

Desarrollo de modelos de fallas de de sistemas Electrónicos utilizando redes bayesianas

Dr. Ing. Jose Luis Roca

Departamento de Ingeniería e Investigaciones Tecnológicas
Universidad Nacional de La Matanza
roca.joseluis@gmail.com

Abstract — The scope of the present work is focused in the use of Belief Bayesian Nets (BBN) in order to model complex electronic system failures with hardware and built-in software. The theory of Bayesian networks can be thought as a fusion of influence diagrams and Bayes Theorem. The present analysis emphasizes their use in replacement of the conventional Fault Tree Analysis (FTA). A later study of software necessary to implement their application completes the proposed objective.

Keywords-component; Belief Bayesian Nets (BBN), Fault Tree Analysis (FTA); Conditional Probability Table (CPT); Influence diagrams (IF).

I. INTRODUCCION

Los sistemas electrónicos en la actualidad están constituidos tanto por hardware como por software delimitando con el ser humano a través de una interfase que permite el intercambio de información (handshake) entre usuario u operador del sistema y el propio sistema. Estos sistemas son demasiado complejos y su complejidad es inherente a su estructura básica. En líneas generales para analizar las probabilidades de fallas de un sistema complejo de este tipo con tiempos de misión fijos, se recurre o a diagramas de confiabilidad RBD (Reliability Block Diagrams) o a los denominados árboles de falla FTA (Fault Tree Analysis). Si se pretende analizar la variación de la confiabilidad de un sistema en el tiempo la técnica de Cadenas de Markov (Markov Chains) es la solución. En el primer caso la introducción de un componente nuevo cuya falla incide directamente sobre algunos de los otros componentes de la red lógica es imposible de modelar con las técnicas RBD o FTA. En el segundo caso la cantidad de estados para sistemas de alta complejidad sería enorme y difícil de manejar aun con medios computacionales. Aquí aparece el beneficio de la modelización de estos sistemas complejos mediante las denominadas Redes de Probabilidad Bayesiana BBN (Belief Bayesian Nets). La utilización de esta última técnica sumada a la selección de una herramienta adecuada que permita resolver esta situación constituye el objetivo de este trabajo de investigación aplicada.

II. PROBLEMÁTICA A RESOLVER

La problemática básica a resolver esta planteada desde dos puntos de vista: el primero es como resolver el modelo de

fallas en un FTA cuando se trata de incorporar un componente nuevo ya sea de software o hardware cuya falla incide directamente en algún otro componente y el segundo estudiar y analizar los diferentes software aplicativos que resuelvan computacionalmente la modelización realizada.

Fundamentalmente la utilización de BBN para la modelización de fallas en un sistema electrónico complejo esta basada en que las mismas soportan estructura gráfica y representación modular del conocimiento, utilizan algoritmos locales y distribuidos para inferencia y aprendizaje, permiten la interpretación intuitiva (causalidades posibles) y contemplan la representación factoreada. Además contienen menos parámetros que la función de distribución conjunta, poseen menor complejidad muestral (menor cantidad de datos para aprendizaje) y menor complejidad temporal (menos tiempo para hacer inferencia estadística). En cuanto a inferencia estadística se refiere, las BBN permiten evaluar probabilidades “a posteriori”, esto es el computo de las probabilidades de cualquier evento dada la evidencia correspondiente, aportan la explicación más verosímil respecto a los posibles escenarios y las evidencias presentes, permite tomar decisiones racionales, maximizando la utilidad esperada y el valor de la información, asimismo tienen en cuenta el efecto y análisis de causalidades. Desde el punto de vista cualitativo su representación es a través de un grafo directo acíclico, cuyos nodos representan variables aleatorias y los arcos distendidos entre los nodos la influencia directa de una variable sobre otra. Desde el punto de vista cuantitativo se trata de la representación de un conjunto de distribuciones de probabilidad condicional. Juntas definen una única distribución en forma factoreada. El modelo probabilístico es descripto cualitativamente por un grafo acíclico como se observa en la Fig.1.

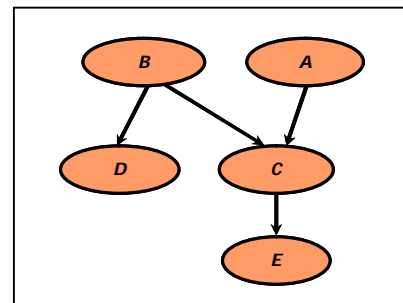


Fig.1 – Ejemplo de BBN

Los vértices del grafo que representan las variables aleatorias son llamados nodos. Estos nodos están representados por círculos que contienen el nombre de la variable aleatoria. Las conexiones entre los nodos están representadas por arcos dirigidos o flechas, y estas flechas representan dependencia entre las variables. Para cualquier par de nodos, aquel en donde se origina la flecha se denomina padre del nodo al cual esta llega, y esta ultima recibe el nombre de hijo de este nodo padre. La independencia es asumida en redes bayesianas por la ausencia de conectores. Por lo tanto, los nodos donde no llega ninguna flecha se dice que no tienen ningún padre y son independientes.

Los nodos donde se originan las flechas son llamados padres. Mientras que los nodos donde las flechas llegan se denominan hijos. Por ejemplo, A es padre de C, entonces C es hijo de A. de una forma más general los nodos que pueden ser alcanzados por otro u otros nodos son denominados descendencia. Por ejemplo, el nodo E es descendiente de A y B. de la misma manera los nodos que encabezan el camino para llegar a otro nodo son denominados ancestros. Por ejemplo, el nodo C es un ancestro de E.

Un aspecto importante en redes bayesianas (BBN) es que no existen lazos. Con esto se quiere decir que ningún hijo puede ser su propio ancestro o descendencia. El punto principal de las redes bayesianas es producir una conclusión basada en la probabilidad de ocurrencia de cada variable. Esto significa que la probabilidad de cada nodo en las redes bayesianas puede ser calculada cuando las probabilidades asociadas a las otras variables son conocidas.

La distribución de probabilidad conjunta esta definida como la probabilidad que tiene una serie de eventos de ocurrir simultáneamente. La probabilidad conjunta de varias variables puede calcularse por el producto de probabilidades individuales de cada nodo. Para el ejemplo de la Fig.1 se tiene:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P[X_i / \text{padres}(X_i)] \quad (1)$$

Aplicando esta expresión al grafo acíclico introducido en la Fig.1, la distribución de probabilidad resulta:

$$P(A, B, C, D, E) = P(A).P(B).P(C/A, B).P(D/B).P(E/C) \quad (2)$$

Si un nodo no tiene ningún nodo padre, como en los nodos A y B, su distribución de probabilidad no está afectada por el resto de los nodos y su distribución es incondicional. Este no es el caso de nodos en los que la distribución de probabilidad es condicional y está afectada por los demás nodos, como es el caso de los nodos C, D y E.

Tres son las conexiones entre nodos que pueden darse definiendo evidencias fuertes y débiles, conexión serie, divergente y convergente. Evidencia fuerte para un nodo X es la evidencia de que el valor de probabilidad asignado al nodo X es en definitiva un valor determinado y particular. Evidencia débil para un nodo X es cualquier evidencia que permita la actualización del valor de probabilidad a priori asignado para el nodo X.

En una conexión serie, Fig.2, cualquier evidencia al comienzo de la conexión puede propagarse a lo largo del camino directo si y solo si no existe ningún nodo intermedio con evidencia fuerte que bloquee el camino directo. En otras palabras la evidencia no puede transmitirse de A a C si B bloquea el camino directo. Se dice que A y C son “d-separados dado B”.

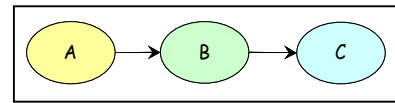


Fig.2 – Conexión Serie

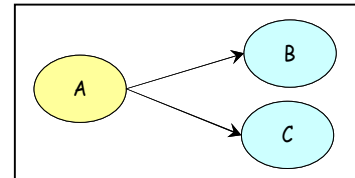


Fig.3 – Conexión Divergente

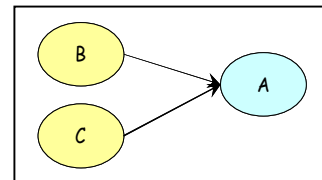


Fig.4 – Conexión Convergente

En una conexión divergente, Fig.3, la evidencia puede propagarse desde un nodo Hijo B a otro nodo Hijo C a través de un nodo Padre A a menos que el nodo padre presente evidencia fuerte. B y C son condicionalmente independientes dado A. Se dice que B y C son “d-separados dado A”. En una conexión convergente, Fig.4, la evidencia puede propagarse desde dos nodos Padres B y C a un nodo Hijo A cuando el nodo hijo recibe alguna evidencia ya sea fuerte como débil.

Cualquier evidencia sobre B o C puede propagarse hacia A. Se dice que B y C son condicionalmente dependientes dado A.

Cada nodo tiene asociada una tabla de probabilidades condicionales CPT (conditional probability table). Estas probabilidades condicionales representan en efecto probabilidades basados en evidencia anterior o posterior.

Esto es que, por cada nodo padre y por cada posible estado de ese nodo padre, existe una columna en la CPT que describe la probabilidad de que el nodo hijo se encuentre en algún estado determinado.

En la Fig.5 se observa una CPT para un nodo padre y en la Fig.6 una CPT para un nodo hijo como ejemplos.

Evento A		
Estado 1	Estado 2	Estado 3
0,800	0,150	0,050

Fig.5 – CPT nodo Padre

Padre	Hijo	
	Evento B	
Evento A	Estado 1	Estado 2
Estado 1	0,050	0,950
Estado 2	0,100	0,900
Estado 3	0,700	0,300

Fig.6 – CPT nodo Hijo

La probabilidad “a posteriori” de que un evento se encuentre en un estado determinado posible esta basada en la adición de alguna evidencia objetiva actual de esa situación. La probabilidad a “priori” resulta un caso particular de la probabilidad basada en información anterior y esta firmemente determinada por la información almacenada en las CPTs de la BBN.

Siendo un BBN un modelo completo para las distintas variables involucradas y sus relaciones, esta puede ser utilizada para realizar inferencia estadística, esto es el computo de la funciones de distribución de probabilidad “a posteriori” a partir de la incorporación de evidencia. Una BBN puede ser así considerada como un mecanismo de aplicación automática del Teorema de Bayes a sistemas complejos.

Las metodologías más generales de realizar inferencia estadística son por eliminación de variables, propagación en árbol y condicionamiento recursivo. La primera elimina ya sea por integración o suma las variables de interés una por una distribuyendo la suma sobre el producto, la segunda metodología actúa sobre varias variables al mismo tiempo de modo que la evidencia se propaga rápidamente a través del grafo generado, agilizando su tratamiento computacional.

Finalmente el condicionamiento recursivo permite una compensación del espacio-tiempo y balancea de algún modo la eficacia de la metodología de eliminación de variables cuando se utiliza bastante espacio de memoria computacional.

III. AVANCES Y RESULTADOS OBTENIDOS

Para la modelización de un sistema electrónico complejo la herramienta que se ha utilizado en esta investigación el es Árbol de Fallas FTA (Fault Tree Analysis) útil para sistemas de tiempo de misión fijo, como es el caso en que se quiere investigar la probabilidad de que se de un evento denominado tope (TOP), generalmente la falla del sistema para un determinado y fijo periodo de tiempo, dadas las probabilidades de falla de sus componentes en el mismo periodo de tiempo. Los árboles de falla están constituidos por compuertas lógicas que enlazan eventos.

En la Fig.7 se observa el diagrama de bloques de un sistema electrónico compuesto por dos procesadores P_1 y P_2 , tres bancos de memoria, dos locales M_1 y M_2 y uno compartido M_3 y un bus N compartido.

Existen además dos unidades de disco duro D_1 y D_2 conformadas cada una de ellas por un disco duro principal D_{11} y D_{21} y un disco duro espejado D_{12} y D_{22} respectivamente.

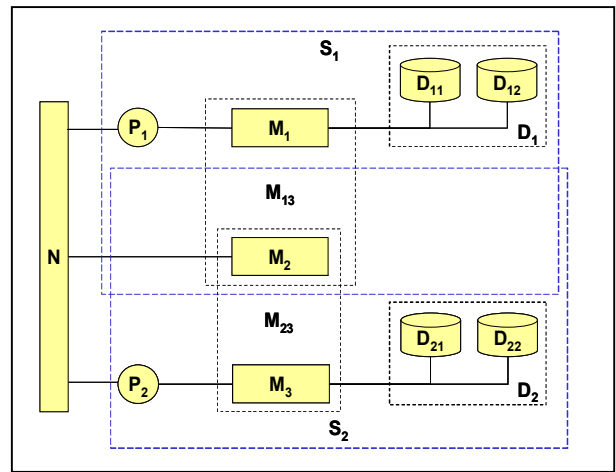


Fig.7 – Esquema de un sistema electrónico

Desde el punto de vista de sistemas de procesamiento se puede observar que son dos. El S_1 formado por P_1, M_1, D_1 y el S_2 formado por P_2, M_2, D_2 , compartiendo ambos el banco de memoria M_3 y el Bus N.

Si se quiere analizar el evento falla del sistema, sea este el evento tope TE, para un tiempo dado fijo, el árbol de fallas correspondiente se muestra en la Fig.8. Obsérvese que sucede cuando se pretende incluir un nuevo evento básico PS como puede ser el asociado a la falla de una fuente de alimentación en forma dependiente que induce fallas en ambos procesadores y en el sistema en general.

Este nuevo evento afecta ambos procesadores y el sistema solamente pero la única forma de introducirlo en el árbol de fallas es mediante el artilugio descrito en líneas de puntos en el árbol de fallas y que no representa la realidad. La conclusión es que resulta imposible modelar este evento en el árbol de fallas.

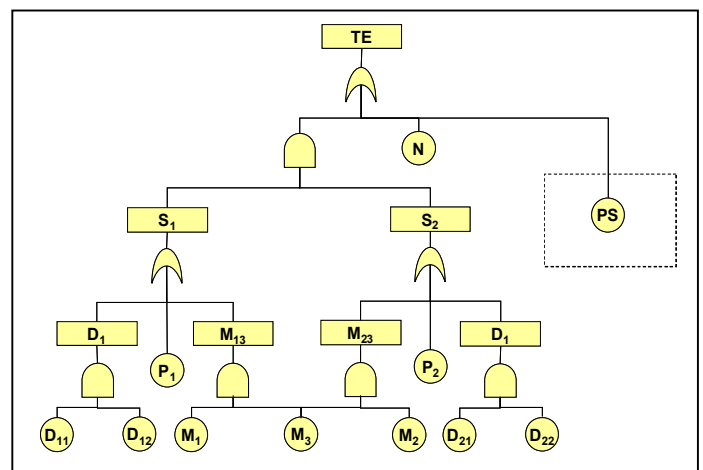


Fig.8 – Árbol de fallas correspondiente al sistema electrónico de la Fig.7

A los efectos de modelar estos eventos y de representar fehacientemente lo que sucede respecto a la interrelación de las distintas fallas en los sistemas electrónicos es que se

procedió a modelar los distintos eventos mediante BBN. En principio se armaron las CPTs correspondientes a las distintas compuertas lógicas presentes en un árbol de fallas. Estas CPTs resultan todas determinísticas.

En las Fig. 9, 10 y 11 se muestran las compuertas típicas de un árbol de fallas, a saber "O", "Y" y mayoritaria "2 de 3" junto con su equivalente diagrama de influencia y su respectiva CPT.

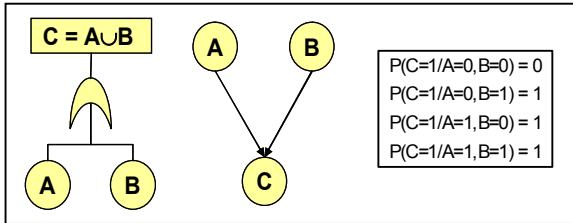


Fig.9 – Compuerta "O" – Diagrama de Influencia – CPT

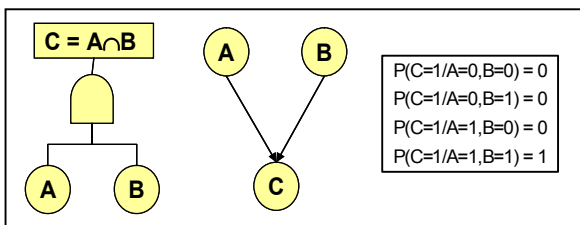


Fig.10 – Compuerta "Y" – Diagrama de Influencia – CPT

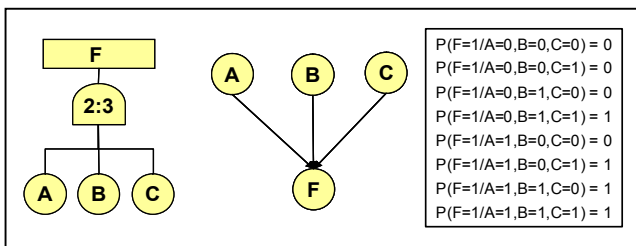


Fig.11 – Compuerta "2 de 3" – Diagrama de Influencia – CPT

En la Fig.12 se observa el modelado de fallas del sistema electrónico de la Fig.7 utilizando BBN. En el diagrama de influencia aparecen todos los eventos raíz que corresponden a los eventos básicos del árbol de fallas, esto es P_1 , P_2 , M_1 , M_2 , M_3 , N , D_{11} , D_{21} , D_{12} y D_{22} respectivamente. Estos eventos raíz son netamente estocásticos. Los nodos determinísticos son los que corresponden a las distintas compuertas lógicas, esto es D_1 , D_2 , M_{13} , M_{23} , S_1 , S_2 , S_{12} y TE .

Obsérvese como es posible modelar el evento correspondiente a la probabilidad de falla de la fuente de alimentación PS cuya falla incide directamente sobre los eventos correspondientes a la probabilidad de falla de los procesadores y del sistema.

En la Fig.13 se observan las CPT correspondientes a los nodos determinísticos S_{12} y TE . En las mismas CPT se indican con

un signo menos sobre el evento correspondiente la negación del mismo, es decir su no ocurrencia.

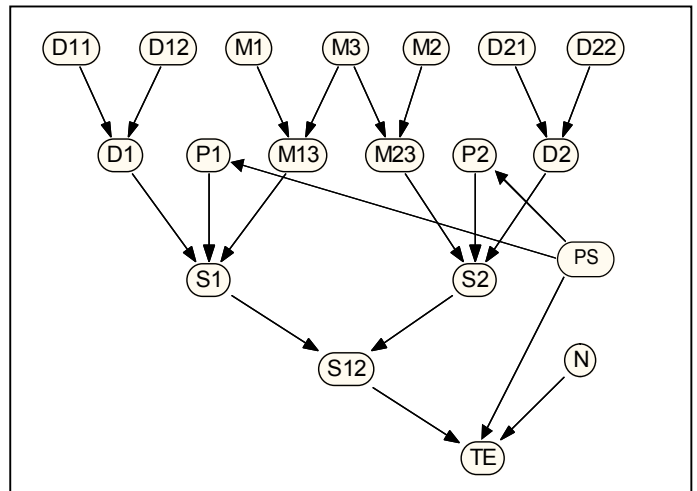


Fig. 12 – BBN correspondiente al sistema electrónico de la Fig.13.

$P(S_{12}/S_1, \overline{S_2})=1$	$P(TE/S_{12}, \overline{N})=1$
$P(S_{12}/S_1, S_2)=0$	$P(TE/S_{12}, N)=1$
$P(S_{12}/\overline{S_1}, \overline{S_2})=0$	$P(TE/S_{12}, N)=1$
$P(S_{12}/\overline{S_1}, S_2)=0$	$P(TE/S_{12}, \overline{N})=0$

Fig.13 – CPTs correspondientes a los nodos S_{12} y TE

El avance obtenido utilizando modelos de fallas en sistemas electrónicos vía BBN permite lograr mejores aproximaciones y solucionar problemas de inserción e interrelación de fallas de los distintos componentes que con las técnicas de árbol de falla (FTA) o diagramas de confiabilidad (RBD) era imposible. Los resultados obtenidos demuestran la versatilidad de esta técnica.

IV. ESTUDIO Y ANÁLISIS DE SOFTWARE

El estudio y análisis de los diversos paquetes de software que permiten modelar y resolver sistemas vía BBN es considerable y de diversa índole. Algunos admiten una interfase grafica, otros no. En algunos casos puede ejecutar inferencia estadística. En la clasificación es posible también encontrar paquetes de software que resuelven redes bayesianas estáticas y/o dinámicas, ampliando el espectro de posibilidades en el mercado.

Los códigos son también variados, desde C++ a Java, pasando por LISP y Matlab. Los sistemas operativos que soportan estas aplicaciones se mueven en toda la franja desde WINDOWS a UNIX pasando por OS de MAC.

En cuanto a costos se encuentran aquellos paquetes de software de uso libre, otros de costo menor para uso

académico, con ciertas restricciones y otros de uso irrestricto y de costo mas elevado.

Para realizar un estudio profundo y exhaustivo de los diversos paquetes de software disponibles es necesario clasificar los mismos de acuerdo a sus atributos.

Los atributos o cualidades que se han seleccionados a esos efectos se refieren a código fuente, interfase de la aplicación, entorno y sistema operativo, interfase grafica, soporte de nodos continuos o discretos, aprendizaje vía parametrización y/o estructura, costos y posibilidad de inferencia estadística.

A continuación se expone un resumen de lo analizado para cada uno de los paquetes de software disponible en el mercado internacional, definiendo las cualidades utilizadas en el análisis. En el cuadro del Apéndice I se exponen en detalle las definiciones de la cualidades utilizadas a lo efectos de su posterior utilización en este estudio

- Cfi = Código fuente incluido (N=no) Si el código fuente esta incluido, en que lenguaje.
- API = Interfase de aplicativo incluida (N significa que el programa no puede ser integrado en el código, esto es solo puede ser corrido en forma aislada.
- Exec = El ejecutable corre en W = Windows (95/98/NT), U = Unix, M = Mac, - = cualquier maquina con compilador.
- Cts = Soporta nodos latentes continuos. G = (condicionalmente) Soporte analítico de nodos Gaussianos, Cs = Soporte de nodos continuos por muestreo, Cd = Soporte de nodos continuos por discretización, Cx = Soporte de nodos continuos por alguna metodología no especificada, D = Soporte solo de nodos discretos.
- GUI = Interfase grafica incluida.
- Params = Aprende vía parametrización.
- Struct = Aprende vía estructura. CI = utiliza pruebas de independencia condicional.
- Utilit = Nodos utilitarios y de decisión (Ejemplo: Soporta diagramas de influencia.
- Costo. 0 = sin costos (disponible solo para uso académico). \$ = Software comercial (algunos con versiones sin costo pero con algunas restricciones de uso). Ejemplo: Limitado numero de nodos, modelos de BBN imposibles de grabar o no existe interfase de aplicativo.
- Undir. Tipos de grafico soportados. U = Solamente grafos no orientados, D = Solamente grafos orientados, UD = Ambos tipos de grafos orientados y no orientados, CG = Grafos encadenados (mezcla de orientados y no orientados)

- Inferencia estadística = Algoritmo utilizado. jtree = junction tree, varelim = eliminación de variables (bucket), MH = Metropolis Hastings, G = Gibbs sampling, IS = importance sampling, sampling = Algún otro método de Monte Carlo, polytree = Algoritmo de Pearl restringido a grafos acíclicos, no = no soporta inferencia (el programa es diseñado solo para aprendizaje de estructura a partir de un conjunto completo de datos observados).

- Comentarios.

V. SELECCIÓN DEL SOFTWARE

Dentro de todos los paquetes de software analizados se escogió el paquete de software NETICA [10]. Se trata de un programa potente, fácil de utilizar, completo para trabajar con redes de probabilidad bayesiana (BBN) y diagramas de influencia.

Posee una interfase con el usuario de características muy intuitivas y de fácil manejo en cuanto a gráficos se refiere. Esto permite dibujar BBN, relacionar las variables utilizadas e incorporar probabilidades individuales bajo la forma de ecuaciones. Una vez creada la red, el conocimiento contenido en ella puede ser transferido a otras redes cortando y pegando o guardando en forma modular de modo de crear una librería de nodos acoplados en forma discontinua.

Las redes y las librerías de nodos pueden ser guardadas o impresas. Netica puede utilizar las redes para realizar varias clases de inferencia estadística usando los algoritmos más rápidos y más modernos. El programa encuentra los valores de probabilidades apropiados para todas las variables desconocidas, en el caso de un nuevo problema a resolver del que se tiene conocimiento limitado.

Estos valores de probabilidades pueden ser mostrados de varias maneras diferentes, incluyendo gráficos de barras y del tipo de medidores. El problema en cuestión puede ser guardado como archivo y puede ser traído nuevamente en la misma red creada u otra nueva, de modo de poder permitir un posterior análisis o tomar en cuenta la nueva información sobre el problema.

Puede utilizar diagramas de influencia para encontrar las decisiones óptimas que maximizan los valores previstos de las variables especificadas. Puesto que las decisiones a futuro pueden depender de las observaciones que se harán y las sincronizaciones y las correlaciones entre las decisiones deben ser consideradas es que Netica puede construir redes condicionales.

Netica puede ser utilizado para transformar una red de varias maneras. Las variables que ya no son de interés pueden ser eliminadas sin cambiar las relaciones preexistentes entre las variables restantes.

Los modelos de probabilidad pueden ser explorados mediante operaciones tales como inversión de acoplamientos individuales de la red, quitando o agregando las influencias

causales, optimizando una decisión en el tiempo, etc. Estas operaciones se pueden hacer simplemente a través de pantalla y Mouse lo que hace de Netica una herramienta muy conveniente para la exploración fácil, y para aprender rápidamente los conceptos de diagrama de la red bayesiana y de influencia. Hay muchas nuevas características y capacidades para Netica actualmente en el desarrollo.

El costo del programa completo esta alrededor de u\$s 685 en su versión comercial y u\$s 285 en su versión académica. Se recomienda vivamente la adquisición de la versión académica. Existe una versión gratis pero limitada en funciones y cantidad de variables.

VI. ESPECIFICACION DEL SOFTWARE SELECCIONADO

- Genera gráficos de calidad que pueden ser incorporados en otros documentos.
- Compila BBN en un árbol acoplado de modo de razonamiento probabilístico rápido.
- Ayuda extensiva en pantalla
- Análisis de sensibilidad Utilidad-libre (versión de Windows solamente)
- Puede probar el funcionamiento de una BBN usando archivos de casos, arrojando como resultado una matriz con las tasas de errores logarítmicos y cuadráticos, una tabla de calibración y los índices correspondientes a cada nodo bajo estudio.
- Permite encontrar las decisiones óptimas para los problemas de decisión secuenciales
- Permite resolver diagramas de influencia eficientemente usando árboles acoplados
- Aprende relaciones probabilísticas entre datos
- Provee una fácil edición de gráficos de BBN y de diagramas de la influencia, incluyendo:
- Corte/pegado/duplicado de nodos sin dejar de lado su relación probabilística.
- Varias formas de exhibición de las variables asociadas a los nodos (gráficos de barra, medidores, etc.)
- Acoplamientos curvos entre nodos a los fines de construir diagramas complejos de forma ordenada.
- Comentarios, sin perder de vista al autor de los mismos.
- Niveles ilimitados de deshacer/hacer de nuevo.
- Permite la entrada de relaciones probabilísticas vía ecuaciones con una extensa biblioteca de funciones de probabilidad y de otras funciones matemáticas incorporada.
- Tiene facilidades para la fácil discretización de variables continuas.
- Puede invertir acoplamientos y sumar nodos de diagramas de influencia o BBN de modo de poder explorar el modelo de red.
- Soporta desconectar acoplamientos entre nodos de modo de hacer posible la existencia de librerías de relaciones de probabilísticas.
- Es posible representar redes con los nodos cuyos valores cambian con el tiempo y tener acoplamientos con demoras. El software puede convertir automáticamente

estas redes en redes regulares ampliadas que cubren un periodo de tiempo limitado.

- Acepta estimaciones de máxima verosimilitud (es decir, evidencia virtual), y resultados de la forma que una cierta variable no esta en un cierto estado.
- Puede exhibir nodos y resultados de inferencia estadística de varias formas, incluyendo gráficos de barra y de un medidor verdadero/falso.
- Fácil customización de pantalla de modo de ser útil al usuario.
- Soporta documentación y trazabilidad para todos y cada uno de los nodos de la red.
- Documentación de las ayudas y seguimiento de cada nodo y red (con los comentarios, títulos, autor, cuando haber cambiado, etc. pasados)
- No tiene límites, en cuanto al tamaño o la complejidad de redes, esta solo limitado por la memoria disponible.
- Puede trabajar con el producto de Netica API, por ejemplo, compartiendo los mismos archivos.

VII. BIBLIOGRAFIA

- [1] David Heckerman, "A tutorial on learning with Bayesian Networks", Technical Report MSR-TR-95-06; Microsoft Research; Advanced Technology Division; Microsoft Corporation; One Microsoft Way; Redmond, WA 98052; USA.
- [2] Judea Pearl, "Causality: Models, Reasoning and Inference", Econometric Theory, Vol. 19, Cambridge University Press, USA, pp. 675-685, 2003.
- [3] Charniak Eugene, "Bayesian Networks without tears", AI Magazine, USA, pp.50-63; Winter 1991.
- [4] Robert Cowell, A.Philip Dawid, Steffen Lauritzen & David Spiegelhalter, "Probabilistic Networks and Expert Systems", Springer Verlag, New York, Inc., pp.5-61; 1999.
- [5] Michael Irwin Jordan, "Learning in Graphical Models", MIT Press, Kluwer Academic Publishers, pp.27-105, 1998.
- [6] Wray Buntine, "Operations for learning with graphical models", Journal of Artificial Intelligence Research, Vol.2, pp.159-225, 1994.
- [7] Peter Spirtes, Clark Glymour & Richard Scheines, "Causation, Prediction and Search", Cambridge, Massachusetts, MIT Press, pp.1-101, 2001.
- [8] M. Neil and N.E. Fenton, "Predicting Software Quality Using Bayesian Belief Networks," *Proc 21st Ann. Software Eng. Workshop*, NASA Goddard Space Flight Centre, pp. 217-230, Dec. 1996.
- [9] M. Neil, B. Littlewood, and N. Fenton, "Applying Bayesian Belief Networks to Systems Dependability Assessment," *Proc. Safety Critical Systems Club Symp.*, Springer-Verlag, Leeds, Feb. 1996.
- [10] Norsys Software Corp. 3512 West 23rd Avenue, Vancouver, BC, CANADA (www.norsys.com)