

Reducción de parámetros en las Redes Bayesianas creadas a partir del Juicio Experto

Joseba Esteban López¹, Daniel Rodríguez², and José Javier Dolado Cosín¹

¹ Facultad de Informática de la Universidad del País Vasco U.P.V./E.H.U.,
P^a Manuel Lardizabal, 1, 20018 Donostia-San Sebastián, Spain

`joseba.esteban@gmail.com, javier.dolado@ehu.es`

² Universidad de Alcalá, Campus Externo, Edificio Politécnico
`daniel.rodriguez@uah.es`

Resumen La aplicación de las Redes Bayesianas en el campo de la estimación de costes a partir del conocimiento experto se ve obstaculizada por la cantidad de parámetros requeridos. Este estudio presenta dos propuestas para la reducción de la cantidad de parámetros requeridos por las Redes Bayesianas. Uno de ellos consiste en independizar la cantidad de parámetros de la cardinalidad de los nodos. La otra propuesta apuesta por reducir el tamaño de las tablas de probabilidad condicionadas de estas redes seleccionando un conjunto reducido de combinaciones entre los estados de las variables predictoras. La aplicación de ambas técnicas propuestas consigue reducir de manera significativa las estimaciones a proporcionar por el experto haciendo que la cantidad de éstas sea únicamente dependiente del número de variables predictoras representadas en la red.

Keywords: Esfuerzo Software, estimación, Redes Bayesianas, juicio experto, probabilidades condicionadas, reducir, distribución Normal

1. Introducción

La estimación del esfuerzo de desarrollo de un proyecto software permite a las compañías desarrolladoras de software evaluar la viabilidad del proyecto así como analizar sus posibles alternativas y gestionar los recursos necesarios para el desarrollo del mismo. Por tanto, disponer de una buena estimación del coste de desarrollo de los proyectos software resulta un aspecto clave en la gestión de la compañía desarrolladora.

A pesar de que se han desarrollado diferentes métodos de estimación a lo largo de las últimas décadas (ver [1] y capítulo 10 de [2]), los resultados esperados en cuanto a la exactitud de las estimaciones obtenidas están aún por alcanzar [3]. Estos métodos deben proporcionar la estimación del coste software antes de iniciarse el desarrollo del proyecto. La incertidumbre ligada al desarrollo del software ha repercutido negativamente en la exactitud de los estudios precedentes, los cuales suelen basar sus estimaciones en fórmulas matemáticas más o menos

complejas sin valorar las particularidades de este tipo de proyectos.

Las redes Bayesianas se presentan como un método eficaz para la gestión de la incertidumbre, siendo capaz de representar las relaciones entre los elementos de un proceso de razonamiento [4], [5] y cap.º10 de [2]. Tanto la estructura de la red como los parámetros necesarios en las redes Bayesianas se pueden aprender o definir a partir de datos almacenados sobre estimaciones anteriores. Mendes et al. realizan un estudio sobre la estimación del esfuerzo software aprendiendo varias redes Bayesianas a partir de datos históricos obteniendo una baja exactitud en sus resultados [6] [7].

Estas redes probabilísticas también permiten obtener sus parámetros a partir del conocimiento experto, permitiendo además manejar la incertidumbre propia de las estimaciones realizadas por el experto. Si el experto debe estimar una gran cantidad de parámetros en el proceso de captura del conocimiento, puede proporcionar estimaciones con inconsistencias y/o prejuicios que provocarán errores en las estimaciones proporcionadas y, por tanto, disminuirán la exactitud de las estimaciones resultantes. Esto implica que la captura de dicho conocimiento debe realizarse de la forma más reducida posible. Los autores de [8] presentan en su estudio una alternativa para reducir la cantidad de parámetros a estimar por el experto aplicando AHP (Analytic Hierarchy Process). Sin embargo, aunque disminuye significativamente la cantidad de parámetros, la reducción de estos no es suficiente para evitar los efectos de la fatiga en el experto [9].

En este estudio se presenta una alternativa para la reducción de los parámetros de las Redes Bayesianas a estimar por el experto. La propuesta tiene como base la aplicación de dos técnicas. Una de ellas está basada en las características de la distribución de probabilidad Normal (sección 3). La otra técnica propone la reducción del espacio muestral de las tablas de probabilidades condicionadas (sección 4). Para facilitar la comprensión de ambas técnicas, este estudio se apoya en un sencillo ejemplo, presentado en la segunda sección. Para terminar se realiza una comparación entre las estimaciones obtenidas aplicando dichas técnicas y sin aplicarlas.

2. Ejemplo y Planteamiento del Problema

Las redes Bayesianas son modelos gráficos probabilísticos que representan una función de distribución conjunta sobre un conjunto finito de variables. Las redes Bayesianas cuentan con dos partes: una parte cualitativa (representada mediante un grafo dirigido acíclico) y otra cuantitativa (representada mediante tablas de probabilidad). El grafo se compone de nodos (representados como circunferencias con el nombre en su interior) y arcos entre los nodos. Los nodos representan variables aleatorias, mientras que los arcos representan las dependencias entre las variables. La influencia entre nodos se representa gráficamente mediante un arco

dirigido, siendo el nodo hijo aquel al que apunta el arco. Las variables aleatorias pueden ser continuas o discretas. En el caso de ser discretas, estas se representan por el conjunto de estados que puede tomar la variable.

Los parámetros necesarios en una red Bayesiana tienen forma de distribuciones de probabilidad y son numerosos. Estos parámetros se dividen en dos grupos: las probabilidades marginales de los nodos padre y Las probabilidades condicionadas de los nodos hijo dadas todas las combinaciones entre los estados de sus nodos padre.

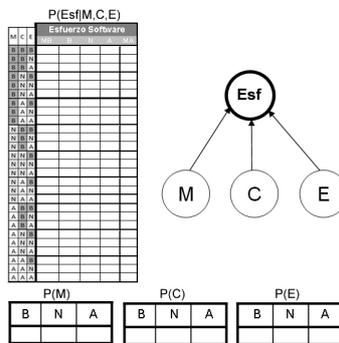


Figura 1. Ejemplo de los parámetros necesarios para utilizar una red probabilística con 3 nodos padre y un nodo hijo. Estos parámetros vienen representados por las tablas $P(Esf|M,C,E)$, $P(M)$, $P(C)$ y $P(E)$.

Con el objetivo de facilitar la comprensión de este estudio, seguiremos el sencillo ejemplo de red Bayesiana de la figura 1. Este ejemplo presenta una red Bayesiana para estimar el esfuerzo de desarrollo de un proyecto software: la clase o variable objetivo es *Esf* (*Esfuerzo Software*). La red del ejemplo sólo tiene un nodo hijo (*Esf*) y tres variables predictoras: la magnitud del proyecto (*M*), la complejidad del proyecto (*C*) y la experiencia de la plantilla involucrada en el desarrollo del proyecto (*E*). Todas ellas son nodos padre de *Esf* y no se influyen unas a otras. Los estados de cada variable aleatoria son:

- Esfuerzo (*Esf*): 5 posibles estados, Muy Bajo (MB), Bajo (B), Nominal (N), Alto (A) y Muy Alto (MA).
- Magnitud (*M*): 3 posibles estados, Baja (B), Nominal (N), Alta (A).
- Complejidad (*C*): 3 posibles estados, Baja (B), Nominal (N), Alta (A).
- Experiencia (*E*): 3 posibles estados, Baja (B), Nominal (N), Alta (A).

Los parámetros de la red están definidos por la regla probabilística $p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i|I_i)$ (sección 6.4.2 de [4]). Dentro del área del desarrollo de proyectos

software, se puede dar cualquier posible combinación de los estados de los nodos padre: en el caso del ejemplo de la figura 1 son 3 nodos padre con 3 estados cada uno, es decir, $3 \cdot 3 \cdot 3$ combinaciones; en general con n nodos padre y c estados cada uno serían $c_1 \cdot c_2 \cdot \dots \cdot c_n$ combinaciones; con 10 nodos padre (factores) y 3 estados cada uno, el número combinaciones asciende a $3^{10} = 59049$ combinaciones. Por tanto, la fórmula matemática que se utiliza debe tener en cuenta cualquier combinación entre los estados de los factores software (ver figura 1). Como se puede apreciar en la figura 1, los parámetros de la red son de dos tipos: probabilidades marginales ($P(M)$, $P(C)$ y $P(E)$) y probabilidades condicionadas ($P(Esf = esfi | M = m_j, C = c_k, E = e_l)$)

$$\begin{aligned}
 & P(Esf = esfi) = \\
 & = \sum_{\forall j,k,l} P(Esf = esfi | M = m_j, C = c_k, E = e_l) * P(M = m_j) * P(C = c_k) * P(E = e_l)
 \end{aligned}
 \tag{1}$$

La influencia entre nodos se representa parametricamente mediante tablas de probabilidad condicionadas. Estas tablas representan la probabilidad de ocurrencia de los estados del nodo influenciado o nodo hijo, dadas todas las combinaciones de los posibles estados que pueden tomar los nodos que lo influyen o nodos padre. Es decir, estas tablas representan la probabilidad de que ocurra cada uno de los posibles estados del nodo hijo dadas cada una de las combinaciones entre los estados de los nodos padre. Dicho de otro modo, cada fila de la tabla de probabilidades condicionadas de la figura 1 representa una distribución de probabilidad del nodo hijo dada la combinación de estados de los nodos padres correspondiente. Como se puede apreciar en la figura 1, la mayor parte de los parámetros de la red se concentra en la especificación de las tablas de probabilidad condicionadas (a partir de ahora nos referiremos a estas tablas por las siglas TPC).

El tamaño de las TPC depende del número de estados del nodo hijo, el número de nodos padre y la cantidad de posibles estados de estos nodos. Por tanto, para el ejemplo de la figura 1 el tamaño de la tabla condicionada es de 135 parámetros ($(n^P) * h$, siendo n el número de estados de los nodos padre, P el número de padres y h el número de estados del nodo hijo). Si se tratara de una red similar pero con 10 nodos padre, el tamaño sería de 295245 parámetros.

Normalmente, las redes Bayesianas se aprenden a partir de repositorios. Sin embargo, si se quiere aprender la red Bayesiana a partir del juicio experto, este deberá estimar todos los parámetros de la red. Por tanto, ante la cantidad de parámetros inherentes a las redes Bayesianas, se hace necesario reducir la cantidad de estimaciones a proporcionar por el experto [5].

3. Independencia del número de estados del nodo hijo: Distribución Normal

En este apartado se propone un método basado en las características de la distribución Normal para la generación de las probabilidades condicionadas independiente del número de estados del nodo hijo.

La distribución Normal es una de las distribuciones de probabilidad más utilizadas para representar variables aleatorias continuas. La Normal tiene dos parámetros: la media, μ , y la desviación típica, σ . La media representa el valor más probable y la desviación típica representa la media de distancias que tienen los datos respecto de su media aritmética. Es decir, la media es equivalente al centro de gravedad, desde el punto de vista físico, y divide la distribución en dos partes iguales, es también la mediana. El parámetro σ es una medida de dispersión respecto de μ de los datos representados en la distribución normal. La distribución Normal tiene forma acampanada y simétrica respecto de μ . Este estudio propone el uso la distribución Normal para la representación del juicio experto.

El juicio experto puede modelarse con dos parámetros al igual que la Normal: la valoración o el valor estimado por el experto y la confianza que se deposita en dicho experto (o cuánto de fiable es el juicio proporcionado por este). El primero de ellos se puede representar mediante el parámetro μ de la Normal, dado que, de los valores representados en la distribución de probabilidad, es el que cuenta con más probabilidad.

Mientras que σ se puede interpretar como el grado de confianza depositado en el juicio aportado por el experto. Como se ha comentado anteriormente, σ es una medida de la dispersión respecto de μ , es decir, respecto del juicio efectuado por el experto. Esto implica que cuanto más confianza depositemos en el juicio proporcionado por el experto, menor será la dispersión del resto de valores representados en la distribución Normal respecto de dicho juicio, μ . Y, a la inversa, cuanto menos confiemos en la opinión del experto mayor será la dispersión de la Normal.

De este modo, si se tiene plena confianza en la opinión del experto, todos los valores de la Normal deberían representarse sólo con μ , es decir, la dispersión sería nula: $\sigma = 0$. De modo inverso, si no se confía nada en el experto, σ debería ser máxima, de forma que la diferencia de probabilidad de los valores representados en la Normal sea mínima (con $\sigma = \infty$ se obtiene una distribución de probabilidad con todos sus valores equiprobables). Por tanto, es posible modelar las estimaciones realizadas por el experto utilizando la Normal.

Sin embargo, la distribución de probabilidad Normal no está acotada (varía entre $-\infty$ y $+\infty$). Esto dificultaría el manejo de las valoraciones proporcionadas por el experto utilizando la Normal, ya que no se podría establecer un valor como

máximo o mínimo. Dicho de otro modo, dado que el objetivo es modelar las opiniones del experto a cerca de determinados aspectos de un proyecto software, sería más sencillo trabajar con estimaciones acotadas en un rango más manejable e intuitivo para el experto, por ejemplo entre 0 y 100 o entre 0 y 10. Para poder utilizar una Normal continua con un rango entre 0 y 100 esta debe ser truncada.

Truncar los valores de la Normal (también denominada Gaussiana) menores a 0 y mayores de 100 implica perder esta información. Esto implica que, al discretizar (ver figura 3) una Normal en varios estados (por ejemplo en 3 estados, el primero de 0 a 33, el segundo de 33 a 66 y el tercero de 66 a 100) la suma de las probabilidades de los mismos no daría 1 (o 100% en caso de representar porcentajes) por lo que dejaría de ser una distribución de probabilidad. Esto se debe a que las probabilidades de los valores que quedan fuera del rango no se han tenido en cuenta en proceso de discretización. Sin embargo, normalizando los estados de la Gaussiana estas probabilidades se reparten entre los estados de la distribución respetando la proporción entre estos. De este modo, se hace factible el empleo de la Normal para la representación de los juicios del experto.

Partiendo de esta idea, en este estudio se ha establecido que la media estará representada por un valor estimado por el experto entre 0 y 100. El cálculo de σ no es tan directo como el de la media ya que primero es necesario conocer el grado de confianza depositado en las estimaciones realizadas por el experto y, a partir de ahí, calcular σ .

La desviación típica es una medida que informa de la media de distancias que tienen los datos respecto de su media aritmética, expresada en las mismas unidades que la variable. Acorde a esta definición, la desviación típica, σ , no se puede definir directamente puesto que una distribución Normal no está acotada, tiene un rango de $(-\infty, +\infty)$. No obstante, se puede establecer el rango donde se encuentren la mayoría de los datos representados en una distribución Normal. Como se puede ver en la figura 2, el 99,7% de los datos se encuentran a una distancia de 3σ respecto de la media. Es decir, el 99,7% de la masa de datos representados en una Normal está en un rango de 6σ centrado en la media μ . Esta propiedad de la Normal se conoce como la regla de 3σ , *68-95-99.7 rule* o *Three sigma rule*.

Asumiendo cierto error (queda un 0,3% de la masa de datos representados en la Normal a una distancia mayor de 3σ respecto de μ) se puede suponer que casi la totalidad de los datos representados en una Normal se encuentran en una distancia 6σ ($6\sigma = 100$). De este modo se puede determinar el valor de σ en una distribución Normal donde los datos están repartidos a lo largo del intervalo $[0, 100]$.

Partiendo de esta idea establecemos una desviación típica (σ) acorde al grado de fiabilidad del experto. Se trata de una relación inversa: cuanto mayor sea la

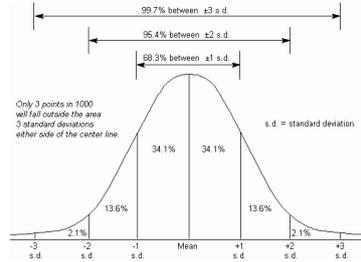


Figura 2. Distribución de los valores de una Normal en torno a σ

fiabilidad asociada al experto, menor debe ser la distancia media de los datos respecto de μ y, por tanto, más estrecho será el intervalo donde se representan los datos, o lo que es lo mismo, menor será la desviación típica de la Normal, y viceversa. La relación entre la desviación típica y la fiabilidad otorgada al experto se puede realizar con diferentes proporciones. La fórmula 2 establece una relación lineal inversa entre ambas.

$$\sigma = \sigma_{max} - Fiabilidad \quad 0 \leq Fiabilidad \leq \sigma_{max} \quad (2)$$

Sin embargo, en el dominio de la estimación del esfuerzo software, resulta complicado determinar de forma absoluta el nivel de fiabilidad del experto (ver estudio [10]). En consecuencia, se ha optado por aproximarlo como la suma ponderada entre la experiencia del experto dentro la ingeniería del software (en años), $ExIS$, la experiencia dentro de la compañía desarrolladora sobre la que se quiere estimar el esfuerzo (en años), $ExCO$, y el porcentaje de acierto en estimaciones posteriores, P . Estos factores no afectan por igual en la fiabilidad del experto, por tanto cada uno de ellos tiene asignado un peso: 10% para W_i , y 30% y 60% para W_c y W_p respectivamente.

Esta suma ponderada la representamos por el índice NEst, acrónimo de *Nivel como Estimator*. Sin embargo, el índice NEst debe ser normalizado para poder utilizar la expresión 2. La normalización de NEst debe realizarse dentro del rango $[0, 100/6]$ para que σ pueda ser máxima ($\sigma = 100/6$) cuando la fiabilidad del experto es mínima, y mínima ($\sigma = 0$) cuando se confía plenamente en las estimaciones del experto. En la figura 3 se muestra la expresión matemática que permite calcular NEst. Por tanto, la ecuación que permite calcular σ quedaría como se muestra en la fórmula 4, siendo $\sigma_{max} = 100/6$.

$$NEst = \frac{(W_{ExIS} \cdot ExIS + W_{ExCO} \cdot ExCO + W_P \cdot P) \cdot (100/6)}{7960} \quad (3)$$

$$\sigma = \sigma_{max} - NEst \quad 0 \leq NEst \leq \sigma_{max} \quad (4)$$

Una vez calculados los parámetros μ y σ , es posible representar las estimaciones realizadas por el experto mediante la *Normal*(μ, σ). Esta distribución es continua. Con el objetivo de obtener una probabilidad para cada uno de los estados que puede tomar la variable aleatoria, la distribución Normal debe ser discretizada aplicando la función de distribución acumulada (ver figura 3).

$$\Phi(x) = \int_{-\infty}^x \phi(t) dt = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt.$$

Figura 3. Función de Distribución de la Normal acumulada

Retomando el ejemplo de la figura 1, la clase (*Esf*) está definida con una cardinalidad de 5 posibles estados. Por tanto el rango de la Normal debe dividirse en 5 intervalos cada uno de los cuales representa un estado. A continuación se muestra la correspondencia entre intervalos y estados así como la probabilidad de cada uno de ellos:

- Estado Muy Bajo: Rango de [0, 19], $P(X = MuyBajo) = \phi(19) - \phi(0)$
- Estado Bajo: Rango de [20, 39], $P(X = Bajo) = \phi(39) - \phi(19)$
- Estado Nominal: Rango de [40, 59], $P(X = Nominal) = \phi(59) - \phi(39)$
- Estado Alto: Rango de [60, 79], $P(X = Alto) = \phi(79) - \phi(59)$
- Estado Muy Alto: Rango de [80, 100], $P(X = MuyAlto) = \phi(100) - \phi(79)$

Esta técnica también se puede aplicar en la estimación de la probabilidades marginales de los factores software. En este caso cada factor cuenta con tres estados, por lo que la probabilidad de cada estado se calcularía del siguiente modo:

- Estado Bajo: Rango de [0, 33], $P(X = Bajo) = \phi(33) - \phi(0)$
- Estado Nominal: Rango de [34, 66], $P(X = Nominal) = \phi(66) - \phi(33)$
- Estado Alto: Rango de [67, 100], $P(X = Alto) = \phi(100) - \phi(66)$

Aplicando esta técnica en el ejemplo de la figura 1, el número de estimaciones a proporcionar por el experto sería de 30: una estimación por cada factor, 3 en total, y una estimación por cada fila de la TPC, 27 estimaciones. La reducción es significativa pero no suficiente, puesto que con 10 factores (un caso más cercano a la realidad) la cantidad de estimaciones ascendería hasta 59059.

4. Reducción del tamaño de la tabla de probabilidades condicionadas

Esta sección muestra una técnica complementaria a la presentada en la sección 3 basada en la reducción del tamaño de la TPC con el objetivo de reducir la cantidad de parámetros a estimar por el experto.

Las TPC representan exhaustivamente la relación entre los nodos padre y el nodo hijo. Con este fin, se muestrean todas las posibles combinaciones entre los estados de los nodos padre, de forma que cada uno de los estados de la clase tiene una probabilidad para cada una de estas combinaciones.

La técnica presentada en esta sección consiste en reducir el tamaño del muestreo. Es decir, reducir la cantidad de combinaciones sobre las que basar la inferencia de la estimación. Retomando el ejemplo de la figura 1, consistiría en reducir las 27 posibles combinaciones entre los estados de los factores software M, C y E a un número reducido que evite el efecto de fatiga a la hora de capturar el juicio experto.

Este proceso implica la selección del conjunto reducido de casos (un caso hace referencia a una combinación entre los estados de los nodos padre). Estos casos se pueden seleccionar de diversas maneras, por ejemplo, aplicando el método de búsqueda de máxima probabilidad (sección 9.10 de [4]) o seleccionándolos aleatoriamente. Sin embargo, dada la incertidumbre intrínseca al desarrollo de proyectos software, no es conveniente basarse únicamente en aquellos casos que se pueden dar con mayor probabilidad, sino que es posible que se de cualquiera de ellos. Una selección aleatoria de casos implica que se obtendrían estimaciones diferentes ante los mismos datos de entrada. Este estudio propone seleccionar un conjunto de casos uniformemente distribuidos en el espacio muestral.

De esta forma, se han elegido 5 casos de acuerdo a su relación de influencia con la clase, *Esfuerzo Software*. Esta relación puede ser directa o indirecta. El factor *Complejidad del Proyecto* (C) tiene una relación directa con la clase: independientemente del resto de factores software, cuanto mayor es el valor de la Complejidad del proyecto, más alto es el *Esfuerzo Software* necesario para desarrollar dicho proyecto, aunque no se pueda determinar en qué grado. La *Magnitud del Proyecto* también tiene una relación directa con el *Esfuerzo Software*. En cambio, la relación de la *Experiencia de la Plantilla* (E) es inversa respecto del *Esfuerzo Software*: con independencia del resto de factores software, cuanto más experiencia tiene la plantilla involucrada en el desarrollo del proyecto software, menor es el esfuerzo necesario para desarrollarlo, aunque no se sepa cuánto menos. El tipo de relaciones entre los factores software y el *Esfuerzo Software* permite seleccionar un conjunto de casos repartido a lo largo del espacio muestral de combinaciones entre los estados de los factores. Los casos elegidos para el ejemplo de la figura 1 son:

- Caso Peor: M = Alto, C = Alto y E = Bajo
- Caso Malo: M = Nominal, C = Nominal y E = Bajo
- Caso Intermedio: M = Nominal, C = Nominal y E = Nominal
- Caso Bueno: M = Bajo, C = Bajo y E = Nominal
- Caso Mejor: M = Bajo, C = Bajo y E = Alto

Cada uno de estos casos tiene una probabilidad de ocurrencia. Esta probabilidad viene dada por el producto de la probabilidad marginal de cada factor, es decir, la probabilidad de que cada factor tome el estado correspondiente al caso (p.e. $p(\text{CasoPeor}) = p(M = \text{Alto}) \cdot p(C = \text{Alto}) \cdot p(E = \text{Bajo})$).

Sin embargo, dado que la TPC no explora todo el espacio muestral, el resultado de la estimación no daría como resultado una distribución de probabilidad (donde la suma de todas las probabilidades representadas en ésta es 1). Esto hace necesario normalizar las probabilidades de cada caso. Estas probabilidades serán las que posteriormente se utilicen en el proceso de inferencia de la estimación de la clase (ver figuras 1 y ecuación 5).

$$P(Esf = esf_i) = \sum_{\forall j,k,l} P(Esf = esf_i | M = m_j, C = c_k, E = e_l) * P(Caso) \quad (5)$$

Esta técnica, combinada con la presentada en la sección 3, permite reducir notablemente la cantidad de parámetros a estimar por el experto, quedándose en 5 estimaciones para la tabla de probabilidades condicionadas y una por factor. En el ejemplo de la figura 1, el experto sólo tendría que proporcionar 8 estimaciones. Para un caso similar pero con 10 factores, el experto debe realizar 15 estimaciones. Por tanto, aplicando ambas técnicas de reducción, el efecto de fatiga o cansancio del experto se mitiga notablemente mejorando la fiabilidad de estas.

5. Comparación de resultados

En esta sección se comparan las estimaciones obtenidas aplicando las técnicas de reducción presentadas en las secciones 3 y 4 con las estimaciones obtenidas sin aplicar dichas técnicas. El objetivo de esta comparación es verificar la semejanza entre las estimaciones obtenidas aplicando esta técnica y las obtenidas sin aplicarlas.

Con este fin, la comparación se realiza en base al ejemplo de la figura 1. La figura 5 muestra las estimaciones proporcionadas por un experto. Se trata de un ejemplo no basado en un caso real, por lo que el resultado de la estimación puede ser insólito. No obstante, la fiabilidad del resultado no es relevante en este estudio puesto que sólo cuenta con 3 variables predictoras, cuando lo habitual

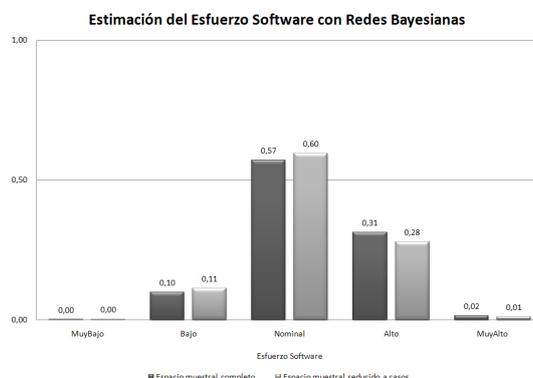


Figura 4. Comparación de las estimaciones obtenidas aplicando la reducción mediante selección de casos y sin dicha selección.

en otros métodos de estimación es que supere los 10 factores software.

A este experto se le han supuesto 10 años de experiencia en el área de la Ingeniería del Software, 5 años de experiencia dentro de la compañía encargada de desarrollar el proyecto software y un 40% de acierto en estimaciones anteriores. Este experto obtiene un índice NEst de 5,55. Por tanto, aplicando la fórmula 4, se obtiene una desviación típica de 11,12 ($\sigma = 11,12$) que se aplicará a todas las estimaciones que proporcione.

La figura 5 muestra las estimaciones del experto aplicando la técnica presentada en la sección 3 de este estudio. Esto implica que el experto ha estimado el valor más probable para cada factor (M,C y E) así como el valor más probable para el *Esfuerzo Software* (Esf) dada cada una de las combinaciones de estados representados en la TPC. En la figura 6 se muestra la probabilidad de cada estado obtenidas discretizando las distribuciones Normales definidas por las estimaciones del experto de la figura 5 y la fiabilidad otorgada al experto (σ).

Estos mismos parámetros han sido utilizados en la estimación del *Esfuerzo Software* aplicando la técnica presentada en la sección 4. Es decir, las probabilidades del *Esfuerzo Software* para cada caso coinciden con las utilizadas en la estimación sin aplicar la reducción del tamaño de la TPC. Esto nos permite evaluar la diferencia entre los resultados de ambas estimaciones. La figura 7 presenta los parámetros utilizados en la estimación de *Esfuerzo Software* a partir de un conjunto reducido de casos.

La figura 4 muestra los resultados de ambas estimaciones: en color gris oscuro la estimación obtenida sin aplicar la técnica de reducción y en gris claro la estimación aplicando la técnica de reducción de la sección 4. Como se puede apreciar ambas estimaciones son similares con una diferencia máxima del 3% para el es-

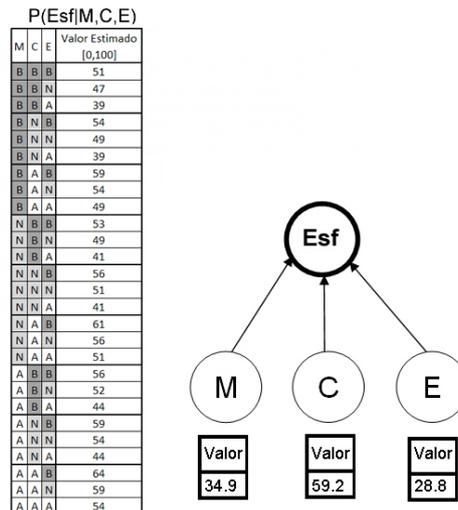


Figura 5. Red Bayesiana con las estimaciones proporcionadas por el experto.

tado *Nominal* y *Alto*.

6. Conclusiones

Una de las características de las redes Bayesianas es su potencial para la representación y manejo de la incertidumbre. Esta característica hace de estas redes una herramienta adecuada en el dominio de la estimación de costes software debido al alto grado de incertidumbre asociado al desarrollo de proyectos en este área. En el caso de aprender dichas redes partiendo del juicio experto, se presenta el inconveniente del gran número de parámetros requeridos por éstas.

Este estudio presenta dos técnicas de reducción de la cantidad de estimaciones a proporcionar por el experto. La primera de ellas permite, en base a las características de la distribución de probabilidad Normal, independizar la cantidad de parámetros a estimar por el experto del número de estados, tanto del nodo hijo como de los nodos padre. No obstante, la aplicación de esta técnica implica asumir que tanto las probabilidades marginales de la red como las probabilidades condicionadas tienen forma de distribución Normal. Además, esta técnica requiere de la evaluación de las habilidades del experto con el fin de establecer el grado de fiabilidad que se deposita en las estimaciones proporcionadas por éste.

La otra técnica propuesta consiste en reducir el grado de exhaustividad con que se representa la relación entre la variable objetivo y las variables predictoras. Esta relación se representa mediante las TCP, siendo en estas tablas donde se

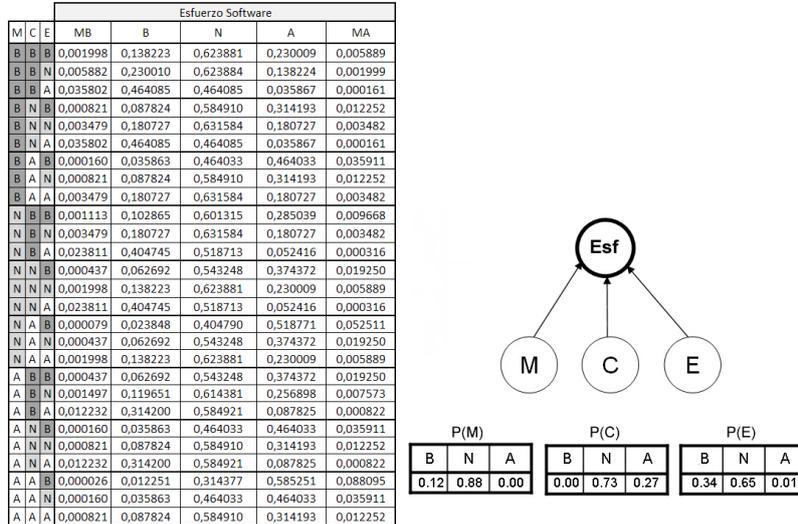


Figura 6. Red Bayesiana con todos los parámetros necesarios.

concentra la mayoría de los parámetros de la red. Esta técnica es la que ofrece un nivel de reducción más significativo, dejando las estimaciones a proporcionar por el experto en una cantidad fija independientemente del número de variables predictoras utilizadas (en este estudio se ha fijado en 5 casos).

Se ha realizado una comparación entre la estimación obtenida aplicando ambas técnicas y únicamente la primera de ellas. Los resultados son similares, lo cual indica la posibilidad de aplicarlas sin que la fiabilidad de la estimación disminuya significativamente. No obstante, la fiabilidad de esta técnica depende tanto de los casos seleccionados como de los datos estimados, por lo que se hace necesario un estudio en profundidad de su viabilidad.

Agradecimientos

Este trabajo se ha desarrollado gracias a la financiación del proyecto UPV/EHU Onproiek EHU08/40.

Referencias

1. Hareton Leung and Zhang Fan. Software cost estimation. *Handbook of Software Engineering and Knowledge Engineering*, World Scientific PUB. CO, River Edge, NJ, 2002.
2. Pablo Javier Tuya González Isabel Ramos Román, José Javier Dolado Cosín. *Técnicas cuantitativas para la gestión en la Ingeniería del software*. Netbiblio, 2007.

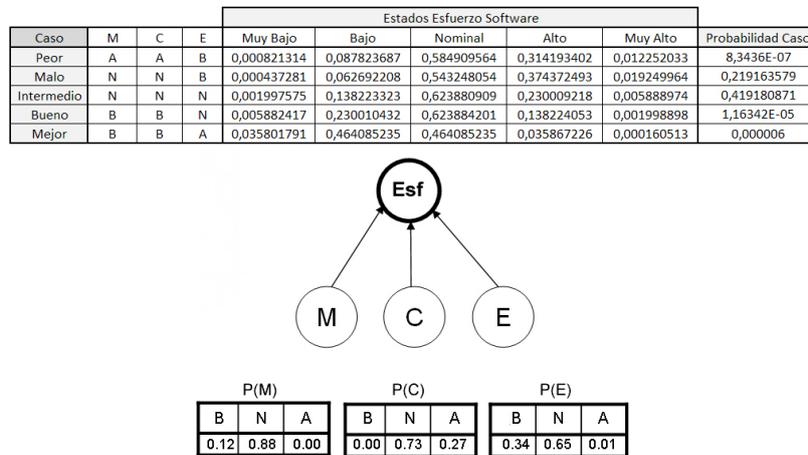


Figura 7. Red Bayesiana con los parámetros reducidos mediante selección de casos.

- J. J. Dolado. On the problem of the software cost function. *Information and Software Technology*, 43(1):61–72, 2001.
- Enrique Castillo, Jose M. Gutierrez, and Ali S. Hadi. *Expert Systems and Probabilistic Network Models*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.
- Norman E. Fenton, Martin Neil, and Jose Galan Caballero. Using ranked nodes to model qualitative judgments in bayesian networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(10):1420–1432, 2007.
- Emilia Mendes. The use of a bayesian network for web effort estimation. In *ICWE 2007, Web Engineering, 7th International Conference, Como, Italy, July 16-20, 2007, Proceedings*, volume 4607 of *Lecture Notes in Computer Science*, pages 90–104. Springer, 2007.
- Sunita Chulani, Barry Boehm, and Bert Steece. Calibrating software cost models using bayesian analysis. *IEEE Transactions on Software Engineering.*, July/August, pages 573–583, 1999.
- Kwai-Sang Chin, Da-Wei Tang, Jian-Bo Yang, Shui Yee Wong, and Hongwei Wang. Assessing new product development project risk by bayesian network with a systematic probability generation methodology. *Expert Systems with Applications*, 36(6):9879 – 9890, 2009.
- Joseba Esteban and Javier Dolado Cosín. Reducción de las probabilidades condicionadas en la estimación del esfuerzo software mediante redes bayesianas. *ADIS - Taller sobre Apoyo a la Decisión en Ingeniería del Software*, 10:35–42, 2010.
- David J. Weiss and James Shanteau. Empirical assessment of expertise. Technical Report 1, Spring 2003.