Redes Bayesianas en la Ingeniería del Software

Daniel Rodríguez[1], Javier Dolado[2]

Universidad de Alcalá[1], Universidad del País Vasco[2]

Resumen. Muchas de las actividades en la ingeniería del software, como por ejemplo, la estimación de costes o esfuerzo, evaluación de riesgos o fiabilidad tratan con valores inciertos o probabilísticas. Por tanto, diversas técnicas estadísticas y la teoría de la probabilidad han sido aplicadas a la ingeniería del software desde sus inicios. Más recientemente, modelos gráficos, que combinan probabilidad y teoría de grafos, están siendo aplicados a problemas de la ingeniería del software donde la incertidumbre está presente. En este capítulo, se proporciona una visión general de las redes Bayesianas, sus fundamentos en la teoría de la probabilidad, la noción de propagación y su construcción, incluyendo técnicas de minería de datos. Además, se describen diferentes extensiones de las redes Bayesianas, así cómo su aplicación a la ingeniería del software y comparación con otras técnicas.

Palabras clave: Redes Bayesianas, estimación en la ingeniería del software, minería de datos en la ingeniería del software

Introducción

Las redes Bayesianas son modelos gráficos probabilísticos utilizados en la toma de decisiones (Castillo et al 1998, Neapolitan 2004, Korb y Nicholson 2004, Jensen 2001, 1996). Una red Bayesiana representa una función de distribución conjunta sobre un conjunto finito de variables. Se componen de dos partes:

- La parte cualitativa, es una estructura gráfica (grafo) que describe las posibles entidades (variables) y dependencias entre ellas.
- La parte cuantitativa esta compuesta por probabilidades condicionadas que representan la incertidumbre del problema, dicho de otro modo, creencias de las relaciones causa efecto entre los nodos.

A modo de ejemplo, la red Bayesiana de la Figura 1 representa un modelo muy simplificado de estimación de errores en la ingeniería del software. Cada nodo del grafo representa una variable del dominio: defectos *insertados*, *detectados* y *residuales* (número de defectos que permanecen en el código una vez entregado al cliente). Cada arco del grafo representa una relación causal entre variables; los defectos *insertados* influyen en el número de defectos *detectados* durante el testeo y el número de defectos *residuales*, a su vez, el número de defectos *detectados* influye

en el número de defectos *residuales*. En este caso, cada variable puede tomar solamente dos valores, *bajo* o *alto*. Las relaciones entre variables son caracterizadas por medio de las tablas de probabilidad, también mostradas en la Figura 1.

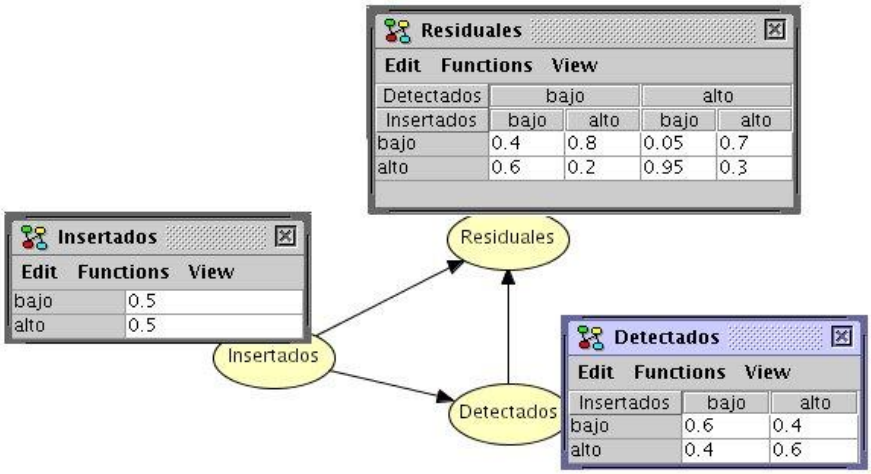

Fig. 1. Red Bayesiana simplificada para la estimación de defectos

Una vez que se tienen evidencias sobre el estado de ciertas variables, es decir, cuando tenemos conocimiento o podemos observar su estado, se actualizan las tablas de probabilidad propagando las nuevas probabilidades.

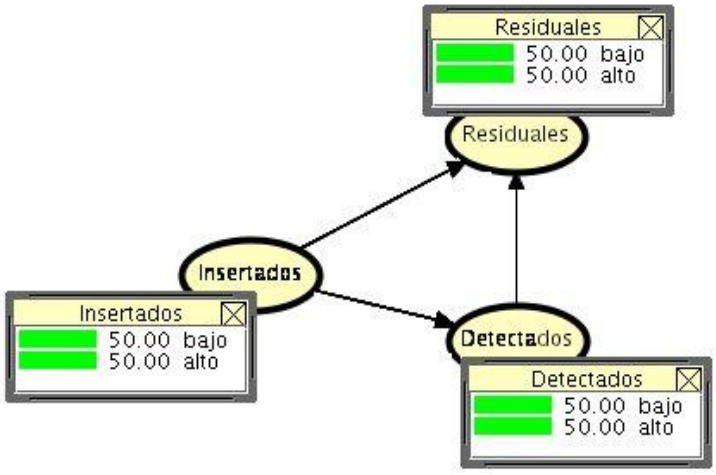

Fig. 2. Modo de ejecución sin evidencias introducidas

En general, los programas para manipular redes Bayesianas tienen dos modos de operar: un modo de edición que permite la creación de redes y la especificación de

tablas de probabilidad como muestra la Figura 1, y un modo de consulta (ver Figuras 2 y 3), donde se introducen las evidencias y se propagan las nuevas probabilidades. En el caso de la Figura 2 muestra las probabilidades de cada variable sin evidencias introducidas. En este caso, las probabilidades están distribuidas uniformemente y no se puede decir mucho del estado del dominio.

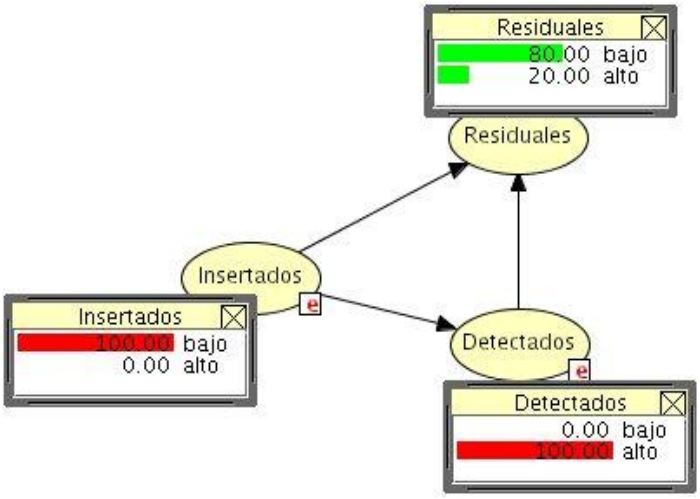

Fig. 3. Modo de ejecución con evidencias introducidas

Sin embargo, si tenemos la certeza (100% de probabilidad) de que el número de defectos *insertados* es *bajo*, y de que un *alto* número de defectos fueron *detectados* durante el testeo, la red Bayesiana propaga las probabilidades estimando que el número de defectos *residuales* será *bajo* con una probabilidad de 0,8 y *alto* con una probabilidad de 0,2. La Figura 3 muestra dicho escenario, las barras que marcan un 100%, indican variables con estado conocido; el resto muestran las nuevas probabilidades después de propagar las evidencias. Nótese que la salida es una distribución de probabilidades en vez de un único valor. Naturalmente, consideraremos el valor con mayor probabilidad como el valor estimado.

Fundamentos de las redes Bayesianas

Formalmente, una red Bayesiana es un grafo dirigido acíclico(Neapolitan 2004, Korb y Nicholson 2004, Jensen 2001, Jensen 1996). Los nodos representan variables aleatorias del dominio $X_1, X_2, \ldots, X_n$ y los arcos representan relaciones de dependencia entre variables. Las redes Bayesianas asumen que un nodo depende solamente de sus padres y que cada nodo esta asociado a una *tabla de probabilidades condicionales*, que definen la probabilidad de cada estado en los que puede estar una variable, dados los posibles estados de sus padres. Una red Bayesiana se mues-

tra la probabilidad de distribución conjunta para un conjunto de $X_1, X_2, \ldots, X_n$ tal que:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1..n} P(x_i \mid padres(X_i)) \quad (1)$$

donde x_i representa el valor que toma la variable X y $padres(X_i)$ denota los valores que tienen el conjunto de los padres en la red Bayesiana del nodo X_i. Por tanto, cada estado de una variable puede ser calculado multiplicando un número reducido de valores en las tablas de probabilidad.

Tipos de Evidencia

Las redes Bayesianas se utilizan en problemas en los que interviene la incertidumbre, es decir, donde no se tiene un completo conocimiento del estado del sistema, pero que sin embargo, podemos realizar observaciones (obtener evidencias) y entonces, actualizar las probabilidades del resto del sistema. Hay dos tipos de evidencia:

- *Evidencia firme* o específica (instanciación), se da cuando se asigna un valor concreto a una variable, es decir, tenemos certeza del estado de dicha variable. Por ejemplo, imagínese que la variable *A* representa el *resultado* de un partido de baloncesto, con dos posibles estados (*ganar*, *perder*); si conocemos que el equipo ha ganado el partido, podremos asignar la probabilidad 1 (certeza absoluta) al estado *ganar* y 0 al estado *perder*.
- *Evidencia parcial* o *virtual* de un nodo, permite actualizar las probabilidades a priori de los estados que puede tomar la variable. Por ejemplo, a la variable *resultado* del partido de baloncesto, si el equipo pierde por 20 puntos a pocos minutos del final, podríamos asignar una probabilidad muy alta a *perder* y muy baja a *ganar*.

Variables Continuas

Las redes Bayesianas permiten usar variables continuas, sin embargo, al haber un número infinito de estados, el problema reside en la especificación de las tablas de probabilidad condicional. Hay dos formas de abordar este problema:

- La *discretización* consiste en dividir el rango de las variables continuas en un número finito de intervalos *exhaustivos* y *exclusivos*. Por ejemplo, si la variable temperatura en un dominio a modelar puede tomar cualquier valor entre 0 y 100, es posible dividir el rango en un número finito de intervalos: (0-20], (20-40] y (40-100]. Naturalmente, al discretizar se pierde información que de-

pende del dominio y el número de intervalos. Es el método más común ya que la mayoría del las herramientas y algoritmos se basan en nodos discretos.

– El segundo método consiste en usar modelos paramétricos como la distribución Gausiana, que es representada por dos parámetros, la media y la varianza.

Inferencia en redes Bayesianas

Una red Bayesiana proporciona un sistema de inferencia, donde una vez encontradas nuevas evidencias sobre el estado de ciertos nodos, se modifican sus tablas de probabilidad; y a su vez, las nuevas probabilidades son propagadas al resto de los nodos. La propagación de probabilidades se conoce como *inferencia* probabilística, es decir, la probabilidad de algunas variables puede ser calculada dadas evidencias en otras variables. Las probabilidades antes de introducir evidencias se conocen como probabilidades *a priori*; una vez introducidas evidencias, las nuevas evidencias propagadas se llaman probabilidades *a posteriori* (Huang y Darwiche 1996, Cooper 1987, Russell y Norvig 2003).

En redes formadas por un gran número de nodos y dependencias, la propagación de probabilidades tiene un alto coste computacional, siendo un problema *NP-complejo* (Cooper 1987). Por ejemplo, una red con n variables booleanas contiene 2^n valores. Sólo a partir de finales de los años 80, con el desarrollo de nuevos algoritmos de propagación ha sido posible usar redes capaces de modelar problemas reales. No es necesario entender los algoritmos de propagación para utilizar redes Bayesianas, sin embargo es importante entender los principios básicos para seleccionar los algoritmos más adecuados y las herramientas que los implementan. Los algoritmos de propagación pueden dividirse en dos grandes grupos: (i) algoritmos de propagación *exactos*, si no hay error en las probabilidades calculadas, y (ii) algoritmos *aproximados*, en cuyo caso las probabilidades de los nodos son estimadas con cierto margen de error.

Los *algoritmos de propagación exactos* más simples incluyen *Inferencia mediante Enumeración* y *Eliminación de variables* (Russell y Norvig 2003). En redes múltiplemente conexas, los algoritmos de agrupamiento desarrollados por Lauritzen y Spiegelhalter (1988) son los más utilizados. Los *algoritmos de propagación exacta* funcionan bien con redes de hasta aproximadamente 30 nodos, sin embargo, no son apropiados para redes con más nodos o altamente conexas. Para estos casos se han desarrollado algoritmos aproximados.

Los métodos aproximados pueden clasificarse en métodos de simulación estocástica y los métodos de búsqueda determinista. Dentro de los métodos aproximados, los de simulación estocástica, también llamados de *Montecarlo*, son los más estudiados y utilizados. Esta clase de algoritmos realizan la inferencia por medio de muestreos de números aleatorios que dependen de las probabilidades de la red. Las

probabilidades condicionales se calculan dependiendo de las frecuencias generadas en el muestreo. Para una descripción más detallada de algoritmos de propagación se pueden consultar los libros de Neapolitan (2004) o Russell y Norvig (2003).

Ingeniería del Conocimiento utilizando redes Bayesianas

Por ingeniería del conocimiento en las redes Bayesianas se entiende el proceso de construcción, validación y utilización de redes Bayesianas. De manera simplificada, consiste en definir la estructura (el grafo) y los parámetros (las tablas de probabilidad) de la red. Para ello es necesario ejecutar los siguientes pasos (generalmente es necesario realizar varias iteraciones). Éstos pueden ser llevados a cabo por ingenieros del dominio o, si se tienen bases de datos del dominio, con ayuda de algoritmos de minería de datos:

- Una vez conocido el dominio que se quiere modelar, el primer paso consiste en seleccionar las variables de interés, seleccionando variables que no añadan complejidad innecesaria a la estructura. En dominios complejos con muchas variables, puede ser difícil enumerar todas las variables importantes y conocer sus relaciones causales.
- Para cada variable del dominio, es necesario decidir si será utilizada como variable entrada, su tipo (Booleano, etiquetas, numérico) y si es necesario discretizarla. En el caso de redes con variables continuas será necesario definir los parámetros de las variables continuas.
- Una vez seleccionadas las variables, es necesario definir la topología de la red, es decir, las relaciones causales entre las variables. Generalmente las redes creadas teniendo en cuenta las relaciones causales son más compactas, maximizando las independencias condicionales sin arcos innecesarios. Es importante distinguir entre correlación y causalidad. Causalidad implica correlación pero no a la inversa. Las técnicas de minería de datos pueden ayudar en la búsqueda de redes, pero generalmente necesitan ser complementadas por los expertos del dominio, bien añadiendo o borrando arcos, bien definiendo o corrigiendo direcciones de relaciones causales.
- El siguiente paso consiste en la definición de las tablas de probabilidad para cada uno de los nodos. En este paso, los ingenieros del conocimiento pueden ayudarse de la minería de datos si existen bases de datos del dominio.
- Una vez creada la red Bayesiana es necesaria su evaluación y verificación de su utilidad; por ejemplo, mediante el análisis de sensibilidad para comprobar cómo la variación de valores introducidos como evidencias en ciertas variables afectan a los resultados en el resto de variables.

En sistemas con pocas variables y estados, estos pasos pueden ser llevados a cabo por expertos del dominio o ingenieros del conocimiento especificando la red y los parámetros a mano. Sin embargo, a menudo se tienen bases de datos del dominio (en la ingeniería del software, por ejemplo, históricos de proyectos), donde los expertos utilizan los pasos anteriores combinados con la minería de datos para obtener las redes. En estos casos, es posible automatizar en cierta medida la creación de redes Bayesianas siguiendo los pasos típicos de la minería de datos que se describen a continuación (Fayyad et al. 1996):

- Preparación de los datos: Los datos son formateados de forma que las herramientas puedan manipularlos, juntar diferentes bases de datos, etc.
- Selección y limpieza de los datos: Típicamente, este paso consiste en la selección de variables, discretizar los datos, decidir sobre valores anómalos, ruido, etc.
- Minería de datos: Es en este paso se produce la extracción del conocimiento, donde se aplican algoritmos para la creación de las redes Bayesianas. El proceso de minería de datos para la creación de redes Bayesianas esta compuesto de dos tareas principales al igual que se ha descrito anteriormente:
    1. Inducción del mejor modelo cualitativo partiendo de los datos y/o conocimiento previo de los expertos del dominio.
    2. Estimación de las tablas de probabilidades: una vez la estructura ha sido definida, se definen las tablas de probabilidad, siendo el método más sencillo, el que se basa en las frecuencias obtenidas de los datos.
- Interpretación de los resultados: en el caso de las redes Bayesianas consistiría en realizar inferencias basándose en las evidencias obtenidas.
- Asimilación y explotación de los resultados.

La Figura 4 muestra los típicos pasos comentados, indicando que el experto de dominio necesita complementar los algoritmos de minería de datos con la limpieza los mismos, añadiendo o quitando arcos del grafo, asignando direcciones a los arcos, etc. El último paso indica que una vez que la red ha sido creada, puede ser utilizada para realizar inferencias.

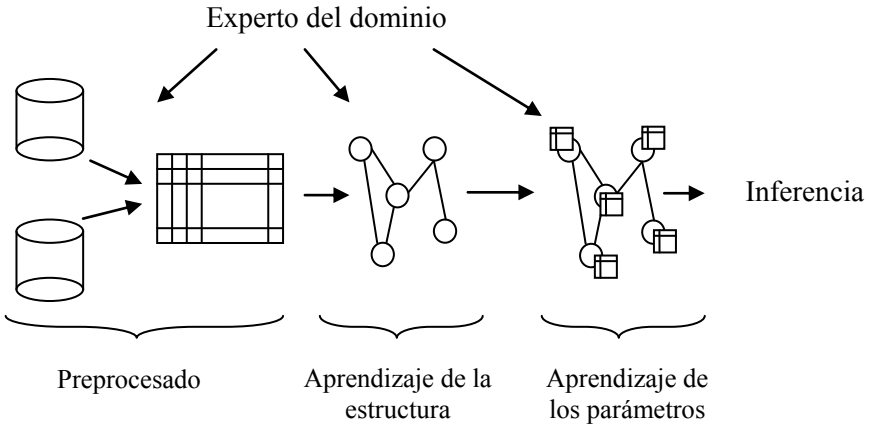

Fig. 4. Pasos típicos en la construcción de redes Bayesianas partiendo de bases de datos

Como ejemplo de la creación de redes Bayesianas en la ingeniería del software, imaginemos que una compañía con una base de datos de proyectos con los atributos definidos en la Tabla 1 y un subconjunto de los datos podría ser como los que se muestran en la

Tabla **2**. En cursiva y negrita se destacan posibles errores en los datos que pueden darse: errores de escritura (p.e., *organic* en vez de orgánico) que los algoritmos de aprendizaje confundirán con atributos diferentes, *outliers* (p.e., 3098 en vez de 309) o finalmente, valores que no fueron almacenados en la base de datos (generalmente son espacios en blanco, guiones, o signos de interrogación).

Tabla 1. Ejemplo de atributos en una base de datos de proyectos

Variable	***Descripción***	***Tipo***
Complejidad	Tipo de proyecto (*Orgánico*, *mediano* y *complejo*).	Discreta
Tamaño	No. de líneas de código (en miles)	Continua
Esfuerzo	Horas (p.e, persona/mes ~ 152h/mes)	Continua
Duración	No. de meses	Continua
Personal	No. de personas a tiempo completo requeridas para llevar a cabo el proyecto.	Continua
EsfDiseño	Calidad/experiencia del equipo de diseño, p.e., *Bajo*, *Medio*, *Alto*	Discreta
DefIntro	No. of defectos introducidos durante el desarrollo	Continua
EsfTest	Calidad/experiencia del equipo de testeo, p.e., *Bajo*, *Medio*, *Alto*	Discreta
DefRes	No. de defectos encontrados posteriores a la entrega	Continua

Tabla 2. Ejemplo de base de datos de proyectos

Tamaño	***Complejidad***	***Esfuerzo***	**…**	***DefIntro***	***EsfTest***	***DefDetec***	***DefRes***
10	Organico	26.9	…	240	Bajo	48	192
12	***Organic***	36.4	…	***3098***	Medio	39	90
	…	..	…	…	…	…	…
45	Embebido	346.5	…	1257	***?***	454	345

En consecuencia, los datos necesitan ser procesados para corregir los errores. Además, en el caso de las redes Bayesianas las herramientas generalmente necesitan los atributos discretizados. Para discretizar los atributos es necesario seleccionar el número de intervalos. Los intervalos se dividen teniendo en cuenta la frecuencia, es decir, el número de instancias en cada intervalo o utilizando el mismo rango, es decir, la misma distancia. Naturalmente, siempre se pierde 'información' al discretizar. La Tabla 3 muestra parcialmente cómo los atributos continuos pueden ser discretizados. Los atributos continuos se transforman en intervalos que pueden ser usados como etiquetas discretas, por ejemplo el *Tamaño* se ha dividido en un numero de intervalos etiquetados como ('<8.5', '>8.5<16.5', …, '>16.5<24.5'). El experto del dominio decidirá el numero de intervalos, rechazar o corregir instancias con *outliers*, etc.

Tabla 3. Ejemplo de variables discretizadas

Tamaño_d	***Esfuerzo_d***	**…**	***DefDetec_d***	***DefRes_d***
<8.5	<24.45	…	<41.5	<38.5
>8.5<16.5	<24.45	…	>41.5<77.5	>171.5<203.5
…	…	…	…	…
>16.5<24.5	>24.45<50.1	…	>77.5<114.5	>316.5<361.5

Una vez que hemos preprocesado los datos, empieza el proceso de aprendizaje (el proceso de minería de datos). En el caso de las redes Bayesianas consiste en aprender la estructura de la red y generar las tablas de probabilidades para cada nodo. En la mayoría de los casos en este paso también serán necesarios expertos del dominio para definir total o parcialmente la estructura de la red, decidiendo la dirección de los arcos (relaciones causa-efecto), nodos raíz/hoja y/o parámetros como umbrales, algoritmos, etc. Una vez obtenida la red es posible que sea necesario editarla para añadir, borrar o invertir los arcos. En la Figura 5 se muestra la pantalla de la herramienta Hugin (2006) para refinar la red obtenida por el algoritmo de minería de aprendizaje de la red.

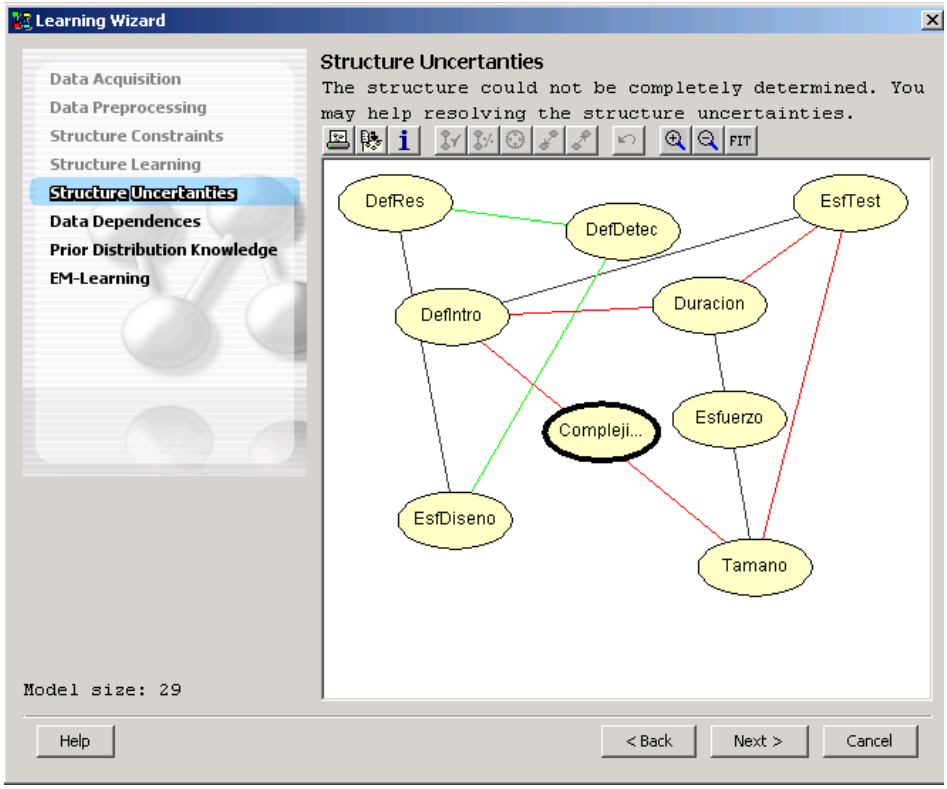

Fig. 5. Pantalla de Hugin para resolver incertidumbres de la estructura de la red.

Finalmente, una vez definida la estructura de la red, los parámetros de los nodos (las tablas de probabilidad) se calculan de acuerdo a la estructura y los datos suministrados. Una vez obtenida la red Bayesiana, se puede utilizar las herramientas de inferencia para razonar sobre el dominio. Debe puntualizarse que generalmente las herramientas no dan un único valor, sino una probabilidad por cada uno de los estados.

Por ejemplo, la Figura 6 muestra que si el *Tamaño* (*Tamano_d*, es la variable discretizada) está entre las 33.5 y 44.5 miles de líneas de código, la probabilidad de que la duración sea menor de 12.35 (etiquetado como '<12.35') es de 0.96, la probabilidad de que esté entre 12.35 y 16.25 es de 19.62, la probabilidad de que esté entre 19.15 y 21.45 es de 0.96 y de que sea mayor de 21.45 es solamente 0.96. Naturalmente, la predicción es que la duración estimada estará dentro el intervalo con mayor probabilidad. Para obtener valores concretos podría usarse la esperanza matemática: $E[X]=\sum_i x_i\, p(x_i)$, donde x_i podría ser el valor medio del intervalo.

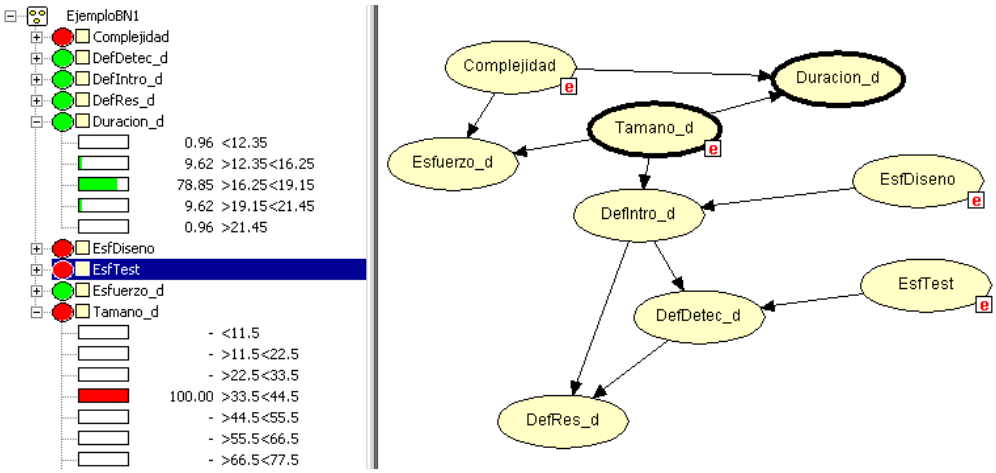

Fig. 6. Ejemplo de inferencia utilizando Hugin

Extensiones de las redes Bayesianas

Las redes Bayesianas estudiadas se pueden extender para acomodar la creación de redes partiendo de subredes, que tengan en consideración el tiempo, etc. En esta sección, se introduce una visión general de extensiones a las redes Bayesianas, aunque debe tenerse en cuenta que no todas las herramientas proveen estas facilidades.

Redes Bayesianas Orientadas a Objetos

Pearl (1986) ya definió *modismos* (*idioms*, en inglés) como fragmentos que pueden ser ensamblados, ya que es frecuente que una misma subred se repita varias veces en un mismo modelo o en otros modelos del mismo dominio. Los modismos o fragmentos pueden actuar como bloques que, unidos, forman sistemas más complejos. Además pueden ayudar a identificar la granularidad de los procesos que modelan.

Las Redes Bayesianas Orientadas a Objetos (OOBN, en sus siglas en inglés) facilitan la representación de redes con un gran número de nodos de una manera modular (Koller y Pfeffer 1997). Por tanto, es posible reutilizar subredes ya construidas dentro del sistema en las que cada subred tiene su propia identidad, ahorrando tiempo y esfuerzo a los ingenieros del conocimiento encargados de modelar el sistema. La topología y las tablas de probabilidad pueden ser reusadas sin ninguna modificación. Las redes pueden estar compuestas de subredes, que a su vez, pueden contener otras subredes, etc. En la creación de este tipo de redes, los autores han he-

redado mucha de la terminología del paradigma orientado a objetos. El mayor beneficio de las redes orientadas a objetos es que la construcción y las inferencias son realizadas de manera modular.

Hasta la fecha, muy pocas herramientas implementan redes orientadas a objetos, una de ellas es Hugin (2006), en la que se incluyen nodos llamados *instancias* que representan subredes. Las *instancias* o *módulos* contienen nodos de *interfaz* que pueden ser de *entrada* o de *salida.* La Figura 7 (a) muestra un ejemplo de subred y su equivalente de forma abstracta (Figura 7 (b)). Las subredes se comportan como redes Bayesianas normales, de manera que también pueden ser usadas para el razonamiento. Para realizar la inferencia en las redes Bayesianas orientadas a objetos, lo normal es transformarlas en redes normales y entonces realizar las inferencias.

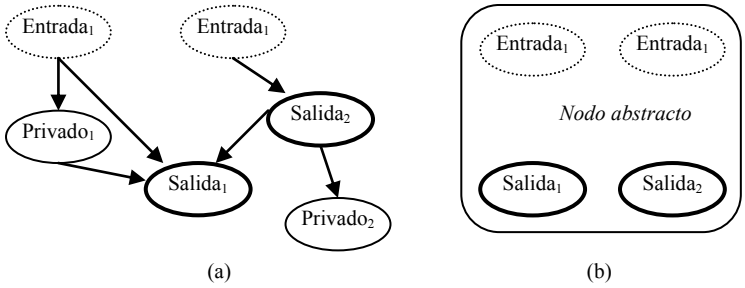

Fig. 7. (a) OOBN extendida (b) OOBN representada de modo abstracto

En ingeniería del software, Neil et al (2000) han aplicado las redes Bayesianas orientadas a objetos y han propuesto una metodología basada en una serie de modismos para ayudar a los expertos del dominio a crear redes complejas con un gran número de nodos en un proyecto llamado SERENE (*SafEty and Risk Evaluation using Bayesian Networks*).

Diagramas de Influencia

Aunque las redes Bayesianas son usadas para la toma de decisiones, los conceptos de utilidad y decisión no están modelados explícitamente. La teoría de la utilidad unida a las redes Bayesianas proporcionan un marco para la toma de decisiones llamado *diagramas de influencia* o *redes de decisión* (Jensen 1996).

Los diagramas de influencia, extienden las redes Bayesianas con dos nuevos tipos de nodos llamados *utilidad* (o *valor*) y *decisión* para modelar explícitamente la toma de decisiones. Además el nodo de forma circular u oval de las redes Bayesianas generales se denomina nodo de *azar* (ver Figura 8). Los valores de los nodos de *decisión* contienen las acciones que puede tomar la persona encargada de la toma de

decisiones (en nuestro caso el gestor de proyectos). El nodo *valor* representa la utilidad y cuantifica las preferencias (generalmente en dinero) de las decisiones.

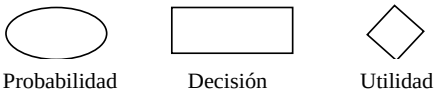

Fig. 8. Tipos de nodos en un diagrama de influencia

Por ejemplo, imagínese un escenario en la ingeniería del software donde el gestor de proyectos necesita decidir si lleva a cabo inspecciones (reuniones formales donde se evalúa el diseño o código) o no. Es de sobra conocido que las inspecciones de código reducen el número de defectos, y que corregir un error en un sistema ya en producción es mucho más caro que lo que sería durante el testeo; sin embargo, las inspecciones son costosas debido al número de horas invertidas por el personal. Por tanto, puede ser beneficioso recabar información adicional antes de decidirse por la realización de inspecciones. La Tabla 4 muestra la función de utilidad combinando el coste de realizar inspecciones junto con el hecho de que si entregamos al cliente un producto con alto número de errores, se perdería dinero en el proyecto. A menor número de errores, mayor beneficio proporcionará el proyecto.

Tabla 4. Función de utilidad

		Residual		
		Alto	***Medio***	***Bajo***
Inspeccionar	si	-50	0	50
	no	-25	5	100

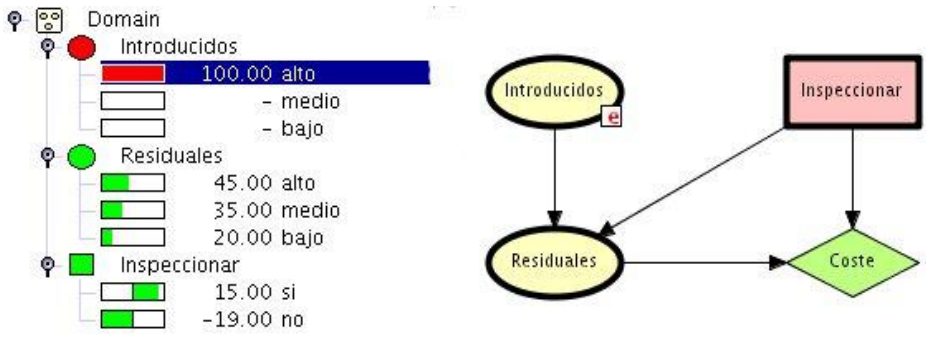

Fig. 9. Diagrama de influencia con bajo número de errores introducidos

Un posible escenario se muestra en el diagrama de influencia de la Figura 9, con el nodo de decisión *Inspeccionar* y el nodo valor *Coste* asociado con el nodo *Resi-*

dual. En este escenario se sabe que se han introducido un alto número de errores durante el desarrollo. El proyecto todavía puede ser rentable si se realizan inspecciones, en caso contrario, generará perdidas. Por otro lado, si sabemos que el número de defectos introducidos durante el desarrollo es bajo, el proyecto generará beneficios se hagan o no inspecciones, pero generará mayor beneficio si no se llevan a cabo las inspecciones (la Figura 10 muestra este escenario).

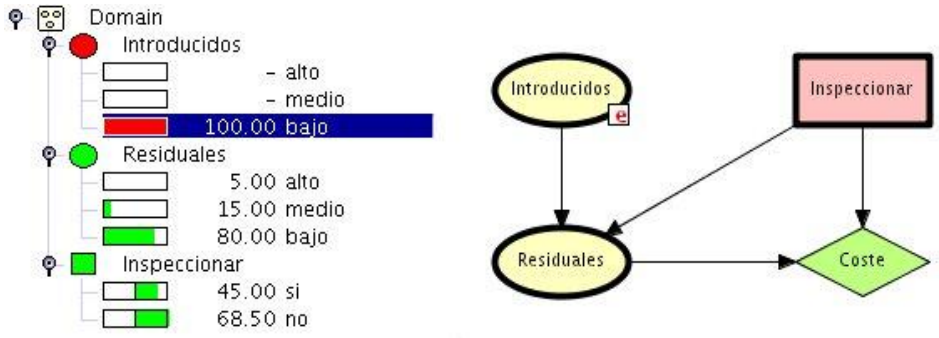

Fig. 10. Diagrama de influencia con un alto número de errores introducidos

Aplicaciones de las redes Bayesianas a la ingeniería del software

Las redes Bayesianas son cada vez más populares en ingeniería, inteligencia artificial y estadística. Se han aplicado con éxito a dominios como medicina, evaluación de riesgos, visión, diagnósticos de sistemas y redes, detección de fraude, *spam*, etc. (Heckerman et al. 1995). En la ingeniería del software las redes Bayesianas han sido usadas en diferentes áreas como por ejemplo:

– *Estimación del esfuerzo y calidad*. Fenton y Neil (1999) proporcionan una revisión crítica de los métodos disponibles en la literatura para la predicción de defectos, argumentando que los modelos basados en tamaño y complejidad no pueden predecir eficientemente. Los autores afirman que la incorporación de redes Bayesianas teniendo en cuenta factores como la habilidad de los programadores/analistas, la complejidad del diseño y los métodos y procedimientos usados, genera predicciones más exactas. Además, han demostrado que es posible introducir atributos del proceso y atributos del producto en una red Bayesiana. Stewart (2002) ha comparado las predicciones del clasificador naïve Bayes para estimar el esfuerzo de proyectos de software con otras técnicas de la minería de datos como árboles de regresión y redes neuronales, mostrando que una técnica tan simple como naïve Bayes tiene el potencial de poder ser usada como una posible técnica de estimación viable.

– *Pruebas del software*. Wooff et al (2002) analizan cómo las redes Bayesianas pueden ser usadas como herramienta de testeo, con preguntas del tipo "*qué pasa si…"* que ayudan a gestores y personal encargado del testeo en la toma de decisiones.

– *Fiabilidad*. Fenton et al. han llevado a cabo varios proyectos, como por ejemplo, DATUM (Dependability Assessment of safety critical systems through the Unification of Measurable) (Fenton 1995), SERENE (SafEty and Risk Evaluation using Bayesian Networks) en los que se han desarrollado redes para estimar la fiabilidad y seguridad de sistemas en distintos ámbitos (Fenton y Neil 2000).

– *Interfaces gráficos e interacción con el usuario*. Como ejemplo, el proyecto Lumiere (Horvitz et al. 1998) de Microsoft investigó formas de mejorar la interacción entre usuarios y software utilizando redes Bayesianas. Sirvió como base para el asistente de *Microsoft Office* cuando el usuario utilizaba la ayuda, puesto que el motor de inferencia de la red Bayesiana considera las acciones de usuario, eventos de las aplicaciones y perfiles de usuarios. También se han utilizado como asistentes de diagnóstico para la detección de problemas de impresión, etc.

Propiedades y limitaciones de las redes Bayesianas

Las redes Bayesianas tienen un número de características que hacen que sean apropiadas para la ingeniería del software:

- *Representación gráfica*. Por su naturaleza, las redes Bayesianas proveen una representación gráfica de las relaciones explícitas de dependencia del dominio. Generalmente las variables en la ingeniería del software como por ejemplo esfuerzo o coste, están influenciados por muchos factores. Las redes Bayesianas nos permiten modelar sistemas complejos permitiéndonos entender las relaciones causales visualizándolas por medio del grafo.

- *Modelado cualitativo y cuantitativo*. Las redes Bayesianas están formadas por un componente cualitativo, el grafo, y una parte cuantitativa, las tablas de probabilidades, que permiten utilizar criterios objetivos (por ejemplo, utilizando proyectos finalizados) y subjetivos (por ejemplo, utilizando creencias de expertos del dominio.

- *Inferencia bidireccional*. Las redes Bayesianas pueden hacer inferencia en ambos sentidos, es decir, las variables de entrada pueden ser usadas para predecir las variables de salida y viceversa. Fijando las variables de salida con los valores deseados, es posible predecir qué valores de las variables de entrada permiten dicha salida. Por ejemplo, usando inferencia hacia adelante, se puede predecir el número final de defectos basándose en variables como tamaño del proyecto, complejidad, esfuerzo en diseño, esfuerzo en testeo, etc. Por el contrario, po-

dríamos fijar un número de defectos y predecir que esfuerzo necesario satisface dicha salida.

- *Análisis de sensibilidad.* Dado un conjunto de evidencias, las redes Bayesianas permiten fácilmente calcular la sensibilidad de ciertas variables, simplemente modificando las evidencias.
- *Incertidumbre*. Las redes bayesianas pueden modelar grados de certidumbre, en vez de valores exactos. Por tanto, permiten modelar la incertidumbre de manera efectiva y explícitamente, por lo que pueden realizar buenas predicciones con información incompleta. De hecho, Kitchenham y Linkman (1997) afirman que las estimaciones en la ingeniería del software son una evaluación probabilística de un suceso futuro, y es por tanto la razón de que los gestores de proyecto no obtengan buenos resultados.
- *Valores de confianza*. La salida de una red Bayesiana es una probabilidad de distribución en vez de valores únicos. Este tipo de información se puede usar para medir la confianza que podemos depositar en la salida de la red Bayesiana, lo cual es esencial si el modelo va a ser usado en la toma de decisiones. Por ejemplo, en una variable con estados *bajo*, *medio*, y *alto*, la red Bayesiana estima la probabilidad de cada uno de los estados.

A pesar de sus ventajas, las redes Bayesianas tienen las siguientes limitaciones o dificultades a la hora de crearlas:

- *Estructura*. Puede ser difícil describir una estructura compleja incluso para expertos del dominio, especialmente si el dominio es nuevo, creando desigualdades entre el problema del dominio y el modelo construido.
- *Variables ocultas*. Si dos variables están relacionadas de forma poco evidente, entonces habrá una dependencia entre ellas y puede que el modelo no tenga esta relación en cuenta.
- *Probabilidades inconsistentes*. En un sistema con un gran número de variables, puede ser difícil asegurar su consistencia. La inferencia Bayesiana es útil sólo si se puede confiar en los parámetros. Una mala estimación de los parámetros distorsionaría toda la red e invalida los resultados.
- Cuando se construyen redes Bayesianas utilizando técnicas de minería de datos, existen numerosas fuentes de error como por ejemplo en la discretización de variables, ruido, valores perdidos en la base de datos, etc.

Comparación con otras técnicas

Las técnicas usadas para la estimación dentro de la ingeniería del software están basadas principalmente en la estadística, especialmente en los modelos de regresión.

- *Modelos de regresión.* Este tipo de modelos no pueden representar relaciones causales y por tanto no pueden predecir con exactitud al no incorporar todos los aspectos del dominio. Pongamos por ejemplo el caso de la estimación de defectos a lo largo del ciclo de vida de un proyecto. En la Figura 11 (a) se muestra el modelo de regresión que sólo tiene en cuenta el tamaño para predecir el número total de defectos del sistema que se está desarrollando, tiene la forma: *DefectosResiduales = f(Tamaño)*. Los modelos basados en la regresión generalmente no tienen en cuenta el testeo y no producen estimaciones fiables. Fenton y Neil (1999) enumeran los problemas de los modelos tradicionales: (i) modelos simplistas que no incorporan variables importantes; (ii) defectos estadísticos y teóricos; (iii) falta de factores causales que explican variaciones en los resultados (iv) son cajas negras que esconden suposiciones cruciales, y (v) no tienen en cuenta la incertidumbre, ni los modelos, ni en las entradas y salidas. Por otro lado, la red Bayesiana mostrada en la Figura 11 (b) tiene en cuenta otras variables, proporcionando un entorno explicativo que facilita la toma de decisiones.

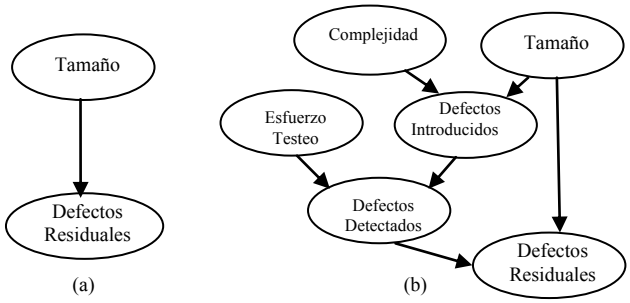

Fig. 11. (a) Modelo basado en regresión; (b) modelo causal

- *Redes neuronales.* Se ha demostrado que las redes neuronales producen muy buenos resultados en bases de datos con muchas instancias. Sin embargo, a diferencia de las redes Bayesianas, las redes neuronales no consideran la incertidumbre. Además, las redes neuronales actúan como una caja negra en el sentido de que no es posible saber como se ha llegado a los resultados obtenidos, ni los nodos intermedios pueden ser interpretados. En las redes Bayesianas todos los nodos y las tablas de probabilidad pueden ser interpretados con respecto al dominio. Otra desventaja de las redes neuronales con respecto a las Bayesianas, es que no pueden incorporar conocimiento de los expertos del dominio, solo pueden ser generadas mediante entrenamiento usando bases de datos y si se incrementa el número de registros en la base de datos hay que entrenar de nuevo a la red. Las redes Bayesianas permiten ser generadas por expertos del dominio, bases de datos

o por la combinación de ambas, con lo que permiten adquirir conocimiento de forma incremental.

- *Sistemas basados en reglas.* Un sistema basado en reglas consiste en una serie de reglas de la forma: *Si* (aserción) *entonces* (acción). Tales reglas son usadas para actuar acorde con la información obtenida. La ventaja de los sistemas basados en reglas es su simplicidad, pero son más apropiados en entornos determinísticos, no siendo este el caso de la ingeniería del software. Para solventar este problema, se han extendido con lógica difusa permitiendo introducir incertidumbre de una manera simple e intuitiva. Otra diferencia consiste en que las redes Bayesianas actúan globalmente; es decir, cualquier nodo puede recibir evidencias y las probabilidades son propagadas globalmente, mientras que en sistemas basados en reglas el orden en que se deben aplicar las reglas viene dado.

Conclusiones

Las redes Bayesianas, modelos que combinan la teoría de grafos y de probabilidad, son aplicadas a la toma de decisiones en dominios donde la incertidumbre representa un papel importante, como es el caso de la ingeniería del software. Aunque este tipo de modelos han sido conocidos desde hace mucho tiempo, solamente han podido ser aplicados desde finales de los años 80, gracias al desarrollo de nuevos algoritmos que permiten la creación y propagación de probabilidades en redes suficientemente complejas como para representar problemas reales. En este capítulo se han introducido sus extensiones y uso en la ingeniería del software donde han sido aplicadas. Se ha proporcionado una visión general de la metodología para su construcción así como algunos algoritmos que podrían considerarse como parte de la minería de datos. Además, se han comparado con otras técnicas más tradicionales dentro de la ingeniería del software y se han descrito sus ventajas e inconvenientes.

Hasta la fecha, la aplicación de las redes Bayesianas en la ingeniería del software ha sido menor que en otras áreas como, por ejemplo, en medicina. Esto puede ser debido a varias causas entre las que se incluyen: las bases de datos en la ingeniería del software son generalmente pequeñas o las mediciones son muy subjetivas; las herramientas están todavía madurando y no son especificas de la ingeniería del software, las relaciones causa efecto pueden ser difíciles de evaluar, etc. Sin embargo, es un área a investigar que está todavía en sus inicios. Fenton (1999) ha descrito: *"La clave de un uso más eficiente de métricas del software, no se encuentra en métricas más potentes, sino en la combinación inteligente de diferentes métricas […] Una vez dominada dicha combinación, se pueden estudiar métodos estadísticos avanzados para obtener buenas predicciones. Las redes Bayesianas son uno de los métodos con un futuro prometedor"*.

Referencias

Castillo E, Gutiérrez J M, Hadi A S (1998) Sistemas Expertos y Modelos de Redes Probabilísticas, Academia Española de Ingeniería.

Cooper G F (1987) Probabilistic Inference Using Belief Networks is NP-Hard, Knowledge Systems Laboratory, Stanford University.

Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) The KDD Process for Extracting Useful Knowledge From Volumes of Data. Communications of the ACM, 39 (11), pp. 27-34.

Fenton N E (1995) Multi-criteria Decision Aid; with emphasis on its relevance of in dependability assessment. DATUM/CSR/02, 1999, City University, London.

Fenton N E, Neil M (1999) A Critique of Software Defect Prediction Models. *IEEE Transactions on Software Engineering*, 25 (5), pp. 675-689.

Fenton N E, Neil M (2000) Software Metrics: A Roadmap. en Finkelstein, A. (Ed.) *The Future of Software Engineering*. ACM Press.

Heckerman D E, Mamdani E H, Wellman M P (1995) Real-World Applications of Bayesian Networks. *Communications of the ACM*, 38(3), pp. 24-30.

Horvitz E, Breese J, Heckerman D, Hovel D, Rommelse K (1998) The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software User. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, pp. 256–265

Huang C, Darwiche A (1996) Inference in Belief Networks: A procedural guide. International Journal of Approximate Reasoning, 15 (3), pp. 225-263.

Hugin (2006) Hugin. http://www.hugin.com/.

Jensen F V (1996) An Introduction to Bayesian Networks, London, UCL Press.

Jensen F V (2001) Bayesian networks and decision graphs, New York, Springer.

Kitchenham B A, Linkman S G (1997) Estimates, Uncertainty, and Risk. *IEEE Software*, 14 (3), pp. 69-74.

Koller D, Pfeffer A (1997) Object-oriented Bayesian networks. Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (UAI). Providence, Rhode Island, USA, pp. 302-313.

Korb K B, Nicholson a E (2004) Bayesian artificial intelligence, Chapman & Hall/CRC.

Lauritzen S L, Spiegelhalter D J (1988) Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society, Series B (Methodological), 50 (2), pp. 157-224.

Neapolitan R E (2004) Learning Bayesian networks, Prentice Hall.

Neil M, Fenton N E, Nielsen L (2000) Building large-scale Bayesian Networks. *The Knowledge Engineering Review*, 15 (3), pp. 257-284.

Pearl J (1986) Fusion, propagation, and structuring in belief networks. Artificial Intelligence, 29 (3), pp. 241 - 288.

Russell S J, Norvig P (2003) Artificial Intelligence: A Modern Approach, Prentice Hall.

Stewart B (2002) Predicting project delivery rates using the Naive-Bayes classifier, *Journal of Software Maintenance*, 14 (3), pp. 161-179.

Wooff DA, Goldstein M, Coolen FPA (2002) Bayesian Graphical Models for Software Testing. *IEEE Transactions on Software Engineering*, 28 (5), pp. 510-525.