Algoritmo de Aprendizaje para Redes Bayesianas de Nodos Temporales

Por

Pablo Francisco Hernández Leal

Tesis sometida como requisito parcial para obtener el grado de

MAESTRO EN CIENCIAS EN EL ÁREA DE CIENCIAS COMPUTACIONALES

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica.
Julio 2011
Tonantzintla, Puebla

Supervisada por:

Dr. Luis Enrique Sucar Succar
Dr. Jesús Antonio González Bernal

Algoritmo de Aprendizaje para Redes Bayesianas de Nodos Temporales

Por:

Pablo Francisco Hernández Leal

Asesores:

Dr. Luis Enrique Sucar Succar

Dr. Jesús Antonio González Bernal

Tesis de Maestría

Instituto Nacional de Astrofísica Óptica y Electrónica

Coordinación de Ciencias Computacionales

Tonanzintla, Puebla

Resumen

Las Redes Bayesianas se han vuelto el modelo de referencia para manejar incertidumbre debido a su facilidad de interpretación y diversos métodos de inferencia y aprendizaje. Sin embargo, las redes bayesianas tradicionales no pueden manejar información temporal. El modelo conocido como Redes Bayesianas de Nodos Temporales (RBNT) es una extensión que combina el manejo de incertidumbre con información temporal, pero su uso no se ha extendido debido a que no existen métodos de aprendizaje para estas redes.

En esta tesis proponemos un algoritmo de aprendizaje de Redes Bayesianas de Nodos Temporales que obtiene la estructura, los intervalos y los parámetros asociados. El algoritmo se compone de tres pasos principales: una discretización inicial de los nodos temporales, la obtención de una estructura inicial y posteriormente un refinamiento de los intervalos usando información de la red. El algoritmo de aprendizaje de intervalos hace uso de un algoritmo basado en agrupamiento para obtener los intervalos temporales. El conjunto de intervalos que obtenga el mejor puntaje predictivo es seleccionado.

El algoritmo fue evaluado con datos sintéticos de tres RBNTs de diferentes tamaños con dos distribuciones diferentes para generar los datos temporales. En los experimentos el algoritmo superó a los algoritmos base y obtuvo la mejor calidad estructural y el menor error temporal. El algoritmo también fue aplicado con datos reales, por un lado, en predicción y diagnóstico de fallas en un subsistema de una planta eléctrica. Para esta aplicación el algoritmo se evaluó con diferente número de casos de entrada en términos de calidad predictiva, error temporal y número de intervalos. Por otro lado, también se probó con datos de pacientes con VIH para obtener redes mutacionales; es decir redes, que muestren la evolución temporal de las mutaciones con respecto a ciertos medicamentos. Para esta aplicación los modelos fueron evaluados cualitativamente por los expertos.

Abstract

Bayesian networks have become the reference model to deal with uncertainty due to its easy understanding and different inference and learning algorithms. However, Bayesian networks can not deal with temporal information. The model known as Temporal Nodes Bayesian Networks (TNBN) is an extension that combines uncertainty reasoning with temporal information, but it has not been used extensively due to a lack of learning algorithms for this type of networks.

In this thesis we propose a learning algorithm for Temporal Nodes Bayesian Networks that obtains the structure, the intervals and the associated parameters. The algorithm has three main steps: an initial discretization of the temporal nodes, learning of an initial structure and a refinement of the intervals using the structure information. The intervals' learning algorithm uses a clustering technique to obtain the temporal intervals.

The algorithm was evaluated with synthetic data of three TNBNs of different sizes with two distributions to generate the temporal data. In the experiments the algorithm obtained better scores than the baselines, particularly in structural quality and temporal error. The algorithm was also applied with real data, on one side it was applied in prediction and fault diagnosis in a subsystem of a power plant. For this application the algorithm was evaluated using different number of cases in terms of predictive score, temporal error and number of intervals. On the other, it was applied with data from patients with HIV in order to obtain mutational networks; i.e. networks that show the temporal evolution of the mutations with respect to certain drugs. For these experiments, the models were qualitatively evaluated by experts.

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico otorgado a través de la beca No. 234507 para estudios de maestría. Al Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) y la Coordinación de de Ciencias Computacionales por la formación académica y todas las facilidades otorgadas.

Al apoyo del CONACYT y de la Comunidad Europea a través del FONCICYT en virtud del contrato de asignación de recursos/contrato de subvención nº 95185.

A mis asesores el Dr. L. Enrique Sucar y el Dr. Jesús A. González por su guía, revisiones y apoyo para llevar a cabo esta investigación.

Al personal del proyecto Dynamo, en particular al Dr. Alberto Reyes Ballesteros y el Dr. Pablo Ibarguengoytia del IIE, al Dr. Felipe Orihuela y la M.C. Alma Rios del INAOE y al Dr. Santiago Ávila del INER, quienes contribuyeron con asesorías, revisiones y sugerencias constructivas para esta tesis.

A mis sinodales, Dr. Eduardo Morales, Dr. Luis Villaseñor y el Dr. Saúl Pomares por sus observaciones y comentarios.

A mis padres por educarme, por quererme, por brindarme siempre su apoyo y sus consejos, sin ellos no podría llegar tan lejos. A Isabel porque contigo soy feliz.

Pablo Fco. Hernández Leal — INAOE

Índice general

Índice de figuras

Índice de Tablas

Lista de Algoritmos

Capítulo 1

Introducción

En muchos dominios como la medicina o la industria, no siempre se tiene la certeza plena del sistema y de las relaciones entre el estado interno y las observaciones externas. Por ejemplo, un médico sabe que no siempre se presentan los mismos síntomas para una misma enfermedad, existe una cierta probabilidad de que cada síntoma aparezca en todos los pacientes. De la misma manera en la industria no se sabe cuándo se presentarán fallas ni de que tipo serán. No obstante, los médicos y los ingenieros pueden trabajar con incertidumbre y aún así obtener diagnósticos acertados en forma eficiente. Este proceso se llama razonamiento bajo incertidumbre y es hoy en día una de las áreas de gran actividad en las ciencias computacionales. En particular ha tenido mucha atención porque ha logrado buenos resultados y porque tiene aplicación en áreas que son prioritarias en la vida humana, tales como la medicina, la economía y la industria.

Formalmente, para razonar bajo incertidumbre se han trabajado varias teorías, la principal es la teoría clásica de la probabilidad. Dentro de los modelos que se basan en ella están los modelos gráficos probabilistas (Koller y Friedman 2009). Éstos tienen la característica principal de que se pueden representar mediante diagramas visuales fáciles de entender, es decir grafos. Un tipo de estos modelos son las redes bayesianas, las cuales son representaciones que modelan dependencias condicionales entre variables aleatorias. De unos años a la fecha, las redes bayesianas se han vuelto populares por varias razones. La primera es lo

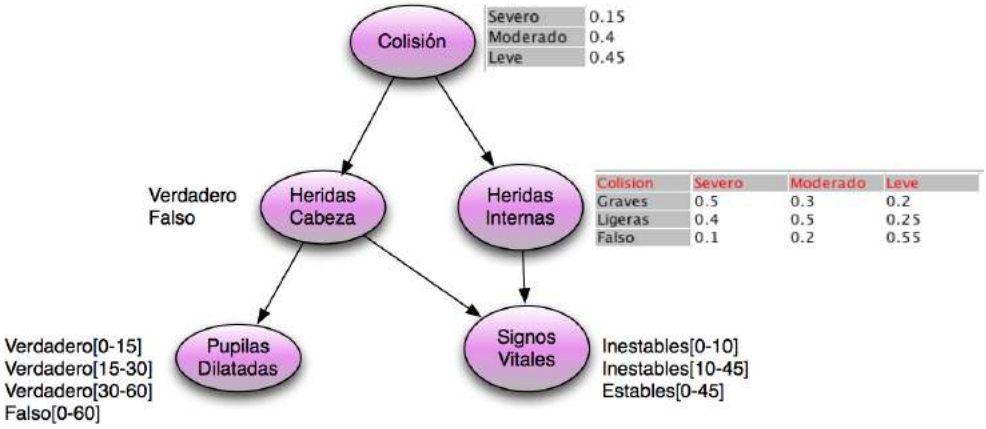


Figura 1.1: Ejemplo de una RBNT. Cada óvalo representa un evento. Los nodos superiores Colisión, Heridas Cabeza y Heridas Internas no contienen intervalos temporales ya que son nodos instantáneos. Los otros dos nodos (Pupilas Dilatadas, Signos Vitales) son nodos temporales con intervalos asociados. Cada nodo tiene asociada una tabla de probabilidades, por simplicidad sólo se muestran las tablas de los nodos Colisión y Heridas Internas.

atractivo de su representación visual, debido a que los nodos representan ciertas variables y los arcos pueden verse como relaciones causales, tienen una representación muy intuitiva para la mayoría de las personas. Otra razón de su éxito es que las teorías en las que se fundamentan, la teoría de grafos y la teoría de la probabilidad, tienen una amplia aceptación y entendimiento. Finalmente, podemos mencionar que han surgido varios métodos para realizar inferencia y aprendizaje de redes bayesianas, lo cual ha aumentado su uso.

Dentro de las redes bayesianas han surgido varias extensiones del modelo básico (conocido como red bayesiana estática), las cuales incluyen el aspecto de temporalidad. En particular se puede mencionar a las Redes Bayesianas Dinámicas, donde su objetivo es modelar procesos que ocurren a través del tiempo, por ejemplo en tareas de monitoreo. Por otra parte, existe otro grupo llamado Redes Bayesianas de Eventos, en este grupo podemos encontrar a las Redes de Eventos Discretos y a las Redes Bayesianas de Nodos Temporales (RBNT), estas últimas serán el objeto de estudio de este trabajo.

En la figura 1.1 se muestra una RBNT. Este ejemplo simplificado representa un accidente automovilístico y sus consecuencias. En un tiempo inicial se presenta una Colisión (nodo superior), la cual puede ser de tres tipos: *severo, moderado* o *leve*. La colisión desencadena

dos eventos instantáneos (representados por flechas que apuntan a dos nodos), herida en la cabeza (la cual puede ocurrir o no) y heridas internas (las cuales pueden ser *graves, ligeras* o *no ocurrir*). Los eventos anteriores, pueden llegar a desestabilizar los signos vitales y lograr que las pupilas se dilaten, en particular para estos eventos nos interesa el tiempo en el que ocurren, por lo que tienen intervalos asociados a ellos, estos nodos reciben el nombre de Nodos Temporales. Los intervalos en cada nodo representan el tiempo (en diferentes unidades de acuerdo a la aplicación) en el que cierto evento de interés ocurre. El número de intervalos y su tamaño pueden variar para diferentes nodos dando así una mayor flexibilidad al modelo. Por ejemplo el nodo Signos Vitales contiene el estado *Inestables [0-10]*, esto significa que los signos del paciente se desestabilizaron dentro de los primeros diez minutos de la colisión.

1.1. Problema

En el ejemplo anterior, figura 1.1, la estructura (los nodos y los arcos), los intervalos de los nodos temporales y las probabilidades de cada nodo se obtuvieron con ayuda de un experto. Este proceso se aplicaba a todas las RB en un principio, el grafo asociado y las probabilidades se obtenían del conocimiento del dominio ya que no existían métodos automáticos para obtenerlos. Lo anterior generaba principalmente dos problemas. El primero es que a veces no existe un concenso entre los expertos de cómo se debe especificar el modelo. Por ejemplo, dos médicos pueden tener diagnósticos diferentes para un mismo caso. El segundo problema surge en tratar de modelar procesos más complejos, ya que la tarea de obtención y especificación de la red y sus parámetros, aumenta en complejidad y tiempo, además de que no se asegura que los resultados sean los mejores.

Debido a los problemas mencionados, surgieron métodos de aprendizaje para redes bayesianas, es decir, algoritmos que a partir de datos obtienen como salida la estructura y los parámetros de una red bayesiana. Estos métodos realizan el aprendizaje en dos partes: (i) Aprendizaje estructural, es decir, las dependencias entre los nodos. (ii) Aprendizaje

paramétrico, es decir, las probabilidades asociadas a cada nodo.

Aún cuando existen diversos métodos de aprendizaje para Redes Bayesianas estáticas, no sucede lo mismo para Redes Bayesianas de Eventos. En específico, para definir una RBNT además de la estructura y las probabilidades, se necesita especificar los intervalos temporales, es por ello que los algoritmos de aprendizaje existentes de RB no se pueden aplicar directamente a las RBNT.

Existen dos algoritmos relacionados al aprendizaje de RBNT. El primero es el algoritmo presentado en (Friedman y Goldszmidt 1996), el cual realiza una discretización de las variables continuas mientras realiza el aprendizaje estructural una red bayesiana. Para realizar la discretización usa un puntaje basado en el principio de descripción de longitud mínima. Vale la pena mencionar que este algoritmo no fue diseñado para aprender RBNT.

El segundo algoritmo relacionado es presentado en (Liu, Song y Yao 2005). Este algoritmo aprende la estructura de una RBNT a partir de una base de datos temporal probabilista; sin embargo, no realiza un aprendizaje de los intervalos de los nodos temporales, el cual es una parte central de esta tesis.

1.2. Objetivos

El objetivo de este trabajo es desarrollar un algoritmo de aprendizaje para Redes Bayesianas de Nodos Temporales, que obtenga la estructura, los intervalos y probabilidades necesarias para realizar procesos de inferencia.

1.2.1. Objetivos Específicos

- Desarrollar un algoritmo para obtener los intervalos en cada nodo temporal de una RBNT.
- Desarrollar un algoritmo que aprenda la estructura y parámetros de una RBNT.
- Evaluar los algoritmos de aprendizaje con datos sintéticos y de algún dominio real.

1.3. Solución Propuesta

En esta tesis se presenta un algoritmo de aprendizaje de redes bayesianas de nodos temporales, el cual consta de tres fases principales:

1. Realizar una discretización inicial de las variables temporales, con lo que se obtiene una aproximación inicial a los intervalos de todos los nodos temporales.

2. Aplicar un aprendizaje estructural estándar, el algoritmo usado es el conocido como K2 (Cooper y Herskovits 1992), con el que se obtiene una estructura.

3. Aplicar un algoritmo de aprendizaje de intervalos para refinar los intervalos de cada nodo temporal por medio de un algoritmo de agrupamiento. Para esto se usa la información de la estructura de la red. Para obtener los intervalos se realiza una aproximación basada en el modelo de mezcla de gausianas. Cada grupo corresponde en principio a un intervalo temporal. Posteriormente estos intervalos se combinan y se selecciona el conjunto de intervalos que mejor calidad predictiva obtenga.

El algoritmo fue evaluado con tres redes sintéticas de diferentes tamaños. Para cada red se variaron los parámetros de tamaño de los datos de entrada y la inicialización. Las redes aprendidas se compararon con las redes de referencia y se evaluaron en términos de calidad estructural, calidad predictiva y calidad de los intervalos. Además el algoritmo se aplicó en dos dominios con datos reales.

1.4. Contribuciones

La contribución principal de esta tesis es un algoritmo de aprendizaje para Redes Bayesianas de Nodos Temporales. De ella se desprende una contribución más:

- Se desarrolló un algoritmo de aprendizaje de intervalos para los Nodos Temporales de una RBNT.

Además se presentaron dos aplicaciones del algoritmo propuesto usando datos reales:

- Se usó el algoritmo para diagnóstico de fallas en un subsistema de una planta eléctrica de ciclo combinado.

- Se aplicó el algoritmo a datos de pacientes de VIH para entender mejor el proceso mutacional del virus.

En esta tesis se realizaron experimentos con tres RBNT sintéticas de distintos tamaños. Para cada una de las redes se evaluaron 3 algoritmos y el propuesto, se realizaron pruebas variando la distribución de los datos temporales, el número de datos de entrada y la inicialización, además se realizaron pruebas de significancia estádistica. De los experimentos se extraen las siguientes conclusiones respecto al algoritmo:

- El algoritmo propuesto obtuvo los mejores resultados en calidad estructural y error temporal en comparación con otros tres algoritmos: discretización uniforme, *K-means* y el algoritmo presentado en (Friedman y Goldszmidt 1996).

- El algoritmo propuesto superó en promedio a los algoritmos base (discretización uniforme y *K-means*) en las medidas estructurales, de intervalos y de predicción.

- Aún cuando el algoritmo propuesto hace la suposición de que los datos siguen una distribución gausiana, al evaluarlo con datos con distribución uniforme se obtuvieron buenos resultados, superando a los algoritmos base.

1.5. Estructura de la tesis

La tesis esta estructurada de la siguiente forma. En el capítulo 2 se presenta una introducción a las Redes Bayesianas. En el capítulo 3 se detalla el modelo de Redes Bayesianas de Nodos Temporales. En el capítulo 4 se presenta el algoritmo propuesto en esta tesis que aprende Redes Bayesianas de Nodos Temporales. En el capítulo 5 se describen los experimentos realizados y los resultados obtenidos para evaluar el algoritmo con datos sintéticos. El algoritmo propuesto también se evaluó con datos reales en dos distintos dominios, uno

industrial y otro médico, estas aplicaciones se presentan en el capítulo 6. Finalmente, en el capítulo 7 se presentan las conclusiones y algunas ideas para trabajo futuro.

Capítulo 2

Redes Bayesianas

En este capítulo se presenta el modelo de redes bayesianas y su definición formal. Posteriormente se presenta un panorama general de los algoritmos de aprendizaje de redes bayesianas. Finalmente se mencionan algunas extensiones que ha tenido el modelo de las redes bayesianas, en particular se mencionan las redes bayesianas dinámicas y las redes bayesianas de eventos.

2.1. Introducción

Las Redes Bayesianas (Pearl 1988) se han vuelto populares en relativamente poco tiempo. Las razones son variadas, pero podemos mencionar 3 muy importantes (Daly y Shen 2009).

- El modelo construido tiene un significado intuitivo, esto ocurre ya que el grafo que la compone da una noción de causalidad. Mas aún, las tablas de probabilidad de cada nodo ayudan a cuantificar esas relaciones.
- Las teorías en las que se basan: la teoría de la probabilidad y la teoría de grafos, están bien estudiadas y fundamentadas.
- La aparición de nuevos métodos para aprender la estructura y los parámetros ha

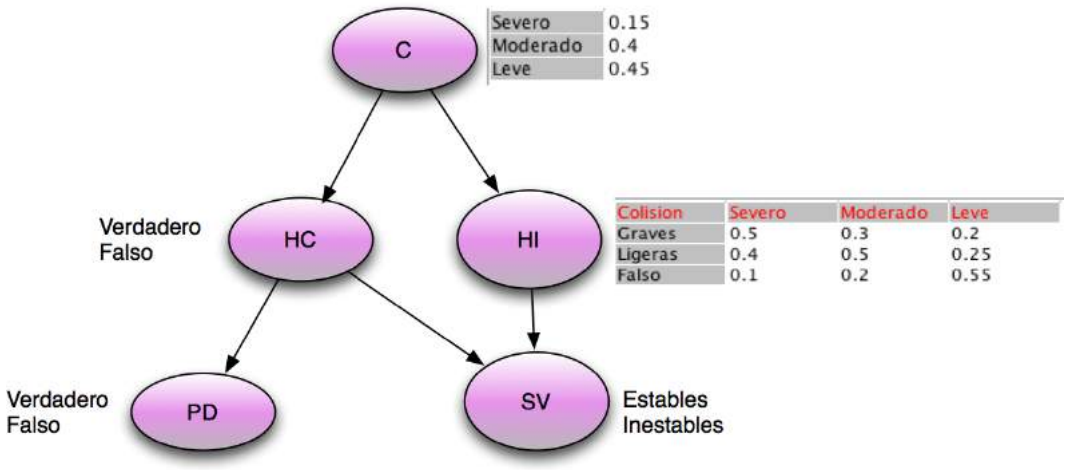


Figura 2.1: Una red bayesiana estática que representa un accidente automovilístico y sus posibles consecuencias. Dentro de los nodos se muestra el nombre y fuera de ellos se muestra los posibles estados. Para los nodos Colisión y Heridas Internas se muestran las tablas de probabilidad condicional. Las flechas indican una relación de dependencia entre los nodos las cuales se cuantifican en las tablas de probabilidad.

causado que su uso aumente en los últimos años.

Además de lo anterior, las redes bayesianas han tenido éxito en diferentes áreas de aplicación tales como: economía (Neapolitan y Jiang 2007), diagnóstico médico (Charitos et al. 2009), diagnóstico industrial de plantas eléctricas (Knox y Mengshoel 2009) e incluso se han adentrado en áreas como la bioinformática (Neapolitan 2009).

2.2. Definición de Red Bayesiana

Antes de definir una red bayesiana necesitamos tener en cuenta algunos conceptos de grafos.

Un grafo dirigido es un par $\mathcal{G} = (V, E)$ donde V es un conjunto finito no vacío, cuyos elementos son llamados nodos y E es un conjunto de pares ordenados de elementos de V. Los elementos de E son llamados arcos dirigidos. Si $(X, Y) \in E$, decimos que existe un arco de X a Y. Un grafo acíclico dirigido (GAD) es un grafo dirigido que no contiene ciclos.

Definición 2.1 *Supongamos que tenemos una distribución de probabilidad conjunta P de*

un conjunto de variables aleatorias V y un GAD $\mathcal{G} = (V, E)$. Decimos que $(\mathcal{G}, P)$ satisface la Condición de Markov *si para cada variable $X \in V$, X es condicionalmente independiente del conjunto de sus no descendientes dado el conjunto de sus padres. Si $(\mathcal{G}, P)$ satisface la condición de Markov entonces $(\mathcal{G}, P)$ es llamada Red Bayesiana.*

Una Red Bayesiana $(\mathcal{G}, P)$ por definición es un GAD $\mathcal{G}$ y una distribución de probabilidad conjunta P que satisfacen la condición de Markov.

Teorema 2.1 *$(\mathcal{G}, P)$ satisface la condición de Markov (es una red bayesiana) si y solo si P es igual al producto de sus distribuciones condicionales de todos los nodos dados los padres en $\mathcal{G}$, siempre que estas distribuciones condicionales existan.*

Una forma de interpretar el teorema 2.1, es que podemos representar una red bayesiana $(\mathcal{G}, P)$ usado un GAD (estructura) $\mathcal{G}$ y las distribuciones condicionales de cada nodo (parámetros). Debido a esto, no es necesario mostrar todas las distribuciones conjuntas. Entonces podemos decir que una Red Bayesiana es una estructura para representar una distribución de probabilidad conjunta de una forma sucinta (Neapolitan 2004).

Por ejemplo, sea B la RB de la figura 2.1, para obtener la probabilidad conjunta, tenemos en base a la regla de la cadena:

$$\begin{aligned} P(B) &= P(C, HC, HI, PD, SV) \\ &= P(SV|PD, HI, HC, C)P(PD|HI, HC, C)P(HI|HC, C)P(HC|C)P(C) \end{aligned}$$

pero sabemos que para una Red Baysiana se cumple

$$P(B) = \prod_{X \in V} P(X|Pa(X))$$

por lo que,

$$\begin{aligned} P(B) &= P(C, HC, HI, PD, SV) && (2.1) \\ &= P(C)P(HC|C)P(HI|C)P(PD|HC)P(SV|HC, HI) && (2.2) \end{aligned}$$

2.3. Aprendizaje de Redes Bayesianas

En un principio las redes bayesianas se definían con ayuda de expertos, la estructura y las probabilidades se obtenían sin ayuda de técnicas computacionales. Esta tarea es laboriosa y especialmente complicada cuando se trata de redes de gran tamaño. Por ello se desarrollaron métodos para aprender tanto la estructura como las probabilidades condicionales a partir de datos.

De unos años a la fecha, han surgido algoritmos de aprendizaje para redes bayesianas estáticas. Generalmente el aprendizaje de las redes ocurre en dos fases: el aprendizaje estructural y el aprendizaje paramétrico.

2.3.1. Aprendizaje Estructural

Para el aprendizaje estructural, el objetivo es obtener un GAD que represente el modelo y sus dependencias.

Cuando el número de variables no es muy grande se podrían enumerar y evaluar de forma exhaustiva todos los posibles GAD y seleccionar el que tenga el *puntaje* mayor. Sin embargo, (Robinson 1977) mostró que el número de GAD que contienen n nodos se define por la siguiente recurrencia:

$$\begin{aligned} f(n) &= \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i) \qquad n > 2 \\ f(0) &= 1 \\ f(1) &= 0 \end{aligned}$$

Por ejemplo $f(2) = 3$, $f(5) = 29,000$, $f(10) = 4{,}2 \times 10^{18}$. En particular, se ha demostrado que encontrar la estructura óptima con variables discretas es un problema NP-difícil (Chickering, Geiger y Heckerman 1994), por lo que es común usar técnicas heurísticas y aproximadas.

Existen en general tres enfoques para resolver este problema (Daly y Shen 2009). Los

primeros tratan de encontrar independencias condicionales en los datos y con ello producen una estructura. Un ejemplo de este tipo de método es (Spirtes y Glymour 1991). Los segundos usan técnicas de programación dinámica y agrupamiento (Eaton y Murphy 2007), estos métodos tienen buenos resultados para redes pequeñas (menos de 30 variables), pero al aumentar el número de variables se vuelven ineficientes. Finalmente, existe un tercer grupo de métodos que realiza una búsqueda en el espacio de redes bayesianas, estos métodos usan heurísticas y funciones de evaluación para obtener una red.

Algoritmo K2

Uno de los algoritmos más conocidos que realiza un proceso de búsqueda en el espacio de los GAD es el algoritmo K2 (Cooper y Herskovits 1992), el cual se muestra en el algoritmo 2.1. Una de las ventajas de este algoritmo es que usa un puntaje de actualización local ya que sólo recalcula unos pocos valores para obtener un puntaje del nuevo modelo, lo cual lo hace más eficiente.

Este algoritmo hace tres suposiciones iniciales: las variables de los casos son discretas, los casos son independientes dada la red bayesiana y se debe determinar un cierto orden en los nodos, con el que se limita la explosión de posibles combinaciones. Con lo anterior se obtiene un algoritmo de aprendizaje estructural con una complejidad polinomial.

El algoritmo funciona de manera incremental, inicia con un grafo desconectado y va agregando padres a los nodos de una forma voraz. Cuando no existe un padre que mejore la estructura, el algoritmo termina.

Generalmente el orden de la variables que se requiere para ejecutar el algoritmo se obtiene de información del dominio tal como el orden temporal de las variables (Neapolitan 2009).

En el algoritmo 2.1 se presenta el pseudocódigo de K2. Cada nodo se representa por X_i. Cada uno de ellos tiene r_i posibles valores o estados. Los padres de cada nodo X_i se representan por PA_i. Denotamos por w_{ij} la j-ésima asignación de PA_i relativa a los datos D y q_i es el número de asignaciones únicas de PA_i.

La función $Pred(X_i)$ obtiene el conjunto de nodos que precede a X_i en el orden de los nodos.

La función puntaje intuitivamente es la probabilidad de los datos D dado que los padres de X_i son PA_i, formalmente se define entonces de la siguiente manera:

$$puntaje(X_i, PA_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{2.3}$$

donde N_{ijk} es el número se casos en D, en que la variable X_i tiene el valor v_{ik} y PA_i esta instanciada como w_{ij}.

N_{ij} se define por: $N_{ij} = \sum_{k=1}^{r_j} N_{ijk}$. Esto es el número de instancias en los datos en que los padres de X_i tienen la asignación w_{ij}.

Algoritmo 2.1: Algoritmo K2

Datos: Un conjunto V de n variables aleatorias; una cota superior u en el número de padres que un nodo puede tener; datos D.
Resultado: n conjuntos de padres PA_i, en un GAD que aproxime el puntaje máximo de G.

```
para (i = 1; i <= n; i++) hacer
    PA_i = ∅  P_anterior = puntaje(X_i, PA_i); // ver la ecuación 2.3.
    buscaMas=verdadero;
    mientras (buscaMas AND |PA_i| < u) hacer
        Z = nodo en Pred(X_i) - PA_i que maximice puntaje(X_i, PA_i ∪ {Z});
        P_nuevo = puntaje(X_i, PA_i ∪ {Z});
        si P_nuevo > P_viejo entonces
            P_viejo = P_nuevo;
            PA_i = PA_i ∪ {Z};
        sino
            buscaMas=falso;
        fin si
    fin mientras
fin para
```

El algoritmo K2 funciona de la siguiente manera. Recorre cada una de las variables en el orden proporcionado por el usuario. Para cada variable obtiene un puntaje inicial, $P_{anterior}$, y activa una bandera, $buscaMas$; posteriormente inicia un ciclo que se repite mientras la

bandera esté activa y no se haya llegado a la cota del número de padres $|PA_i| < u$. En el ciclo obtiene un puntaje P_{nuevo} y verifica si es mejor que $P_{anterior}$, si es así entonces se actualiza el puntaje y se agrega ese nodo como padre, sino entonces la bandera se actualiza falso.

El algoritmo propuesto en esta tesis hace uso de un algoritmo de aprendizaje estructural, en específico se seleccionó el algoritmo K2.

2.3.2. Aprendizaje paramétrico

En una red bayesiana las probabilidades condicionales son también llamados parámetros. Por lo tanto, el aprendizaje paramétrico consiste en la obtención de las tablas de probabilidad condicional (TPC) para cada nodo. Estas tablas representan las probabilidades de cada estado de ese nodo dados los valores de los padres. Para este problema se sabe que si se realiza la suposición de que no existen datos faltantes entonces el aprendizaje se reduce a un simple conteo de frecuencias. No obstante, cuando se elimina esta suposición (lo cual es razonable) entonces la solución ya no es directa. En general esto es un problema intratable y se requieren métodos aproximados (Heckerman 2008; Feelders y Gaag 2006). Diversos métodos para resolver este problema hacen uso del algoritmo de Maximización de la Expectancia EM (Moon 1996), el cual está garantizado que converge a un máximo local.

2.4. Extensiones de Redes Bayesianas

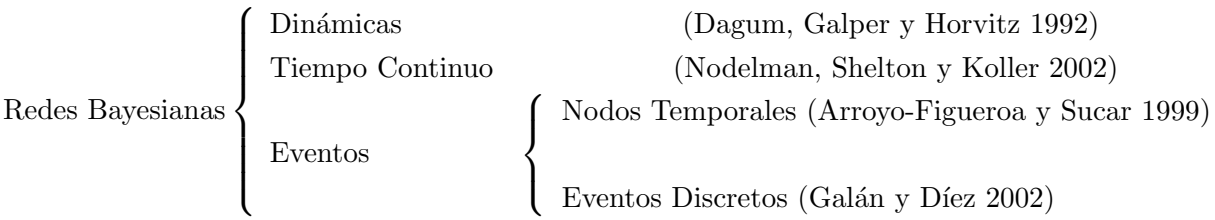


Figura 2.2: Clasificación de las Redes Bayesianas Temporales

Debido al éxito de las Redes Bayesianas estáticas han surgido muchas extensiones. En

la figura 2.2 se muestran algunas de las extensiones más importantes que consideran el tiempo. Las más conocidas son las Redes Bayesianas Dinámicas, otras más recientes son las Redes Bayesianas de Tiempo Continuo. Ambas redes han sido pensadas para modelar procesos que ocurren durante un cierto tiempo y donde se dan muchos cambios de estado, como ejemplo específico se puede pensar en tareas de monitoreo.

La mayoría de los métodos conocidos de aprendizaje se aplican para Redes Bayesianas estáticas. No obstante, existen algoritmos de aprendizaje para Redes Bayesianas Dinámicas (Ghahramani 1998) y para Redes Bayesianas de Tiempo Continuo (Nodelman, Shelton y Koller 2003).

Otro tipo de redes son las Redes de Eventos, las cuales tienen la característica que modelan procesos continuos donde sólo unos pocos cambios de estado en las variables son importantes, con lo cual se trata de simplificar el modelo obtenido.

2.4.1. Redes Bayesianas Dinámicas

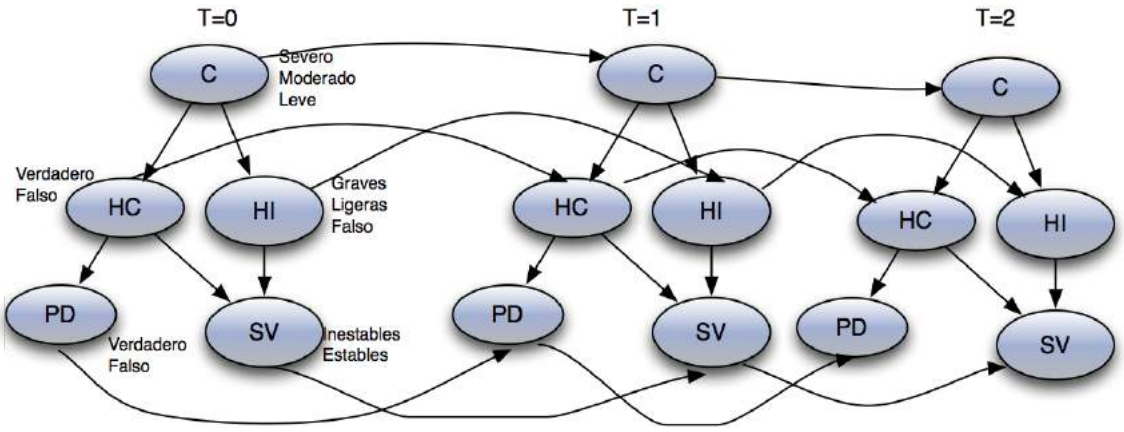


Figura 2.3: Una Red Bayesiana Dinámica para representar un accidente automovilístico y sus posibles consecuencias. Una RBD está compuesta por múltiples copias de RB estáticas en el tiempo, en este caso se muestran tres etapas o tiempos del modelo.

Una Red Bayesiana Dinámica (RBD) es un modelo probabilista que representa una secuencia de variables en el tiempo. Generalmente se puede ver una RBD como múltiples

copias de Redes Bayesianas estáticas con arcos que las unen. Un ejemplo de RBD es presentado en la figura 2.3. En el contexto de modelos gráficos probabilistas, las RBD son el método usual para representar procesos temporales. Para aplicar este modelo el tiempo es discretizado y por cada instante de tiempo deseado se crea una copia de la red estática, cada nodo de la red representa el estado de una variable en ese tiempo particular. Una RBD se puede pensar como una serie de fotos instantánea en el tiempo t de todo el sistema. Las RBD pueden llegar a ser muy complejas, por lo que en general se realizan ciertas suposiciones básicas: la primera es que el tiempo se divide en rebanadas, que son intervalos regulares con una granualiradad predeterminada Δ. Otra simplificación es pensar que el valor de las variables en cada estado depende sólo de la inmediata anterior, formalmente se dice que un proceso es Markoviano si se asume que el futuro es condicionalmente independiente del pasado dado el presente:

$$P(S_{t+1}|S_t, S_{t-1}, S_{t-2}, ...) = P(S_{t+1}|S_t)$$

Además se asume que la red es estacionaria, es decir que la estructura de la red no cambia entre estados de tiempo. Por lo que es común dividir a los arcos de una RBD en dos tipos, los arcos intra red, que son los que existen en un solo tiempo de la red, y los arcos inter redes, que representan los arcos que van de una red de un tiempo a la red del tiempo siguiente.

Formalmente una RBD se define como un par $(B_0, B_{\rightarrow})$, donde B_0 es una red bayesiana en el tiempo t_0 representando la distribución inicial de los estados y$B_{\rightarrow}$ es una red bayesiana de dos tiempos del proceso.

Aprendizaje de Redes Bayesianas Dinámicas

Para el aprendizaje paramétrico en RBD generalmente se ocupan las mismas técnicas que para RB estáticas (Murphy 2002), tales como el algoritmo EM. La única diferencia es que los parámetros iniciales se obtienen de forma independiente que la matriz de transición.

Para el aprendizaje estructural de RB Dinámicas lo que se hace comúnmente es aprender

la intra-conectividad (las conexiones dentro una rebanada de tiempo), la cual debe ser un GAD, y la inter-conectiviad (las conexiones de una rebanada a otra), que es equivalente a un problema de selección de variables, dado que para cada nodo en una etapa t, debemos elegir sus padres de la etapa $t-1$. Si se asume que la intra-conectividad es fija entonces aprender una RBD se reduce a una selección de atributos (Murphy 2002).

2.4.2. Redes Bayesianas de Eventos

Otras extensiones de las RB son las Redes Bayesianas de Eventos. En ellas se encuentran las Redes Probabilísticas de Tiempo Discreto (Galán y Díez 2002) y las Redes Bayesianas de Nodos Temporales (Arroyo-Figueroa y Sucar 1999) las cuales serán objeto de estudio en la presente tesis. En estas redes, a diferencia de las RBD, cada valor de la variable representa el tiempo en el que cierto evento ocurrió. Estas dos redes fueron diseñadas pensando en tareas en las que lo importante son los cambios de estado de ciertas variables, además de que tratan de modelar eventos irreversibles; es decir, eventos que sólo ocurren una vez en el tiempo. Ejemplos de aplicaciones de las redes de eventos son la predicción de fallas en plantas eléctricas (Arroyo-Figueroa, Sucar y Villavicencio 1998) y el pronóstico de cáncer nasofaríngeo (Galán et al. 2001).

Para redes de eventos, a la fecha sólo se conoce una forma automática para aprender una RBNT, este método se presenta en la sección 3.2.1.

2.5. Resumen

En este capítulo se presentó un panorama general de las redes bayesianas y sus métodos de aprendizaje, en particular se describió el algoritmo K2 de aprendizaje estructural. Además se presentaron algunas de las extensiones como las redes bayesianas dinámicas. Algunas de las limitaciones de las redes bayesianas dinámicas es su alta complejidad cuando se trata de representar procesos temporales donde ocurren pocos cambios de estados durante un largo tiempo. Por otro lado, dentro de las redes bayesianas de eventos se encuentran las

redes bayesianas de nodos temporales, las cuales tienen una representación más compacta y simple en ciertos dominios donde ocurren pocos eventos temporales de importancia. Por ello son el objeto de estudio de esta tesis y se presentan en el siguiente capítulo.

Capítulo 3

Redes Bayesianas de Nodos Temporales

En este capítulo se presentará la definición formal de Redes Bayesianas de Nodos Temporales. Además se presentarán los algoritmos existentes que están relacionados al aprendizaje de las mismas, para ello se introduce el concepto de discretización.

3.1. Representación

Una red bayesiana de nodos temporales (RBNT) (Arroyo-Figueroa y Sucar 1999) es una extensión de una red bayesiana que está compuesta por un conjunto de Nodos Temporales (NT) y de Nodos Instantáneos (NI). Los nodos están conectados por arcos, cada arco representa una relación causal-temporal entre ellos. Los nodos que no cuentan con intervalos, reciben el nombre de Nodos Instantáneos. Para los Nodos Temporales existe a lo más un cambio de estado para cada variable en el rango temporal de interés. El valor tomado por la variable representa el intervalo en el que el evento ocurre. El tiempo es discretizado en un número finito de intervalos para cada variable, lo cual permite que se tenga un número diferente de intervalos y de tamaño de los mismos, dando lugar a múltiple granularidad. Cada intervalo definido para un nodo hijo representa el posible retraso entre la ocurrencia

de uno de sus padres (causa) y el correspondiente evento hijo (efecto).

Definición 3.1 *Una RBNT es un par* $B = (\mathcal{G}, \Theta)$*, donde* $\mathcal{G}$ *es un GAD* $\mathcal{G} = (\mathbf{V}, \mathbf{E})$*.* $\mathcal{G}$ *se compone de* $\mathbf{V}$*, un conjunto de Nodos Temporales e Instantáneos, y* $\mathbf{E}$*, un conjunto de arcos entre nodos. El segundo componente,* Θ*, corresponde al conjunto de parámetros que cuantifica la red.* Θ *contiene los valores* $\Theta_{v_i} = P(v_i|\Pi_{v_i})$ *para cada valor* v_i *de* V *y donde* Π_{v_i} *representa el conjunto de padres de* V_i *en* $\mathcal{G}$*.*

Definición 3.2 *Un Nodo Temporal,* v_i*, está definido mediante un conjunto de estados* $\mathbf{S}$*. Cada estado se define como un par ordenado* $S = (\lambda, \tau)$*, donde* λ *es un valor de una variable aleatoria y* $\tau = [a, b]$ *es el intervalo de tiempo asociado, con valor inicial* a *y valor final* b*, que corresponde al tiempo en que ocurre el cambio de valor de la variable aleatoria* λ*. Además, cada Nodo Temporal cuenta con un estado extra* $s = (\text{'Default'}, \emptyset)$*, el cual no tiene intervalo asociado y corresponde al valor inicial (por defecto) de cada Nodo Temporal. Si todos los estados de un Nodo Temporal no contienen intervalos asociados entonces ese nodo recibe el nombre de Nodo Instantáneo.*

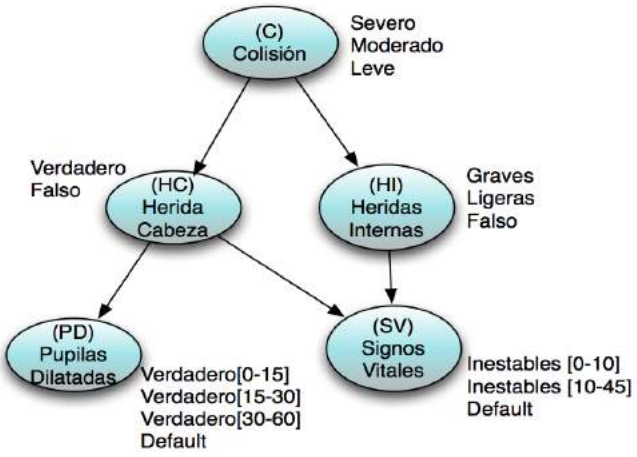


Figura 3.1: Una RBNT que representa un accidente automovilístico y sus posibles consecuencias en el tiempo. Existen 3 Nodos Instantáneos: Colisión, Heridas Cabeza y Heridas Internas. Los otros dos nodos, Pupilas Dilatadas y Signos Vitales, son Nodos Temporales con intervalos asociados.

La siguiente RBNT es un ejemplo basado en (Arroyo-Figueroa y Sucar 1999), su correspondiente representación gráfica se muestra en la figura 3.1.

Ejemplo 3.1 *Asumamos que ocurre un accidente en el tiempo* $t = 0$*, una Colisión. Este tipo de accidente puede ser clasificado como* Severo*,* Moderado *o* Leve*. Por simplicidad sólo consideraremos dos consecuencias inmediatas, Heridas en la Cabeza y Heridas Internas. Heridas Cabeza tiene los estados* Verdadero *y* Falso*, Heridas Internas tiene tres estados:* Graves*,* Ligeras *y* Falso*. Estos eventos son (Nodos) instantáneos que no tienen intervalos asociados, pero generarán subsecuentes cambios tales como Pupilas Dilatadas y desestabilización de Signos Vitales. Estos cambios no son inmediatos, ellos dependen de la severidad del accidente, por lo tanto tienen intervalos asociados a ellos.*

En este ejemplo existen 3 eventos instantáneos: Colisión, Heridas en la Cabeza, y Heridas Internas. De la misma forma existen 2 eventos (Nodos) Temporales con intervalos asociados. Pupilas Dilatadas es causado por Heridas Internas y en este caso nos interesa si las pupilas se dilatan en los primeros 15 minutos (intervalo [0-15]), antes de la media hora (intervalo [15-30]) o dentro de la primera hora (intervalo [30-60]). Por otra parte, Signos Vitales es causado por Heridas Cabeza o por Heridas Internas, aquí sólo existen dos intervalos de interés: [0-10] y [10-45].

3.2. Aprendizaje de RBNT

Para el caso particular del aprendizaje de RBNT, además del aprendizaje estructural y paramétrico, se necesita un paso intermedio que no existe en otras RB: *encontrar los intervalos asociados a cada Nodo Temporal*. Este paso es importante y soluciones triviales no funcionan de forma eficiente. Veamos dos ejemplos:

- Un intento simple sería eliminar los intervalos y crear un solo intervalo para cada NT. Sin embargo, de esta forma se perdería todo el sentido de los eventos temporales y se comportaría de forma muy parecida a una RB estática.

- Otra posible solución es crear k intervalos del mismo tamaño. En este caso el problema ahora es obtener ese k, el número de intervalos. Un número pequeño podría reducir la calidad de la red mientras que un número elevado podría hacer que la red fuera demasiado compleja para realizar procesos de inferencia.

De aquí se obtiene una idea de porque los métodos simples no son convenientes y se tienen que aplicar otras técnicas más elaboradas.

Es importante mencionar que los métodos de aprendizaje para Redes Bayesianas no son aplicables directamente, debido a que para especificar una RBNT se necesita la estructura, las probabilidades y los intervalos temporales. Estos últimos no existen en otras redes y por tanto no se puede realizar el aprendizaje usando los algoritmos ya existentes.

3.2.1. Algoritmo de Liu

A la fecha existe un método para aprender una RBNT (Liu, Song y Yao 2005), este algoritmo proviene de la teoría de Bases de Datos Temporales (Jensen, Snodgrass y Soo 2002) y Dependencias Funcionales Temporales (Wijsen 1995). En base a ellas, se realizan ciertas restricciones que ayudan a obtener la estructura y más tarde los parámetros de la RBNT. Sin embargo, para aplicar este método es necesario que los datos provengan de una *base de datos temporal probabilista*, lo cual limita su aplicación.

El método obtiene la estructura de la RBNT mediante un conjunto de dependencias temporales en un Modelo Relacional Temporal Probabilista (MRTP). Para construir la RBNT, se obtiene un orden de las variables que maximiza el conjunto de relaciones de independencia lo que es implicado por un grafo obtenido del MRTP. Basado en este orden, un grafo acíclico dirigido es obtenido el cual corresponde a las relaciones de independencia. Una vez que la estructura fue obtenida, las correspondientes tablas de probabilidad son inferidas.

En particular, este trabajo asume que se tiene un modelo relacional temporal probabilista, lo cual no es siempre el caso. Obtener este modelo puede ser tan difícil como construir una RBNT. En contraste el algoritmo presentado en esta tesis usa datos directamente sin

información adicional. Mas aún, (Liu, Song y Yao 2005) no atacan directamente el problema de aprendizaje de intervalos, que es una parte central de esta tesis.

3.2.2. Discretización

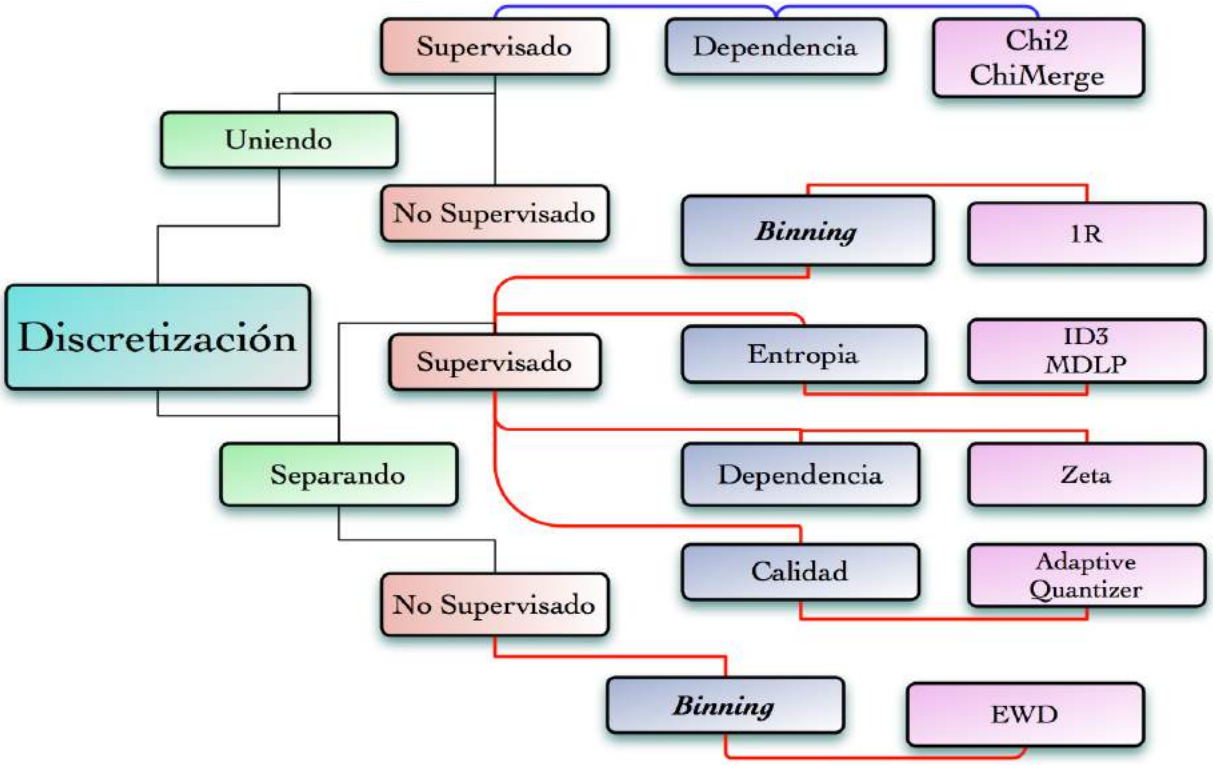


Figura 3.2: Esquema jerárquico de algunos métodos de discretización (Liu et al. 2002). Se divide a los métodos primero por su forma de obtener intervalos, después por si usan información de la clase, y finalmente por la medida que usan. En el último nivel se muestran algunos algoritmos conocidos de cada tipo.

Como se ha mencionado anteriormente, el problema del aprendizaje de intervalos es una parte fundamental del aprendizaje de las RBNT. Ahora presentaremos la relación que existe entre aprender intervalos para una RBNT y discretizar atributos continuos.

El objetivo de la discretización es encontrar un conjunto de puntos de corte para particionar el rango en un número pequeño de intervalos que tengan coherencia (Kotsiantis

y Kanellopoulos 2006). En la figura 3.2.2 se presenta un esquema jerárquico de algunos métodos de discretización. En el esquema se muestra una forma de agrupar los métodos conocidos. En particular se hace una división entre los métodos de unión o fusión (*merge*) y los que usan una técnica de separación (*split*). Ambas categorías se pueden subdividir en métodos supervisados y no supervisados dependiendo si se usa información de las clases. En el siguiente nivel los métodos se dividen en cuanto a la medida usada. En el nivel más bajo se muestran algunos de los algoritmos de discretización más conocidos.

Para aprender las RBNT tenemos eventos que ocurren en el tiempo, por lo que un conjunto de eventos es similar a tener un atributo continuo. Además, el resultado deseado para las RBNT son diversos intervalos que puedan ser considerados como una categoría, lo cual es similar al resultado que se desea en la discretización. Dado lo anterior, podemos ahora tratar el problema de obtener los intervalos temporales como un problema de discretización; sin embargo, no es tan trivial ya que existen varios retos.

- El número de posibles discretizaciones es exponencial en el número de umbrales de corte.
- No existe información de clases, por lo que se necesita un procedimiento de discretización no supervisado y casi no existen métodos de este tipo en la literatura (Liu et al. 2002).
- La información que se podría usar es tomar en cuenta la estructura de la red, sin embargo no siempre se conoce de antemano.

3.2.3. Algoritmo de Friedman

Un trabajo relacionado con discretización y aprendizaje de Redes Bayesianas es el algoritmo presentado en (Friedman y Goldszmidt 1996), el cual realiza una discretización de variables continuas y a la vez realiza un aprendizaje estructural de una red bayesiana. En particular este algoritmo hace uso de un puntaje basado en el concepto de Descripción de Longitud Mínima (Lam et al. 1994), conocido como MDL por sus siglas en inglés. Este

puntaje es la suma de la codificación de la red, los parámetros, y la discretización de las variables continuas.

El algoritmo realiza una búsqueda para cada variable continua para obtener la mejor discretización mediante un algoritmo voraz. El algoritmo para aprender la red funciona de forma iterativa alternando la obtención de una política de discretización y aprendiendo la estructura hasta converger.

Debido a que un cambio local puede afectar el puntaje de forma global, obtener un puntaje óptimo no es tratable computacionalmente y por ello se opta por una estrategia de encontrar máximos locales, maximizando cada nodo a la vez. Aún con ello, dependiendo de la estructura, este algoritmo puede ser tardado si las variables continuas interactuan entre ellas. Una variable se dice que interactua con otra cuando se encuentra en su cobija de Markov. La cobija de Markov de un nodo se forma por sus padres, sus hijos y otros padres de los hijos. Para mostrar cómo calcular la fórmula del puntaje MDL usada en el algoritmo, necesitamos definir previamente algunos conceptos. Una política de discretización Λ es el conjunto de discretizaciones λ_i para cada $X_i \in U_{cont}$, donde U_{cont} es el conjunto de variables continuas. Usando la política de discretización se obtiene $X_i^* = f_{\lambda_i}(X_i)$ que corresponden a la variable X_i ya discretizada.

Friedman y Goldszmidt 1996 muestran que el puntaje de una Red Bayesiana discretizada corresponde a la ecuación 3.1 . Este puntaje es básicamente la suma de la descripción de la estructura de la red G, la política de la discretización Λ y la descripción de los datos D.

$$DL^*(G, \Lambda, D) = DL_{red}(U^*, G) + DL_{\Lambda}(\Lambda) - N \sum_i I(X_i^*; \pi_{Xi}) \tag{3.1}$$

donde:

$$DL_{red}(U, G) = \sum_i (log||X_i|| + (1 + |\pi_{Xi}| \text{ log n}) + \frac{\text{log N}}{2} \sum_i ||\pi_{Xi}||(||X_i|| - 1)$$

$$DL_{\Lambda}(\Lambda) = \sum_{X_i \in U_{cont}} (N_i - 1) H(\frac{k_i - 1}{N_i - 1})$$

$I(A, B)$ se refiere a información mutua, $H(A)$ es la función de entropía, π_{Xi} son los padres del nodo Xi, el valor $k_i = ||X_i^*||$ es la cardinalidad de la discretización de X_i, n es el número de variables (nodos) y N es el número de instancias (datos).

Algoritmo 3.1: Algoritmo de Friedman

Datos: Una discretización inicial

Resultado: Una discretización con puntajes óptimos locales

Agregar a la cola Q todas las variables continuas

mientras Q *no este vacía* **hacer**

  Eliminar de la cola el Q primer elemento X

  Obtener una nueva política de discretización λ'_X para X

  si $DL*(G, \Lambda[\lambda'_X], D) < DL*(G, \Lambda, D)$ **entonces**

    Reemplazar λ_X por λ'_X

    Para todo Y interactuando con X, si $Y \notin Q$, encolar Y en Q.

  fin si

fin mientras

regresar Λ

En el Algoritmo 3.1 se muestra el pseudocódigo del proceso de discretización de varias variables para una RB. El algoritmo inicia agregando a una cola todas la variables continuas, después para cada variable de la cola se obtiene una discretización mediante un proceso voraz y se agregan a la cola las variables que interactuen con la variable discretizada. Este proceso termina cuando la cola esté vacía.

3.3. Resumen

En este capítulo se presentó el modelo de Redes Bayesianas de Nodos Temporales así como los algoritmos existentes relacionados con su aprendizaje. En particular se mencionaron dos algoritmos, el primero es el presentado en (Liu, Song y Yao 2005) el cual tiene las

limitantes de que para ser usado los datos deben de provenir de una base de datos temporal probabilista y además no realiza un aprendizaje de intervalos. Nuestro algoritmo, no necesita que los datos provengan de algún tipo especial de base de datos y sí realiza un aprendizaje de los intervalos para cada nodo temporal. El segundo algoritmo relacionado es el presentado en (Friedman y Goldszmidt 1996), una limitante es que este algoritmo no fue específicamente diseñado para aprender RBNTs, ademas de que puede ser computacionalmente complejo cuando la red en la que se aplica contiene muchos arcos. Debido a ello, en esta tesis se propone un algoritmo de aprendizaje para RBNTs, el cual se presentará en el siguiente capítulo.

Capítulo 4

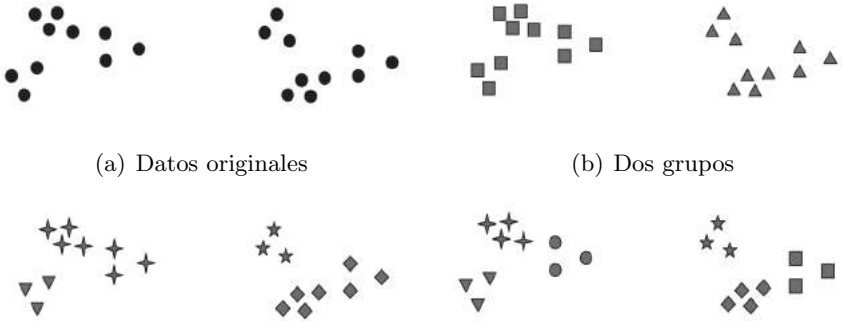
Algoritmo de aprendizaje de RBNT

En este capítulo se presenta la parte central de la tesis: es decir, el algoritmo de aprendizaje para RBNT. Debido a que parte del algoritmo se basa en técnicas de agrupamiento, se incluyen algunos conceptos y algoritmos importantes de esta área, en particular se describe el algoritmo de agrupamiento *K-means* y el modelo de mezcla de Gausianas. Posteriormente se presenta el algoritmo de aprendizaje de intervalos y el algoritmo de aprendizaje estructural. Finalmente se presenta un ejemplo sencillo de la aplicación del algoritmo.

4.1. Agrupamiento

Una definición de agrupamiento es: *aquella tarea que divide a los datos en grupos significativos, por lo tanto, los grupos resultantes deben capturar la estructura "natural" de los datos* (Tan, Steinbach, Kumar et al. 2006). Sin embargo, lo que constituye un grupo no está bien definido y en muchas aplicaciones los grupos "naturales" no son fáciles de encontrar.

Para entender mejor la dificultad de lo que constituye un grupo consideremos la figura 4.1, la cual muestra 20 puntos y tres diferentes maneras en las que se pueden dividir en



(a) Datos originales (b) Dos grupos

(c) Cuatro grupos (d) Seis grupos

Figura 4.1: Datos y diferentes grupos encontrados

grupos. Dependiendo del punto de vista de cada persona alguno de los tres resultados presentados debería de ser el mejor. Esto muestra que la definición de lo que es un grupo es ambigua y que el mejor agrupamiento depende del tipo de datos y los resultados esperados.

4.1.1. Algoritmo *K-means*

El algoritmo *K-means* es un algoritmo de agrupamiento simple y muy conocido, el pseudocódigo del mismo se presenta en el algoritmo 4.1.

Algoritmo 4.1: Algoritmo *K-means*

Datos: Número de grupos a buscar K, datos D
Resultado: K grupos
Seleccionar K puntos de los datos D como centroides iniciales.
mientras *Centroides sigan cambiando* **hacer**
  Asignar todos los puntos al centroide más cercano
  Recalcular el centroide de cada grupo
fin mientras

El algoritmo *K-means* incluye, básicamente, una inicialización de centroides y posteriormente se entra en un ciclo, dentro de él se realizan dos pasos: asignar los puntos al centroide más cercano y recalcular el centroide, este proceso converge a un máximo local. También es importante mencionar que es usado muy a menudo para inicializar los parámetros del modelo de mezcla de Gausianas (Bishop 2006), antes de aplicar el algoritmo EM .

4.1.2. Modelo de Mezcla de Gausianas

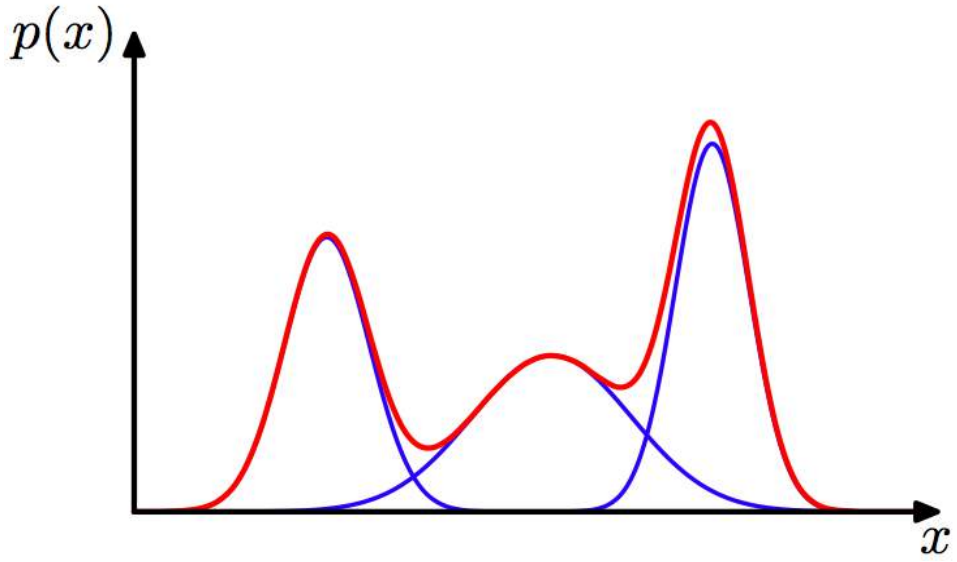


Figura 4.2: Una Mezcla de Gausianas, en azul se muestran las differentes Gausianas por separado y en rojo la suma de ellas.

Un modelo de agrupamiento similar a *K-means* pero que tiene bases estadísticas es llamado Modelo de Mezcla de Gausianas (MMG). La idea de este modelo es considerar que los datos son generados por una mezcla de diferentes distribuciones. La suposición que generalmente se hace para simplificar el modelo es que las distribuciones son Gausianas.

De manera formal, el modelo de mezcla de Gausianas es una función de densidad de probabilidad parametrizada por la suma de varias Gausianas. Un ejemplo de un MMG formado por 3 distribuciones se presenta en la figura 4.2.

Un MMG se describe formalmente por:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2) \tag{4.1}$$

donde cada $\mathcal{N}(x|\mu_k, \sigma_k^2)$ es una Gausiana, parametrizada por su media μ_k y su varianza σ_k^2. Los parámetros π_k son llamados coeficientes de mezcla y deben de cumplir $0 \leq \pi_k \leq 1$ y

$\sum_{i=1}^{k} \pi_k = 1$.

La forma de la Mezcla de Gausianas está determinada por los parámetros $\lambda = \{\pi, \mu, \sigma^2\}$. El problema que se tiene comúnmente es que dados varios datos (asumimos que son datos generados por una mezcla de Gausianas) deseamos estimar los parámetros que mejor se ajusten a un MMG. Existen varios algoritmos (Day 1969) para estimar los parámetros, sin embargo, la forma más común de hacerlo es mediante estimación de máxima verosimilitud.

El objetivo es encontrar los parámetros del modelo λ que maximicen la verosimilitud del MMG dados los datos de entrenamiento. Para una secuencia de datos de entrenamiento $X = \{x_i, \ldots, x_T\}$ y asumiendo independencia entre ellos, entonces:

$$p(X|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda) = L(\lambda|X)$$

donde $L(\lambda|X)$ es llamada la función de verosimilitud de los parámetros (λ) dados los datos. Esta verosimilitud es una función que depende de los parámetros, en donde los datos usados se mantienen fijos. El problema de estimar la máxima verosimilitud es encontrar los parámetros que maximicen $L(\lambda|X)$. Esta función no siempre es maximizable de forma analítica, sin embargo los parámetros pueden ser estimados mediante un caso especial del algoritmo iterativo EM.

4.1.3. Algoritmo EM

La idea del algoritmo EM es iniciar con un modelo inicial λ y estimar un nuevo modelo $\bar{\lambda}$ tal que $p(X|\bar{\lambda}) > p(X|\lambda)$. Entonces el nuevo modelo se convierte en el modelo inicial para la siguiente iteración y el proceso se repite hasta converger, esto sucede cuando la diferencia entre los modelos no sea significativa. El pseudocódigo del algoritmo EM para obtener los parámetros de la Mezcla de Gausianas se presenta en el Algoritmo 4.2. El algoritmo consiste de 3 pasos: el primero es inicializar los valores de las medias, las varianzas y los coeficientes de mezcla. Posteriormente se entra en un ciclo de dos pasos, estimar la probabilidad con los parámetros actuales y maximizar los parámetros con la probabilidad calculada en el

paso anterior, este ciclo esta garantizado a converger a un máximo local (Dempster, Laird y Rubin 1977).

Algoritmo 4.2: Algoritmo EM para Mezcla de Gausianas

Inicializar los parámetros μ_k, σ_k^2, π_k
mientras *no converja a máximo local* **hacer**
Estimar probabilidad:

$$Pr = \frac{\pi_k \mathcal{N}(x|\mu_k, \sigma_k^2)}{\sum_{j=1}^{M} \mathcal{N}(x|\mu_k, \sigma_k)}$$

Maximizar parámetros:

$$\bar{\pi_k} = \frac{1}{T} \sum_{t=1}^{T} Pr(i|x_t, \lambda)$$

$$\bar{\mu_k} = \frac{\sum_{t=1}^{T} Pr(i|x_t, \lambda) x_t}{\sum_{t=1}^{T} Pr(i|x_t, \lambda)}$$

$$\bar{\sigma_k^2} = \frac{\sum_{t=1}^{T} Pr(i|x_t, \lambda)(x_t - \bar{\mu_i})^2}{\sum_{t=1}^{T} Pr(i|x_t, \lambda)}$$

fin mientras

4.2. Algoritmo propuesto "LIPS"

A continuación presentaremos el algoritmo propuesto de aprendizaje de RBNT al que denominamos LIPS (*Learning Intervals Parameters and Structure*). Inicialmente se muestra cómo son los datos de entrada que se requieren. Posteriormente se detallan los tres pasos del algoritmo: la inicialización, el aprendizaje de intervalos y el aprendizaje estructural. Finalmente se concluye con un ejemplo de la aplicación del algoritmo.

4.2.1. Datos de entrada

En la tabla 4.1 se puede observar un ejemplo de cómo serían los datos obtenidos para aprender una RBNT. En este ejemplo cada fila corresponde a un caso. Del lado izquierdo,

Colisión	Heridas Cabeza	Heridas Internas	Pupilas dilatadas (minuto de ocurrencia)	Signos vitales (minuto de ocurrencia)
severo	verdadero	graves	14	20
moderado	verdadero	graves	25	25
moderado	verdadero	ligeras	21	8
moderado	falso	falso	32	-
leve	falso	falso	-	-
leve	falso	ligero	-	-
⋮	⋮	⋮	⋮	⋮

Tabla 4.1: Ejemplo de un conjunto de datos para aprender la red de la figura 4.7. Estos datos pertenecen a un accidente automovilístico y algunos de sus efectos.

en las primeras tres columnas, se muestran datos estáticos que corresponderán a nodos instantáneos y del lado derecho se muestran datos continuos (últimas dos columnas), estos corresponderán a nodos temporales a los que se deberán obtener intervalos. En caso de que el evento no ocurrió se muestra el símbolo "-" en la tabla 4.1.

4.2.2. Idea del algoritmo

El algoritmo de aprendizaje propone usar una estrategia parecida a la que usa el algoritmo EM (Expectation-Maximization)(Moon 1996). El algoritmo se compone de dos pasos que se alternan; en un primer paso se aprende una estructura con la información disponible, asumiendo una discretización inicial de los datos, y en un segundo momento se aprenden los intervalos con la estructura obtenida anteriormente.

Algoritmo 4.3: Algoritmo para aprender una RBNT

```
Resultado: Una RBNT
discretizarDatos();
mientras estructura no converja hacer
    aprenderEstructura();
    afinarIntervalos();
fin mientras
```

Colisión	Heridas Cabeza	Heridas Internas	Pupilas dilatadas (minuto de ocurrencia)	Signos vitales (minuto de ocurrencia)
severo	verdadero	graves	[10-20]	[15-30]
moderado	verdadero	graves	[20-30]	[15-30]
moderado	verdadero	ligeras	[20-30]	[0-15]
moderado	falso	falso	[30-40]	-
leve	falso	falso	-	-
leve	falso	ligero	-	-
⋮	⋮	⋮	⋮	⋮

Tabla 4.2: Ejemplo de un conjunto de datos ya discretizados de la tabla 4.1

.

Debido a que inicialmente no se tiene más información, se propone usar una estimación inicial de los intervalos temporales. Por ello, como primer paso se aplica un algoritmo que como resultado obtenga varios segmentos, los cuales serán tomados como una primera aproximación a los intervalos de cada Nodo Temporal. Una vez que se tienen los intervalos de los NTs, entonces los datos cambian ligeramente, un ejemplo se muestra en la figura 4.2. Con estos datos es posible aplicar un algoritmo de aprendizaje estructural. Finalmente, ya que se tiene la primera aproximación completa, es decir la estructura con los intervalos, se aplica un nuevo algoritmo que refina los intervalos dada la estructura encontrada. Más aún, este proceso se podría repetir. El pseudocódigo se muestra en el algoritmo 4.3.

4.2.3. Inicio del algoritmo

Como ya se mencionó, al inicio del proceso, no se cuenta con más información que los datos mismos. Además, se ha mencionado que uno de los problemas son los datos continuos y la obtención de los intervalos temporales. Por lo tanto el objetivo es obtener una primera aproximación a los intervalos de los nodos temporales en base a los datos. Para nuestros experimentos se consideraron 2 formas de obtener los intervalos iniciales: (i) una discretización uniforme y (ii) una discretización basada en el algoritmo *K-means*.

Discretización Uniforme

La aproximación más simple y rápida es aplicar una discretización uniforme en cada uno de los nodos temporales y así obtener intervalos. Para realizar una discretización uniforme sólo es necesario determinar el parámetro n, número de intervalos, y los datos de entrada. Posteriormente se deben encontrar los $n-1$ puntos de división en el rango de los datos, para ello se ordenan los datos y se obtienen los puntos de corte de tal forma que todos sean del mismo tamaño.

K-Means

La discretización uniforme es un método muy simple pero que no usa información para determinar los intervalos. Por ello, una mejor aproximación es hacer uso del algoritmo *K-means* presentado en la sección 4.1.1. El resultado de aplicar el algoritmo en los datos serán n puntos que nos indicarán los centroides encontrados en los datos. Posteriormente podemos convertir esos centroides en intervalos temporales, esto se realiza mediante el algoritmo 4.4.

Algoritmo 4.4: Algoritmo basado en *K-means* para obtener los intervalos iniciales.

Datos: Puntos ordenados *Pi* obtenidos mediante el algoritmo *k-means*, valores continuos *data* del nodo n.
Resultado: Conjunto de intervalos iniciales para el nodo n

```
min=mínimo(data)
max=máximo(data)
Intervalo[0].inicio=min
Intervalo[0].fin=promedio(Pi[0],Pi[1])
para i=0 to size(Pi)-2 hacer
    Intervalo[i+1].inicio=promedio(Pi[i],Pi[i+1])
    Intervalo[i+1].fin=promedio(Pi[i+1],Pi[i+2])
fin para
i=size(Pi)-1
Intervalo[i].inicio=promedio(Pi[i],Pi[i+1])
Intervalo[i].fin=max
```

El algoritmo 4.4 toma como parámetros el resultado del algoritmo *K-means*, es decir un conjunto ordenado de centroides P_i y el conjunto de datos. El primer paso es obtener los

valores mínimo y máximo del conjunto de los datos. Posteriormente se obtiene el primer intervalo el cual se conforma con el valor mínimo del conjunto de datos y el promedio de los primeros dos puntos del conjunto de centroides. Después el algoritmo entra en un ciclo para obtener los $P_i - 2$ intervalos, cada intervalo se compone del promedio de puntos consecutivos obtenidos por el algoritmo *K-means*. El ciclo termina y se obtiene el intervalo final el cual se conforma con el promedio de los dos últimos centroides y el valor máximo del conjunto de datos. El algoritmo obtiene como resultado un conjunto de intervalos basado en los valores obtenidos por el algoritmo *K-means*.

4.2.4. Algoritmo de aprendizaje de intervalos

Las aproximaciones anteriores sólo hacen uso de los datos, sin embargo, si asumimos que conocemos la estructura de la RBNT, podemos ayudarnos de esa información y mejorar la calidad de los intervalos. El algoritmo presentado a continuación usa información de la red, en particular de los padres de los Nodos Temporales, para encontrar una mejor aproximación a los intervalos.

El algoritmo se presenta en dos etapas: la primera es una versión simple para entender la idea del algoritmo en donde no usamos información de la red, la segunda es la versión completa del algoritmo.

En la figura 4.3 se presenta un esquema general del algoritmo de aprendizaje de intervalos. En la parte (a) de la figura se muestra un histograma de algunos eventos temporales que ocurrieron en un tiempo determinado. Nuestro objetivo es encontrar intervalos como los que se muestran en la parte (d) de la figura. Para ello, el algoritmo asume que los datos (eventos) siguen una distribución normal. Más aún, se usa el Modelo de Mezcla de Gausianas para obtener n distribuciones, parametrizadas por su media y su desviación estándar. Para los datos de la figura 4.3 (a) se obtuvieron las gausianas de la figura 4.3 (b). Posteriormente usamos los parámetros con que se presentan las gausianas para obtener la primera aproximación de los intervalos, los cuales se muestran en la parte (c) de la figura 4.3. Por último se aplica un algoritmo de refinamiento para obtener los intervalos finales, los cuales

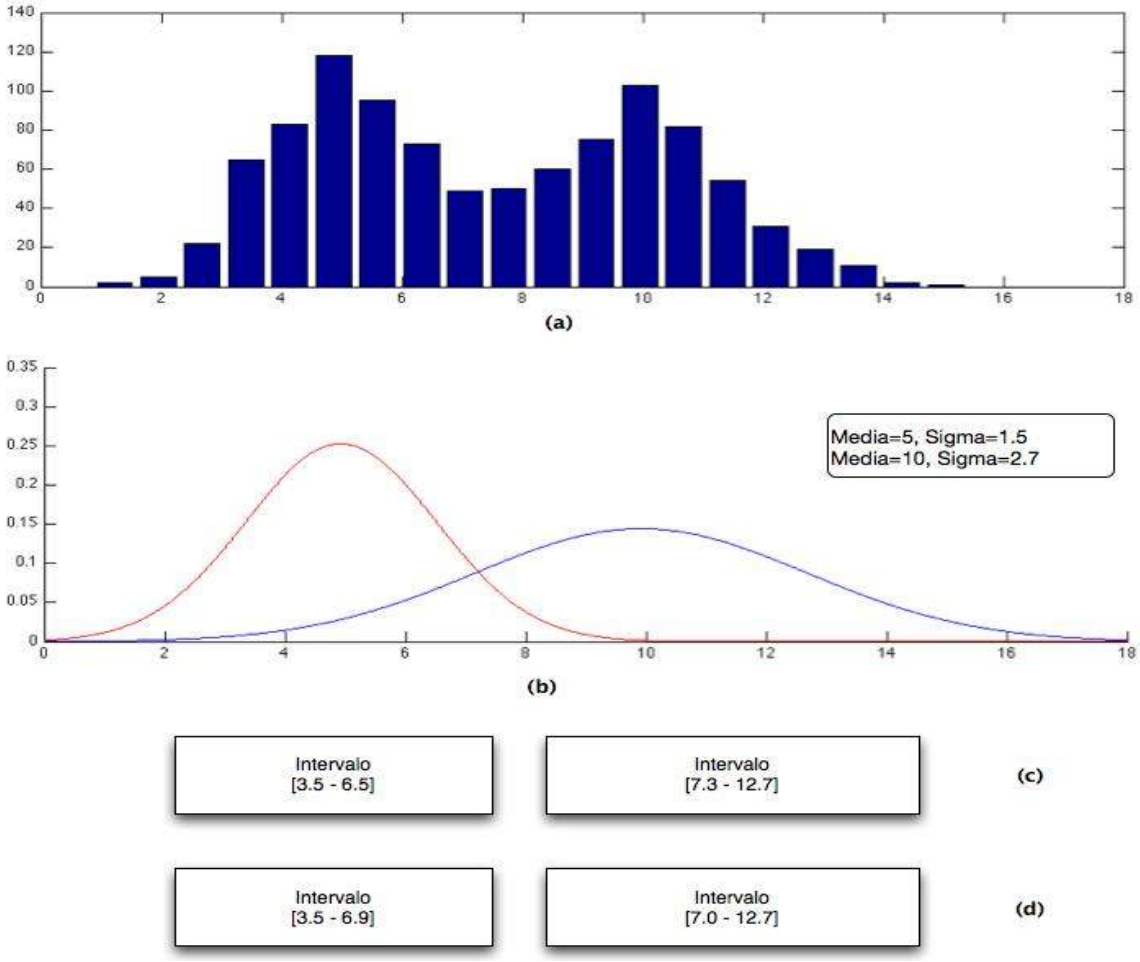


Figura 4.3: (a) histograma de eventos en el tiempo, (b) Modelo de Mezcla de Gausianas obtenidas de los datos temporales, (c) primera aproximación de los intervalos, (d) intervalos finales obtenidos

se muestran en la parte (d) de la figura 4.3. A continuación se detalla el algoritmo de aprendizaje de intervalos, el cual se presenta en dos aproximaciones, la primera es una versión simplificada sin usar información de la red, la segunda es la versión completa que usa la información de los nodos padres para obtener los intervalos.

Primera aproximación:

En primera instancia vamos a considerar a cada nodo temporal de forma independiente y se van a obtener los intervalos en dos fases.

Fase 1: determinando los intervalos Nuestro método usa el modelo de mezcla de gausianas para realizar una aproximación a los datos. Por lo tanto se puede usar el algoritmo (EM) (sección 4.1.3) para obtener los parámetros de las distribuciones. Aplicando el algoritmo EM, se obtienen distintas gausianas (grupos) que son especificados por su media y su desviación estándar.

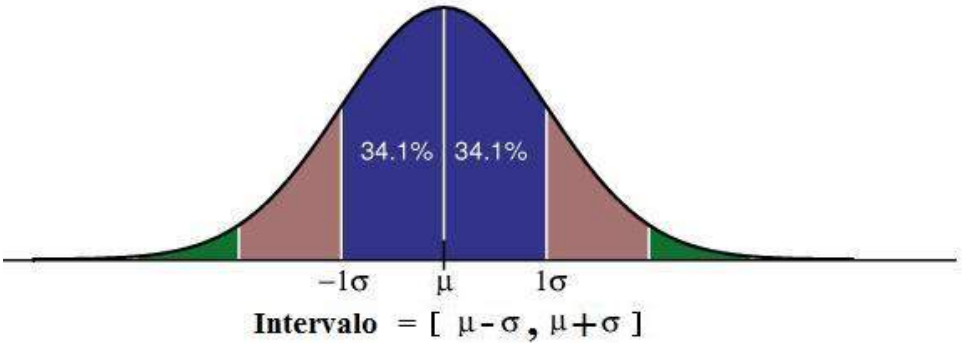


Figura 4.4: La aproximación inicial de los intervalos se obtiene con la media y la desviación estándar.

Por ahora se asume que conocemos el número de intervalos temporales (gausianas) k. Para cada nodo temporal se tiene un conjunto de puntos en el tiempo y estos son agrupados usando el modelo de mezcla de gausianas. Basados en los parámetros de cada gausiana, cada intervalo temporal estará definido por:

$$[\mu - \sigma, \mu + \sigma]$$

tal como se muestra en la figura 4.4.

Fase 2: determinando el número de intervalos En la fase anterior asumimos que conocíamos el número de gausianas, ahora presentaremos cómo encontrar ese número. La solución ideal tiene que cumplir dos condiciones: (i) el número de intervalos debe ser pequeño para reducir la complejidad de la red, y (ii) los intervalos deben de dar buenas

estimaciones cuando se realicen procesos de inferencia sobre la red.

Basándonos en las ideas mencionadas, nuestra aproximación usa el algoritmo EM con el parámetro de número de grupos de 1 a ℓ, donde ℓ es un valor máximo (para nuestros experimentos $\ell = 3$). Para seleccionar el mejor conjunto de intervalos, una evaluación es hecha sobre la red, la cual es una medida indirecta de la calidad de los intervalos. En particular, se usó la puntuación relativa de Brier para medir la exactitud predictiva de la red.

La puntuación de Brier esta definida como

$$PB = \frac{1}{n} \sum_{i=1}^{n} (1 - P_i)^2$$

donde P_i es la probabilidad posterior marginal de el valor correcto de cada nodo dada la evidencia. La puntuación relativa de Brier (PRB) se define como:

$$PRB \ \ (en\,\%) = (1 - PB) \times 100.$$

Este PRB es usado para evaluar la RBNT. El proceso que se sigue es que para cada instancia de datos que se tenga, se selecciona al azar un subconjunto de variables del modelo, a las variables de este subconjunto se les asigna el valor que se tenga de los datos, posteriormente se aplica inferencia sobre la red y se predicen las variables no asignadas y se compara con el valor que se tenga en los datos, con esto se obtiene el PRB para esas predicciones. Este proceso se aplica a los diferentes conjuntos de intervalos y se selecciona el conjunto que obtenga el mayor PRB.

Segunda aproximación: considerando la topología de la red

La primera aproximación no usa información alguna de la topología de la RBNT. No obstante, podemos usar información de los padres de cada nodo temporal para obtener una mejor aproximación basada en la primera.

Para la nueva aproximación usaremos las configuraciones de los padres. El número de

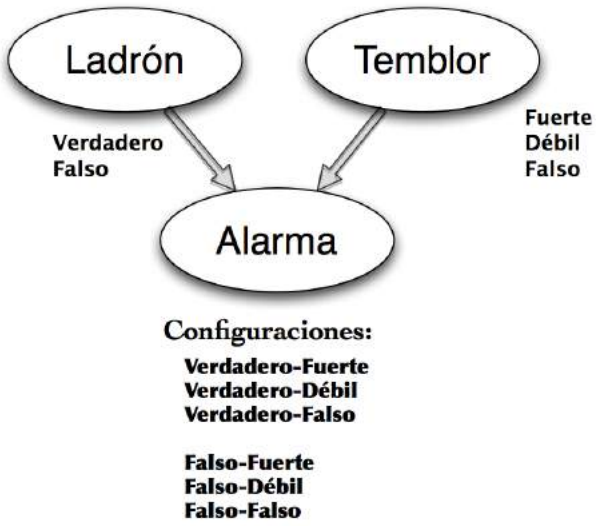


Figura 4.5: Una red bayesiana con 3 nodos. Para los nodos raíz se muestran los estados, para el nodo Alarma se muestran las configuraciones posibles de los padres.

configuraciones de cada nodo i es $q_i = \prod_{X \in Pa(i)} |s_X|$ (esto es el producto del número de estados de cada nodo padre). Un ejemplo de las configuraciones se muestra en la figura 4.5.

Formalmente, se construirán particiones de los datos (conjuntos disjuntos de valores) para cada configuración. Con las distintas particiones construimos las combinaciones binarias (por simplicidad), tomando por ejemplo particiones p_i,p_j del total, ver figura 4.6 (a). Lo anterior genera $q(q-1)/2$ diferentes combinaciones de particiones. Cabe mencionar que sólo se usaron combinaciones binarias para reducir la complejidad del algoritmo. Para cada p_i y p_j se aplica la primera aproximación (figura 4.6 (b)) para obtener ℓ conjuntos de intervalos para cada partición. Posteriormente se obtiene la combinación de los conjuntos de intervalos, esto se presenta en la figura 4.6 (c), lo que genera ℓ^2 conjuntos de intervalos para cada p_i,p_j, por último es necesario ajustar cada conjunto de intervalos para que sean continuos (figura 4.6 (d)) usando el algoritmo 4.5. El proceso descrito anteriormente se presenta gráficamente en la figura 4.6

Por ejemplo, tomando la figura 4.5, el nodo *Alarma* tiene padres: *Ladrón* (con estados $Verdadero, Falso$) y *Temblor* (con estados *Fuerte,Débil,Falso*), existen 6 particiones en total. Posteriormente se toman las combinaciones binarias, con lo cual se obtienen $\binom{6}{2} = 15$

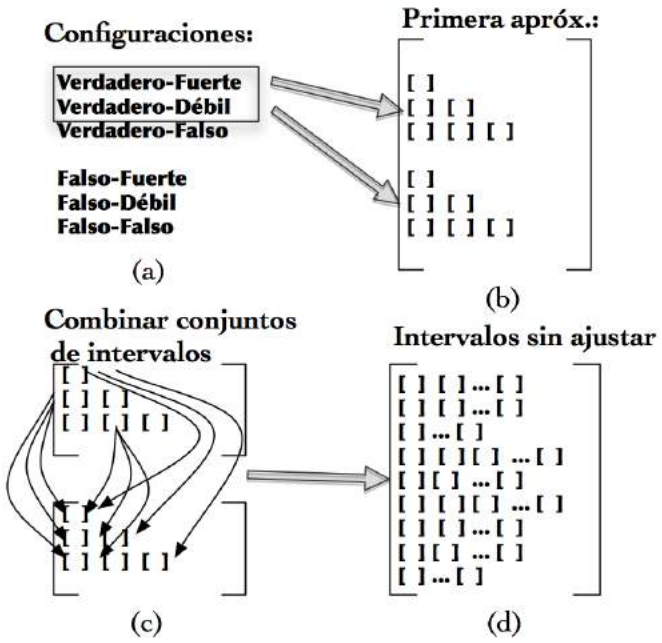


Figura 4.6: Descripción gráfica de la segunda aproximación. En (a) se seleccionan dos particiones. En (b) a cada partición se le aplica la primera aproximación generando dos conjuntos de intervalos. En (c) se combinan los conjuntos de intervalos generados en el paso anterior. El resultado de combinar en (c) se presenta en (d), a estos conjuntos de intervalos se les aplica el algoritmo 4.5.

combinaciones diferentes en total, para cada una de éstas combinaciones binarias se aplica el proceso descrito en la figura 4.6.

Algoritmo 4.5: Algoritmo para ajustar los intervalos.

```
Datos: Conjunto de intervalos
Resultado: Conjunto de intervalos ajustados
para cada conjunto de intervalos s hacer
    ordenarIntervalosPorValorInicial(s);
    mientras Intervalo i este contenido en Intervalo j hacer
        tmp=intervaloPromedio(i,j);
        s.remplazarIntervalo(i,j,tmp);
    fin mientras
    para i=0 número de intervalos en s -1 hacer
        Intervalo[i].fin=Intervalo[i].fin + Intervalo[i+1].inicio /2;
    fin para
fin para
```

El algoritmo 4.5 se detalla a continuación. Para cada conjunto de intervalos éste se ordena por su límite inferior. Posteriormente se verifica si un intervalo está contenido en otro, mientras esto suceda, el algoritmo obtiene un intervalo promedio, tomando la media de los puntos iniciales y la media los puntos finales, este nuevo intervalo reemplaza a los dos anteriores. Después, se deben refinar los intervalos para que sean continuos (para que el final de uno coincida con el inicio del siguiente). Esto se logra obteniendo la media del valor final y el valor inicial de intervalos adyacentes. Un ejemplo del algoritmo se presenta en la sección 4.2.7.

Así como en la primera aproximación, el mejor conjunto de intervalos para cada Nodo Temporal es seleccionado basado en la calidad predictiva medida por el PRB (explicado en la sección 4.2.4). No obstante cuando un NT tiene como padre otro NT, un problema ocurre, los estados del nodo padre no son inicialmente conocidos, así que no es posible aplicar la segunda aproximación de forma directa. Para resolver este problema, los intervalos son seleccionados de una forma secuencial de arriba-abajo de acuerdo a la estructura de la RBNT. Esto quiere decir que primero se seleccionan los nodos del segundo nivel (los nodos raíz siempre son instantáneos por definición). Una vez que estos intervalos fueron

encontrados, entonces se procede a obtener los nodos del tercer nivel, y así sucesivamente hasta llegar a los nodos hoja.

4.2.5. Poda

Obtener las combinaciones y unir los intervalos es computacionalmente complejo, el número de conjuntos de intervalos por nodo está en $O(q^2\ell^2)$, donde q es el número de configuraciones de los padres y ℓ es un parámetro del número máximo de intervalos iniciales a buscar. Por esta razón se usaron dos técnicas de poda las cuales se aplican a cada Nodo Temporal para reducir el tiempo de procesamiento.

La primera técnica discrimina las particiones que proveen poca información al modelo. Para lograr esto, se cuenta el número de instancias en cada partición y si es mayor que un valor

$$\beta = \frac{\text{Número de instancias}}{\text{Número de particiones} \times 2}$$

la configuración es usada, sino se descarta.

La segunda técnica es aplicada cuando se obtienen los intervalos finales para cada combinación. Si el conjunto de intervalos finales contiene un sólo intervalo, entonces no hay información temporal de utilidad y entonces ese intervalo se descarta. Al contrario, cuando se tienen más de α intervalos, entonces se tiene una red demasiado compleja, por lo que de la misma manera el conjunto de intervalos se descarta. En los experimentos se usó el valor $\alpha = 4$.

4.2.6. Aprendizaje estructural

Para aplicar el algoritmo de aprendizaje estructural, lo que se debe hacer es convertir la información temporal en información discreta, de tal forma que un algoritmo de aprendizaje estructural de RB pueda ser usado.

En particular se decidió usar el algoritmo K2 (Cooper y Herskovits 1992) (ver Algoritmo 2.1) debido a que este algoritmo tiene como parámetro un ordenamiento de los nodos, el

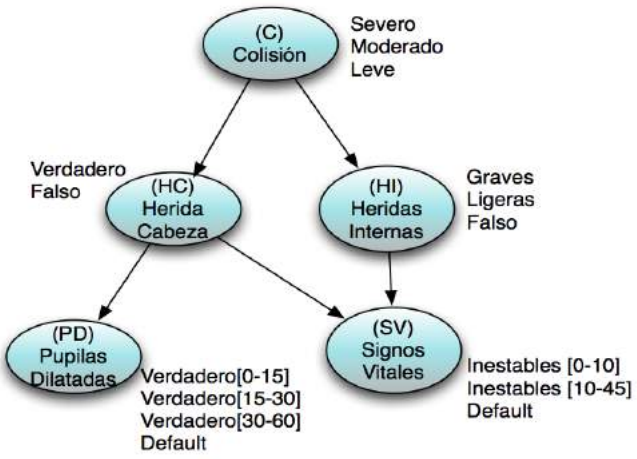


Figura 4.7: Una RBNT que representa un accidente automovilístico y sus posibles consecuencias en el tiempo. Existen 3 Nodos Instantáneos: Colisión, Heridas Cabeza y Heridas Internas. Los otros dos Nodos: Pupilas Dilatadas y Signos Vitales son Nodos Temporales con intervalos asociados.

cual se puede usar para describir la información básica temporal que se tenga del dominio del problema. En particular lo que se realiza es colocar al inicio de la lista las variables estáticas y al final las variables temporales.

4.2.7. Ejemplo del algoritmo

Ahora se mostrará la aplicación del algoritmo en un ejemplo sencillo. Supongamos que tenemos n datos como los de la tabla 4.1. Inicialmente tenemos 2 nodos temporales, Pupilas Dilatadas y Signos Vitales, para ellos aplicamos un algoritmo de discretización uniforme con un número de intervalos predefinido, para este ejemplo usamos 3 intervalos, con lo que obtenemos [1- 20], [21-40], [41-60] para PD y [1-15], [16-30], [31-45] para SV. De esta forma ya tenemos datos como los de la tabla 4.2.

Ahora con ellos ya podemos aplicar un algoritmo de aprendizaje estructural de RB, tal como el algoritmo K2. Para aplicar este algoritmo se debe de especificar un orden de los nodos, en particular esto resulta útil para las RBNT ya que tenemos cierta información temporal ya conocida (tales como los eventos instantáneos, C, HC y HI en este ejemplo, y los que se desencadenarán después PD y VS). Por ello definimos el orden C, HC, HI, PD, SV

Tabla 4.3: Intervalos obtenidos para el nodo PD. Hay 3 conjuntos de intervalos por cada partición.

Partición	**Intervalos**
HC=true	$[11-35]$
	$[11-27][32-53]$
	$[8-21][25-32][45-59]$
HC=false	$[3-48]$
	$[0-19][39-62]$
	$[0-14][28-40][47-65]$

(primero las variables estáticas y luego las temporales). Al aplicar el algoritmo obtenemos una red como la de la Figura 4.7.

Ahora falta refinar los intervalos de los nodos temporales. Para ello vamos a aplicar el algoritmo presentado en la sección 4.2.4. Para ejemplificar sólo vamos a mostrar el proceso para el nodo PD.

El nodo PD tiene como padre al nodo HC, el cual tiene dos valores ($Verdadero$ y $Falso$). Entonces podemos hacer una separación de los datos en dos particiones (una por configuración), posteriormente se aplica el algoritmo EM para obtener una mezcla de gausianas con parámetros 1 a 3 como el número de grupos. Esto nos daría 6 conjuntos de intervalos presentados en la Tabla 4.3.

Posteriormente debemos combinar estos conjuntos de intervalos de cada partición con las restantes, por lo que podría haber $3 \times 3 = 9$ conjuntos diferentes de intervalos, sin embargo algunas de ellas son eliminadas de acuerdo al Algoritmo 4.5.

Consideremos los intervalos de la tabla 4.3, tomamos primero $[11-35]$ y $[3-48]$, si los concatenamos obtenemos $[11-35][3-48]$. A este conjunto de intervalos le aplicamos el Algoritmo 4.5. Para este caso debemos ordenar y así obtener $[3-48][11-35]$. Lo siguiente a realizar es verificar si uno de los intervalos está contenido dentro de otro, lo cual es verdadero ($[11-35]$ en $[3-48]$), por lo tanto, se obtiene un nuevo intervalo tomando las medias de los intervalos anteriores, por lo que el nuevo intervalo es $[7-41]$, este intervalo no se agrega a

Tabla 4.4: Intervalos obtenidos para el nodo PD

Intervalos Nodo PD
$[0-15][16-37][38-62]$
$[0-12][13-31][32-43][44-65]$
$[7-37][38-53]$
$[15-39][40-59]$
$[0-13][14-23][24-38][39-60]$

Tabla 4.5: Intervalos obtenidos para el nodo SV

Intervalos Nodo SV
$[0-9][10-38][39-53]$
$[0-6][7-29][30-43][44-53]$

la lista de conjuntos de intervalos finales ya que por la poda se eliminará debido a que sólo tiene un intervalo. Ahora se toman $[11-35]$ y $[0-19][39-62]$. Concatenando obtenemos $[11-35][0-19][39-62]$. Aplicando el Algoritmo 4.5 se ordenan los intervalos para obtener $[0-19][11-35][39-62]$. En este conjunto verificamos que ningún intervalo está contenido en otro así que solamente se refinan los intervalos para obtener $[0-15][16-37][38-62]$. Este es otro conjunto de intervalos finales. Este proceso se repite para obtener los cuatro conjuntos de intervalos mostrados en la tabla 4.4

El proceso anterior se aplica a cada nodo temporal, para este ejemplo se aplicaría al nodo SV y se obtendrían los intervalos mostrados en la tabla 4.5.

Para seleccionar "el mejor" conjunto de intervalos se aplican pruebas de inferencia y se elige el conjunto de intervalos que minimize el puntaje relativo de Brier. En la tabla 4.6 se muestran los intervalos que obtuvieron la mayor puntuación y por tanto los que serán usados en la red, además se muestra los intervalos de la red original. En la figura 4.8 se muestra la representación final de la RBNT aprendida.

Tabla 4.6: Intervalos finales obtenidos para los nodos PD y SV

Nodo PD		**Nodo SV**	
Aprendido	**Original**	*Aprendido*	**Original**
$[0-15]$	$[0-15]$	$[0-9]$	$[0-10]$
$[16-37]$	$[16-30]$	$[10-38]$	$[11-45]$
$[38-62]$	$[30-60]$	$[39-53]$	

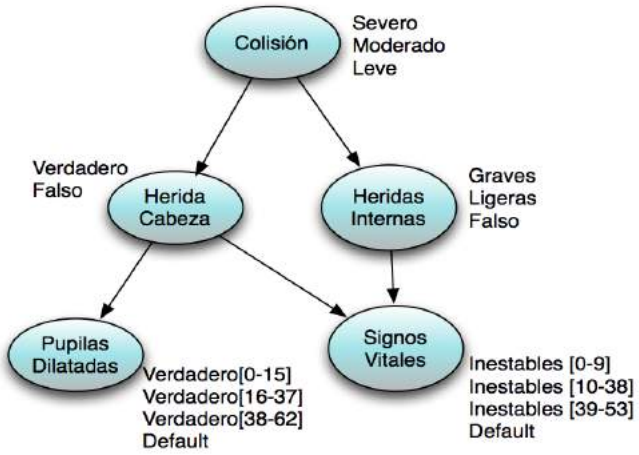


Figura 4.8: La RBNT aprendida descrita en la sección 4.2.7. La estructura y los intervalos fueron obtenidos usando el algoritmo descrito en este capítulo.

4.3. Resumen

En este capítulo se presentó el algoritmo de aprendizaje de Redes Bayesianas de Nodos Temporales. El algoritmo obtiene una aproximación inicial a los intervalos, con lo anterior es posible realizar un aprendizaje estructural haciendo uso del algoritmo K2. Posteriormente, con la estructura inicial el algoritmo hace uso de la información de los padres de los nodos temporales para obtener un modelo de mezcla de gausianas que a su vez se convertirá en diferentes conjuntos de intervalos. Los intervalos que obtengan la mejor calidad predictiva serán usados en el modelo. Este algoritmo de aprendizaje de RBNT es la principal aportación de la tesis. Para evaluar la calidad del algoritmo se realizaron pruebas en 3 RBNTs de diferentes tamaños variando el número de casos y la inicialización, estas pruebas y el análisis de los resultados se presentan el el siguiente capítulo.

Capítulo 5

Experimentos

En este capítulo se presentan los experimentos que se realizaron para evaluar el algoritmo de aprendizaje. En la primera sección se habla de las medidas de evaluación utilizadas. Posteriormente se muestran los experimentos con datos sintéticos con 3 redes de diferentes tamaños.

5.1. Medidas de evaluación

Las diferentes medidas que se usaron para evaluar las redes las podemos dividir en tres tipos: las que evalúan la estructura (el grafo), las que evalúan los intervalos y las que evalúan a la red en general.

5.1.1. Medidas que evalúan la calidad de la estructura

Cuando se tiene una red de referencia (red original) entonces lo que comúnmente se hace para evaluar el aprendizaje de una red es comparar la red aprendida con la red de referencia. Entre más parecida sea la red aprendida con la original mayor puntaje se otorga. En particular, para nuestros experimentos se usaron las siguientes medidas:

- Número de arcos faltantes, respecto a la red de referencia tomando en cuenta la dirección de los arcos.

- Número de arcos sobrantes, respecto a la red de referencia tomando en cuenta la dirección de los arcos.

- Similitud estructural (Wu et al. 2001). Sea C la matriz de adyacencia $n \times n$ del grafo dirigido G, sea $s(C) = \sum_{i,j} c_{i,j}$ la suma de todos los componentes de la matriz C. Más aún, se define la conjunción $\bigwedge$ de dos matrices de adyacencia C y C' como $D = C \bigwedge C'$, con $d_{i,j} = c_{i,j} \wedge c'_{i,j}$. La similitud estructural para dos grafos G y G' se define como $SE(G, G') = \frac{s(C \wedge C')}{s(C)}$, donde C es la matriz de adyacencia de G y C' es la matriz de adyacencia de G'.

La red ideal debería obtener valores de 0 para arcos faltantes y sobrantes y un puntaje de 1 en similitud estructural.

5.1.2. Medidas que evalúan la calidad de los intervalos

Una parte importante de las RBNT son los intervalos que contienen los nodos temporales. Lo que se desea es que los intervalos sean pocos pero suficientes para obtener buenos resultados en la calidad predictiva de la red. En nuestro trabajo se usaron dos medidas de evaluación.

1. Número total de intervalos.

2. Error de predicción temporal. El tiempo esperado se define como

$$t_e = (t_{fin} + t_{ini})/2$$

donde t_{ini} y t_{fin} son los valores iniciales y finales del intervalo correspondiente, por lo que el tiempo esperado es la media del intervalo. El error de predicción temporal se define como

$$error = |t_e - t_{orig}|$$

lo cual es la diferencia del tiempo esperado y el tiempo original (el que se tiene de los

datos). Esto significa que los intervalos más grandes tendrán un error mayor que un intervalo de menor tamaño.

La red ideal debería de tener un número bajo de intervalos (para reducir la complejidad) y un valor cercano a 0 en el error de predicción temporal (para mostrar que los intervalos son representativos de los datos).

5.1.3. Medidas que evalúan la red en general

Una medida para evaluar de forma indirecta la calidad de la red es conocida como Puntaje de Brier. Esta medida se define como

$$PB = \sum_{i=1}^{N} (1 - P(v_i|\mathbf{e}))^2$$

lo cual es la suma del error cuadrático de la probabilidad inferida para un nodo v_i con la evidencia dada $\mathbf{e}$. Posteriormente se puede normalizar este puntaje obteniendo el Puntaje Relativo de Brier (PRB):

$$PRB(en\,\%) = (1 - PB) \times 100$$

El PRB es una medida de la calidad predictiva de la red representada en porcentaje (el porcentaje 100 representaría una predicción perfecta).

La forma en la que se usó en los experimentos para evaluar una red con N nodos es seleccionar aleatoriamente un subconjunto s de tamaño $1 \leq |s| \leq N-1$. Posteriormente los nodos de s se asignan de acuerdo a los datos originales y se aplica inferencia a la red para obtener los valores de los nodos restantes y así obtener el PRB.

5.1.4. Prueba de Kruskal-Wallis con correción de Bonferroni

Para evaluar de forma estadística si los resultados obtenidos por nuestro algoritmo eran mejores que los resultados de otros algoritmo se usó la prueba de Kruskal-Wallis, la cual se detalla a continuación.

En estadística, la prueba de Kruskal-Wallis es un método no paramétrico para probar si un grupo de datos proviene de la misma población. La prueba de Kruskal-Wallis no asume normalidad en los datos y es una extensión de la prueba de Mann-Whitney-Wilcoxon para 3 o más grupos.

Para aplicar esta prueba las muestras se combinan y se ordenan de forma incremental y se le da un puntaje (*ranking*). Cuando existe un empate, entonces se obtiene la media de los puntajes. Posteriormente, la suma de los puntajes para cada uno de los K grupos se calcula.

El estadístico esta dado por:

$$H = \left(\frac{12}{N(N+1)} \sum_{j=1}^{K} \frac{R_i^2}{n_j} \right) - 3(N+1) \tag{5.1}$$

donde: n_j es el número de observaciones en el grupo j, R_j es la suma de los puntajes (rankings) del grupo j. N es el número total de observaciones entre todos los grupos.

La hipótesis nula de que las medias son iguales, es rechazada cuando H excede el valor crítico.

Usando Kruskal-Wallis sólo podemos saber si un par de grupos es significativamente diferente, pero no nos dice cual. Para ello, debemos usar una prueba de pares con la corrección de Bonferroni. El ajuste de Bonferroni es simple, sólo se tienen que dividir el α original sobre el número de pares de pruebas. Si se tienen K grupos entonces el número de pares de grupos esta dado por $\ell = K(K-1)/2$, por lo que para cada prueba de pares si el resultado es menor que α/ℓ entonces es significativa.

5.2. Panorama de los experimentos

Los experimentos realizados se hicieron con datos generados de forma sintética de 3 redes de diferentes tamaños: pequeña, mediana y grande[1].

[1]Considaremos una red pequeña aquella con 5 nodos o menos, una red mediana con más de 5 y menos de 10 nodos y una red grande es aquella con más de 10 nodos.

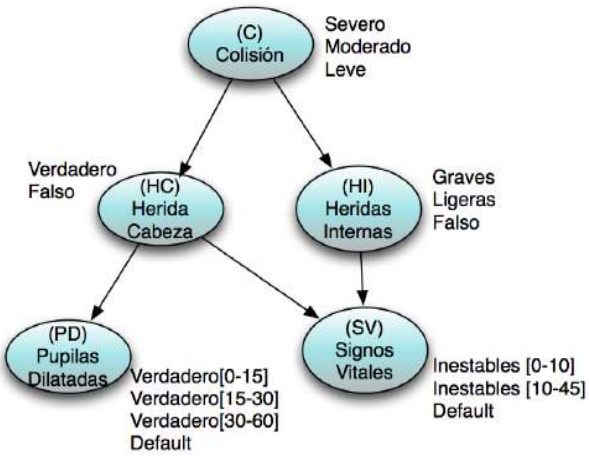


Figura 5.1: Ejemplo de una RBNT pequeña. Esta red representa un modelo simple de un accidente automovilístico, contiene 5 nodos en total y 2 nodos temporales.

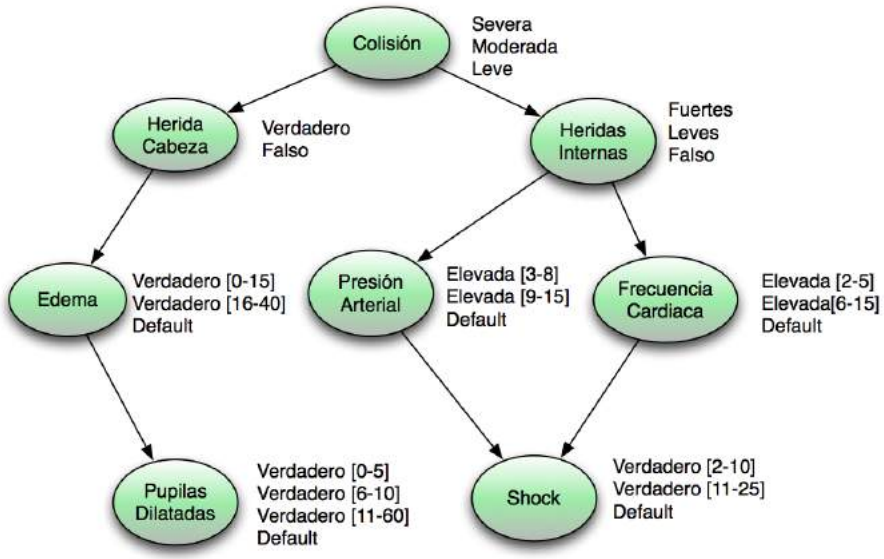


Figura 5.2: Ejemplo de una RBNT mediana. Esta red representa un modelo extendido de un accidente automovilístico, contiene 8 nodos en total y 5 nodos temporales.

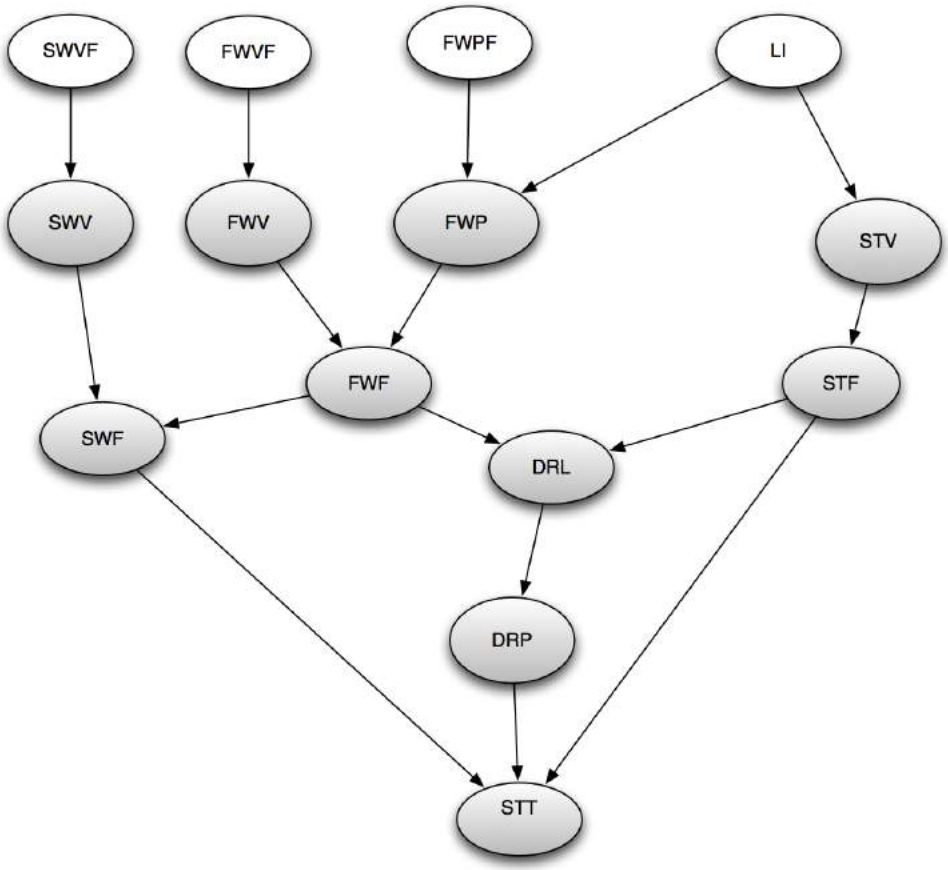


Figura 5.3: Ejemplo de una RBNT grande. Esta red representa un modelo de predicción de fallas en una planta de combustible fósil, contiene 14 nodos en total y 10 nodos temporales (los cuales se muestran en gris). Los intervalos de los nodos temporales se omitieron por claridad.

5.2.1. Generación de los datos

Para realizar los experimentos se generaron datos a partir de las redes de las figuras 5.1-5.3. Estas redes serán las RBNTs de referencia ya que tienen definida la estructura, intervalos y tablas de probabilidad. A partir de estos modelos se generaron datos, en particular para generar los valores de los nodos temporales (los que contienen intervalos), los datos se generaron de dos formas diferentes:

- Basados en una distribución gausiana, usando los parámetros:

$$\mu = \frac{Intervalo\_inicio + Intervalo\_fin}{2}$$

$$\sigma = \frac{Intervalo\_fin - Intervalo\_inicio}{2}$$

- Basados en una distribución uniforme.

5.2.2. Metodología

Para todos los experimentos se siguió la siguiente metodología. Se generaron datos de una red de referencia, para los datos temporales se generaron datos de acuerdo a dos distribuciones: Gausiana y Uniforme. Se usaron 4 algoritmos diferentes en cada uno de los experimentos:

1. Discretización Uniforme (D-U). Corresponde a obtener los intervalos temporales con un algoritmo de discretización uniforme. Este algoritmo tiene como parámetro el número de intervalos a crear. Este algoritmo es uno de los algoritmos base debido a su simplicidad.

2. *K-means* (K-M). Corresponde a obtener los intervalos temporales con un algoritmo basado en *K-means* presentado en el algoritmo 4.4. Este algoritmo tiene como parámetro el número de grupos a buscar. Este es un segundo algoritmo base debido a su simplicidad.

3. Algoritmo presentado en Friedman y Goldszmidt 1996 (Fried.). Este algoritmo realiza una discretización de los atributos continuos mientras aprende la estructura de la red, se presentó en la sección 3.2.3.

4. Algoritmo propuesto en esta tesis (LIPS).

Para los experimentos se varió el número de casos de entrada para evaluar cómo se comportaban los algoritmos con diferente número de datos. Además, se varió el parámetro inicial de los algoritmos (número de intervalos iniciales) de 2 a 4 para todos los experimentos, estos valores se seleccionaron debido a que deseamos que las redes iniciales no contengan un número excesivo de intervalos. Cada experimento se repitió 10 veces.

En las tablas presentadas, además de mostrar los mejores resultados en negritas, se hicieron pruebas de significancia estadística, en particular se realizó la prueba de Kruskal-Wallis con corrección de Bonferroni (presentada en la sección 5.1.4) con un valor $\alpha = 0{,}05$. Para estas pruebas se compararon los 4 algoritmos mencionados anteriormente. En particular se hicieron pruebas de nuestro algoritmo contra los tres restantes por lo que se muestra un '$*$', si el resultado de nuestro algoritmo es significativamente mejor que D-U, '†', si es significativamente mejor que *K-means* y un '§', si es significativamente mejor que el algoritmo de Friedman.

5.3. Red pequeña

El primer conjunto de experimentos se realizó con datos generados de la red de la figura 5.1. El objetivo de este experimento es evaluar los algoritmos con una red muy sencilla.

En las tablas 5.1 y 5.2 se muestran los resultados de diversos experimentos usando datos generados por distribución Gausiana o Uniforme respectivamente. Para cada una de las formas anteriores se presentan 6 medidas de evaluación presentadas en la sección 5.1. La columna Algoritmo presenta los 4 diferentes algoritmos a evaluar los cuales se mencionaron en la sección 5.2.2. La columna Núm-C representa el número de casos de entrada para aprender la red, para esta red empezó en 200 casos aumentando de 100 en 100 hasta llegar

Tabla 5.1: Resultados obtenidos para la red de la figura 5.1 con datos generados con una distribución Gausiana. Se muestran experimentos con diferente número de casos de 200 a 1000. Para cada una de esas filas se muestra el promedio de variar el número de intervalos iniciales, además cada experimento se repitió 10 veces. Se muestra los resultados de arcos añadidos (A+), arcos faltantes (A-), similitud estructural (S-E), puntaje relativo de brier(PRB), error temporal (E-T) y número de intervalos (#Int).

Gausiana							
Núm-C	Algoritmo	A+	A-	S-E	PRB	E-T	#Int
200	D-U	0.33	3.00	0.87	76.06	**18.78**	6.00
	K-M	**0.00**	**2.87**	0.93	73.99	19.55	6.00
	Fried.	1.00	3.00	0.69	**80.99**	19.22	**3.33**
	LIPS	**0.00**∗§	**2.87**	**1.00**†	77.22*	18.98	5.80
300	D-U	**0.00**	2.00	0.96	76.52	17.34	6.00
	K-M	**0.00**	2.00	**1.00**	73.22	18.33	6.00
	Fried.	0.33	2.00	**1.00**	**78.87**	19.48	**3.33**
	LIPS	**0.00**	**1.60**∗†§	**1.00**	78.37†	**14.38**∗†§	5.80
400	D-U	0.13	0.73	0.98	76.93	18.42	6.00
	K-M	**0.00**	0.53	0.96	75.42	17.66	6.00
	Fried.	0.33	1.40	**1.00**	**79.50**	19.43	**4.00**
	LIPS	**0.00**§	**0.13**∗§	**1.00**	77.55	**17.61**§	5.93
500	D-U	0.20	2.00	0.97	77.48	17.22	6.00
	K-M	0.07	2.00	0.98	75.14	17.16	6.00
	Fried	**0.00**	**1.00**	**1.00**	**79.33**	21.46	**4.00**
	LIPS	**0.00**	1.87	**1.00**	78.00	**15.71**§	5.73
1000	D-U	0.07	1.00	0.98	77.66	17.97	6.00
	K-M	0.07	1.00	0.98	75.14	18.13	6.00
	Fried	**0.00**	1.00	**1.00**	**81.64**	18.23	4.00
	LIPS	**0.00**	**0.47**∗†§	**1.00**	78.43	**16.74**†§	5.80
Promedio	D-U	0.15	1.75	0.95	77.05	17.95	6.00
	K-M	0.03	1.68	0.97	74.58	18.17	6.00
	Fried.	0.33	1.68	0.94	**80.07**	19.57	**3.73**
	LIPS	**0.00**§	**1.39**∗	**1.00**∗§	77.98†	**16.68**∗†§	5.81

Tabla 5.2: Resultados obtenidos para la red de la figura 5.1 con datos generados basados en una distribución Uniforme. Se muestran experimentos con diferente número de casos de 200 a 1000. Para cada una de esas filas se muestra el promedio de variar el número de intervalos iniciales, además cada experimento se repitió 10 veces.

Uniforme							
Núm-C	Algoritmo	A+	A-	S-E	PRB	E-T	#Int
200	D-U	0.20	3.00	0.95	73.94	18.84	6.00
	K-M	0.67	**2.33**	0.83	74.14	**16.88**	6.00
	Fried.	**0.00**	**2.33**	**1.00**	**79.98**	17.95	**4.00**
	LIPS	**0.00**∗†	**2.33**∗	**1.00**∗†	73.23	18.15	5.33
300	D-U	0.33	**1.67**	0.93	74.31	18.85	6.00
	K-M	**0.00**	2.00	**1.00**	73.76	19.24	6.00
	Fried.	1.00	2.00	0.75	**77.81**	18.84	**4.00**
	LIPS	**0.00**∗§	**1.67**	**1.00**∗§	75.72	**17.97**∗†	4.33
400	D-U	0.20	**1.00**	0.93	74.55	19.12	6.00
	K-M	0.13	**0.67**	0.96	74.09	18.47	6.00
	Fried.	0.20	1.00	0.96	**81.23**	**16.87**	**4.33**
	LIPS	**0.07**∗	**0.67**	**0.98**∗	77.19†	17.57	**4.33**
500	D-U	0.07	1.67	0.98	73.93	18.06	6.00
	K-M	0.27	1.67	0.93	74.84	18.36	6.00
	Fried	**0.00**	**1.00**	**1.00**	75.57	18.44	4.67
	LIPS	**0.00**†	1.33	**1.00**†	**76.98**∗†	**17.83**†§	**4.33**
1000	D-U	**0.00**	1.20	1.00	74.73	18.51	6.00
	K-M	0.13	1.13	**0.97**	74.92	18.89	6.00
	Fried	**0.00**	**1.00**	1.00	74.24	18.38	5.00
	LIPS	**0.00**†	**1.00**∗	1.00†	**77.64**∗†§	**17.48**†	**4.00**
Promedio	D-U	0.16	1.71	0.96	74.29	18.67	6.00
	K-M	0.24	1.56	0.94	74.35	18.37	6.00
	Fried.	0.24	1.46	0.94	**77.77**	18.10	**4.40**
	LIPS	**0.01**∗†§	**1.40**∗	**0.99**∗†§	76.15∗†	**17.80**∗†	4.47

Tabla 5.3: Resultados de la red de la figura 5.1 variando el número de intervalos iniciales.

		Gausiana						Uniforme					
I-Ini.	Algoritmo	A+	A-	S-E	PRB	E-T	#Int	A+	A-	S-E	PRB	E-T	#Int
2	D-U	0.40	1.64	0.90	79.12	15.78	**4.00**	0.08	1.44	0.98	76.19	16.94	**4.00**
	K-M	**0.00**	1.76	0.96	78.68	16.83	**4.00**	**0.00**	1.60	**1.00**	77.25	17.77	**4.00**
	Fried.	0.20	1.72	0.95	**80.10**	19.54	**4.00**	0.20	1.60	0.95	**77.80**	18.11	4.20
	LIPS	**0.00**	**1.36**	**1.00**	77.76	**16.59**	5.92	0.04	**1.20**	0.99	75.88	**18.09**	4.60
3	D-U	**0.00**	1.80	0.99	77.41	17.96	6.00	0.20	1.84	0.95	73.92	19.99	6.00
	K-M	**0.00**	1.60	**1.00**	73.57	18.90	6.00	0.40	1.48	0.90	73.41	18.95	6.00
	Fried.	0.60	1.72	0.93	**80.05**	19.49	**3.60**	0.20	**1.60**	0.95	**77.96**	18.17	**4.40**
	LIPS	**0.00**	**1.44**	**1.00**	77.87	**16.86**	5.76	**0.00**	**1.60**	**1.00**	76.20	**17.73**	**4.40**
4	D-U	0.04	1.80	0.96	74.27	20.10	8.00	0.20	1.84	0.95	72.77	19.09	8.00
	K-M	0.08	1.68	0.95	71.50	18.77	8.00	0.32	1.60	0.92	72.39	18.38	8.00
	Fried.	0.20	1.60	0.93	**80.06**	19.66	**3.60**	0.32	**1.40**	0.92	**77.54**	18.01	4.60
	LIPS	**0.00**	**1.36**	**1.00**	78.11	**16.60**	5.76	**0.00**	**1.40**	**1.00**	76.38	**17.58**	**4.40**

a 500 y por último se realizó un experimento con 1000 casos. En la última fila se presentan los promedios para esta red.

Algunas conclusiones obtenidas de los resultados son las siguientes:

- El algoritmo propuesto superó a los dos algoritmos base en prácticamente todos los experimentos y en todas las medidas.
- En promedio, el algoritmo propuesto obtuvo los mejores puntajes de evaluación estructural, es decir en número de arcos añadidos, eliminados y similitud estructural ya sea cuando los datos fueron generados por distribución uniforme o gausiana.
- Nuestro algoritmo, en promedio obtuvo el menor error temporal tanto con datos gausianos o uniformes.

Adicionalmente en la tabla 5.3 se presenta otra vista de los datos, en este caso se presentan los resultados por diferente inicialización. La columna I-Ini representa el número inicial de intervalos. Los resultados que se presentan son el promedio de variar el número de casos de 200 a 500 de 100 en 100 y un experimento final con 1000 casos.

De los resultados se puede resaltar que cuando el número de intervalos es pequeño (2) los algoritmos *K-means* y discretización uniforme obtienen los mejores resultados. Conforme se aumenta el número de intervalos iniciales la calidad predictiva disminuye y con 4 intervalos se obtienen los peores resultados. Sin embargo, el algoritmo propuesto es estable y obtiene la mejor calidad estructural y el menor error temporal se obtiene con 3 y 4 intervalos iniciales. Además, es interesante notar que el puntaje predictivo no disminuye de forma signitificativa si se cambia la inicialización.

5.4. Red Mediana

El segundo conjunto de experimentos se realizó con datos de la figura 5.2. Esta red es una extensión de la anterior ya que cuenta con un mayor número de nodos temporales y además existe dependencia entre nodos temporales, cosa que no ocurría en el primer experimento.

En las tablas 5.4 y 5.5 se muestran los resultados. Se pueden destacar las siguientes observaciones:

- El algoritmo propuesto obtuvo en promedio el mejor puntaje en calidad estructural y error temporal, en datos generados de forma gausiana y uniforme.
- Cuando los datos fueron uniformes el algoritmo propuesto obtuvo en promedio el mejor puntaje predictivo y para la distribución gausiana estuvo cerca del mejor resultado.
- El algoritmo propuesto superó en promedio a los algoritmos base en todas las medidas.

En la tabla 5.6 se presentan los resultados por inicialización. Para estos experimentos se varió el número de casos de 200 a 1000 casos.

Con los resultados obtenidos se puede notar que nuevamente cuando el número de intervalos iniciales es 2 los algoritmos base obtienen buenos resultados. Sin embargo, cuando se cambia la inicialización sus puntajes disminuyen notablemente sobre todo en calidad predictiva. El algoritmo propuesto y el de Friedman obtienen los mejores resultados con 3 y 4 intervalos: el algoritmo de Friedman obtiene un número muy bajo de intervalos, por otro

Tabla 5.4: Resultados obtenidos para la red de la figura 5.2 usando datos generados con distribución Gausiana. Se muestran experimentos con diferente número de casos de 200 a 1000. Para cada una de esas filas se muestra el promedio de variar el número de intervalos iniciales, además cada experimento se repitió 10 veces.

Gausiana							
Núm-C	Algoritmo	A+	A-	S-E	PRB	E-T	#Int
200	D-U	1.67	1.00	0.81	78.29	6.98	15.00
	K-M	1.87	0.60	0.80	77.00	6.54	15.00
	Fried.	1.67	**0.13**	0.80	**82.21**	6.75	**10.33**
	LIPS	**1.20**†	0.27∗	**0.86**†	79.27	**6.16**∗§	16.00
300	D-U	0.47	**1.67**	0.94	77.50	7.12	15.00
	K-M	0.20	2.33	0.97	74.89	6.81	15.00
	Fried.	**0.00**	2.00	**1.00**	**81.66**	8.23	**11.33**
	LIPS	**0.00**∗	2.00	**1.00**∗	78.57	**6.77**§	13.27
400	D-U	0.27	1.67	0.97	77.40	6.82	15.00
	K-M	0.13	1.20	0.98	75.61	**6.73**	15.00
	Fried.	**0.00**	**1.00**	**1.00**	**81.02**	9.63	**10.67**
	LIPS	**0.00**	**1.00**∗	**1.00**∗	79.71	6.94§	13.07
500	D-U	0.33	1.67	0.94	76.24	7.48	15.00
	K-M	0.13	1.33	0.98	74.84	**6.76**	15.00
	Fried.	**0.00**	**1.00**	**1.00**	**78.55**	11.33	**10.00**
	LIPS	**0.00**	**1.00**∗	**1.00**∗	78.30	7.41§	13.00
1000	D-U	0.13	0.67	0.97	78.33	7.03	15.00
	K-M	0.07	0.13	0.98	76.27	6.81	15.00
	Fried.	**0.00**	**0.00**	**1.00**	78.73	10.54	**11.00**
	LIPS	**0.00**	**0.00**∗†	**1.00**	**79.74**	**6.79**∗§	13.00
Promedio	D-U	0.57	1.33	0.93	77.55	7.09	15.00
	K-M	0.48	1.12	0.94	75.72	6.73	15.00
	Fried.	0.33	**0.83**	0.96	**80.43**	9.30	**10.67**
	LIPS	**0.24**∗	0.85∗	**0.97**∗	79.12†	**6.69**∗§	13.67

Tabla 5.5: Resultados obtenidos para la red de la figura 5.2 usando datos generados con distribución Uniforme. Se muestran experimentos con diferente número de casos de 200 a 1000. Para cada una de esas filas se muestra el promedio de variar el número de intervalos iniciales, además cada experimento se repitió 10 veces.

Uniforme							
Núm-C	Algoritmo	A+	A-	S-E	PRB	E-T	#Int
200	D-U	1.67	0.67	0.81	79.52	5.92	15.00
	K-M	1.67	0.13	0.83	77.34	6.24	15.00
	Fried.	1.07	**0.00**	0.88	80.98	**6.13**	**10.00**
	LIPS	**1.00**∗	**0.00**∗	**0.89**	**81.43**†	6.16	14.33
300	D-U	0.40	1.67	0.94	76.89	6.77	15.00
	K-M	0.40	1.73	0.95	74.82	6.67	15.00
	Fried.	**0.13**	**1.00**	**0.97**	**79.63**	7.15	**10.00**
	LIPS	**0.13**	**1.00**∗†	**0.97**	79.39	**6.15**∗†§	14.33
400	D-U	0.67	1.00	0.92	77.25	6.89	15.00
	K-M	0.80	**1.00**	0.89	73.01	6.35	15.00
	Fried.	**0.00**	**1.00**	**1.00**	**80.22**	8.08	**10.00**
	LIPS	**0.00**∗†	**1.00**	**1.00**∗†	79.89†	**6.34**∗§	13.00
500	D-U	0.33	1.33	0.94	76.22	7.02	15.00
	K-M	0.33	1.00	0.95	75.37	6.42	15.00
	Fried.	**0.00**	**0.00**	**1.00**	79.73	7.77	**10.00**
	LIPS	**0.00**	1.00	**1.00**	**79.82**∗†	**6.33**§	13.67
1000	D-U	0.07	0.27	0.98	77.87	6.87	15.00
	K-M	0.33	0.33	0.95	75.54	6.67	15.00
	Fried.	**0.00**	**0.00**	**1.00**	81.11	7.97	**10.00**
	LIPS	**0.00**†	**0.00**∗†	**1.00**†	**81.37**∗†	**6.22**∗§	12.00
Promedio	D-U	0.63	0.99	0.92	77.55	6.69	15.00
	K-M	0.71	0.84	0.92	75.21	6.47	15.00
	Fried.	0.24	**0.40**	**0.97**	80.33	7.42	**10.00**
	LIPS	**0.23**∗†	0.60	**0.97**∗†	**80.38**∗†	**6.24**∗†§	13.47

Tabla 5.6: Resultados de la red de la figura 5.2 variando el número de intervalos iniciales.

		Gausiana						Uniforme					
I-Ini.	Alg.	A+	A-	S-E	PRB	E-T	#Int	A+	A-	S-E	PRB	E-T	#Int
2	D-U	0.40	**0.60**	0.95	79.12	7.49	**10.00**	0.56	0.44	0.93	80.78	6.81	**10.00**
	K-M	0.44	0.68	0.96	**80.84**	6.95	**10.00**	0.60	0.44	0.92	79.95	**6.27**	**10.00**
	Fried.	**0.20**	0.84	0.96	80.49	9.07	10.80	**0.24**	**0.40**	**0.97**	**80.52**	7.60	**10.00**
	LIPS	**0.20**	0.84	**0.98**	79.15	**6.90**	12.72	**0.24**	0.60	**0.97**	80.29	6.31	13.20
3	D-U	0.64	1.20	0.93	78.43	6.96	15.00	0.48	0.92	0.94	77.30	6.81	15.00
	K-M	0.48	1.08	0.95	75.36	**6.68**	15.00	0.68	0.88	0.92	74.23	6.76	15.00
	Fried.	0.40	**0.84**	0.96	**80.53**	8.95	**10.20**	**0.24**	**0.40**	**0.97**	80.15	7.52	**10.00**
	LIPS	**0.20**	**0.84**	**0.98**	79.04	6.86	13.28	**0.24**	0.60	**0.97**	**80.49**	**6.20**	13.20
4	D-U	0.68	2.20	0.90	75.11	6.81	20.00	0.84	1.60	0.89	74.58	6.47	20.00
	K-M	0.52	1.60	0.93	70.96	**6.56**	20.00	0.84	1.20	0.90	71.46	6.38	20.00
	Fried.	0.40	**0.80**	**0.96**	**80.29**	9.88	**11.00**	0.24	**0.40**	0.97	80.33	7.13	**10.00**
	LIPS	**0.32**	0.88	**0.96**	79.15	6.70	15.00	**0.20**	0.60	**0.98**	**80.36**	**6.22**	14.00

lado, el algoritmo propuesto obtiene la mejor calidad estructural y el menor error temporal. Estos dos algoritmos son estables debido a que no tienen una gran variación en su calidad predictiva con diferentes inicializaciones.

5.5. Red grande

El tercer conjunto de experimentos se realizó con datos de la red de la figura 5.3. Esta red fue presentada en (Galán et al. 2007) y corresponde a una red para diagnosticar fallas en una planta de combustible fósil. Es una red mucho más compleja que cuenta con 10 nodos temporales y 4 nodos instantáneos. Debido a esto el número de casos con los que se aprendió la red fue aumentado, empezando en 250 casos y finalizando en 3000 casos.

En las tablas 5.7 y 5.8 se muestran los resultados. De los resultados se pueden obtener la siguientes conclusiones:

- El algoritmo propuesto obtuvo en promedio el mejor puntaje en calidad estructural y error temporal, en datos generados de forma gausiana y uniforme.
- El algoritmo obtuvo la mejor puntuación en arcos añadidos en todos los experimentos

Tabla 5.7: Resultados obtenidos para la red de la figura 5.3 usando datos generados con distribución Gausiana. Se muestran experimentos con diferente número de casos de 250 a 3000. Para cada una de esas filas se muestra el promedio de variar el número de intervalos iniciales, además cada experimento se repitió 10 veces.

Gausiana							
Núm-C	Alg.	A+	A-	S-E	PRB	E-T	#Int
250	D-U	0.73	14.67	0.68	78.39	37.69	30.00
	K-M	0.73	14.47	0.66	76.40	35.71	30.00
	Fried.	**0.67**	14.33	0.76	**89.96**	42.35	**20.00**
	LIPS	**0.67**	**14.07**∗	**0.81**	82.55	**33.85**∗§	28.80
500	D-U	0.40	14.73	0.80	82.48	36.23	30.00
	K-M	0.60	14.87	0.75	77.33	35.55	30.00
	Fried.	1.33	15.00	0.44	**89.49**	44.09	**20.00**
	LIPS	**0.33**§	**14.13**∗†§	**0.87**§	84.78	**34.05**§	28.27
1000	D-U	0.13	12.93	0.95	84.47	36.84	30.00
	K-M	0.07	12.13	0.94	77.10	35.15	30.00
	Fried.	**0.00**	**10.67**	0.94	**88.87**	48.25	**20.00**
	LIPS	0.07	12.40	**0.98**	85.50	**33.78**§	27.73
2000	D-U	1.20	12.20	0.76	83.41	38.79	30.00
	K-M	1.20	11.60	0.79	76.85	37.20	30.00
	Fried.	**1.00**	11.33	**0.80**	**88.96**	41.50	**20.00**
	LIPS	**1.00**	**10.87**∗	**0.80**	85.28†	**36.18**∗§	28.20
3000	D-U	0.27	11.67	0.94	82.98	37.17	30.00
	K-M	0.20	11.33	0.92	76.86	35.93	30.00
	Fried.	**0.00**	10.67	**1.00**	**88.17**	43.83	**20.00**
	LIPS	**0.00**∗†	**10.60**∗	**1.00**	85.83∗†	**35.69**§	28.20
Promedio	D-U	0.55	13.24	0.83	82.34	37.08	30.00
	K-M	0.56	12.88	0.81	76.91	35.74	30.00
	Fried.	0.60	12.47	0.79	**89.08**	44.00	**20.00**
	LIPS	**0.41**	**12.41**∗	**0.89**∗†	84.79 †	**34.71**∗§	28.24

Tabla 5.8: Resultados obtenidos para la red de la figura 5.3 usando datos generados con distribución Uniforme. Se muestran experimentos con diferente número de casos de 250 a 3000. Para cada una de esas filas se muestra el promedio de variar el número de intervalos iniciales, además cada experimento se repitió 10 veces.

Uniforme							
Núm-C	Alg.	A+	A-	S-E	PRB	E-T	#Int
250	D-U	0.00	15.00	**1.00**	77.19	34.47	30.00
	K-M	1.00	14.67	0.75	75.23	34.27	30.00
	Fried.	0.67	**14.33**	0.78	**84.79**	40.82	**20.00**
	LIPS	**0.33**	**14.33**∗	0.92	78.04	**32.87**∗§	24.33
500	D-U	1.67	**14.33**	0.58	78.03	34.10	30.00
	K-M	1.00	14.67	**0.80**	75.06	33.94	30.00
	Fried.	**1.33**	**14.33**	0.64	**86.73**	41.21	**20.00**
	LIPS	**1.33**	**14.33**	0.64	79.91	**34.60**§	24.67
1000	D-U	0.47	11.00	0.87	78.52	35.05	30.00
	K-M	0.67	11.67	0.87	75.71	34.34	30.00
	Fried.	**0.00**	**10.33**	**1.00**	**86.94**	39.39	**20.00**
	LIPS	**0.00**∗†	11.67	**1.00**∗†	79.53	**33.94**§	25.67
2000	D-U	1.00	10.67	0.84	77.94	35.82	30.00
	K-M	1.33	11.00	0.78	75.14	34.78	30.00
	Fried.	1.00	**10.33**	0.86	**88.47**	39.66	**20.00**
	LIPS	**0.00**∗†§	**10.33**	**1.00**∗†§	78.39	**33.60**§	25.33
3000	D-U	**0.00**	10.33	0.93	78.32	34.87	30.00
	K-M	0.33	10.67	0.92	74.64	34.97	30.00
	Fried.	**0.00**	**8.33**	**1.00**	**88.36**	38.77	**20.00**
	LIPS	**0.00**†	**8.33**∗†	**1.00**	79.98	**34.75**§	26.00
Promedio	D-U	0.62	12.27	0.82	78.00	34.86	30.00
	K-M	0.87	12.53	0.82	75.17	34.46	30.00
	Fried.	0.60	**11.53**	0.86	**87.06**	39.97	**20.00**
	LIPS	**0.33**†	11.80	**0.91**∗†	79.17†	**33.95**§	25.20

Tabla 5.9: Resultados de la red de la figura 5.3 variando el número de casos para cada inicialización.

		Gausiana						Uniforme					
I-Ini.	Alg.	A+	A-	S-E	PRB	E-T	#Int	A+	A-	S-E	PRB	E-T	#Int
2	D-U	0.80	12.56	0.80	83.19	41.40	**20.00**	1.28	11.20	0.73	81.45	36.24	20.00
	K-M	0.68	12.48	0.76	85.01	**36.38**	**20.00**	0.80	11.60	0.86	84.01	32.81	20.00
	Fried.	1.00	**11.80**	0.73	**87.41**	39.42	**20.00**	0.80	**10.80**	0.78	**86.26**	40.65	**20.00**
	LIPS	**0.56**	12.08	**0.83**	84.67	37.12	28.40	**0.60**	12.00	**0.87**	79.51	**33.54**	25.40
3	D-U	0.32	13.32	0.84	83.40	37.73	30.00	0.33	10.33	0.76	64.89	29.14	25.00
	K-M	0.44	12.88	0.86	76.62	35.71	30.00	0.67	10.50	0.70	62.17	29.17	25.00
	Fried.	0.40	**12.40**	0.83	**90.64**	44.73	**20.00**	0.17	**9.50**	0.81	**72.05**	33.08	**16.67**
	LIPS	**0.24**	12.60	**0.95**	84.91	**33.68**	28.20	**0.00**	9.83	**0.83**	65.93	**27.76**	20.83
4	D-U	0.52	13.84	0.85	78.76	33.95	40.00	0.60	13.20	0.83	73.41	**34.11**	40.00
	K-M	0.56	13.28	0.82	69.08	35.63	40.00	1.00	13.40	0.77	66.85	35.84	40.00
	Fried.	0.40	13.00	0.81	**91.98**	47.86	**20.00**	0.80	12.40	0.82	**88.05**	40.96	**20.00**
	LIPS	**0.44**	**12.56**	**0.90**	84.79	**33.33**	28.12	**0.40**	**11.60**	**0.87**	79.48	34.61	25.20

en ambas distribuciones.

En la tabla 5.9 se muestran los resultados por número de intervalos iniciales. En este experimento se presenta el promedio de variar el número de casos de 250 a 3000.

Para estos resultados, aún cuando la red es mucho más compleja el algoritmo propuesto obtuvo nuevamente los mejores resultados en calidad estructural y error temporal. Más aún, la calidad predictiva se mantiene sin importar la inicialización.

5.5.1. Análisis de complejidad temporal

En la tabla 5.10 se presenta el promedio y desviaciones estándar de los tiempos de ejecución (en segundos) para las 3 redes de las figuras 5.1-5.3. De los resultados se puede notar que para la primer red, la cual es la más pequeña de todas y para cuando el número de casos es pequeño, el algoritmo de Friedman es más rápido que nuestro algoritmo. Sin embargo, cuando el número de casos aumenta, nuestro algoritmo obtiene significativamente un menor tiempo de ejecución. Para la segunda red, ocurre algo similar, cuando el número de casos es pequeño, Friedman obtiene bajo tiempo de ejecución, pero al aumentar, nuestro

Tabla 5.10: Tabla que muestra los promedios y desviaciones estándar de los tiempos de ejecución del algoritmo de Friedman y el propuesto (LIPS) en las 3 redes usadas en los experimentos. En la columna Núm-C se muestra el número de casos en la red-1 y red-2, entre paréntesis se muestra el número de casos para la red-3. Para cada red se muestra el promedio de 10 ejecuciones.

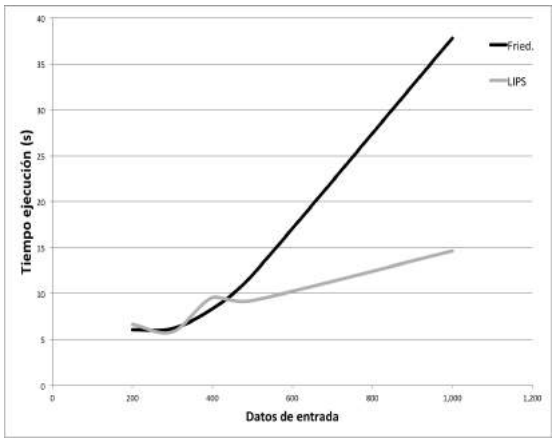
Núm-C	Algoritmo	Red-1	Red-2	Red-3
200 (250)	Fried.	**6.04±3.55**	**25.13±5.10**	44.24±11.28
	LIPS	6.65±2.09	34.18±4.09	**31.99±2.55**
300 (500)	Fried.	6.18±2.80	**30.85±3.96**	134.21±57.70
	LIPS	**5.82±1.07**	28.11±4.02	**37.03±3.86**
400 (1000)	Fried.	**8.34±2.72**	46.79±7.93	502.30±53.03
	LIPS	9.57±0.80	**34.84±2.58**	**65.27±4.96**
500 (2000)	Fried.	12.01±2.33	69.16±6.47	2,275.99±19.17
	LIPS	**9.24±0.48**	**53.64±2.41**	**95.19±3.81**
1000 (3000)	Fried.	37.81±1.68	256.69±9.15	6,342.38±220.22
	LIPS	**14.63±1.67**	**90.60±2.65**	**591.19±29.43**

algoritmo obtiene un tiempo hasta 3 veces menor en promedio. Para la red más grande, para todos los casos, nuestro algoritmo obtuvo un menor tiempo de ejecución. Cuando el número de datos de entrada fue el mayor, nuestro algoritmo fue aproximadamente 20 veces más rápido en promedio.

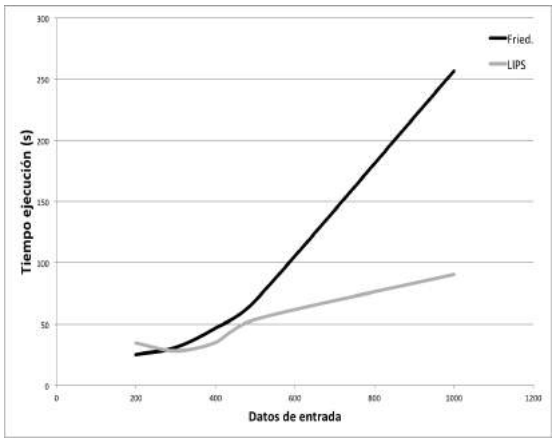
En la figura 5.4 se muestra cómo aumentan los tiempos de ejecución con respecto al número de datos. Se puede observar que nuestro algoritmo crece mucho más lentamente que el algoritmo (Friedman y Goldszmidt 1996). Algunas razones por las que el algoritmo de Friedman obtiene tiempos considerablemente mayores son:

- El algoritmo tiene que evaluar todos los puntos medios existentes en el conjunto de datos para el nodo a discretizar, esto puede ser un número elevado dependiendo de la dispersión de los datos.

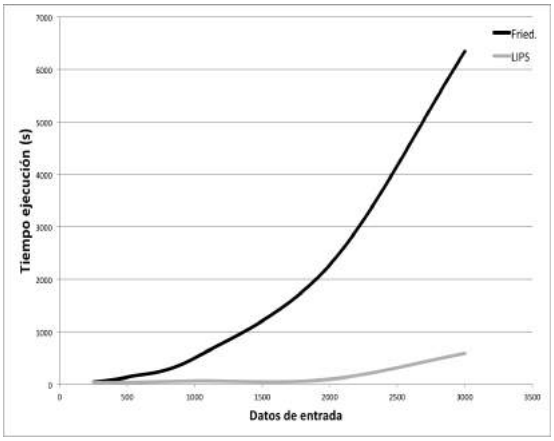
- El algoritmo contiene un ciclo mientras, el cual no termina hasta que la cola con la lista de nodos este vacía. De la lista se obtiene un nodo y se encuentra una discretización, después se agregan los nodos con los que tiene relación el nodo que se está optimizando. Si es una red conexa, es decir tiene un gran número de arcos, entonces es probable que



(a) Tiempos de ejecución para la red pequeña



(b) Tiempos de ejecución para la red mediana



(c) Tiempos de ejecución para la red grande

Figura 5.4: Tiempos de ejecución para las redes de las figuras 5.1- 5.3, en gris obscuro se muestra el tiempo obtenido por (Friedman y Goldszmidt 1996), mientras que en color gris claro se muestra el tiempo de nuestro algoritmo

.

Tabla 5.11: Promedios y desviaciones estándar de los resultados de los 3 conjuntos de experimentos con datos generados de forma gausiana.

Experimento	Alg.	A+	A-	S-E	PRB	E-T	#Int
Red-1	D-U	0.15±0.3	1.75±0.86	0.95±0.09	77.05±2.33	17.95±1.96	6.00±1.69
	K-M	0.03±0.07	1.68±0.87	0.97±0.05	74.58±3.26	18.17±1.40	6.00±1.69
	Fried.	0.33±0.49	1.68±0.79	0.94±0.13	**80.07±1.10**	19.57±1.14	**3.73±0.46**
	LIPS	**0.00**§	**1.39±1.03**∗	**1.00**∗§	77.98±0.68†	**16.68±1.65**∗†§	5.81±0.14
Red-2	D-U	0.57±0.71	1.33±0.9	0.93±0.09	77.55±2.09	7.09±0.43	15.00±4.23
	K-M	0.48±0.74	1.12±0.9	0.94±0.08	75.72±4.29	6.73±0.25	15.00±4.23
	Fried.	0.33±0.72	**0.83±0.75**	0.96±0.08	**80.43±1.97**	9.30±1.89	**10.67±0.82**
	LIPS	**0.24±0.51**∗	0.85±0.72∗	**0.97±0.06**∗	79.12±0.70†	**6.69±0.42**∗†	13.67±1.88
Red-3	D-U	0.55±0.51	13.24±1.45	0.83±0.12	82.34±4.6	37.69±3.4	30.00±8.45
	K-M	0.56±0.52	12.88±1.60	0.81±0.13	76.91±6.76	35.91±0.96	30.00±8.45
	Fried.	0.60±0.74	12.47±2.06	0.79±0.23	**89.08±2.27**	44.00±4.99	**20.00**
	LIPS	**0.41±0.47**	**12.41±1.60**∗	**0.89±0.13**∗†	84.79±1.25†	**34.71±2.18**∗§	28.24±0.45

el número de nodos que se agrega a la cola sea alto y tenga un impacto en el tiempo de procesamiento.

- Aún cuando el algoritmo está garantizado a converger a un máximo local, no se sabe con que rapidez lo hará.

5.5.2. Análisis de resultados generales

En la tabla 5.11 se presenta un resumen de los experimentos realizados con datos generados con distribución gausiana. Se muestra una fila por cada conjunto de experimentos (3 experimentos, uno por cada red). Para cada red se muestran los promedios de los subexperimentos para los 4 algoritmos.

De los resultados podemos hacer las siguientes conclusiones:

- El algoritmo propuesto obtuvo los mejores resultados en calidad estructural y error temporal. Para el error temporal fue significativamente mejor que al menos otros dos algoritmos para las tres redes. Para similitud estructural fue significativamente mejor que al menos un algoritmo en las tres redes.

- El algoritmo de Friedman obtuvo el mejor resultado en puntaje predictivo y número

Tabla 5.12: Promedios y desviaciones estándar de los resultados de los 3 conjuntos de experimentos con datos generados uniformemente.

Experimento	Alg.	A+	A-	S-E	PRB	E-T	#Int
Red-1	D-U	0.16±0.19	1.71±0.79	0.96±0.04	74.29±1.67	18.67±1.6	6.00±1.69
	K-M	0.24±0.35	1.56±0.73	0.94±0.09	74.35±2.33	18.37±1.17	6.00±1.69
	Fried.	0.24±0.42	1.46±0.74	0.94±0.1	**77.77±2.73**	18.10±0.76	**4.40±0.51**
	LIPS	**0.01±0.05**∗†§	**1.40±0.74**∗	**0.99±0.02**∗†§	76.15±1.7∗†	**17.80±0.61**∗†	4.47±0.64
Red-2	D-U	0.63±0.80	0.99±0.92	0.92±0.09	77.55±2.98	6.69±0.54	15.00±4.23
	K-M	0.71±0.64	0.84±0.76	0.92±0.07	75.21±4.05	6.47±0.36	15.00±4.23
	Fried.	0.24±0.44	**0.40±0.51**	**0.97±0.05**	80.33±0.74	7.42±0.83	**10.00**
	LIPS	**0.23±0.41**∗†	0.60±0.51	**0.97±0.05**∗†	**80.38±0.91**∗†	**6.24±0.13**∗†§	13.47±1.13
Red-3	D-U	0.62±0.84	12.27±2.31	0.88±0.21	78.00±3.54	34.86±1.67	30.00±8.45
	K-M	0.87±0.92	12.53±2.13	0.82±0.19	75.16±7.29	34.46±1.76	30.00±8.45
	Fried.	0.60±0.91	**11.53±2.75**	0.86±0.25	**87.06±2.02**	39.97±1.09	**20.00**
	LIPS	**0.33±0.72**†	11.80±2.46	**0.91±0.2**∗†	79.17±1.0†	**33.95±1.36**§	25.20±0.86

de intervalos.

- El algoritmo *K-means* tiene alta desviación estándar en su puntaje predictivo en todos los experimentos.
- El algoritmo de discretización uniforme obtuvo mejores resultados que el algoritmo de *K-means* aun cuando es el más simple de los 4 algoritmos.
- Los algoritmos más elaborados: el propuesto y el de Friedman, superaron en promedio a los algoritmos simples: discretización y *K-means*.
- El algoritmo propuesto obtuvo buenos resultados debido a que usa la suposición de que los datos son gausianos.

En la tabla 5.12 se presentan un resumen de los experimentos realizados con datos generados con distribución uniforme.

De los resultados podemos hacer las siguientes conclusiones:

- El algoritmo propuesto obtuvo los mejores resultados en calidad estructural y error temporal. Para el error temporal fue significativamente mejor que al menos un algoritmo para las tres redes. Para similitud estructural fue significativamente mejor que al menos dos algoritmos en las tres redes.

- El algoritmo *K-means* tiene alta desviación estándar en su puntaje predictivo en todos los experimentos.

- El algoritmo de discretización uniforme obtuvo mejores resultados que el algoritmo de *K-means* aun cuando es el más simple de los 4 algoritmos.

- Los algoritmos más elaborados: el propuesto y el de Friedman, superaron en promedio a los algoritmos simples: discretización y *K-means*.

- Aún cuando el algoritmo propuesto hace la suposición de que los datos son gausianos y en este caso no se cumple, se obtuvieron buenos resultados, comparables contra el algoritmo de Friedman y superando a los algoritmos base.

- Algo interesante a notar es que el algoritmo de Friedman en las tres redes generalmente obtuvo el menor número de intervalos por nodo temporal (casi siempre son 2 intervalos). Debido a que el algoritmo usa el mínimo y máximo de los valores temporales, el algoritmo de Friedman obtiene dos intervalos de la forma: $[mínimo - partición]$ $[partición - máximo]$. Dependiendo de los datos, esto puede resultar en que los intervalos sean muy grandes, lo cual se puede notar en el alto error temporal obtenido por el algoritmo. Sin embargo, creemos que esto también tiene impacto en la calidad predictiva. El bajo número de intervalos y su gran tamaño hace menos probable que los datos estén fuera de ellos con lo que obtienen un alto puntaje predictivo. Por lo expresado anteriormente sería interesante considerar una medida combinada que tome en cuenta el número de intervalos y la calidad predictiva.

5.6. Resumen

En este capítulo se presentaron los experimentos realizados con datos sintéticos. Se realizaron tres conjuntos de experimentos, cada uno de ellos representa una red de nodos temporales de diferente tamaño. Para cada conjunto de experimentos se mostró como se comportan los algoritmos cuando se varía la distribución de los datos, el número de casos y

la inicialización. Además se mostraron los resultados en general de todos los experimentos. En promedio, el algoritmo propuesto obtiene la mejor calidad estructural y el menor error temporal. Se realizaron pruebas estadísticas de las que se concluye que el algoritmo obtiene mejores resultados significativamente en las 3 redes usadas en los experimento en términos de similitud estructural y error temporal. Además, nuestro algoritmo obtiene buenos puntajes en arcos añadidos y eliminados. En calidad predictiva y número de intervalos el algoritmo propuesto supera a los algoritmos base (Discretización uniforme y *K-means*). Con estos experimentos se muestra que el algoritmo propuesto es robusto y tiene resultados aceptables en distintos casos. Además se realizaron experimentos para evaluar la complejidad de nuestro algoritmo y compararla contra el algoritmo de (Friedman y Goldszmidt 1996), en los resultados obtenidos, del promedio de varias ejecuciones donde se fue aumentando el número de casos, el tiempo de nuestro algoritmo fue significativamente menor, aproximadamente polinomial, en tanto que el algoritmo de Friedman parece tener un comportamiento exponencial en el tiempo. Una de las limitaciones del algoritmo es que no obtiene la mejor calidad predictiva, ya que es superado por el algoritmo de Friedman, esto puede suceder debido a que este algoritmo obtiene menos intervalos de gran tamaño. Nuestro algoritmo también se usó en dos aplicaciones con datos reales, una aplicación industrial y una médica, las cuales se presentan en el siguiente capítulo.

Capítulo 6

Aplicaciones con datos reales

Además de los experimentos con datos sintéticos, el algoritmo propuesto en esta tesis se aplicó a dos dominios con datos reales: en la primera aplicación se usaron datos de un simulador de una planta eléctrica para predicción y diagnóstico de fallas. En la segunda, se obtuvieron modelos para predicción de mutaciones de la enzima proteasa, la cual es un componente importante del VIH (Virus de la Inmunodeficiencia Humana).

6.1. Planta eléctrica

En esta aplicación se usó el algoritmo propuesto en esta tesis para obtener un modelo para predicción y diagnóstico de fallas en un subsistema de una planta eléctrica de ciclo combinado.

6.1.1. Introducción

Una planta eléctrica consiste básicamente de 3 partes: un generador de calor, una turbina de vapor y un generador eléctrico. El generador de calor contiene varios tubos y un gran tanque llamado domo. Un diagrama simplificado de la planta se muestra en la figura 6.1.

En el domo existe una combinación de vapor y agua, la cual debe de estar en niveles estables para que se pueda llevar a cabo el proceso de generación eléctrica. Un descenso

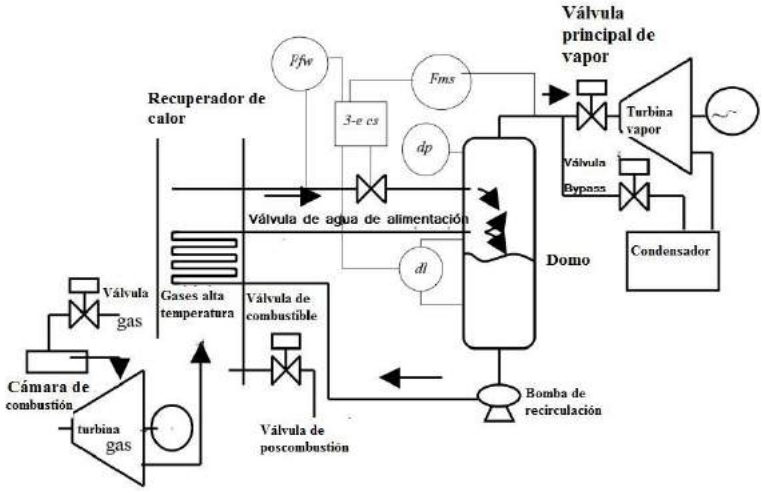


Figura 6.1: Descripción esquemática de la planta eléctrica mostrando algunos componentes importantes tales como la válvula de agua y la válvula de vapor. Además, se muestra la presión del domo, *dp* y el nivel del mismo, *dl*.

en el nivel del domo puede causar un exceso de calor y destruir otros componentes. Por el contrario, un nivel alto del domo puede acarrear humedad en la turbina y causarle daño. En ambos casos, si los valores del domo no están controlados, el ciclo de generación eléctrica disminuye su capacidad, lo cual representa pérdida de dinero, ya que los componentes no están operando a su capacidad óptima.

Para controlar la planta existen operarios que deben estar atentos a los niveles de control y tomar las acciones necesarias en aquellos casos en que la planta no funcione adecuadamente. Sin embargo, sería de mucha utilidad un modelo que pueda ayudar a diagnosticar una falla y que muestre cómo esa falla afecta a otros componentes, más aún sí se obtuviera información temporal de cuanto tiempo se tiene para resolver esa falla antes de ciertos eventos. Debido a lo anterior, se decidió usar el modelo de RBNT para diagnosticar ciertas fallas y ver sus efectos temporales en la planta.

6.1.2. Variables y datos

Un planta eléctrica contiene muchos elementos importantes los cuales están expuestos a fallas. Para simplificar el problema, en nuestros experimentos tomamos en cuenta una parte

de la planta y algunos componentes importantes, en particular:

- Válvula principal de agua. Componente del que se simulará una falla.
- Válvula principal de vapor. Componente del que se simulará una falla.
- Flujo de agua. Flujo de entrada de agua al domo.
- Flujo de vapor. Flujo de vapor que va a la turbina.
- Presión del domo. Valor que debe de controlarse para no afectar el funcionamiento de la planta.
- Nivel del domo. Otro valor importante del componente domo que debe ser controlado para el buen funcionamiento de la planta.
- Generación eléctrica. Capacidad de generación de electricidad que debe mantenerse en condiciones óptimas.

Cada uno de los componentes mencionados representará un nodo en el modelo de la RBNT construida. En este dominio, consideramos que un evento ocurre cuando una señal o componente excede su límite normal de funcionamiento.

Para obtener los datos usados en los experimentos se usó un simulador de la planta eléctrica desarrollado por el Instituto de Investigaciones Eléctricas, además se usó la interfaz proporcionada por el programa ASISTO (Reyes-Ballesteros 2006), presentado en la figura 6.2, el cual se conecta al simulador. Para los experimentos se alternaron aleatoriamente 3 diferentes escenarios: falla en la válvula de agua, falla en la válvula de vapor o funcionamiento correcto (sin fallos). Para cada uno de los diferentes escenarios se obtuvieron diversos casos que se usaron en los experimentos.

6.1.3. Evaluación y Resultados

El algoritmo propuesto se comparó contra dos algoritmos base: discretización uniforme y el algoritmo 4.4 (basado en *K-means*) para obtener los intervalos iniciales de cada nodo

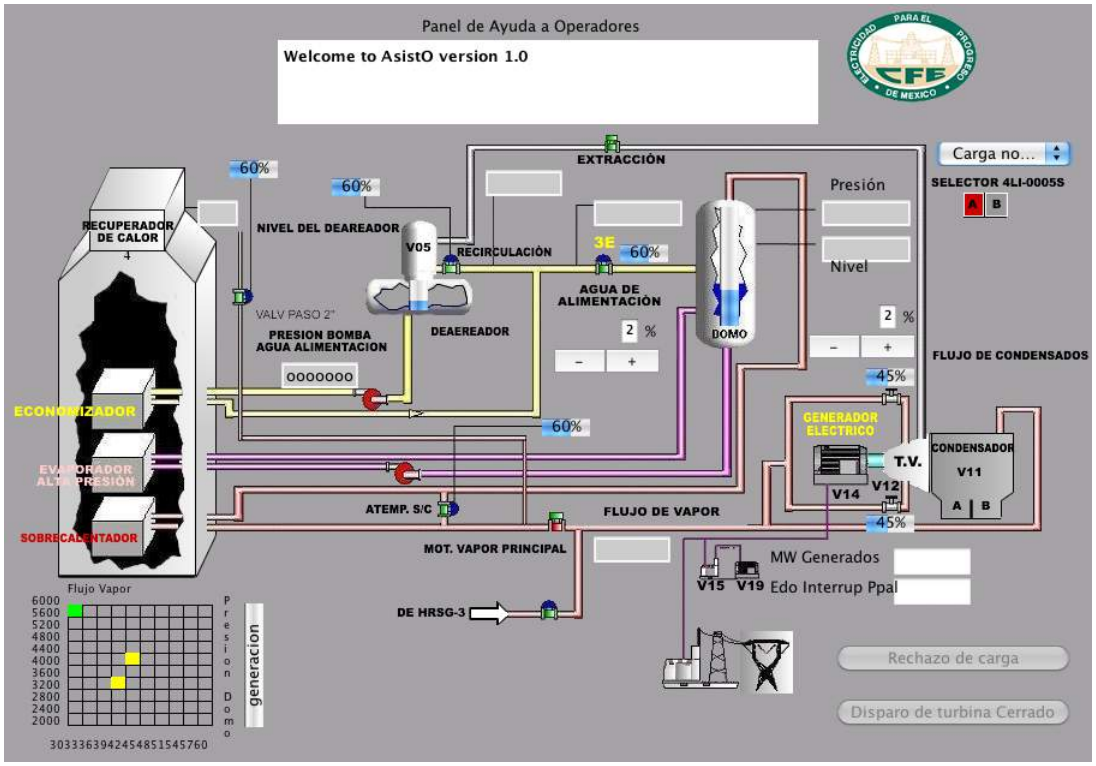


Figura 6.2: Captura de pantalla de la interfaz del simulador ASISTO (Reyes-Ballesteros 2006). En él se pueden apreciar algunos componentes importantes como el recuperador de calor, el domo y el generador eléctrico.

Tabla 6.1: Evaluación para el dominio de la planta eléctrica. Se compara el algoritmo propuesto (Prop), el algoritmo K-means y la discretización uniforme (D-U) en términos de calidad predictiva (PRB), el error temporal y el número de intervalos generados.

Núm. de casos	Algoritmo	PRB (Max 100)	Error Temporal	Núm. intervalos promedio
50	LIPS	**93.26**∗†	18.02	**16.25**∗†
50	K-means	83.57	**15.6**	24.5
50	D-U	85.3	16.5	24.5
75	LIPS	**93.7**∗†	17.8	**16**∗†
75	K-means	85.7	**16.3**	24.5
75	D-U	86.9	17.2	24.5
100	LIPS	**93.37**†	17.7	**17**∗†
100	K-Means	90.4	17.1	24.5
100	D-U	91.9	**15.29**	24.5

temporal. Se usaron tres medidas de evaluación: (i) la calidad predictiva usando el puntaje relativo de Brier (PRB), (ii) el error temporal y (iii) el número de intervalos en la red. El mejor modelo debe obtener una alta calidad predictiva, un bajo error temporal y una baja complejidad, dada por un bajo número de intervalos.

Se realizaron tres experimentos variando el número de casos usados para aprender la red. Para todos los experimentos se generaron datos con el simulador de la planta y con esos datos se usó el algoritmo de aprendizaje para obtener la estructura y los intervalos. Los modelos fueron evaluados usando las medidas mencionadas en la sección 5.1. Se realizaron pruebas de significancia estadística de Kruskal-Wallis con $\alpha = 0{,}05$ (ver Sección 5.1.4), si el algoritmo propuesto es mejor que el algoritmo de discretización uniforme se muestra un $*$, por otro lado si nuestro algoritmo es mejor que el algoritmo *K-means* se muestra un † en la tabla. Un resumen de los experimentos y resultados se muestra en la tabla 6.1. Además, la red que obtuvo el mejor puntaje predictivo se muestra en la figura 6.3.

Las siguientes observaciones se pueden obtener de los resultados. En todos los experimentos el algoritmo propuesto obtuvo el mejor puntaje predictivo y el menor número de intervalos. Los algoritmos *K-means* y discretización uniforme obtuvieron el menor error

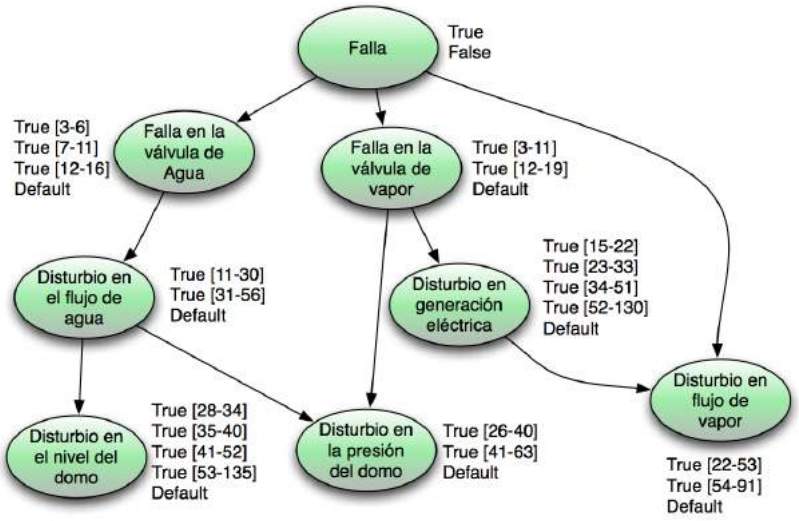


Figura 6.3: RBNT aprendida con el algoritmo propuesto para el dominio de la planta eléctrica.

temporal. Sin embargo, esto ocurre debido que generaron un alto número de intervalos, con lo cual se disminuye la diferencia entre la media del intervalo y el evento real. Aún cuando el algoritmo propuesto no obtiene el menor error temporal no está lejos de los otros algoritmos. Además, es importante notar que nuestro algoritmo obtuvo la mejor calidad predictiva con un modelo más simple.

Para la evaluación cualitativa de la red, los expertos[1] revisaron los modelos y confirmaron que la posible sucesión de eventos que presenta la RBNT son acordes a los principios de cómo opera la planta. Debido a la gran incertidumbre y alto número de componentes en el proceso de generación eléctrica, una evaluación más a fondo no pudo ser realizada.

6.2. Mutaciones de Proteasa en VIH

La segunda aplicación del algoritmo propuesto en esta tesis fue para modelar la ocurrencia de las mutaciones en una enzima del VIH. A continuación se presenta una breve descripción del virus y cómo esta formado, además se mencionan los fármacos utilizados

[1]El Dr. Pablo Ibarguengoytia, investigador del Instituto de Investigaciones Eléctricas y algunos de sus colaboradores evaluaron los modelos obtenidos.

para atacar el VIH y los problemas que estos fármacos tienen. Posteriormente se habla del trabajo relacionado y finalmente se muestran los experimentos y resultados obtenidos.

6.2.1. Introducción

La evolución viral es un aspecto importante de la epidemiología de ciertas enfermedades virales tales como la influenza, la hepatitis y el virus de inmunodeficiencia adquirida (VIH). Esta evolución impacta en el desarrollo de vacunas y antiretrovirales eficientes ya que las mutaciones proveen resistencia a los fármacos. En el VIH esto es muy relevante ya que es uno de los organismos con más rápida evolución (Freeman, Herron y Payton 1998). La enorme velocidad de replicación está asociada con la alta tasa de mutación y con una alta probabilidad de recombinación en el genoma del virus durante el periodo de replicación. Estas características permiten al VIH presumir de una alta variabilidad genética, aún considerando sólo la población dentro de un sólo individuo (huésped). Esta elevada capacidad de variación da al virus una habilidad notable para adaptarse a múltiples presiones selectivas, incluyendo la respuesta inmune y la terapia de antiretrovirales. Debido a lo mencionado anteriormente surgen varias preguntas, por ejemplo: ¿Cuánto de esta diversidad se debe a la respuesta inmune de cada individuo y cuánto a la terapia antiretroviral? ¿Cuál es la relación entre diversidad genética y la respuesta clínica? ¿Es posible que entendiendo mejor la evolución del VIH se pueda reducir la resistencia a los fármacos?

Motivados por la última pregunta, se puede decir que sería deseable construir terapias *proactivas* que consideren las mutaciones futuras, lo que reduciría el riesgo de resistencia a los fármacos, en lugar de esperar a que el virus desarrolle resistencia y entonces *reactivamente* se realice el cambio de la terapia. Por lo tanto, si fuéramos capaces de predecir la evolución más probable del virus en cualquier huésped, entonces se podrían desarrollar terapias antiretrovirales que inhiban la aparición de ciertas mutaciones, incrementando efectivamente el control del VIH.

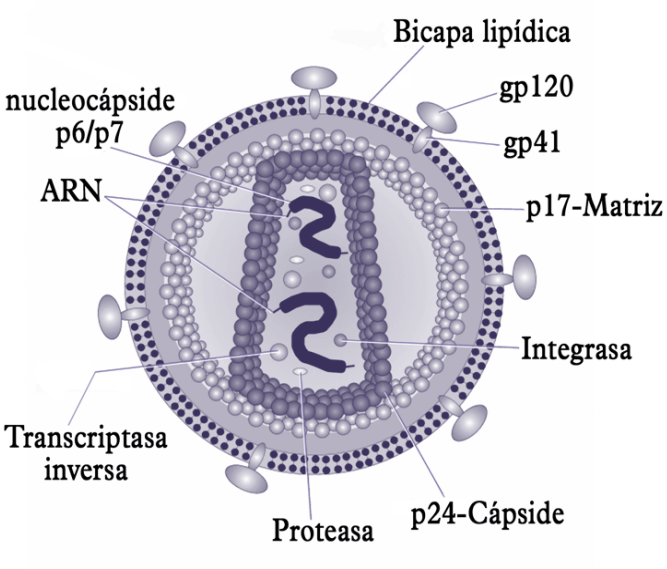


Figura 6.4: Estructura del VIH. Se presentan sus componentes principales entre los que destacan las enzimas: transcriptasa inversa, integrasa y proteasa.

6.2.2. VIH

El virus de la inmunodeficiencia humana (VIH) fue descubierto y considerado como el agente de la naciente epidemia de SIDA en 1983 (Córdoba Villalobos, Leon Rosales y Valdespino 2009). EL VIH es un virus que corresponde a la familia de los retrovirus, los cuales se caracterizan por la transcripción inversa (ARN $\rightarrow$ ADN) (Weiss 1993). El virus tiene un diámetro de aproximadamente 100 nanómetros y tiene una estructura esférica. Su parte exterior es la "cubierta", una membrana que originalmente pertenecía a la célula de donde el virus emergió. El núcleo tiene una "cápside", compuesta por la proteína p24. En su interior está el ARN, la forma de información genética del VIH. El ARN debe copiarse provisionalmente al ADN para poder multiplicarse e integrarse en el genoma de la célula que infecta. Dentro de la cápside, además de las dos copias idénticas del ARN viral, hay ejemplares de tres enzimas necesarias para la multiplicación del virus: una transcriptasa inversa, una integrasa y una proteasa. En la figura 6.4 se presenta una representación simple de la composición del VIH.

Como ya se mencionó, el virus cuenta con un conjunto de enzimas que son usadas para

su replicación, a continuación se presentan las funciones de esas enzimas:

- La proteasa. Actúa cortando las piezas de las proteínas. Una parte de los fármacos empleados contra el VIH son inhibidores de su función.
- La transcriptasa inversa. Su función es la síntesis del ADN de doble cadena usando como patrón la cadena singular del ARN viral. También existen múltiples fármacos contra la actividad de la transcriptasa inversa.
- La integrasa realiza la inserción del ADN proviral en el genoma de la célula huésped. En la actualidad existe un fármaco comercial contra la actividad de la integrasa.

Fármacos antriretrovirales

Los fármacos antirretrovirales son medicamentos para el tratamiento de la infección por el VIH, causante del SIDA. Diferentes antirretrovirales actúan en varias etapas del ciclo vital del VIH.

Es importante mencionar que las guías de tratamiento están en cambio constante, desde un inicio más agresivo a un enfoque más conservador. En sí, no existe una pauta exacta para saber cuando iniciar el tratamiento, salvo el uso de la clínica y el estado inmunológico del paciente. Además, los regímenes antirretrovirales son complejos, con posibles efectos colaterales serios con potencial desarrollo de resistencia viral.

Debido a las complicaciones mencionadas, actualmente se usan combinaciones de fármacos, ya que actúan incrementando el número de obstáculos para la mutación viral, manteniendo bajo el número de copias virales. Sin embargo, no todas las combinaciones de antirretrovirales pueden ser efectivas, lo que limita el número de combinaciones disponibles.

Resistencia del VIH a los fármacos

La resistencia del VIH a los fármacos ocurre cuando existe una mutación en el virus tal que lo hace tolerante a los tratamientos antiretrovirales. Cuando un medicamento deja de hacer efecto, el sistema inmune del paciente vuelve a ser comprometido. El VIH contiene

intrínsecamente ciertas características que lo ayudan a facilitar la resistencia, la más importante es su elevada tasa de mutación. De hecho, el VIH es el organismo con la tasa más alta de mutación (Freeman, Herron y Payton 1998). A final de cuentas esta elevada tasa de mutación puede dar lugar a una población completamente nueva de virus que es totalmente resistente al fármaco.

Por lo anterior, es muy importante conocer las mutaciones que ocurren en el VIH. Analizando las mutaciones y los fármacos se puede tener una idea de cuales mutaciones hacen que el fármaco deje de funcionar. Por consiguiente, se pueden modificar los tratamientos para que tengan mayor efectividad.

6.2.3. Trabajo relacionado

Existen algunos trabajos reportados en los que se han usado métodos computacionales con datos de VIH para predecir la resistencia de los fármacos usando árboles de decisión (Beerenwinkel et al. 2002) y redes neuronales (Draghici y Potter 2003). Otros trabajos buscan identificar asociaciones importantes en las variables clínicas y el VIH (Ramirez et al. 2000). Sin embargo, existen pocos trabajos que traten de identificar patrones temporales en los datos. Uno de los pocos trabajos que considera información temporal para obtener reglas de asociación es (Chausa et al. 2009). En este trabajo se buscan relaciones temporales entre variables clínicas y un posible fallo del tratamiento antiretroviral. Sin embargo, a la fecha no existen trabajos que traten de modelar como ocurren las mutaciones respecto al tiempo y los fármacos usados. En particular, para estos experimentos nos enfocamos en buscar un modelo temporal de las mutaciones de la proteasa tomando como variable inicial alguno de los medicamentos (inhibidores de proteasa) usados.

6.2.4. Datos usados y preprocesamiento

Los datos fueron obtenidos de la base de datos de VIH de Stanford (HIVDB) (Rhee et al. 2003). El VIH contiene una gran cantidad de tipos y subtipos, por ello decidimos trabajar con el subtipo B, el cual es el más común en América (Hemelaara et al. 2006). En

Tabla 6.2: Un ejemplo de los datos de pacientes con VIH. El paciente P_1 tiene tres estudios temporales y el paciente P_2 sólo tiene dos estudios temporales. Las mutaciones tienen la forma Aminoacido Inicial-Posición-Aminoacido Mutado, donde cada aminoácido se representa por una letra del alfabeto.

Paciente	**Conjunto inicial de medicamentos**	**Lista de mutaciones**	**Semanas**
P_1	LPV, FPV, RTV	L63P, L10I	15
		V77I	30
		I62V	10
P_2	NFV, RTV, SQV	L10I	25
		V77I	45

total se obtuvieron 2373 pacientes con subtipo B.

Para cada paciente, se tiene una historia clínica, la cual consiste en un número de estudios. La información de cada estudio consiste en un tratamiento (conjunto de fármacos antiretrovirales) administrado al paciente, las semanas que ese tratamiento fue usado, y la lista de mutaciones obtenidas cuando el tratamiento fue suspendido (cambiado por un nuevo tratamiento). Un ejemplo de los datos se presenta en la tabla 6.2. Para los experimentos, el conjunto inicial de medicamentos corresponderán a las variables estáticas y cada mutación corresponderá a las variables temporales, donde el valor de ellas será la semana de ocurrencia de esa mutación en el paciente.

Los antiretrovirales usualmente se clasifican de acuerdo a la enzima en la que actúan. Para los experimentos nos enfocamos en la proteasa, la cual es la más pequeña de las enzimas importantes. De los datos se obtuvieron 9 medicamentos Inhibidores de Protesasa (IP), los cuales se presentan en la tabla 6.3. Estos nueve medicamentos se consideraron en los experimentos realizados. En la figura 6.5 se presenta el histograma de la administración de diferentes IP en el conjunto de datos completo.

Es importante mencionar que los datos de la HIVDB corresponden a diferentes estudios y en ciertos casos están incompletos. Por lo anterior en la figura 6.5 es posible notar que una pequeña porción de los medicamentos usados se reporta *PI* (Inhibidor de Proteasa), sin

Tabla 6.3: Medicamentos inhibidores de proteasa

Nombre del medicamento	**Nombre corto**
Atazanavir	ATV
Darunavir	DRV
Amprenavir	APV
Lopinavir	LPV
Indinavir	IDV
Nelfinavir	NFV
Ritonavir	RTV
Tripanavir	TPV
Saquinavir	SQV

especificar el medicamento. De la misma manera existe otra porción en la cual se reporta como *Desconocido* el medicamento usado.

El número de estudios por paciente varía de 1 a 10. Debido a que estamos interesados en la evolución temporal de las redes mutacionales, eliminamos aquellos pacientes que tuvieran sólo un estudio, con lo cual obtuvimos 973 en el conjunto final.

Aún cuando ya definimos los medicamentos a usar, es necesario definir el conjunto de mutaciones de interés. En la figura 6.6 se presenta el histograma de mutaciones que ocurrieron en el conjunto de datos. En total se obtuvieron 733 diferentes mutaciones que ocurrieron al menos una vez. Sin embargo, es evidente del histograma, que la mayoría de las mutaciones se presentan con poca frecuencia; por lo que pocas mutaciones exhiben alta frecuencia. Estas mutaciones son las que vamos a usar en los experimentos.

6.2.5. Evaluación y Resultados

Debido a que el algoritmo propuesto en esta tesis llega a un máximo local dependiendo de la inicialización (número de intervalos inciales), para los experimentos se varió el número de intervalos iniciales de 2 a 4 usando discretización uniforme (Liu et al. 2002). Para evaluar el modelo se usaron tres medidas: el puntaje relativo de Brier (PRB), el error temporal

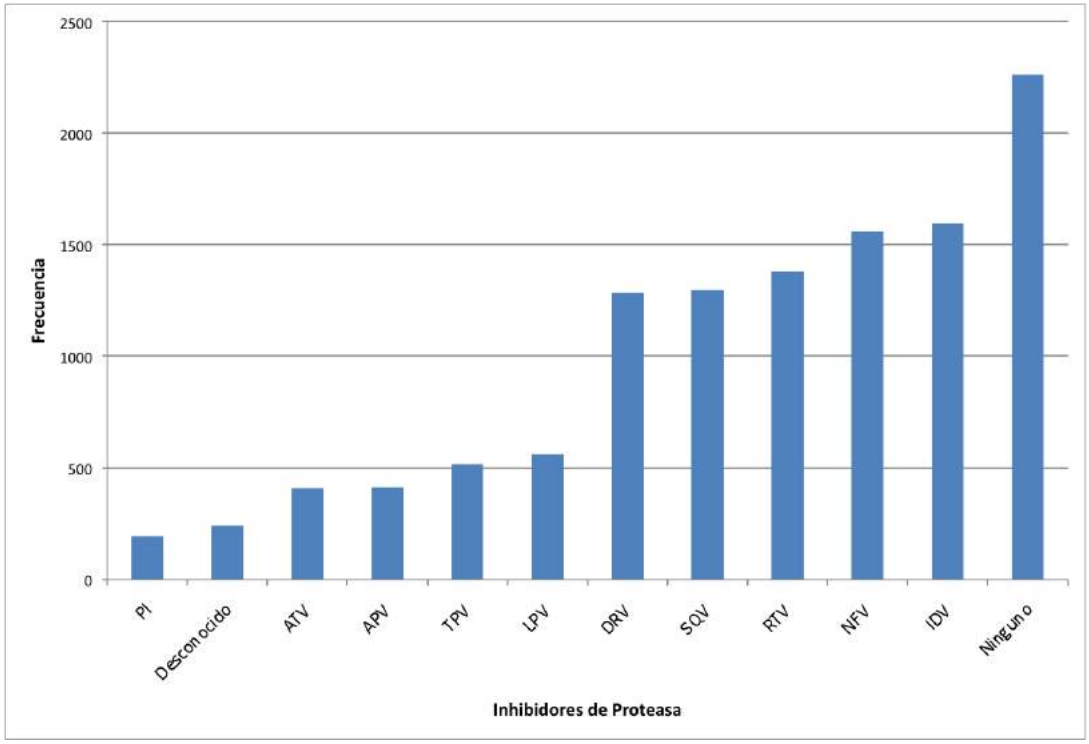


Figura 6.5: Histograma de la administración de Inhibidores de proteasa en el conjunto completo de 2373 pacientes.

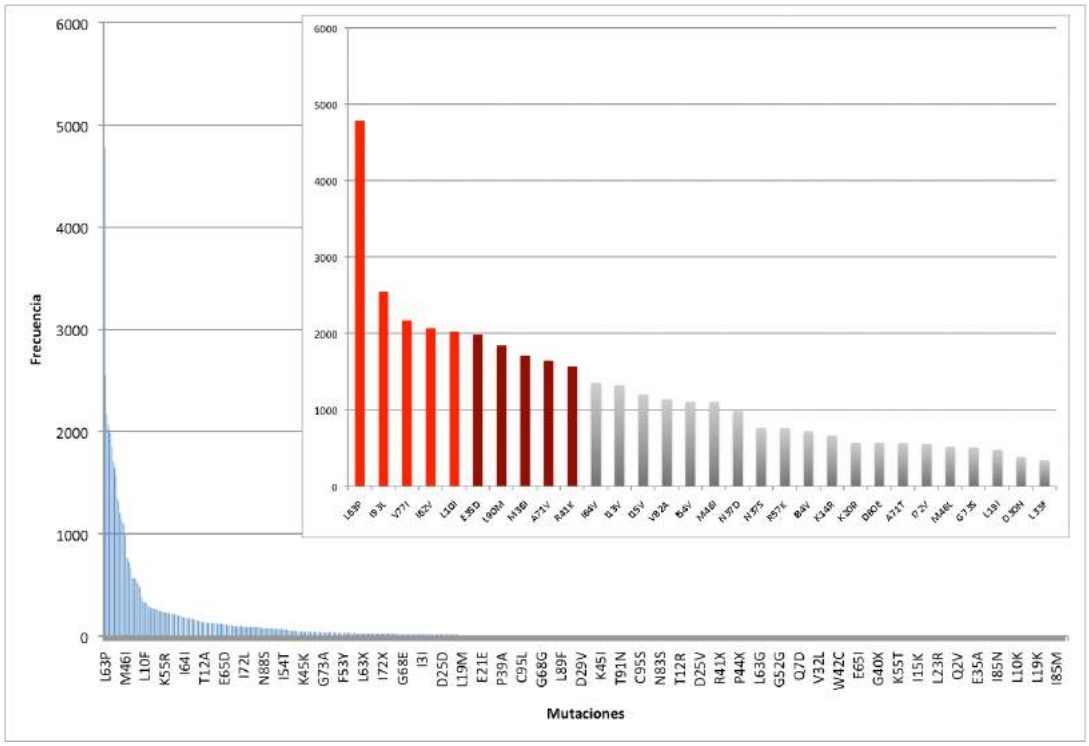


Figura 6.6: Un grupo de mutaciones y su frecuencia usando el conjunto completo de 2373 pacientes. Las mutaciones con alta frecuencia se muestran en un acercamiento. En particular las columnas en rojo representan las mutaciones seleccionadas en el primer experimento, esas y las columnas en rojo obscuro presentan las mutaciones usadas en el segundo experimento.

relativo y el número total de intervalos. El mejor modelo debería obtener un alto puntaje predictivo, un bajo error temporal y una baja complejidad (bajo número de intervalos).

Se realizaron dos experimentos. El primer experimento realizado con un modelo pequeño, tiene como objetivo evaluar la capacidad de las RBNT de capturar relaciones previamente conocidas y de esta forma dar una validación cualitativa al modelo. El segundo experimento, con un conjunto de mutaciones ampliado, tiene como objetivo descubrir las redes mutacionales más comunes además de capturar el aspecto temporal.

En el primer experimento, sólo se usaron las mutaciones con más de 2000 ocurrencias: L63P, I93L, V77I, I62V y L10I. Para el segundo experimento se usaron aquellas mutaciones con una ocurrencia mayor a 1500 veces: L63P, I93L, V77I, I62V, L10I, E35D, L90M, M36I, A71V y R41K.

Tabla 6.4: Evaluación de los modelos obtenidos variando la inicialización para dos experimentos con 5 y 10 mutaciones de proteasa. Los resultados son el promedio de 5 repeticiones, los modelos se evaluaron en términos de calidad predictiva (PRB), error temporal y número de intervalos.

Exp.	**Intervalos iniciales**	**PRB**	**Error temporal**	**Núm. total de intervalos**
1	2	**89.8**	**47.7**	**17.4**
	3	88.3	49.1	20.2
	4	88.5	49.5	18.8
2	2	87.3	54.9	**30.2**
	3	**88.5**	**54.6**	30.8
	4	87.5	55.1	35.0

La tabla 6.4 resume los resultados para los dos experimentos. Para ambos experimentos se usó un 80 % de los datos para aprender el modelo y el 20 % restante para evaluación, cada experimento se repitió 5 veces y se muestran los promedios de ellos.

En la figura 6.7 se presenta la RBNT del primer experimento que obtuvo los mejores puntajes. En esta figura se presenta la estructura de la red, los intervalos y las probabilidades a priori para los nodos temporales.

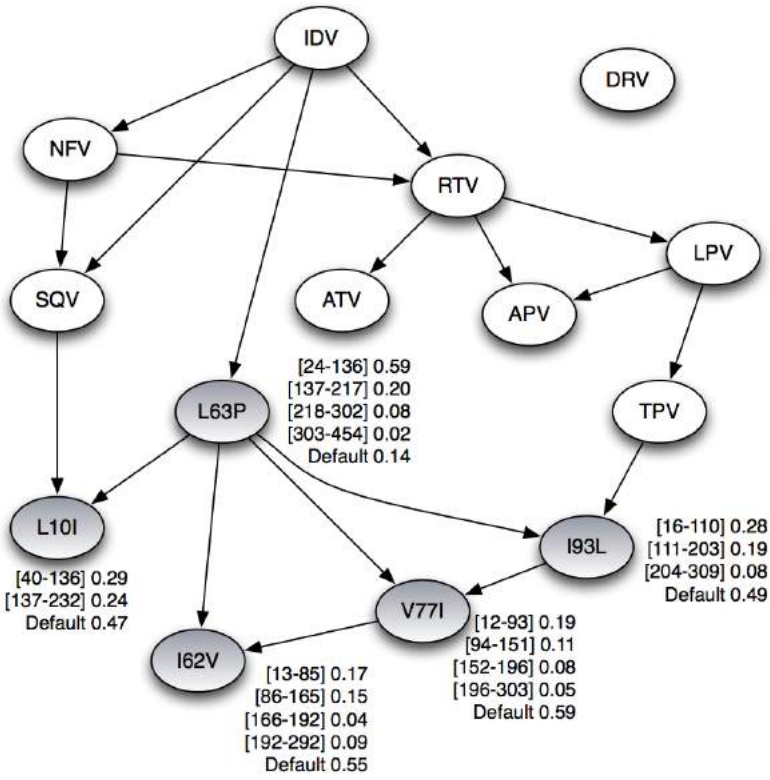


Figura 6.7: Una RBNT aprendida con 9 inhibidores de proteasa y 5 mutaciones que aparecen frecuentemente. Este modelo presenta en nodos en blanco a los medicamentos y en nodos color gris a las mutaciones, además estos nodos tienen intervalos asociados y probabilidades.

Cabe mencionar que las observaciones presentadas a continuación fueron validadas por los expertos en el dominio[2].

- RTV esta relacionado con IDV, NFV, SQV, ATV, APV y LPV. Esta relación de RTV con otros medicamentos, es explicada debido al hecho de que el medicamento Ritonavir ha sido probado como catalizador de otros IP y por lo tanto la mayoría de las ocasiones siempre se administra en combinación con otros medicamentos.

- La relación entre SQV y L10I era previamente conocida por los expertos (Johnson et al. 2010) y nuestro modelo fue exitoso al descubrirlo usando solamente datos.

- El nodo DRV en el modelo esta aislado debido a que en los datos nunca fue usado como parte del tratamiento inicial de los pacientes. La razón es que DRV es un nuevo medicamento.

- La mutación L63P es extremadamente común, tal como lo muestra el histograma de la figura 6.6. Sin embargo, el modelo obtenido sugiere que en la mayoría de las veces esta mutación tiende a ocurrir inicialmente y las probabilidades de ocurrir disminuyen conforme pasa el tiempo.

En la figura 6.8 se muestra la RBNT que obtuvo el mayor puntaje predictivo para el segundo experimento. La mayoría de los arcos del modelo más pequeño se mantuvieron. Solo la relación entre I62V y V77I desapareció. Además, se añadieron dos nuevos arcos de SVQ y TPV hacia L63P, entre los elementos considerados previamente. Esta ligera variación de los dos modelos es un buen indicador de la robustez del modelo.

En este modelo más completo se muestra la evidente relación de L63P con la mayoría de los medicamentos. De ahí se divide en diferentes mutaciones. Existen dos posibles explicaciones para esta observación: la frecuencia de ocurrencia de aparición de L63P esta sesgando el modelo, o L63P es una mutación importante que desencadena las demás. Por último podemos mencionar que es evidente que existen dos redes mutacionales importantes:

[2]El experto que evaluó los modelos fue el Dr. Santiago Ávila quién es investigador del Centro de Investigación en Enfermedades Infecciosas y tiene una larga experiencia en el estudio del VIH.

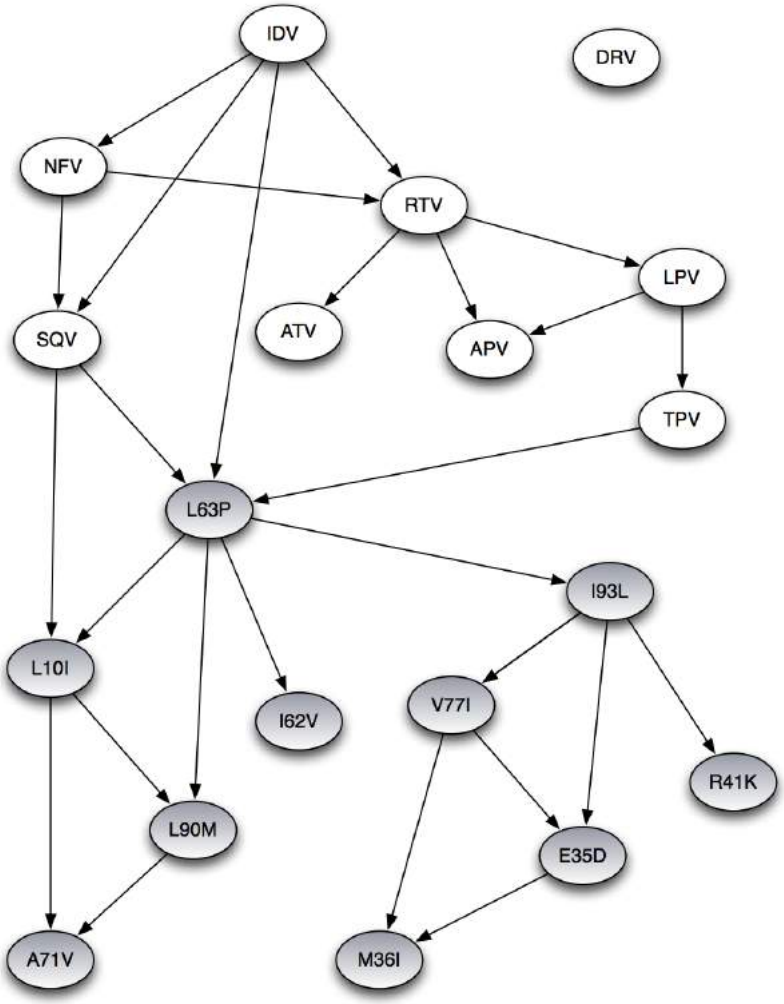


Figura 6.8: Una RBNT aprendida con 9 inhibidores de proteasa y 10 mutaciones que aparecen frecuentemente. Este modelo presenta en nodos en blanco a los medicamentos y en color gris a las mutaciones.

* L63P, L10I, L90M y A71V

* I93L, V77I, M36I, E35D y R41K

Sabemos que los resultados presentados son preliminares pero cumplen con el objetivo de hacer un análisis exploratorio de los datos. Más aún, sabemos que es necesario ampliar los experimentos y realizar un análisis exhaustivo de ellos, sin embargo escapan del alance de esta tesis.

6.3. Resumen

En este capítulo se presentaron los experimentos realizados con datos reales en dominios muy distintos: el primero un problema industrial de diagnóstico de fallas en una planta eléctrica, el segundo un problema médico para encontrar redes mutacionales importantes en un componente del VIH. Con estas aplicaciones se muestra que las RBNTs son una alternativa para modelar procesos temporales que resulta en modelos intuitivos y simples.

En ambas aplicaciones los algoritmos fueron evaluados de forma cualitativa por expertos, debido a que no es posible comparar contra redes de referencia puesto que no existen. Es importante mencionar que los expertos en los dominios aplicados pudieron entender los modelos obtenidos aún sin conocer toda la especificación formal o las teorias en las que se fundamente, lo que muestra que los modelos gráficos probabilistas y en particular las RBNT tienen una interpretación natural.

En específico para la aplicación en datos de VIH, creemos que los resultados presentandos aquí son una aproximación inicial pero exitosa que puede dar lugar a una linea de investigación para usar modelos temporales para explorar las mutaciones en las enzimas de VIH, que pueda derivar en un mejor entendimiento del virus y por lo consiguiente en una mejor aplicación de las terapias antiretrovirales.

Las pruebas presentadas en este capítulo son un indicador de que el algoritmo propuesto puede ser usado en diferentes aplicaciones reales obteniendo buenos resultados.

Capítulo 7

Conclusiones y Trabajo Futuro

7.1. Resumen

Dentro de las redes bayesianas han surgido varias extensiones del modelo básico (conocido como red bayesiana estática), las cuales incorporan al aspecto de temporalidad. En particular podemos encontrar al grupo de Redes Bayesianas de Eventos. En este grupo se encuentran las Redes Bayesianas de Nodos Temporales (RBNT).

Los trabajos existentes sobre RBNT obtienen la estructura, las probabilidades y los intervalos temporales con ayuda de expertos. Este proceso se aplicaba debido a que no existían métodos automáticos para obtenerlos. Lo anterior generaba principalmente dos problemas. El primero es que a veces no existe un consenso entre los expertos de cómo se debe especificar el modelo. El segundo problema surge en tratar de modelar procesos más complejos, ya que esta tarea de obtención de la estructura de la red y sus parámetros, es compleja y requiere de mucho tiempo de los expertos que normalmente están muy ocupados; además de que no se asegura que los resultados sean los mejores.

En esta tesis se presenta un algoritmo de aprendizaje de redes bayesianas de nodos temporales, el cual consta de tres fases principales:

1. Realizar una discretización inicial de las variables temporales, por ejemplo usando un algoritmo de discretización uniforme o un algoritmo basando en *K-means*. Con

este proceso se obtiene una aproximación inicial a los intervalos de todos los nodos temporales.

2. Se realiza un aprendizaje estructural estándar, el algoritmo usado es el conocido como K2 (Cooper y Herskovits 1992), con el se obtiene una estructura inicial la cual será usada en el tercer paso, el algoritmo de aprendizaje de intervalos.

3. El algoritmo de aprendizaje de intervalos refina los intervalos de cada nodo temporal por medio de un algoritmo de agrupamiento. Para esto se usan las configuraciones de los nodos padres. Para obtener los intervalos se usa una aproximación basada en el modelo de mezcla de gausianas. Cada grupo corresponde en principio a un intervalo temporal. Posteriormente estos intervalos se combinan y se selecciona el conjunto de intervalos que mejor calidad predictiva obtenga. Cuando se da el caso que un nodo temporal es padre de otro nodo temporal, entonces los intervalos se obtienen de forma secuencial de arriba hacia abajo de acuerdo a la estructura de la RBNT.

Para evaluar el algoritmo se realizaron pruebas con 3 RBNTs de diferente tamaños, además los datos temporales de las redes se generaron basándose en dos distintas distribuciones, gausiana y uniforme. Se evaluaron las redes obtenidas en términos de calidad estructural, en calidad de los intervalos y calidad de la red en su conjunto. Con el algoritmo desarrollado en esta tesis se realizaron varios experimentos de los cuales se extraen las siguientes conclusiones:

- El algoritmo propuesto obtuvo los mejores resultados en calidad estructural y error temporal en comparación con dos algoritmos base: discretización uniforme y *K-means*.

- El algoritmo propuesto superó en promedio a los algoritmos base en las medidas estructurales, de intervalos y de predicción.

- Aún cuando el algoritmo propuesto hace la suposición de que los datos son gausianos, el algoritmo se evaluó con datos uniformes obteniendo buenos resultados incluso superando a los algoritmos base.

- Nuestro algoritmo se base en agrupamiento para obtener los intervalos temporales, mientras que (Friedman y Goldszmidt 1996) hace uso de un puntaje basado en el principio de descripción de longitud mínima para obtener una discretización, la cual será interpretada como intervalos temporales.

- Nuestro algoritmo realiza el aprendizaje de intervalos de arriba-abajo, por niveles, dependiendo de la estructura, el algoritmo de (Friedman y Goldszmidt 1996) discretiza variable por variable asumiendo las demás discretizaciones fijas.

- Nuestro algoritmo obtiene tiempos de ejecución menores a los obtenidos por el algoritmo de Friedman. Cuando el número de casos de entrada aumenta, nuestro algoritmo tiene un comportamiento aproximadamente polinomial mientras que el algoritmo de Friedman es aproximadamente exponencial.

- Una de las limitaciones de nuestro algoritmo es que en algunas ocasiones obtuvo una calidad predictiva menor a la obtenida por el algoritmo de (Friedman y Goldszmidt 1996). Una posible razón a esto es que el algoritmo de Friedman obtiene generalmente menos intervalos y de mayor tamaño.

7.2. Aportaciones

La aportación principal de esta tesis es un algoritmo de aprendizaje para Redes Bayesianas de Nodos Temporales. En particular:

- Se desarrolló un algoritmo de aprendizaje de intervalos para los Nodos Temporales de una RBNT.

El algoritmo fue usado en dos dominios reales:

- Para diagnosticar de fallas en una parte de un subsistema de una planta eléctrica de ciclo combinado.

- Para obtener redes mutacionales, es decir, redes temporales en donde se relacionan los fármacos usados y las mutaciones ocurridas en la enzima proteasa del VIH.

7.3. Trabajo Futuro

Algunas ideas de trabajo futuro se presentan a continuación:

- Respecto al algoritmo se realizaron experimentos con distribuciones uniforme y gausiana. Sería interesante probar con más distribuciones como la distribución exponencial o probar mezclando distribuciones.

- Ya se mencionó que el tamaño de los intervalos puede tener impacto en la calidad predictiva, por lo que una medida combinada, también se propone para trabajo futuro.

- Otro aspecto interesante es que las inicializaciones que se usaron en los experimentos varían el número de intervalos iniciales para *todos* los nodos temporales. Una propuesta para trabajo futuro es obtener cual es el mejor número de intervalos iniciales para cada uno de los nodos temporales de la red.

- En los experimentos con datos de pacientes de VIH se trabajó con la enzima proteasa. Una propuesta de trabajo futuro es aplicar el algoritmo a datos de otra enzima del VIH conocida como transcriptasa reversa.

- Por último, sería interesante hacer una investigación más a fondo de cómo se comporta el algoritmo en cada una de las iteraciones de los pasos de aprendizaje estructural y aprendizaje de intervalos.

Bibliografía

Arroyo-Figueroa, G. y L. E. Sucar (1999). "A temporal Bayesian network for diagnosis and prediction". En: *Proceedings of the 15th UAI Conference*. Stockholm, Sweden, págs. 13-22.

Arroyo-Figueroa, G., L.E. Sucar y A. Villavicencio (1998). "Probabilistic temporal reasoning and its application to fossil power plant operation". En: *Expert Systems with Applications* 15.3, págs. 317-324.

Beerenwinkel, N., B. Schmidt, H. Walter, R. Kaiser, T. Lengauer, D. Hoffmann, K. Korn y J. Selbig (2002). "Diversity and complexity of HIV-1 drug resistance: a bioinformatics approach to predicting phenotype from genotype". En: *Proceedings of the National Academy of Sciences of the United States of America* 99.12, págs. 8271-8276.

Bishop, C.M. (2006). *Pattern recognition and machine learning*. Springer New York.

Charitos, T., L.C. van der Gaag, S. Visscher, K.A.M. Schurink y P.J.F. Lucas (2009). "A dynamic Bayesian network for diagnosing ventilator-associated pneumonia in ICU patients". En: *Expert Systems with Applications* 36.2, págs. 1249-1258.

Chausa, P., C. Cáceres, L. Sacchi, A. León, F. García, R. Bellazzi y E. Gómez (2009). "Temporal Data Mining of HIV Registries: Results from a 25 Years Follow-Up". En: *Artificial Intelligence in Medicine*, págs. 56-60.

Chickering, D.M., D. Geiger y D. Heckerman (1994). "Learning Bayesian networks is NP-hard". En: *Microsoft Research*, págs. 94-17.

Cooper, G.F. y E. Herskovits (1992). "A Bayesian method for the induction of probabilistic networks from data". En: *Machine learning* 9.4, págs. 309-347.

Córdoba Villalobos, J. A., Samuel Ponce de Leon Rosales y José Luis Valdespino, eds. (2009). *25 años de SIDA en México: Logros, desaciertos y retos*. 2.ª ed. Instituto Nacional de Salud Pública - Secretaría de Salud.

Dagum, P., A. Galper y E. Horvitz (1992). "Dynamic network models for forecasting". En: *Proceedings of the 8th Workshop UAI*. Stanford, California, USA, págs. 41-48.

Daly, R. y Q. Shen (2009). "Learning Bayesian network equivalence classes with ant colony optimization". En: *Artificial Intelligence Research* 35, págs. 391-447.

Day, N.E. (1969). "Estimating the components of a mixture of normal distributions". En: *Biometrika* 56.3, págs. 463-474. ISSN: 0006-3444.

Dempster, A.P., N.M. Laird y D.B. Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". En: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1, págs. 1-38. ISSN: 0035-9246.

Draghici, Sorin y R. Brian Potter (2003). "Predicting HIV drug resistance with neural networks". En: *Bioinformatics* 19.1, págs. 98-107.

Eaton, D. y K. Murphy (2007). "Bayesian structure learning using dynamic programming and MCMC". En: *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*. Vancouver, BC, Canada.

Feelders, A. y L.C. Van der Gaag (2006). "Learning Bayesian network parameters under order constraints". En: *International Journal of Approximate Reasoning* 42.1-2, págs. 37-53.

Freeman, S., J.C. Herron y M. Payton (1998). *Evolutionary analysis*. Prentice Hall Upper Saddle River, NJ: ISBN: 0135680239.

Friedman, N. y M. Goldszmidt (1996). "Discretizing continuous attributes while learning Bayesian networks". En: *Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96)*. Bari, Italy, págs. 157-165.

Galán, Severino F., Severino F. Galán, Francisco Aguado, Francisco J. Díez y José" Mira (2001). "NasoNet: joining Bayesian networks and time to model nasopharyngeal cancer spread". En: *Proceedings of the Eighth International Conference on Artificial Intelligence in Medicine in Europe (AIME 2001)*. Cascais, Portugal: Springer-Verlag, págs. 207-216.

Galán, S.F. y F.J. Díez (2002). "Networks of probabilistic events in discrete time". En: *International Journal of Approximate Reasoning* 30.3, págs. 181-202.

Galán, S.F., G. Arroyo-Figueroa, F.J. Díez y L.E. Sucar (2007). "Comparison of two types of event Bayesian networks: A case study". En: *Applied Artificial Intelligence* 21.3, pág. 185.

Ghahramani, Z. (1998). "Learning dynamic Bayesian networks". En: *Adaptive Processing of Sequences and Data Structures*, pág. 168.

Heckerman, D. (2008). "A tutorial on learning with Bayesian networks". En: *Innovations in Bayesian Networks*, págs. 33-82.

Hemelaara, Joris, Eleanor Gouws, Peter D. Ghys y Saladin Osmanov (2006). "Global and regional distribution of HIV-1 genetic subtypes and recombinants in 2004". En: *AIDS* 20, W13-W23.

Jensen, C.S., R.T. Snodgrass y M.D. Soo (2002). "Extending existing dependency theory to temporal databases". En: *Knowledge and Data Engineering, IEEE Transactions on* 8.4, págs. 563-582. ISSN: 1041-4347.

Johnson, V.A., F. Brun-Vézinet, B. Clotet, H.F. G
ünthard, D.R. Kuritzkes, D. Pillay, J.M. Schapiro y D.D. Richman (2010). "Update of the Drug Resistance Mutations in HIV-1: December 2010". En: *Topics in HIV medicine* 17, págs. 138-145.

Knox, W.B. y O. Mengshoel (2009). "Diagnosis and Reconfiguration using Bayesian Networks: An Electrical Power System Case Study". En: *Proceedings of IJCAI-09 Workshop on Self-* and Autonomous Systems (SAS): Reasoning and Integration Challenges*. Pasadena, California, US.

Koller, D. y N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press.

Kotsiantis, S. y D. Kanellopoulos (2006). "Discretization techniques: A recent survey". En: *GESTS International Transactions on Computer Science and Engineering* 32.1, págs. 47-58.

Lam, W., F. Bacchus, University of Waterloo. Dept. of Computer Science y University of Waterloo. Faculty of Mathematics (1994). "Learning Bayesian belief networks: An approach based on the MDL principle". En: *Computational intelligence* 10.4, págs. 269-293.

Liu, H., F. Hussain, C.L. Tan y M. Dash (2002). "Discretization: An enabling technique". En: *Data Mining and Knowledge Discovery* 6.4, págs. 393-423. ISSN: 1384-5810.

Liu, W.Y., N. Song y H. Yao (2005). "Temporal functional dependencies and temporal nodes Bayesian networks". En: *The Computer Journal* 48.1, págs. 30-41.

Moon, T.K. (1996). "The expectation-maximization algorithm". En: *IEEE Signal processing magazine* 13.6, págs. 47-60.

Murphy, Kevin Patrick (2002). "Dynamic Bayesian Networks: Representation, Inference and Learning". Tesis doct. University of California, Berkeley.

Neapolitan, R.E. (2004). *Learning bayesian networks*. Pearson Prentice Hall Upper Saddle River, NJ. ISBN: 0130125342.

— (2009). *Probabilistic Methods for Bioinformatics: With an Introduction to Bayesian Networks*. Morgan Kaufmann.

Neapolitan, R.E. y X. Jiang (2007). *Probabilistic methods for financial and marketing informatics*. Morgan Kaufmann.

Nodelman, U., C.R. Shelton y D. Koller (2002). "Continuous time Bayesian networks". En: *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*. Edmonton, Alberta, Canada, págs. 378-387.

— (2003). "Learning continuous time Bayesian networks". En: *Proceedings of the Nineteenth International Conference on Uncertainty in Artificial Intelligence*. Vol. 451458. Acapulco, Mexico.

Pearl, Judea (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 0-934613-73-7.

Ramirez, J.C.G., D.J. Cook, L.L. Peterson y D.M. Peterson (2000). "Temporal pattern discovery in course-of-disease data". En: *Engineering in Medicine and Biology Magazine, IEEE* 19.4, págs. 63-71. ISSN: 0739-5175.

Reyes-Ballesteros, Alberto (2006). "Representación y Aprendizaje de Procesos de Decisión de Markov Cualitativos." Tesis doct. Instituto Tecnológico y de Estudios Superiores de Monterrey.

Rhee, S.Y., M.J. Gonzales, R. Kantor, B.J. Betts, J. Ravela y R.W. Shafer (2003). "Human immunodeficiency virus reverse transcriptase and protease sequence database". En: *Nucleic acids research* 31.1, págs. 298-303. ISSN: 0305-1048.

Robinson, R. (1977). "Counting unlabeled acyclic digraphs". En: *Combinatorial mathematics V*, págs. 28-43.

Spirtes, P. y C. Glymour (1991). "An algorithm for fast recovery of sparse causal graphs". En: *Social Science Computer Review* 9.1, pág. 62.

Tan, P.N., M. Steinbach, V. Kumar et al. (2006). *Introduction to data mining*. Pearson Addison Wesley Boston. ISBN: 0321321367.

Weiss, RA (1993). "How does HIV cause AIDS?" En: *Science* 260.5112, págs. 1273-1279. DOI: 10.1126/science.8493571.

Wijsen, J. (1995). "Design of temporal relational databases based on dynamic and temporal functional dependencies". En: *Proceedings of the International Workshop on Temporal Databases: Recent Advances in Temporal Databases*. Springer-Verlag, págs. 61-76. ISBN: 3540199454.

Wu, X., P. Lucas, S. Kerr y R. Dijkhuizen (2001). "Learning bayesian-network topologies in realistic medical domains". En: *Medical Data Analysis*, págs. 302-307.

Apéndice A

Artículos aceptados

Como resultado de este trabajo de investigación se realizaron las siguientes publicaciones:

- Hernandez-Leal, Pablo, L. Enrique Sucar y Jesus A. Gonzalez (2011). "Learning Temporal Nodes Bayesian Networks". En: The 24th Florida Artificial Intelligence Research Society Conference (FLAIRS-24). Palm Beach, Florida, USA.

- Hernandez-Leal, Pablo, L. Enrique Sucar, Jesus A. Gonzalez, Eduardo F. Morales y Pablo H. Ibarguengoytia (2011). "Learning temporal Bayesian networks for power plant diagnosis". En: The Twenty-fourth International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE 2011). Syracuse, New York, USA.

- Hernandez-Leal, Pablo, Alma Rios-Flores, Felipe Orihuela-Espina, Santiago Ávila-Rios, Gustavo Reyes-Terán, Jesus A. González, Eduardo F. Morales y L. Enrique Sucar (2011) "Unveiling HIV mutational networks associated to pharmacological selective pressure: a temporal Bayesian approach" En: AIME'11 Workshop on Probabilistic Problem Solving in Biomedicine. Bled, Eslovenia.