros para abrir ficheros, el clasificador a utilizar, etc.

Esta opción es más difícil, de hecho para usar *Weka* en la versión *Weka* parallel (Arboláez 2008), que permite ejecute la validación cruzada (Kohavi 1995), (Efron y Tibshirani 1997), (Fu y Carroll 2005), (Varma y Simon 2006), con varias terminales, se crearon dos ficheros que ejecutan *Weka* para poder indicar los puertos de conexión, uno para *weka* cliente y uno *weka* servidor, y si no se va a realizar la validación cruzada con varios subconjuntos de datos con distintas terminales, porque no merece la pena hacerlo, se puede usar el *Weka* cliente. En el anexo 11 se pueden ver las características de estos dos ficheros.

Cuando se utiliza el algoritmo PSO se selecciona una medida de calidad, en *Weka* ya se encuentran incorporadas medidas basada en validaciones cruzadas o la medida basada en el método *LOO-CV* (*Leave one out crossvalidation: validación cruzada dejando uno fuera*). Se hizo la extensión a *Weka* de otras medidas de calidad global: dos medidas presentadas en (Eitrich et al. 2007), que a su vez se consideran medidas de calidad robustas para clasificación, se trata del coeficiente de correlación de Matthews y una medida de calidad basada en la sensibilidad y una medida que mide la precisión de la clasificación de (Fawcett 2004), las cuales se han descrito en el capítulo dos.

Alternativamente se incorporó a *Weka* un filtro para selección de atributos propuesto por (Wang et al. 2006) el cual se usa previamente a la construcción de la RB, especialmente cuando hay demasiados atributos (digamos, más de 100) (Chávez et al. 2007b).

3.2 Planteamiento del problema sobre predicción de interacciones de proteínas

Se trata de predecir interacciones de proteínas desde una base de datos de *Arabidopsis thaliana*, la misma se obtuvo por el Departamento de Biología de Sistemas de Plantas[14], a partir de documentación reportada en la literatura. Dicha base contiene información relevante de las interacciones de proteínas de la *Arabidopsis thaliana*: atributos de

[14] Department of Plant Systems Biology, Flanders Interuniversity Institute for Biotechnology (VIB), Ghent University, Belgium.

dominios conservados, valores de expresión para calcular coeficientes de correlación de Pearson, información de anotaciones de GO (*Gene Ontology*, genes ontólogos), OG (*Orthologous Group*, grupos ortólogos), entre otros.

Resultados sobre interacciones de proteínas en esta planta se pueden ver en (Cui et al. 2007). Otras investigaciones de interacciones de proteínas se pueden ver en los trabajos de (Long et al. 2005) en el que se utiliza el CBN para predicción de interacciones de proteínas en hongos, se utiliza por su simplicidad para integrar distintas fuentes de datos genómicos, pero se tiene una alta dependencia estadística entre rasgos, reportan coeficientes de correlación entre 0.904 y 0.97. En (Qi et al. 2006) se explica la importancia del estudio de las interacciones de proteínas en hongos, pero a veces se tienen resultados incompletos, por lo que se obtienen valores altos de la razón de falsos positivos y razón de falsos negativos, en este trabajo se enfrenta el problema con seis clasificadores: árboles aleatorios (*Random Forest*, RF), *k* vecinos más cercanos (*k* NN)-RF, árboles de decisión, CBN, regresión logística y máquinas de vectores soportes (*Supor Vector Machine*), los mejores resultados se obtienen con el *k* NN –RF, la mayor exactitud es del 68 %.

En humanos hay resultados muy interesantes con RB en (Scott y Barton 2007), aquí se reporta un estimado de la razón de falsos positivos del 90%, dada la existencia de pocas bases de datos para el mayor proteoma: el humano.

3.2.1 Análisis de los datos

El conjunto de datos consta de 4314 pares de proteínas, 1438 son ejemplos de verdaderas interacciones y 2876 son ejemplos negativos (o al menos dudosos). Los resultados reportados anteriormente demuestran que identificar simultáneamente ejemplos positivos y negativos resulta difícil, pues es raro encontrar reportes de pares de proteínas que no interactúan, especialmente a gran escala y los casos negativos para el aprendizaje no son del todo seguros. Se siguió el enfoque descrito por (Zhang et al. 2004): se usa un conjunto aleatorio de pares de proteínas (después de filtrar las positivas); esto se justifica porque la fracción de pares de proteínas que interactúan, en el conjunto total de pares de proteínas es pequeño.

3.2.2 Rasgos del problema

Se seleccionaron en total 11 rasgos, mas la variable especial denominada clase, la cual identifica si hay o no una interacción de proteína:

1. "*GO similarity score biological process*: *average" (GO_sim_bp_avg)*
2. "*GO similarity score biological process*: sum" (GO_sim_bp_sum)
3. "*GO similarity score biological process*: maximum" (GO_sim_bp_max)
4. "*GO similarity score cellular component*: *average*" (GO_sim_cc_avg)
5. "*GO similarity score cellular component*: sum" (GO_sim_cc_sum)
6. "*GO similarity score cellular component*: maximum" (GO_sim_cc_max)
7. "*Pearson correlation coefficient for micro-array type* 1" (PCC_1_devtissues)
8. "*Pearson correlation coefficient for micro-array type* 2" (PCC_2_heterog)
9. "*Domain score* 1: *number of common domains*" (domain_match)
10. "*Domain score* 2: *number of common domains*/ *total number of different domains for the two proteins together*" (domain_score)
11. "*Orthology information*" (orthology_score)
12. clase (valor cero identifica que no interactúan las proteínas, y el valor uno que hay una interacción de proteínas)

3.2.3 Discusión de los resultados

La tarea de predicción de interacciones de proteínas se ha enfrentado con múltiples métodos de clasificación, con técnicas estadísticas, de IA y en este caso con RB.

En la Tabla 3.1 se muestran los resultados con técnicas estadísticas: análisis discriminante (AD), regresión logística (RL), árboles de clasificación CHAID (AC CHAID) y árboles de clasificación QUEST (AC QUEST), con técnicas de IA: dos métodos de RB: RB K2 y RB obtenidas mediante pruebas de independencia condicional (RB CI), este caso no se utiliza el CBN, pues se tiene una alta correlación entre los rasgos del problema, el valor mayor del coeficiente de correlación es 0.782, lo que indica dependencia estadística entre los rasgos

del problema, además se muestran resultados de árbol de decisión con el algoritmo ID3 (AD ID3) (Quinlan 1986).

Tabla 3.1. Resultados con otras técnicas de aprendizaje supervisado para el problema de predicción de interacciones de proteínas

Algoritmo	***Exactitud***	***Área bajo la curva ROC***	***rVP***	***rVN***
AD	82.2 %	0.825	0.523	0.971
RL	82.4%	0.813	0.522	0.974
AC CHAID	81.8 %	0.819	0.479	0.988
AC QUEST	82.4 %	0.818	0.514	0.979
RB K2	82.89 %	0.834	0.571	0.958
RB CI	81.75%	0.835	0.567	0.943
AD ID3	81.68%	0.813	0.600	0.940

Con ninguno de los métodos de clasificación utilizados se ha logrado incrementar los porcientos de verdaderas interacciones de proteínas. Tanto la exactitud como el área bajo la curva ROC se mantienen con valores similares, igual sucede con la razón de negativos y positivos.

Los resultados con los algoritmos que se proponen en la tesis se muestran en la Tabla 3.2, estos son similares a los que se muestran en la Tabla 3.1 para otras técnicas de aprendizaje supervisado, pero estos resultados son mejores a los que se describieron en el epígrafe 3.2 para interacciones de proteínas de otros organismos.

Tabla 3.2. Resultados de los algoritmos propuestos en la tesis para obtener RB en el problema de predicción de interacciones de proteínas

Algoritmo	***Exactitud***	***rVP***	***rVN***	***Área bajo la curva ROC***	***mcc***
ByNet	81.45 %	0.461	0.991	0.775	0.583
BayesChaid	82.10 %	**0.605**	0.929	0.838	0.582
BayesPSO	**82.75** %	0.565	**0.959**	**0.839**	**0.600**

En todas las ejecuciones de los algoritmos se usó la validación cruzada con 10 subconjuntos para estimar la predicción de error de los métodos propuestos y ver el comportamiento frente a otros algoritmos (Varma y Simon 2006).

Si se evalúa los algoritmos por el área bajo la curva ROC y la razón de verdaderas interacciones, el algoritmo BayesChaid muestra el mejor desempeño, si se mide por la exactitud es el algoritmo BayesPSO.

3.2.4 Validación mediante el uso del modelo obtenido con el algoritmo ByNet

Cuando se obtiene un modelo de RB, este se puede evaluar por el uso que se le da a la red, en este caso se escoge el algoritmo ByNet para mostrar cómo se puede usar las RB, a pesar de que no es el que mejores resultados obtuvo se demostró en el capítulo anterior que es el de menos complejidad temporal. Con la RB se puede inferir cualquier variable tanto predictivas como la variable dependiente o clase.

Como se explicó, desde el punto de vista estadístico es posible definir cuál es la importancia de las variables con respecto a la variable dependiente.

En este caso las 11 variables están correlacionadas, y en la red están todas las variables, pero como se aprecia en la Figura 3.1 las variables que más se relacionan con la clase son *domainmatch* y *PCC1devtissues*.

Según los resultados de la prueba Chi-cuadrado *domainscore* y *PCC2hetreog* están más relacionados con la variable clase que *PCC1devtissues*, pero como hay correlación entre todas las variables predictivas, estas variables forman parte del primer árbol que se crea, y la red queda como se observa en la Figura 3.1.

Debe quedar claro que esta dependencia es puramente estadística, y si coincide con el significado biológico, las predicciones deben ser mejores.

Cuando no se tienen evidencias en la red se tienen las probabilidades a priori para cada valor de las variables, por ejemplo la probabilidad a priori de una interacción de proteínas es 0.34, lo que se observa en la Figura 3.2.

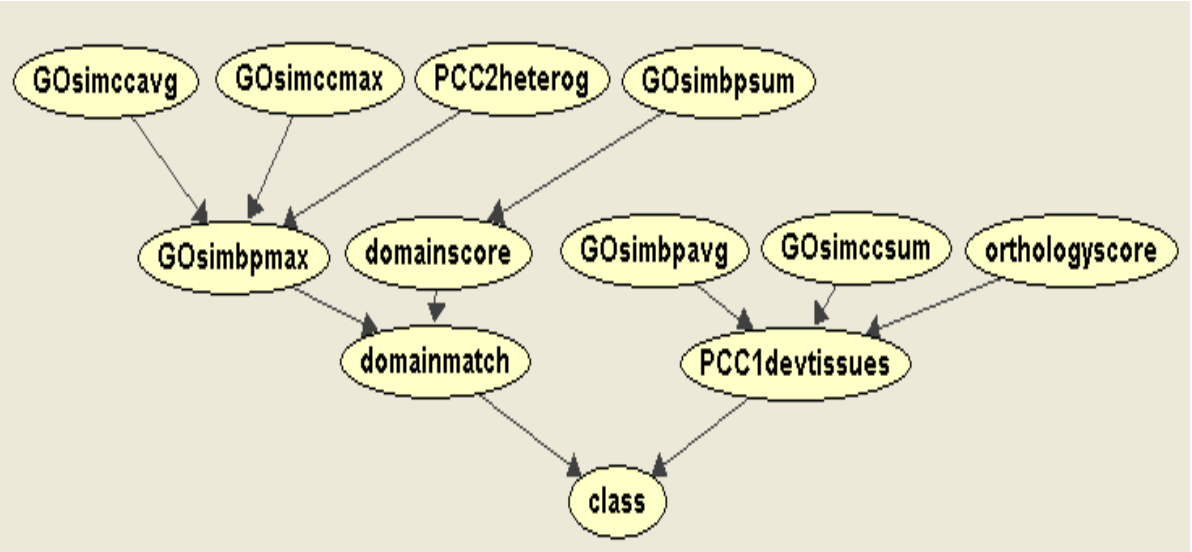


Figura 3.1. RB obtenida con el algoritmo ByNet

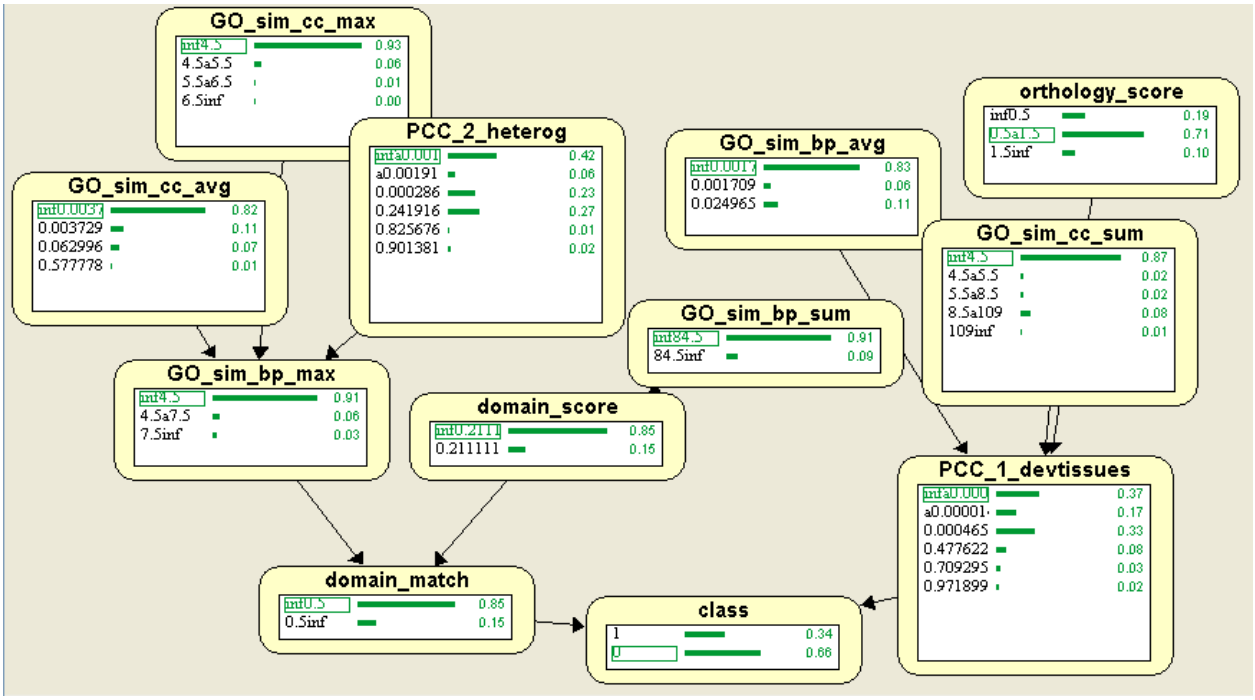


Figura 3.2. Red Bayesiana obtenida con el algoritmo ByNet sin evidencias

Si *domainmach* es mayor que 0.5 se obtiene una probabilidad de 0.97 de que exista una interacción de proteína, lo cual se observa en la Figura 3.3.

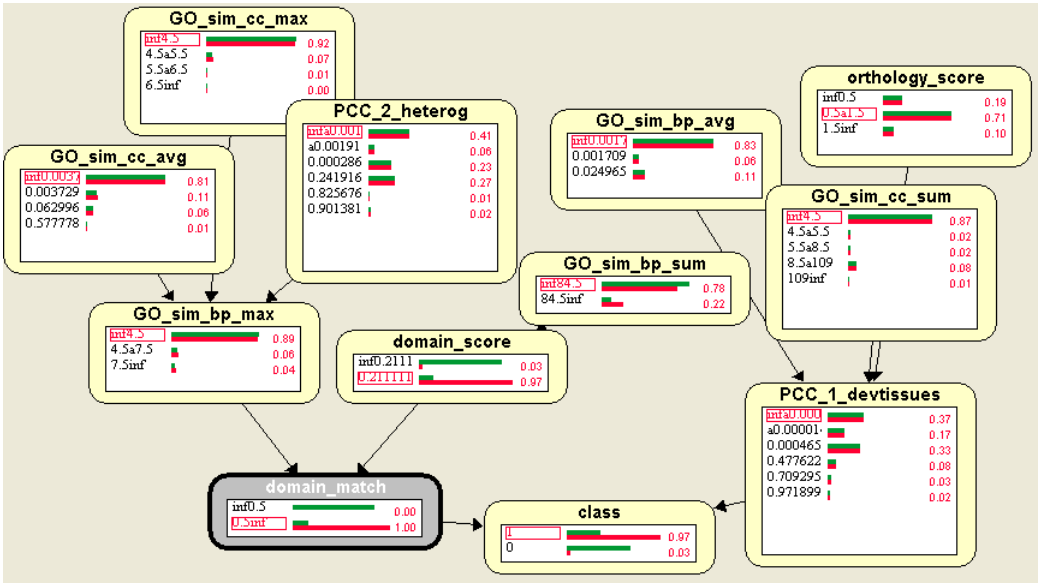


Figura 3.3. Red Bayesiana cuando la evidencia es: domainmach mayor que 0.5

Si las evidencias son que *GO_sim_bp_max* es menor que 4.5, *domain_score* es menor que 0.21 y *PCC_devtissues* es menor que 0.00001 la probabilidad de una interacción de proteína es 0.85, además se tiene probabilidad uno de que *domain_match* es menor que 0.5.

Si se quiere saber cómo se comportan las variables cuando se tiene certeza de una interacción, los resultados en la red son los que se aprecian en la Figura 3.4.

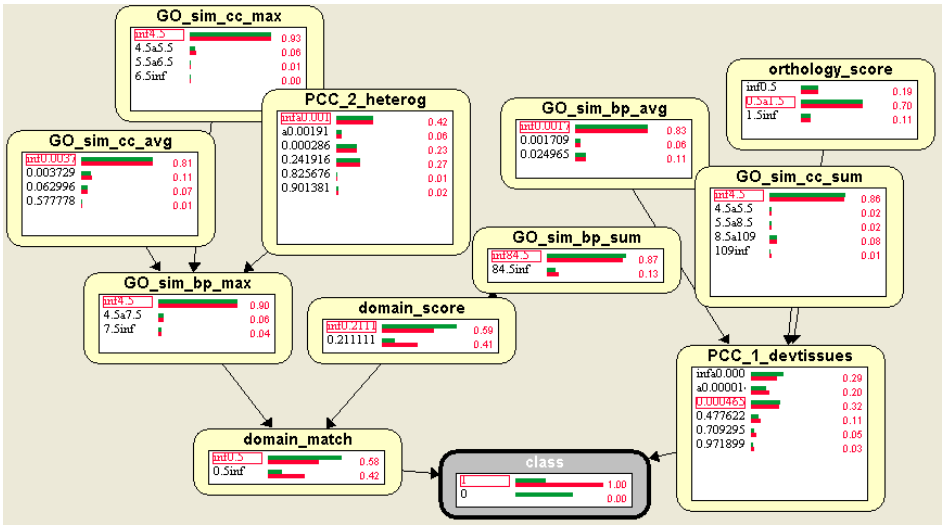


Figura 3.4. Red Bayesiana cuando la evidencia es: se conoce que hay una interacción de proteína, o sea la variable *clase* toma el valor 1.

3.2.5 Mejorando el balance de verdaderos positivos y negativos

Todo aprendizaje supervisado es más eficiente y eficaz si la base de entrenamiento tiene casos muy bien distinguidos en cada clase, digamos, se conocen positivos sin duda y negativos sin duda. Es un principio natural y obvio, incluso del aprendizaje no computarizado, por ejemplo el de un niño.

La base de entrenamiento de interacciones de proteínas no presenta estas características idóneas. De hecho, de los denominados negativos, no se está absolutamente seguro que lo sean, sólo se conoce que esas interacciones no están reportadas como positivas. Entonces el aprendizaje puede estar sesgado y hasta es natural que este grupo tenga el mejor porcentaje de clasificación, simplemente porque el clasificador aprende con dudas.

Una técnica para aliviar este sesgo es filtrar los falsos negativos, concretamente seleccionar dentro de los negativos, aquellos de los cuáles podemos tener "un poco más de confianza" de que sean negativos.

Nos basamos en reglas puramente probabilísticas y se siguen los siguientes criterios:

1. Eliminar de la muestra de aprendizaje aquellos casos en que la predicción de falsos negativos tuvo una probabilidad por debajo de un cierto umbral.

2. Al mismo tiempo, evitar eliminar demasiados casos para no perder información que puede ser útil. Esto se logra fijando dicho umbral no demasiado alto y al mismo tiempo escogiendo el clasificador que más Verdaderos Positivos (y por tanto menos Falsos Negativos) tuviese: resultó el BayesChaid, ver Tabla 3.2.

Los resultados con los algoritmos, una vez que se ha reducido la base de datos con los falsos positivos que obtienen el algoritmo *BayesChaid* se muestran en la Tabla 3.3. Estos mejoran sustancialmente, pues son superiores a los de las Tablas 3.1 y 3.2, para todas las medidas de validación.

Mahdavi ya había utilizado esta técnica, auxiliándose de criterios biológicos pero este reduce los falsos positivos (Mahdavi y Lin 2007).

Los resultados con respecto a otros clasificadores, confirma el buen desempeño de los algoritmos ByNet, BayesChaid y BayesPSO.

Tabla 3.3. Resultados de los algoritmos propuestos para el problema de interacciones de proteínas cuando se reducen falsos negativos.

Algoritmo	***Exactitud***	***Área bajo la curva ROC***	***rVP***	***rVN***
ByNet	93.41 %	0.930	0.818	0.970
BayesChaid	92.32 %	0.974	**0.906**	0.918
BayesPSO	**95.69** %	**0.990**	0.863	0.986
AD	93.1 %	0.962	0.719	0.997
RL	94.7 %	0.963	0.823	0.985
AD CHAID	93.9 %	0.978	0.775	0.99
AD QUEST	94.5 %	0.921	0.818	0.984
RB K2	93.08 %	0.968	0.761	0.983
RB CI	91.88%	0.976	0.816	0.953
AD ID3	92.4%	0.942	0.873	0.958

Otra alternativa consiste en estudiar el comportamiento de los modelos desde las posibilidades de análisis que permiten las curvas ROC. En la Tabla 3.4 se muestra el comportamiento del modelo obtenido con el algoritmo BayesChaid en función de distintos puntos de corte. Puede observarse que cuando la probabilidad de ser positivo en la clasificación se reduce a 0.18 se logra un balance de las razones de *FP* y *FN*, pero por supuesto, a costa de pérdidas en la exactitud.

Tabla 3.4. Resultados del análisis de balance de verdaderas y falsas interacciones de proteínas

Punto de corte	***rVN***	***rFP***	***rVP***	***rFN***	***Exactitud***
0.50	0.928	0.072	0.616	0.384	82.38%
0.40	0.912	0.088	0.636	0.364	82.01%
0.42	0.913	0.087	0.627	0.373	81.78%
0.30	0.867	0.133	0.672	0.328	80.23%
0.28	0.850	0.150	0.690	0.310	79.65%
0.18	0.758	0.242	0.752	0.248	75.59%

Para este problema de predicción de interacciones de proteínas, los resultados de los algoritmos que se proponen en la tesis muestran un comportamiento similar a los que se

reportan en la literatura, además para algunas de las medidas que evalúan la calidad de los resultados estas se mejoran. Por ejemplo una de los problemas fundamentales en esta aplicación es el bajo porciento de verdaderos positivos, sin embargo con el algoritmo BayeChaid este valor es relativamente mejor al obtenido por el resto de los clasificadores aplicados. En cuanto a exactitud el algoritmo BayesPSO muestra los mejores resultados.

3.3 Planteamiento del problema sobre localización de *splice sites*

Uno de los problemas fundamentales de la Bioinformática, específicamente de la genómica consiste en la localización de genes dentro del genoma de un cierto organismo. Las secuencias de ADN de la mayoría de los genes se transcriben en ARN mensajero que a su vez se traducen en las proteínas. En los procariotas (organismos menos desarrollados) el ARN mensajero es una mera copia del ADN. Sin embargo, en los eucariotas, el ADN contiene en los genes segmentos codificantes (*exones*) y no codificantes (*intrones*) y estos últimos son "cortados" durante el proceso de transcripción, mecanismo conocido como *splicing* que coloca a los exones de un gen consecutivamente, y listos para traducirse en la secuencia de aminoácidos que conforman la proteína (Figura 3.5) (Foley y Lewin 2004). La detección de intrones y exones constituye una de las vías para abordar el problema de la localización de los genes.

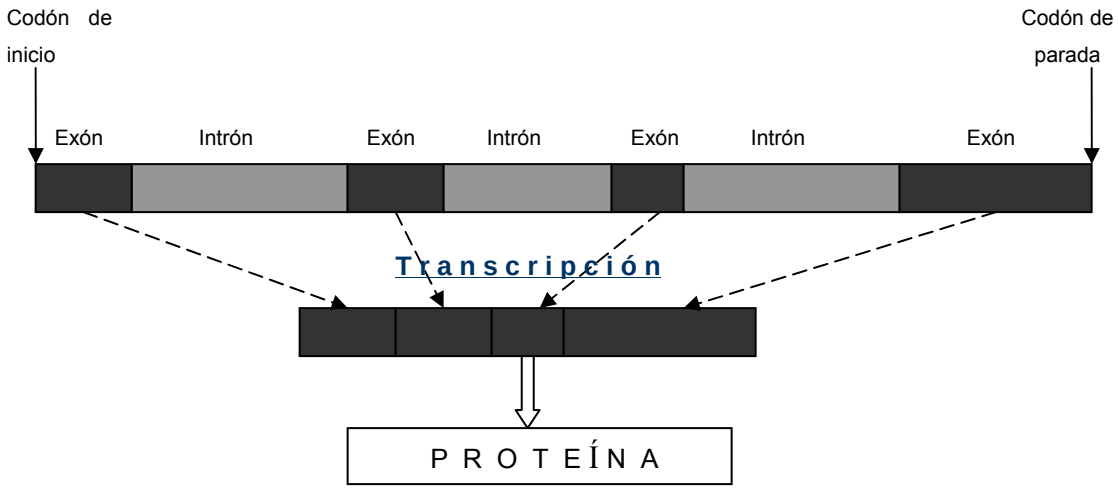


Figura 3.5. Esquema de la conformación de un gen como sucesión de exones e intrones. En la transcripción a RNA mensajero se desechan los intrones y se colocan los exones consecutivamente para traducirse en la proteína

Los bordes de estos intrones es lo que se conoce como *splice sites*. El *splice sites* en el principio del intrón es conocido como *donor*, mientras que el que lo finaliza es conocido como *acceptor*. Los "*donors*" se caracterizan por la presencia del par de nucleótidos "GT" al inicio del intrón, los "*acceptors*" se identifican por el par "AG" al final del intrón (Figura 3.6). Entonces se podría intentar reconocer *donors* y *acceptors* a través de estos dinucleótidos y con ellos los intrones. La limitación de este enfoque es que estos dinucleótidos abundan en el genoma, y sólo un pequeño por ciento de estas combinaciones son *splice sites* reales (Saeys 2004).

Si se tienen secuencias con el par "GT" de las cuales se conozca si son verdaderos o falsos *donors* se puede intentar "aprender" a clasificarlos utilizando la información de las bases nucleotídicas de su entorno y otro tanto podría hacerse a partir de secuencias con el par "AG" de las cuales se conozca si son verdaderos o falsos *acceptors*. Así el problema original se descompone en dos problemas de clasificación.

Figura 3.6. Esquema general de un intrón entre dos exones. Observe como el inicio y el fin del intrón se marcan por los *splice sites*

Se ha abordado el problema de detectar si las combinaciones de estos nucleótidos ("GT" o "AG") son verdaderos o falsos *splice sites* desde diversas técnicas de clasificación de Estadística y de IA. En la presente tesis se aplican las RB para afrontar el problema.

3.3.1 Análisis de datos

La base de datos de *splice sites* para humanos fue construida en la Universidad de Ghent, Bélgica, a partir de obtener ARN mensajero desde la base de datos pública EMBL (Base de datos de secuencias nucleotídicas) (EMBL 2009). Se eliminaron los genes redundantes y se obtuvo una base de 1115 genes. Desde estos genes sólo se tuvieron en cuenta los intrones con *splice sites* canónicos, o sea GT para *donor* y AG para *acceptor*, que fueron usados como positivos. Las instancias negativas fueron definidas a partir de los dinucléotidos GT y AG que se encontraban en sitios que no eran de *splice sites*.

Quedaron construidas dos bases de datos de secuencias de nucleótidos, con el objetivo de clasificar verdaderos y falsos *splice sites*: identificación de *donors* y *acceptors*. Las bases de datos para este trabajo se conformaron con 7000 casos cada una, 6000 falsos y 1000 verdaderos, tal como sugiere la proporción aproximada real de verdaderos y falsos *splice sites* en los genomas (Saeys 2004).

3.3.2 Rasgos del problema

La longitud de las secuencias de pares intrón-exón es muy variable. Debido a esto, se realizó un análisis estadístico acerca de la distribución de la longitud de exones e intrones en los genes del *homo sapiens* con el objetivo de encontrar una ventana que permitiera todas las secuencias de la misma longitud (Grau et al. 2007a). De este estudio se obtuvo que es suficiente con 80 bases nucleótidas en los intrones y 22 en los exones. En la base de los *donors*, por ejemplo, ello se traduce en 22 posiciones (del exón) a la izquierda del par GT y otras 78 posiciones (del intrón) a la derecha de dicho par.

En las dos bases de datos se representan las posiciones nucleotídicas a partir del álgebra Booleana primal (Sánchez 2006). Esto es: G-00, T-10, A-01, C-11. De esta manera, como se tienen 102 posiciones (22 exón, 80 intrón), se convierten en 204 variables predictivas, que se etiquetan como v1_1, v1_2, v2_1, v2_2,…, v102_1, v102_2 y cada una de ellas es un valor binario (0 ó 1). En particular, en la base de *donors* se tienen las posiciones v23 y v24 constantes porque representan a GT, esto es v23_1=v23_2=0 porque corresponden a G y v24_1=1, v24_2=0 porque corresponden a T. Como son constantes, no intervendrán en el

proceso de clasificación pero se incluyen en la base para claridad de la interpretación de la misma.

3.3.3 Discusión de los resultados

El problema de clasificación de *donors* se seleccionó para medir el desempeño de uno de los métodos propuestos, cuando se utiliza la validación cruzada con 10 subconjuntos distribuida entre distintas máquinas, debido a que la base de datos es la que mayor cantidad de variables y de casos cuenta entre los tres problemas que se intentan resolver. Se realizó con el método *BayesChaid*, pero se puede probar con cualquier método, ya sea de los que se proponen, o cualesquiera de los implementados en Weka. Los parámetros seleccionados para realizar las pruebas fueron: dos padres, tres niveles y 100 casos como mínimo en las sub-poblaciones., pero esto fue sólo para realizar el experimento, podrían escogerse otros. Las ejecuciones se hicieron en varias computadoras utilizando desde dos hasta cinco terminales remotas para realizar la validación cruzada.

El resultado de este experimento es que se logró disminuir el tiempo de más de cinco horas a menos de una hora. Esta mejora en tiempo resulta fundamental si el problema a resolver es de Bioinformática, porque en la mayoría de los problemas que se presentan tienen grandes volúmenes de información a procesar. Los resultados obtenidos por los algoritmos propuestos para el aprendizaje de la estructura de RB, descritos en el capítulo dos se muestran en las Tablas 3.5 y 3.6.

Tabla 3.5. Resultados en *donnor* con varios clasificadores

Algoritmo	***Exactitud***	***Área bajo la curva ROC***	***rVP***	***rVN***
ByNet	**93.64**	**0.975**	**0.771**	0.964
BayesChaid	**92.9**	0.958	0.713	0.965
BayesPSO	92.2	0.949	0.669	0.964
AD QUEST[1]	91.1	0.946	0.731	0.956
RB K2	92.44	0.959	0.678	0.966
RB TAN	92.52	0.959	0.674	0.967

[1] Obtenido con el Algoritmo QUEST para obtener árboles de decisión, resultados reportados en (Grau et al. 2007b)

En ambos problemas la clasificación con los modelos obtenidos es satisfactoria, el método BayesChaid obtiene los mejores resultados de falsos positivos en *acceptors* y ByNet en *donors*, esta medida es una de las que se quiere minimizar, una de las causas de este resultado es el desbalance de los verdaderos *splice sites* con respecto a los sitios aleatorios o casos negativos.

En (Degroeve et al. 2002) se obtienen mejores resultados para clasificación de *splice site*s en *arabidopsis thaliana*, lo que se logra con una buena selección de rasgos, resultados similares para esta planta en la tesis de (Saeys 2004), pero no ocurre lo mismo en la clasificación de sitios de *splice sites* en humanos.

Cuando se usa el método BayesPSO no se reportan mejoras, aún usando las medidas *fitness* que han sido implementadas por otros autores para estas situaciones de desbalance y que se han incluido en la tesis como métricas de calidad, las cuales se describen en el capítulo dos de la tesis. Sin embargo es conveniente usar PSO cuando se hace una selección de atributos previa a la clasificación como se propone en (Chávez et al. 2007b), lo que ha reportado mejores resultados.

Tabla 3.6. Resultados en *Acceptors* utilizando distintos algoritmos para clasificar

Algoritmo	***Exactitud***	***Área bajo la curva ROC***	***rVP***	***rVN***
ByNet	91.38	0.953	0.615	0.964
BayesChaid	92.3	0.955	**0.784**	0.946
BayesPSO	91.04	0.937	0.671	0.95
AD QUEST	89.2	0.913	0.487	0.95
RB K2	92.38	0.953	0.729	0.956
RB TAN	92.32	0.953	0.72	0.957

3.3.4 Validación mediante el uso del modelo obtenido con el algoritmo *ByNet*

Para clasificación de *donors* y *acceptors* con el algoritmo ByNet el mejor modelo de RB se obtiene cuando la red tiene un solo nivel, o sea la clase depende de las variables más relacionadas con esta desde el punto de vista estadístico. Se escogen 10 variables, para tener en cuenta algunos nucleótidos alrededor del sitio de *splicing*.

La red para *donor* se muestra en la Figura 3.7, las variables que forman la red están alrededor del sitio *donor*, las variables que le corresponden a este sitio son v23 y v24:

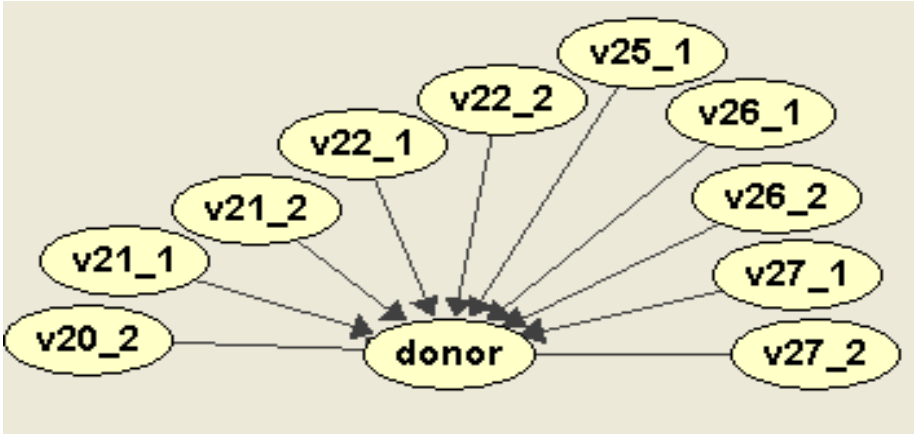


Figura 3.7. Red bayesiana para clasificación de *donors* con el algoritmo *ByNet*

Cuando aún no se tienen evidencias, la red muestra las probabilidades a priori para cada una de las variables, y a medida que se conoce el valor de determinada variable, se infieren otras probabilidades en base a la evidencia. A continuación se describen algunos ejemplos, en los que para dar la evidencia de un nucleótido es necesario conocer dos posiciones acorde a la representación que se explicó en el epígrafe 3.3.2, pero si sólo se conoce una de las variables asociada a un nucleótido la posición puede ser uno u otro.

Si se tienen evidencias de posiciones en la secuencia (las que se indican en las columnas a la izquierda de la Tabla 3.7 y 3.8, la probabilidad de inferir o no un sitio *donor* se indica en la última columna de dichas tablas. Las redes con los resultados de la propagación ante distintas evidencias para *donors* y *acceptors* se pueden ver en los anexos 7 y 8.

La red obtenida para *acceptors* se muestra en la Figura 3.8, y en las Tablas 3.9 a la 3.11 se muestran resultados de la propagación de evidencias en la clasificación de *acceptors* (posiciones que corresponden a las variables v81 y v82) del mismo modo que se explicó para la clasificación de *donors*.

Tabla 3.7. Probabilidad de que se tenga un sitio *donor* cuando se conocen tres posiciones antes y tres después de este sitio

V20	V21	V22	V23	V24	V25	V26	V27	P(*donor* | evidencias)
T o C	A	G	G	T	G o A	A	G	0.94

Tabla 3.8. Probabilidad de que no se tenga un sitio *donor* cuando se conocen tres posiciones después de este sitio

V20	V21	V22	V23	V24	V25	V26	V27	P(no *donor* | evidencias)
			G	T	T o C	C	C	0.91

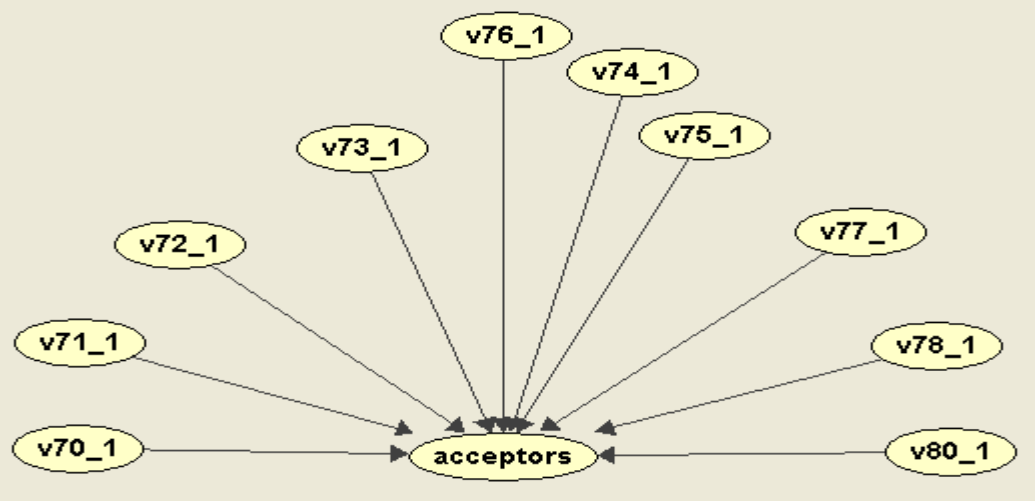


Figura 3.8. Red bayesiana para clasificación de *acceptors* con el algoritmo *ByNet*

Tabla 3.9. Probabilidad de se tenga un sitio *acceptors* cuando se evidencia la presencia de los nucleótidos T o C en diez posiciones antes de este sitio.

V70	V71	V72	V73	V74	V75	V76	V77	V78	V79	V80	P(*acceptors* | evidencias)
T o C	T o C	T o C	T o C	T o C	T o C	T o C	T o C	T o C		T o C	0.82

Tabla 3.10. Probabilidad de no se tenga un sitio *acceptors* cuando se evidencia la presencia de los nucleótidos A o G en siete posiciones antes este sitio

V70	V71	V72	V73	V74	V75	V76	V77	V78	V79	V80	P(no *acceptors* | evidencias)
				A o G	A o G	A o G	A o G	A o G	A o G	A o G	0.96

Tabla 3.11. Probabilidad de que no se tenga un sitio *acceptors* cuando se evidencia la presencia de los nucleótidos A o G en once posiciones antes de este sitio

V70	V71	V72	V73	V74	V75	V76	V77	V78	V79	V80	P(no *acceptors* | evidencias)
A o G	A o G	A o G	A o G	A o G	A o G	A o G	A o G	A o G	A o G	A o G	0.99

3.3.5 Mejorando la predicción de verdaderos splice sites

Para lograr mejorar la predicción de verdaderos *splice sites* se plantea la necesidad de reducir los casos considerados falsos positivos (FP), o sea los casos negativos de *donors* o *acceptors* que se predicen erróneamente como positivos. En la predicción de genes (en este caso a través de la localización de sitios de *splicing*), la disminución de los FP es fundamental porque dicha clasificación es apenas un primer paso para un conjunto de investigaciones costosas en dinero y tiempo destinadas a caracterizar completamente el gen y su funcionalidad y no pueden desgastarse estos recursos en un gen dudoso (Chuang y Roth 2001).

En la Tabla 3.12 se muestra cómo se comporta esta modificación, usando el punto de corte que utiliza el clasificador para emitir una respuesta ante un caso dado. Los resultados se obtienen desde la curva ROC. Se escogieron los resultados en *donors* con el algoritmo BayesChaid para mostrar este resultado. Como se aprecia en la Tabla 3.12 el punto de corte se escoge por encima de 0.5 con el objetivo de lograr una sensibilidad dada, y en consecuencia la reducción de los falsos positivos.

Tabla 3.12. Resultados en la predicción de verdaderos *splice sites* cuando se reducen los falsos positivos para la base de *donors*.

Punto de corte	*rVN*	*FP*	*rFP*	*rVP*	*rFN*	*Exactitud*
0.5	0.965	210	0.035	0.713	0.287	92.90%
0.57	0.973	164	0.027	0.669	0.331	92.93%
0.67	0.983	105	0.018	0.585	0.415	92.57%
0.77	0.990	59	0.010	0.497	0.503	91.97%
0.89	0.998	14	0.002	0.308	0.692	89.77%
0.95	1.000	3	0.001	0.153	0.847	87.86%

De esta forma se verifica la presencia de un *donor* solamente si se está muy seguro de ello, aunque existe pérdida de exactitud esta no es significativa (Chuang y Roth 2001).

Los mejores resultados los obtiene el algoritmo ByNet, la red muestra relaciones desde el punto de vista estadístico de variables alrededor del sitio *donor* (*acceptor*) directas con la variable clase. Los algoritmos que se proponen en la tesis muestran mejores medidas de validación que el resto de los clasificadores aplicados.

3.4 Predicción de la Hipertensión arterial

Los modelos de RB pueden ser utilizados en otros dominios de aplicación, específicamente se han desarrollado aplicaciones en varios problemas biomédicos, uno de los cuales, relacionados con la HTA, se describe en el trabajo. También han sido utilizados en otros campos, por ejemplo, en la elaboración de sistemas tutoriales inteligentes, ver por ejemplo la tesis de maestría de (Medina 2007), (Medina et al. 2007).

La HTA es un factor de riesgo para numerosas enfermedades, entre las que se destacan con una incidencia mayor las coronarias. Muchos autores coinciden en asegurar que la HTA es por sí misma una enfermedad.

El corazón bombea sangre a través de las arterías a todo el cuerpo, la tensión que se genera en el interior arterial se denomina presión arterial. La HTA ó presión alta es la elevación de esta presión arriba de un límite que se considera normal (140/90 mmHg). Es la principal enfermedad crónica degenerativa y la más común causa de muerte, afecta aproximadamente al 20% de la población mundial. La elevación anormal de la presión constituye un importante factor de riesgo coronario.

Varios estudios realizados consideran esta enfermedad como la primera causa de muerte en el mundo (Ordúñez et al. 2001), (Silva 2009). En Cuba y en particular en el municipio de Santa Clara, es la segunda causa de muerte.

Al medirse la presión arterial se anotan dos números, el mayor es la presión sistólica, corresponde a la presión del corazón al contraerse para bombear la sangre y el número menor es la presión diastólica, que es la presión de la sangre en las arterias en la fase de relajamiento del corazón. Para un correcto diagnóstico de hipertensión, el médico mide

varias veces la presión arterial, en diferentes condiciones de esfuerzo y en diferentes horas del día. En personas hipertensas, la variación es mayor y permanece alta la mayor parte del día, incluso en los períodos de descanso.

La Organización Mundial de la Salud la ha denominado epidemia silenciosa pues por lo regular se presenta de forma asintomática, ocasionando daños como: trombosis, hemorragias cerebrales, infarto del miocardio, muerte súbita, insuficiencia renal, entre otras.

El conocimiento actual de éste problema de salud pública a nivel mundial, obliga a buscar estrategias de detección, control y tratamiento (Benet et al. 2003).

Los factores de riesgo de esta enfermedad son tan disímiles que pueden ir desde factores económicos y sociales, hasta ambientales y étnicos, por lo que su diagnóstico no debe limitarse simplemente a la toma de la presión arterial sistólica y diastólica sino analizar cada uno de estos factores. Sin lugar a dudas el estudio de todos los factores requiere de una gran cantidad de recursos materiales y humanos de los que no siempre es posible disponer.

Como parte de un proyecto de investigación conjunta entre la UCLV y la Universidad de Oviedo, hace algunos años se creó la "Proyección del Centro de Desarrollo Electrónico hacia la Comunidad" (PROCDEC) cuyo objetivo principal es desarrollar un estudio de personas supuestamente normotensas primero en la ciudad de Santa Clara y luego en toda la nación de modo que se desarrolle una Campaña contra la HTA y el riesgo vascular.

En el desarrollo de este proyecto participa un grupo multidisciplinario formado por psicólogos, cardiólogos, nefrólogos, genetistas, fisiólogos, clínicos, médicos de laboratorio, ingenieros, matemáticos especialistas en estadística y cibernéticos.

Participan además especialistas en Medicina Integral General de los centros hospitalarios José Ramón León, Chíqui Gómez y Ramón Pando Ferrer. Estos especialistas realizan la captura de los datos, mientras el grupo multidisciplinario es quien realiza el diagnóstico (Gutiérrez 2003).

3.4.1 Análisis de los datos

En este estudio la muestra estuvo constituida por un total de 849 individuos entre 18 a 78 años de edad, de ambos sexos, pertenecientes a cinco policlínicos de la ciudad de Santa Clara.

Se confeccionó una historia clínica con información del paciente contenida en las siguientes variables: edad, sexo, raza, índice de masa corporal, consumo de bebidas alcohólicas, fuma, diabetes mellitus, dislipidemia, número de padres con HTA, número de abuelos con HTA, tensión arterial sistólica y diastólica basal, al primer y segundo minuto, presión arterial media, glicemia, triglicéridos, colesterol total hdl y ldl, entre otras. A partir del análisis de todas las variables, el comité de expertos clasificó a cada paciente como normotensos o hipertensos. La base de casos cuenta con 23 rasgos además de la clase (diagnóstico).

Los casos hipertensos incluyen los casos declarados hipertensos o hiperreactivos (pre-hipertensos). El estado de hiperreactividad vascular se consiguió mediante una ergometría isométrica denominada Prueba del Peso Sostenido (PPS) (Benet et al. 2001). Esta prueba basa su principio en introducir al método clásico de la medición de la tensión arterial la condición de que los pacientes realicen, en posición sentada, un ejercicio físico isométrico, que consiste en mantener un peso de 500 gramos con el brazo izquierdo extendido en ángulo recto al cuerpo durante 2 minutos. La presión arterial se toma en el brazo contrario antes del ejercicio y a partir del segundo 50 del segundo minuto.

3.4.2 Discusión de los resultados

Se ejecutaron los algoritmos propuestos y los resultados de las principales medidas para su evaluación se aprecian en la Tabla 3.13.

Se usó la validación cruzada con 10 subconjuntos en la estimación del error de la clasificación. En esta aplicación los resultados se consideran suficientemente buenos.

En el capítulo dos de la tesis se ha expresado que el algoritmo ByNet se ha utilizado en problemas de diagnóstico médico con buenos resultados, se confirma en la predicción de la HTA que aunque no es el mejor clasificador, el resultado es satisfactorio.

Tabla 3.13. Resultados para la predicción de la HTA usando los algoritmos propuestos y otros tres clasificadores.

Algoritmo	***Exactitud***	***Área bajo la curva ROC***	***rVP***[1]	***rVN***[2]
ByNet	94.69 %	0.980	0.976	0.864
BayesChaid	98.35 %	**0.999**	**0.989**	0.978
BayesPSO	97.05 %	0.996	0.975	0.959
AD QUEST	96.60 %	0.996	0.952	0.980
RB K2	96.80%	0.997	0.971	0.959
RB TAN	97.10%	0.997	0.979	0.95

[1] La clase de los positivos se corresponde con los casos hipertensos.

[2] La clase de los negativos son los casos normotensos.

3.4.3 Validación mediante el uso del modelo obtenido con el algoritmo *BayesChaid*

Para mostrar el uso de la RB obtenida para el diagnóstico de la HTA, se puede escoger cualquiera de los modelos obtenidos con buenas características.

Para ejemplificar tales inferencias se utilizó el modelo obtenido con el algoritmo BayesChaid que es el que mejores resultados dio como modelo clasificador.

Para la propagación de evidencias se utilizó el *software* ELVIRA descrito en el capítulo uno. El algoritmo de propagación utilizado es el de eliminación de variables.

Algunos ejemplos para ver el comportamiento del modelo de RB obtenido: por ejemplo sin evidencias la probabilidad de ser hipertenso es de 0.52. Otro ejemplo, un caso con la presión sistólica al minuto uno alta, se eleva la probabilidad de hipertenso a 0.97, también aumenta la probabilidad de la presión sistólica al segundo minutos, así como las presiones diastólicas y la presión arterial media. En el anexo 9 se muestran otros ejemplos para distintas evidencias. Existe la posibilidad de propagación conocidas varias evidencias simultáneamente, o por ejemplo si se conoce a priori que el paciente es hipertenso, como se comportan las otras variables que lo caracterizan.

3.4.4 Mejorando la predicción de falsos sanos

Se hace necesario reducir los casos falsos negativos, pues en los problemas de diagnóstico médico es más peligroso declarar un enfermo como sano, que lo inverso. Ello se logra si se reduce el umbral de probabilidad de enfermo por debajo de 0.5. Se escoge para este análisis

el modelo obtenido con el algoritmo BayNet pues es el que peores resultados obtiene en cuanto a verdaderos sanos o normotensos.

Este comportamiento para distintos puntos de corte para realizar esta predicción se aprecia en la Tabla 3.14. A medida que se disminuye el punto de corte se reducen los falsos negativos de treinta casos a cinco. La exactitud en la clasificación se mantiene alta.

Tabla 3.14. Resultados de HTA cuando se reducen casos falsos negativos

Punto de corte	***rVN***	***rFP***	***rVP***	***FN***	***rFN***	***Exactitud***
0.5	0.976	0.024	0.864	30	0.136	94.70%
0.41	0.965	0.035	0.868	29	0.132	93.99%
0.37	0.940	0.060	0.882	26	0.118	92.46%
0.33	0.914	0.086	0.914	19	0.086	91.40%
0.316	0.909	0.091	0.941	13	0.059	91.76%
0.312	0.908	0.092	0.964	8	0.036	92.23%
0.25	0.903	0.097	0.977	5	0.023	92.23%

Los resultados obtenidos para este problema de predicción HTA son suficientemente buenos, tanto con los algoritmos de aprendizaje estructural de RB que se proponen, como con los algoritmos utilizados en la comparación.

Estos resultados consolidan que los algoritmos que se proponen en la tesis muestran desempeño similar a los que se reportan en la literatura ante tareas similares.

Conclusiones parciales del capítulo

Se han aplicado los algoritmos que se describen en el capítulo anterior para el aprendizaje estructural de RB a tres problemas, dos de la Bioinformática y al diagnóstico de la HTA. En todos los casos los resultados son satisfactorios. Los mejores resultados se obtienen con los algoritmos BayesChaid y Bayes PSO.

Para cada una de las aplicaciones se hace un análisis de cómo mejorar los resultados del clasificador en función del dominio de la aplicación que se desarrolla. Para el caso de la predicción de interacciones de proteínas se trata de balancear los resultados de la

clasificación, cuando se clasifican sitios de *splice sites* se trata de reducir los falsos positivos, y cuando se diagnóstica la HTA se reducen los falsos sanos.

Se muestra para cada aplicación un ejemplo de cómo utilizar un modelo de RB y así mostrar la generalidad en las posibilidades del uso de las mismas.

Teniendo en cuenta la factibilidad de los resultados, se sugiere que los algoritmos propuestos se pueden usar en otros campos de aplicación además de la bioinformática y la medicina.

CONCLUSIONES Y RECOMENDACIONES

Como resultado de esta investigación, se arriba a las siguientes conclusiones:

1. Se propone un algoritmo de aprendizaje estructural de RB basado en árboles de decisión con la técnica CHAID al que se le llamó ByNet. Este algoritmo reporta los mejores resultados en dominios Biomédicos, pero se puede utilizar en dominios de la Bioinformática, pues permite tener en el modelo subgrupos de variables interrelacionadas entre sí y con la variable dependiente o clase.

2. Se presenta un algoritmo de aprendizaje de la estructura de RB al que se le llamó BayesChaid, que utiliza la técnica CHAID, pero no construye árboles de decisión, sino que primero selecciona las variables más relacionadas con la variable dependiente, y a su vez analiza la relación entre las variables predictoras hasta un nivel en profundidad que especifica el usuario, lo cual mejora el resultado anterior.

3. Se presenta un algoritmo para la búsqueda de la estructura de una RB, que denominamos BayesPSO y que se basa en optimización inspirada en modelos de inteligencia colectiva, específicamente la optimización de enjambre de partículas. Este algoritmo garantiza buenas soluciones porque es más exhaustivo en la búsqueda de la estructura

4. Se logra la implementación de los algoritmos propuestos y se añaden a la plataforma de aprendizaje automático *Weka*, donde se incorporan como nuevos clasificadores bayesianos. Se implementaron además en versiones que permiten la paralelización de las validaciones cruzadas. La adición a *Weka* de los nuevos clasificadores propició la validación de los algoritmos propuestos en el contexto de otros reportados en la literatura. Los algoritmos BayesChaid y BayesPSO muestran mejores resultados que el algoritmo ByNet en los problemas resueltos en la tesis. De manera general los tres algoritmos se pueden utilizar en dominios Bioinformáticos, de hecho la semántica de las RB se presta para este tipo de dominio del conocimiento, en el que se tienen grandes volúmenes de datos, caracterizado por datos ruidosos y sujetos a errores. La incertidumbre de estos datos, hace que el uso de las RB resulte apropiado, pues estas ofrecen ventajas en el análisis de este tipo de datos sobre otros métodos estadísticos convencionales y de la IA.

5. Los algoritmos desarrollados se aplicaron satisfactoriamente a la solución de los problemas de análisis de secuencia siguientes: predicción de interacciones de proteínas en la *Arabidopsis thaliana* y clasificación de verdaderos y falsos *donors* y/o *acceptors* en un problema de localización de *splice sites*. Los resultados obtenidos en ambas aplicaciones resultan acordes a los que se obtienen por otras técnicas clásicas de aprendizaje de RB, de estadística y de IA y pueden combinarse con estos para obtener soluciones más plausibles de estos problemas. Los algoritmos propuestos se aplicaron también a un problema de diagnóstico de la HTA, lo que demuestra que los algoritmos se pueden aplicar de modo general para la búsqueda de la estructura de una RB desde datos.

Los resultados obtenidos de ninguna forma agotan el desarrollo ulterior de esta temática. Estos, al igual que los resultados de cualquier desarrollo teórico, constituyen las bases para nuevas líneas de investigación. A continuación se enumeran algunos temas que pudieran ser fuentes de trabajos futuros a manera de recomendaciones:

1. Realizar un análisis de los algoritmos propuestos para determinar si es posible obtener versiones paralelizadas. Ello aumentaría notablemente las posibilidades de aplicación en dominios Bioinformáticos.

2. Incorporar los nuevos clasificadores a los sistemas multiclasificadores que se elaboran en el grupo de Bioinformática con el objetivo de ganar en los resultados de los problemas analizados o en nuevos problemas Bioinformáticos.

3. Analizar la posible aplicación de otros algoritmos bioinspirados, como el de colonia de hormigas o bandadas de insectos a la creación de nuevos métodos para el aprendizaje estructural en RB.

REFERENCIAS BIBLIOGRÁFICAS

Acid, S. y De Campos, L. M. (2003). Searching for Bayesian Network Structures in the Space Restricted Acyclic Partially Directed Graphs. *Artificial Intelligence Research* 18: 445- 490.

Acid, S., De Campos, L. M. y Castellanos, J. G. (2005). Learning Bayesian Network Classifiers: Searching in a Space of Partially Directed Acyclic Graphs. *Machine Learning* 59(**3**): 213 - 235

Arboláez, A. (2008). Extensiones a Weka-Parallel con distintos algoritmos de aprendizaje en redes bayesianas. Aplicaciones Bioinformáticas. *Trabajo de Diploma. Tutores: Chávez, M.C., Casas, G., Departamento Ciencia de la Computación, UCLV, Cuba*.

Armañanzas, R., Inza, I., Santana, R., Saeys, Y., Flores, J. L., Lozano, J. A., Van de Peer, Y., Blanco, R., Robles, V. y Larrañaga, P. (2008). A review of estimation of distribution algorithms in bioinformatics. *BioData Min.* .

Asthana, S., King, O. D., Gibbons, F. D. y Roth, F. P. (2007). Predicting Protein Complex Membership Using Probabilistic Network Reliability. *Cold Spring Harbor Laboratory Press*.

Asuncion, A. y Newman, D. J. (2007). UCI Machine Learning Repository. http://www.ics.uci.edu/~mlearn\mlrepository.htm.

Aytug, H. (2000). Decision Tree Induction. *University of Florida*.

Baldi, P. y Soren, B. (2001). Bioinformatics: The machine learning approach. *2nd Ed., Massachusetts Institute of Technology: MIT Press*: 365-369.

Beielstein, T., Parsopoulos, K. E. y Vrahatis, M. N. (2002). Tuning PSO parameters through sensitivity analysis. *Technical Report of the Collaborative Research Center, University of Dortmund*: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.6598.

Benet, M., Apollinaire, J. J. y Peraza, S. (2003). Cardiovascular Risk Factors among Individuals under Age 40 with Normal Blood Pressure. *Revista Española de Salud Pública* 77(**1**): 143-150.

Benet, M., Yanes, A. J., González, J., Apollinaire, A. y García, J. (2001). Criterios diagnósticos de la prueba del peso sostenido en la detección de pacientes con hipertensión arterial. *Medicina Clinica* 116(**17**): 645-648.

Benson, D. A., Karsch-Mizrachi, I., J., L. D., Ostell, O. y Wheeler, D. L. (2005). GenBank. *Nucleic Acids Research* 33.

Billingsley, P. (1995). Probability and Measure., 3ra. Edic. John Wiley and Sons, New York.

Bouckaert, R. R. (1995). Bayesian Belief Networks: From Construction to Inference. *Promotor: Prof. Dr. J. Van Leeuwen, Co-promotor: Dr. L.C. Van der Gaag, Faculteit Wiskunde en Informatica, Utrecht University*.

Bouckaert, R. R. (2007). Bayesian Network Classifiers in Weka for Version 3-5-7. http://www.cs.waikato.ac.nz/~remco/weka_bn/.

Brender, J., Talmon, J., Egmont-Petersen, M. y McNair, P. (1994). Measuring quality of medical knowledge. *Medical Informatics in Europe* Lisbon.

Buntine, W. L. (1994). Operation for Learning with Graphical Models. *Artificial Intelligence Research* 2: 159- 225

Buntine, W. L. (1995). Graphical Models for Discovery Knowledge and Data Mining. *International Conference on Discovery Science, LNCS* 2534(**5**): 2-11.

Buntine, W. L. (1996). A guide to literarture on learning graphical models. *IEEE Transactions and Knowledge Data Engeniering. 8, 195- 210*.

Caballero, Y. (2007). Aplicación de la Teoría de los conjuntos aproximados en el preprocesamiento de los conjuntos de entrenamiento para los algoritmos de aprendizaje automatizado. *Tesis en opción del grado de Doctor en Ciencias Técnicas, Universidad Central "Marta Abreu" de Las Villas, Cuba* Tutor: Dr. Rafael Bello.

Cai, D., Delcher, A., Kao, B. y Kasif, F. (2000). Modeling splice sites with Bayes network. *Bioinformatics* 16(**2**): 152- 158.

Castillo, E., Gutiérrez, J. M. y Hadi, A. S. (1997). Expert Systems and Probabilistic Network Models,, Springer-Verlag, New York.

Charles River Analytics, I. (2004). About Bayesian Belief Networks. *Charles River Laboratories International Inc.*: http://www.cra.com/commercial-solutions/belief-network-modeling.asp.

Chávez, M. C. y Rodríguez, L. O. (2002). Bayshell, Software para crear redes Bayesianas e inferir evidencias en la misma. *Copyright*

Chávez, M. C., Grau, R. y García, M. M. (1999). Un método para construir Redes Bayesianas. *Revista Facultad de Ingenieria* 19: 76-84

Chávez, M. C., Grau, R. y Sánchez, R. (2005). Construcción de árboles filogenéticos a partir de secuencias de ADN y su integración en una red bayesiana. *Memorias de la XI Convención Expo Internacional de Informática, INFORMÁTICA 2005, La Habana, Cuba* ISBN: 978-959-716-487-6.

Chávez, M. C., Casas, G., González, E. y Grau, R. (2007a). BYNET Herramienta computacional para aprendizaje e inferencias de redes bayesianas en aplicaciones Bioinformáticas. *Memorias de la XII Convención y Expo Internacional de Informática, INFORMÁTICA 2007, La Habana, Cuba, ISBN:978-959-286-002-5.*

Chávez, M. C., Casas, G., Bello, R. y Grau, R. (2008a). Modelo de red bayesiana para predicción de mutaciones en secuencias de la transcriptasa inversa del VIH usando PSO. *Memorias de XIV Congreso Latino-Iberoamericano en Investigación de Operaciones (CLAIO 2008)*: http://socio.org.co/CLAIO2008.

Chávez, M. C., Casas, G., Moya, I. y Grau, R. (2008b). A new Method for Learning Bayesian Networks. Application to Data Splice site Classification. *Proceedings of the Second Workshop on Bioinformatics Cuba Flanders IWOBI 2008, Santa Clara, Cuba, 2008* ISBN:978-959-250-394-6.

Chávez, M. C., Casas, G., Falcón, R., Moreira, J. L. y Grau, R. (2007b). Building Fine Bayesian Networks Aided by PSO-based Feature Selection. IN: ALEXANDER, G., MORALES, K. & FERNANDO, A. (Eds.) *MICAI 2007, LNAI* 4827: 441- 451.

Chávez, M. C., Silveira, P., Casas, G., Grau, R. y Bello, R. (2007c). Aprendizaje estructural de redes bayesianas utilizando PSO. *Memorias en Boletín de la Sociedad Cubana de Matemática, Trabajo IA7, Número Especial en CD de COMPUMAT, Holguín, Cuba* 5.

Chávez, M. C., Casas, G., Moreira, J., González, E., Bello, R. y Grau, R. (2008c). Uso de redes bayesianas obtenidas mediante Optimización de Enjambre de Partículas para

el diagnóstico de la Hipertensión Arterial. *Octavo Congreso Internacional de Investigación de Operaciones, Revista Investigación Operacional* 30(**1**): 52-59.

Chávez, M. C., Casas, G., Moreira, J., Silveira, P., Moya, I., Bello, R. y Grau, R. (2008d). Predicción de mutaciones en secuencias de la proteína transcriptasa inversa del VIH usando nuevos métodos para Aprendizaje Estructural de Redes Bayesianas. *Revista Avances en Sistemas e Informática* 4(2): 77-85.

Chickering, D. M. (1996). Learning Bayesian networks is NP-complete. . *Learning from Data: Artificial Intelligence and Statistics, Springer-Verlag, University of California at Los Angeles*: 121-130.

Chow, C. y Liu, C. (1968). Approximating discrete probability distribution with dependence trees. *IEEE Transactions on Information Theory* 114: 462- 467.

Christos, A. O. y Valencia, A. (2003). Early bioinformatics: the birth of a discipline - a personal view. *Bioinformatics* 19(**17**): 2176- 2190.

Chuang, J. S. y Roth, D. (2001). Splice Site Prediction Using a Sparse Network of Winnows. *Technical Report, University of Illinois, Urbana-Champaign, USA* http://portal.acm.org/citation.cfm?id=871219.

Cohen, J. (2004). Bioinformatics - An Introduction for Computer Scientists. *ACM Computing Surveys* 36(**2**): 122- 158.

Consortium, I. H. G. S. (2004). Finishing the euchromatic sequence of the human genome. *International Human Genome Sequencing Consortium, Nature* 431 (**7011**): 931-45.

Cooper, G. F. (1990). Computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence* 42: 393-405.

Cooper, G. F. y Herskovits, E. H. (1992). A Bayesian methods for the induction of probabilistic networks from data. *Machine Learning* 9(**4**): 309- 348.

Correa, E. S., Freitas, A. A. y Johnson, C. G. (2007). Particle Swarm and Bayesian Networks Applied to Attribute Selection for Protein Functional Classification. *Proceedings of the GECCO: Conference companion on Genetic and evolutionary computation, N.Y., USA* 2651-2658

Cui, J., Li, P., Li, G., Xu, F., Zhao, C., Li, Y., Yang, Z., Wang, G., Yu, Q., Li, Y. y Shi, T. (2007). AtPID: Arabidopsis thaliana protein interactome database-an integrative platform for plant systems biology. *Nucleic Acids Research*: 1-10.

Daalen, V. C. (1992). Evaluating Medical Knowledge Based Systems. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* 3: 888-889.

Davis, L. (1991). Handbook of Genetics Algorithms. *Van Nostrand Reinhold Company, New York* II: 100 páginas

Degroeve, S., De Baets, B., Van de Peer, Y. y Rouzé, P. (2002). Feature subset selection for splice site prediction. *Bioinformatics* 18(**2**): 75- 83.

DeGroot, M. H. (1987). Probability and Statistics. *3rd Edition, Addison-Wesley*.

Dillon, W. y Goldstein, M. (1984). Multivariate Analysis. Methods and Aplications. *John Wiley & Sons*.

Doldán, F. (2007). Redes Bayesianas y Riesgo Operacional. *Revista Gallega de Economía* 16 (**Número extraordinario**): http://www.usc.es/econo/RGE/Vol16_ex/Castelan/art1c.pdf.

Donald, M., Spiegelhalter, C., Taylor y, J. y Campbell, E. (1994). Machine learning, neural and statistical classification *Ellis Horwood Limited*: 289 páginas.

Dopazo, J. y Valencia, A. (2002). Bioinformática y Genómica. *Genómica y mejora vegetal*: 147-198

Dorigo, M. y Stützle, T. (2002). The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. *Handbook of Metaheuristics*: 250-285.

Dorigo, M. y Stützle, T. (2004). Ant Colony Optimization. *MIT Press*: 324 páginas.

Dorigo, M., Birattari, M. y Stützle, T. (2006). Ant Colony Optimization-- Artificial Ants as a Computational Intelligence Technique. *IEEE Computational Intelligence Magazine* 1(**4**): 28 - 39.

Dorigo, M., Stützle, T. (2007). An Introduction to Ant Colony Optimization. *In T. F. Gonzalez, editor, Handbook of Approximation Algorithms and Metaheuristics, CRC Press* 26(**14**): 1 -26.

Duda, R. O. y Hart, P. E. (1973). Pattern Classification and scene analysis. *Jonh Wiley Sons*.

Durrett, R. (1991). Probability: Theory and Examples. *Wadsworth, Pacific Grove, CA.*

EBI (1999). The European Bioinformatics Institute http://www.ebi.ac.uk.

Efron, B. y Tibshirani, R. J. (1997). Improvements on cross-validation: The bootstrap method. *J. Am. Stat. Assoc.* 92: 548-560

Eitrich, T., Kless, A., Druska, C., Meyer, W. y Grotendorst, J. (2007). Classification of Highly Unbalanced CYP450 Data of Drugs Using Cost Sensitive Machine Learning Techniques. *American Chemical Society: J. Chem. Inf. Model* 47(**1**): 92-103.

El-Hay, T. (2001). Efficient Methods for exact and aproximate inference in discrete Graphicals Models. *Master of Science Thesis, Supervisor Nir Friedman*: 17-18.

EMBL (2009). Base de datos de secuencias nucleotídicas. http://www.ebi.ec.uk/embl/index.html.

Escofier, B. y Pages, J. (1992). Análisis Factoriales Simples y Múltiples. *Universidad del País Vasco. Bilbao*.

Fawcett, T. (2004). ROC Graphs: Notes and Practical Considerations for Researchers. *Machine Learning*: http://citeseer.ist.psu.edu/fawcett04roc.html.

Ferat, S., Yavuz, M. C., Arnavut, Z. y Uluyol, O. (2007). Fault diagnosis for airplane engines using Bayesian networks and distributed particle swarm optimization. *Parallel Computing, Elsevier* 33: 124–143.

Foley, R. A. y Lewin, R. (2004). Principles of Human Evolution. *Segunda edición, Backwell publishing, Review from Times Higher Education Supplement, University of Durham*.

Friedman, N. (2004). Infering Cellular Networks Using Probabilistic Graphical Models. *Mathematic Biology* 303(**5659**): 799-805.

Friedman, N. y Goldszmidt, M. (1996). Building Classifiers using Bayesian Networks. *Proceedings of Thirteen National Conference on Artificial Intelligence* 2: 1277-1284.

Friedman, N., Geiger, D. y Goldszmidt, M. (1997a). Bayesian Network Classifiers. *Mach. Learn.* 29(**2-3**): 131-163.

Friedman, N., Goldszmidt, M., Heckerman, D. y Russell, S. (1997b). Challenge: Where is the impact of Bayesian networks in learning? . *Proceedings of the Fifteenth International Joint Conference on Artifcial Intelligence* 1: 10 -15.

Fu, W. J. y Carroll, R. J. (2005). Estimating misclassification error with small samples via bootstrap cross-validation. *Bioinformatics* 21(**7**): 3301.

Galperin, M. Y. (2007). The Molecular Biology Database Collection 2007 update. *Nucleic Acids Res.* 35: D3-D4.

García, L. (1990). Probabilidad e Inteligencia Artificial. *Conferencias de Laureano García, Universidad de la Habana, Cuba*.

Gibas, C. y Per, J. (2001). Developing Bioinformatics Computer Skills. *O'Reilly & Associates* 6641: 448 páginas.

Gilbert, D. (2004). Bioinformatics software resources. *Briefings in Bioinformatics* 5(**3**): 300-304.

Grau, R., Correa, C. y Rojas, M. (2004). Metodología de la Investigación *Segunda Edición, EL POIRA Editores S.A., Ibagué, Colombia, ISBN: 958-8028-10-8*.

Grau, R., Galpert, D., Chávez, M. C., Sánchez, R., Casas, G. y Morgado, E. (2006). Algunas aplicaciones de la estructura booleana del Código Genético. *Revista Cubana de Ciencias Informáticas* 1(**1**): 94-109.

Grau, R., Chávez, M. C., Sánchez, R., Morgado, E., Casas, G. y Bonet, I. (2007a). Boolean algebraic structures of the genetic code. Possibilities of applications. *Lecture Notes on Bioinformatics, Knowledge Discovery and Emergent Complexity in Bioinformatics* 4366: 10-21.

Grau, R., Chávez, M. C., Sánchez, R., Morgado, E., Casas, G. y Bonet, I. (2007b). Boolean algebraic structures of the genetic code. Possibilities of applications. IN: TUYLS, K. et al. (Eds.). *KDEB 2006, LNBI* 4366: 10–21.

Guo, H. y Viktor, H. L. (2007). Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach. *SIGKDD Explorations* 6(**1**): 30 - 39.

Gutiérrez, I. (2003). Un Modelo para la Toma de Decisiones usando Razonamiento Basado en Casos en condiciones de Incertidumbre. *Tesis en opción del grado de Doctor en Ciencias Técnicas, Universidad Central "Marta Abreu" de Las Villas, Cuba* Tutor: Dr. Rafael Bello.

Harley, C. y Reynolds, R. (1987). Analysis of E. Coli Promoter Sequences. *Nucleic Acids Res.* 15: 2343-2361.

Headquarters, C. (2007). Visual Paradigm for UML 6.0. http://www.visual-paradigm.com

Heckerman, D. (1996). A Tutorial on Learning With Bayesian Networks. *Microsoft Research Tech. Report MSR-TR-95- 06, Redmond, WA*: ftp://ftp.research.microsoft.com/pub/dtg/david/tutorial.ps.

Heckerman, D. (1997). Bayesian networks for data mining. *Data Mining and Knowledge Discovery* 1: 79–119.

Hernández, A. G. (2004). Aprendizaje Automático: Árboles de Decisión.

Hernandis, J. A. (2005). Visual Paradigm for UML (VP- UML) 6.0.

Hogg, R. V. (1993). Probability and Statistical Inference. *Maxwell Macmillan International, New York*.

Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N. J., Chung, S., Emili, A., Snyder, M., Greenblatt, J. F. y Gerstein, M. (2003). A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data. *American Association for the Advancement of Science, Washington, USA* 302(**5644**): 449-453.

Jensen, F. V. (2001). Bayesian Network and Decision Graphs. *Springer-Verlag, Nueva York*.

Jensen, F. V. y Nielsen, T. D. (2007). Bayesian Networks and Decisions Graphs. *Information Science and Statistics Series, Springer Verlag, New York* segunda edición: 294 páginas.

Jeroen, H. H., Donkers, L. M. y Tuyls, K. (2008). Belief Networks for Bioinformatics. *Computational Intelligence in Bioinformatics, Springer Berlin / Heidelberg*: 75-111.

Jobson, J. D. (1992). Applied Multivariate Data Analysis *Categorical and Multivariate Methods, Springer, New York* Vol. II: 11-54.

John, G., Kohavi, R. y Pfleger, K. (1994). Irrelevant features and the subset selection problem. *In Machine Learning: Proceeding of Eleventh International Conference, Morgan Kaufman*: 121- 129.

KDnuggets (2008). Bayesian Networks and Bayesian Classifier Software. http://www.kdnuggets.com/software/bayesian.html.

Kenley, C. R. (1986). Infuence Diagram Models with Continuous Variables. Ph.D. Thesis: http://www.kenley.org/Kenley1986.pdf.

Kennedy, J. (1997). The particle swarm: social adaptation of knowledge. *IEEE International Conference on Evolutionary Computation, April 13–16*: 303–308.

Kennedy, J. y Eberhart, R. C. (1995a). Particle swarm optimization. *In: Proceedings of IEEE International Conference on Neural Networks, Perth*: 1942–1948.

Kennedy, J. y Eberhart, R. C. (1995b). A new optimizer using particle swarm theory. *In: Sixth International Symposium on Micro Machine and Human Science. Nagoya*: 39–43.

Kennedy, J. y Spears, W. M. (1998). Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. *Proceedings of the IEEE International Conference on Evolutionary Computation*: 39- 43.

Kennedy, J., Eberhart, R. C. y Y., S. (2001). Swarm Intelligence. *Morgan Kaufmann Series in Artificial Intelligence*: 510 páginas.

Kjærulff, U. B. y Madsen, A. L. (2008). Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis. *Springer Verlag, Series: Information Science and Statistics , New York* XVIII 318 páginas.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *14th International Joint Conference on Artificial Intelligence (IJCAI)*: 1137-1145.

Lanzi, P. (2006). Feature Subset Selection Using Effective Combine of Filter and Wrapper Approaches. Tesis de Grado: 139 páginas.

Larrañaga, P. (2000). Aprendizaje automatico de Modelos Graficos II. Aplicaciones a la Clasificación Supervisada. *Sistemas expertos probabilísticos* 141-162.

Larrañaga, P., Inza, I. y Moujahid, A. (2003). Modelos Probabilísticos para la Inteligencia Artificial y la Minería de Datos: Selección de Variables. *Curso de Doctorado*.

Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armañanzas, R., Santafé, G., Pérez, A. y Robles, V. (2005). Machine learning in bioinformatics. *Briefings in Bioinformatics* 7(**1**): 86-112.

Lauritzen, S. L. y Spiegelhalter, D. J. (1988). Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *J. R. Stat. Soc. B* 157–224.

Lebart, M. (1998). Statistique Exploratoire Multidimensionnelle. *Dunod. París*.

Li, T., Zhang, C. y Ogihara, M. (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. 20: 2429 - 2437.

Liu, J. S. y Logvinenco, T. (2003). Bayesian methods in Biological sequences analysis. *In D.J. Balding, M. Bishop, C., Cannings editors, Handbook of Statistical Genetics, Wiley, New York* chapter 3(**second edition**).

Long, J. L., Xia, Y., Paccanaro, A., Yu, H. y Gerstein, M. (2005). Assessing the limits of genomic data integration for predicting protein networks. *Genome Res.* 15: 945-953.

Lu, L. J., Xia, Y., Paccanaro, A., Yu, H. y Gerstein, M. (2005). Assessing the limits of genomic data integration for predicting protein networks. *Genome Res.* 15: 945-953.

Madsen, A. L. y Jensen, F. V. (1999). Lazy propagation: A junction tree inference algorithm based or lazy evaluation. *Artificial Intelligence* 113(**1-2**): 203- 245.

Madsen, A. L., Jensen, F., Kjærulff, U. y Lang, M. (2005). The HUGIN tool for probabilistic graphical models. *International Journal of Artifcial Intelligence Tools* 14(**3**): 507-543.

Mahamed, G. H. O., Engelbrecht, A. P. y Salman , A. (2005). Dynamic Clustering using PSO with Application in Unsupervised Image Classification. *Proc. 5th World Enformatika Conf. (ICCI), Transactions on Engineering, Computing and Technology* 9: http://cie.szu.edu.cn/dsp/research/areas/T08/papers/Clustering/.

Mahdavi, M. A. y Lin, Y. (2007). False positive reduction in protein-protein interaction predictions using gene ontology annotations. *BMC Bioinformatics* 8(**262**): 1471-2105.

Medina, D. (2007). Redes Bayesianas y Mapas Conceptuales en la elaboración de Sistemas de Enseñanza-Aprendizaje Inteligentes. *Tesis de Maestría en Ciencia de la Computación, UCLV, Santa Clara, Cuba Tutor: Dra. Zenaida García, Consultante: Chávez, M.C.*

Medina, D., Martínez, N., García, Z., Chávez, M. C. y García, M. M. (2007). Putting Artificial Intelligence Techniques into a Concept Map to Build Educational Tools. *IWINAC 2007, Springer-Verlag Berlin Heidelberg* Part II(**LNCS 4528**): 617–627.

Morales, E. (2006). "Aprendizaje Bayesiano."

Morell, C., Rodríguez, Y., Matías, H. y Araujo, L. I. (2006). Una metodología para extender el ambiente de aprendizaje automatizado WEKA. *Monografía publicada en Biblioteca Samuel Feijó, Santa Clara, UCLV, Cuba*.

Murphy, K. (2005). Software Packages for Graphical Models / Bayesian Networks http://http.cs.berkeley.edu/~murphyk/Bayes/bnsoft.html.

Neapolitan, R. E. (1990). Probabilistic Reasoning in Expert Systems: Theory and Algorithms *Wiley-Interscience, New York*: 433 páginas.

Ochoa, A., Mühlenbein, H. y Soto, M. (2000). A Factorized Distribution Algorithm Using Single Connected Bayesian Networks *LNCS* 1917, Springer Berlin / Heidelberg: 787-796.

Ochoa, A., Höns, R., Soto, M. y Mühlenbein, H. (2003). A Maximum Entropy Approach to Sampling in EDA - The Single Connected Case. *LNCS* 2905, Springer Berlin / Heidelberg: 683-690.

Ordúñez, P., Silva, L. C., Paz, M. y Robles, S. (2001). Prevalence estimates for hypertension in Latin America and the Caribbean: are they useful for surveillance? *Panamerican Journal of Public Health* 10(**4**): 226-231.

Parzen, E. (1960). Modern Probability Theory and its Applications. *La Habana. Instituto Cubano del Libro*.

Pazani, M. J. (1996). Searching for dependences in Bayesian classifiers. *Learning from data: Artificial Intelligence. Proceeding of the Twelft Conference, Horvitz, E. Jensen, F. (eds), Morgan Kaufman*: 414-419.

Pe'er, D., Regev, A., Elidan, G. y Friedman, N. (2001). Inferring Subnetworks Expression Profiles. *Bioinformatics* 1(**1**): 1-9.

Pearl, J. L. (1988). Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann, San Francisco.

Pearl, J. L. (1993). Graphical Models, Causality and Intervention. *Stat. Sci.* 8(**3**): 266- 273.

Peña, D. (2002). Análisis de Datos Multivariantes. *MacGraw Hill*: 556 páginas.

Piñero, P. Y. (2005). Un modelo para el aprendizaje y la clasificación automática basado en técnicas de Softcomputing. *Tesis presentada en opción al grado de Doctor en Ciencias Técnicas, Universidad de Ciencias Informáticas, Cuba* Tutor: Dra. María Matilde García.

Qi, Y. Y., Bar-Joseph, Z. y Klein-Seetharaman, J. (2006). Evaluation of Different Biological Data and Computational Classification Methods for Use in Protein Interaction Prediction. *PROTEINS: Structure, Function, and Bioinformatics, Wiley InterScience* 63: 490–500.

Quinlan, J. R. (1986). Induction of Decision Trees. *Mach. Learn.* 1(**1**): 81-106

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning *Morgan Kaufmann Series in Machine Learning*: 302 páginas.

Rebane, G. y Pearl, J. (1988). The recovery of causal poly- trees from statistical data. *Int. J. Approxi. Reasoning* 2 (3): 341.

Rodríguez, A., Mondeja, Y. y Díaz, Y. (2006). Herramienta computacional para hacer inferencias Bayesianas, aplicaciones a Bioinformática *Trabajo de Diploma, Tutores: Chávez, M.C., Casas, G., Departamento Ciencia de la Computación, UCLV, Cuba*.

Ruiz-Shulcloper, J. (2000). Logical Combinatorial Pattern Recognition.

Ruiz, R. (2006). Heurísticas de selección de atributos para datos de gran dimensionalidad. *Tesis presentada en opción al grado de Doctor en Informática, Universidad de Sevilla, España*.

Saeys, Y. (2004). Feature Selection for Classification of Nucleic Acid Sequences. *PhD Thesis, Promotor: Prof. Dr. Yves Van de Peer, co-promotor: Prof. Dr. ir. Dirk Aeyels, Bioinformatics & Evolutionary Genomics, Ghent University/VIB, Belgium.* .

Sahami, M. (1996). Learning limited dependence Bayesian Classifiers. *In Proceeding of the Second International Conference on Knowledge Discovery and Data Mining*: 335-338.

Sánchez, R. (2006). Regularidades algebraicas del código genético: aplicaciones a la evolución molecular. *Tesis presentada en opción al grado científico de Doctor en Ciencias Biológicas, Universidad de la Habana, Cuba, Tutor: Dr. Ricardo Grau*.

Sánchez, R. y Grau, R. (2005). A genetic code Boolean structure. II. The genetic information system as a Boolean information system. *Bull. Math. Biol.* 67(**5**): 1017-1029.

Sánchez, R., Grau, R. y Morgado, E. (2004). Genetic code boolean algebras. *WSEAS transactions on Biology and Biomedicine* 1: 190-197.

Saucier, R. (2000). Computer Generation of Statistical Distributions. *Report of Army Research Laboratory paper ARL-TR-2168*.

Schachter, R. D. (1990). Evidence absorption and propagation through arc reversals. *Uncertainty in Artficial Intelligence, Elsevier Science Publishers B. V. (North-Holland)! Amsterdam*: 173-190.

Schachter, R. D., Anderson, S. K.,Szolovits, P. (1994). Global Conditioning for Probabilistic Inference in Belief Networks. *In Proceedings of the Uncertainty in AI Conference, San Francisco, CA, Morgan Kaufman*: 514–522.

Scott, M. S. y Barton, G. J. (2007). Probabilistic prediction and ranking of human protein-protein interactions. *BMC Bioinformatics* 8: 239-260

Shenoy, P. P. (1992). Valuation-based systems for Bayesian decision analysis. *Operation Research* 40(**3**): 463-484.

Shi, Y. y Eberhart, R. (1998). Parameter Selection in Particle Swarm Optimization. *In Proceedings of the Seventh Annual Conference on Evolutionary Programming*: 591-601.

Siegel, S. (1987). Diseño Experimental no paramétrico. *Edic. Rev.*: 346 páginas.

Silva, L. C. (1997). Cultura estadística e investigación científica en el campo de la salud: una mirada crítica. *Ediciones Díaz de Santos, S.A. Juan Bravo, 3A. 28006 MADRID España*: 416 páginas.

Silva, L. C. (2009). La investigación biomédica y sus laberintos: en defensa de la racionalidad para la ciencia del Siglo XXI *Rústica Hilo*: 499 páginas.

Silva, L. C. y Muñoz, A. (2000). Debate sobre métodos frecuentistas vs bayesianos. *Gaceta Sanitaria* 14(**6**): 482-494.

Spirtes, P. y Meek, C. (1995). Learning Bayesian networks with discrete variables from data. *In Proceeding of the First International Conference on Knowledge Discovery and Data Mining*: 294- 299.

Spirtes, P., Glaymour, C., Sheines, R. (1993). Causation, Prediction and Search *Springer Verlag, New York*.

SPSS_Inc (1994). CHAID para SPSS sobre Windows. Técnicas de segmentación basadas en razones de verosimilitud Chi-cuadrado, Release 6.0. *User Manual* Chicago http://e-spacio.uned.es/fez/eserv.php?pid=bibliuned:Empiria-1998-DB19A741-F905-77F0-77D0-D0DF22E2872F&dsID=PDF.

Stuart, J. R. y Norvig, N. (1996). Inteligencia Artificial: Un enfoque Moderno. *Prentice Hall, Englewood Cliffs, N.J.*

Stuart, J. R. y Norvig, N. (2003). Artificial Intelligence: A Modern Approach. *Prentice Hall; 2 edition*: 1132 páginas.

Towell, G., Shavlik, J. y Noordewier, M. (1990). Refinement of Approximate Domain Theories by Knowledge-Based Artificial Neural Networks. *In Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*.

Troyanskaya, O. G., Dolinski, K., Owen, A. B., Altman, R. B. y Botstein, D. (2003). A Bayesian framework for combining heterogeneous data sources for gene function

prediction (in Saccharomyces cerevisiae). *Proceedings of the National Academy of Sciences USA* 100(**14**): 8348-8353.

Van Rijsbergen, C. J. (1979). Information Retrieval. *London, Butterworths*.

Varma, S. y Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics* 7(**91**): http://www.biomedcentral.com/1471-2105/7/91.

Wang, X., J., Y., X., T., W., X. y R, J. (2006). Feature Selection Based on Rough Sets and Particle Swarm Optimization. *Pattern Recognition Letter, Elsevier* 28(**4**): 459-471

Wilkinson, D. J. (2007). Bayesian methods in bioinformatics and computational systems biology. *Briefings in Bioinformatics* 8(**2**): 109 -116.

Wiltaker, J. (1990). Graphical Models in Applied Multivariate Statistical. *Wiley Series in Probability & Statistics*: 462 páginas.

Witten, I. H. y Frank, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques.: 525 páginas.

Wu, X., Zhu, L., Guo, J., D., Z. y Lin, K. (2006). Prediction of yeast protein–protein interaction network: insights from the Gene Ontology and annotations. *Nucleic Acids Res.* 34 (**7**): 2137–2150.

Ye, N. (2003). The Handbook of Data Mining. *Lawrence Erlbaum Associates, Publishers Mahwah, New Jersey*: Capítulo 5 y 17.

Zhang, L., Wong, S., King, O. D. y Roth, F. P. (2004). Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC Bioinformatics* 5-38.

Producción Científica del autor sobre el tema de la tesis

Chávez, M.C. et al. Memorias de Informática 2000, Red Bayesiana de pronóstico de trastornos neuropsíquicos leves.

Chávez, M.C. et al. Memorias de Informática 2005, Construcción de árboles filogenéticos a partir de secuencias de ADN y su integración en una red bayesiana. Memorias de la XI Convención Internacional de Informática, INFORMÁTICA 2005, La Habana, Cuba, ISBN: 959-7164-87-6

Chávez, M.C., Cuadrado, S; Casas, G, Martínez, N. (2006). Red bayesiana a partir de factores de riesgo de la hipertensión arterial. Memorias III Simposio Internacional de Hipertensión Arterial 2006. ISBN 925-250-27-4.

Grau, R., **Chávez, M.C**., Sánchez, R., Morgado, E., Casas, G., and Bonet, I. (2006) "Boolean algebraic structures of the genetic code. Possibilities of applications". Knowledge Discovery and Emergent Complexity in Bioinformatics, pp. 10-21, LNBI 4366.

Grau, R., **Chávez, M.C.,** Sánchez, R., Morgado, E., Casas, G., and Bonet, I. (2006) "Boolean algebraic structures of the genetic code. Possibilities of applications". In Proceedings of Knowledge Discovery and Emergent Complexity in Bioinformatics Workshop, KDEBC'2006, (Westra R. and Tuyls K., ed.), pp. 1-12, University of Ghent, Belgium.

Grau R; Galpert D., **Chávez, M.C**., Sánchez, R., Casas, G., Morgado E., (2006) "Algunas aplicaciones de la estructura booleana del Código Genético", Revista Cubana de Ciencias Informáticas, Año 1, Vol 1.

Chávez, M.C., Casas, G., Grau, R., Sánchez, R. "Learning Bayesian Networks from Data Bases a Protein Mutant", Proceedings of First International Workshop on Bioinformatics Cuba-Flanders' 2006, Santa Clara, Feb. 7-10, ISBN:959-250-239-0.

Chávez, M.C. "Aplicaciones de la Inteligencia Artificial en la Bioinformática", 3er Congreso Internacional de Ingeniería en Computación, México, noviembre 2006.

Martínez Sánchez, N; León Espinosa, M; García Valdivia, Z; Ferreira Lorenzo, G; **Chávez M.C**. (2006). Mapas Conceptuales y Redes Bayesianas: Una perspectiva para los Sistemas de Enseñanza Inteligentes. Memorias UCIENCIA 2006. Habana Cuba, ISBN 959-16-0463-7.

Chávez, M.C. et al. Byshell, *Software* de inferencia bayesiana, PREMIO PROVINCIAL DEL XVI FORUM DE CIENCIA Y TÉCNICA OBTENIDO EN EL Año 2006 (Destacado).

Chávez, M.C. Memorias de Informática 2007, BYNET Herramienta computacional para aprendizaje e inferencias de redes bayesianas en aplicaciones Bioinformáticas.

Chávez, M.C. et al. Uso de las redes bayesianas combinado con técnicas estadísticas para el diagnostico de la Hipertensión arterial, CIE 2007. Publicado en Revista Automática, Comunicaciones y Electrónica, XXXVIII (2) pp. 45- 48, 2007.

Medina, D., Martínez, N., García, Z., **Chávez, M. C**. and García, M.M.: Putting Artificial Intelligence Techniques into a Concept Map to Build Educational Tools. IWINAC 2007, Part II, LNCS 4528, pp. 617–627, 2007, Springer-Verlag Berlin Heidelberg.

Medina, D., Martínez, N., García, Z., **Chávez, M.C.**,. Redes Bayesianas y Mapas Conceptuales: Una contribución al modelo del estudiante. CIE 2007.

Chávez, M.C., Casas, G., Moreira, J., Falcon, R., Grau, R.: Building Fine Bayesian Networks Aided by PSO-based Feature, Selection, 6th Mexican International Conference on ARTIFICIAL INTELLIGENCE, MICAI 2007 LNAI.

Medina D; Martínez N, García Z, **Chávez, M.C.** (2007). Using Artificial Intelligence Techniques to Build Adaptative Tutoring Systems. EATIS 2007. ACM Digital Library. Copyright © 2007 by the Association for Computing Machinery, Inc ISBN: 978-1-59593-598-4.

Chávez, M.C., Casas, G., Grau, R., Sánchez, R. "Learning Bayesian Networks from Data Bases a Protein Mutant", Proceedings of First International Workshop on Bioinformatics Cuba-Flanders' 2006, Santa Clara, Feb. 7-10, ISBN:959-250-239-0

Chávez, M.C., Silveira, P., Casas, G., Grau, R., Bello, R.: Aprendizaje estructural de redes bayesianas utilizando PSO. *Memórias de COMPUMAT 2007*

Chávez, M.C. et al. A new Method for Learning Bayesian Networks. Application to Data Splice site Classification, Proceedings of Second Workshop on Bioinformatics Cuba – Flanders, February, 2008.

Chávez, M.C. et al., Uso de redes bayesianas obtenidas mediante Optimización de Enjambre de Partículas para el diagnóstico de la Hipertensión Arterial., Octavo Congreso Internacional de Investigación de Operaciones, Habana y publicado en Revista Investigación Operacional 30 (1) pp. 52-59 (2009).

Chávez, M. C., Casas, G., Bello, R., Grau, R. (2008). "Modelo de red bayesiana para predicción de mutaciones en secuencias de la transcriptasa inversa del VIH usando PSO." Memorias de XIV CONGRESO LATINO-IBEROAMERICANO EN INVESTIGACIÓN DE OPERACIONES (CLAIO). (*9 al 12 de septiembre*)

Chávez, M. C., Casas, G., Moreira, J., Silveira, P., Moya, I., Bello, R., Grau, R. (2008). "Predicción de mutaciones en secuencias de la proteína transcriptasa inversa del VIH usando nuevos métodos para Aprendizaje Estructural de Redes Bayesianas " Revista Avances en Sistemas e Informática **4** (2) pp. 77-85.

Chávez, M. C., Casas, G., Moreira, J., Bello, R., Grau, R. (2009), "Perfeccionamiento de la matriz de confusión que resulta de un clasificador, en dependencia del dominio de aplicación" Memorias de XIII Congreso de Informática ISBN 978-959-486-010-0. Presentación virtual en evento INFOSALUD (VII Congreso Internacional de Informática en la Salud).

Se tiene además el siguiente registro de *software*:

Rodríguez L.O., **Chávez M. C.**, Registro de Software número 09358-9358 del Centro Nacional de Derecho de Autor a favor de: Bayshell, *Software* para crear redes bayesianas e inferir evidencias en la misma, 2002.

ANEXOS

Anexo 1. Conceptos básicos

1. Probabilidades.

El cálculo de probabilidades suministra las reglas apropiadas para cuantificar la incertidumbre y constituye la base para la estadística inductiva o inferencial. Para estudiar con mayor profundidad, se puede consultar cualquiera de los libros clásicos de teoría de la probabilidad y estadística, por ejemplo, (DeGroot 1987), (Durrett 1991), (Hogg 1993), (Billingsley 1995). En este anexo se resumirán sólo algunos conceptos básicos que son utilizados y no son definidos en el texto.

Distribución de Probabilidad

Sea $\{X_1, \ldots, X_n\}$ un conjunto de variables aleatorias discretas y $\{x_1, \ldots, x_n\}$ el conjunto de sus posibles realizaciones. Nótese que las variables aleatorias se denotan con mayúsculas y que sus realizaciones se denotan con minúsculas. Por ejemplo, si X_i es una variable binaria, entonces x_i puede ser 1 ó 0. Los resultados que siguen son también válidos si las variables son continuas, pero en este caso los símbolos de suma deben sustituirse por integrales.

Distribución de Probabilidad Conjunta (DPC): Dado un n+1 – *plus* $(X_1, X_2, \ldots, X_n, Y)$ de variables aleatorias, se llama DPC a la función $F[x_1, x_2, \ldots, x_n, y] = prob[X_i \leq x_i \quad i = 1,\ldots, n, \quad Y \leq y]$. Dicha probabilidad no puede calcularse en términos de las distribuciones individuales de $X_1, X_2, \ldots, X_n, Y$, a menos que haya independencia.

Sea $p(x_1,..., x_n)$ la función de probabilidad conjunta[15] como se describe en A1.1:

$$p(x_1, ..., x_n) = p(X_1 = x_1, ..., X_n = x_n) \qquad \text{(A1.1)}$$

Entonces, la *función de probabilidad marginal* de la i-ésima variable se obtiene mediante la formula:

---

[15] Cuando las variables son discretas, $p(x_1, \ldots, x_n)$ se llama *función de probabilidad*, y cuando las variables son continuas, se llama *función de densidad*. Por simplicidad, nos referiremos a ambas como *función de probabilidad conjunta* de las variables.

$$p\ (x_i) = p\ (X_i = x_i) = \sum_{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n} p\ (x_1, \ldots, x_n) \qquad \text{(A1.2)}$$

La expresión A1.3 se conoce como Teorema de Bayes, en la que $p(x_i)$ se conoce como probabilidad "*a priori*" o *inicial* de x_i, $p(x_i \mid x_1, \ldots, x_k)$ es la probabilidad "*a posteriori*" o *condicional*, $p(x_1, \ldots, x_k \mid x_i)$ se conoce como verosimilitudes (Castillo et al. 1997).

$$p\ (x_i \mid x_1, \ldots, x_k) = \frac{p\ (x_i, x_1, \ldots, x_k)}{\sum_{x_i} p\ (x_i, x_1, \ldots, x_k)} = \frac{p(x_i)\ p\ (x_1, \ldots, x_k \mid x_i)}{\sum_{x_i} p(x_i)\ p\ (x_1, \ldots, x_k \mid x_i)} \qquad \text{(A1.3)}$$

Precisamente en esta teoría matemática desarrollada por el Reverendo Thomas Bayes[16] se basan las RB.

Dependencia e Independencia Condicional

Sean X, Y y Z tres conjuntos disjuntos de variables, entonces X se dice condicionalmente independiente de Y dado Z, si y sólo sí $p(x \mid z, y) = p(x \mid z)$, para todos los valores posibles de x, y, z en X, Y y Z; en otro caso X e Y se dicen condicionalmente dependientes dado Z.

Cuando X e Y son condicionalmente independientes dado Z, se escribe $I(X, Y \mid Z)$. La relación $I(X, Y \mid Z)$ se denomina relación de independencia condicional. Similarmente, cuando X e Y son condicionalmente dependientes dado Z, se escribe $D(X, Y \mid Z)$, que se conoce como una relación de dependencia condicional. A veces se escribe $I(X, Y \mid Z)p$ o $D(X, Y \mid Z)p$ para indicar que la relación se deriva, o es implicada, por el modelo probabilístico asociado a la probabilidad p (la función de probabilidad conjunta).

La definición de independencia condicional lleva en sí la idea de que una vez que es conocida Z, el conocimiento de Y no altera la probabilidad de X. En otras palabras, si Z ya se conoce, el conocimiento de Y no añade información alguna sobre X (Castillo et al. 1997).

---

[16] Fue uno de los seis primeros reverendos protestantes ordenados en Inglaterra. Comenzó como ayudante de su padre. Abandonó los hábitos en 1752. Publicó su teoría en el artículo titulado: *"Easy towards solving a problem in the doctrine of chances"*, publicado por: "*The philosophical Transactions of the Royal Society of London*". Las conclusiones presentadas por él fueron aceptadas por Laplace en una memoria de 1781. Fue elegido miembro de la *Royal Society* en 1742, a pesar de que en aquella época no tenía ninguna publicación en el área de las Matemáticas. De hecho no se publicó nada a su nombre mientras vivió, ya que enviaba sus trabajos de forma anónima.

2. Grafos

Un modelo probabilístico puede definirse usando un grafo que describa las relaciones existentes entre las variables. Supongamos que el conjunto de variables $X= \{X_1, ..., X_n\}$ puede relacionarse entre sí. El conjunto anterior puede representarse gráficamente por una colección de nodos o vértices, asociando un nodo a cada elemento de X. Estos nodos pueden conectarse por arcos, indicando las relaciones existentes entre los mismos. Un arco entre X_i y X_j se denotará mediante L_{ij}. Así mismo, el conjunto de todos los arcos se denotará por $L= \{L_{ij} \mid \ X_i \text{ y } X_j \text{ están conectados}\}$. Por tanto, un grafo se define mediante el conjunto de nodos: X y las relaciones entre los mismos: L. Los términos grafo y red se emplean como sinónimos en este trabajo.

Un grafo es un par de conjuntos $G= (X, L)$ donde $X= \{X_1,..., X_n\}$ es un conjunto finito de elementos (nodos) y L es un conjunto de arcos, es decir, un subconjunto de pares ordenados de elementos distintos de X. Los arcos de un grafo pueden ser dirigidos o no dirigidos, dependiendo de si se considera o no el orden de los nodos.

Grafos dirigidos y no dirigidos, cíclicos y no cíclicos

Un grafo en el que todos los arcos son dirigidos se denomina grafo dirigido. Un grafo en el que todos sus arcos son no dirigidos se denomina no dirigido. Por tanto, en un grafo dirigido es importante el orden del par de nodos que define cada arco, mientras que en un grafo no dirigido, el orden carece de importancia.

Ciclo: Un ciclo es un camino cerrado en un grafo dirigido.

Grafo dirigido cíclico: Un grafo dirigido se denomina cíclico si contiene al menos un ciclo; en caso contrario se denomina grafo dirigido acíclico (GDA).

Arco dirigido: Dado un grafo $G= (X, L)$, si $L_{ij} \in L$ y $L_{ji} \notin L$, el arco L_{ij} entre los nodos X_i y X_j se denomina dirigido y se denota mediante $X_i \rightarrow X_j$.

Arco no dirigido: Dado un grafo $G= (X, L)$, si $L_{ij} \in L$ y $L_{ji} \in L$, el arco L_{ij} entre los nodos X_i y X_j se denomina no dirigido y se denota mediante $X_i - X_j$ o $X_j - X_i$.

Camino: Un camino del nodo X_i al nodo X_j es un sucesión de nodos $\{X_{i1}, ..., X_{ir}\}$, comenzando en $X_i = X_{i1}$ y finalizando en $X_j = X_{ir}$, de forma que existe un arco del nodo X_{ik} al nodo X_{ik+1}, $k = 1, \ldots, r-1$. La longitud del camino $(r-1)$, se define como el número de arcos que contiene.

Un camino $\{X_{i1}, ..., X_{ir}\}$ se dice que es cerrado si el nodo inicial coincide con el final, es decir, $X_{i1} = X_{ir}$.

Padre de un nodo: Cuando existe un arco dirigido, $X_i \rightarrow X_j$, del nodo X_i al nodo X_j, entonces se dice que el nodo X_i es un padre del nodo X_j, y que el nodo X_j es un hijo de X_i. El conjunto de los padres de un nodo X_i se denota por Pa_i.

Ascendientes de un nodo. Un nodo X_j se denomina ascendiente del nodo X_i si existe un camino de X_j a X_i.

Descendientes de un nodo. Un nodo X_j se denomina descendiente del nodo X_i si existe un camino de X_i a X_j

Grafo moral: El grafo obtenido uniendo primeramente cada par de nodos con hijos comunes en un grafo dirigido y luego se elimina la direccionalidad de las conexiones, se llama grafo moral.

Cuerda: Una cuerda es una conexión entre dos nodos de un lazo que no está contenida en el lazo. Los lazos de longitud tres no pueden contener una cuerda y se llaman triángulos.

Grafo triangulado: Un grafo no dirigido se dice que es triangulado o cordal, si cada lazo de longitud cuatro o más tiene al menos una cuerda.

Subconjunto completo de un grafo: Un subconjunto de nodos S de un grafo G, se dice que es completo si existe una conexión entre cada par de los nodos en S.

Un conjunto completo de nodos C, es **un conglomerado** si es máximo, esto es, no es un subconjunto propio de otro conjunto completo.

Grafo agrupado asociado con un grafo: Dado un grafo $G = (X, L)$ y un conjunto de grupos de nodos de X, $C = \{C_1, ..., Cm\}$, tal que $X = C_1 \cup ... \cup C_m$, entonces el grafo $G' = (C, L')$ se llama grafo agrupado (Acid y De Campos 2003) de G si L' contiene solamente

conexiones entre grupos que contienen nodos comunes, esto es, $(C_i, C_j) \in L' \Rightarrow C_i \cap C_j \neq \varnothing$.

Grafos simples y poliárboles: Un árbol dirigido se denomina un árbol simple si cada nodo tiene como máximo un padre; en caso contrario se denomina un poliárbol.

Nodo de aristas convergentes o cabeza-cabeza: Dado un grafo dirigido y un camino no dirigido (. . . − *U* − *A* − *V* − . . .), el nodo *A* se denomina un nodo de aristas convergentes en este camino si las dos aristas del camino convergen a este nodo en el grafo dirigido, es decir, si el grafo dirigido contiene las aristas U → A y V → A).

Grafo de conglomerados: Un grafo agrupado se llama grafo de conglomerados si sus grupos son los conglomerados del grafo asociado.

Árbol de unión: Un grafo de conglomerados se llama un árbol de unión si es un árbol y si cada nodo que pertenece a dos grupos también pertenece a cada grupo en el camino entre ellos.

Familia de un nodo: El conjunto formado por un nodo y sus padres, se llama la familia del nodo.

Árbol de familias: Un árbol de familias de un grafo dirigido *D*, es un árbol de unión de algún grafo no dirigido *G*, en el cual la familia de cada nodo está contenida al menos en un grupo.

Variables sumidero: Variables sin sucesores que no forman parte de la evidencia. Resultan irrelevantes para el cálculo de las distribuciones a posteriori.

d-separación (Jensen y Nielsen 2007): dos variables distintas *A* y *B* en una red causal están d-separadas (d para grafos dirigidos) si para todos los caminos entre *A* y *B*, hay una variable intermedia *V* (distinta de *A* y *B*) tal que se cumple una de las dos proposiciones siguientes: la conexión es serial o divergente y *V* está instanciada o la conexión es convergente y ni *V* ni ninguno de sus descendientes ha recibido evidencia. Si *A* y *B* no están d -separadas se llaman d-conectadas

Cuando *Z* d-separa *X* e *Y* en *G*, se escribe $I(X, Y \mid Z)_G$ para indicar que la relación de independencia viene dada por el grafo *G*; en caso contrario, se escribe $D(X, Y \mid Z)_G$ para indicar que X e Y son condicionalmente dependientes dado Z en el grafo *G*.

3. Términos biológicos.

La **genómica** es la disciplina que estudia el genoma de los seres vivos, en particular los genes que los componen y sus funciones.

El **genoma** es todo el material genético contenido en los cromosomas de un organismo en particular.

En un gen, la secuencia de los nucleótidos a lo largo de la cadena de ADN define una **proteína**, que un organismo es capaz de sintetizar o "expresar" en uno o varios momentos de su vida, usando la información de dicha secuencia.

La relación entre la secuencia de nucleótidos y la secuencia de aminoácidos de la proteína es determinada por un mecanismo celular de traducción, conocido de forma general como **código genético**. A, T, G, y C son las "letras" del código genético y representan las bases nitrogenadas adenina, timina, guanina y citosina, respectivamente.

En cada gen se combinan las cuatro bases en diversas formas, para crear palabras de tres letras (codón) que especifican qué **aminoácido** es necesario en cada paso de la elaboración de la proteína.

Las alrededor de treinta mil proteínas diferentes en el cuerpo humano están hechas de veinte aminoácidos diferentes, y una molécula de ADN debe especificar la secuencia en que se unan dichos aminoácidos. Aquí se sitúa la **proteómica**, como disciplina que correlaciona las proteínas con sus genes, estudia el conjunto completo de proteínas que se pueden obtener de un genoma.

Anexo 2. Comparación de paquetes de *software* de Modelos Gráficos: RB

Src = si no contiene código fuente incluido, N, sino el lenguaje

API = Si N no se puede integrar a nuestro código, debe ejecutar desde ejecutable

Exec = Sistema operativo: W = Windows (95/98/NT), U = Unix, M = Mac, or - = otro compilador.

Nombre	Autores	Src	API	Exec	Comentarios
AgenaRisk	Agena	N	Y	W,U	Simulación por discretización Dinámica
Analytica	Lumina	N	Y	W,M	Propagación compatible
Banjo	Hartemink	Java	Y	W,U,M	Estructura de aprendizaje dinámica o discreta, redes de variable discreta
Bassist	U. Helsinki	C++	Y	U	Genera C++ para MCMC.
Bayda	U. Helsinki	Java	Y	WUM	Clasificador Naïve bayes
BayesBuilder	Nijman (U. Nijmegen)	N	N	W	-
BayesiaLab	Bayesia Ltd	N	N	-	Aprendizaje estructural, modelos dinámicos
Bayesware Discoverer	Bayesware	N	N	WUM	Usa límite y compresión para aprendizaje con datos ausentes
B-course	U. Helsinki	N	N	WUM	Ejecuta en un servidor
Belief net power constructor	Cheng (U.Alberta)	N	W	W	-
BNT	Murphy (U.C.Berkeley)	Matlab /C	Y	WUM	Maneja modelos dinámicos como Modelos ocultos de Markov y filtros Kalman
BNJ	Hsu (Kansas)	Java	-	-	-
BucketElim	Rish (U.C.Irvine)	C++	Y	WU	-
BUGS	MRC/Imperial College	N	N	WU	-
Business Navigator 5	Data Digest Corp	N	N	W	-
CABeN	Cousins et al. (Wash. U.)	C	Y	WU	-

Causal discoverer	Vanderbilt	N	N	W	Estructura de aprendizaje
CoCo+Xlisp	Badsberg (U. Aalborg)	C/lisp	Y	U	Diseñado para tablas de contingencia
CIspace	Poole et al. (UBC)	Java	N	WU	-
DBNbox	Roberts et al	Matlab	-	-	-
Deal	Bottcher et al	R	-	-	Estructura de aprendizaje
DeriveIt	DeriveIt LLC	N	-	-	Explota estructura local de la distribución de probabilidad conjunta
Ergo	Noetic systems	N	Y	W,M	-
GDAGsim	Wilkinson (U. Newcastle)	C	Y	WUM	Análisis bayesiano de modelos dirigidos gaussianos
Genie	U. Pittsburgh	N	WU	WU	-
GMRFsim	Rue (U. Trondheim)	C	Y	WUM	Análisis bayesiano de modelos no dirigidos gaussianos
GMTk	Bilmes (UW), Zweig (IBM)	N	Y	U	Diseñado para reconocimiento del habla
gR	Lauritzen et al.	R	-	-	Aún no esta en el mercado
Grappa	Green (Bristol)	R	-	-	-
Hugin Expert	Hugin	N	Y	W	-
Hydra	Warnes (U.Wash.)	Java	-	-	-
Ideal	Rockwell	Lisp	Y	WUM	GUI requiere Allegro Lisp.
Java Bayes	Cozman (CMU)	Java	Y	WUM	-
KBaseAI	Codeas	N	Y	W,U	Arquitectura cliente/servidor, múltiples usuarios, control acceso, lenguaje de consultas
LibB	Friedman (Hebrew U)	N	Y	W	Estructura de aprendizaje
MIM	HyperGraph Software	N	N	W	Hasta 52 variables
MSBNx	Microsoft	N	Y	W	-
Netica	Norsys	N	WUM	W	-
Optimal Reinsertion	Moore, Wong (CMU)	N	N	W,U	Estructura de aprendizaje
PMT	Pavlovic (BU)	Matlab /C	-	-	-
PNL	Eruhimov (Intel)	C++	-	-	Versión C++ de BNT

Pulcinella	IRIDIA	Lisp	Y	WUM	Usa sistema de evaluación para calculo no probabilístico
RISO	Dodier (U.Colorado)	Java	Y	WUM	Implementación distribuida
Sam Iam	Darwiche (UCLA)	N	N ?	WU ? (Java ejecutable)	Solo hace análisis de sensitividad
Tetrad	CMU	N	N	WU	-
UnBBayes	?	Java	-	-	Usa K2 para estructura de aprendizaje
Vibes	Winn & Bishop (U. Cambridge)	Java	Y	WU	No disponible
Web Weaver	Xiang (U. Regina)	Java	Y	WUM	-
WinMine	Microsoft	N	N	W	Aprendizaje de la estructura
XBAIES 2.0	Cowell (City U.)	N	N	W	-

Anexo 3. Clasificación de *Software* de Redes Bayesianas y Clasificadores Bayesianos en propietario y libre

***Software* propietario**
AgenaRisk, herramienta visual que combina RB y simulación estadística, libre un mes para evaluación.
Analytica, basado en diagramas de influencia, ambiente visual para crear y analizar modelos probabilisticos. (Win/Mac).
AT-Sigma Data Chopper, para analizar y buscar relaciones causales en bases de datos.
BayesiaLab, herramienta de RB para aprendizaje supervisado y no supervisado, y una herramienta de análisis.
Bayesware Discovery 1.0, herramienta de modelación automática de RB desde datos buscando el modelo más probable.
BNet, incluye BNet.Builder para crear una RB, entrar información y obtener resultados y BNet.EngineKit para incorporar la tecnología RC (Redes de Creencia) a nuestras aplicaciones
DXpress, herramienta sobre Windows para crear y compilar RB.
Ergo™, Editor y resolvedor de RB (Win, Mac, demos disponibles).
Flint, combina RB, factores de certeza, y lógica difusa con un ambiente de programación lógica basado en reglas.
Hugin, colección completa de herramientas de razonamiento en RB.
KnowledgeMiner, usa redes neuronales autoorganizadas para descubrir la estructura del problema (Mac).
Netica, Herramienta de RB (Win 95-NT, demo disponible).
PrecisionTree, una macro de Microsoft Excel para crear árboles y diagramas de influencia.

***Software* Libre**
Bayda 1.0, sistema experto para ecocardiografía. *Bayesian belief network software,* de J. Cheng, incluye un *PowerConstructor*: Sistema eficiente para aprendizaje estructural y paramétrico de RB. Constantemente actualizado desde 1997 y un *PowerPredictor:* Programa de Minería de datos para modelación, clasificación y predicción de datos. *Bayesian Logistic Regression Software,* regresión logística bayesiana a gran escala (Win y Linux). *Bayesian Network tools in Java (BNJ),* colección de código Fuentes de herramientas en java para aprendizaje y razonamiento probabilístico (Universidad del estado de Kansas, KDD Lab.). *FDEP*, induce dependencia funcional desde una entrada de datos. *GeNle,* ambiente de modelos de decisión mediante diagramas de influencia y RB (Win, tiene sobre 2000 usuarios). *JavaBayes*, *software* de edición y uso de RB. *jBNC*, conjunto de programas en Java para entrenamiento, prueba y aplicación de clasificadores de RB. *JNCC, Naïve Credal Classifier 2,* herramienta en java que hace una extensión al Naïve bayes con resultados robustos aún cuando se tengan pequeños conjuntos de datos y/o información incompleta. *MSBN: Microsoft Belief Network Tools,* herramienta para crear y evaluar RC bayesianas (libre para investigaciones no comerciales). *PNL,* librería de código Fuentes de RB. *Pulcinella,* herramienta para propagar incertidumbre basada en cálculos locales *(Lisp).*

Anexo 4. Técnicas y Herramientas de Genómica y Proteómica (Gibas y Per 2001)

¿Qué hacer?	**¿Por qué hacerlo?**	**¿Con qué hacerlo?**
Recursos genómicos en línea	Para encontrar información acerca de la localización y función de determinados genes en un genoma	Herramientas NCBI (*National Center for Biotechnology Information*, *EE.UU.*), Herramientas *TIGR, EnsEMBL*, y bases de datos de determinados genomas
Escala base	Para convertir intensidades fluorescents desde experimentos de secuenciación en letras de las bases nucleotídicas	*Phred*
Mapeo y ensamble de Genomas	Para organizar las secuencias de fragmentos cortos de ADN en algo coherente	*Phrap, Staden package*
Anotación de Genomas	Para conectar información funcional acerca de un genoma en localizaciones específicas de la secuencia	*MAGPIE*
Comparación de Genomas	Para identificar componentes de la estructura de un genoma que diferencia un organismo de otro	*PipMaker*, *MUMmer*
Análisis de imágenes de micro-arreglos	Para identificar y cuantificar sitios en datos de micro-arreglos	*CrazyQuant*, *SpotFinder*, *ArrayViewer*
Análisis de agrupamiento de datos	Para identificar genes que aparecen o se expresan en	*Cluster*, *TreeView*

de micro-arreglos	determinados grupo	
Análisis Páginas - 2D	Para analizar, visualizar y cuantificar imágenes en páginas -2D	*Melanie3, Melanie Viewer*
Análisis Proteómico	Para analizar espectrometría de masa e identificar proteínas	*ExPASy tools, ProteinProspector, PROWL*
Herramientas de caminos metabólicos	Para buscar caminos metabólicos y descubrir relaciones funcionales, para reconstruir caminos metabólicos	*PATH-DB, WIT, KEGG*
Simulación celular y metabólico	Para modelar procesos metabólicos y celulares basados en propiedades conocidas e inferencia	*Gepasi, XPP, Virtual Cell*

Anexo 5. La Prueba Chi-cuadrado y la técnica de CHAID

Suponga que se trabaja con dos variables aleatorias discretas (nominales u ordinales) con las cuales se ha realizado una tabla de contingencia (*m* x *n*), esto es una tabla de doble entrada con las frecuencias de casos con cada par de valores de las variables que se asocian. De acuerdo a la definición de independencia de la Teoría de Probabilidades (Parzen 1960), las dos variables serán independientes si la probabilidad de que un caso quede en una celda dada de la tabla es igual al producto de las probabilidades marginales de las dos categorías que definen la celda. Tal probabilidad define las frecuencias esperadas en una tabla bajo el supuesto de independencia y ello debe manifestarse aproximadamente así con las frecuencias observadas.

Para construir un estadístico que mide la independencia precisamente se calculan las diferencias entre las frecuencias esperadas y las observadas y ello se realiza para cada celda de la tabla. Si las variables son independientes, la probabilidad de que una observación caiga en la celda (*i, j*) se estima por la expresión:

$$P(fila = i \ y \ columna = j) = \frac{cantidad\ en\ fila\ i}{N} * \frac{cantidad\ en\ columna\ j}{N} \qquad \text{(A5.1)}$$

Para obtener la frecuencia esperada E_{ij} se multiplica la probabilidad anterior por el volumen de la muestra según expresión:

$$E_{ij} = \frac{(cantidad\ en\ fila\ i) * (cantidad\ en\ \ columna\ j)}{N} \qquad \text{(A5.2)}$$

Las frecuencias esperadas se comparan con las frecuencias observadas O_{ij} en la tabla.

Las diferencias E_{ij}-O_{ij} se llaman residuales, se elevan al cuadrado para evitar la compensación de diferencias positivas y negativas y se dividen por las frecuencias esperadas E_{ij} para establecer magnitudes relativas. Resulta en el estadístico de la expresión:

$$\chi^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \qquad \text{(A5.3)}$$

Si la hipótesis fundamental de independencia es cierta, este estadístico tiene distribución aproximadamente igual a la Chi-cuadrado, con grados de libertad determinado por el

producto (*m-1*)*(*n-1*), donde *m* y *n* son el número de filas y columnas de la tabla. La idea de los grados de libertad es aquí clara pues este es el número de celdas de la tabla que "*podrían llenarse libremente*" si están fijados los totales marginales de filas y columnas.

El valor del estadístico anterior, conocido como Chi-cuadrado de Pearson, se compara con los valores teóricos de la distribución Chi-cuadrado, lo que determina la significación del valor y por tanto un criterio para rechazar o no la hipótesis de independencia.

La prueba Chi-cuadrado tiene realmente muchas limitaciones y los principales detractores llegan incluso a decir que el único caso en que él puede ser aplicado con fiabilidad, es el caso de las tablas 2x2. Esta restricción ha sido ampliamente discutida, pero en esencia es cierto que las tablas de contingencia no pueden tener dimensiones demasiado grandes pues ello puede redundar en frecuencias esperadas excesivamente bajas que exacerbarían el valor del estadístico χ^2. Si se quiere eliminar frecuencias esperadas bajas, se debe reducir las dimensiones de la tabla aunque esto haga que se pierda información (Jobson 1992).

Algoritmo de detección de Interacciones basado en Chi-cuadrado (CHAID)

El método CHAID surge como una técnica de segmentación. Su propósito es segmentar o dividir una población en dos o más grupos en las categorías del mejor predictor de una variable dependiente. El algoritmo se basa en la prueba Chi-cuadrado para seleccionar la mejor división en cada paso, la división se realiza hasta que no haya más variables predictoras significativas o hasta que se satisfaga algún otro criterio de parada, relacionado por ejemplo con el número mínimo de casos en un nodo para analizar su divisibilidad.

En un estudio real existen frecuentemente múltiples variables (predictivas o independientes) que pueden tener asociación con una variable dependiente y además efectos de interacción entre ellas sobre dicha variable dependiente. La presentación de muchas tablas de contingencia, no siempre refleja las asociaciones esenciales, y usualmente se convierte en un listado inútil de tablas que desinforman en lugar de orientar, aún cuando se utilicen estadísticos (como la *V* de Cramer) para ordenar la fortaleza de las asociaciones. Un estudio multivariado trata de enfocar el efecto posible de todas las variables conjuntamente incluyendo sus posibles correlaciones; pero puede ser particularmente interesante, si considera además la posibilidad de la interacción entre las variables

predictivas sobre la variable dependiente. Cuando el número de variables crece, el conjunto de las posibles interacciones crece en demasía, resulta prácticamente imposible analizarlas todas y por ello adquiere especial interés una técnica de detección automática de interacciones fundamentales. CHAID es exactamente eso (SPSS_Inc 1994).

Un análisis de CHAID automático comienza dividiendo la población total en dos o más subgrupos distintos basado en las categorías del mejor predictor de la variable dependiente (en principio por el estadígrafo Chi-cuadrado de Pearson). Divide cada uno de estos subgrupos en pequeños sub-subgrupos y así sucesivamente. CHAID visualiza los resultados de la segmentación en forma de un diagrama tipo árbol cuyas ramas (nodos) corresponden a los grupos (subgrupos conformados en cada nivel). Entiéndase en este caso que está seleccionando sucesivamente las variables más significativamente asociadas con la clase y las variables que deben ser fuentes de estratificaciones sucesivas.

Algoritmo CHAID

Estado 1.Fusionar (Merging)

Para cada predictor $X_{1, ..., } X_{k, ...} X_N$ CHAID une categorías no significativas por los siguientes pasos:

1. Formar todas las crostabulaciones con la variable dependiente (*a full two-way*).

2. Para cada par de categorías aplicar la prueba Chi-cuadrado para probar dependencia de dos categorías y la variable dependiente (Usar todas las categorías de la variable dependiente).

3. Calcular el p-*value* para cada par. Si hay dos pares no significativos unirlos e ir al paso 4. Si todos los pares se mantienen significativos ir al paso 5.

4. En el caso que se tienen más de dos categorías, probar si es posible aplicar el proceso de dividir categorías a una previamente mezclada. Si el valor del estadístico Chi-cuadrado es significativo dividir la categoría de las demás. Si es posible dividir más de una categoría, dividir la de mayor significación. Retornar al paso 3.

5. Mezclar cualesquiera categorías que tienen menos observaciones que el mínimo tamaño de grupo fijado (después de dividir) con la categoría más similar.

Estado 2. Dividir (Splitting)

Para variables predictoras con p-*value* significativos, dividir el grupo por el predictor de menor p-*value*. Cada una de las categorías mezcladas se convierte en un nuevo subgrupo del grupo padre. Si no hay p-*value* significativo, no dividir el grupo.

Estado 3.Parada (Stopping)

Retornar al estado 1 para analizar el próximo subgrupo que contiene más observaciones que lo especificado por el mínimo tamaño de subgrupo (después de dividir). Parar cuando todos los subgrupos han sido analizados o cuando estos contienen pocos casos.

Anexo 6. Características de las bases de datos del repositorio de la UCIML utilizadas para validar los algoritmos de aprendizaje estructural de Redes Bayesianas

	Base de Datos	*Rasgos discretos*	*Rasgos continuos*	*Clases*	*Casos*	*Distribución por clase*
1	promoters	57	0	2	106	53 53
2	mammographic	4	1	2	961	516 545
3	Lung-cancer	56	0	3	32	9 13 10
4	hepatitis	13	6	2	155	32 123
5	e colic	0	7	8	336	143 77 52 35 20 5 2 2
6	crx	9	6	2	690	307 383
7	breast-cancer-w	10	0	2	683	444 239
8	contac-lenses	4	0	3	24	4 5 15
9	hayes-roth-m	4	0	3	132	51 51 30
10	kr-vs-kp	36	0	2	3196	1669 1527
11	Monk 1	6	0	2	415	229 186
12	vote	16	0	2	300	116 184
13	Balance-scale	4	0	3	625	288 49 288
14	tic-tac-toe	9	0	2	958	626 332
15	iris	0	4	3	150	50 50
16	labor	8	8	2	57	20 37
17	segment-challenge	0	19	7	2310	330 330 330 330 330 330 330
18	soybean	35	0	19	683	-

Anexo 7. Red Bayesiana de clasificación de *donors* con el algoritmo ByNet. Ejemplos de propagación de evidencias con el *software* ELVIRA.

Cuando aún no se tienen evidencias la red se muestra según Figura 7.1.

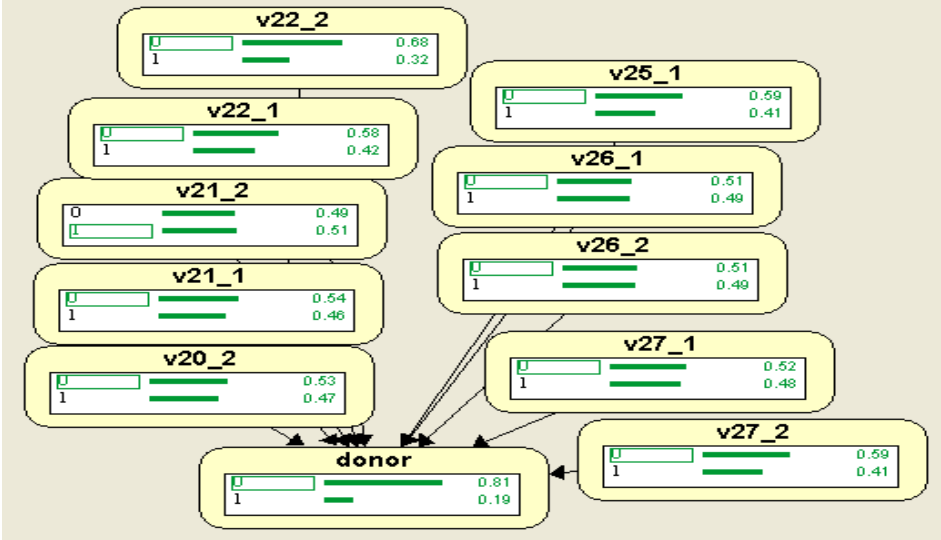


Figura 7.1. RB obtenida con el algoritmo ByNet para *donors* sin evidencias.

Tabla 7.1. Propagación de evidencias de *donors*.

V20	V21	V22	V23	V24	V25	V26	V27	P(*donors* | evidencias)
T o C	A	G	G	T	G o A	A	G	0.94

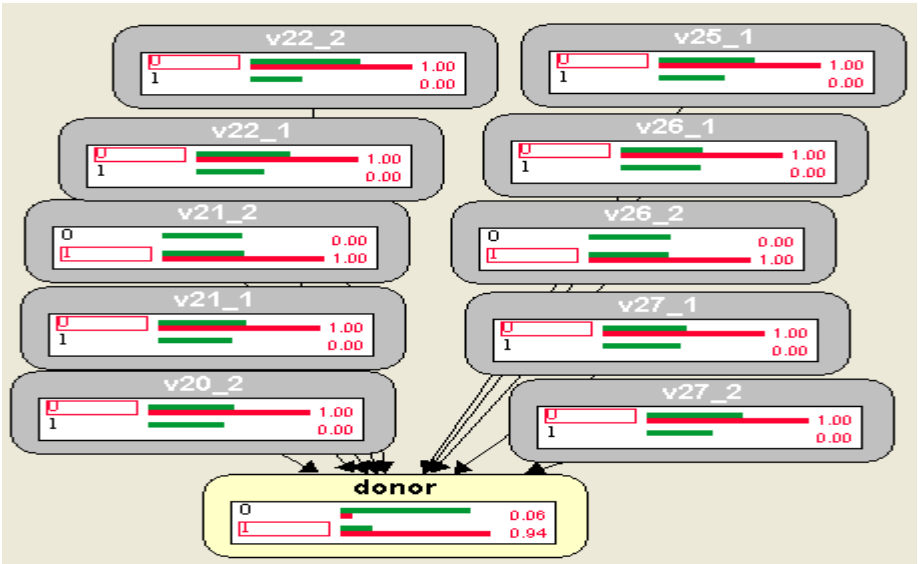


Figura 7.2. RB para *donors* con las evidencias de la Tabla 7.1.

Tabla 7.2. Propagación de evidencias de no presencia de *donors*

V20	V21	V22	V23	V24	V25	V26	V27	P(no *donors* | evidencias)
			G	T	T o C	C	C	0.91

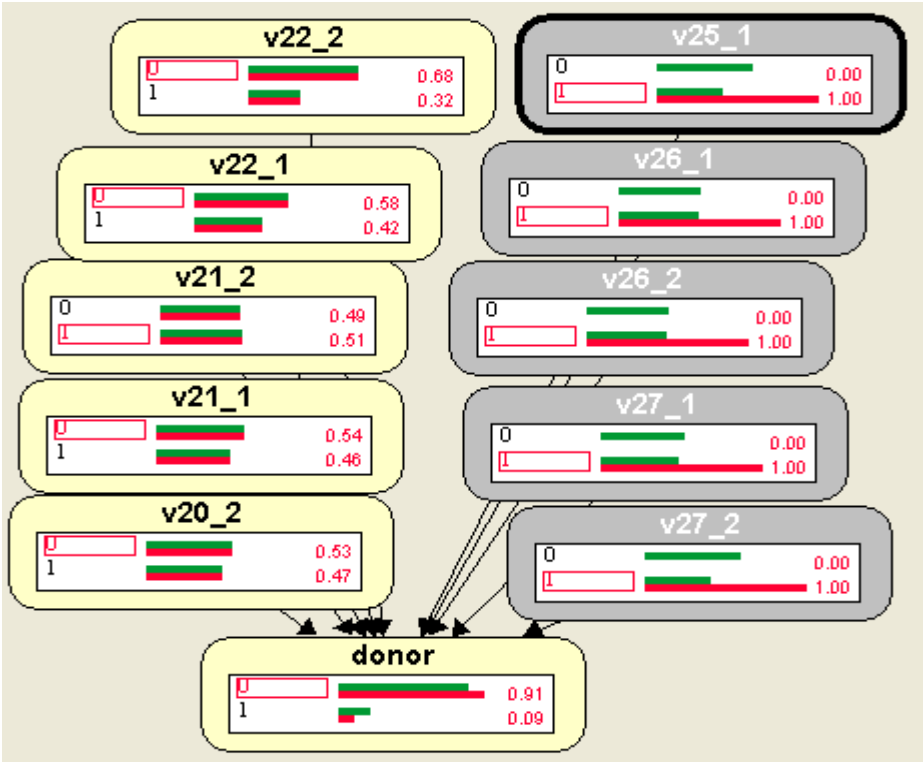


Figura 7.3. RB para las evidencias de la Tabla 7.2

Anexo 8. Red Bayesiana de clasificación de *acceptors* con el algoritmo ByNet. Ejemplos de propagación de evidencias con el *software* ELVIRA.

Cuando aún no se tienen evidencias la red se muestra como en la Figura 8.1.

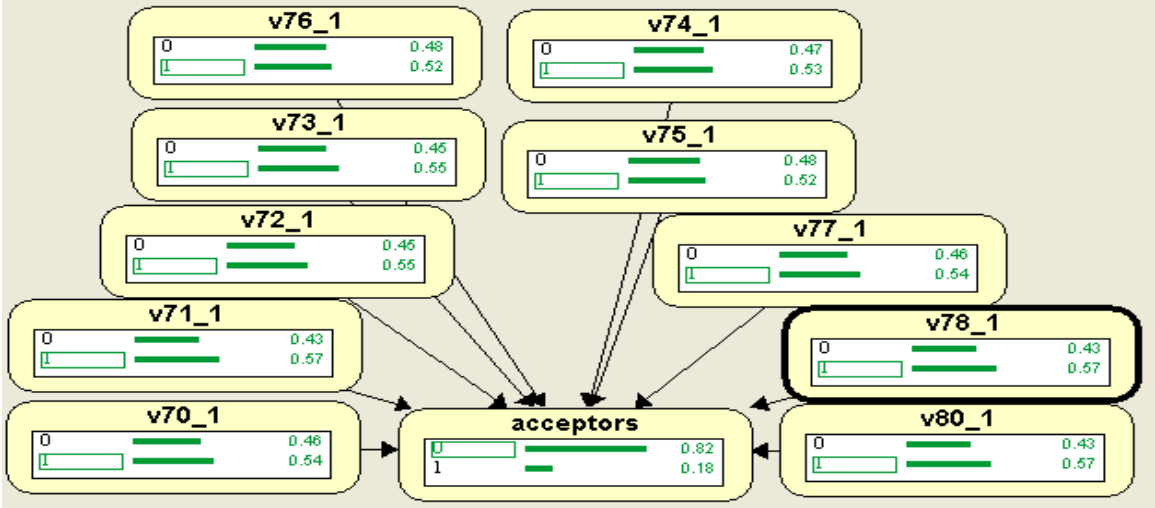


Figura 8.1. RB obtenida con el algoritmo ByNet para *acceptors* sin evidencias

Tabla 8.1. Propagación de evidencias de *acceptors*

V70	V71	V72	V73	V74	V75	V76	V77	V78	V79	V80	P(*acceptors* | evidencias)
T o C	T o C	T o C	T o C	T o C	T o C	T o C	T o C	T o C		T o C	0.82

Tabla 8.2. Propagación de evidencias de no *acceptors*

V70	V71	V72	V73	V74	V75	V76	V77	V78	V79	V80	P(no *acceptors* | evidencias)
				A o G	A o G	A o G	A o G	A o G	A o G	A o G	0.96

Tabla 8.3. Propagación de evidencias de no *acceptors*

V70	V71	V72	V73	V74	V75	V76	V77	V78	V79	V80	P(no *acceptors* | evidencias)
A o G	A o G	A o G	A o G	A o G	A o G	A o G	A o G	A o G	A o G	A o G	0.99

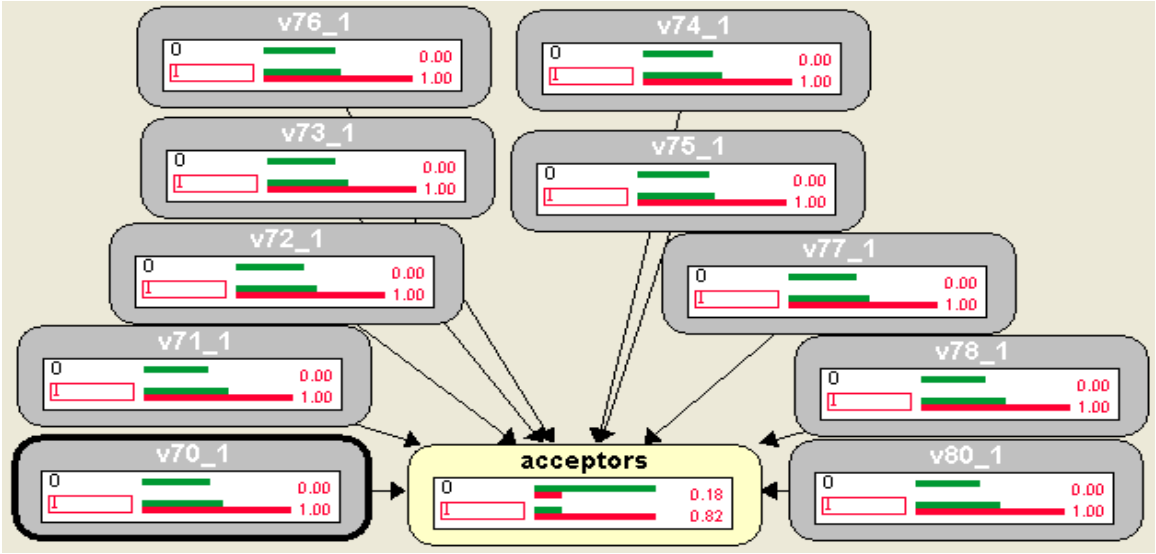


Figura 8.2. RB para las evidencias de la Tabla 8.1.

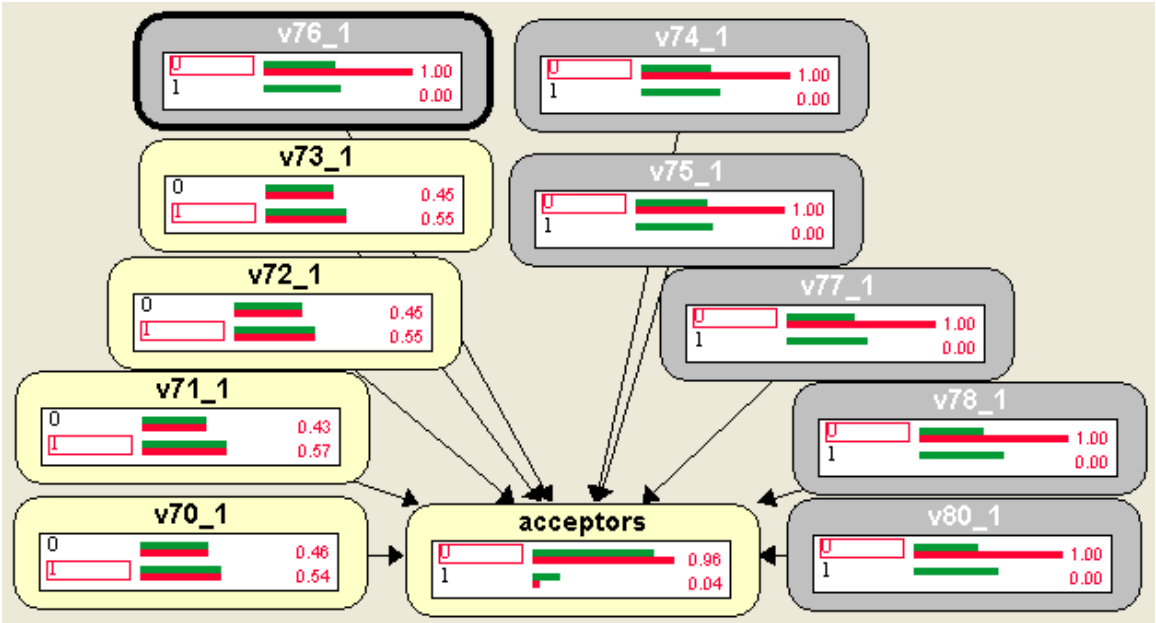


Figura 8.3. RB para las evidencias de la Tabla 8.2.

Anexo 9. Red Bayesiana de diagnóstico de la HTA con el algoritmo BayesChaid. Ejemplos de propagación de evidencias con el *software* ELVIRA

En las Figuras 9.1 y 9.2 se muestra la RB que se obtuvo mediante el algoritmo BayesChaid, cuando el número de padres es dos, el número de niveles en la red es tres, y sub-poblaciones hasta 30 casos.

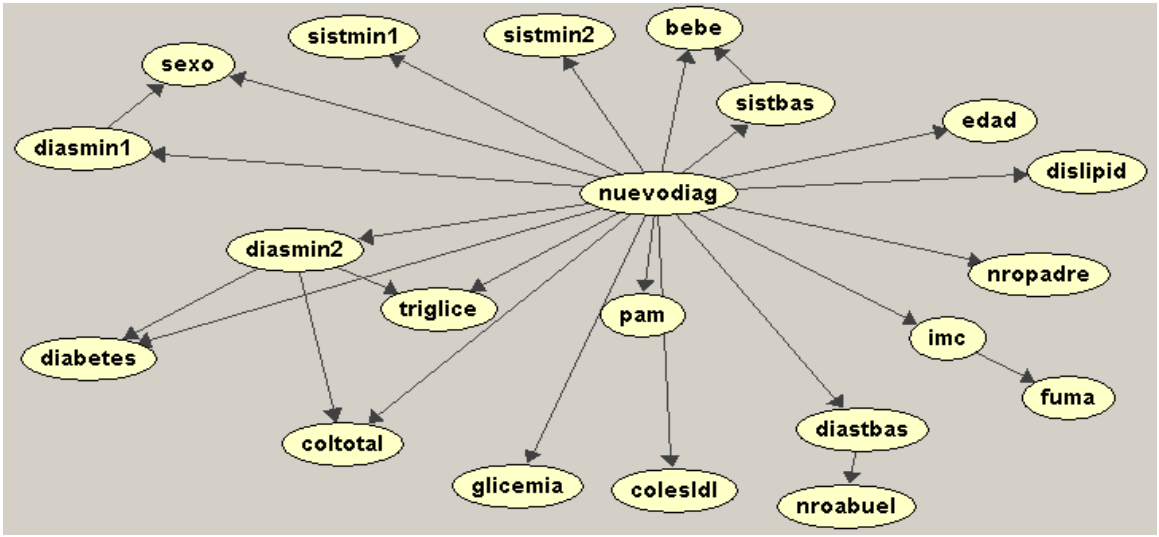


Figura 9.1. RB obtenida con el algoritmo BayesChaid

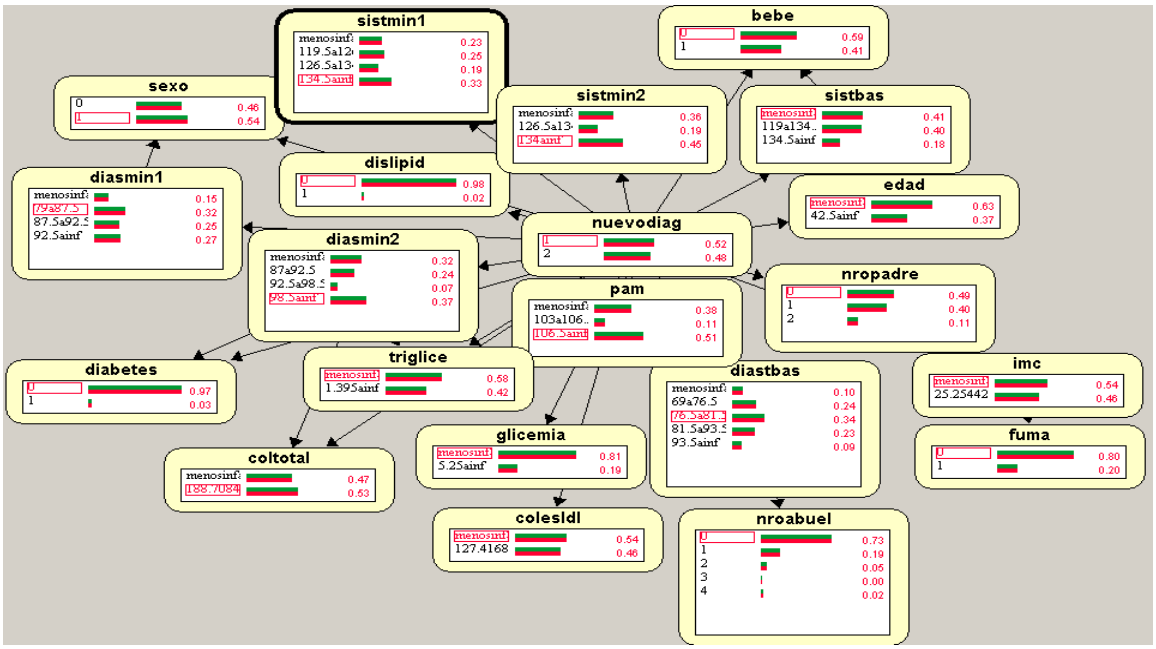


Figura 9.2. RB inicial sin evidencias

En la Figura 9.3 se muestra la RB ante un caso con la presión sistólica al minuto uno alta, lo que hace que se eleve la probabilidad de hipertenso a 0.97, también aumenta la probabilidad de la presión sistólica al segundo minuto, así como las presiones diastólicas y PAM.

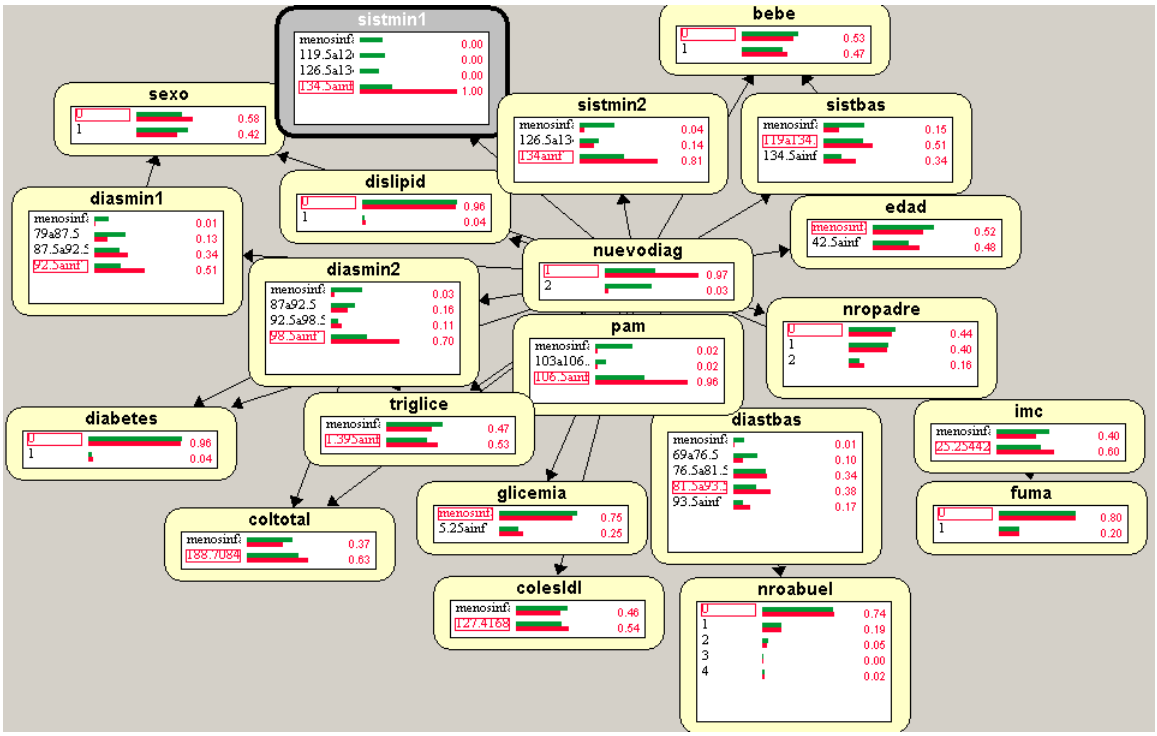
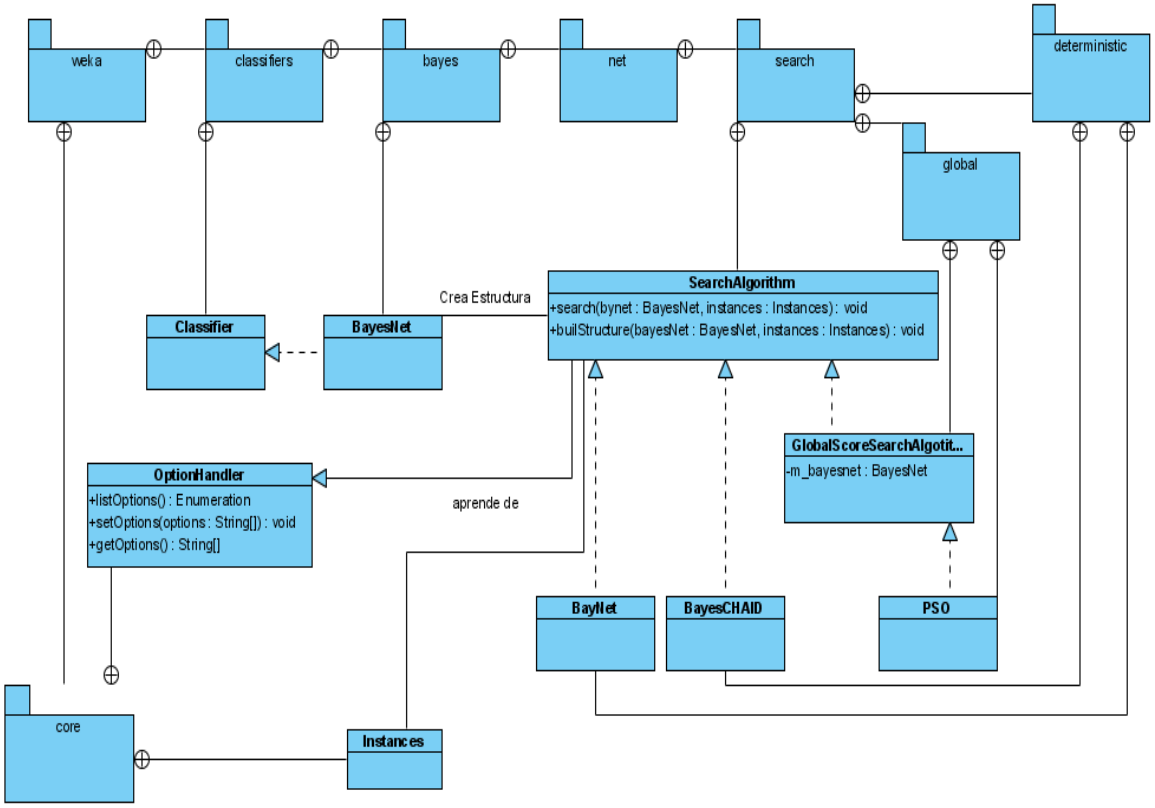


Figura 9.3. RB cuando la presión sistólica al minuto uno es muy alta

Es posible incluir varias evidencias simultáneamente, por ejemplo para un paciente diabético con alto índice de masa corporal, se incrementa la probabilidad de tener HTA a 0.94

Anexo 10. Diagrama de relación de Clases



Anexo 11. Sintaxis de los ficheros de datos para Weka y comandos para ejecutar Weka *parallel*

El formato ARFF está compuesto por una estructura claramente diferenciada en tres partes:

1. **Cabecera.** Se define el nombre de la relación. Su formato es el siguiente:
   @relation <nombre-de-la-relación>

   Donde *<nombre-de-la-relación>* es de tipo *String*. Si dicho nombre contiene algún espacio será necesario ponerlo entre comillas.

2. **Declaraciones de atributos.** En esta sección se declaran los atributos junto a su tipo.
   @attribute <nombre-del-atributo> <tipo>
   Donde *<nombre-del-atributo>* es de tipo *String* teniendo las mismas restricciones que el caso anterior.

Weka acepta diversos tipos de datos, estos son:

a) NUMERIC Expresa números reales.

b) INTEGER Expresa números enteros.

c) DATE Expresa fechas, para ello este tipo debe ir precedido de una etiqueta de formato entre comillas. La etiqueta de formato está compuesta por caracteres separadores (guiones y/o espacios) y unidades de tiempo: dd Día, MM Mes, yyyy Año, HH Horas, mm Minutos, ss Segundos.

d) STRING Expresa cadenas de texto, con las restricciones del tipo *String*

e) ENUMERADO El identificador de este tipo consiste en expresar entre llaves y separados por comas los posibles valores que puede tomar el atributo. Por ejemplo, si tenemos un atributo que indica el tiempo se define:

@attribute tiempo {soleado,lluvioso,nublado}

3. **Sección de datos**. Declaramos los datos que componen la relación, los atributos se separan entre comas y las relaciones con saltos de línea.

@data
4,3.2

Si algún dato es desconocido se representa con un símbolo de cerrar interrogación (“?"). Además es posible añadir comentarios con el símbolo “ %”, que indica que desde ese símbolo hasta el final de la línea es todo un comentario. Los comentarios pueden situarse en cualquier lugar del fichero.

Ejemplo de un fichero ARFF: prueba.**arff**

% Archivo de prueba para Weka.

@relation prueba
@attribute nombre STRING
@attribute ojo_izquierdo {Bien,Mal}
@attribute dimension NUMERIC
@attribute fecha_analisis DATE "dd-MM-yyyy HH:mm"

@data
Antonio,Bien,38.43,"12-04-2003 12:23"
’Maria Jose’,?,34.53,"14-05-2003 13:45"
Juan,?,?,"03-04-2003 11:03"

Otro formato es un fichero tipo CSV. En la primera línea del fichero se ubica el nombre de las variables separadas por coma y a continuación las instancias de casos.

Ejemplo: Fichero prueba2.**csv**

nombre, ojo_izquierdo, dimension, fecha_analisis
Antonio,Bien,38.43,"12-04-2003 12:23"
’Maria Jose’,?,34.53,"14-05-2003 13:45"

Si se resuelven problemas de aprendizaje supervisado, se debe indicar en el fichero la variable dependiente o clase al final.

Ejemplos:

Fichero prueba1**.arff**

@relation prueba1
@attribute V1 NUMERIC
@attribute V2 NUMERIC
@attribute V3 NUMERIC
@attribute **clase** {0, 1}

@data
163,0,0,0
8.67,0,5,1

Fichero prueba1**.csv**

V1,V2,V3,V4,**Clase**
0.163,0,0,0,0
8.67,0,1,5,1

Sintaxis de ficheros para ejecutar *Weka* parallel

Comando que ejecuta *weka parallel cliente.bat*:

java -Xmx290m -classpath new-weka-paralell.jar weka.gui.GUIChooser 6050

La sentencia indica memoria mínima 290MB, la clase que se debe ejecutar en Weka y el puerto que se utiliza para la conexión.

Comando que ejecuta *weka parallel server.bat*:

java -Xmx290m -classpath new-weka-paralell.jar weka.core.DistributedServer 6050

Con esta sentencia se indica la memoria virtual mínima y el puerto por el que la terminal donde se ejecute debe establecer la conexión con el cliente que la solicita.