

Método Híbrido para Inferencia en Redes Bayesianas

SALMERÓN, A.

Dpto. Estadística y Matemática Aplicada
Facultad de Ciencias Experimentales
Universidad de Almería
e-mail: asc@stat.ualm.es

BOLAÑOS, M.J.

Dpto. Estadística e I.O.
Facultad de Ciencias
Universidad de Granada
e-mail: mjbc@robinson.ugr.es

Resumen

It is known that probabilistic inference in Bayesian Networks by exact methods is a NP-hard problem. In this paper, we propose a method for combining exact and Monte Carlo techniques, in order to improve the performance of approximate algorithms without a high computational cost.

Palabras clave: Bayesian networks, probability propagation, hybrid propagation.

Clasificación AMS: 68T30.

1 Introducción

El problema de la inferencia en redes Bayesianas se plantea como la obtención de la distribución a posteriori de ciertas variables de la red dado que se ha observado el valor que toman otras variables. Esto es lo que se conoce como "propagación de probabilidades". Diversos métodos de propagación, tanto exactos [7, 9, 10] como basados en técnicas de Monte Carlo [1, 4, 5], han sido desarrollados. Sin embargo, se ha demostrado que éste es un problema NP-duro [2] y por lo tanto intratable a nivel computacional para redes de tamaño suficientemente grande. Las técnicas de Monte Carlo permiten trabajar con redes de mayor tamaño a cambio de perder la exactitud de los cálculos, aunque siguen

teniendo complejidad exponencial [3]. En este trabajo proponemos un método de pre-proceso de la red mediante métodos exactos antes de aplicar una técnica de Monte Carlo, con lo cual esperamos reducir el error en los resultados.

En la sección 2 se expone el desarrollo del método propuesto. A continuación (sección 3) se detalla el algoritmo de propagación. Los resultados experimentales pueden verse en la sección 4, finalizando con las conclusiones en la sección 5.

2 Desarrollo del Método

Una Red Bayesiana (figura 1) es un grafo dirigido acíclico en el cual cada nodo representa una variable aleatoria y la existencia de un arco entre dos variables indica dependencia causal entre dichas variables. Para cada variable de la red se tiene una distribución de probabilidad para esa variable dados sus padres en la red.

Dada una red Bayesiana asociada a las variables $\{X_1, \dots, X_n\}$ el objetivo es calcular la distribución 'a posteriori' $P(X_i|X_E)$ con $i \in I \subset \{1, \dots, n\}$, $I \cap E = \phi$ de un conjunto de variables no observadas, dado el conjunto de variables observadas X_E . Puesto que sólo nos interesa la distribución de las variables X_I , habrá ciertos nodos que podrán ser eliminados, permitiendo esto contraer la red antes de aplicar la técnica de Monte Carlo. En concreto, nuestro método consiste en contraer el grafo eliminando las zonas del mismo que sean "flecos" o "como cadenas" (ver figura 1) mediante la técnica de inversión de arcos propuesta por Shachter [8], para aplicar un método de Monte Carlo en el resto de la red.

El proceso de eliminación de una variable X_i se lleva a cabo en dos pasos:

1. Invertir todos los arcos que parten de X_i .
2. Eliminar el nodo X_i y todos los arcos incidentes en el.

Si se tiene un arco desde X_i a X_j , después de invertirlo las nuevas distribuciones de probabilidad, P^* , para ambas variables vienen dadas por las siguientes expresiones (ver [8]):

$$P^*(X_j|X_{F(X_i) \cup F(X_j) - \{i\}}) = \sum_{X_i} P(X_j|X_{F(X_j)}) P(X_i|X_{F(X_i)}) \quad (1)$$

$$P^*(X_i|X_{\{j\} \cup F(X_i) \cup F(X_j) - \{i\}}) = \frac{P(X_j|X_{F(X_j)}) P(X_i|X_{F(X_i)})}{P^*(X_j|X_{F(X_i) \cup F(X_j) - \{i\}})} \quad (2)$$

Nótese que para el preproceso de la red sólo se tiene en cuenta la estructura. Esto permite hacer la contracción de la red de manera rápida, ya que no se entra en consideraciones acerca del tamaño de las nuevas distribuciones de probabilidad que se generan. A cambio de la rapidez de los cálculos, algunas situaciones en que se podría simplificar la red aún más, puede que no sean consideradas.

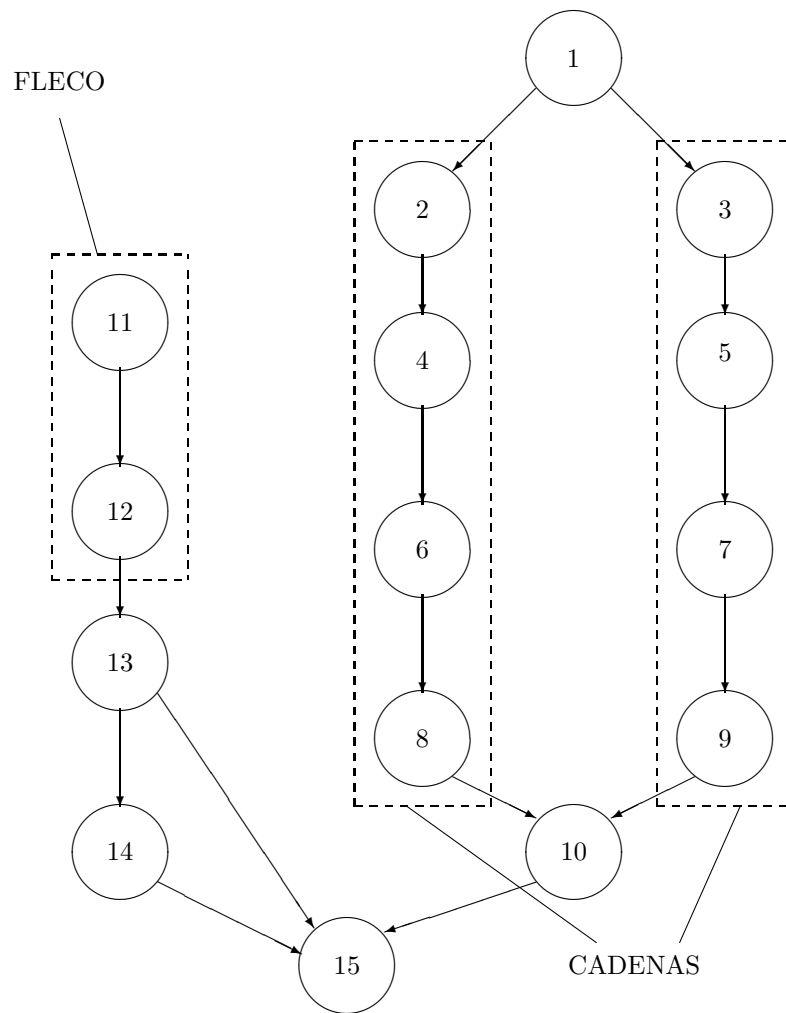


Figura 1: Una red Bayesiana.

Denotaremos por $A(X_j)$ al conjunto de ascendientes de X_j , y por $F(X_j)$ al conjunto de los padres, por X_I a las variables de interés y por X_E a las variables observadas. Diremos que una variable X_j cumple la condición para ser eliminada si verifica que $X_j \notin A(X_I \cup X_E)$.

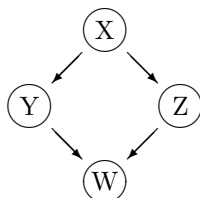


Figura 2: Una red Bayesiana.

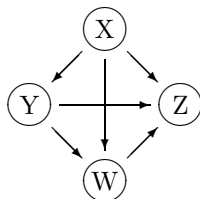


Figura 3: La red de la figura 2 después de invertir el arco entre Z y W

En las figuras (2) y (3) puede verse el cambio que se produce en la red después de invertir un arco.

3 El algoritmo

En esta sección presentamos el algoritmo que recoge la idea expuesta anteriormente. El primer paso consiste en eliminar directamente las variables que no intervienen en el proceso de inferencia (ver [8]). A continuación se comprueba si hay alguna variable que puede ser eliminada de forma rápida, es decir, aquellas que no añadan más complejidad a la red, en el sentido de aumentar el número de arcos que inciden en un nodo.

Obsérvese en la figura (1) que las variables más fáciles de eliminar son precisamente aquellas que están en "flecós" o "cadenas".

El algoritmo detallado es el siguiente:

ALGORITMO DE PROPAGACIÓN

1. Eliminar toda variable X_j que cumpla la condición para ser eliminada.
2. Mientras se produzcan cambios en la red hacer:
Para cada variable $X_j \notin X_I \cup X_E$ que cumpla que, o bien X_j no tiene padres y tiene 1 ó 2 hijos y éstos tienen como único padre a X_j , o bien X_j tiene un único padre y un único hijo (con X_j como único padre), o no tiene padres y un solo hijo hacer:
 - (a) Invertir los arcos desde X_j a sus hijos, actualizando las probabilidades de éstos según la fórmula (1). No es necesario calcular la distribución para X_j cuando se invierte el último arco, pues ésta va a ser eliminada.
 - (b) Eliminar el nodo X_j .
3. Aplicar cualquier método basado en técnicas de Monte Carlo a la red resultante.

4 Experimentación y Resultados

Se ha comparado el método anteriormente expuesto con los algoritmos de muestreo lógico probabilístico [5] y evidence weighting [4], en redes de tres tipos: un árbol (G1), una red poco conectada con flecos y cadenas (G2) y una red muy conectada sin flecos (G3), todas ellas contenían entre 20 y 25 variables con 2,3 ó 4 valores cada una.

Para la experimentación se han considerado tres variables objetivo y dos variables instanciadas. Se ha probado cada algoritmo con 2.000, 5.000 y 10.000 iteraciones de simulación, y se ha calculado el error medio.

Se observa que se reduce significativamente el error en el caso 2, y en menor medida en el caso 3. En la red de tipo 1 los errores no disminuyen, pero sí lo hace de forma considerable el tiempo de ejecución del algoritmo. Los resultados experimentales pueden verse en la tabla (1), donde las leyendas tienen el siguiente significado:

- H : Algoritmo del Muestreo Lógico Probabilístico.
- HP : El anterior con preproceso.
- E : Algoritmo Evidence Weighting.
- EP : El anterior con preproceso.

	H	HP	E	EP
G1	0.015462	0.023268	0.007782	0.014059
G2	0.043093	0.022659	0.013476	0.011784
G3	0.013576	0.013033	0.005916	0.005687

Tabla 1: Resultados experimentales.

5 Conclusiones

A la luz de los resultados obtenidos, se pueden extraer las siguientes conclusiones:

- Reducción en general del tiempo de ejecución.
- Reducción de los errores en general excepto para el caso más sencillo (un árbol).
- Rapidez de la operación de preproceso, al considerar sólo cuestiones de estructura de la red.
- En determinados casos en que no funcionan los algoritmos aproximados, el preproceso puede eliminar las variables problemáticas. Por ejemplo, en la figura (1), si las variables 2 y 3 están instanciadas y la probabilidad de que se de el valor observado es cero para todas las configuraciones de la variable 1 excepto para una, el algoritmo evidence weighting puede fallar, en el sentido de pesar todas las simulaciones con peso cero. El preproceso puede eliminar la variable 1 antes de realizar la simulación, con lo que este problema se evitaría.
- Este método puede servir de base para la obtención de las funciones de muestreo de cara al desarrollo de un algoritmo de muestreo por importancia.

Referencias

- [1] Cano J.E., Hernández L.D., Moral S. (1995) *Importance sampling algorithms for belief networks*. Submitted to International Journal of Approximate Reasoning.
- [2] Cooper G.F. (1990) *The computational complexity of probabilistic inference using Bayesian belief networks*. Artificial Intelligence 42, 393-405.
- [3] Dagum P., Luby M. (1993) *Approximating probabilistic inference in Bayesian networks is NP-hard*. Artificial Intelligence 60 (1), 141-153.

- [4] Fung, R. Chang, K.C. Weighting and integrating evidence for stochastic simulation in Bayesian networks. *Uncertainty in Artificial Intelligence* (5):209-219. Elsevier science publisher B.V. (North Holland), 1990.
- [5] Henrion, M. Propagation of uncertainty by probabilistic logic sampling in Bayes networks. *Uncertainty in Artificial Intelligence* (2),pp: 149-164. Elsevier Science Publisher B.V. (North Holland), 1988.
- [6] Hernández, L.D. Diseño y validación de nuevos algoritmos para el tratamiento de grafos de dependencias. *Tesis Doctoral*. Universidad de Granada. 1995.
- [7] 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, B 50, 157-224.
- [8] Shachter, R.D. Probabilistic inference and influence diagrams. *Op. Research*, 4(36):589-604, 1988.
- [9] Shachter R.D., Andersen S.K., Szolovits P. (1991) *The equivalence of exact methods for probabilistic inference on belief networks*. Submitted to Artificial Intelligence.
- [10] Shafer G., Shenoy P.P. (1990) *Probability propagation*. Annals of Mathematical and Artificial Intelligence 2, 327-351.