

# Aprendizaje Paramétrico en las redes bayesianas para el Diagnóstico de enfermedades cardiovasculares

Parametric Learning in Bayesian networks for diagnosis of cardiovascular diseases

Guillermo Roberto Solarte-Martínez<sup>1</sup>, José A. Soto-Mejía<sup>2</sup>, Carlos Augusto Meneses-Escobar<sup>3</sup>

<sup>1,3</sup>Facultad de Ingenierías, programa de Sistemas y Computación. Semillero de Optimización GNTO, Universidad Tecnológica de Pereira, Pereira Risaralda, Colombia

Correo-e: gsolarte294@gmail.com, cmeneses@utp.edu.co

<sup>2</sup>Facultad de Ingeniería Industrial, Semillero de Optimización GNTO, Universidad Tecnológica de Pereira, Pereira-Risaralda, Colombia.

Correo-e: jomejia@utp.edu.co

**Resumen**— En esta investigación se muestra que el aprendizaje paramétrico en una red bayesiana puede ser usado en el diagnóstico de enfermedades cardiovasculares. El objetivo es determinar si se debe o no aplicar fármacos a un paciente con enfermedad cardiovascular utilizando como herramienta una aplicación de software en Java, realizada por los autores. Las redes bayesianas se utilizan como representación gráfica del conocimiento previo y métodos de razonamiento en modelos probabilísticos. Para la creación de una red existen dos fases de aprendizaje: aprendizaje estructural y aprendizaje paramétrico. En este estudio se hizo uso del aprendizaje paramétrico.

**Palabras clave**— redes Bayesianas, Bases de Datos, diagnóstico, enfermedades, aprendizaje paramétrico.

**Abstract**—This paper describes an ongoing research on Bayesian networks used to diagnose a cardiovascular disease with parametric learning. The objective is to determine whether or not to apply drug to patients with cardiovascular disease using a Java software application written by the authors. Bayesian networks are used as a graphical representation of prior knowledge and reasoning methods on probabilistic models. For the creation of a network there are two phases of learning: structural learning and parametric learning. In this study the parametric learning was used.

**Key Word** — Bayesian networks, databases, diagnostics, disease, parametric learning

## I. INTRODUCCIÓN

Una red bayesiana [1], es una representación gráfica (grafos dirigidos a cíclicos), que contiene información probabilística para realizar un razonamiento probabilístico. Esta red está compuesta por nodos y arcos, donde los nodos representan variables aleatorias que pueden ser continuas o discretas y los arcos interpretan influencias causales. El que un nodo sea padre de otro implica que es causa directa del mismo.

Conceptos claves de redes bayesianas<sup>1</sup>

• **Probabilidad a priori:** Es la probabilidad que tiene una variable en ausencia de evidencia

• **Probabilidad a posteriori:** Es la probabilidad de una variable condicionada a la existencia de una evidencia determinada.

• **Probabilidad conjunta :** Si se tiene un colección de variables  $\{X, Y, K, Z\}$ , la probabilidad conjunta especifica la probabilidad de cada combinación posible de estados de cada variable  $P(X_i | X_{i-1}, \dots, Z_k) \forall i, j, \dots, K$  de manera que se cumple que;

$$\sum_{i, j, \dots, k} P(X_i, Y_j, \dots, Z_k) = 1$$

• **Probabilidad condicional:** Proporcionadas dos variables X e Y, la probabilidad de que ocurra  $y_i$  dado que ocurrió el evento  $x_i$  es la probabilidad condicional de Y dado X.

En esta investigación se demuestra empíricamente que es posible diagnosticar la necesidad o no de administrar fármacos en pacientes con síntomas de enfermedad cardiovascular, usando las variables presión arterial, índice de colesterol, azúcar en la sangre, alergias a antibióticos y otras alergias, mediante la utilización de redes bayesianas con aprendizaje paramétrico a través de una aplicación implementado por los autores en el lenguaje Java. En la segunda sección de este artículo se realiza una descripción del aprendizaje paramétrico y del algoritmo de propagación implementado y en la sección 3 se muestran los resultados de esta investigación. El artículo termina con una sección de conclusiones.

## II. CONTENIDO

### A. El Aprendizaje en las Redes Bayesianas

Se fundamenta en definir en redes probabilísticas la información de datos almacenados en bases de datos con el objeto de tomar de decisiones en lugar de utilizar la ayuda de un experto.

Por lo tanto este tipo de almacenamiento ofrece la posibilidad de definir la estructura grafica de la red a partir de los datos almacenados (observados) definiendo las asociaciones entre los nodos o variables.

<sup>1</sup>Modelos Gráficos Probabilistas, Principios y Aplicaciones Página Curso: [ccc.inaoep.mx/~esucar/Clases-mgp/mgp.html](http://ccc.inaoep.mx/~esucar/Clases-mgp/mgp.html)  
Dr. L. Enrique Sucar S. Ciencias Computacionales, INAOE  
Publicado 20 junio de 2011

Según Pearl [7] existen dos fases de aprendizaje: aprendizaje estructural y aprendizaje paramétrico.

#### 1. Aprendizaje estructural

Consiste en obtener la estructura de la red bayesiana a partir de la base de datos, o sea, las relaciones de dependencia y/o independencia entre las variables involucradas. Las técnicas de aprendizaje estructural dependen del tipo de organización de la red (árboles, poli árboles o redes multiconectadas).

#### 2. Aprendizaje paramétrico:

El aprendizaje paramétrico se fundamenta en descubrir los parámetros asociados a la estructura dada de una red bayesiana [6]. Es decir, lo que se pretende es encontrar las probabilidades a priori de los nodos raíz y las probabilidades condicionales de las demás variables dados sus padres, de acuerdo con la información disponible.

Uno de los principales trabajos en el campo del aprendizaje de redes bayesianas es el de Herkovits y Copper [4]. Si se conocen los valores de todas las variables es relativamente sencillo obtener las probabilidades requeridas, ya que las probabilidades previas corresponden a las marginales de los nodos raíz y las condicionales se obtienen del agrupamiento de cada nodo con su(s) padre(s).

Para actualizar las probabilidades de cada nodo observado, cuando se utiliza una estructura en árbol, se hace uso de las siguientes fórmulas:

Fórmula de probabilidades previas:

$$P(A_i) = \frac{\binom{a_i + 1}{s + 1}}{\binom{s + 1}{s + 1}}; \quad i = k$$

$$P(A_i) = \frac{\binom{a_i}{s + 1}}{\binom{s + 1}{s + 1}}; \quad i \neq k$$

Fórmula probabilidades condicionales

$$P(B_j|A_i) = \binom{b_j + 1}{a_i + 1} \quad i = k \quad j = 1$$

$$P(B_j|A_i) = \binom{b_j}{a_i + 1} \quad i = k \quad j \neq 1$$

$$P(B_j|A_i) = \binom{b_j}{a_i} \quad i \neq k$$

Donde:

- ✓  $P(A_i)$ : es la probabilidad asociada con el evento (nodo) de interés.
- ✓  $s$ , corresponde al número de casos totales,
- ✓  $i, j$  los índices de las variables,  $k$ .
- ✓  $b_i, a_i$  es un conjunto de eventos
- ✓  $I$  los índices de las variables específicas siendo evaluadas

### B. Inferencia Bayesiana

Es un método de inferencia estadística [2] que permite ingresar nueva información u observaciones para recalcular las probabilidades que alcanzarán el resto de variables dada la información nueva conocida.

Este método lo que hace es calcular las probabilidades a posteriori  $P(X|Y = y_i)$  de un conjunto de variables  $X$  después de obtener un conjunto de observaciones  $Y = y_i$  donde  $Y$  es la lista de variables observadas e  $y_i$  es un valor particular tomado por ella.

#### Predicción

Si se tienen datos de un suceso anterior y este suceso se está representando en la red como un nodo padre la red puede conjeturar cuáles serán sus efectos. Para lograr esto se debe conservar la información en el nodo correspondiente y difundir esta información hacia el resto de los nodos [5].

#### 1. Interpretación de datos

Otra manera de predicción es a través de las mismas relaciones que se muestran en la red, donde conociendo las consecuencias, se puede saber cuáles son sus posibles causas. El conocimiento es el mismo que en el caso anterior: “si  $a$  entonces  $b$ ” pero ahora el hecho conocido es “ $b$ ” y el hecho desconocido es “ $a$ ”. Las probabilidades a posteriori  $P(X|Y = y_i)$ , se pueden conseguir desde la probabilidad marginal  $P(X|Y)$ , que a su vez puede obtenerse de la probabilidad conjunta  $P(x_1, x_2, \dots, x_i)$  sumando los valores para todas las variables que no pertenezcan al conjunto  $X \cup Y$ .

### C. Algoritmos de Propagación

Como se ha mencionado anteriormente una de las ventajas de las redes bayesianas es contar con una estructura grafica que facilita la representación de las relaciones entre las variables, ya que estas relaciones

reducen considerablemente el número de operaciones para obtener probabilidades a posteriori

1. Propagación en árboles

Suponga que cada nodo corresponde a una variable discreta  $A = (A_1, A_2, \dots, A_n)$  con su respectiva matriz de probabilidad condicional  $P(B | A) = P(B_j | A_i)$ . Dada cierta evidencia  $E$  (representada por la instanciación de ciertas variables) la probabilidad posterior de cualquier variable  $B$  es, por el teorema de Bayes:

$$P(B_i | E) = \frac{P(E | B_i) \cdot P(B_i)}{P(E)}$$

Suponiendo que la tipología escogida para la evaluación de la red es un árbol, el nodo  $B$ , se divide en dos subárboles; de este modo podemos separar la evidencia en dos grupos ( $E^-$ ;  $E^+$ ) así:

$E^-$ : Datos en el árbol cuya raíz es  $B$ .

$E^+$ : Datos en el resto del árbol.

Por lo tanto tenemos que:

$$P(B_i | E) = \frac{P(E^-, E^+ | B_i) \cdot P(B_i)}{P(E)}$$

Asumiendo la independencia de los dos grupos podemos aplicar nuevamente el teorema de Bayes, así:

$$P(B_i | E) = \alpha P[(B_i | E^+) (E^- | B_i)] \text{ donde } \alpha \text{ es una constante de normalización.}$$

La anterior expresión divide la evidencia para recalcular la probabilidad de  $B$ . Igualmente se observa que no se requiere de la probabilidad a priori en caso de que el nodo no posea padres.

Así se obtiene que:  $P(A_i | E^+) = P(A_i)$

Para reducir el proceso se definen los siguientes dos términos, llamados pi y lambda mensajes:

$$\lambda(B_i) = P(E^- | B_i)$$

$$\pi(B_i) = P(B_i | E^+)$$

Entonces,  $P(B_i | E) = \lambda \pi(B_i) P(B_i)$

Dado que los hijos son condicionalmente independientes dado los padres

$$\lambda(B_i) = \prod_k P(E_k^- | B_i) = \prod_k \lambda_k(B_i)$$

Donde  $E_k$  pertenece a la evidencia que se origina del hijo  $k$  de  $B$  denotado por  $S_k$ . Condicionando cada término en la ecuación anterior respecto de todos los posibles valores de cada nodo hijo se obtiene que  $B$  es condicionalmente independiente de la evidencia bajo cada hijo por lo tanto se usa la siguiente expresión de  $\lambda$

$$\lambda(B_i) = \prod_k \left[ \sum_j P(E_k^- | B_i, S_j^k) P(S_j^k | B_i) \right] \text{ De}$$

igual forma se obtiene una ecuación para  $\pi$ . Primero se le condiciona sobre todos los posibles valores del padre:

$$\pi(B_i) = \sum_j P(B_i | E^+, A_j) P(A_j | E^+) \text{ Luego}$$

podemos eliminar  $E^+$  del primer término dada la independencia condicional. El segundo término representa la probabilidad a posterior de  $A$  sin contar la evidencia del subárbol de  $B$ , por lo que podemos expresarla usando la

ecuación para  $P(B_j | E)$  y la descomposición de  $\lambda$ .

$$\pi(B_i) = \sum_j P(B_i | A_j) \left[ \alpha \pi(A_j) \prod_k \lambda_k(A_j) \right]$$

Donde  $k$  incluye a todos los hijos de  $A$  excepto  $B$ . Mediante estas ecuaciones se integra un algoritmo de propagación de probabilidades en árboles, donde cada nodo guarda los valores de los vectores  $\pi$  y  $\lambda$ , así también como las matrices de probabilidades,  $P(B)$ . la propagación se hace por un mecanismo de paso de mensajes en donde cada nodo envía los mensajes correspondientes a su padre e hijos:

Mensaje al padre (del hijo, nodo  $B$ , a su padre  $A$ )

$$\lambda(A_i) = \sum_j P(B_i | A_i) \lambda(B_j)$$

Mensaje a los hijos (nodo  $B$  a su hijo,  $S_k$ )

$$\pi_k(B_i) = \alpha \pi(B_i) \prod_{l \neq k} \lambda_l(B_j)$$

Al instanciarse ciertos nodos, éstos envían mensajes a sus padres e hijos y se propagan hasta llegar a la raíz u hojas o hasta encontrar un nodo instanciado, así que la propagación se hace en un solo paso en un tiempo proporcional al diámetro de la red. Esto se puede hacer en forma iterativa instanciando ciertas variables, propagando su efecto y luego instanciando otras variables y propagando la nueva información combinando ambas evidencias.

D. Algoritmo de Propagación de Probabilidades

Se han desarrollado varios algoritmos para el cálculo de las probabilidades. Se presenta a continuación el algoritmo para redes con forma de árbol. Este algoritmo consta de dos etapas:

1. Etapa de inicialización.

En esta etapa se registran las probabilidades a priori de cada uno de los nodos de la red, obteniéndose un estado inicial de la red que denotaremos por  $S_0$ .

2. Etapa de actualización.

Cuando una variable adquiere un valor nuevo, se actualiza el estado de la red (árbol), obteniéndose las probabilidades a posteriori de las variables del árbol basadas en la evidencia considerada, adoptando la red un estado que denotaremos por  $S_j$ .

Este proceso se repite cada vez que una variable tome un valor nuevo, obteniendo así la actualización de los estados

sucesivos de la red. La idea básica del algoritmo de propagación de probabilidades es enviar mensajes a toda la red (árbol), cada vez que existe un cambio en uno de los nodos (variables), se informa a sus nodos más cercanos mediante utilización de mensajes, de la siguiente forma:

- La variable manda a su padre un mensaje, que llamaremos el  $\lambda$ -mensaje, para comunicarle de que ha cambiado su valor.
- La variable manda a todos sus hijos un mensaje, que llamaremos el  $\pi$ -mensaje, para comunicarle de que ha cambiado su valor. Además a cada uno de los mensajes se le asignan unos valores que llamaremos  $\lambda$ -valor y  $\pi$ -valor. Por ejemplo.



Figura 1. Representación de un arco A a B  
Si se tiene un arco de A a B en el que la variable A toma tres valores posibles  $a_1, a_2, a_3$ , y la variable B toma dos,  $b_1$  y  $b_2$ , tendríamos que:

- Si B se instancia, enviará un  $\lambda$ -mensaje a A,  $\lambda_B(A) = (\lambda_B(a_1), \lambda_B(a_2), \lambda_B(a_3))$ .
- Si A se instancia, enviará un  $\pi$ -mensaje a B,  $\pi_B(A) = (\pi_B(b_1), \pi_B(b_2))$ .

E. Fórmulas<sup>2</sup>

Para el cálculo de  $\lambda$  y  $\pi$ -mensajes,  $\lambda$  y  $\pi$ -valores y probabilidades de  $P^*$  (esto es la probabilidad a posteriori dada una evidencia observada,  $(P^*(x) = P(x/e))$ ), se tienen las siguientes ecuaciones.

1. Si B es un hijo de A, y B tiene k valores posibles, y A m valores posibles, entonces para  $j=1,2,\dots, m$ , el  $\lambda$ -mensaje de B a A viene dado por:

$$\lambda_B(a_j) = \sum_{i=1}^k P(b_i / a_j) \lambda(b_i) \quad \text{Ecuación 1}$$

2. Si B es hijo de A, y A tiene m valores posible, entonces para  $j=1,2,\dots, m$ , el  $\pi$ -mensaje de A a B viene dado por:

$$\pi_B(a_j) = \begin{cases} \pi(a_j) \cdot \prod_{\substack{C \in S(A) \\ C \neq B}} \lambda_C(a_j) & \text{si A no ha sido instanciada} \\ 1 & \text{Si } A = a_j \\ 0 & \text{Si } A \neq a_j \end{cases} \quad \text{Ecuación 2}$$

Donde  $s(A)$  denota al conjunto de hijos de A.

3. Si B tiene k valores posibles y  $s(B)$  es el conjunto de los hijos de B, entonces para  $i=1,2,\dots,k$ , el  $\lambda$ -valor de B viene dado por:

$$\lambda_B(b_i) = \begin{cases} \prod_{C \in S(B)} \lambda_C(b_i) & \text{si B no ha sido instanciada} \\ 1 & \text{Si } B = b_i \\ 0 & \text{Si } B \neq b_i \end{cases} \quad \text{Ecuación 3}$$

4. Si A es padre de B, B tiene k valores posibles y A tiene m valores posibles, entonces para  $i=1,2,\dots,k$ , el  $\pi$ -valor de B viene dado por:

$$\pi(b_i) = \sum_{j=1}^m P(b_i / a_j) \pi_B(a_j) \quad \text{Ecuación 4}$$

<sup>2,3</sup> Tomado de [Neapolitan, Richard E, 1990] Probabilistic reasoning in expert systems: Theory and Algorithms <http://www.getcited.org/pub/102782375>

5. Si B es una variable con k posibles valores, entonces para  $i = 1,2,\dots,k$  la probabilidad a posteriori ( $P^*$ ) basada en las variables instanciadas se calcula como:

$$P^*(b_i) = \alpha(b) \cdot \pi(b_i) \quad \text{Ecuación 5}$$

Para ilustrar el algoritmo de propagación de probabilidades vamos a utilizar la estructura de la red mostrada en la figura 2. Este estructura, para decidir si se debe o no administrar fármacos a un paciente con enfermedad cardiovascular, se obtuvo mediante la técnica de arboles de decisión ID3[3]. (Induction Decisión tree)<sup>4</sup>

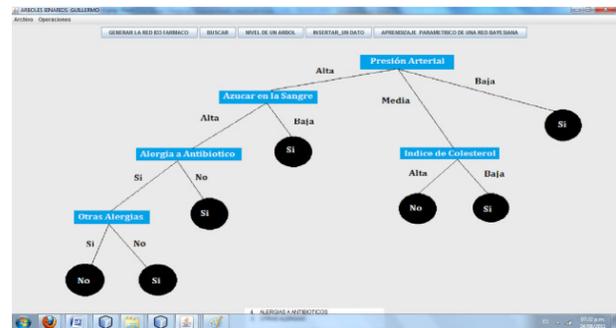


Imagen 1. Estructura de la red

<sup>4,5</sup> Detalles sobre este proceso pueden ser consultados en el artículo de los mismos autores: en la revista Scientia et Technica Vol 1, No 49 (2011). Universidad Tecnológica de Pereira

Paciente	Presión Arterial	Azúcar en la Sangre	Índice de Colesterol	Alergias a Antibióticos	Otras Alergias	Administrar Fármacos
1	Alta	Alto	Alto	NO	NO	SI
2	Alta	Alto	Alto	SI	NO	SI
3	Baja	Alto	Alto	NO	NO	SI
4	Media	Alto	Alto	SI	SI	SI
5	Media	Bajo	Alto	SI	SI	NO
6	Baja	Bajo	Alto	SI	SI	SI
7	Alta	Bajo	Alto	SI	NO	SI
8	Alta	Alto	Bajo	SI	SI	NO
9	Alta	Alto	Bajo	SI	SI	SI
10	Baja	Bajo	Alto	SI	SI	SI
11	Media	Bajo	Bajo	NO	SI	SI
12	Alta	Alto	Alto	SI	SI	NO
13	Baja	Bajo	Alto	NO	SI	SI
14	Baja	Bajo	Bajo	NO	NO	No

Tabla 1 Datos médicos de 14 Pacientes<sup>5</sup>

Primero se procede a calcular las probabilidades asociadas al nodo raíz, “Presión Arterial”. Suponga que en la evaluación de 14 pacientes se obtuvo que 4 de ellos tienen presión arterial alta y que solo 1 de ellos tiene presión arterial media como se observa en la tabla 2 de contingencia para la presión arterial, esta tabla se obtiene a partir de los datos de la tabla 1. Datos médicos de 14 Pacientes.

	Alto	Medio	Bajo	Total
Si	4	2	5	10
No	2	1	0	4
Total	6	3	5	14

Tabla 2 Contingencia para presión arterial  
Fuente: autores

A manera de ilustración el cálculo de la probabilidad de los pacientes que si tienen presión alta es:

$$\frac{4}{10} = 0.4$$

La probabilidad de **no** de presión alta sería  $\frac{2}{4} = 0,5$

De igual manera se realiza los cálculos de las probabilidades iniciales para los demás nodos.

#### F. Resultados de la investigación

A continuación se muestra el código Java que lee la Matriz de frecuencias absolutas y genera la matriz de probabilidades iniciales.

```
public double [ ][ ] matrizProb(double b[ ][ ])
{
    int i,j;
    for (i = 0; i < b.length; i++){
        for (j = 0; j < b[i].length; j++) {
```

```
            if (i==1) {b[i][j]=b[i][j]/ps;}
            if (i==2) {b[i][j]=b[i][j]/pn;}
        }
    }
    return b; }
```

#### Código 1 Cálculo de probabilidades iniciales de datos

A continuación se muestran las probabilidades iniciales para cada uno de los nodos generadas por el código anterior.

Probabilidad	Alta	Media	Baja
Si	0.4	0.1	0.5
No	0.5	0.5	0.0

Tabla 3 Probabilidad Presión Arterial

Probabilidad	Alta	Baja
Si	0.5	0.5
No	0.5	0.5

Tabla 4 Probabilidad Azúcar en Sangre

Probabilidad	Alta	Baja
Si	0.6	0.4
No	0.75	0.25

Tabla 5 Probabilidad Índice de Colesterol

Probabilidad	SI	NO
Si	0.4	0.4
No	1.25	0.25

Tabla 6 Probabilidad Alergia Antibiótico

Probabilidad	SI	NO
Si	0.5	0.4
No	1	0.25

Tabla 7 Probabilidad Alergia Antibiótico

	Probabilidad
Si	0.71428
No	0,2857

Tabla 8 Probabilidad Administrar Fármacos

A continuación con el objeto de comparar los resultados obtenidos con el código Java arriba propuesto por el autores de este trabajo se procesan los mismo datos con otra herramienta libre llamada la herramienta bayesiana Network tool java. Mayor información consultar en <http://bnj.sourceforge.net/>

#### ¿Qué es BNJ<sup>6</sup>?

BNJ<sup>6</sup> es un conjunto de herramientas de software open-source para investigación y desarrollo usando modelos gráficos de probabilidad. Está implementado en 100% puro Java y distribuido bajo la Licencia Pública General de GNU (GPL por sus siglas en inglés) por el Laboratorio para Descubrimiento de Conocimiento en Bases de Datos

(KDD por sus siglas en inglés) de Kansas State University .

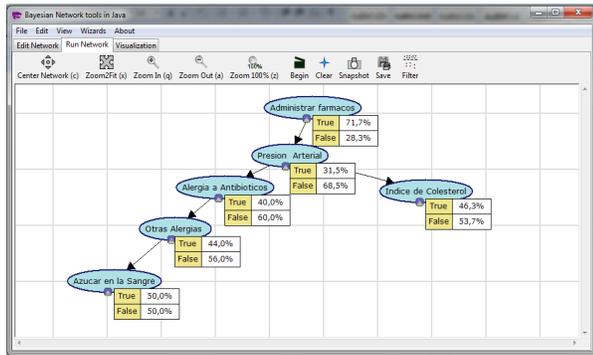


Imagen 1.0 Ventana Run Network  
Fuente: Bayesian a Network tool

Utilizando en algoritmo de propagación ilustrado en la sección 2.3 calculamos el estado inicial del árbol. Como los valores anteriores son los valores iniciales, ahora vamos a recalculamos los valores haciendo uso del algoritmo de propagación que calcula los lambda mensajes.

```

Public void CalcLamdaValor(nodoABB nodo, double
[]lamda_v, int flag) {
if(nodo != null) { if(flag==0){lamda_v[0] = 1;
lamda_v[1] = 1; flag++;
CalcLamdaValor(nodo.izq, lamda_v, flag);
CalcLamdaValor(nodo.der, lamda_v, flag);
}else{ lamda_v[0] *= nodo.lamda_msg[0];
lamda_v[1] *= nodo.lamda_msg[1];
CalcLamdaValor(nodo.izq, lamda_v, flag);
CalcLamdaValor(nodo.der, lamda_v, flag);
} } }

```

**Código 2** Calcula el lambda valor del nodo si este no ha sido instanciado.

```

public void lamdaMsg(nodoABB nodo) {
if(nodo.padre != null) {
nodo.lamda_msg[0] = nodo.matriz[0] * nodo.lamda_valor[0] +
nodo.matriz[2] * nodo.lamda_valor[1];
nodo.lamda_msg[1] = nodo.matriz[1] * nodo.lamda_valor[0] +
nodo.matriz[3] * nodo.lamda_valor[1]; } }

```

**Código 3.** Calcula el lamda-msg del nodo que le enviemos

```

public void piMsg(nodoABB nodo) {
if(!nodo.instanciada) { CalcPiMsg(nodo, nodo.pi_msg, 0);
}else{ nodo.pi_msg[0] = nodo.probabilidad[0];
nodo.pi_msg[1] = nodo.probabilidad[1]; }
}

```

**Código 4.** Calcula el pi\_msg del nodo.

```

public void CalcPiMsg(nodoABB nodo, double[] pmsg, int
flag) { if(nodo != null) { if(flag == 0) { flag++;
pmsg[0] = nodo.pi_valor[0];

```

```

pmsg[1] = nodo.pi_valor[1];
CalcPiMsg(nodo.izq, pmsg, flag);
CalcPiMsg(nodo.der, pmsg, flag);
}else if(flag == 1) { flag++;
CalcPiMsg(nodo.izq, pmsg, flag);
CalcPiMsg(nodo.der, pmsg, flag);
}else { pmsg[0] *= nodo.lamda_msg[0];
pmsg[1] *= nodo.lamda_msg[1];
CalcPiMsg(nodo.izq, pmsg, flag);
CalcPiMsg(nodo.der, pmsg, flag);
} } }

```

**Código 5.0** función recursiva que calcula el pi\_msg del nodo

```

public void preorden(nodoABB p) {
if(p != null) {
System.out.print ("nodo (" + p.name + "\nid=" + p.id + ")
{Matriz: {"");
for(int i=0; i<p.matriz.length; i++) {
System.out.print (p.matriz[i] + "|");
System.out.print("\n\tlamda_valor
{" + p.lamda_valor[0] + ", " + p.lamda_valor[1] + "});
System.out.print("\n\tpi_valor {" + p.pi_valor[0] + " ,
" + p.pi_valor[1] + "});
System.out.print("\n\tlamda_msg
{" + p.lamda_msg[0] + ", " + p.lamda_msg[1] + "});
System.out.print("\n\tpi_msg {" + p.pi_msg[0] + " ,
" + p.pi_msg[1] + "});\n");
System.out.print("\nprobabilidad: tverdad:" + p.probabilidad
[0] + " falso:" + p.probabilidad[1] + "\n");
preorden(p.izq);
preorden(p.der); } } }

```

**Código 6.0** Imprime el árbol. Recorrido en preorden

Función de control elimina marcas utilizadas en la recursión.

```

public void EliminarMarca(nodoABB p) {
if(p != null) { p.marca = false;
EliminarMarca(p.izq); EliminarMarca(p.der);
} }

```

**Código 7.0** Eliminar marcas

**Resultados del software**

Nodo ("Administrar Fármacos" id = 8) Matriz: {0.71 | 0.0 | 0.29 | 0.0 | }  
lamda\_valor {1.0, 1.0} pi\_valor {0.71, 0.29}  
lamda\_msg {1.0, 1.0} pi\_msg {0.71, 0.29}  
Probabilidad: **Verdad: 0.71 Falso: 0.29**

Nodo ("Presión Arterial" id = 6) {Matriz: {0.4 | 0.1 | 0.6 | 0.9 | }  
lamda\_valor {1.0, 1.0} pi\_valor {0.313, 0.687}  
lamda\_msg {1.0, 1.0} pi\_msg {0.313, 0.687}  
Probabilidad: **Verdad: 0.313 Falso: 0.687**

Nodo ("Alergia a Antibióticos" id = 5){Matriz: {0.4|0.4|0.6|0.6| }  
 lamda\_valor {1.0, 1.0} pi\_valor {0.4, 0.6}  
 lamda\_msg {1.0, 1.0} pi\_msg {0.4, 0.6}  
 Probabilidad: **Verdad: 0.4 Falso: 0.6**

Nodo ("Otras Alergia" id = 4){ Matriz: {0.5|0.4|0.5|0.6| }  
 }  
 lamda\_valor {1.0, 1.0} pi\_valor {0.44, 0.56}  
 lamda\_msg {1.0, 1.0} pi\_msg {0.44, 0.56}  
 Probabilidad: **Verdad: 0.44 Falso: 0.56**

Nodo ("Azúcar en la Sangre" id = 3){Matriz: {0.5|0.5|0.5|0.5| }  
 }  
 lamda\_valor {1.0, 1.0} pi\_valor {0.5, 0.5 }  
 lamda\_msg {1.0, 1.0} pi\_msg {0.5, 0.5}  
 Probabilidad: **Verdad: 0.5 Falso: 0.5**

Nodo ("Índice de Colesterol" id = 7){ Matriz: {0.6|0.4|0.4|0.6| } lamda\_valor {1.0,1.0} pi\_valor {0.4626, 0.5374 }  
 lamda\_msg {1.0, 1.0} pi\_msg {0.4626, 0.5374}  
 Probabilidad: **Verdad: 0.4626 Falso: 0.5374**

Observe que la probabilidad inicial de administrar fármacos a un paciente es: Probabilidad: **Verdad: 0.71 Falso: 0.29**. Esto quiere decir que existe 71% de probabilidad de administrar fármacos a un paciente con síntomas de enfermedad cardiovascular, de acuerdo a los datos arrojados por el software implementado con el modelo probabilístico (redes bayesianas) utilizando las siguientes variables de decisión (presión arterial, índice de colesterol, azúcar en la sangre, alergias a antibióticos y otras alergias).

Pero en el caso que se tenga una nueva evidencia del cambio de colesterol de un paciente las siguientes probabilidades arriba relacionadas:

Probabilidad **Verdad: 0.4626 Falso: 0.5374** a cambiar a Probabilidad **Verdad: 1.0 Falso: 0.0**, como se puede observar en la siguiente pagina en el nodo de índice de colesterol:

Nodo ("Índice de Colesterol" id = 7){  
 Matriz: {0.6|0.4|0.4|0.6| }  
 lamda\_valor {1.0, 0.0} pi\_valor {0.46, 0.54}  
 lamda\_msg {0.6, 0.4} pi\_msg {1.0, 0.0}  
**Probabilidad: Verdad: 1.0 Falso: 0.0**

Entonces, tenemos que el paciente presenta colesterol alto. En el software el  $\lambda$ -valor tomaría un valor de uno.

lamda\_valor {1.0, 0.0} pi\_valor {0.46, 0.5374}

Ahora se procede a la actualización de la red pasando por cada uno de los nodos a través de los mensajes:  $\lambda$ -mensaje y  $\pi$ -mensaje.

Después de haber actualizado cada uno de los nodos (probabilidades) a través de los  $\lambda$ -valor y  $\pi$ -valor, obtenemos como resultado los siguientes valores de cada uno de los nodos:

Nodo ("Administrar Fármacos" id = 8) {Matriz: {0.71|0.0|0.29|0.0| }  
 lamda\_valor {0.63, 0.168} pi\_valor {0.71, 0.29}  
 lamda\_msg {1.0, 1.0} pi\_msg {0.428, 0.11}  
**Probabilidad: Verdad: 0.62 Falso: 0.48**

Nodo ("Presión Arterial" id = 6){ Matriz: {0.4|0.1|0.6|0.9| }  
 lamda\_valor {0.6, 0.4} pi\_valor {0.31, 0.69}  
 lamda\_msg {0.48, 0.420000000004} pi\_msg {0.313, 0.687}  
 Probabilidad: **Verdad: 0.41 Falso: 0.59**

Nodo ("Alergia a antibióticos" id = 5){ Matriz: {0.4|0.4|0.6|0.6| }  
 lamda\_valor {1.0, 1.0} pi\_valor {0.22, 0.33}  
 lamda\_msg {1.0, 1.0} pi\_msg {0.22, 0.33}  
 Probabilidad: **Verdad: 0.22 Falso: 0.33**

Nodo ("Otras Alergia" id = 4){ Matriz: {0.5|0.4|0.5|0.6| }  
 lamda\_valor {1.0, 1.0} pi\_valor {0.24, 0.36}  
 lamda\_msg {1.0, 1.0} pi\_msg {0.24, 0.36}  
 Probabilidad: **Verdad: 0.24 Falso: 0.36**

Nodo ("Azúcar en la Sangre" id = 3){Matriz: {0.5|0.5|0.5|0.5| }  
 lamda\_valor {1.0, 1.0} pi\_valor {0.271, 0.271}  
 lamda\_msg {1.0, 1.0} pi\_msg {0.271, 0.271}  
 Probabilidad: **Verdad: 0.271 Falso: 0.271**

Nodo ("Índice de Colesterol" id = 7){Matriz: {0.6|0.4|0.4|0.6| }  
 lamda\_valor {1.0, 0.0} pi\_valor {0.46, 0.54}  
 lamda\_msg {0.6, 0.4} pi\_msg {1.0, 0.0}  
**Probabilidad: Verdad: 1.0 Falso: 0.0**

Observe que antes, la probabilidad inicial de administrar fármacos a un paciente era de: Probabilidad: **Verdad: 0.71 Falso: 0.29**, como se observa en el nodo inicial:

Nodo ("Administrar Fármacos" id = 8){Matriz: {0.71|0.0|0.29|0.0| }  
 lamda\_valor {1.0, 1.0} pi\_valor {0.71, 0.29}  
 lamda\_msg {1.0, 1.0} pi\_msg {0.71, 0.29}  
 Probabilidad: **Verdad: 0.71 Falso: 0.29**

Así las probabilidades de los nodos de la red cambiaron. La posibilidad de administrar fármacos cambio de: Probabilidad: **Verdad: 0.71 Falso: 0.29** a Probabilidad: **Verdad: 0.62 Falso: 0.48**. Es decir disminuye en un 9%.

Y la de presión arterial que cambió así:

Probabilidad: **Verdad: 0.313** Falso: 0.687 a

Probabilidad: **Verdad: 0.41** Falso: 0.59, aumentó en un en un 10 %.

Como se observa la probabilidad de administrar fármacos aun siendo baja (0.62), permite administrar fármacos (según criterio medico se puede administrar fármacos cuando se tiene una probabilidad de más del .50, si es más baja la decisión depende de otros criterios médicos).

De igual forma se puede considerar nuevas evidencias para cada uno de los nodos de la red ya que el algoritmo al recorrer todos los nodos hacia abajo y hacia arriba permite la actualización de la red.

### III. CONCLUSIONES

- Se demostró empíricamente que a partir de una estructura de red es posible diagnosticar ó predecir la administración ó no de fármacos en pacientes con síntomas de enfermedad cardiovascular, usando variables tales como: presión arterial, índice de colesterol, azúcar en la sangre, alergias a antibióticos y otras alergias y mediante la utilización del modelo probabilístico de las redes Bayesianas.

-El algoritmo de propagación implementado genera resultados coherente con otras herramientas (por ejemplo, Bayesian Network tool).

- Como se observa en la implementación el manejo de arboles y la recursividad tiene un alto costo computacional ya que el consumo de memoria y procesamiento aumenta con un numero de variables de decisión.

### REFERENCIAS

- [1]. [Bayesian Artificial Intelligence. Kevin B. Korb, Ann Nicholson. Boca Raton, Fla: CRC Press, 2011. Jensen,
- [2]. [Ezequiel Felgaer], p. (2008, abril). Optimización de redes bayesianas basado en técnicas de aprendizaje por inducción disponible en: [www.fi.uba.ar/materias/7500/felgaer-tesisingenieriainformatica.pdf](http://www.fi.uba.ar/materias/7500/felgaer-tesisingenieriainformatica.pdf)
- [3]. Frías Blanco Isvani; Ortiz Díaz Agustín. Clasificadores y Multiclasificadores con cambio de concepto basados en Árboles De Decisión. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, Vol. 14, Núm. 45, 2010, pp. 32-43 Asociación Española para la Inteligencia

Artificial

España

<http://redalyc.uaemex/src/inicio/ArtPdfed.jsp?iCve=>

- [4]. [4] [Herkovits y Copper] Herskovits, E. H. Computer-based probabilistic network construction. Doctoral Dissertation. Medical Information Sciences, Stanford University, Stanford, CA, 1991.
- [5]. Joseph Bockhorst, Mark Craven, David Page, Jude Shavlik, Jeremy Glasner. "A Bayesian network approach to operation prediction" Bioinformatics Vol. 19 no. 10 2005, pages 1227-1235.
- [7]. [Miguel ángel Rueda Morales] Modelos de recomendación basado en redes bayesianas. Tesis de doctorado en informática Granada Septiembre 2011. <http://digibug.ugr.es/bitstream/10481/20245/1/20184979.pdf>
- [8]. [Pearl, 1988] J. Pearl. Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley, 1984