Diferentes estrategias para aproximar árboles de probabilidad en propagación Penniless*

Andrés Cano, Serafín Moral
Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
Avda. de Andalucía, 38
18071 Granada (España)
{acu,smc}@decsai.ugr.es

Antonio Salmerón
Dpto. Estadística y Matemática Aplicada
Universidad de Almería
La Cañada de San Urbano s/n
04120 Almería (España)
Antonio.Salmeron@ual.es

Resumen

En este trabajo proponemos algunas modificaciones sobre el algoritmo Penniless. Usamos una medida de información mejorada a la hora de calcular el error de las aproximaciones, lo que lleva a una nueva forma de podar multiples valores en un árbol de probabilidad reemplazándolos por uno solo, que se calcula a partir de los valores almacenados en el árbol que se está podando, teniendo en cuenta el mensaje presente en la dirección opuesta. Además, hemos considerado la posibilidad de sustituir valores de probabilidad muy pequeños por ceros. Localmente, esta estrategia de aproximación no es óptima, pero en este problema se dan numerosos pasos de aproximación antes de obtener el valor final, con lo que globalmente puede ser beneficioso. En algunos experimentos se puede comprobar que sustituir por ceros puede mejorar la calidad de las aproximaciones.

1 Introducción

Las redes bayesianas son modelos gráficos que permiten manejar de forma eficiente la incertidumbre en sistemas expertos probabilistas (sistemas expertos donde la incertidumbre se mide en términos de probabilidad). Una *red bayesiana* es un grafo dirigido acíclico donde cada node representa una variable aleatoria y la topología de la red codifica las relaciones de independencia entre las variables, de acuerdo con el criterio de d-separación. Asociada al grafo, hay una distribución de probabilidad para cada nodo, condicionada a sus padres, de tal manera que la distribución conjunta sobre todas las variables de la red se factoriza como el producto de todas esas distribuciones condicionadas.

La tarea de razonamiento, también llamada *propagación de probabilidad*, consiste en la obtención de las marginales a posteriori sobre

*Parcialmente subvencionado por la Junta de Andalucía, grupos de investigación TIC103 y FQM244

algún conjunto de variables de interés dado que se conoce el valor de algunas otras variables. Se han propuesto algoritmos de propagación que que obtienen las distribuciones a posteriori sin necesidad de calcular previamente la distribución conjunta; en su lugar, éstas se obtienen mediante una serie de cálculos locales sobre una estructura auxiliar llamada *árbol de uniones* [4, 9, 7]. Sin embargo, en redes suficientemente complicadas, el uso de estos algoritmos no resulta factible..

Con el objetivo de poder tratas redes de gran tamaño, es posible usar algoritmos aproximados, que proporcionan resultados (quizás aproximados) en un tiempo menor. Algunos de estos métodos están basados en simulación Monte Carlo [8], mientras que otros son de carácter determinista. Una de las contribuciones más recientes dentro del grupo de algoritmos deterministas, es el llamado *Algoritmo de propagación Penniless* [3], que está basado en el algoritmo de propagación sobre árboles de uniones binarios de Shenoy-Shafer [9], con la salvedad de que la información probabilista se representa mediante *árboles de probabilidad* [8]. El uso de árboles de probabilidad permite aproximar grandes potenciales por otros más pequeños, gracias a la posibilidad de podar algunas de las ramas del árbol, haciendo factible la propagación incluso bajo recursos limitados (RAM y CPU).

En este trabajo proponemos algunas modificaciones sobre el algoritmo de propagación Penniless. Usamos una medida de información mejorada a la hora de calcular el error de las aproximaciones. Esta nueva medida sugiere una forma novedosa de podar multiples valores de probabilidad y reemplazarlos por uno solo, obtenido a partir de los valores almacenados en las hojas que se podan y teniendo en cuenta el mensaje almacenado en la dirección contraria en el correspondiente arco del árbol de uniones.

Además, hemos considerado la posibilidad de reemplazar valores de probabilidad pequeños por ceros, con el objetivo de acelerar los cálculos y de controlar la complejidad de los árboles usados para representar los potenciales. Demostraremos un teorema que establece que la mejor aproximación de un potencial, condicionado en otro potencial, se obtiene sustituyendo un nodo tal que sus hijos son hojas, por una media ponderada de los valores en esas hojas. Sin embargo, aquí consideraremos una estrategia diferente: sustituir por cero cuando la suma de los valores en las hojas es muy pequeña. Aunque esta estrategia no es óptima, potencialmente tiene una ventaja. El algoritmo Penniless lleva a cabo muchos pasos de aproximación consecutivos. Cada mensaje se aproxima y luego se usa en ulteriores cálculos (multiplicaciones y marginalizaciones). En estas operaciones, la complejidad de los resultados es, en el peor caso, exponencial en relación a la complejidad de los operandos, siendo la multiplicación la mayor fuente de obtención de potenciales más complicados (el tamaño del dominio se incrementa, mientras que en la marginalización se reduce).

Si aproximamos una rama por un cero en lugar de por un valor positivo, la complejidad de la representación de los potenciales en las partes nulas no se incrementa mediante la multiplicación. El resultado de multiplicar por un cero contenido en una hoja de un árbol que representa un cierto potencial, siempre vale cero para cada valor del otro potencial, y este resultado se puede almacenar usando de nuevo el mismo nodo. De esta manera, aunque la aproximación no es óptima, obtenemos aproximaciones menos complejas, lo que puede simplificar también pasos posteriores, lo que redunda en un beneficio global. Esta afirmación está avalada por los resultados experimentales que presentaremos en este artículo.

Comenzaremos con una breve introducción a la propagación Shenoy-Shafer en la sección 2 y a la propagación Penniless en la sección 3, donde también presentaremos las novedades contenidas en este trabajo, concretamente en la subsección 3.2. Los experimentos que hemos llevado a cabo para contrastar la validez de los nuevos algoritmos se describen en la sección 4, y el artículo finaliza con las conclusiones en la sección 5.

2 Propagación sobre árboles de uniones

A lo largo de este trabajo consideraremos una red bayesiana definida sobre un conjunto de variables $\mathbf{X} = \{X_1, \ldots, X_n\}$, donde cada variable X_i toma valores en un conjunto finito U_i con $|U_i|$ elementos. Si $I \subseteq N = \{1, \ldots n\}$ es un conjunto de índices, denotaremos por $\mathbf{X}_I$ al

conjunto de variables $\{X_i | i \in I\}$, definido sobre $U_I = \times_{i \in I} U_i$. Dado $\mathbf{x} \in U_I$ y $J \subseteq I$, $\mathbf{x}_J$ es el elemento de U_J obtenido a partir de $\mathbf{x}$ eliminando las coordenadas que no estén en J. Dados $\mathbf{x} \in U_J$ y $J \subseteq I$, denotaremos por $A^I_{\mathbf{x}}$ el conjunto de valores $\mathbf{y} \in U_I$ tales que $\mathbf{y}_J = \mathbf{x}$, es decir, el conjunto de elementos de U_I coincidentes con $\mathbf{x}$ en las coordenadas de J. Si ϕ es un potencial[1] definido sobre U_I, $\mathrm{dom}(\phi)$ es el conjunto de índices de las variables para las cuales ϕ está definido (i.e. $\mathrm{dom}(\phi) = I$).

La *marginal* de un potencial ϕ sobre un conjunto de variables $\mathbf{X}_J$ con $J \subseteq I$ se denota por $\phi^{\downarrow J}$ y es una función definida para las variables $\mathbf{X}_J$ como $\phi^{\downarrow J}(\mathbf{y}) = \sum_{\mathbf{x}_J = \mathbf{y}} \phi(\mathbf{x})$ para todo $\mathbf{y} \in U_J$.

La *combinación o producto* de dos potenciales ϕ y ϕ' es un nuevo potencial $\phi \cdot \phi'$ definido para las variables $\mathbf{X}_{\mathrm{dom}(\phi) \cup \mathrm{dom}(\phi')}$ y que se obtiene mediante la multiplicación punto a punto.

La distribución condicionada de cada variable X_i, $i = 1, \ldots, n$, dados sus padres en la red, $\mathbf{X}_{pa(i)}$, se denota por un potencial $p_i(x_i | \mathbf{x}_{pa(i)})$ definido sobre $U_{\{i\} \cup pa(i)}$, y la distribución conjunta de la variable n-dimensional $\mathbf{X}_N$ puede expresarse como

$$p(\mathbf{x}) = \prod_{i \in N} p_i(x_i | \mathbf{x}_{pa(i)}) \quad \forall \mathbf{x} \in U_N \ . \qquad (1)$$

Se denotamos por $\mathbf{e}$ los valores de las variables observadas y por E a sus índices, la propagación de probabilidades puede verse como el cálculo de la distribución a posteriori $p(x'_k | \mathbf{e}) = p(x'_k, \mathbf{e}) / p(\mathbf{e})$, para todo $x'_k \in U_k$, $k \in \{1, \ldots, n\} \setminus E$.

Usando notación potencial, si llamamos H al conjunto de potenciales correspondientes a las distribuciones condicionadas de la red restringidas a las observaciones $\mathbf{e}$, el objetivo de la propagación de probabilidad es obtener, para cada variable de interés X_k,

$$\phi^m_{X_k} = \left(\prod_{\phi \in H} \phi \right)^{\downarrow k} \ , \qquad (2)$$

[1] Un potencial es una función no negativa que representa una distribución condicional, conjunta o marginal

donde el cuperíndice m indica *marginal a posteriori*. Después, la distribución condicionada se consigue normalizando $\phi^m_{X_k}$.

El cálculo de $\phi^m_{X_k}$ se puede organizar en un árbol de uniones, que es un árbol donde cada nodo V es un subconjunto de $\mathbf{X}_N$, y tal que si una variable está en dos nodos distintos, V_1 y V_2, entonces también está en todos los nodos que se encuentren en el camino entre ambos. Un árbol de uniones se dice *binario* si cada nodo no tiene más de dos vecinos. Cada uno de los potenciales iniciales $\phi \in H$, se asigna a un nodo V_j tal que $\mathbf{X}_{\mathrm{dom}(\phi)} \subseteq V_j$. De esta manera, almacenado en cada nodo V_i habrá un potencial ϕ_{V_i} definido sobre el conjunto de variables V_i y que es igual al producto de todos los potenciales asignados previamente a ese nodo. El algoritmo Penniless opera sobre un árbol de uniones binario [3, 9], y se basa en el esquema de propagación de Shenoy-Shafer, que describiremos brevemente a continuación.

El *algoritmo de propagación de Shenoy-Shafer* se lleva a cabo mediante un paso de mensajes en las dos direcciones de cada arco del árbol de uniones. Los *mensajes* entre dos nodos adyacentes V_i y V_j son potenciales definidos sobre $V_i \cap V_j$ (ver [10] para más detalles). El mensaje que sale de V_i y entra en V_j se calcula como

$$\phi_{V_i \to V_j} = \left\{ \phi_{V_i} \cdot \left(\prod_{V_k \in ne(V_i) \setminus \{V_j\}} \phi_{V_k \to V_i} \right) \right\}^{\downarrow V_i \cap V_j} \ , \qquad (3)$$

donde ϕ_{V_i} es el potencial inicial sobre V_i reducido a las observaciones $\mathbf{e}$, $\phi_{V_k \to V_i}$ son los mensajes que van desde V_k hasta V_i y $ne(V_i)$ son los vecinos de V_i.

La propagación se organiza en dos etapas. En la primera, los mensajes se mandan desde las hojas hacia un nodo raíz seleccionado de antemano *propagación ascendente*, y en la segunda fase, los mensajes se mandan desde la raíz hacia las hojas *propagación descendente*. Después de estas dos etapas, de cara a obtener la distribución marginal a posteriori para la variable X_k, primero determinamos un nodo V_i que contenga a X_k y calculamos

$$\phi_{V_i}^m = \phi_{V_i} \cdot (\prod_{V_k \in ne(V_i)} \phi_{V_k \to V_i}) \ .$$

La distribución de X_k dado $\mathbf{e}$ se puede calcular marginalizando $\phi_{V_i}^m$ sobre X_k (obteniendo$\phi_{X_k}^m$) y normalizando a continuación en resultado.

3 Propagación Penniless

El algoritmo de propagación *Penniless* [3] es un método determinista y aproximado basado en el esquema de Shenoy-Shafer, y que tiene como objetivo proporcionar resultados (aproximados) bajo recursos limitados. Una de sus principales características es el uso de *árboles de probabilidad* [1], que permiten representar potenciales de forma aproximada dentro de un límite máximo de tamaño (de nodos hoja) [2, 5, 8].

Dado que el algoritmo Penniless se basa en la arquitectura de Shenoy-Shafer, éste opera sobre árboles de uniones binarios, dado que se ha demostrado que este esquema es máseficiente en este tipo de estructuras [9].

Uno de los rasgos característicos del algoritmo Penniless es que los mensajes que se mandan durante la propagación sufren un proceso de aproximación para reducir su tamaño. Otra diferencia con respecto al algoritmo de Shenoy-Shafer es el número de etapas de la propagación: el Penniless puede llevar a cabo más de dos etapas, en las cuales los mensajes se mejoran gradualmente haciendo uso de la información que fluye por el árbol de uniones. Por lo tanto, la base del algoritmo Penniless es el uso de árboles de probabilidad como representación aproximada de los mensajes, y la mejora incremental de la calidad de las aproximaciones conforme el número de etapas de propagación se incrementa.

3.1 Árboles de probabilidad

Un *árbol de probabilidad* [1, 2, 8] es un árbol dirigido y etiquetado donde cada nodo interior representa una variable aleatoria y cada nodo hoja es un número real no negativo. El número de nodos hoja de un árbol $\mathcal{T}$ es su *tamaño*. De cada nodo interior parten tantos arcos como estados tiene la variable que representa.

Un árbol de probabilidad $\mathcal{T}$ con variables $\mathbf{X}_I$ representa a un potencial ϕ si para cada $\mathbf{x}_I \in U_I$ el valor $\phi(\mathbf{x}_I)$ es el número almacenado en la hoja que se alcanza partiendo del nodo raíz y seleccionando, en cada nodo interior visitado etiquetado con X_i, el arco que se corresponde con el valor x_i contenido en $\mathbf{x}_I$.

Dos importantes cualidades de los árboles de probabilidad es que pueden representar la misma información que una tabla de probabilidad, pero en menos espacio, y que pueden aproximar el árbol original sustituyendo algunos valores por uno solo (ver figura 1).

Las operaciones que tienen lugar durante los algoritmos de propagación (combinación, marginalización y restricción), pueden realizarse directamente sobre árboles de probabilidad (ver [3, 8] para más detalles). En el caso de la propagación Penniless, hay otra operación que es especialmente importante: la aproximación. EN esta operación nos centraremos a continuación.

3.2 Árboles de probabilidad aproximados

Aproximar un árbol $\mathcal{T}_1$ que representa a un potencial ϕ quiere decir obtener un árbol $\mathcal{T}$ más pequeño que $\mathcal{T}_1$, pero tratando de mantener una representación ajustada del potencial ϕ. Una manera de obtener el árbol aproximado es mediante la poda del árbol original. Una *poda* de un árbol de probabilidad consiste en seleccionar un nodo tal que todos sus hijos son hojas y reemplazar ese nodo y sus hijos por un solo nodo que contenga un número. En general, el valor óptimo que se situaría en ese nodo es la media de los valores contenidos en las hojas que se podan (esto minimiza la distancia de Kullback-Leibler [6] entre el árbol original y el aproximado [2, 8]).

Sin embargo, en el esquema de propagación Penniless, los árboles que representan los mensajes a través de los arcos se aproximan teniendo en cuenta el mensaje (un árbol de probabilidad) que pasa por el mismo arco pero en dirección contraria. Concretamente, el objetivo es aproximar un potencial ϕ representado por

x_1	x_2	x_3	$\phi(x_1, x_2, x_3)$
1	1	1	0.2
1	1	2	0.5
1	2	1	0.7
1	2	2	0.7
2	1	1	0.3
2	1	2	0.5
2	2	1	0.3
2	2	2	0.3

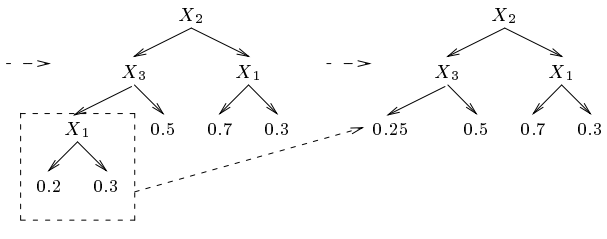

Figura 1: Un potencial, un árbol de probabilidad que lo representa y una aproximación del árbol original. Los arcos que salen de cada variable, de izquierda a derecha, se corresponden con los valores de dicha variable en orden lexicográfico.

un árbol $\mathcal{T}$, por otro potencial ϕ' representado por otro árbol $\mathcal{T}'$ de menor tamaño, condicionado a otro potencial ψ. En toda esta sección consideraremos todos los potenciales definidos sobre un conjunto U_I. Dado un potencial ϕ, usaremos la siguiente notación:

- $\text{sum}(\phi|A) = \sum_{\mathbf{x}\in A} \phi(\mathbf{x})$, donde $A \subseteq U_I$.
- $\text{sum}(\phi) = \text{sum}(\phi|U_I) = \sum_{\mathbf{x}\in U_I} \phi(\mathbf{x})$.
- Si $\text{sum}(\phi) \neq 0$, $N(\phi) = \phi/\text{sum}(\phi)$.

Mediremos la distancia entre dos potenciales ϕ y ϕ' condicionada a ψ como la divergencia de Kullback-Leibler entre los potenciales normalizados como sigue:

$$D(\phi, \phi'|\psi) = \sum_{\mathbf{x}\in U_I} N(\phi(\mathbf{x})\psi(\mathbf{x})) \log\left(\frac{\phi(\mathbf{x})\text{sum}(\phi'\cdot\psi)}{\phi'(\mathbf{x})\text{sum}(\phi\cdot\psi)}\right) . \quad (4)$$

Dado que no hay diferencia entre las distancias $D(\phi, \phi'|\psi)$ y $D(\phi, \phi''|\psi)$ cuando $N(\phi') = N(\phi'')$, es decir, la distancia es independiente del factor de normalización, entonces ϕ' se determinará salvo un factor de normalización. En [3] se asumía que ϕ' y ϕ eran tales que $\text{sum}(\phi') = \text{sum}(\phi)$, pero aquí supondremos que $\text{sum}(\phi' \cdot \psi) = \text{sum}(\phi \cdot \psi)$. La selección del factor de normalización no tiene ningún efecto en la calidad de la aproximación, pero bajo esta suposición los resultados son más simples de expresar y de demostrar.

Tal y como se ponía de manifiesto en [1, 2], la dificultad de la aproximación radica en encontrar la *estructura* del árbol, es decir, el árbol sin números en las hojas. En [3] se suponía que dada una estructura $\mathcal{S}$ se podía construir un árbol aproximado denotado por $\mathcal{T}_\mathcal{S}$ a partir de ϕ asignando a cada hoja caracterizada por la configuración $\mathbf{X}_J = \mathbf{x}_J$, la media del potencial ϕ en los puntos de $A^I_{\mathbf{x}_J}$ (los puntos de U_I para los cuales $\mathbf{X}_J = \mathbf{x}_J$). Sin embargo, esta estrategia no es óptima.; es apropiada cuando no tenemos un potencial condicionante, ψ, o cuando ese potencial es igual a 1, pero no en el caso general. El problema se puede formular de la siguiente manera: tenemos un potencial ϕ definido sobre U_I y una partición $\mathcal{A}$ de este referencial. Queremos encontrar un potencial ϕ' constante en cada conjunto $A \in \mathcal{A}$ y tal que la distancia de ϕ a ϕ' condicionada a ψ sea mínima. En este caso, dada una estructura $\mathcal{S}$ los elementos de la partición vienen definidos por las hojas de la estructura. Si una hoja se caracteriza por una configuración $\mathbf{X}_J = \mathbf{x}_J$, entonces dicha hoja define un conjunto $A = A^I_{\mathbf{x}_J}$. En estas condiciones se puede demostrar el siguiente resultado, que muestra que ahora la estrategia óptima es asignar a los elementos de A la media de ϕ ponderada por los valores de ψ.

Teorema 1 *Si ϕ es un potencial definido sobre U_I y $\mathcal{A}$ es una partición de U_I, entonces el potencial ϕ' que es constante en los elementos de cada conjunto $A \in \mathcal{A}$, con* $\text{sum}(\phi \cdot \psi) = \text{sum}(\phi' \cdot \psi)$ *y que minimiza la distancia (4) de ϕ a ϕ' dado ψ es aquel potencial ϕ' que asigna a cada elemento $\mathbf{x} \in A$ el valor*

$$\phi'(\mathbf{x}) = \frac{\text{sum}(\phi \cdot \psi|A)}{\text{sum}(\psi|A)} , \quad (5)$$

y que es igual a ϕ en el resto de los puntos de U_I.

Demostración: Llamemos $\phi'(A)$ al valor constante de ϕ' para los elementos de A. Tenemos que,

$$D(\phi,\phi'|\psi) = \sum_{\mathbf{x}\in U_I} N(\phi(\mathbf{x})\psi(\mathbf{x}))\log\left(\frac{\phi(\mathbf{x})\text{sum}(\phi'\cdot\psi)}{\phi'(\mathbf{x})\text{sum}(\phi\cdot\psi)}\right) = \sum_{\mathbf{x}\in U_I}\frac{\phi(\mathbf{x})\psi(\mathbf{x})}{\text{sum}(\phi\cdot\psi)}\log\left(\frac{\phi(\mathbf{x})}{\phi'(\mathbf{x})}\right) = \frac{1}{\text{sum}(\phi\cdot\psi)}\sum_{A\in\mathcal{A}}\sum_{\mathbf{x}\in A}\phi(\mathbf{x})\psi(\mathbf{x})\log\left(\frac{\phi(\mathbf{x})}{\phi'(A)}\right) = \frac{1}{\text{sum}(\phi\cdot\psi)}\sum_{A\in\mathcal{A}}\left(\sum_{\mathbf{x}\in A}\phi(\mathbf{x})\psi(\mathbf{x})\log(\phi(\mathbf{x}))- (\sum_{\mathbf{x}\in A}\phi(\mathbf{x})\psi(\mathbf{x}))\log(\phi'(A))\right) \ .$$

Sin tener en cuenta las partes constantes que no dependan de ϕ', se tiene que minimizar la expresión anterior es lo mismo que maximizar

$$\sum_{A\in\mathcal{A}}\left(\sum_{\mathbf{x}\in A}\phi(\mathbf{x})\psi(\mathbf{x})\right)\log(\phi'(A)) = \sum_{A\in\mathcal{A}}\text{sum}(\phi\cdot\psi|A)\log(\phi'(A)) \ .$$

Sumando la constante

$$\sum_{A\in\mathcal{A}}\text{sum}(\phi\cdot\psi|A)\log(\text{sum}(\psi|A)) \ ,$$

que no depende de ϕ', vemos que tenemos que maximizar

$$\sum_{A\in\mathcal{A}}\text{sum}(\phi\cdot\psi|A)\log(\phi'(A)\cdot\text{sum}(\psi|A)) \ . \quad (6)$$

Dado que ϕ' es constante en A,

$$\text{sum}(\phi'\cdot\psi) = \sum_{A\in\mathcal{A}}\phi'(A)\text{sum}(\psi|A) \ ,$$

y como

$$\sum_{A\in\mathcal{A}}\text{sum}(\phi\cdot\psi|A) = \text{sum}(\phi\cdot\psi) \ ,$$

y

$$\text{sum}(\phi\cdot\psi) = \text{sum}(\phi'\cdot\psi) \ ,$$

tenemos que

$$\sum_{A\in\mathcal{A}}\text{sum}(\phi\cdot\psi|A) = \sum_{A\in\mathcal{A}}\phi'(A)\text{sum}(\psi|A) \ ,$$

y, por el lema de Gibbs,

$$\sum_{A\in\mathcal{A}}\text{sum}(\phi\cdot\psi|A)\log(\phi'(A)\cdot\text{sum}(\psi|A)) \leq \sum_{A\in\mathcal{A}}\text{sum}(\phi\cdot\psi|A)\log(\text{sum}(\phi\cdot\psi|A)) \ ,$$

lo que quiere decir que la expresión (6) se maximiza para

$$\phi'(A) = \text{sum}(\phi\cdot\psi|A)/\text{sum}(\psi|A) \ ,$$

y por lo tanto la distancia se minimiza en ese punto. □

Si tenemos una estructura $\mathcal{S}'$, $\mathcal{S}''$ es la estructura que se obtiene podando $\mathcal{S}'$, y ϕ' y ϕ'' son los potenciales asociados a los árboles $\mathcal{T}_{\mathcal{S}'}$ y $\mathcal{T}_{\mathcal{S}''}$ respectivamente, entonces la poda se lleva a cabo intentando minimizar $D(\phi',\phi''|\psi)$.

Esto conlleva el cálculo de la distancia de Kullback-Leibler de ϕ a ϕ' dado un tercer potencial ψ. El valor $\phi'(\mathbf{x})$ es igual a $\phi(\mathbf{x})$ en todos los puntos de U_I, excepto para un subconjunto $A \subseteq U_I$ en el cual

$$\phi'(\mathbf{x}) = \frac{\text{sum}(\phi\cdot\psi|A)}{\text{sum}(\psi|A)} \ .$$

En este caso, el conjunto A se corresponde con todos los valores $\mathbf{x}_I \in U_I$ tales que siguiendo el camino correspondiente se llega a el nodo resultante de la poda. Haciendo algunos cálculos sencillos, se observa que $D(\phi,\phi'|\psi)$ puede obtenerse de acuerdo con la siguiente fórmula:

$$\frac{1}{\text{sum}(\phi\cdot\psi|A)}\left(\left(\sum_{\mathbf{x}\in A}\phi(\mathbf{x})\psi(\mathbf{x})\log(\phi(\mathbf{x}))\right) + \text{sum}(\phi\cdot\psi|A) + \log\left(\frac{\text{sum}(\psi|A)}{\text{sum}(\phi\cdot\psi|A)}\right)\right) \ . \quad (7)$$

Esta fórmula es mucho más sencilla que la usada en el esquema Penniless original:

Tabla 1: Resultados (tiempo en segundos y divergencia K-L) de `penni` y `new-penni` para una red de pedigree con numerosos nodos observados.

Δ	4 etapas		5 etapas		6 etapas	
	`penni`	`new-penni`	`penni`	`new-penni`	`penni`	`new-penni`
0.01	1.16 - 0.004434	1.11 - 0.004366	1.18 - 0.003754	1.14 - 0.003609	1.24 - 0.003739	1.17 - 0.003575
0.005	1.20 - 0.001004	1.14 - 0.003376	1.21 - 0.001004	1.20 - 0.002932	1.24 - 0.001004	1.18 - 0.001162
0.001	1.22 - 1.505E-4	1.18 - 4.375E-5	1.22 - 1.505E-4	1.17 - 4.816E-5	1.24 - 1.505E-4	1.17 - 4.375E-5
0.0005	1.23 - 4.042E-5	1.19 - 1.978E-5	1.25 - 4.042E-5	1.19 - 1.883E-5	1.28 - 4.042E-5	1.20 - 1.978E-5

Tabla 2: Resultados (tiempo en segundos y divergencia K-L) en el segundo experimento con la red de pedigree.

ϵ	4 etapas		5 etapas		6 etapas	
	`new-penni-av`	`new-penni-ze`	`new-penni-av`	`new-penni-ze`	`new-penni-av`	`new-penni-ze`
0.01	1.22 - 1.392E-4	1.16 - 1.742E-4	1.18 - 1.383E-4	1.15 - 1.744E-4	1.18 - 1.392E-4	1.16 - 1.742E-4
0.006	1.18 - 6.391E-5	1.17 - 2.862E-4	1.17 - 6.310E-5	1.16 - 2.715E-4	1.18 - 6.391E-5	1.18 - 2.862E-4
0.003	1.20 - 2.959E-5	1.24 - 3.015E-5	1.18 - 2.550E-5	1.19 - 2.749E-5	1.19 - 2.959E-5	1.22 - 3.015E-5
0.001	1.20 - 1.978E-5	1.20 - 1.914E-5	1.19 - 2.017E-5	1.18 - 1.813E-5	1.20 - 1.978E-5	1.20 - 1.914E-5

$$(\sum_{\mathbf{x}\in A}(\phi(\mathbf{x})\psi(\mathbf{x})\log(\phi(\mathbf{x})))- \\ \mathrm{sum}(\phi\cdot\psi|A)\log\left(\mathrm{sum}(\phi|A)/|A|\right))/\mathrm{sum}(\phi\cdot\psi)+ \\ \log\left(\frac{\mathrm{sum}(\phi\cdot\psi)-\mathrm{sum}(\phi\cdot\psi|A)+(\mathrm{sum}(\phi|A)\mathrm{sum}(\psi|A))/|A|}{\mathrm{sum}(\phi\cdot\psi)}\right) . \quad (8)$$

Sea ϕ un potencial representado por un árbol $\mathcal{T}$ y supongamos que queremos obtener una aproximación condicionada en los valores de otro potencial ψ. Consideremos un nodo de un árbol tal que todos sus hijos son hojas. Sea X_k la variable almacenada en ese nodo y $(\mathbf{X}_J = \mathbf{x}_J)$ la configuración de valores que definen el camino desde la raíz hasta dicho nodo. Proponemos a continuación diferentes maneras de llevar a cabo la poda de un árbol de probabilidad:

1. Considerar un umbral $\Delta > 0$ y aproximar los hijos de X_k por su media si el valor de la fórmula (8), con $A = A^I_{\mathbf{x}_J}$, es menor que Δ. Esto coincide con el algoritmo Penniless original, denotado como `penni`. También hemos considerado el mismo esquema pero sustituyendo por la suma ponderada de la expresión (5) con $A = A^I_{\mathbf{x}_J}$, en lugar de reemplazar simplemente por la media. Denotaremos este algoritmo como `new-penni`.

2. Considerar un valor $0 < \epsilon < 1$ y podar un nodo X_k si $\mathrm{sum}(\phi\cdot\psi|A^I_{\mathbf{x}_J}) \leq \epsilon\cdot\mathrm{sum}(\phi\cdot\psi)$, i.e. podamos todo nodo tal que, por debajo de él, la proporción del total de masa de probabilidad del producto de los potenciales es menor que ϵ. Aquí hemos distinguido dos posibilidades: reemplazar los valores borrados por la suma ponderada en (5), denotado como `new-penni-av`, o reemplazarlos por un cero, en cuyo caso el algoritmo será denotado por `new-penni-ze`. El objetivo de esta forma de poda es el evitar invertir demasiado esfuerzo en manejar valores poco significativos. En cierto sentido, sustituir por ceros está inspirado por los algoritmos de simulación, en los cuales las configuraciones con probabilidad muy baja en la práctica es como si tuvieran probabilidad nula, pues tienen tendencia a no aparecer en ninguna muestra.

Aunque los criterios anteriores están expresados en términos de potenciales, los cálculos pueden llevarse a cabo directamente sobre sus representaciones mediante árboles, siendo la complejidad de las operaciones una función de las estructuras de los árboles y no de los tamaños de los referenciales sobre los cuales están definidos los potenciales.

Los pasos de aproximación se hacen de forma recursiva, comenzando en los nodos cuyos hijos son hojas y retrocediendo hacia la raíz. De esta manera, si todos los hijos de un nodo interior son hojas o han sido previamente podados y sustituidos por un número, ese nodo es considerado también para la poda.

4 Evaluación experimental

Hemos llevado a cabo dos experimentos. En el primero de ellos se trata de contrastar la conveniencia de la nueva forma de calcular la divergencia entre un potencial exacto y una apro-

Tabla 3: Resultados (tiempo en segundos y divergencia K-L) en el segundo experimento para la red Munin 1.

ϵ	4 etapas		5 etapas		6 etapas	
	`new-penni-av`	`new-penni-ze`	`new-penni-av`	`new-penni-ze`	`new-penni-av`	`new-penni-ze`
0.01	202.8 - 0.16978	102.1 - 0.1942	204.2 - 0.1782	107.0 - 0.1933	284.4 - 0.1450	139.1 - 0.1766
0.006	310.4 - 0.1232	167.8 - 0.0857	274.6 - 0.1526	178.5 - 0.0889	403.5 - 0.1390	236.8 - 0.0954
0.003	440.6 - 0.0849	236.8 - 0.0606	444.4 - 0.0851	233.8 - 0.0647	650.5 - 0.0851	312.4 - 0.0888
0.001	2436.8 - 0.0375	639.7 - 0.0136	2418.5 - 0.0376	646.4 - 0.0135	3777.0 - 0.0375	920.3 - 0.0664

ximación suya introducida en la fórmula (7); en definitiva, se trata de comparar `penni` con `new-penni`.

El segundo experimento está diseñado para comparar los efectos de reemplazar pequeños valores de probabilidad bien por ceros o bien por la suma ponderada de la ecuación (5), es decir, comparar `new-penni-av` con `new-penni-ze`.

En el primer experimento hemos elegido una red de pedigree de gran tamaño (441 variables), con un número considerable de observaciones (166). La razón de esta elección es que en redes donde no haya muchas observaciones, reemplazar por la media o por cero no marca grandes diferencias, debido a la forma en que opera el algoritmo Penniless, donde muchos mensajes hacia arriba pueden ser constantes e iguales a 1, en cuyo caso ambos esquemas de sustitución son equivalentes. Hemos llevado a cabo pruebas con 4, 5 y 6 etapas de propagación, y la precisión de las aproximaciones la hemos controlado mediante el parámetro Δ. Los resultados de este experimento se muestran en la tabla 1. Estos resultados sugieren que `new-penni`, en general, proporciona iguales o mejores resultados para la red de prueba, y además en un tiempo menor.

Para el segundo experimento hemos elegido la misma red que en el experimento anterior, además de otra red con menos variables (189 con 8 observaciones) pero con una mayor complejidad (tamaños de potencial más elevados). Esta red se denomina Munin 1. Ambas redes han sido proporcionadas por el grupo de sistemas de ayuda a la decisión de la Universidad de Aalborg (Dinamarca).

En el segundo experimento hemos tomado un valor fijo de $\Delta = 0.001$, y hemos probado los algoritmos variando el valor del parámetro ϵ. Los resultados de este experimento se muestran en las tablas 2 y 3. De acuerdo con estos resultados, el uso de `new-penni-ze` no parece proporcionar ninguna mejora, pero en el caso de la complicada red Munin 1, el tiempo de cómputo es considerablemente reducido, y la calidad de las aproximaciones mejora. Hemos realizado otros muchos experimentos no reflejados aquí, que parecen apoyar la misma conclusión.

Los algoritmos han sido implementados en Java 2 versión 1.3, y se han integrado en el sistema Elvira.

Todas las pruebas han sido llevadas a cabo en un ordenador con procesador AMD K7 (800 MHz), equipado con 512MB de memoria RAM y sistema operativo Linux RedHat con kernel 2.2.16.22.

5 Conclusiones

En este trabajo hemos introducido nuevos enfoques para la realización de aproximaciones en la propagación Penniless. Bajo las suposiciones del teorema 1, hemos probado que reemplazar las hojas podadas en un árbol de probabilidad por la suma ponderada de la ecuación (5) es una estrategia óptima.

Además, hemos visto que sustituir pequeños valores de probabilidad por ceros permite al esquema Penniless adoptar dos buenas cualidades propias de los algoritmos de Monte-Carlo: velocidad y bajos requisitos de memoria. En algunos casos, esto lleva a la obtención de aproximaciones de mayor calidad.

Los métodos presentados en este trabajo aun pueden ser refinados. Por ejemplo, pensamos perfeccionar el método de triangulación por el cual se obtiene el árbol de uniones sobre el que se lleva a cabo la propagación.

Referencias

[1] J. Boutilier, N. Friedman, M. Goldszmidt, y D. Koller. Context-specific independence in Bayesian networks. En E. Horvitz y F.V. Jensen, eds., *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pp. 115–123. Morgan & Kauffman, 1996.

[2] A. Cano y S. Moral. Propagación exacta y aproximada con árboles de probabilidad. En *Actas de la VII Conferencia de la Asociación Española para la Inteligencia Artificial*, pp. 635–644, 1997.

[3] A. Cano, S. Moral, y A. Salmerón. Penniless propagation in join trees. *International Journal of Intelligent Systems*, 15:1027–1059, 2000.

[4] F.V. Jensen, S.L. Lauritzen, y K.G. Olesen. Bayesian updating in causal probabilistic networks by local computation. *Computational Statistics Quarterly*, 4:269–282, 1990.

[5] A.V. Kozlov. *Efficient inference in Bayesian networks*. Tesis doctoral, Stanford University, 1998.

[6] S. Kullback y R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:76–86, 1951.

[7] A.L. Madsen y F.V. Jensen. Lazy propagation: a junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, 113:203–245, 1999.

[8] A. Salmerón, A. Cano, y S. Moral. Importance sampling in Bayesian networks using probability trees. *Computational Statistics and Data Analysis*, 34:387–413, 2000.

[9] P.P. Shenoy. Binary join trees for computing marginals in the Shenoy-Shafer architecture. *International Journal of Approximate Reasoning*, 17:239–263, 1997.

[10] P.P. Shenoy y G. Shafer. Axioms for probability and belief function propagation. En R.D. Shachter, T.S. Levitt, J.F. Lemmer, y L.N. Kanal, (eds.), *Uncertainty in Artificial Intelligence 4*, pp. 169–198. North Holland, Amsterdam, 1990.