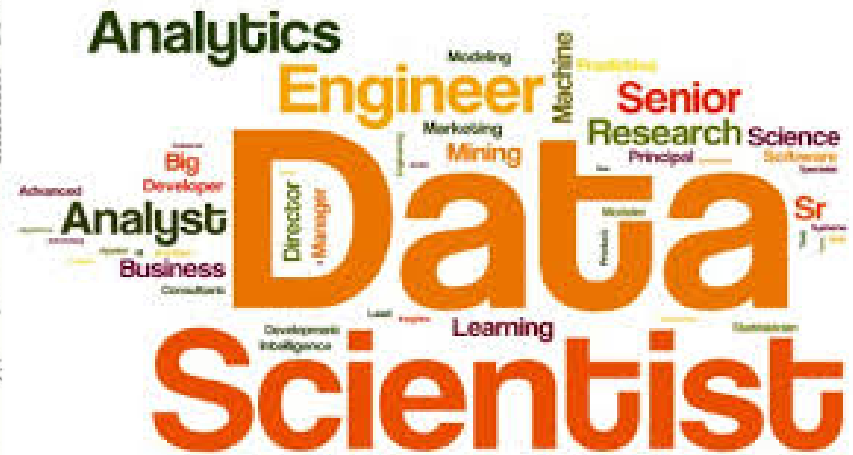
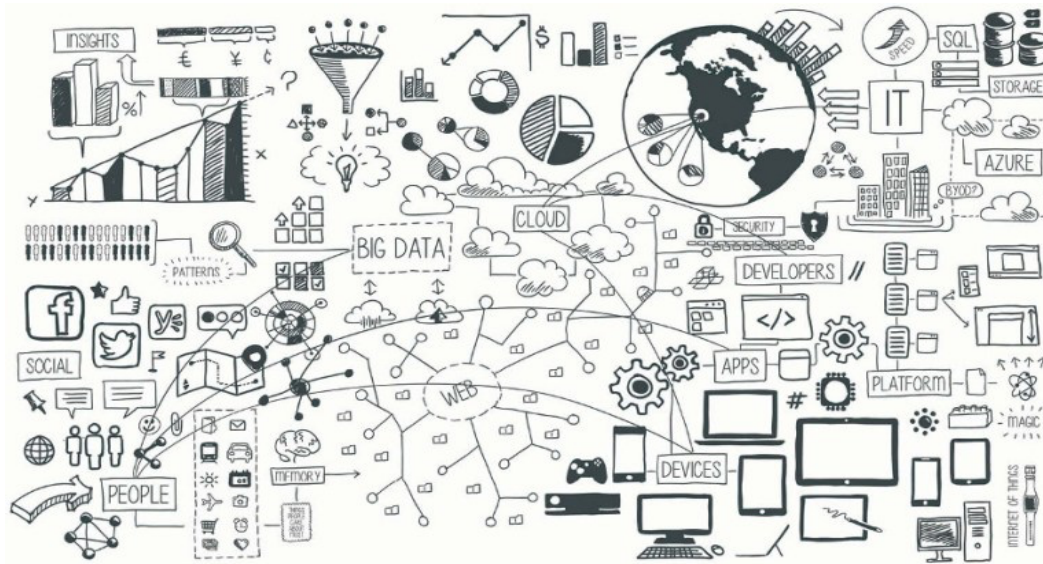


M1970 – Machine Learning II

Redes Probabilísticas Discretas



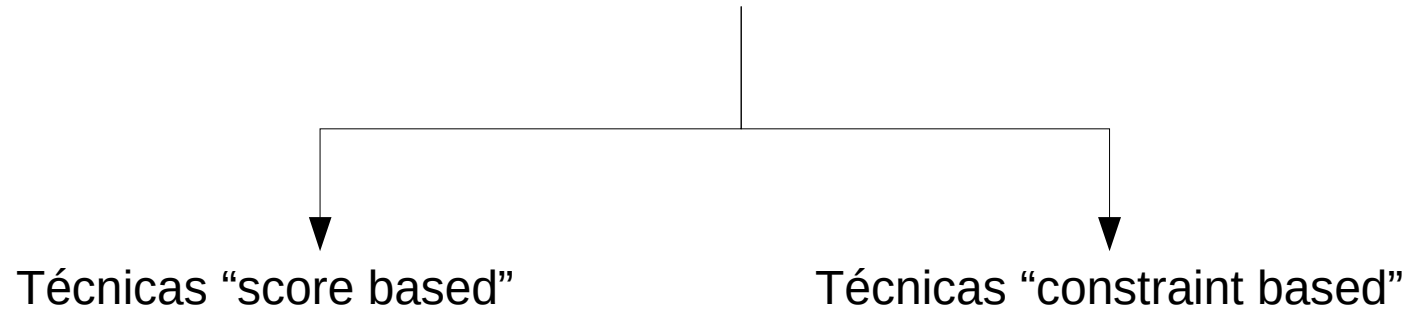
Sixto Herrera (sixto.herrera@unican.es)
José M. Gutiérrez, Mikel Legasa

Grupo de Meteorología
Univ. de Cantabria – CSIC
MACC / IFCA



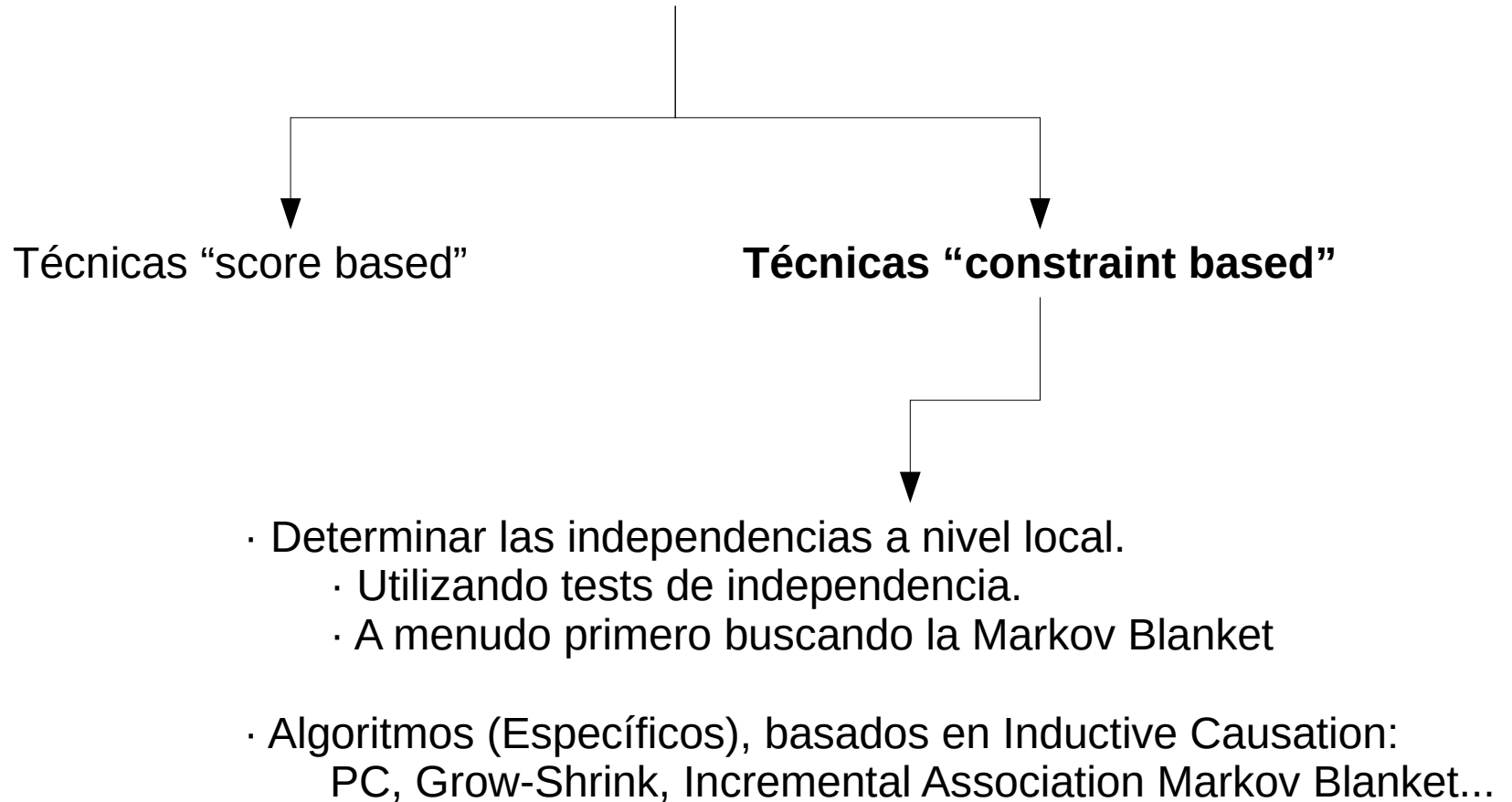
Aprendizaje Estructural:

Aprender la estructura (grafo) a partir del dataset



Aprendizaje Estructural:

Aprender la estructura (grafo) a partir del dataset



Algoritmo Inductive Causation

1) (Identificar pares dependientes)

1.1) Para cada par A, B buscar $S \subset X$ t.q. $A \perp B \mid S$.

1.2) Si $\nexists S$, colocar un arco no dirigido $A - B$

Algoritmo Inductive Causation

1) (Identificar pares dependientes)

1.1) Para cada par A, B buscar $S \subset X$ t.q. $A \perp B \mid S$.

1.2) Si $\nexists S$, colocar un arco no dirigido $A - B$

2) (Identificar v-estructuras)

2.1) Para cada par A, B no adyacentes con vecino común C .

2.2) Si $C \notin S$: dirigir $A - C, C - B$ como $A \rightarrow C \leftarrow B$.

2.3) Si $C \in S$: no hacer nada.

Algoritmo Inductive Causation

1) (Identificar pares dependientes)

1.1) Para cada par A, B buscar $S \subset X$ minimal t.q. $A \perp B \mid S$.

1.2) Si $\nexists S$, colocar un arco no dirigido $A - B$

2) (Identificar y construir v-estructuras)

2.1) Para cada par A, B no adyacentes con vecino común C .

2.2) Si $C \notin S$: dirigir $A - C, C - B$ como $A \rightarrow C \leftarrow B$.

2.3) Si $C \in S$: no hacer nada.

3) (Dirigir arcos) Repetir para los arcos no dirigidos:

3.1) Si $A - B$ y hay un camino estrictamente dirigido de A a B :

Fijar dirección de $A - B$ a $A \rightarrow B$.

3.2) Si A y B no son adyacentes pero $\exists C$ t.q $A \rightarrow C$ y $C - B$:

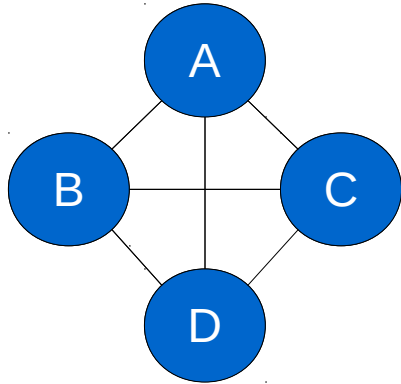
Fijar dirección de $C - B$ a $C \rightarrow B$.

4. Devolver el grafo (Partially Directed Acyclic Graph).

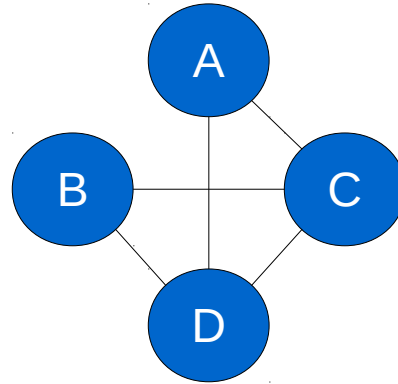
Algoritmo Inductive Causation

(todos dependientes)

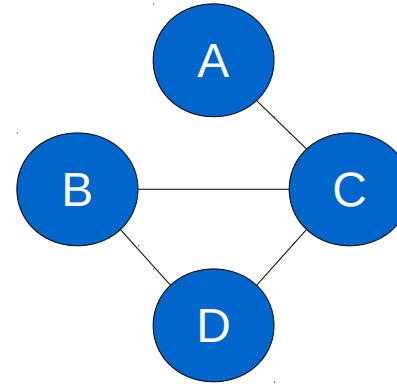
1)



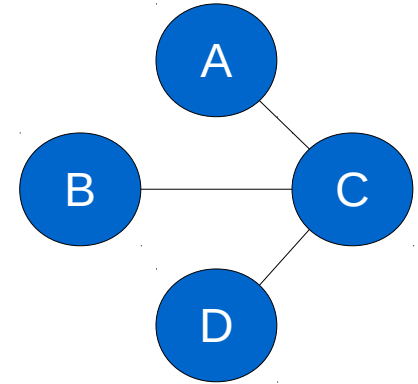
$A \perp B$



$A \perp D \mid C$



$B \perp D \mid C$



Algoritmo Inductive Causation

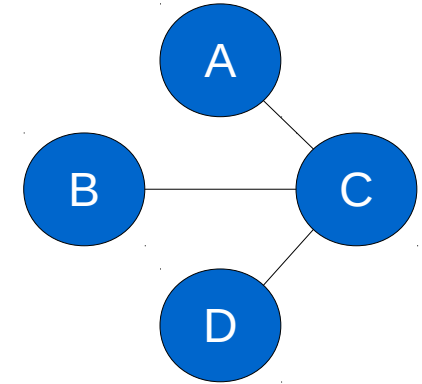
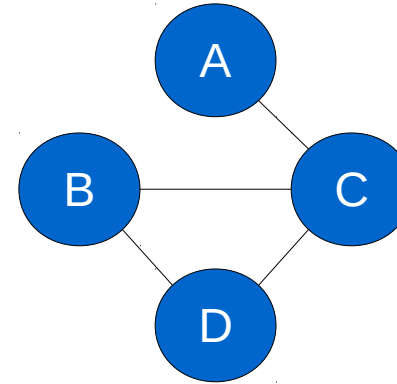
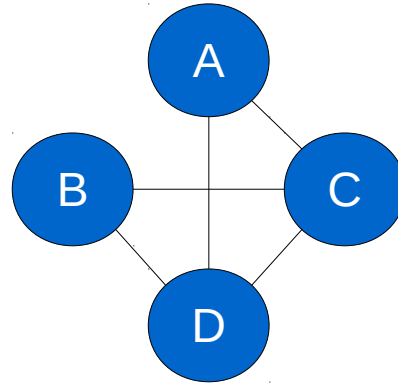
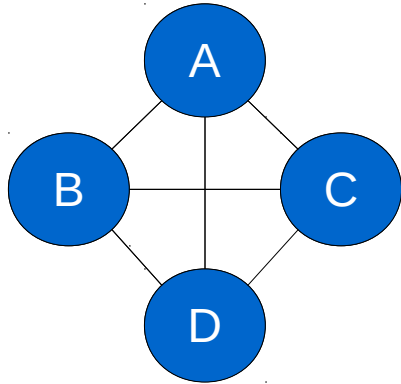
(todos dependientes)

$A \perp B$

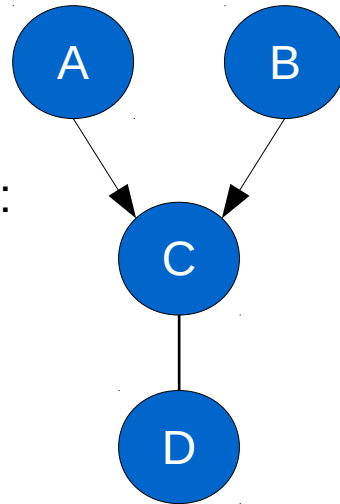
$A \perp D \mid C$

$B \perp D \mid C$

1)



2) V-estructura sobre C:



Algoritmo Inductive Causation

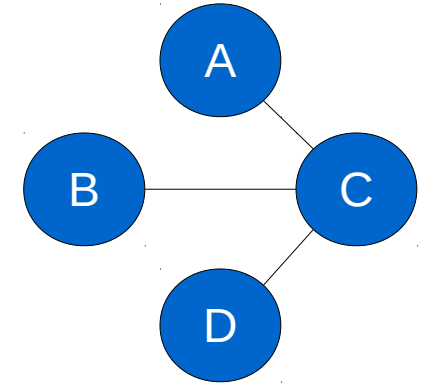
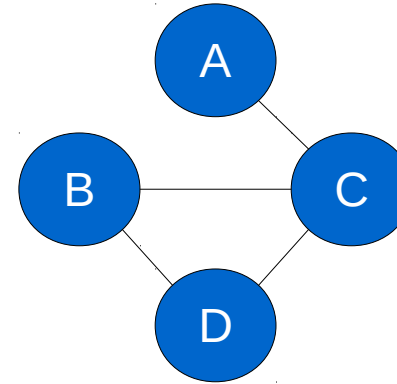
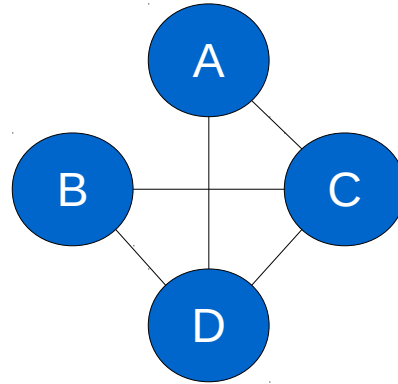
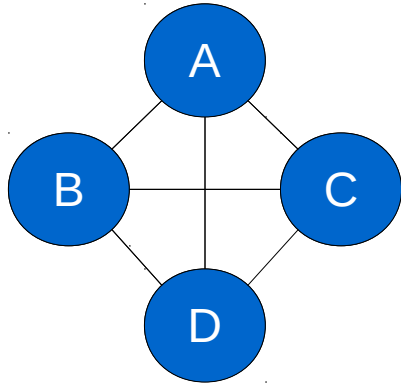
(todos dependientes)

$A \perp B$

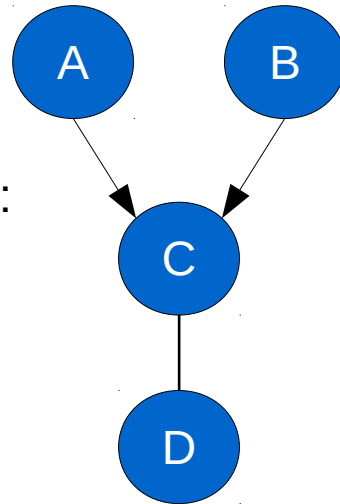
$A \perp D \mid C$

$B \perp D \mid C$

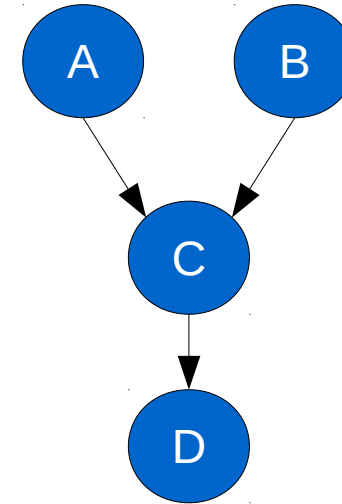
1)



2) V-estructura sobre C:



3.2)



Test de Independencia

Test χ^2 :

$$\chi^2(X, Y | \mathbf{Z}) = \sum_{i=1}^R \sum_{j=1}^C \sum_{k=1}^L \frac{(n_{ijk} - m_{ijk})^2}{m_{ijk}}, \quad \text{where} \quad m_{ijk} = \frac{n_{i+k} n_{+jk}}{n_{++k}}.$$

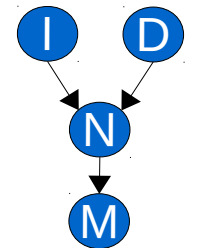
Para las L combinaciones de padres \mathbf{Z} y los R, C estados de X,Y, respectivamente

Otros :

Mutual Information, correlación...

Ejercicios:

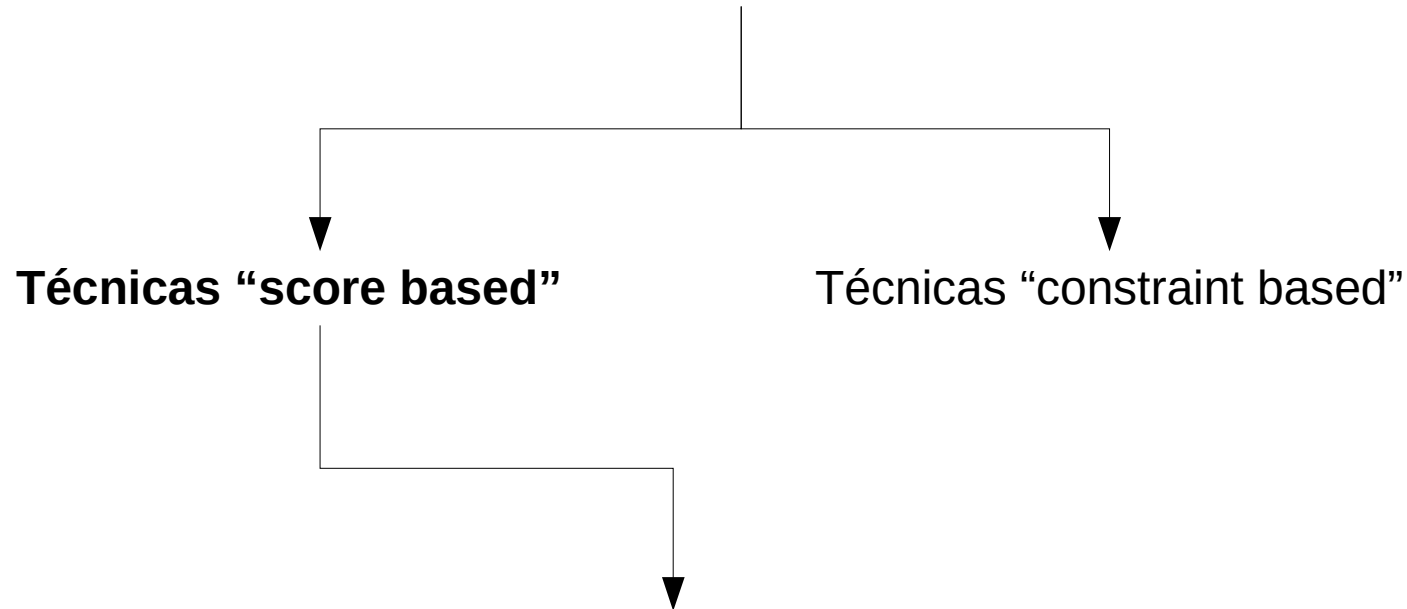
- 1) En el algoritmo Inductive Causation, ¿Por qué es necesario el paso 3.1?
- 2) En el algoritmo Inductive Causation, ¿Por qué es necesario el paso 3.2?
- 3) ¿Puede aplicarse Inductive Causation para un problema suficientemente grande?
- 4) Recorre manualmente los pasos del algoritmo IC para construir un DAG con 4 nodos: Inteligencia (I), Dificultad de Examen (D), Nota (N), Nota Media (M). Debe construir el DAG



- 5) **Bnlearn** implementa el algoritmo PC, función `pc.stable()`, aplícalo al dataset `survey` utilizando el argumento `debug = TRUE`, explora los argumentos `alpha`, `max.sx`.

Aprendizaje Estructural:

Aprender la estructura (grafo) a partir del dataset



- 1) Determinar "score", medida de bondad de ajuste.
- 2) Maximizar el "score".

· Algoritmos: (¡No específicos de RB!)
Cualquiera de optimización, e.g. hill-climbing.

Algoritmo Hill-Climbing

- 1) Inicializar grafo G (e.g. $G = \emptyset$)
- 2) $\text{maxscore} \leftarrow \text{score}(G)$

Algoritmo Hill-Climbing

- 1) Inicializar grafo G (e.g. $G = \emptyset$)
- 2) $\text{maxscore} \leftarrow \text{score}(G)$
- 3) Repetir **mientras** maxscore aumente:
 - 3.1) **Para cada** operador $\omega(G)$:
 - 3.1.1) $G' \leftarrow \omega(G)$
 - 3.1.2) **Si** $\text{score}(G') > \text{score}(G)$
 $\text{maxscore} \leftarrow \text{score}(G')$
 $G \leftarrow G'$
- 4) Devolver G (**Directed Acyclic Graph**)

Operadores $\omega()$:

- a) añadir arco
- b) borrar arco
- c) invertir arco

Score de bondad de ajuste

G que maximice $P(G \mid \mathcal{D})$

$$P(G \mid \mathcal{D}) = \frac{P(G, \mathcal{D})}{P(\mathcal{D})} = \frac{P(\mathcal{D} \mid G)P(G)}{P(\mathcal{D})}.$$

Score de bondad de ajuste

G que maximice $P(G \mid \mathcal{D})$

$$P(G \mid \mathcal{D}) = \frac{P(G, \mathcal{D})}{P(\mathcal{D})} = \frac{P(\mathcal{D} \mid G)P(G)}{P(\mathcal{D})}.$$

$$P(\mathcal{D} \mid G) = \int P(\mathcal{D} \mid G, \Theta)P(\Theta \mid G) d\Theta =$$

$$\prod_{i=1}^p \left(\int P(X_i \mid \pi(X_i), \Theta(X_i))P(\Theta(X_i) \mid \pi(X_i)) d\Theta(X_i) \right).$$

$\Theta(X_i)$: Parámetros CPT

Score de bondad de ajuste, estimándolo...

G que maximice

$$\text{BIC}(G, \mathcal{D}) = \sum_{i=1}^p \left[\log \Pr(X_i \mid \Pi_{X_i}) - \frac{|\Theta_{X_i}|}{2} \log n \right]$$

Con:

$$|\mathcal{D}| = n.$$

$|\Theta_{X_i}|$ el número de parámetros.

Score de bondad de ajuste, estimándolo...

G que maximice $P(G \mid \mathcal{D})$

Ejemplo:

De:

$$\Pr(A, S, E, O, R, T) = \Pr(A) \Pr(S) \Pr(E \mid A, S) \Pr(O \mid E) \Pr(R \mid E) \Pr(T \mid O, R)$$

Obtenemos:

Score de bondad de ajuste, estimándolo...

G que maximice $P(G \mid \mathcal{D})$

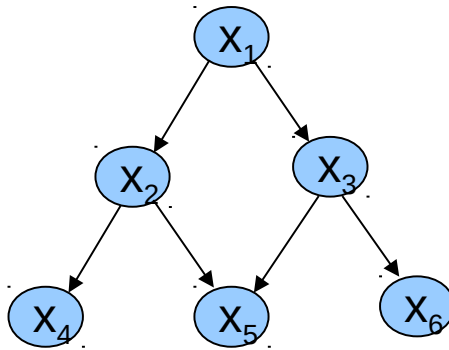
Ejemplo:

De:

$$\Pr(A, S, E, O, R, T) = \Pr(A) \Pr(S) \Pr(E \mid A, S) \Pr(O \mid E) \Pr(R \mid E) \Pr(T \mid O, R)$$

Obtenemos:

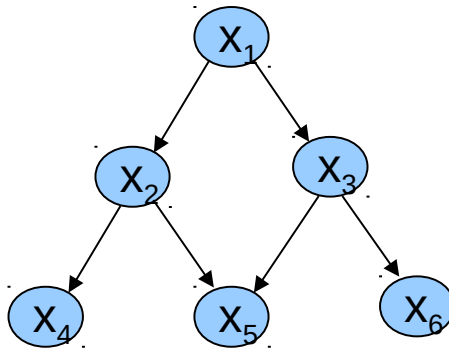
$$\begin{aligned} \text{BIC} &= \log \widehat{\Pr}(A, S, E, O, R, T) - \frac{d}{2} \log n = \\ &= \left[\log \widehat{\Pr}(A) - \frac{d_A}{2} \log n \right] + \left[\log \widehat{\Pr}(S) - \frac{d_S}{2} \log n \right] + \\ &+ \left[\log \widehat{\Pr}(E \mid A, S) - \frac{d_E}{2} \log n \right] + \left[\log \widehat{\Pr}(O \mid E) - \frac{d_O}{2} \log n \right] + \\ &+ \left[\log \widehat{\Pr}(R \mid E) - \frac{d_R}{2} \log n \right] + \left[\log \widehat{\Pr}(T \mid O, R) - \frac{d_T}{2} \log n \right] \end{aligned}$$



Score de bondad de ajuste, estimándolo...

x_1	$p(x_1)$
0	0.3
1	0.7

$$p(0,1,1,1,0,0) = p(x_1=0)$$

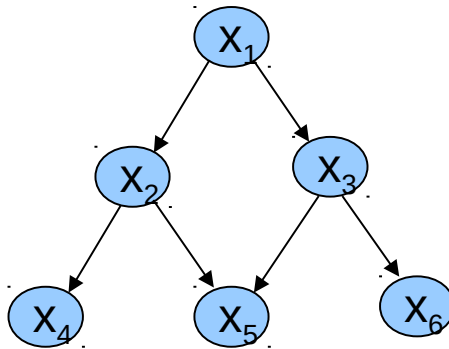


Score de bondad de ajuste, estimándolo...

x_1	$p(x_1)$
0	<u>0.3</u>
1	0.7

x_1	x_2	$p(x_2 x_1)$
0	0	0.4
<u>0</u>	<u>1</u>	<u>0.6</u>
1	0	0.1
1	1	0.9

$$p(0,1,1,1,0,0) = p(x_1=0)p(x_2=1|x_1=0)$$



Score de bondad de ajuste, estimándolo...

x_1	$p(x_1)$
0	<u>0.3</u>
1	0.7

x_1	x_2	$p(x_2 x_1)$	x_1	x_3	$p(x_3 x_1)$	x_2	x_4	$p(x_4 x_2)$	x_3	x_6	$p(x_6 x_3)$
0	0	0.4	0	0	0.2	0	0	0.3	0	0	0.1
<u>0</u>	<u>1</u>	<u>0.6</u>	<u>0</u>	<u>1</u>	<u>0.8</u>	0	1	0.7	0	1	0.9
1	0	0.1	1	0	0.5	1	0	0.2	<u>1</u>	<u>0</u>	<u>0.4</u>
1	1	0.9	1	1	0.5	<u>1</u>	<u>1</u>	<u>0.8</u>	1	1	0.6

x_2	x_3	x_5	$p(x_5 x_2, x_3)$
0	0	0	0.4
0	0	1	0.6
0	1	0	0.5
0	1	1	0.5
1	0	0	0.7
1	0	1	0.3
<u>1</u>	<u>1</u>	<u>0</u>	<u>0.2</u>
1	1	1	0.8

$$\begin{aligned}
 &p(0,1,1,1,0,0) = \\
 &p(x_1=0)p(x_2=1|x_1=0)p(x_3=1|x_1=0)p(x_4=1|x_2=1) \\
 &p(x_6=0|x_3=1)p(x_5=0|x_2=1, x_3=1) =
 \end{aligned}$$

$$0.3 \times 0.6 \times 0.8 \times 0.8 \times 0.4 \times 0.2 = \mathbf{0.009216}$$

Ejercicios:

5) En los algoritmos score-based se utilizan tres operadores para los arcos: adición, borrado e inversión. ¿Por qué es necesario este último?

6) Explica el término final en la expresión del BIC:

$$\text{BIC}(G, \mathcal{D}) = \sum_{i=1}^p \left[\log \Pr(X_i \mid \Pi_{X_i}) - \frac{|\Theta_{X_i}|}{2} \log n \right]$$

¿Qué pasaría si este término no existiera?

7) **Bnlearn** implementa el algoritmo hill-climbing, función *hc()*, aplícalo al dataset survey utilizando el argumento *debug = TRUE*. Explora los argumentos *start*, *maxp* y *k*.

Referencias de ampliación:

D. Koller, N. Friedman: *Probabilistic Graphical Models*

D. Heckerman, D. Geiger, D. M. Chickering: *Learning Bayesian Networks: The combination of Knowledge and Statistical Data*