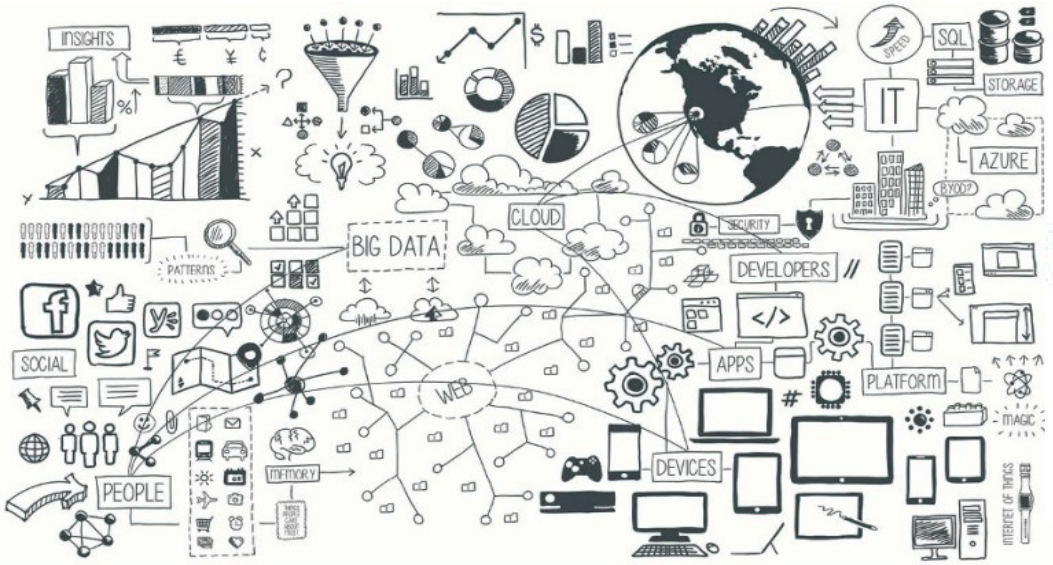
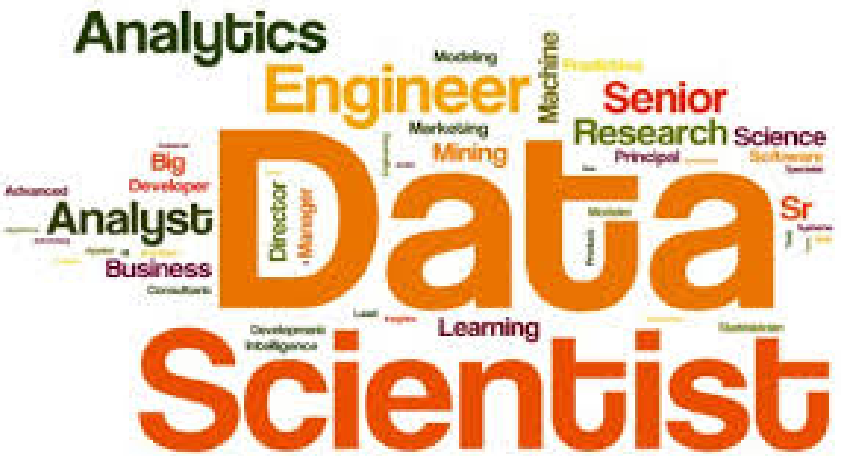
M1970 – Machine Learning II
Redes Probabilísticas Discretas





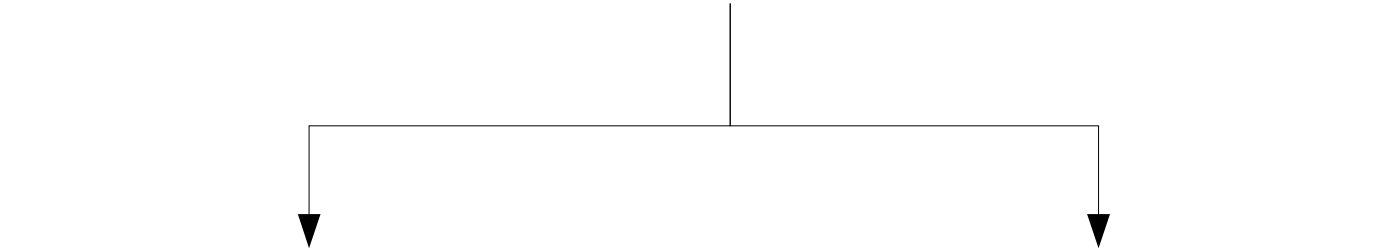
Sixto Herrera (sixto.herrera@unican.es)
José M. Gutiérrez, Mikel Legasa

Grupo de Meteorología

Univ. de Cantabria – CSIC
MACC / IFCA

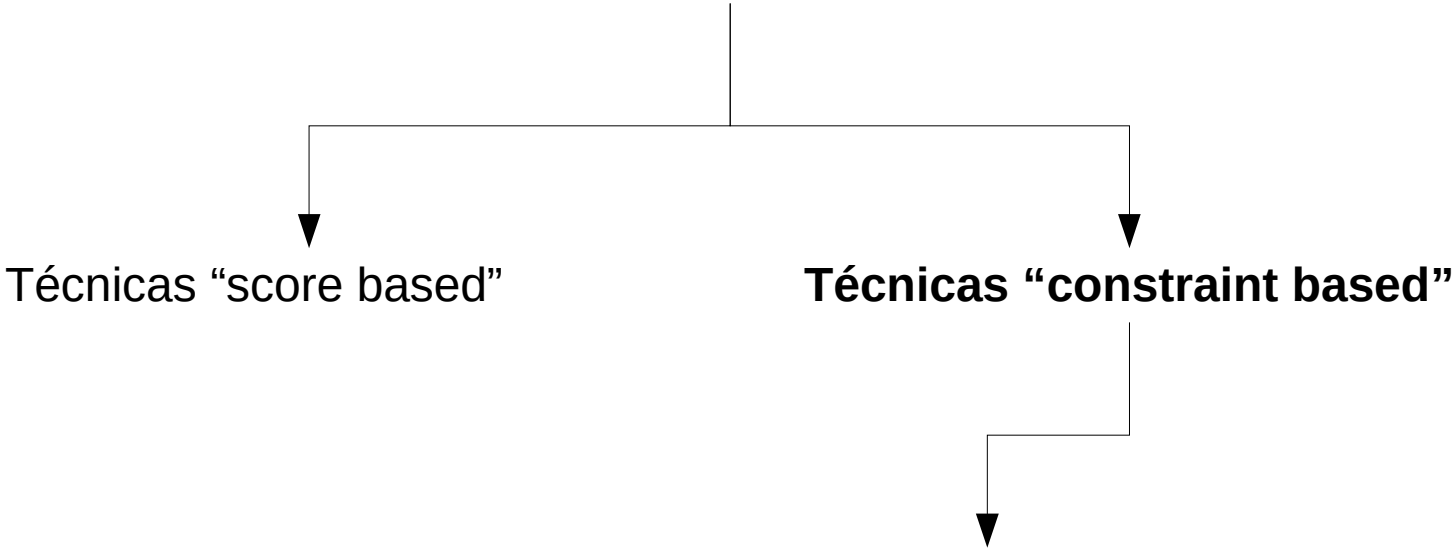
Aprendizaje Estructural:

Aprender la estructura (grafo) a partir del dataset



Aprendizaje Estructural:

Aprender la estructura (grafo) a partir del dataset



· Determinar las independencias a nivel local.
  · Utilizando tests de independencia.
  · A menudo primero buscando la Markov Blanket

· Algoritmos (Específicos), basados en Inductive Causation:
  PC, Grow-Shrink, Incremental Association Markov Blanket...

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

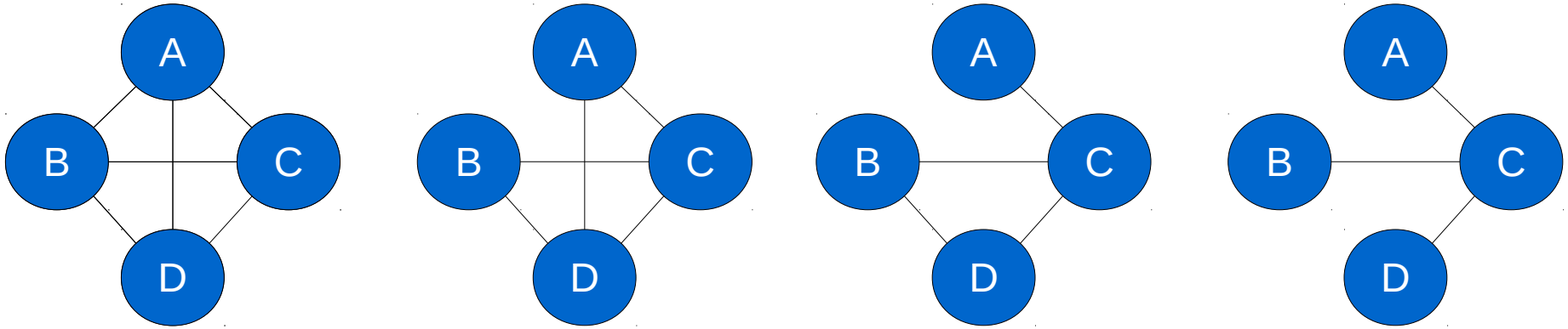
(todos dependientes)

$A \perp B$

$A \perp D \mid C$

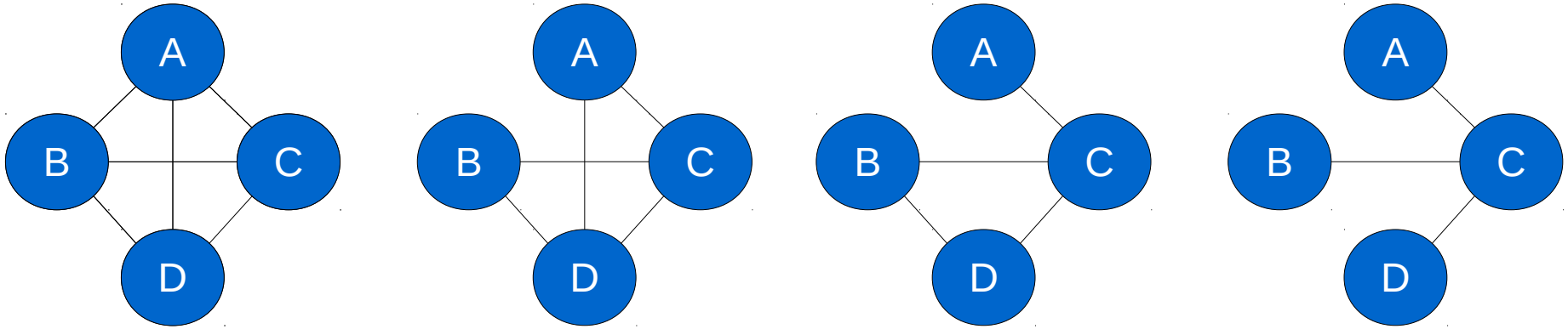
$B \perp D \mid C$

1)

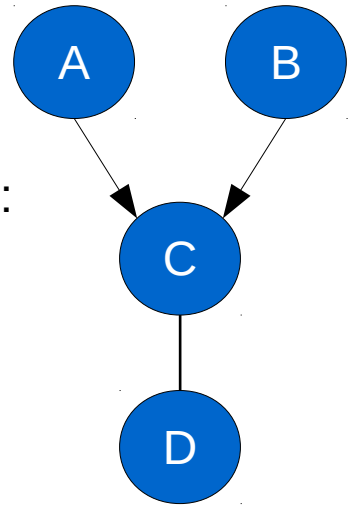


Algoritmo Inductive Causation

1) (todos dependientes) $A \perp B$ $A \perp D \mid C$ $B \perp D \mid C$



2) V-estructura sobre C:



Algoritmo Inductive Causation

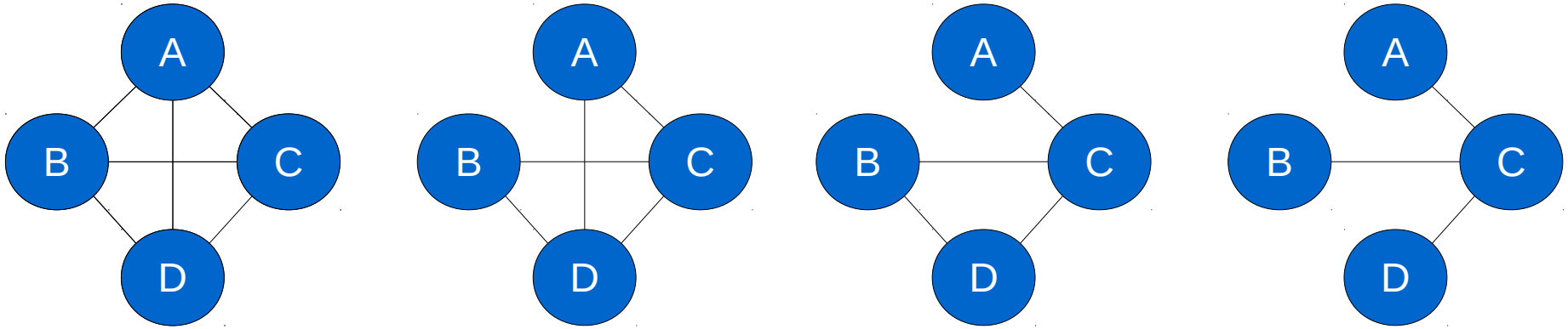
(todos dependientes)

$A \perp B$

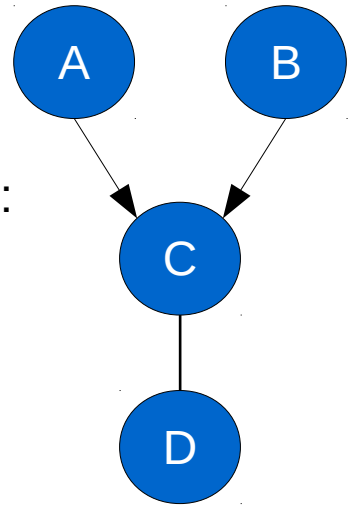
$A \perp D \mid C$

$B \perp D \mid C$

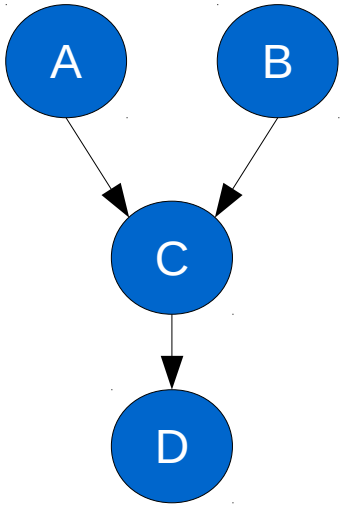
1)



2) V-estructura sobre C:



3.2)



Test de Independencia

Test χ^2 :

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

Para las L combinaciones de padres **Z** y los R, C estados de X,Y, respectivamente

Otros :

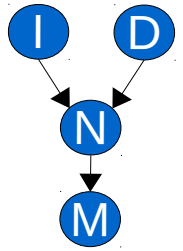
Mutual Information, correlación...

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?

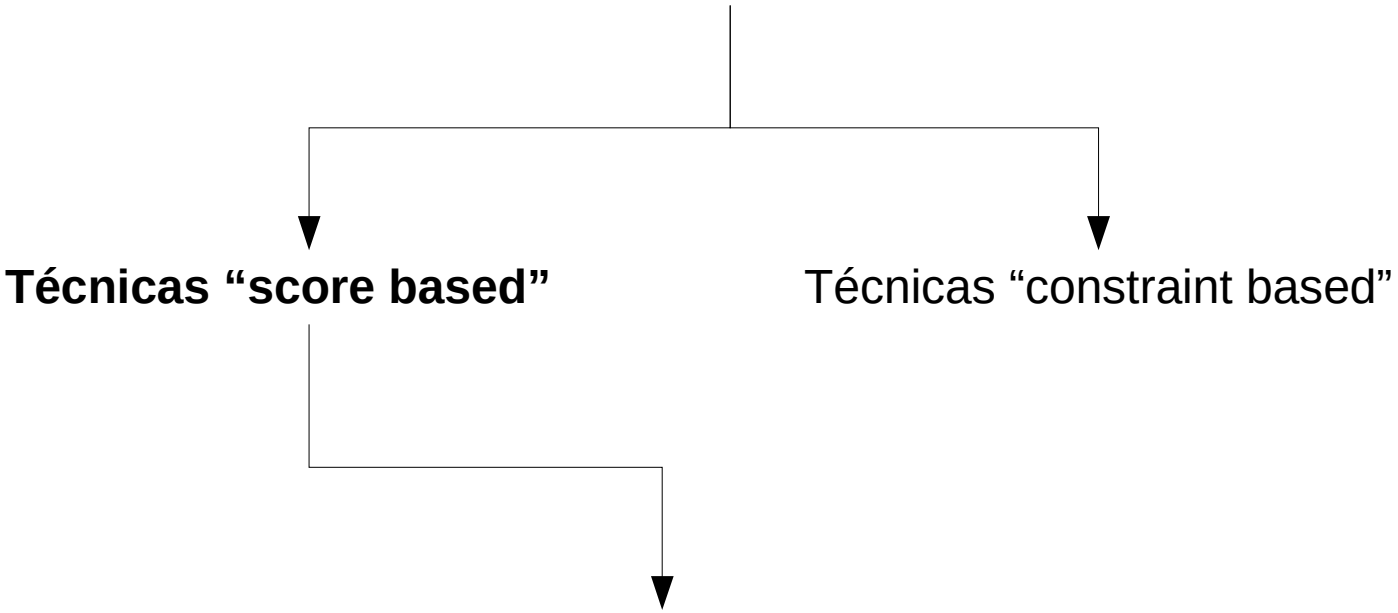
4) Recorre manualmente los pasos del algoritmo IC para construir un DAG con 4 nodos: Inteligencia (I), Dificultad de Examen (D), Nota (N), Nota Media (M). Debe construir el DAG 

.

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 = Ø)

2) maxscore ← score(G)

Algoritmo Hill-Climbing

1) Inicializar grafo G (e.g. G = Ø)

2) maxscore ← score(G)

3) Repetir **mientras** maxscore aumente:
   3.1) **Para cada** operador ω(G):
      3.1.1) G' ← ω(G)
      3.1.2) **Si** score(G') > score(G)
         maxscore ← score(G')
         G ← G'

4) Devolver G **(Directed Acyclic Graph)**

Operadores ω():
- 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 $P(G \mid \mathcal{D})$

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

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

¿Cómo calcularlo?

Utilizando la factorización del propio grafo

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

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

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:

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

Obtenemos:

Score de bondad de ajuste, estimándolo...

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

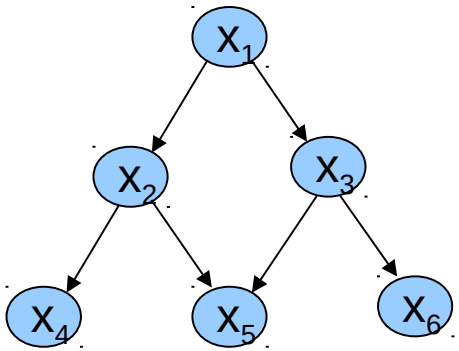
Ejemplo:

De:

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

Obtenemos:

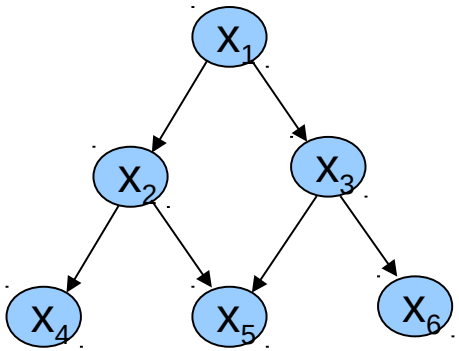
$$\begin{aligned}
\mathrm{BIC} &= \log \widehat{\Pr}(\mathtt{A}, \mathtt{S}, \mathtt{E}, \mathtt{O}, \mathtt{R}, \mathtt{T}) - \frac{d}{2} \log n = \\
&= \left[\log \widehat{\Pr}(\mathtt{A}) - \frac{d_{\mathtt{A}}}{2} \log n\right] + \left[\log \widehat{\Pr}(\mathtt{S}) - \frac{d_{\mathtt{S}}}{2} \log n\right] + \\
&+ \left[\log \widehat{\Pr}(\mathtt{E} \mid \mathtt{A}, \mathtt{S}) - \frac{d_{\mathtt{E}}}{2} \log n\right] + \left[\log \widehat{\Pr}(\mathtt{O} \mid \mathtt{E}) - \frac{d_{\mathtt{O}}}{2} \log n\right] + \\
&+ \left[\log \widehat{\Pr}(\mathtt{R} \mid \mathtt{E}) - \frac{d_{\mathtt{R}}}{2} \log n\right] + \left[\log \widehat{\Pr}(\mathtt{T} \mid \mathtt{O}, \mathtt{R}) - \frac{d_{\mathtt{T}}}{2} \log n\right]
\end{aligned}$$



Score de bondad de ajuste, estimándolo...

x_1	$p(x_1)$
0	0.3
1	0.7

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

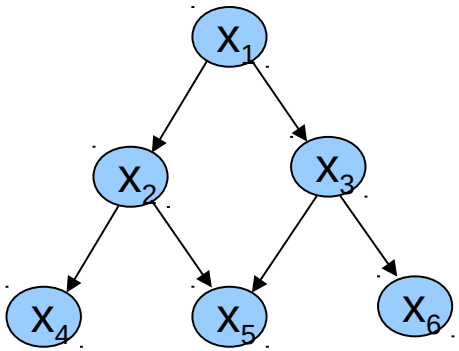


Score de bondad de ajuste, estimándolo...

x_1	$p(x_1)$
0	0.3
1	0.7

x_1	x_2	$p(x_2\|x_1)$
0	0	0.4
0	1	0.6
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	0.3
1	0.7

x_1	x_2	$p(x_2\|x_1)$
0	0	0.4
0	1	0.6
1	0	0.1
1	1	0.9

x_1	x_3	$p(x_3\|x_1)$
0	0	0.2
0	1	0.8
1	0	0.5
1	1	0.5

x_2	x_4	$p(x_4\|x_2)$
0	0	0.3
0	1	0.7
1	0	0.2
1	1	0.8

x_3	x_6	$p(x_6\|x_3)$
0	0	0.1
0	1	0.9
1	0	0.4
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
1	1	0	0.2
1	1	1	0.8

$p(0,1,1,1,0,0) =$
$p(x_1=0)p(x_2=1|x_1=0)p(x_3=1|x_1=0)p(x_4=1|x_2=1)$
$p(x_6=0|x_3=1)p(x_5=0|x_2=1,x_3=1) =$

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

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:

¿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*