El Clasificador Grafo de Decisión Probabilístico

Nielsen, Jens y Rumí, Rafael y Salmerón, Antonio

Resumen

Proponemos un algoritmo de aprendizaje supervisado de grafos de decisión probabilísticos (PDG) orientados a clasificación. El modelo PDG captura de forma natural independencias basadas en el contexto difícilmente representables con otros modelos gráficos como el Naïve Bayes (NB) o los árboles de clasificación (CT), lo que significa que los PDGs son capaces de representar ciertas distribuciones de probabilidad con menos parámetros, disminuyendo el riesgo de sobreajuste. Comparamos experimentalmente el modelo propuesto con varios clasificadores conocidos, observando un comportamiento competitivo del modelo PDG en la mayoría de los casos.

Keywords: Clasificación supervisada, Modelos gráficos
AMS: 62H30 , 68T10

1. Notación Básica

Notaremos mediante letras mayúsculas a las variables aleatorias, y mediante letras mayúsculas enfatizadas a los conjuntos de variables aleatorias, e.g. $\mathbf{X} = \{X_0, X_1, \ldots, X_n\}$. Notamos por $R(X)$ al conjunto de posibles estados de la variable X, extendiéndose de forma natural a conjuntos de variables aleatorias $R(\mathbf{X}) = \times_{X_i \in \mathbf{X}} R(X_i)$. Por letras minúsculas x (ó $\mathbf{x}$) notamos a algunos elementos de $R(X)$ (ó $R(\mathbf{X})$). Cuando $\mathbf{x} \in R(\mathbf{X})$ y $\mathbf{Y} \subseteq \mathbf{X}$, notamos por $\mathbf{x}[\mathbf{Y}]$ a la proyección de $\mathbf{x}$ sobre las coordenadas $\mathbf{Y}$.

2. El modelo Grafo de Decisión Probabilístico

El modelo Grafo de Decisión Probabilístico (PDG) fue presentado por primera vez por Bozga y Maler [2], y se propuso originalmente como una representación eficiente de sistemas de transición probabilística. En este estudio, consideramos la versión más general de los PDGs presentada por Jaeger [8].

Un PDG codifica una distribución de probabilidad conjunta sobre un conjunto de variables aleatorias discretas $\mathbf{X}$ mediante la representación de cada variable aleatoria $X_i \in \mathbf{X}$ por un conjunto de nodos V_i. Los nodos se organizan en un conjunto de estructuras DAG con raíz, consistentes con una estructura de árbol subyacente sobre las variables $\mathbf{X}$. La estructura se define formalmente como sigue:

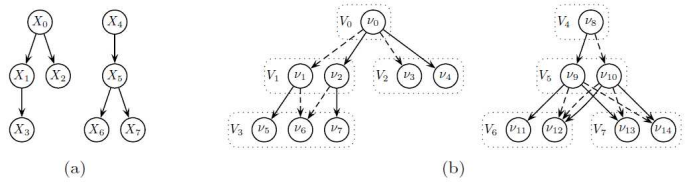


Figura 1: En (a) vemos un bosque de variables F sobre las variables binarias $\mathbf{X} = \{X_0, \ldots, X_7\}$, y en (b) una estructura-PDG sobre $\mathbf{X}$ con respecto al bosque de variables F.

Definición 2.1 (La Estructura PDG) *Sea F un bosque de variables sobre el dominio $\mathbf{X}$. Una* estructura-PDG $G = (\mathbf{V}, \mathbf{E})$ *para $\mathbf{X}$ con respecto a F es un conjunto de DAGs* con raíz*, tales que:*

1. *Cada nodo $\nu \in \mathbf{V}$ se etiqueta con alguna $X_i \in \mathbf{X}$. Nos referimos con V_i, al conjunto de todos los nodos en una estructura-PDG etiquetada con la misma variable X_i.*
2. *Para cada nodo ν_i etiquetado con X_i, cada posible estado $x_{i,h}$ de X_i y cada sucesor $X_j \in ch_F(X_i)$ existe exactamente* un *arco etiquetado con $x_{i,h}$ desde ν_i hasta algún nodo ν_j etiquetado con la variable aleatoria X_j. Sean $X_j \in ch_F(X_i)$ y $\nu_i \in V_i$. Nos referimos por $succ(\nu_i, X_j, x_{i,h})$ al nodo único $\nu_j \in V_j$ alcanzado desde ν_i por un arco etiquetado $x_{i,h}$.*

Ejemplo 2.1 *Podemos ver en la Figura 1(a) un bosque de variables F sobre las variables binarias $\mathbf{X} = \{X_0, \ldots, X_7\}$, y una estructura-PDG sobre $\mathbf{X}$ con respecto a F en la Figura 1(b). El etiquetado de los nodos ν en la estructura-PDG se indica mediante las cajas punteadas, e.g., los nodos etiquetados con X_2 se visualizan como el conjunto $V_2 = \{\nu_3, \nu_4\}$. Los arcos punteados corresponden a arcos etiquetados con el 0 y los arcos sólidos corresponden a arcos etiquetados con el 1, por ejemplo $succ(\nu_9, X_6, 0) = \nu_{12}$.*

Se instancia una estructura-PDG asignando a cada nodo ν una distribución multinomial local sobre la variable que representa. Nos referiremos a tales distribuciones locales como $\mathbf{p}^\nu = (p_1^\nu, \ldots p_{k_i}^\nu) \in \mathbb{R}^{k_i}$, donde $k_i = |R(X_i)|$ es la cantidad de estados distintos de X_i. Entonces, nos referimos con $p_{x_{i,h}}^\nu$ al h'ésimo elemento de $\mathbf{p}^\nu$ bajo algún orden de $R(X_i)$.

Definición 2.2 (El modelo PDG) *Un modelo PDG $\mathcal{G}$ es un par $\mathcal{G} = (G, \theta)$, donde G es una estructura-PDG válida (Def. 2.1) sobre algún conjunto $\mathbf{X}$ de variables aleatorias discretas y θ es un conjunto de distribuciones locales que instancia completamente a G.*

Definición 2.3 (Alcance) *Sea G una estructura-PDG sobre las variables $\mathbf{X}$ con respecto al bosque F. Un nodo ν en D etiquetado con X_i es* alcanzado *por $\mathbf{x} \in R(\mathbf{X})$ si*

- ν *es una raíz en* G, *o*
- $X_i \in ch_F(X_j)$, $\nu' \in V_j$, ν' *es alcanzado por* $\mathbf{x}$ *y* $\nu = succ(\nu', X_i, \mathbf{x}[X_j])$.

Notamos por $reach_G(i, \mathbf{x})$ al único nodo-parámetro $\nu \in V_i$ alcanzado por $\mathbf{x}$ en la estructura-PDG G.

Un modelo PDG instanciado $\mathcal{G} = (G, \theta)$ sobre las variables $\mathbf{X}$ representa una distribución conjunta $P^{\mathcal{G}}$ de acuerdo a la siguiente factorización:

$$P^{\mathcal{G}}(\mathbf{x}) = \prod_{X_i \in \mathbf{X}} p_{\mathbf{x}[X_i]}^{reach_G(i,\mathbf{x})}. \tag{1}$$

Un conjunto de nodos V_i en una estructura-PDG G sobre las variables $\mathbf{X}$ particiona el espacio de estados $R(\mathbf{X})$ en un conjunto de subconjuntos disjuntos, concretamente $(\nu \in V_i)\{\mathbf{x} \in R(\mathbf{X}) : reach_G(i, \mathbf{x}) = \nu\}$. Notamos por $\mathcal{A}_G(X_i)$ a la partición de $R(\mathbf{X})$ definida por V_i en G. Entonces, la estructura-PDG G impone las siguientes relaciones de independencia condicional:

$$X_i \perp\!\!\!\perp nd_G(X_i) | \mathcal{A}_G(X_i)\,, \tag{2}$$

donde $nd_G(X_i)$ denota a los no-descendientes de X_i en la estructura G.

3. Clasificación

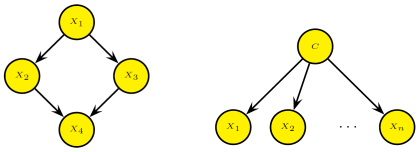
Uno de los usos más importantes de los modelos gráficos es la **clasificación** de individuos con respecto a alguna información conocida. En un problema de clasificación, una de las variables es la variable **clase**, C, y el resto de ellas son las **características**, $X_1, \ldots, X_n$. El objetivo del modelo de clasificación es predecir el valor de la variable clase, c^*, de un individuo, dada una configuración de las variables características, $x_1, \ldots, x_n$.

3.1. Clasificadores Comunes

Existen diferentes tipos de modelos de clasificación. Primero describiremos los llamados **clasificadores bayesianos**, que son tipos concretos de redes bayesianas [5]. Una red bayesiana (ver Fig. 2 (a)) es un grafo dirigido acíclico en el que cada nodo representa una variable aleatoria, y la existencia de un arco entre dos nodos indica una dependencia entre las correspondientes variables aleatorias. Cada nodo tiene asociado una distribución de probabilidad de la variable aleatoria correspondiente condicionada a las variables aleatorias correspondientes a sus padres en el grafo.

Una propiedad muy importante de las redes bayesianas es que la distribución de probabilidad conjunta de las variables de la red factoriza según el concepto de d-separación como sigue:

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | pa(x_i)) \tag{3}$$



(a) (b)

Figura 2: (a) Ejemplo de una red bayesiana - (b) Modelo Naive Bayes

donde $pa(x_i)$ denota el conjunto de padres de la variable x_i. Esta factorización implica que la distribución conjunta de todas las variables de la red se puede especificar con una importante reducción de complejidad.

Por ejemplo, la red bayesiana de la Figura 2 (a) induce la siguiente factorización de la distribución de probabilidad conjunta:

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2, x_3)$$

Los clasificadores bayesianos son redes bayesianas con una estructura restringida. A la hora de clasificar, se asigna a un individuo el valor más probable de la variable clase, dado el valor que toman sus variables características

$$c^* = \underset{c \in R(C)}{argmax}\ p(c|x_1, \ldots, x_n) \ .$$

Usando la fórmula de Bayes, el cálculo de arriba se resuelve fácilmente como

$$p(c|x_1, \ldots, x_n) = \frac{p(c, x_1, \ldots, x_n)}{p(x_1, \ldots, x_n)} = \frac{p(c)p(x_1, \ldots, x_n|c)}{p(x_1, \ldots, x_n)} \propto p(c)p(x_1, \ldots, x_n|c) \ . \quad (4)$$

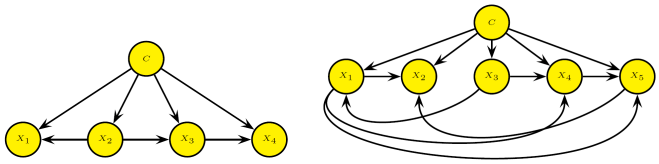
El modelo bayesiano más sencillo, y también uno de los más precisos, es el modelo **Naive Bayes** (NB) o Bayes ingenuo [9]. En el clasificador NB, las variables características se suponen independientes dada la clase, de forma que las únicas relaciones de dependencia existentes en el modelo relacionan la variable clase con todas las características, dando como resultado un red bayesiana muy simple, como la que aparece en la Fig. 2 (b). Aunque esta suposición no siempre se cumple, no suele afectar en exceso los resultados del modelo.

Esta simplicidad en la estructura también implica una reducción en el número de parámetros del modelo. De acuerdo con la Eq. (3), en el modelo NB, la Eq. (4) se reduce a

$$p(c|x_1, \ldots, x_n) = p(c) \prod_{i=1}^{n} p(x_1|c) \ ,$$

de tal forma que, cuando se aprende el modelo, se necesitan n distribuciones condicionadas 1-dimensionales, en lugar de 1 distribución de probabilidad condicionada n-dimensional.

El modelo **Tree Augmented Network** (TAN) [7] es otro clasificador bayesiano que relaja la suposición del NB. En el modelo TAN se permite a las variables características estar conectadas a 1 variable más de la red, aparte de la variable clase, i.e., las variables características no han de ser independientes dada la clase (ver Fig. 3 (a)).



(a) (b)

Figura 3: (a) Ejemplo de un modelo TAN con 4 características. - (b) Ejemplo de un modelo 2DB con 5 características

El modelo Naive bayes y el modelo TAN son casos particulares de un modelo más general, el modelo **k-dependence Bayesian network** (KDB) [13]. El clasificador KDB es una red bayesiana en la que cada característica puede tener un máximo de k variables características como padres, aparte de la variable clase, que es un padre de todas las características (ver Fig. 3 (b)).

Así, el modelo Naive Bayes es un modelo 0DB, y el modelo TAN es un modelo 1DB.

Otro tipo de modelos de clasificación son los **árboles de clasificación**. Un árbol de clasificación es un modelo jerárquico, compuesto por hojas terminales y nodos de decisión. Cada nodo de decisión representa una pregunta acerca de una característica, con un número finito de respuestas. A través de cada respuesta se conecta ese nodo de decisión con otro nodo de decisión o bien con un nodo hoja. Los nodos hoja no tienen más enlaces, pero continen un valor para la variable clase (ver Fig. 4).

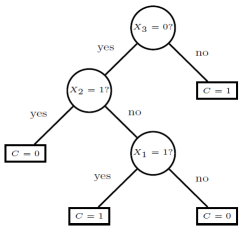


Figura 4: Ejemplo de un árbol de clasificación con 3 características binarias y una variable clase también binaria. Los nodos circulares son nodos de decisión y los nodos rectángulares nodos hoja o nodos terminales.

Para aprender un árbol de clasificación a partir de un conjunto de datos se debe primero responder a una serie de cuestiones que se plantean, como por ejemplo qué tipo de preguntas se pueden realizar, cómo asignar los valores de la clase a las hojas o si se poda el árbol para reducir su tamaño. Dependiendo de las respuestas a estas preguntas, se pueden aprender diferentes tipos de árboles de clasificación, de entre los cuales podemos destacar el árbol **CART** (Classification and regression trees) [3], el árbol **C4.5** [12] y su antecesor, el **ID3** [11] y el **árbol de clasificación Dirichlet** [1].

3.2. El Clasificador PDG

Como sugiere el título, en este trabajo investigamos el comportamiento del modelo PDG en problemas de clasificación supervisada. En esta sección introducimos el modelo específico de clasificación PDG en el que nos centraremos en el trabajo y daremos algunos ejemplos de su potencial como modelo de clasificación. En lo siguiente, nos referiremos por C a la variable clase, por $\mathbf{X}$ al conjunto de variables discretas características y $\mathbf{C} = \{C\} \cup \mathbf{X}$.

Definición 3.1 (El clasificador PDG) *Un* clasificador PDG $\mathcal{C}$ *es un modelo PDG que además de las restricciones estructurales de la Def. 2.1 satisface las siguientes dos características estructurales:*

1. *G define un árbol conteniendo un sólo árbol sobre las variables $\mathbf{C}$, y*
2. *C es la raíz de este árbol.*

El modelo PDG se inspiró inicialmente en ROBDDs [4], un marco de modelización que permite la representación eficiente de expresiones booleanas. Como veremos en el siguiente ejemplo, el modelo PDG ha heredado la habilidad para representar expresiones booleanas de forma eficiente, al menos hasta cierto punto.

Ejemplo 3.1 *Consideremos el problema de clasificación en el que todas las variables características $\mathbf{X}$ son verdadero-falso (esto es, para todo $X \in \mathbf{X}$ tenemos $R(X) = \{\texttt{true}, \texttt{false}\}$) con distribución uniforme. Supongamos también que $R(C) = \{\texttt{true}, \texttt{false}\}$, y que el estado de C dada una configuración $\mathbf{x} \in R(\mathbf{X})$ se define de forma determinística como:*

$$P(C = \texttt{true}|\mathbf{X} = \mathbf{x}) = \begin{cases} 1 & si \ \underset{X \in \mathbf{X}}{\texttt{XOR}}\, \mathbf{x}[X], \\ 0 & en\ otro\ caso. \end{cases} \tag{5}$$

El concepto definido en (5) se recoge de forma eficiente en la estructura PDG de la Fig. 5(a) donde los parámetros $\{p^{\nu_i}_{\texttt{true}}, p^{\nu_i}_{\texttt{false}}\}$ son $\{0, 1\}$ para $i \in \{7, 12, 14\}$, $\{1, 0\}$ para $i \in \{13, 15\}$, $\{\frac{1}{4}, \frac{3}{4}\}$ para $i = 0$ y $\{\frac{1}{2}, \frac{1}{2}\}$ para otro i. Esta estructura define un modelo que contiene 16 parámetros libres, de tal forma que añadir más variables características a la función o-exclusivo que determina la etiqueta de la instancia sólo conlleva una adición constante de parámetros al modelo. Ni el clasificador NB ni el TAN pueden representar la distribución de $C, \mathbf{X}$ definida por este modelo.

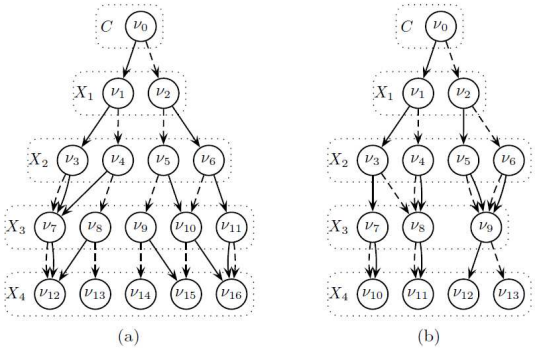


Figura 5: Las dos estrcuturas diferentes de clasificadores PDG comentadas en el Ejemplo 3.1. Los arcos sólidos se corresponden con el valor `true` y los arcos punteados se corresponden con `false`.

La Fig. 5(b) muestra una estructura PDG capaz de representar de forma eficiente el modelo en el que la etiqueta clase de una instancia se determina como la disyunción de conjunciones de pares de variables características:

$$P(C = \mathit{true}|\mathbf{X} = \mathbf{x}) = \begin{cases} 1 & \textit{si } (\mathbf{x}[X_0] \wedge \mathbf{x}[X_1]) \vee (\mathbf{x}[X_2] \vee \mathbf{x}[X_3]), \\ 0 & \textit{en otro caso.} \end{cases} \tag{6}$$

El concepto expresado en (6) se recoge en la instantiación de la estructura de la Fig. 5(b) con parámetros $\{p^{\nu_i}_{\mathit{true}}, p^{\nu_i}_{\mathit{false}}\}$ *:* $\{1, 0\}$ *para* $i \in \{8, 11\}$, $\{0, 1\}$ *para* $i \in \{5, 12\}$, $\{\frac{7}{16}, \frac{9}{16}\}$ *para* $i = 0$ *y* $\{\frac{1}{2}, \frac{1}{2}\}$ *en otro caso. Este modelo define 13 parámetros libres, y de nuevo, este número crece de forma lineal conforme añadimos pares de variables características a la disyunción de (6). Este modelo también define una distribución que no puede ser representada por los modelos NB o TAN.*

3.3. Clasificadores Bayesianos como PDGs

Se sabe que un árbol de cliques obtenido a partir de una red bayesiana puede representarse mediante un PDG con un número de parámetros libre lineal en el tamaño del árbol de cliques [8]. Si consideramos redes bayesianas orientadas a la clasificación, tales como el TAN o el naive Bayes, entonces la equivalencia en número de parámetros no se da solamente para el árbol de cliques, sino también para la red bayesiana, tal y como establece el siguiente teorema.

Teorema 3.1 *Sea B una red bayesiana con estructura TAN. Entonces, existe al menos un PDG G con tamaño lineal en el tamaño de B tal que $P_B = P_G$, donde P_B y P_G son las distribuciones conjuntas representadas por B y G respectivamente.*

Demostración: Jaeger [8] demostró que dado un árbol de cliques, es posible encontrar un PDG con tamaño (número de parámetros libres) lineal en el tamaño del árbol de cliques. Por tanto, es suficiente demostrar que es posible construir un árbol de cliques a partir de un TAN con el mismo tamaño que la red original. Obsérvese que el grafo moral de un TAN se obtiene simplemente ignorando la dirección de los arcos de la red original, dado que los padres de cada variable característica (la clase y otra característica) están conectados por definición (la clase está conectada con todas las características). Además, es un grafo triangulado, por estar formado por la unión de triángulos compuestos por cada característica y sus padres. Ahora, dado que ningún triángulo está contenido en otro subgrafo completo más grande, dado que eso implicaría que alguna variable característica tendría más de dos padres, se sigue que los cliques del grafo triangulado son los triángulos correspondientes a cada familia de la red original. En consecuencia, podemos distribuir los cliques de forma que obtengamos un árbol de cliques donde cada uno corresponda a una familia en la red original, y por tanto sus tamaños sean iguales. En realidad, el tamaño del árbol de cliques puede ser ligeramente inferior, dado que la distribución de la variable clase y la condicionada de la primera característica dada la clase pueden incluirse en el mismo clique que la familia de la segunda variable característica. □

Corolario 3.1 *Sea B una red bayesiana con estructura FAN. Entonces, hay al menos un PDG G con tamaño lineal en el tamaño de B tal que $P_B = P_G$, donde P_B y P_G son las distribuciones conjuntas representadas por B y G respectivamente.*

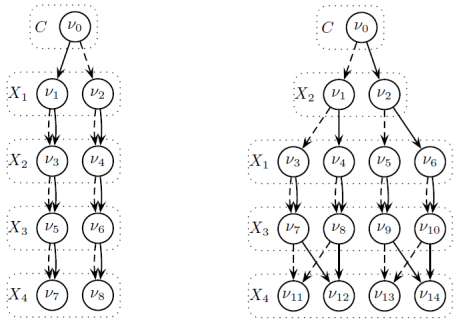
Demostración: La demostración es similar a la del teorema 3.1, dado que un FAN es un TAN donde algunos enlaces entre características no existen. □

Corolario 3.2 *Sea B una red bayesiana con estructura naive Bayes. Entonces, hay al menos un PDG G con tamaño lineal en el tamaño de B tal que $P_B = P_G$, donde P_B y P_G son las distribuciones conjuntas representadas por B y G respectivamente.*

Demostración: La demostración es similar a la del teorema 3.1. En este caso, en lugar de triángulos, los cliques están formados por pares de variables, formados por cada característica y la clase. □

Cuadro 1: Instanciación de probabilidades para el PDG de la Figura 6 (a).

$p^{\nu 0} = P(C)$			
$p^{\nu 1} = P(X_1 \mid C = 0)$	$p^{\nu 2} = P(X_1 \mid C = 1)$	$p^{\nu 3} = P(X_2 \mid C = 0)$	$p^{\nu 4} = P(X_2 \mid C = 1)$
$p^{\nu 5} = P(X_3 \mid C = 0)$	$p^{\nu 6} = P(X_3 \mid C = 1)$	$p^{\nu 7} = P(X_4 \mid C = 0)$	$p^{\nu 8} = P(X_4 \mid C = 1)$



(a) (b)

Figura 6: (a) Un clasificador naive Bayes con cuatro variables características representado como un PDG - (b) Un PDG equivalente al TAN de la Fig. 3 (a)

Cuadro 2: Instanciación de probabilidades para el PDG de la Figura 6 (b).

$p^{\nu_0} = P(C)$	
$p^{\nu_1} = P(X_2 \mid C = 0)$	$p^{\nu_2} = P(X_2 \mid C = 1)$
$p^{\nu_3} = P(X_1 \mid X_2 = 0, C = 0)$	$p^{\nu_4} = P(X_1 \mid X_2 = 1, C = 0)$
$p^{\nu_5} = P(X_1 \mid X_2 = 0, C = 1)$	$p^{\nu_6} = P(X_1 \mid X_2 = 1, C = 1)$
$p^{\nu_7} = P(X_3 \mid X_2 = 0, C = 0)$	$p^{\nu_8} = P(X_3 \mid X_2 = 1, C = 0)$
$p^{\nu_9} = P(X_3 \mid X_2 = 0, C = 1)$	$p^{\nu_{10}} = P(X_3 \mid X_2 = 1, C = 1)$
$p^{\nu_{11}} = P(X_4 \mid X_3 = 0, C = 0)$	$p^{\nu_{12}} = P(X_4 \mid X_3 = 1, C = 0)$
$p^{\nu_{13}} = P(X_4 \mid X_3 = 0, C = 1)$	$p^{\nu_{14}} = P(X_4 \mid X_3 = 1, C = 1)$

Los resultados anteriores garantizan que algunos clasificadores basados en redes bayesianas tienen un PDG equivalente (en términos de tamaño). Otros clasificadores como el KDB no admiten un árbol de cliques con tamaño lineal en el de la red original. Más concretamente, aquellos que contengan subestructuras tipo *diamante* [8], son ejemplos de ello.

En general, no es necesario obtener un árbol de cliques y luego transformarlo en un PDG usando el método descrito en [8]. El algoritmo 3.1 detalla un procedimiento para obtener un PDG a partir de un clasificador bayesiano sin necesidad de recurrir a la transformación previa en árbol de cliques.

La idea del algoritmo es construir un PDG con bosque de variables con estructura lineal, donde la raíz es la variable clase y las características se disponen linealmente en orden topológico. Los nodos paramétricos se conectan de tal forma que haya un camino entre cada uno de ellos y la configuración de sus variables padre en la red bayesiana original que se corresponde con la distribución contenida en dicho nodo. El algoritmo es válido para los clasificadores mencionados anteriormente, excepto para algunos como los KDB con subestructuras tipo diamante, para los cuales el paso 18 fallaría.

Algoritmo 3.1 (Construcción de un PDG a partir de un clasif. bayesiano)

1. Sea *net* un clasificador bayesiano.
2. Sea C la variable clase de *net*.
3. Crear un PDG G con un solo nodo C en el bosque de variables.
4. Añadir un nodo paramétrico ν con $p^{\nu} = P(C)$ al nodo C.
5. $\mathbf{C} \leftarrow \{C\}$.
6. Sea $\mathbf{X}$ el conjunto de características de G.
7. Ordenar $\mathbf{X}$ según un orden topológico.
8. **While** ($\mathbf{X} \neq \emptyset$)
9. Sea Y la primera variable de $\mathbf{X}$.
10. $\mathbf{X} \leftarrow \mathbf{X} \setminus \{Y\}$.
11. $\mathbf{C} \leftarrow \mathbf{C} \cup \{Y\}$.
12. Sea L la hoja del bosque de variables de G.
13. Insertar Y en el bosque de variables de G como hija de L.
14. Sea $Pa(Y)$ el conjunto de padres de Y en *net*.
15. **Para todo** $\mathbf{w} \in R(Pa(Y))$
16. Añadir un nodo paramétrico ν a Y.
17. Instanciar ν con $p^{\nu} = P(Y|Pa(Y) = \mathbf{w})$.
18. Hacer que ν se alcanzado por cualquier $\mathbf{x} \in R(\mathbf{C})$ tal que $\mathbf{x}[Pa(Y)] = \mathbf{w}$.
19. **return** G.

Ejemplo 3.2 (Construcción de un PDG a partir de un NB) *Sea un naive Bayes con cuatro variables características ($X_1, \ldots, X_4$) y clase C. Podemos construir un PDG a partir de él usando el algoritmo 3.1. En primer lugar, insertamos C como raíz del bosque de variables del PDG, y la distribución marginal de C se instancia en el nodo paramétrico. Entonces, las variables características se ordenan en orden topológico y se insertan de esa manera en el bosque de variables. Obsérvese que las conexiones entre los nodos paramétricos correspondientes a las variables características no discriminan entre los diferentes valores, dado que las características son independientes dada la clase. El PDG resultante puede verse en la Figura 6 (a), y la instanciación de los nodos paramétricos en la Tabla 1.*

Ejemplo 3.3 (Construcción de un PDG a partir de un TAN) *Sea el TAN mostrado en la Figura 3. La construcción de un PDG a partir de él usando el algoritmo 3.1 es similar al caso del NB. La principal diferencia es que las conexiones entre nodos paramétricos ahora reflejan las dependencias entre las características. El PDG resultante puede verse en la Figura 6 (b), y la instanciación de los nodos paramétricos en la Tabla 2.*

4. Aprendizaje del Clasificador PDG a partir de Datos

En esta sección proponemos un algoritmo para aprender clasificadores PDG a partir de datos etiquetados. El algoritmo construye la estructura del PDG incrementalmente añadiendo variables de $\mathbf{X}$ a la estructura del árbol de variables con raíz C de acuerdo con la máxima información mutua (condicional). La estructura del PDG sobre los nodos paramétricos se optimiza usando operaciones guiadas por la **tasa de clasificación (TC)** sobre un conjunto de validación separado.

Algoritmo 4.1 (Función LearnPDGC($\mathcal{D}$))
Input: *Conjunto de datos $\mathcal{D}$ de observaciones completas de las variables características* $\mathbf{X}$ *y la clase* C.

1. Instanciar un modelo PDG $\mathcal{G}$ con un único árbol formado por C como raíz.
2. $\mathbf{C} \leftarrow \{C\}$.
3. **While** ($\mathbf{X} \neq \emptyset$)
4. $\quad \langle X_c, X_p \rangle \leftarrow \underset{X_c \in \mathbf{X}, X_p \in \mathbf{C}}{argmax} \; CMI_{\mathcal{D}}(X_c, X_p, \mathscr{A}_G(X_p))$.
5. $\quad \mathbf{X} \leftarrow \mathbf{X} \setminus \{X_c\}$.
6. $\quad \mathbf{C} \leftarrow \mathbf{C} \cup \{X_c\}$.
7. $\quad$ Añadir X_c a $\mathcal{G}$ bajo X_p.
8. $\quad$ **Para todo** $\nu \in V_p$ y $x_{p,i} \in R(X_p)$
9. $\quad\quad$ Añadir un nodo paramétrico ν' representando a X_c.
10. $\quad\quad$ Añadir conexiones tales que $\nu' = succ(\nu, X_c, x_{p,i})$.
11. $\quad$ Unir los nodos paramétricos hacia arriba desde la nueva hoja X_c.
12. **return** $\mathcal{G}$.

En el algoritmo 4.1, la función $CMI_{\mathcal{D}}(X, Y, \mathscr{A}_G(Y))$ calcula la información mutua condicionada entre X e Y dada la partición definida por el conjunto de nodos que representan a Y en la estructura G, y definida como:

$$CMI_{\mathcal{D}}(X, Z, \mathscr{A}_G(Y)) = \sum_{a \in \mathcal{D}[\mathscr{A}_G(Y)]} \sum_{x \in R(X)} \sum_{z \in R(Z)} \hat{P}_a(X = x, Z = z) \log \frac{\hat{P}_a(X = x, Z = z)}{\hat{P}_a(X = x)\hat{P}_a(Z = z)}, \quad (7)$$

donde $\hat{P}_a$ es la estimación de máxima verosimilitud de la probabilidad dada, calculada usando solamente los datos de a y $\mathcal{D}[\mathscr{A}_G(Y)]$ es la partición de los datos $\mathcal{D}$ definida mediante la partición $\mathscr{A}_G(Y)$ de $R(\mathbf{X})$.

En las líneas 7 a 10 del algoritmo 4.1, incluimos la variable X_c en la estructura PDG actual G bajo X_p incluyendo un nodo paramétrico para cada nodo paramétrico de X_p y cada valor $x_{p,i} \in R(X_p)$. De esta forma no restringimos a priori la expresividad del modelo, sino que incluimos el número máximo de parámetros.

En la línea 11 del algoritmo 4.1, los nodos paramétricos se fusionan de abajo hacia arriba a partir de los nodos paramétricos recién creados. La fusión de nodos paramétricos está guiada por un conjunto de datos para validación, tal que dos nodos se juntan solamente si el PDG resultante tiene mayor tasa de clasificación sobre el conjunto de validación.

La fusión de nodos paramétricos es la operación que debería identificar las independencias basadas en el contexto que existan en el modelo subyacente, y que podrían producir un modelo con menos parámetros. Nos guiamos por la tasa de clasificación puesto que nuestro objetivo final es aprender un clasificador con tasa de éxito lo más alta posible.

Las variables que no son independientes deberían colocarse próximas en la estructura de forma que la operación de fusión pueda capturar las independencias basadas en el contexto que sean relevantes. Esto justifica el uso de la información mutua para guiar la inserción de nuevas variables a la estructura del PDG.

5. Experimentos y Resultados

En esta sección describimos los experimentos llevados a cabo para evaluar el clasificador PDG de la Def. 3.1 y el procedimiento de aprendizaje descrito en el Alg. 4.1.

Hemos comparado nuestro algoritmo frente a clasificadores conocidos como NB, TAN, KDB y los árboles de clasificación (CT) (ver sección 3.1). Para el clasificador KDB, hemos usado $k = 4$. En todos los casos, hemos empleado la implementación disponible en el sistema Elvira.

Hemos empleado dos grupos de bases de datos, uno formado por datos sintéticos muestreados a partir de PDGs y otro conteniendo datos reales tomados del repositorio de la UCI [10].

Los datos sintéticos que hemos usado son de dos tipos. Primero, hemos generado al azar 3 PDGs de diferente complejidad, con 9, 19 y 29 variables característica, denotados como Rnd10, Rnd20 y Rnd30 respectivamente. En cada uno de ellos las variables tienen 2 0 3 posibles valores, y las muestras obtenidas constan de 1000 individuos. El otro tipo de datos sintéticos fue generado a partir de fórmulas booleanas, algunas con ruido (ver tabla 3). De nuevo tomamos un tamaño muestral de 1000.

Datos	Descripción
3-of-10	La clase es `true` cuando exactamente 3 de las 10 características son `true`.
5-of-10	La clase es `true` cuando exactamente 5 de las 10 características son `true`.
discon	La clase es `true` cuando $(X_1 \wedge X_2) \vee (X_2 \wedge X_3) \vee (X_3 \wedge X_4) \vee (X_4 \wedge X_5)$ es `true`, y `false` en otro caso.
noisy-discon	Similar al anterior, pero con ruido.
noisy-or	La clase se calcula como una función noisy or sobre las 5 variables característica.

Cuadro 3: Datos sintéticos muestreados a partir de fórmulas booleanas.

Datos	#Características	#Clases	#Registros
car	6	4	1727
chess	36	2	3195
iris	4	3	150
mushroom	21	2	5643
monks-1	6	2	431
monks-3	6	2	431
nursery	8	5	12959
postop	8	3	86

Cuadro 4: Datos tomados del repositorio UCI ML.

Finalmente, consideramos una serie de bases de datos tomadas del repositorio UCI Machine Learning Repository, tanto reales como sintéticos (ver tabla 4), a las que se les han eliminado los registros faltantes y se han discretizado las variables continuas, usando el procedimiento k-means implementado en Elvira.

En las tablas mostradas en esta sección, denotamos por TC a la tasa de clasificación y por Tam. al número de parámetros libres del modelo. En todos los experimentos, los valores de TC y Tam. están estimados por validación cruzada con 5 hojas. Las filas etiquetadas como c45-E, c45-R y c45-N se corresponden con el algoritmo C45 de aprendizaje de árboles de clasificación, usando com método de poda EBP, REP y sin poda respectivamente, y de forma similar para las filas etiquetadas con id3-/dir-.

	Rnd10		Rnd20		Rnd30	
Modelo	TC	Tam.	TC	Tam.	TC	Tam.
PDG	0.912	90.4	0.426	376.4	0.892	133.4
TAN	0.940	38.6	0.399	233.8	0.715	243.4
NB	0.920	21.0	0.439	86.0	0.717	93.0
KDB	0.945	267.8	0.408	3757.4	0.777	4260.2
c45-E	0.929	2.4	0.467	3.0	0.889	47.2
c45-R	0.930	24.4	0.458	484.8	0.878	140.8
c45-N	0.933	172.8	0.410	1535.4	0.865	257.2
id3-E	0.928	2.0	0.466	7.2	0.891	41.6
id3-R	0.928	3.6	0.437	629.4	0.869	166.4
id3-N	0.934	175.2	0.394	1525.2	0.860	273.2
dir-E	0.928	2.0	0.466	6.0	0.885	28.8
dir-R	0.933	16.0	0.458	112.8	0.885	108.4
dir-N	0.946	88.0	0.411	866.4	0.868	195.2

Cuadro 5: Resultados para los datos sintéticos muestreados a partir de PDGs.

	3-of-10		5-of-10		discon		discon-noisy		noisy-or	
Modelo	TC	Tam.	TC	Tam.	TC	Tam.	TC	Tam.	TC	Tam.
PDG	0.877	236.8	0.684	190.8	1.000	44.8	0.861	43.4	0.974	32.4
TAN	0.881	39.0	0.727	39.0	0.885	18.2	0.833	18.2	0.858	18.2
NB	0.877	21.0	0.737	21.0	0.806	11.0	0.807	11.0	0.964	11.0
KDB	0.891	223.0	0.726	223.0	1.000	63.0	0.856	63.0	0.980	63.0
c45-E	0.893	2.0	0.737	2.0	0.986	17.2	0.792	2.0	0.964	2.0
c45-R	0.894	37.6	0.740	61.6	1.000	18.0	0.860	50.0	0.969	19.6
c45-N	0.893	306.4	0.770	592.0	1.000	18.0	0.857	57.2	0.980	43.6
id3-E	0.893	2.0	0.737	2.0	0.986	17.2	0.792	2.0	0.964	2.0
id3-R	0.894	25.2	0.736	28.4	1.000	18.0	0.860	50.0	0.969	19.6
id3-N	0.892	306.8	0.768	589.6	1.000	18.0	0.857	57.2	0.980	43.6
dir-E	0.893	2.0	0.737	2.0	0.885	10.0	0.792	2.0	0.967	7.2
dir-R	0.893	11.2	0.737	2.8	0.986	17.2	0.847	32.0	0.980	21.6
dir-N	0.889	220.0	0.759	536.4	1.000	18.0	0.857	38.8	0.980	21.6

Cuadro 6: Resultados para los datos sintéticos generados a partir de funciones booleanas.

La tabla 5 muestra los resultados obtenidos a partir de los datos muestreados de PDGs. Cabría esperar que los clasificadores PDG fueran superiores para estos conjuntos de datos, pero ese efecto no se aprecia para los modelos menos complejos Rnd10 y Rnd20. Para Rnd30, por contra, se aprecia en general el mejor comportamiento del modelo PDG, especialmente frente a los TAN, NB y KDB models. Los árboles de clasificación se comportan de forma competitiva para las bases de datos Rnd10-30.

La tabla 6 muestra los resultados de aprender de los datos muestreados de las funciones booleanas descritas en la tabla 3. Se observa que el modelo PDG obtiene valores de precisión competitivos en la mayoría de los casos (excepto para 5-of-10). En particular, para los datos noisy-or, el modelo PDG es comparable a los clasificadores bayesianos. De nuevo, los árboles de clasificación obtienen excelentes resultados.

La tabla 7 muestra los resultados de los modelos obtenidos a partir de los datos procedentes del repositorio UCI descritos en la tabla 4. En estos experimentos, no hay un modelo que se comporte siempre mejor o peor que los demás.

	car		chess		iris		mushroom	
Modelo	TC	Tam.	TC	Tam.	TC	Tam.	TC	Tam.
PDG	0.883	192.0	0.872	116.4	0.906	50.0	0.999	1017.6
TAN	0.936	181.4	0.922	148.2	0.940	194.0	0.999	773.8
NB	0.854	63.0	0.875	75.0	0.933	50.0	0.970	153.0
KDB	0.947	3109.4	0.961	1307.8	0.926	1874.0	1.000	113822.2
c45-E	0.700	4.0	0.948	15.6	0.926	22.2	0.998	30.0
c45-R	0.837	732.0	0.967	46.0	0.886	108.6	0.999	31.6
c45-N	0.898	1063.2	0.996	91.2	0.886	108.6	1.000	34.0
id3-E	0.700	4.0	0.940	11.2	0.920	27.0	1.000	38.0
id3-R	0.806	504.0	0.964	42.8	0.893	106.2	1.000	38.0
id3-N	0.900	1060.8	0.994	91.2	0.893	106.2	1.000	38.0
dir-E	0.700	4.0	0.940	10.0	0.920	15.0	1.000	34.0
dir-R	0.700	4.0	0.979	52.0	0.920	17.4	1.000	34.0
dir-N	0.870	462.4	0.992	82.8	0.920	22.2	1.000	34.0

	monks-1		monks-3		nursery		postop	
Model	TC	Size	TC	Size	TC	Size	TC	Size
PDG	0.810	94.0	0.986	95.4	0.908	687.2	0.618	46.4
TAN	0.802	66.0	0.990	56.6	0.924	330.0	0.584	129.2
NB	0.750	23.0	0.972	23.0	0.903	99.0	0.722	47.0
KDB	0.759	589.4	0.861	589.4	0.967	7949.0	0.596	2099.6
c45-E	0.750	8.0	0.972	18.0	0.918	440.0	0.711	5.4
c45-R	0.817	57.6	0.993	26.0	0.938	1877.0	0.573	90.6
c45-N	0.976	114.0	1.000	28.0	0.974	4043.0	0.573	161.4
id3-E	0.750	8.0	1.000	24.0	0.918	440.0	0.711	5.4
id3-R	0.821	44.8	1.000	24.0	0.938	2003.0	0.596	66.0
id3-N	0.990	90.4	1.000	24.0	0.974	4047.0	0.550	165.6
dir-E	0.750	8.0	0.986	21.6	0.912	266.0	0.711	3.0
dir-R	0.863	30.8	1.000	25.6	0.921	434.0	0.711	3.0
dir-N	0.867	38.0	1.000	25.6	0.951	903.0	0.711	3.0

Cuadro 7: Resultados para los datos del repositorio UCI.

6. Conclusiones

En este trabajo hemos presentado el clasificador PDG, ampliando la clase de modelos gráficos que pueden ser usados en problemas de clasificación. Los experimentos llevados a cabo muestran que los PDGs construidos con el algoritmo de aprendizaje descrito en este artículo son competitivos en algunos problemas, principalmente los procedentes de funciones booleanas. En un futuro, esperamos ampliar el estudio mediante la construcción de clasificadores PDG a partir de transformaciones de redes bayesianas.

Referencias

[1] Abellán, J. y Moral, S. (2003). Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18:1215 – 1225.

[2] Bozga, M. y Maler, O. (1999). On the Representation of Probabilities over Structured Domains. En *Proceedings of the 11th International Conference on Computer Aided Verification*, páginas 261–273. Springer.

[3] Breiman, L., Friedman, J. H., Olshen, R. A., y J.Stone, C. (1993). *Classification and Regression Trees*. Chapman & Hall.

[4] Bryant, R. E. (1992). Symbolic boolean manipulation with ordered binary decision diagrams. *ACM Computing Surveys*, 24(3):293–318.

[5] Castillo, E., Gutiérrez, J. M., y Hadi, A. S. (1997). *Expert Systems and Probabilistic Network Models*. Springer-Verlag.

[6] Chow, C. K. y Liu, C. N. (1968). Approximating Discrete Probability Distributions with Dependence Trees. *IEE Transactions on Information Theory*, 14(3):462–467.

[7] Friedman, N., Geiger, D., y Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29:131–163.

[8] Jaeger, M. (2004). Probabilistic Decision Graphs - Combining Verification and AI Techniques for Probabilistic Inference. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12:19–42.

[9] Minsky, M. (1961). Steps toward artificial intelligence. *Transactions on Institute of Radio Engineers*, 46:8–30.

[10] Newman, D., Hettich, S., Blake, C., y Merz, C. (1998). UCI repository of machine learning databases:

[11] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1:81–106.

[12] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

[13] Sahami, M. (1996). Learning limited dependence bayesian classifiers. En *KDD-96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, páginas 335–338.

Jens D. Nielsen

Universidad de Almería, La Cañada de San Urbano s/n , 04120 Almería
dalgaard@ual.es

Rafael Rumí

Universidad de Almería, La Cañada de San Urbano s/n , 04120 Almería
rrumi@ual.es

Antonio Salmerón

Universidad de Almería, La Cañada de San Urbano s/n , 04120 Almería
antonio.salmeron@ual.es

Trabajo subvencionado por el Ministerio de Educación y Ciencia

a través del proyecto TIN2004-06204-C03-01