

Aproximación de Redes Causales mediante Poliárboles *

Luis M. de Campos, Juan F. Huete
Dpto. Ciencias de la Computación e I.A.
Facultad de Ciencias
Universidad de Granada
18071 - Granada

Resumen

Las redes causales nos van a permitir modelar nuestro conocimiento en base a relaciones del tipo *Causa-Efecto*. Una estructura sin ciclos, va a permitir que los algoritmos de propagación sean bastantes eficientes. Sin embargo, la presencia de ciclos es común en este tipo de redes. En este trabajo, se presentan dos algoritmos, el primero permite aprender una estructura de poliárbol que refleja fielmente un modelo de dependencias y el segundo, nos permite aproximar cualquier modelo por un poliárbol.

1 Introducción

A la hora de realizar cualquier razonamiento, el distinguir qué variables son causa y cuáles son efecto, nos permitirá modelar determinados problemas. Este tipo de relaciones *Causa-Efecto* pueden identificarse a través de estructuras gráficas donde las variables están representadas por nodos y cuando una variable A es causa directa de otra variable B en el problema, existirá un arco entre A y B en nuestra estructura, esto es $A \rightarrow B$. En este tipo de estructuras gráficas no se permite la presencia de ciclos dirigidos, representándose a través de DAG (Grafos Dirigidos Acíclicos). Sin embargo, la presencia de un ciclo no dirigido es permitida y provoca que los algoritmos creados para trabajar con ellas sean poco eficientes.

La dirección de los arcos dentro de una estructura permite determinar relaciones de independencia existentes entre las variables del modelo. Una relación de independencia la notaremos por $I(\dots)$. En una representación gráfica de un modelo simple, podemos encontrar variables x con arcos cola a cola, $(\alpha \leftarrow x \rightarrow \beta)$ o variables x con arcos cabeza a cola, $(\alpha \rightarrow x \rightarrow \beta)$ que representan una relación de dependencia marginal entre α y β , notada por $I(\alpha, \beta) = \text{False}$. Estas variables se hacen independientes cuando x es conocida y por tanto se verifica una relación de independencia condicional, $I(\alpha \mid x \mid \beta) = \text{True}$. Por otra parte, también podremos encontrar variables x donde los arcos son cabeza a cabeza, $(\alpha \rightarrow x \leftarrow \beta)$, representando una relación de independencia marginal entre α y β

*Este trabajo ha sido subvencionado por el proyecto DGICYT número PB92-0939

$I(\alpha, \beta) = \text{True}$, sin embargo, estas variables se hacen dependientes cuando x es conocido, esto es $I(\alpha | x | \beta) = \text{False}$.

Las relaciones de dependencia o independencia entre variables son extensibles, mediante el criterio de *d-separación* [10], a cualquier tipo de estructura gráfica. El concepto de independencia gráfica se puede notar como $\langle X | Z | Y \rangle$. Las siguientes definiciones las podemos encontrar en [6].

Definición Sea Z un conjunto de nodos en un DAG. Un camino (no dirigido) t , se dice que está activo por Z si (1) Todo nodo cabeza a cabeza con respecto a t está en Z o tiene un descendiente en Z y (2) Cualquier otro nodo en t no pertenece a Z . En cualquier otro caso, decimos que el camino está bloqueado por Z .

Definición Sea D una red causal y X, Z y T tres subconjuntos disjuntos de nodos. Entonces X e Y se dice que son gráficamente independientes dado Z si no existe un camino (no dirigido) activado por Z entre cualquier nodo en X y cualquier nodo en Y . En otro caso, X e Y son gráficamente dependientes dado Z .

Siguiendo la notación dada por Pearl [10] llamaremos poliárbol a toda estructura donde la presencia de cualquier tipo de ciclos está prohibida. Entre dos nodos cualesquiera del poliárbol existe un único camino, en el que podemos encontrar arcos cabeza a cabeza, arcos cola a cola o arcos cabeza a cola.

Se han desarrollado técnicas que permiten el aprendizaje de redes en base a cálculos de medidas de entropía entre variables [4, 8, 11], o bien utilizando test de independencia condicional [1, 2], técnicas que permiten el aprendizaje de estructura simples, como [6, 7] y aquéllas que utilizan el conocimiento de un experto [13].

En la siguiente sección presentamos un algoritmo que, cuando el modelo se puede representar a través de un poliárbol, aprende la estructura de forma eficiente. En la sección tercera se presenta otro algoritmo que, utilizando una medida distancia entre distribuciones, obtiene como salida un poliárbol que aproxima al modelo original.

2 Aprendiendo estructuras de Poliarbol

Como hemos comentado, cuando el dominio del problema puede representarse mediante una estructura de poliárbol, el siguiente algoritmo recupera la estructura del mismo realizando test de independencia locales. Antes de comentar el algoritmo, veremos algunos conceptos previos.

Para cada nodo x , denominamos Λ_x al conjunto de variables que son dependientes con x (en el sentido de dependencia marginal). Si consideramos una estructura de poliárbol, dos variables x, y son dependientes si no existe un arco cabeza a cabeza en el único camino que las une, en cualquier otro caso se dice que son independientes.

La idea en la que se basa el algoritmo es la siguiente:

Supongamos que nuestro dominio del problema está formado por un conjunto U de variables y que conocemos, para cada nodo x , el conjunto de variables que son marginalmente dependientes con x , Λ_x .

Partimos de una estructura formada por un conjunto $T = \{\emptyset\}$ de variables, y en cada paso, una nueva variable se irá incorporando a la estructura. En el primer paso se selecciona una variable x , de forma que $T = \{x\}$ y una a una se incorporan a T aquéllas variables z pertenecientes a Λ_x (son marginalmente dependientes con x). A este proceso lo denominamos la Expansión de Λ_x . El esqueleto de la estructura obtenido lo llamamos

Algoritmo Aprendizaje de Poliárboles

1. Para cada variable x en M Inicializar Λ_x .
 - Visitado[x]=False
 - Expandido[x]=False
 - Para cada variable y en M . tal que $x \neq y$
 - If $I(x, y) = \text{False}$ Then unir y a Λ_x
2. Seleccionar un nodo x de M , asignar x a T ; Expandido[x]=True
3. While existan nodos no visitados en T
 - (a) Seleccionar un nodo no visitado x de T ; Visitado[x]= True
 - (b) While existan nodos no expandidos en Λ_x , Seleccionar un nodo no expandido z de Λ_x , el nuevo nodo a insertar en T
 - i. Expandido[z]=True
 - ii. Forward=True
 - iii. Inserted=False
 - iv. While Forward=True do
 - If todo $y \in \Psi_x$ se ha testado
 - Then Forward=False
 - Else
 - Selecciona un nuevo nodo y en Ψ_x
 - If $(x | y | z) = \text{True}$
 - Then $x = y$
 - v. Para todo y en Ψ_x hacer
 - If $(x | z | y) = \text{True}$
 - Incluir (x, z) y (z, y) en T
 - Borrar (x, y) de T
 - Inserted=True
 - vi. If Inserted=False Then Incluir (x, z) en T
 - vii. Reinicializar x al valor original

variables [4, 11], o tripletas de variables [12]. Todos estos algoritmos producen aproximaciones de la estructura de forma eficiente. Sin embargo, puede ser útil el conocer no solamente qué estructura de poliárbol aproxima al modelo, sino también las variables de la estructura que se ven afectadas por la presencia de un ciclo en el modelo original, así como la bondad de la aproximación sobre cada nodo. Esta información puede ser de utilidad al hacer la propagación del conocimiento sobre la estructura, o bien dada a un experto, sirviendo de guía para posibles modificaciones de la estructura.

La red causal permite representar una distribución de probabilidad P sobre el conjunto de variables U del modelo, asociando a cada nodo x en la red, la probabilidad $P(x | \Pi_x)$, donde Π_x representa el conjunto de padres (causas directas) de la variable x . A partir de las probabilidades condicionales, podemos obtener la distribución de probabilidad conjunta mediante la siguiente relación:

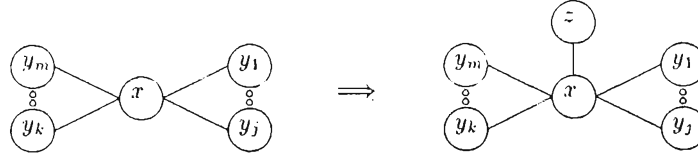


Figure 2: Teorema1 : Condicion dos

Si el Teorema2 se satisface, supongamos que Ψ_x está formado por el conjunto de variables $\{w_1, \dots, w_k, y\}$, con y la variable que hace verdadera la relación $I(x | y | z)$. Entonces nosotros conocemos que y está en el camino que conecta x con z , y por tanto, nuestro problema original se puede resolver estudiando si z pertenece al Haz de Nodos para y (Ψ_y). (ver fig. 3)

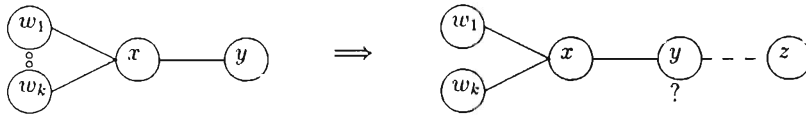


Figure 3: Teorema2 se satisface.

El algoritmo, listado en la siguiente página, permite construir el poliárbol que refleja el modelo en $O(n^2)$ pasos, con n representando el número de variables en el modelo. Para ello solamente utilizamos tests de independencia marginal y tests de independencia condicional de primer orden. Podemos encontrar otros algoritmos que permiten obtener la estructura del poliárbol utilizando un número polinomial de tests de independencia sobre el conjunto de variables en el modelo [6, 7].

El algoritmo únicamente reconstruye el esqueleto de la estructura, la dirección de los arcos se puede detectar haciendo uso de tests de independencia condicional [3, 11].

3 Aproximación

Si tenemos un modelo representable mediante una estructura de poliárbol, el algoritmo de la sección anterior permite recuperar las dependencias en el modelo de forma eficiente. Cuando en el modelo original existen ciclos, la salida del algoritmo es un poliárbol que refleja algunas de las relaciones de independencia entre variables, pero la eliminación de ciclos impone un conjunto de independencias no existentes en el modelo. La topología de la estructura depende en gran medida del orden con que se toman los nodos a expandir, el cual determina la pérdida de determinados arcos y la inclusión de otros en el poliárbol de salida.

Existen algoritmos que permiten recuperar árboles o poliárboles a partir de un conjunto de variables discretas utilizando una medida de entropía cruzada entre pares de

"Poliárbol Parcial". y representa el poliárbol que se obtendría al considerar únicamente las variables en T . Cuando se han insertado en T todas las variables de Λ_x , se selecciona una nueva variable x' a expandir de T , considerando todas aquellas variables z pertenecientes a $\Lambda_{x'}$ y que no se encuentren ya en T . El algoritmo terminará cuando todas las variables se hayan incluido en T .

Los siguientes teoremas nos permitirán detectar cómo la inclusión de una nueva variable en una estructura parcial T afecta a cada Haz de Nodos.

Definición: Sea cualquier variable del poliárbol. Definimos el *Haz de Nodos* para x , y lo notamos por Ψ_x , como el conjunto de variables que están conectadas directamente con x . Un Haz de Nodos para x contendrá las causas y los efectos directos de x .

Teorema1: Sea x cualquier variable en T y Ψ_x el Haz de nodos para x . Sea z la nueva variable a incluir en T , con $z \in \Lambda_x$. Tras incluir z en T , Ψ_x se verá modificado si y solo si alguna de las siguientes condiciones se satisface:

1. $I(x | z | y) = \text{True}$ para algún $y \in \Psi_x$.
2. $I(x | y | z) = \text{False}$ para todo $y \in \Psi_x$.

Cuando el teorema anterior no se satisface para un nodo x , por tanto $z \notin \Psi_x$, el siguiente teorema permite dirigir la búsqueda a través de la estructura T , hasta detectar aquellas variables donde el Haz de Nodos deba ser modificado, encontrando aquel nodo $y \in \Psi_x$ que separa a x de z .

Teorema2: En las mismas condiciones que el teorema anterior, cuando (1) y (2) no se satisfacen, entonces debe existir un nodo y solo uno $y \in \Psi_x$ tal que $I(x | y | z) = \text{True}$.

Teorema3: Las condiciones del Teorema1 y Teorema2 no se satisfacen simultáneamente.

Veamos gráficamente cómo la inclusión de un nuevo nodo z , perteneciente a Λ_x puede afectar a un Haz de Nodos para x cuando el Teorema1 se verifica. En este caso, si la condición uno se satisface, el conjunto de nodos Ψ_x se divide en dos subconjuntos, el formado por los nodos y_i , con $i = 1, \dots, j$, tal que $y_i \in \Psi_x$, para los que se cumple que $I(x | z | y_i)$ es cierto, y aquel formado por los nodos w_i con $i = 1, \dots, k$ que no se ven afectados por la inclusión de z . Debemos de crear nuevos arcos, uno entre x y z y el resto entre z y cada uno de los y_i , y borrar los arcos del grafo que unían x con y_i . Con estos cambios, z se encuentra ahora en Ψ_x y los nodos $y_i \notin \Psi_x$. Esto representa el hecho de que si z es conocido las variables x e y_i son independientes. (ver fig. 1).

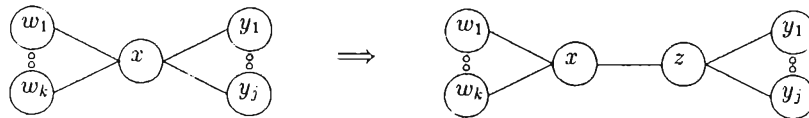


Figure 1: Teorema1 : Condición uno

Si la condición dos del Teorema1 se verifica, y la condición uno no se verifica, se incluye un nuevo arco entre x y z en el poliárbol reflejando el conocimiento de que x y z son variables relacionadas (son dependientes, recordemos que z pertenece a Λ_x) y ninguno de los nodos en Ψ_x se ven afectados por esta relación. (ver fig. 2).

$$P(x_1, \dots, x_n) = \prod_{x \in U} P(x | \Pi_x)$$

Si la estructura aprendida no representa fielmente al modelo, sino que es una aproximación del mismo, tendremos que distinguir entre la distribución de probabilidad generada por el modelo, P y la distribución de probabilidad que se obtiene, si se considera la estructura aprendida, al aplicar la anterior relación, y que llamaremos P^T .

Para seleccionar la mejor aproximación se considerarán ambas distribuciones. Se busca aquella estructura que minimize una medida distancia (usaremos la de Kullback-Leibler [9]) entre P y P^T . El cálculo de la distancia entre ambas distribuciones tiene un coste exponencial en el número de variables que componen el modelo, siendo un proceso bastante costoso. Proponemos aproximarla tratando de minimizar la suma de la medida distancia sobre el conjunto de variables que forman cada Haz de Nodos. Se trata de buscar, para cada variable n , el Haz de Nodos que mejor aproxime la distribución maginal de P sobre Ψ_n .

Definición: Para cada nodo n en el poliárbol, llamaremos D_n a la medida distancia entre la maginal de P sobre las variables en Ψ_n y la distribución conjunta sobre el mismo conjunto de variables, P^T , que se obtiene al considerar únicamente a Ψ_n como conjunto de variables en el modelo, esto es:

$$D_n(P(n_1, \dots, n_m), P^T(n_1, \dots, n_m)) = \sum P(n_1, \dots, n_m) \lg \frac{P(n_1, \dots, n_m)}{P^T(n_1, \dots, n_m)}$$

siendo n_1, \dots, n_m el conjunto de variables en Ψ_n . El considerar únicamente a las variables en Ψ_n , permite eliminar posibles errores provocados por la ruptura de algún ciclo, que se obtendrían al marginalizar la distribución P^T sobre Ψ_n .

La siguiente proposición nos asegura que cuando, para cada Haz de nodos, la medida distancia toma un valor nulo, el modelo original se puede representar a través de un poliárbol.

Proposición: Si para cada variable x en el poliárbol, D_x toma un valor nulo, entonces el modelo original se puede considerar isomorfo a un poliárbol.

Para todas aquellas variables n del modelo, tal que D_n tome un valor cero, podemos asegurar que, al marginalizar P sobre Ψ_n , la estructura resultante refleja fielmente al modelo. Por tanto, todas aquellas componentes del modelo representables a través de una estructura de poliárbol serán detectadas. Podemos utilizar, para cada variable n , la medida D_n como un estimador de la bondad de la aproximación para n , detectando las partes de la estructura que reflejan todas las dependencias del modelo entre variables y las partes que son una aproximación.

Un procedimiento sería aquel, que basandose en un poliárbol obtenido por un algoritmo de aprendizaje, seleccione aquellos nodos x , para los cuales D_x tome un valor mayor que un umbral ϵ . Este nodo x está afectado por un ciclo, y por tanto se debería estudiar todos los posibles poliárboles que se puedan generar como aproximación del ciclo y, de entre todos ellos, tomar aquel que haga mínimo el valor $\sum_n D_n$, con n el conjunto de variables que intervengan en el mismo. El número de posibles poliárboles que se pueden construir, cuando el número de ciclos en el modelo es grande, es bastante elevado, y encontrar la mejor topología puede dar lugar a un problema combinatorio. Por tanto, se propone un

método aproximado donde, tomando una a una las variables, se intente encontrar el mejor Haz para cada una de ellas.

En la búsqueda del mejor Haz, tomaremos cada nodo x del modelo y buscamos aquellos nodos que tienen una dependencia directa con x , o bien una dependencia provocada por la presencia de un ciclo. Para ello basta con ejecutar el paso 3(b) del algoritmo anterior para cada nodo x . Como solamente estamos interesados en el Haz para x , podemos realizar una poda sobre los nodos z , tal que $I(x | y | z) = \text{True}$, con $y \in \Psi_x$.

Aquellos nodos x , para los que $D_x < \epsilon$, se consideran estructuras correctas en la salida. Para el resto de nodos, influenciados por un ciclo, habrá que buscar la estructura que se considere más correcta, es decir, se tenga un valor D_x menor y que al insertarla en el poliárbol de salida, no genere un ciclo con las estructuras consideradas correctas en pasos anteriores. En cada paso, se estudian aquellos haces cuya inserción provocaría un ciclo, eliminando los arcos que lo pudiesen causar y nos quedamos con aquella estructura que da un valor distancia menor para cada haz. El algoritmo se repite hasta que todos los haces formen una única componente conexa.

La principal ventaja del método, es que además de obtener un árbol que nos aproxime una estructura causal, nos permite detectar qué variables forman parte de un ciclo, y cómo de buena es la aproximación al considerar únicamente los nodos que pertenecen a un Haz. Esta información puede sernos útil a la hora de realizar la propagación del conocimiento sobre la estructura aproximada.

4 Conclusiones

Hemos presentado un algoritmo capaz de recuperar una estructura de poliárbol a partir de datos de forma eficiente realizando únicamente tests de independencia marginal y condicional de primer orden. En la sección anterior se presenta un algoritmo que aproxima una red mediante una estructura de poliárboles. Existen algoritmos eficientes para realizar esta tarea, sin embargo, el algoritmo aquí propuesto incluye información en la salida del mismo que puede ser utilizada bien por el experto o bien en los programas de propagación. El algoritmo de aproximación es eficiente cuando las variables en el modelo están dispersas, ya que si el número de elementos en el Haz es elevado, los cálculos de la medida de entropía pueden ser costosos.

Queda trabajo a desarrollar, como puede ser un estudio comparativo entre los distintos algoritmos de aprendizaje y aproximación, así como el desarrollo de técnicas que permitan utilizar la distancia entre las distribuciones en cada Haz para realizar la propagación de creencias sobre la estructura.

Bibliografía

- [1] Acid S., Campos L.M. de, González A. Molina R., Pérez de la Blanca N. (1991) Learning with CASTLE, in Symbolic and Quantitative approaches to Uncertainty. Lecture Notes in Computer Science 584, R. Kruse R., Siegel P. (Eds). Springer Verlag, 99-106.

-
- [2] Acid S., Campos L.M. de, González A. Molina R., Pérez de la Blanca N. (1991) CASTLE: A tool for bayesian learning. Proceedings of the ESPRIT 91 Conference, Commission of the European Communities, 363-377.
 - [3] Campos L.M.de, Huete J.(1993) Learning Causal Politrees. Proceedings of the EC-SQARU'93 Conference.
 - [4] Chow C.K., Liu C.N.(1968) Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory 14, 462-467.
 - [5] Cooper G., Herskovits E. (1992) A bayesian method for the induction of Probabilistic Networks from data. Machine Learning 9, 309-347.
 - [6] Geiger D., Paz A., Pearl J. (1990) Learning causal trees from Dependence information. Proceedings of the Eighth National Conference on A.I. 770-776.
 - [7] Geiger D., Paz A., Pearl J. (1993) Learning Simple Causal Structures International Journal of Intelligent Systems, Vol 8, 231-247.
 - [8] Herskovits E., Cooper G. (1990) KUTATO: An Entropy-Driven system for construction of probabilistic expert systems from Databases. Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence 54-62, Cambridge MA.
 - [9] Kullback S.(1959) Information Theory and Statistic. Wiley. New York.
 - [10] Pearl J.(1988) Probabilistic Reasoning in intelligent systems: Networks of plausible inference. Morgan and Kaufman, San Mateo.
 - [11] Rebane G., Pearl J., (1989) The recovery of causal poly-trees from statistical data, Uncertainty in Artificial Intelligence 3, Kanal L.N., Levitt T.S. and Lemmer J.F. Eds., 175-182, North-Holland.
 - [12] Sarkar S. (1993) Using tre-decomposable Structures to aproximate belief networks. Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence. 376-382.
 - [13] Srinivas S., Russell S., Agogino A. (1990) Automated construction of sparse Bayesian networks from unstructured probabilistic models and domain information. Uncertainty in Artificial Intelligence 5, Henrion M., Shachter R.D., Kanal L.N., Lemmer J.F. (Eds) 295-308, North-Holland, Amsterdam.