Redes Bayesianas

CC52A - Inteligencia Artificial

Gonzalo Ríos D.

DCC - UChile

Otoño 2009

Dado un vector de variables aleatorias X=$(x_1, ..., x_n)$, tenemos una medida de probabilidades conjunta:

$$\Pr : dom(X) \rightarrow [0, 1]$$

donde $dom(X) = dom(x_1) \times ... \times dom(x_n)$. Si conocemos la probabilidad conjunta, podemos calcular cualquier probabilidad sobre las variables $x_1 ... x_n$

Proposición

- *Regla de Probabilidad Condicional:* $\Pr(X|Y) = \frac{\Pr(X,Y)}{\Pr(Y)}$
- *Regla de Marginación:* $\Pr(A) = \sum_{i \in I} \Pr(A, B_i)$, B_i *disjuntos,* $\bigcup_{i \in I} B_i = \Omega$

El siguiente teorema muestra una simple pero poderosa relación entre probabilidades condicionales, que será la base de nuestra teoría.

Teorema

Teorema de Bayes

$$\Pr(C = c|X = x) = \frac{\Pr(X = x|C = c) * \Pr(C = c)}{\Pr(X = x)}$$

- $\Pr(C = c|X = x)$: *Posterior*
- $\Pr(X = x|C = c)$: *Verosimilitud*
- $\Pr(C = c)$: *Prior*
- $\Pr(X = x)$: *Evidencia*

Definición

- *Se dice que X,Y va's son independientes ssi* $\forall x, y \;\; \Pr(X = x, Y = y) = \Pr(X = x) * \Pr(Y = y)$
- *Se dice que X,Y va's son independientes dada la evidencia E ssi* $\Pr(X, Y|E) = \Pr(X|E) * \Pr(Y|E)$

Proposición

- *X e Y son indep. ssi* $\Pr(X|Y) = \Pr(X)$
- *X e Y son indep. dada la evidencia E ssi* $\Pr(X|Y, E) = \Pr(X|E)$

La independencia entre variables permite reducir la complejidad de la función de probabilidades conjunta, y en vez de modelar una única función, la separamos en partes más simples.

- Supongamos que tenemos datos de la forma $(X_1, ..., X_n, C)$, donde C es la variable de la clase, y deseamos predecir el valor de la clase para un vector $(x_1, ..., x_n)$.
- El enfoque probabilistico asignará la clase más probable, es decir:

$$\bar{c} = \arg\max_c \Pr(C = c | X_1 = x_1, ..., X_n = x_n)$$

- Luego, si aplicamos el teorema de Bayes, obtenemos

$$\bar{c} = \arg\max_c \frac{\Pr(X_1 = x_1, ..., X_n = x_n | C = c) * \Pr(C = c)}{\Pr(X_1 = x_1, ..., X_n = x_n)}$$

- Luego, para cada clase c_i, basta modelar las funciones

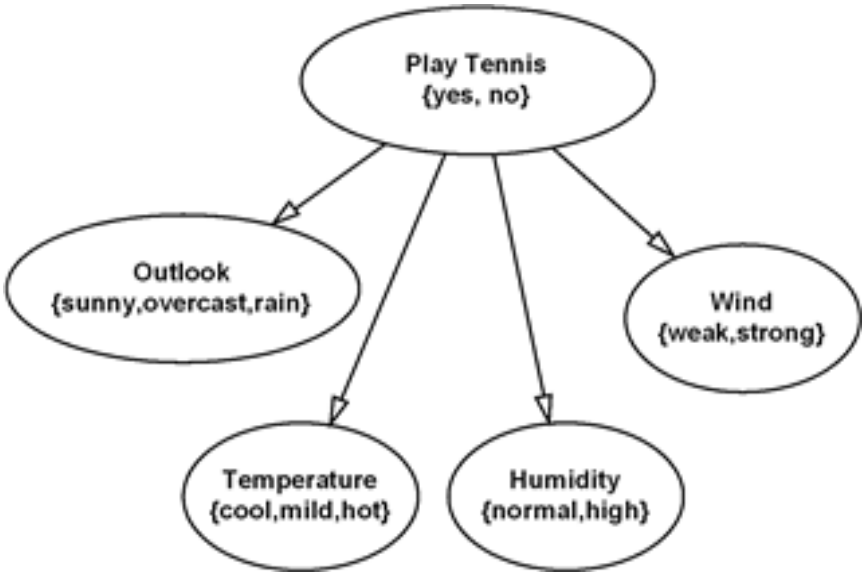
$$g_i(x_1, ..., x_n) = \Pr(X_1 = x_1, ..., X_n = x_n | C = c_i) * \Pr(C = c_i)$$

- Supongamos que las variables X_i son binarias.
- Luego para estimar $\Pr(X_1 = x_1, ..., X_n = x_n | C = c_i) * \Pr(C = c_i)$, necesitaremos 2^n parámetros.
- Podemos ver que el problema más simple se convierte en exponencial.
- Si las variables son discretas o continuas, el problema se vuelve más complejo aún.
- Pero si las variables son independientes, entonces el problema se simplifica significativamente:

$$\begin{aligned} \Pr(X_1 &= x_1, ..., X_n = x_n | C = c_i) \\ &= \Pr(X_1 = x_1 | C = c_i) * ... * \Pr(X_n = x_n | C = c_i) \end{aligned}$$

- Luego, si las variables X_i son binarias, bajo el supuesto de independencia, necesitaremos n parámetros, es decir, el problema se vuelve lineal.
- Este modelo se conoce como Bayes Naive

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

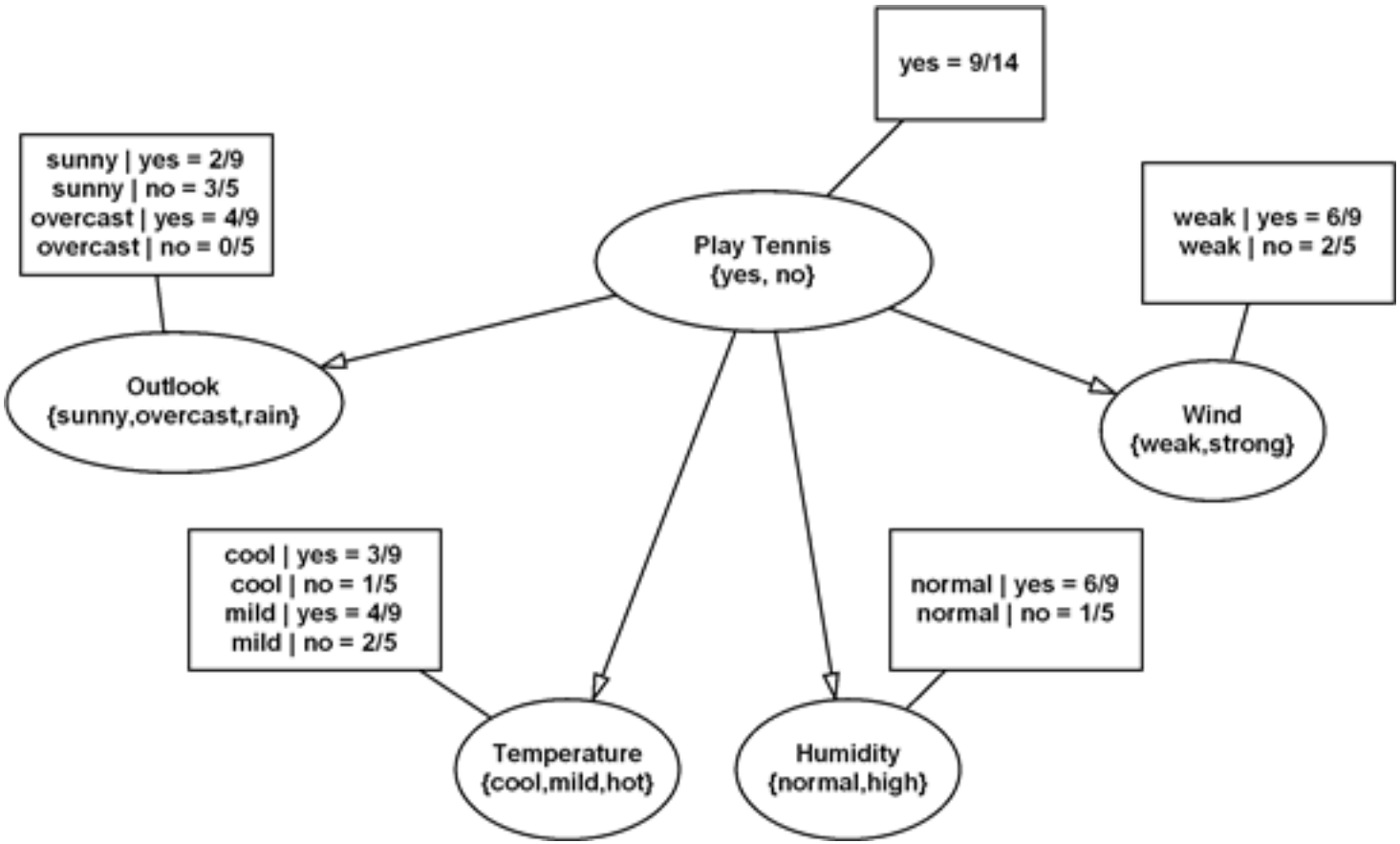


Los cantidad de parámetros que debemos estimar es 13, versus los 71 sin el supuesto de independencia.

La estimación más simple es por frecuencias:

- $\Pr(Play = yes) = \frac{9}{14}$
- $\Pr(Outlook = sunny | Play = yes) = \frac{2}{9}$
- $\Pr(Outlook = overcast | Play = yes) = \frac{4}{9}$
- $\Pr(Outlook = sunny | Play = no) = \frac{3}{5}$
- $\Pr(Outlook = overcast | Play = no) = \frac{0}{5}$
- $\Pr(Temperature = cold | Play = yes) = \frac{3}{9}$
- $\Pr(Temperature = mild | Play = yes) = \frac{4}{9}$
- $\Pr(Temperature = cold | Play = no) = \frac{1}{5}$
- $\Pr(Temperature = mild | Play = no) = \frac{2}{5}$
- $\Pr(Humidity = normal | Play = yes) = \frac{6}{9}$
- $\Pr(Humidity = normal | Play = no) = \frac{1}{5}$
- $\Pr(Wind = weak | Play = yes) = \frac{6}{9}$
- $\Pr(Wind = weak | Play = no) = \frac{2}{5}$

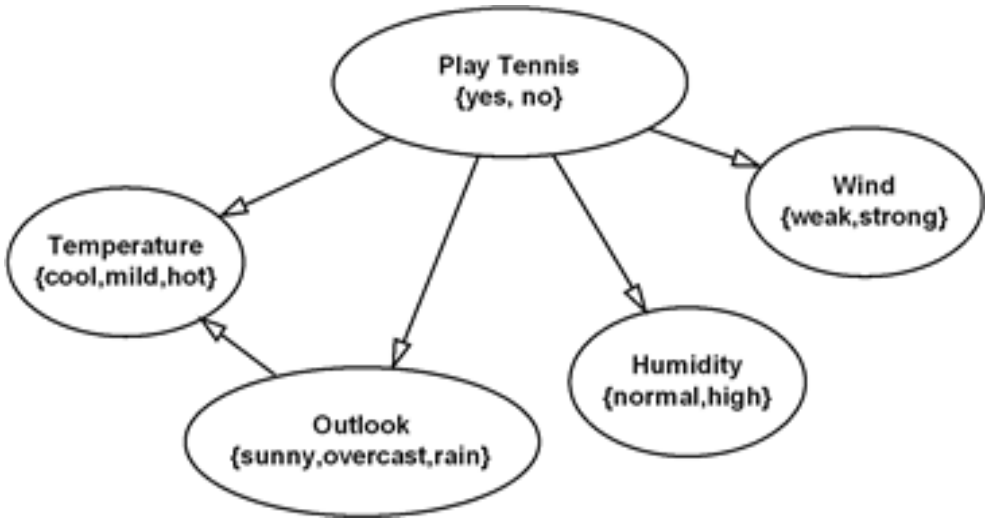
Red Bayesiana Naive



Redes Bayesianas

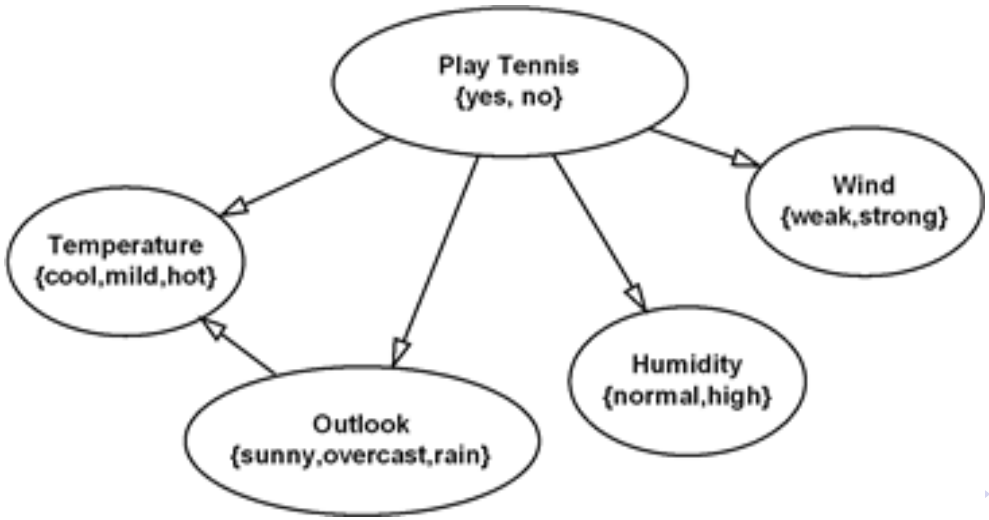
Suposiciones de Independencia Erróneas

- Recordemos que hicimos la suposición de independencia, ¿qué sucede si esto no es verdad para todas las variables?
- Podriamos encontrar ciertas dependencias entre las variables, como por ejemplo, entre Outlook y Temperature



- $\Pr(Outlook = rain, Temperature = hot | Play = y) = 0$
- $\Pr(Outlook = rain | Play = y) * \Pr(Temperature = hot | Play = y) = \frac{3}{9} * \frac{2}{9} \neq 0$, luego no son independientes

- Las Redes Bayesianas son un tipo de modelos denominados "modelos de grafos", que codifican eficientemente la probabilidad conjunta, evitando suposiciones de independencia erróneas.
- Una Red Bayesiana consta de dos partes:
  - Grafo dirigido acíclico (DAG) que contiene un nodo por variable (G).
  - Tablas de probabilidades condicionales (TPC), que almacenan los parámetros del modelo (P).
- Los arcos entre nodos indican dependencia entre las variables.



Proposición

Regla de la Cadena

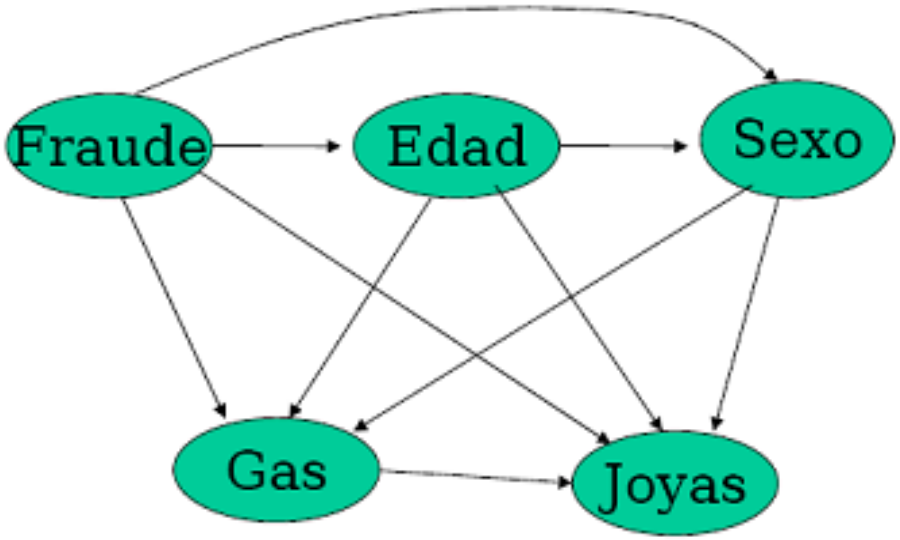
$$\Pr(x_1, ..., x_n) = \prod_{i=1}^{n} \Pr(x_i | x_1, ..., x_{i-1})$$

Para cada orden de las variables, podemos reformular Pr *usando la regla de la cadena. Tenemos* $n!$ *reformulaciones distintas.*

Usando el orden F,E,S,G,J tenemos

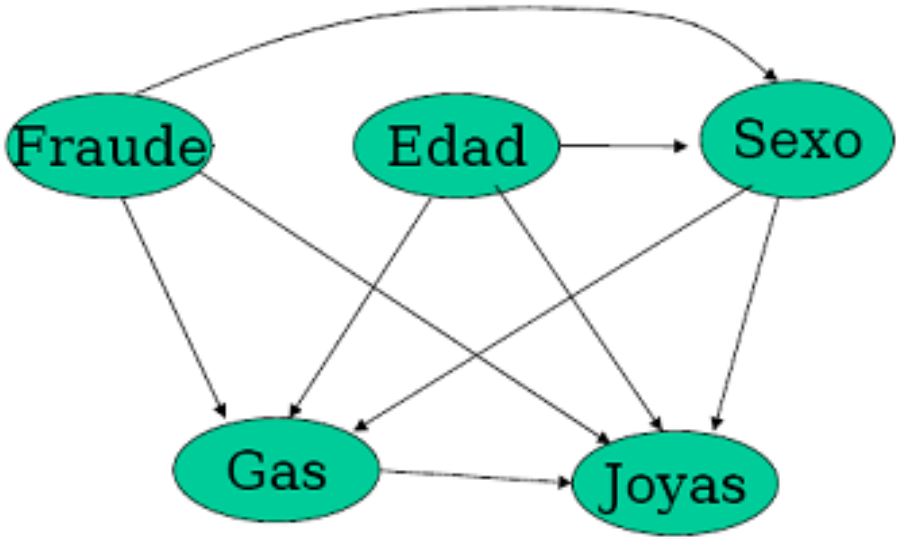
$$\Pr(f, e, s, g, j) = \Pr(f) \Pr(e|f) \Pr(s|f, e) \Pr(g|f, e, s) \Pr(j|f, e, s, g)$$

Usando el orden F,E,S,G,J tenemos



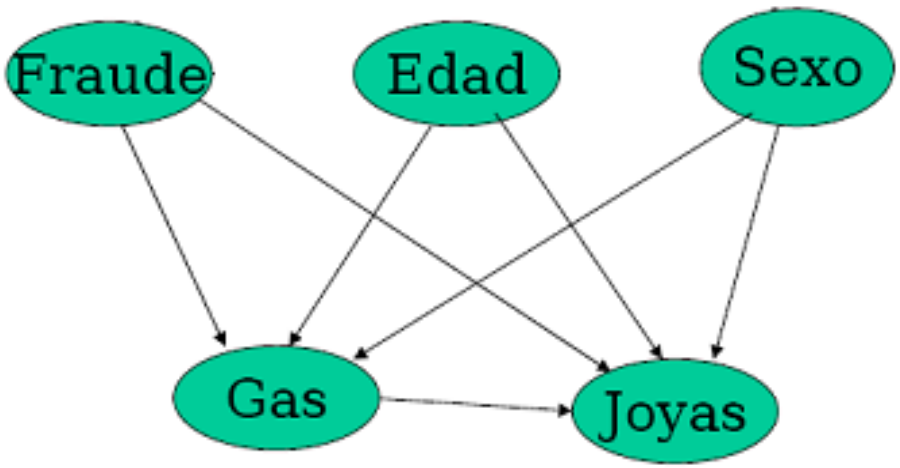
La independencia de algunos variables nos lleva a una expresión más simple de codificar. Un buen ordenamiento hará uso de estas independencias, mientras que ordenamientos ineficientes no.

$\Pr(e|f) = \Pr(e)$



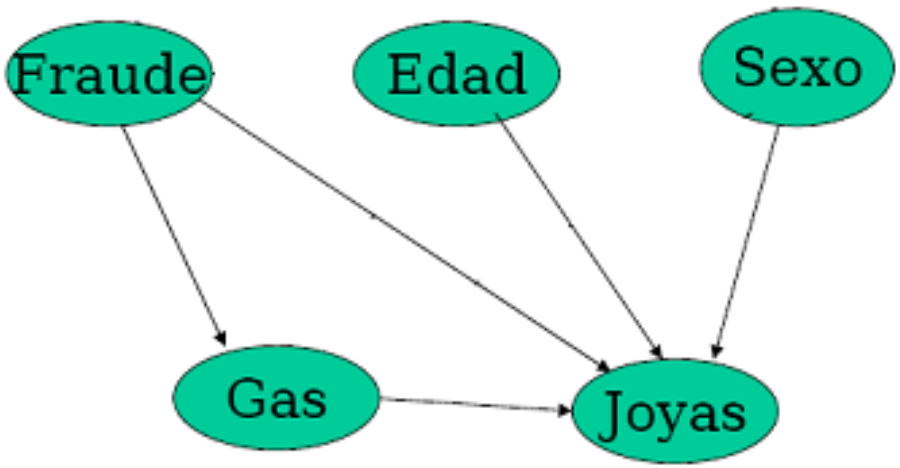
E y F son independientes

$\Pr(s|f,e) = \Pr(s)$



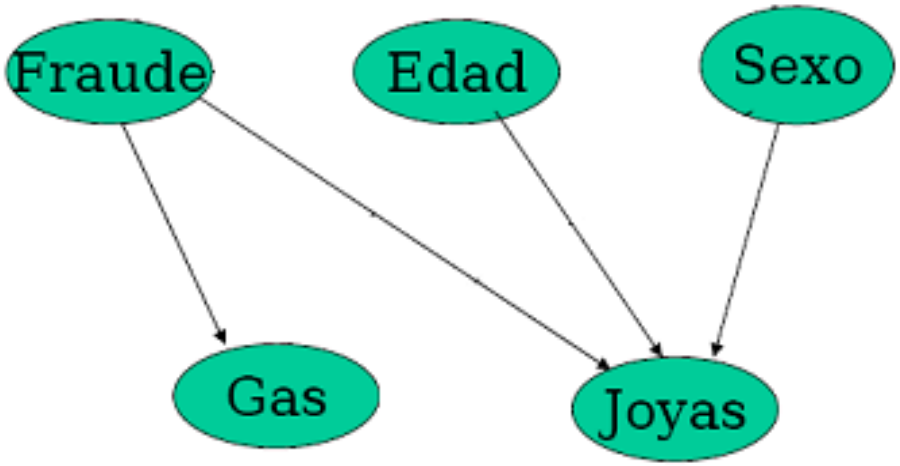
S es independiente de F,E

$$\Pr(g|f,e,s) = \Pr(g|f)$$



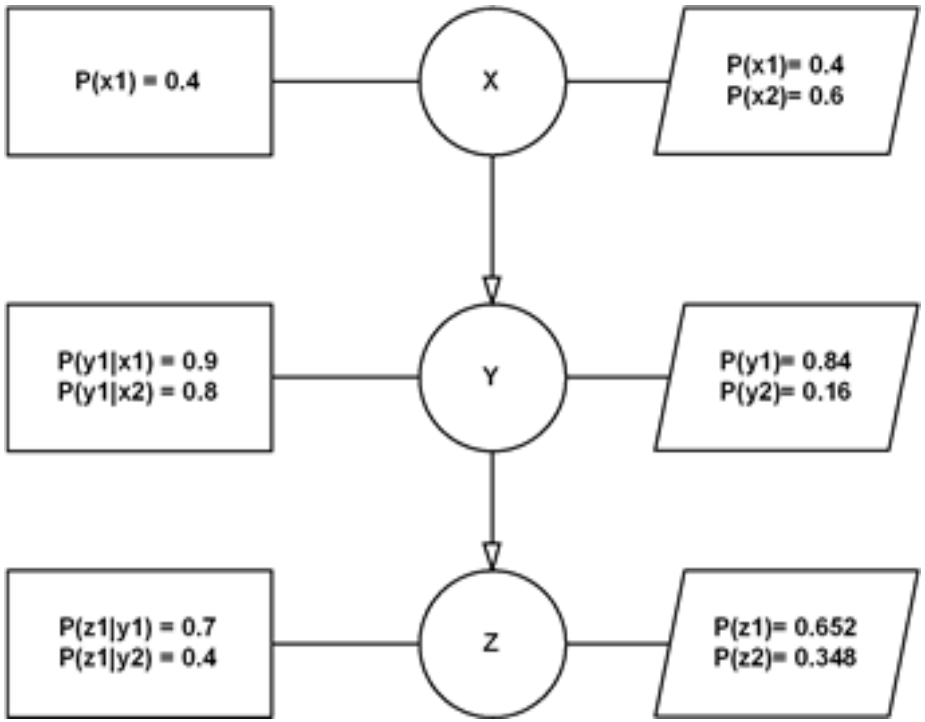
G es independiente de E,S dado F

$$\Pr(j|f,e,s,g) = \Pr(j|f,e,s)$$



J es independiente de G dado F,E,S
Podemos ver que este ordenamiento es eficiente. Concidere el ordenamiento J,G,S,E,F y verifique que no se puede eliminar ningún arco de la red bayesiana.

Como vimos, una red bayesiana es una forma eficiente de codificar la probabilidad conjunta de un conjunto de variables, pero a nosotros nos interesa inferir distintas probabilidades.



$$\Pr(y_1) = \Pr(y_1|x_1) * \Pr(x_1) + \Pr(y_1|x_2) * \Pr(x_2)$$
$$= 0.9 * 0.4 + 0.8 * 0.6 = 0.84$$

Pero que sucede si nosotros conocemos algunos de los datos?

- Si sabemos que Y=y_1, entonces $\Pr^*(y_1) = \Pr(y_1|y_1) = 1$ y $\Pr^*(y_2) = 0$
- Luego, $\Pr^*(z_1) = \Pr(z_1|y_1) = 0.7$
- Entonces, $\Pr^*(z_2) = 0.3$
- $\Pr^*(x_1) = \Pr(x_1|y_1) = \frac{\Pr(y_1|x_1)*\Pr(x_1)}{\Pr(y_1)} = \frac{0.9*0.4}{0.84} = 0.428\,57$

Al instanciar una variable, la información debe propagarse por la red. Se puede observar que la información fluye a los hijos y a los padres de la variable instanciada, de forma distinta:

- A los hijos se aplica la regla de marginación
- A los padres se aplica el teorema de bayes

Veamos como realizar este flujo de información en redes más complejas.

Definición

Sea (G, P) una red bayesiana donde G=(V,E) es un árbol. Sea a un conjunto de instancias de un subconjunto $A \subseteq V$. Para cada variable X definimos:

1. $\lambda -$ *mensaje*
   Para cada hijo Y de X : $\lambda_Y(x) = \sum_y \Pr(y|x)\lambda(y)$
2. $\lambda -$ *valor*
   1. *Si $X \in A$ y el valor de X es $\bar{x}$:* $\lambda(\bar{x})=1$, $\lambda(x)=0$, $\forall x \neq \bar{x}$
   2. *Si $X \notin A$, y X es una hoja:* $\lambda(x)=1$
   3. *Si $X \notin A$, y X no es una hoja:* $\lambda(x) = \prod_{U \in hijos(X)} \lambda_U(x)$
3. $\pi -$ *mensaje*
   Si Z es el padre de X : $\pi_X(z) = \pi(z) \prod_{U \in hijos(Z)\setminus\{X\}} \lambda_U(z)$

Definición

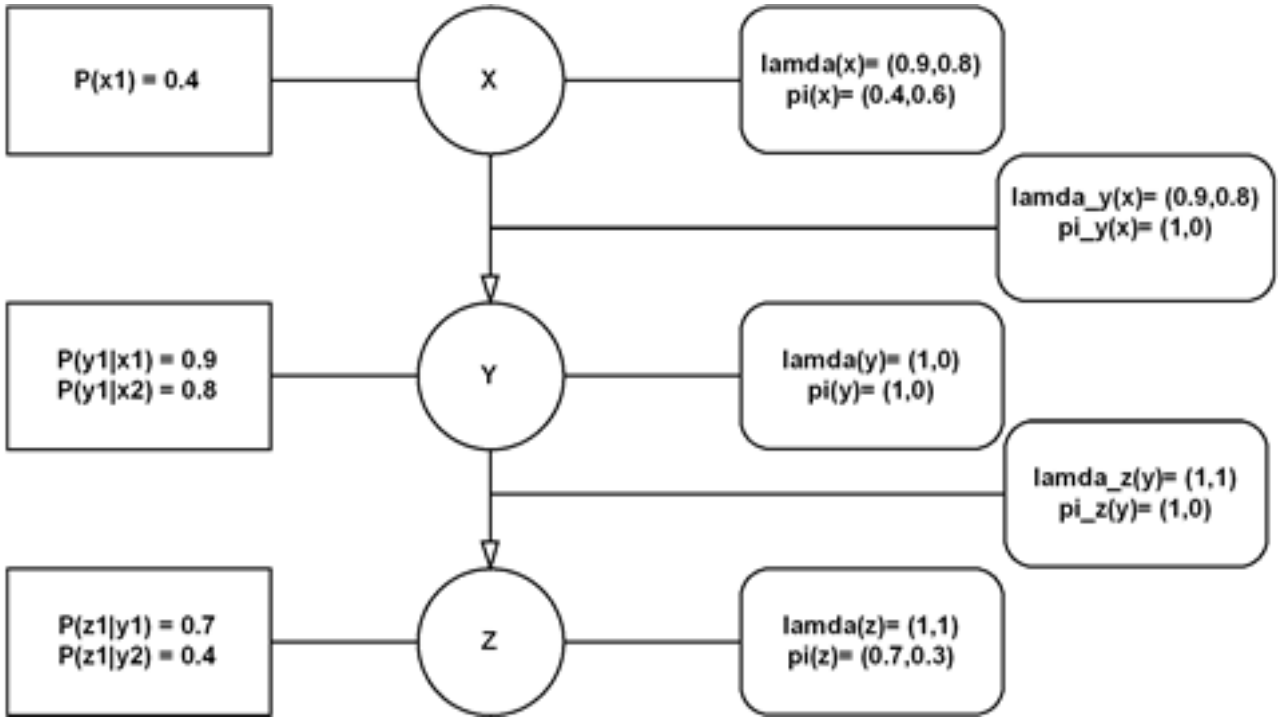
4. $\pi - valor$
   1. *Si* $X \in A$ *y el valor de* X *es* $\bar{x}$: $\pi(\bar{x})=1$, $\pi(x)=0$, $\forall x \neq \bar{x}$
   2. *Si* $X \notin A$, *y* X *es la raiz:* $\pi(x) = \Pr(x)$
   3. *Si* $X \notin A$, *y* X *no es la raiz:* $\pi(x) = \sum_z \Pr(x|z)\pi_X(z)$

Teorema

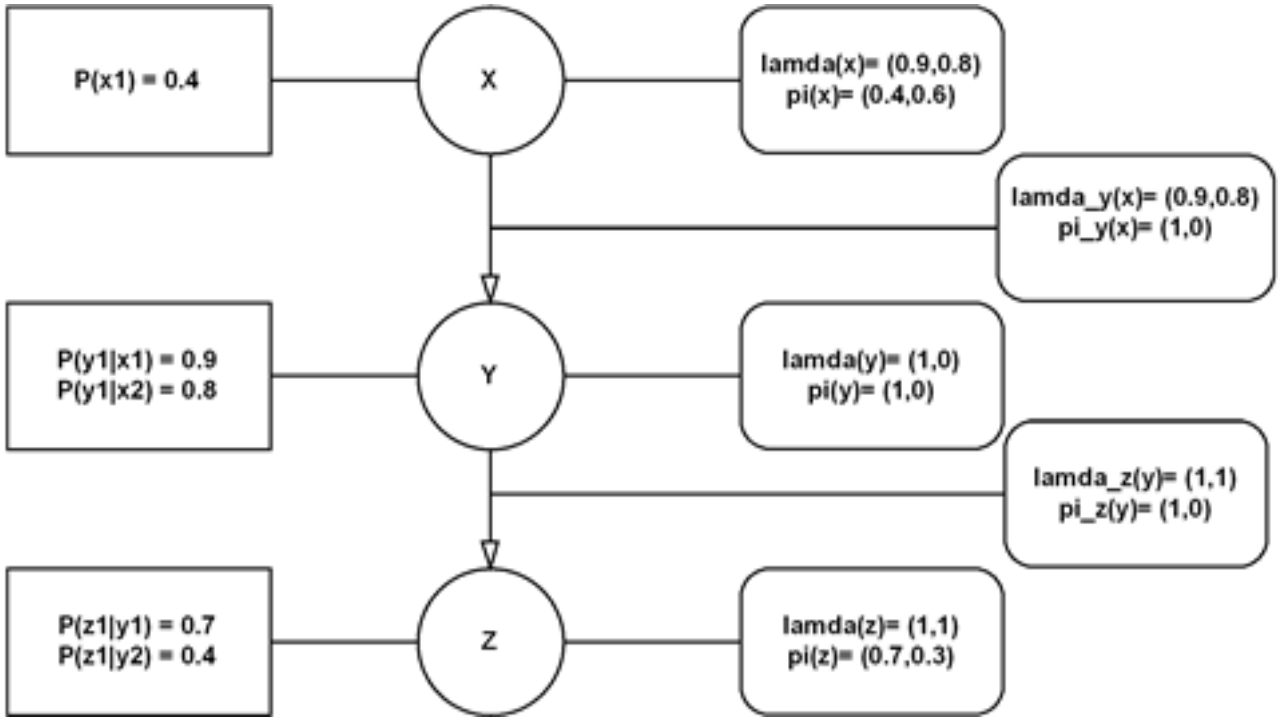
Dadas las definiciones anteriores, tenemos que para cada variable X

$$\Pr^*(x) = \Pr(x|a) = \alpha\lambda(x)\pi(x)$$

donde α *es una constante de normalización.*

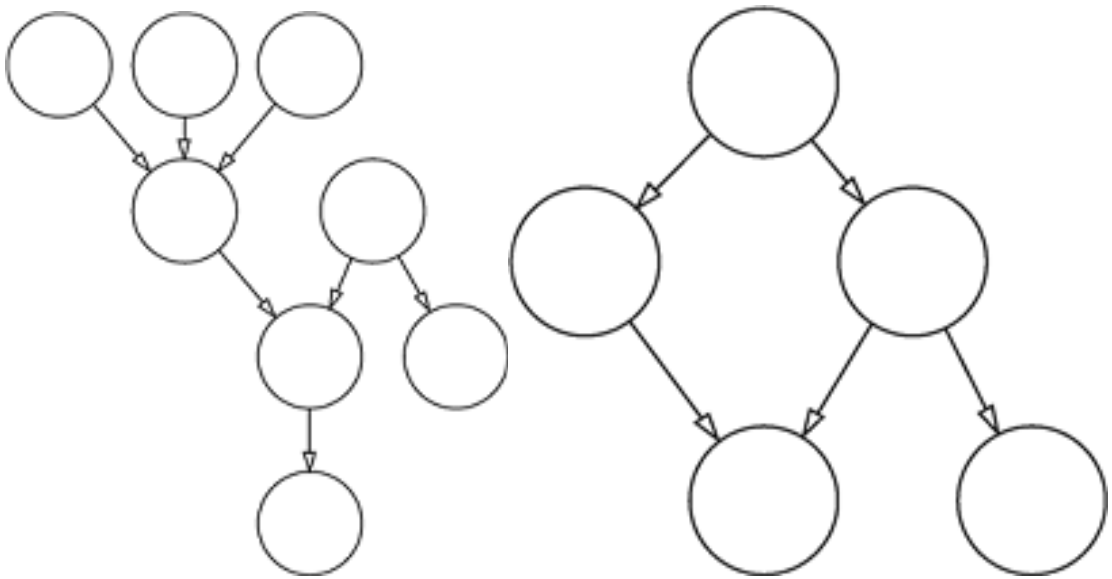


- A=Y, a={y1}
- $\Pr^*(x_1) = \alpha\lambda(x_1)\pi(x_1) = \alpha * 0.9 * 0.4 = \alpha 0.36$
- $\Pr^*(x_2) = \alpha * 0.8 * 0.6 = \alpha 0.48$
- $\alpha 0.36 + \alpha 0.48 = 0.84\alpha = 1 \Longrightarrow \alpha = \frac{1}{0.84}$



- $\Pr^*(x_1) = \frac{0.36}{0.84} = 0.428\,57, \;\; \Pr^*(x_2) = \frac{0.48}{0.84} = 0.571\,43$
- $\Pr^*(y_1) = 1, \;\; \Pr^*(y_2) = 0$
- $\Pr^*(z_1) = \beta 0.7, \;\; \Pr^*(z_2) = \beta 0.3 \Longrightarrow \beta = 1$
- $\Pr^*(z_1) = 0.7, \;\; \Pr^*(z_2) = 0.3$

El teorema anterior solo funciona para el caso que el DAG de la red bayesiana sea un árbol, pero esto no sucede en muchos casos.



Definición

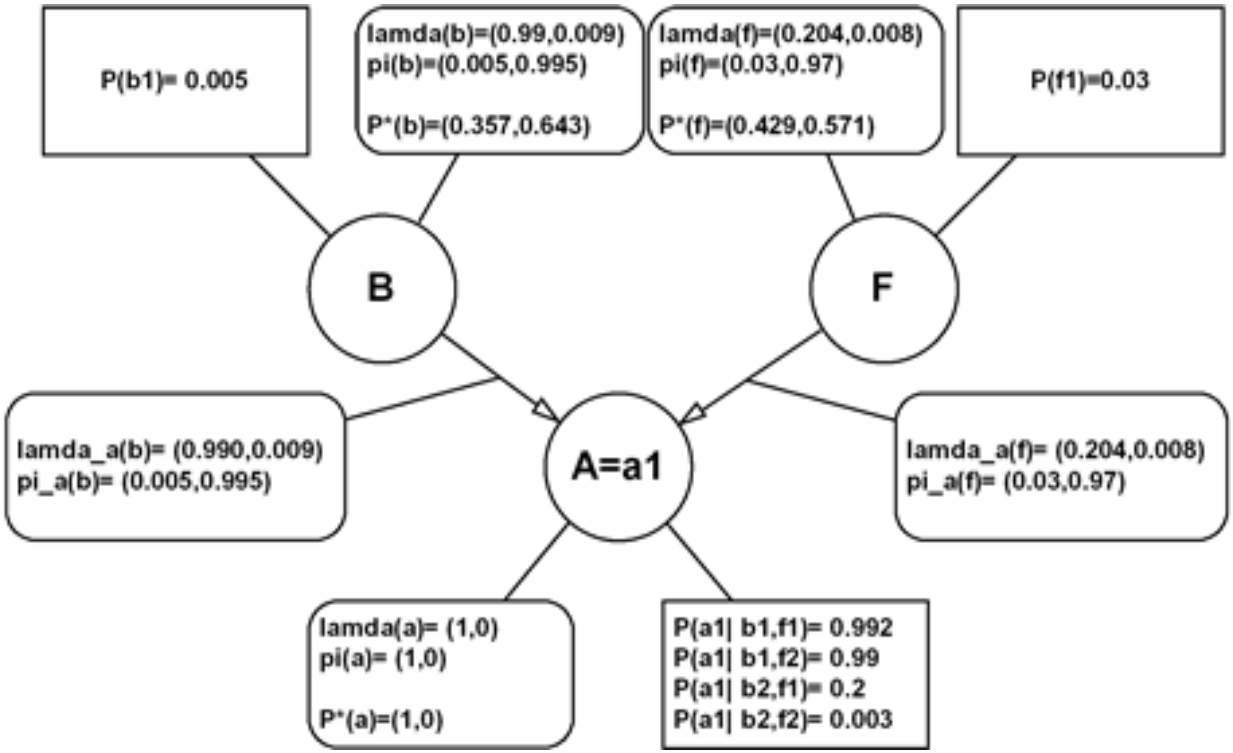
Una red se dice simplemente conectada si, para todo par de nodos existe a lo más una cadena que los conecta. En caso contrario, se dice que la red es multiplemente conectada.

Teorema

Para el caso de redes simplemente conectadas, se debe modificar los $\lambda -$ *mensaje y* $\pi -$ *valor :*

Definición

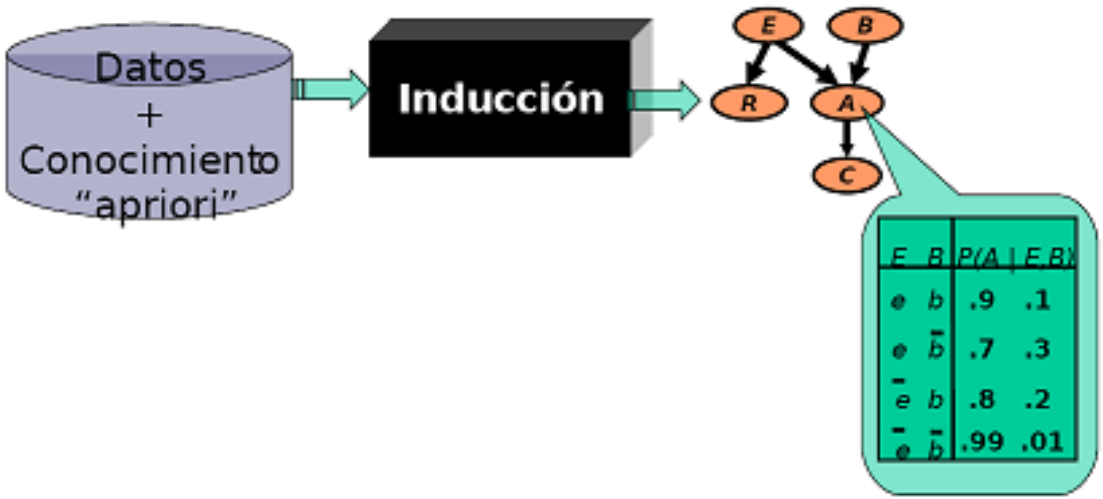
1. $\lambda -$ *mensaje*
   Para cada hijo Y *de* X*, donde* $W_1, .., W_k$ *son los otros padres de* Y *:*
   $$\lambda_Y(x) = \sum_y \left[\sum_{w1,...,w_k} \left(\Pr(y|x, w_1, ..., w_k) \prod_{i=1}^{k} \pi_Y(w_i) \right) \right] \lambda(y)$$
2. $\pi -$ *valor*
   Si $X \notin A$*,* X *no es una raiz y* $Z_1, ..., Z_j$ *son los padres de* X*:*
   $$\pi(x) = \sum_{z_1,..,z_j} \left(\Pr(x|z_1, .., z_j) \prod_{i=1}^{j} \pi_X(z_i) \right)$$

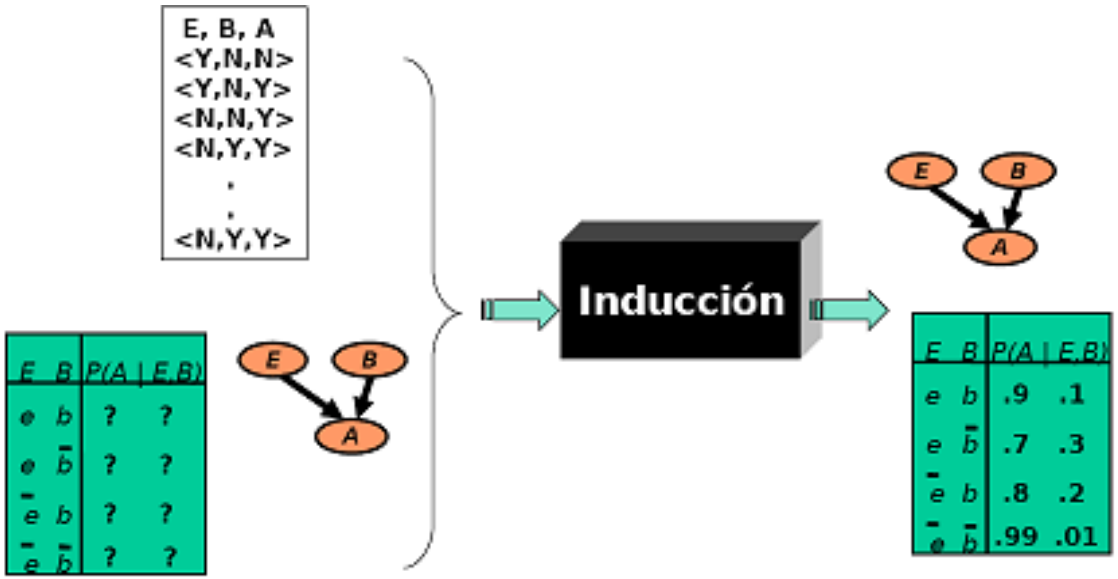


$$\lambda_A(f_1) = (0.992 * 0.005 + 0.2 * 0.995) * 1$$
$$+(0.008 * 0.005 + 0.8 * 0.995) * 0 \approx 0.204$$
$$\lambda_A(f_2) = 0.99 * 0.005 + 0.003 * 0.995 \approx 0.008$$

Existen tres escenarios en la inducción de Redes Bayesianas:

1. Estructura y TPCs se definen por expertos
2. Estructura Fija y TPCs inducidas de datos
3. Estructura y TPCs inducidas de datos





El problema se reduce a estimar las distribuciones de probabilidades de cada TPC. Esta estimación puede ser:

- Paramétrica: suponemos que la distribución es de un cierto tipo y estimamos sus parámetros.
- No Paramétrica: no hay suposición del tipo de la distibución.

- Variable continua: en general se supone distribución Normal

$$N(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

- Variable discreta: en general se supone distribución Multinomial

$$\Pr(Y_1 = y_1, ..., Y_k = y_k) = \frac{n!}{y_1!...y_k!} p_1^{y_1}...p_k^{y_k}, \ \textit{donde} \sum_{i=1}^{k} y_i = n$$

- Estimación de Máxima Verosimilitud
  - Los datos determinan completamente los parámetros
  - Problema: sobreajuste.
- Estimación Bayesiana
  - Tenemos información a priori de los parámetros (prior)
  - Los datos nos aportan información adicional para ajustar los parámetros

Definición

Sean los datos de entrenamiento $D = (x_1, ..., x_n)$ *y los parámetros* θ. *La función de verosimilitud es* $L(\theta : D) = \Pr(D|\theta)$. *Se llama el estimador de máxima verosimilitud (EMV) a*

$$\hat{\theta} = \arg\max_{\theta} L(\theta : D)$$

Proposición

- *Si los datos son iid tenemos* $L(\theta : D) = \prod_{i=1}^{n} \Pr(x_i|\theta)$
- *Se cumple que el EMV* $\hat{\theta} = \arg\max_\theta \log -L(\theta : D)$, *y en el caso iid,* $\log -L(\theta : D) = \sum_{i=1}^{n} \Pr(x_i|\theta)$

- Cada dato $x_i \in D$ es una instancia de una variable X, dom(X) = $\{v_1,...,v_k\}$
- Sea $\Pr(X = v_i) = p_i$
- Y=$(Y_1, ..., Y_k)$ ~Multinomial$(p_1, ..., p_k, n)$, donde Y_i es el número de ocurrencias del valor v_i en D
- $\theta = (p_1, ..., p_k)$
- $L(\theta : D) = \prod_{i=1}^{n} \Pr(x_i|\theta) = \prod_{i=1}^{k} p_i^{n_i}$, donde n_i es el número de datos en la instancia v_i
- $\log -L(\theta : D) = \sum_{i=1}^{k} n_i \log p_i = \sum_{i=1}^{k-1} n_i \log p_i + n_k \log(1 - \sum_{i=1}^{k-1} p_i)$
- $\frac{d(\log -L(\theta:D))}{dp_j} = \frac{n_j}{p_j} - \frac{n_k}{p_k} = 0 \Longrightarrow p_j = \frac{n_j}{c}$
- $\sum_{i=1}^{k} p_i = 1 \Longrightarrow \sum_{i=1}^{k} n_i = c = n \Longrightarrow \hat{p}_j = \frac{n_j}{n},\ j = 1..k$

En el caso continuo, suponiendo distribución normal, tenemos que

$$\log -L(D:\theta) = -\frac{n}{2}\log(2\pi) - n\log\sigma - \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2}$$

- $\frac{d(\log -L(D:\theta))}{d\mu} = \sum_{i=1}^{n}\frac{(x_i-\mu)}{\sigma^2} = 0 \Longrightarrow \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$
- $\frac{d(\log -L(D:\theta))}{d\sigma} = \frac{-n}{\sigma} + 2\sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^3} = 0 \Longrightarrow \hat{\sigma}^2 = \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{n}$
- Los EMV pueden llevar a sobreajuste

- Los EMV pueden llevar a sobreajuste.
- La estimación Bayesiana intenta solucionar este problema usando info a priori.
- La idea es que partimos con conocimiento de los parámetros y los datos modifican ese conocimiento.
- Por Teo. de Bayes tenemos $\Pr(\theta|D) = \frac{\Pr(D|\theta)\Pr(\theta)}{\Pr(D)} = \frac{\Pr(D|\theta)\Pr(\theta)}{\int_\theta \Pr(D|\theta)\Pr(\theta)}$

Definición

- *Estimador Máximo Posterior (MAP):* $\hat{\theta}_{MAP} = \arg\max_\theta \Pr(\theta|D)$
- *Estimador de Bayes:* $\hat{\theta}_{Bayes} = E(\theta|D)$

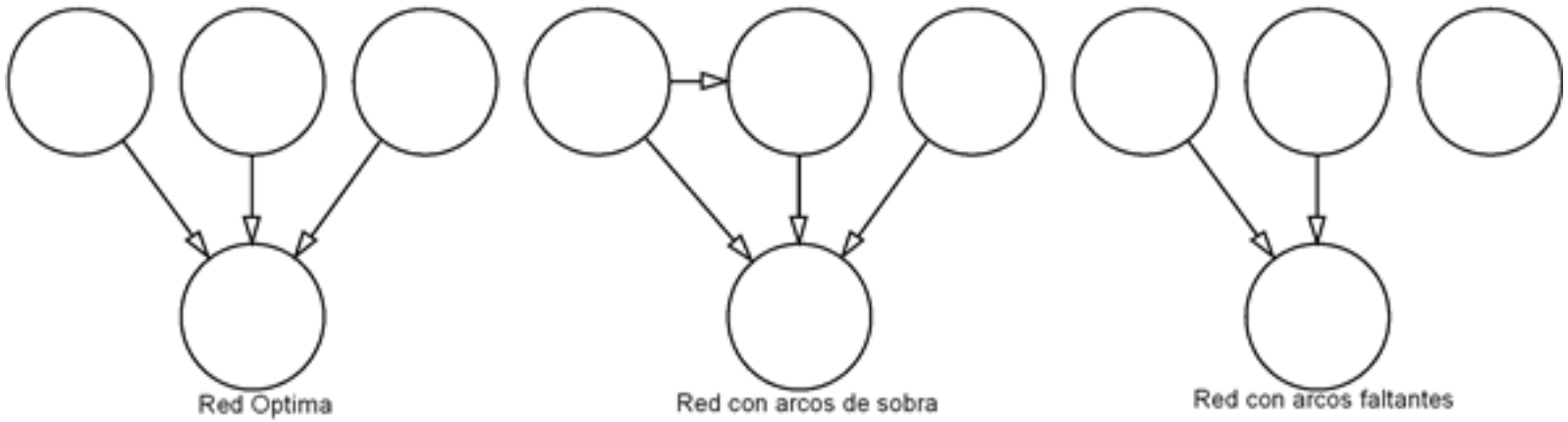
- Recordemos la distribución Binomial: probabilidad de tener s éxitos
- $\Pr(s|p,n) = \left(\begin{array}{c} \text{n} \\ \text{s} \end{array} \right) p^s(1-p)^{n-s}$, n fijo
- $\Pr(p|D) = \frac{\Pr(D|p)\Pr(p)}{\int_\theta \Pr(D|p)\Pr(p)} = \frac{p^s(1-p)^{n-s}\Pr(p)}{\int_\theta p^s(1-p)^{n-s}\Pr(p)dp}$
- Si no tenemos info a priori, asumimos que $p \sim U[0,1]$
- $\Pr(p|D) = \frac{p^s(1-p)^{n-s}}{\int_0^1 p^s(1-p)^{n-s}dp} \Longrightarrow p|D \sim Beta(s+1, n-s+1)$
- $\frac{d(p^s(1-p)^{n-s})}{dp} = sp^{s-1}(1-p)^{n-s} - (n-s)p^s(1-p)^{n-s-1} = 0$
- $\hat{p}_{MAP} = \frac{s}{n} \Longrightarrow$ *igual* al EMV
- $\hat{p}_{Bayes} = \frac{\int_0^1 p^{s+1}(1-p)^{n-s}dp}{\int_0^1 p^s(1-p)^{n-s}dp} = \frac{\Gamma(n+2)\Gamma(s+2)}{\Gamma(n+3)\Gamma(s+1)} = \frac{s+1}{n+2}$

  $\Longrightarrow$estimador de Laplace

- En general, si tenemos el prior $p \sim Beta(s', n' - s')$
- Obtenemos el posterior $p|D \sim Beta(s + s', n + n' - s - s')$
  - $\hat{p}_{Bayes} = \frac{s+s'}{n+n'}$
  - $\hat{p}_{MAP} = \frac{s+s'-1}{n+n'-2}$
- Supongamos que tenemos una moneda "cargada" y la lanzamos 30 veces, obteniendo 10 caras.
- Con prior uniorme obtenemos: $p|D_1 \sim Beta(10, 20) \implies \hat{p}_{Bayes} = \frac{10}{30}$
- Supongamos que lanzamos la moneda nuevamente 100 veces y observamos 50 caras.
- Obtenemos el posterior:
  $p|D_2 \sim Beta(10 + 50, 20 + 50) \implies \hat{p}_{Bayes} = \frac{60}{130}$

- El método de inferencia Bayesiano es una generalización del caso binominal.
- Se usa la distribución de Dirichlet que generaliza la distribución Beta.
- En general, tenemos el prior: $(p_1, ..., p_k) \sim Dir(s'_1, ..., s'_k)$
- Obtenemos el posterior: $(p_1, ..., p_k)|D \sim Dir(s_1 + s'_1, ..., s_k + s'_k)$
  - $\hat{p}_{iBayes} = \frac{s_i + s'_i}{n + n'}$
  - $\hat{p}_{iMAP} = \frac{s_i + s'_i - 1}{n + n' - k}$
- Se puede usar el estimador simple: $\hat{p}_{i\,\mathrm{Simple}} = \frac{s_i + \alpha}{n + k\alpha}$
  - Si $\alpha = 0 \Longrightarrow$obtenemos el EMV
  - Si $\alpha = 1 \Longrightarrow$obtenemos el Laplaciano

- La estructura de la red bayesiana determinará la eficiencia de la red y las suposiciones de independencia condicional.
- Si la estructura tiene un ordenamiento de las variables que no es adecuado, entonces la red puede no aprovechar las independencias condicionales de las variables, y hacer muy complejo el modelo.
- Si la estructura tiene arcos de sobra, el modelo es más complejo y se genera sobreajuste
- Si la estructura tiene arcos faltantes, el modelo hará falsas suposiciones de independencia, generando error por sesgo.



Existen dos tipos principales de métodos para la inducción de estructuras

- Métodos basados en puntaje
  - Cada posible estructura tiene un puntaje
  - El puntaje indica que tan bien la estructura representa los datos
  - Encontrar estructura que maximice el puntaje
- Métodos basados en test de independencia
  - Comienzan con una red completa e intentan eliminar arcos cuando se verifica independencia
  - Comienza con una red vacía e intenta agregar arcos cuando no se verifica independencia
  - El test más usuado para verificar independencia es el test de chi-cuadrado

Definición

Dada una estructura G y datos de entrenamiento D, tenemos:

- *Prob. conjunta observada:* $\Pr_D$
- *Prob. conjunta de la red:* $\Pr_{G,D}$

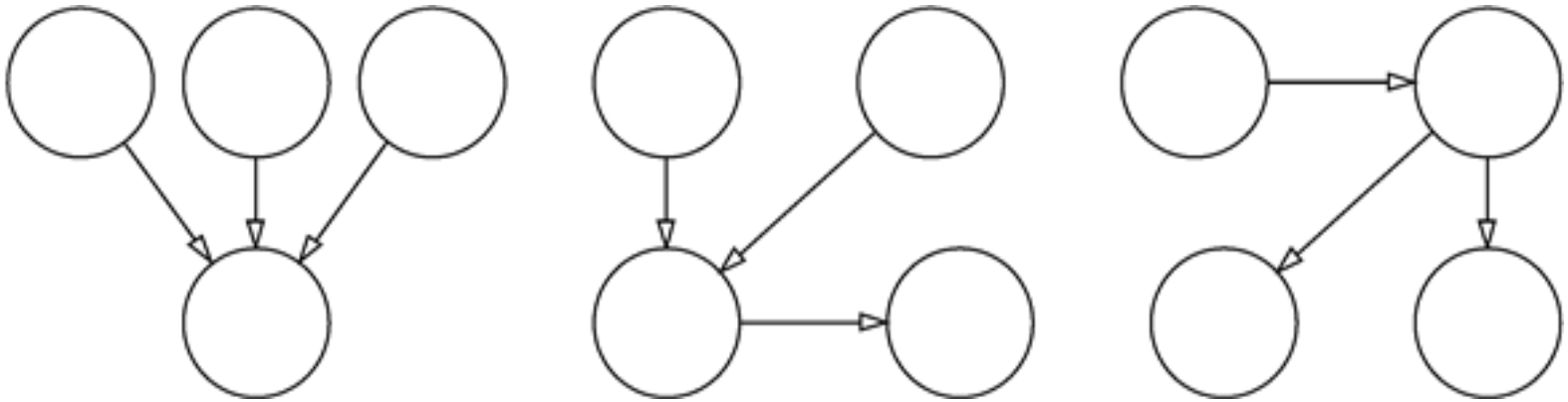
- Puntajes Locales
  - Miden qué tan bien la estructura representa la probabilidad conjunta
  - Es decir qué tanto se parecen $\Pr_D$ y $\Pr_{G,D}$
  - Puntaje de Versosimilitud
  - Puntaje MDL (Minimum Description Length)
- Puntajes Globales
  - Miden qué tan bien la estructura se comporta para predecir
  - Por ejemplo, se puede usar error de clasificación

- Bajo suposición de que los datos son idd, obtenemos:
  $L(G:D) = \prod_{x_i \in D} \Pr_{G,D}(x_i)$
- Aplicamos logaritmo: $\log -L(G:D) = \sum_{x_i \in D} \log \Pr_{G,D}(x_i)$
- Luego $\sum_{x_i \in D} \log \Pr_{G,D}(x_i) = \sum_{x_i \in dom(X)} n_i \log \Pr_{G,D}(x_i)$,
  donde n_i es la cantidad de datos x_i
- Finalmente, como $\Pr_D(x_i) = \frac{n_i}{n}$

$$\log -L(G:D) = n \sum_{x_i \in dom(X)} \Pr_D(x_i) \log \Pr_{G,D}(x_i)$$

- Función de pérdida de información:
  $\text{fpi}(\Pr_0, \Pr_1) = \int_X \Pr_0(x) \log \Pr_1(x) dx$
- Mide similitud entre dos distribuciones de probabilidad

- La red que maximiza el puntaje de verosimilitud es la red "completa".
- El puntaje de verosimilitud no considera la complejidad de la red, sino solo la codificación de los datos en la red.
- Esto lleva a sobreajuste
- Este puntaje se debe usar en un conjunto reducido de estructuras, y no sobre el espacio completo.



Definición

El costo de descripción es la cantidad de información necesaria para codificar modelo y codificar los datos usando el modelo.
$Costo(M, D) = Costo(M) + Costo(D|M)$

- Supongamos que transmitimos en un canal (o codificamos) símbolos en $\{x_1, ..., x_n\}$
- A primera vista, necesitamos $\log n$ bits por mensaje.
- Si el emisor y el receptor conocen la distribución de los datos $Pr(X = x_i) = p_i$ se puede elabrar un código que requiera menos bits por mensaje.

Si usamos este código (Huffman) para enviar mensajes (A,B,C,D), en promedio cada mensaje require 1.75 bits, versus los 2 bits usuales.

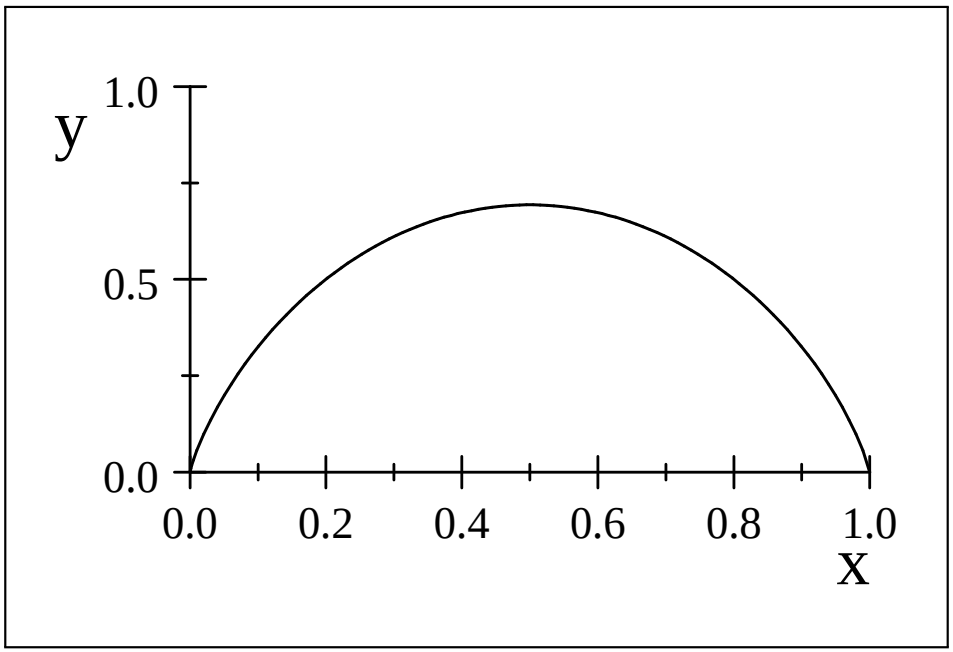
M	cod	long	prob	prom
A	000	3	0.125	0.375
B	001	3	0.125	0.375
C	01	2	0.25	0.5
D	1	1	0.5	0.5

Definición

La entropía es el mínimo teórico de bits promedio necesarios para trasmitir un conjunto de mensajes sobre $\{x_1, ..., x_n\}$ *con distribución de prob.* $Pr(X = x_i) = p_i$. *Es la información asociada a la distribución de probabilidades P.*

$$Entropy(P) = -\sum p_i \log(p_i)$$

- Mientras más uniforme es P, mayor es su entropía
  - Si P es (0.5, 0.5), Entropy(P) = 1
  - Si P es (0.67, 0.33), Entropy(P) = 0.92
  - Si P is (1, 0), Entropy(P)=0



Proposición

- $Costo(D|M) = -n * \sum p_i^D \log(p_i^{M,D}) = n * fpi(\Pr_D, \Pr_{M,D})$, *donde* $\Pr_D$ *es la distribución observada en los datos y* $\Pr_{M,D}$ *es la distribución del modelo.*
- $Costo(M) = \frac{dim(M) * \log n}{2}$

Definición

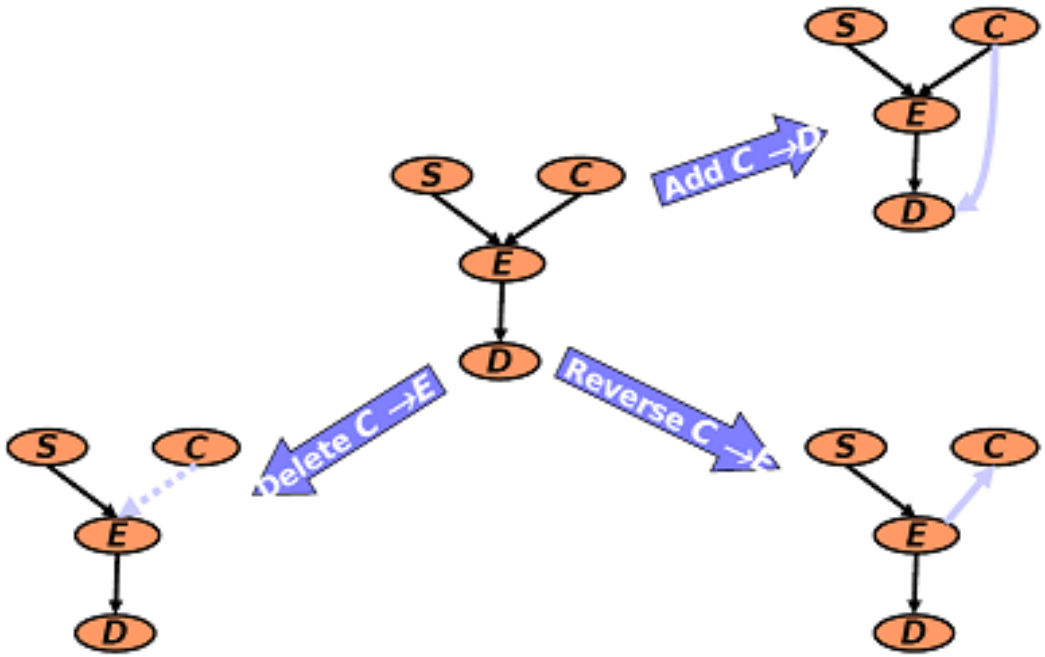
- $PuntajeMDL(G{:}\theta) = n * fpi(\Pr_D, \Pr_{M,D}) + \frac{dim(G) * \log n}{2}$
  Mide el costo de codificar los datos usando la probabilidad de la red más el costo de codificar la red.
- $PuntajeAIC(G{:}\theta) = n * fpi(\Pr_D, \Pr_{M,D}) + dim(G)$
- *Entropía:* $H(G{:}\theta) = -\log L(G : \theta)$

Los puntajes anteriores son "costos" de los modelos.

- Entrada
  - Datos de Entrenamiento
  - Función de puntaje
  - Posibles estructuras
- Salida
  - Red que maximiza (o minimiza) puntaje
  - TPC
- Espacio de búsqueda
  - Estados son posibles estructuras
  - Operaciones modifican la red y generan nuevos estados a evaluar
- Algoritmos de Búsqueda
  - Recorren el espacio en busca de estructuras con mejor puntaje

Las operaciones más comunes al realizar la búsqueda son:

- Agregar arco
- Eliminar arco
- Invertir arco



Algunos de los Algoritmos de Búsqueda en Weka son:

- Búsqueda local
  - Hill Climbing
  - Reapeated Hill Climbing
  - LAGD Hill Climbing
  - K2: Hill Climbing con un orden fijo de variables
- Búsqueda Heurística
  - Simmulated Annealing
  - Tabu Search
  - Genetic Search
- TAN: búsqueda de estructuras de árbol
- Métodos Basados en Test de Independencia
  - CISearchAlgorithm
  - ICSSearchAlgorithm

- La Búsqueda de estructuras con puntaje máximo, para redes con al menos k $>$ 1 padres por nodo es NP-hard.
- Este problema es un problema de Optimización Combinatorial.
- Para k=1 (árboles) se puede resolver en tiempo polinomial.
- Arboles tienen pocos parámetros por lo que en general evitan sobreajuste, pero no permiten codificar funciones muy complejas
- La prueba final siempre debe ser con un conjunto separado de datos, el conjunto test, que evalúa el desempeño de la estructura y sus TPCs en la predicción de la red