

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

Director: Ing. Aldo Sacerdoti (asacerdo@unlam.edu.ar)

Integrantes: Dr. Ing. José Luis Roca, Ing. Ricardo Julia, Ing. Ariel Serra

Introducción

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

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.

Para el cómputo y planificación de una BBN existen en la actualidad una diversidad de paquetes de software que encaran ese problema. Parte final de esta investigación la constituye el estudio de los mismos a los fines de seleccionar alguno de ellos que pueda ser adquirido a un precio adecuado y con el cual pueda empezarse a trabaja a futuro en la utilización de esta herramienta en el campo de estudio y análisis de fallas de sistemas electrónicos complejos.

Estudio y Análisis de Paquetes de Software

El estudio y análisis de los diversos paquetes de software que permiten modelizar 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 que sigue 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.

Nombre	Autores	Cfi	API	Exec	Cts	GUI	Params	Struct	Utilit	Costo	Undir	Inferencia	Comentarios
AgenaRisk	Agena	N	S	W,U	Cx	S	S	N	N	\$	D	JTree	Simulación vía discretización dinámica
Analytica	Lumina	N	S	W,M	G	S	N	N	S	\$	D	sampling	Compatible con planilla de cálculo
Banjo	Hartemink	Java	S	W,U,M	Cd	N	N	S	N	0	D	no	Aprendizaje vía estructura de redes estáticas o dinámicas de variable discreta
Bassist	U. Helsinki	C++	S	U	G	N	S	N	N	0	D	MH	Generación de C++ para MCMC.
Bayda	U. Helsinki	Java	S	WUM	G	S	S	N	N	0	D	?	Bayesiano Nativo Clasifica Bayes
BayesBuilder	Nijman (U. Nijmegen)	N	N	W	D	S	N	N	N	0	D	?	-
BayesiaLab	Bayesia Ltd	N	N	-	Cd	S	S	S	N	\$	CG	jtree,G	Aprendizaje vía estructura, interrogación adaptiva, modelos dinámicos
Bayesware Discoverer	Bayesware	N	N	WUM	Cd	S	S	S	N	\$	D	?	Aprendizaje vía colapso y limitación con datos parciales
B-course	U. Helsinki	N	N	WUM	Cd	S	S	S	N	0	D	?	Corre en su Server: Vista de resultados vistos vía web browser.
Belief net power constructor	Cheng (U.Alberta)	N	W	W	D	S	S	CI	N	0	D	?	-

BNT	Murphy (U.C.Berkeley)	Matlab/C	S	WUM	G	N	S	S	S	0	D,U	Varios	Gestiona modelos dinámicos como HMMs y filtros de Kalman
BNJ	Hsu (Kansas)	Java	-	-	D	S	N	S	N	0	D	jtree, IS	-
BucketElim	Rish (U.C.Irvine)	C++	S	WU	D	N	N	N	N	0	D	Varelim	-
BUGS	MRC/Imperial College	N	N	WU	Cs	W	S	N	N	0	D	Gibbs	-
Business Navigator 5	Data Digest Corp	N	N	W	Cd	S	S	S	N	\$	D	Jtree	-
CABeN	Cousins et al. (Wash. U.)	C	S	WU	D	N	N	N	N	0	D	5 Métodos de Muestreo	-
Causal discoverer	Vanderbilt	N	N	W	-	-	N	S	N	0	D	-	Aprendizaje solo ví estructuras
CoCo+Xlisp	Badsberg (U. Aalborg)	C/lisp	S	U	D	S	S	CI	N	0	U	Jtree	Diseñado para tablas de contingencia
CIspace	Poole et al. (UBC)	Java	N	WU	D	S	N	N	N	0	D	Varelim	-
DBNbox	Roberts et al	Matlab	-	-	Y	N	S	N	N	S	D	Varios	DBNs
Deal	Bottcher et al	R	-	-	G	S	S	S	N	0	D	No	Aprendizaje vía estructuras
DeriveIt	DeriveIt LLC	N	-	-	?	?	S	S	?	\$	D	Jtree	Explota la estructura local en CPDs.
Ergo	Noetic systems	N	S	W,M	D	S	N	N	N	\$	D	jtree	-
GDAGsim	Wilkinson (U. Newcastle)	C	S	WUM	G	N	N	N	N	0	D	Exact	Análisis Bayesiano de grandes modelos lineales directos Gaussianos
Genie	U. Pittsburgh	N	WU	WU	D	W	N	N	S	0	D	Jtree	-

GMRFSim	Rue (U. Trondheim)	C	S	WUM	G	N	N	N	N	0	U	MCMC	Análisis Bayesiano de grandes modelos lineales indirectos Gaussianos
GMTk	Bilmes (UW), Zweig (IBM)	N	S	U	D	N	S	S	N	0	D	Jtree	Diseñado para reconocimiento de voces.
gR	Lauritzen et al.	R	-	-	-	-	-	-	-	0	-	-	-
Grappa	Green (Bristol)	R	-	-	D	N	N	N	N	0	D	Jtree	-
Hugin Expert	Hugin	N	S	W	G	W	S	CI	S	\$	CG	Jtree	-
Hydra	Warnes (U.Wash.)	Java	-	-	Cs	S	S	N	N	0	U,D	MCMC	-
Ideal	Rockwell	Lisp	S	WUM	D	S	N	N	S	0	D	Jtree	GUI requiere Allegro Lisp.
Java Bayes	Cozman (CMU)	Java	S	WUM	D	S	N	N	S	0	D	Varelim, jtree	-
KBaseAI	Codeas	N	S	W,U	D	N	N	N	N	\$	D	varelim	Arquitectura cliente/servidor, múltiples usuarios, control de acceso, lenguaje "query"
LibB	Friedman (Hebrew U)	N	S	W	D	N	S	S	N	0	D	none	Aprendizaje vía estructura
MIM	HyperGraph Software	N	N	W	G	S	S	S	N	\$	CG	Jtree	Maneja hasta 52 variables.
MSBNx	Microsoft	N	S	W	D	W	N	N	S	0	D	Jtree	-
Netica	Norsys	N	WUM	W	G	W	S	N	S	\$	D	jtree	-
Optimal Reinsertion	Moore, Wong (CMU)	N	N	W,U	D	N	S	S	N	0	D	none	Aprendizaje vía estructura

<u>PMT</u>	Pavlovic (BU)	Matlab/C	-	-	D	N	S	N	N	0	D	Propósito especial	-
<u>PNL</u>	Eruhimov (Intel)	C++	-	-	D	N	S	Y	N	0	U,D	Jtree	Versión C++ de BNT
<u>Pulcinella</u>	IRIDIA	Lisp	S	WUM	D	S	N	N	N	0	D	?	Evalúa sistemas con computo no probabilístico
<u>RISO</u>	Dodier (U.Colorado)	Java	S	WUM	G	S	N	N	N	0	D	Polytree	Implementación distribuida.
<u>Sam Iam</u>	Darwiche (UCLA)	N	N ?	WU ? (Java ejecutable)	G ?	S	S	N ?	S	0	D	Recursivo condicional	Efectúa análisis sensitivo
<u>Tetrad</u>	CMU	N	N	WU	G	N	S	CI	N	0	U,D	No	-
<u>UnBBayes</u>	?	Java	-	-	D	S	N	S	N	0	D	Jtree	K2 para aprendizaje vía estructura
<u>Vibes</u>	Winn & Bishop (U. Cambridge)	Java	S	WU	Cx	S	S	N	N	0	D	Variacional	No disponible
<u>Web Weaver</u>	Xiang (U.Regina)	Java	S	WUM	D	S	N	N	S	0	D	?	-
<u>WinMine</u>	Microsoft	N	N	W	Cx	S	S	S	N	0	U,D	None	Aprende vía BN o estructuras de red dependientes
<u>XBAIES 2.0</u>	Cowell (City U.)	N	N	W	G	S	S	N	S	0	CG	Jtree	-

Selección del Paquete de Software

Dentro de todos los paquetes de software analizados se escogió el paquete de software NETICA de la empresa Norsys Software Corp. 3512 West 23rd Avenue, Vancouver, BC, CANADA (www.norsys.com). 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.

Especificaciones Generales de NETICA

- 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.

Bibliografía

- [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.