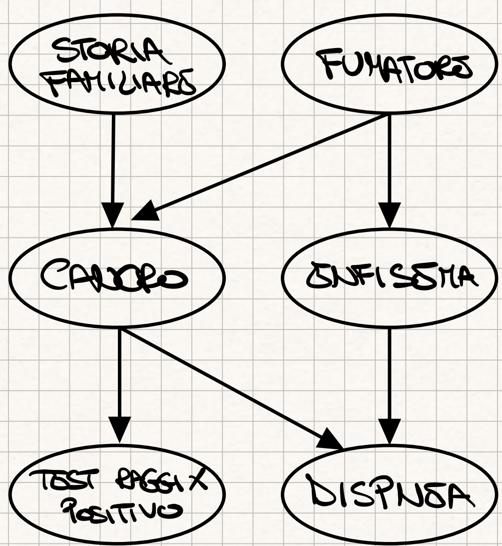
BAYESIAN BELIEF NETWORK

È diffatti una miglioria del classico Bayesian, che si modella alle dipendenze fra vari attributi.
Usa l'idea di grafo dove:

- **NODI**: Variabili aleatorie (Random Variable)
- **LINK**: Dipendenze fra le variabili

Questo grafo non ha cicli.

Esempio



Conditional Probability Tab
Per la variabile CANCRO

	SF S	SF NS	NSF S	NSF NS
C	0,8	0,5	0,7	0,1
NC	0,2	0,5	0,3	0,9

P(Cancro = si | StoriaFamigliare = yes, Fumatore = yes) = 0.8
P(Cancro = no | StoriaFamigliare = no, Fumatore = no) = 0.9

Una Belief Network ha una Conditional Probably Table per ogni variabile Y. La CPT per una variabile Y specifica la distribuzione di probabilità condizionata

$$P(Y \mid Parents(Y))$$

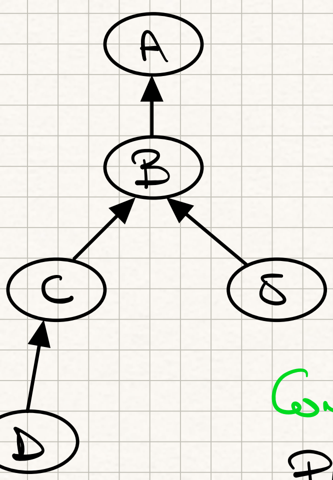
La tabella dell'esempio di sopra specifica la CPT per la variabile CANCRO i cui parenti sono STORIA FAMIGLIARE e FUMATORE.

Prendiamo $X=(x_1 \dots x_n)$ le tuple da analizzare, e diciamo che ogni tupla è descritta dagli attributi $Y_1 \dots Y_n$. Ogni variabile è indipendente da chi non è direttamente collegato con un link nel grafo, questo fa in modo che possiamo rappresentare l'intera distribuzione di probabilità condizionata con la seguente equazione:

$$P(x_1 \dots x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(Y_i))$$

Dove per $Parents(Y_i)$ si intendono le entrate della Conditional Probably Table (CPT). Difatti la formula rappresenta la distribuzione della joint pobability, ovvero la probabilità che una determinata combinazione dei valori di X si verifichi.

Esempio



Non considerando la rete, supponendo l'indipendenza fra tutte le variabili:

$$P(A,B,C,D,E) = P(A \mid B,C,D,E) \cdot P(B \mid C,D,E) \cdot P(C \mid D,E) \cdot P(D \mid E) \cdot P(E)$$

Considerando la rete, quindi le dipendenze:

$$P(A,B,C,D,E) = P(A \mid B) \cdot P(B \mid C,E) \cdot P(C \mid D) \cdot P(D) \cdot P(E)$$

Di fatti il classificatore di tipo Network Bayesian, presuppone che uno o più di uno dei nodi nella rete siano la classe - labeled target.
Il classificatore può anche dare in output la distribuzione di probabilità che dato un esempio quale è la probabilità di essere classificato con la classe C_1 piuttosto che C_2. Distribuzione di probabilità fra le classi.

SCENARI:

1) Abbiamo la struttura della rete e le variabili osservate bisogna solo calcolare la CPT.

2) Rete conosciuta e solo qualche variabile nota, non tutte. Possiamo usare il gradiente discendente per aggiornare le CPT entry o pesi. Questi pesi sono inizializzati randomicamente.

3) Rete sconosciuta, variabili osservabili: Ricostruire la rete attraverso lo spazio dei modelli. Credo che semplicemente bisogna valutare la dipendenza fra le variabili

4) Se non conosci nulla ti attacchi.

METODO DEL GRADIENTE DISCENDENTE

Prendiamo in considerazione il dataset D con le tuple $X_1 \dots X_{|D|}$.

Indichiamo con w_{ijk} le entrate della CPT o pesi.

Il peso w_{ijk} fa riferimento all'entrata della CPT per la variabile $Y_i = y_{ij}$ che ha i parenti $U_i = u_{ik}$.

$$w_{ijk} = P(Y_i = y_{ij} \mid U_i = u_{ik})$$

All'inizio tutti i pesi sono scelti randomicamente.

L'algoritmo funziona scalando in modo greedy la distribuzione di probabilità:

$$\boxed{P_w(D) = \prod_{d=1}^{|D|} P_w(X_d)}$$

Cercando difatti l'ottimo. Per farlo bisogna seguire il gradiente negativo del $\ln(P_w(D))$.

A ogni iterazione i passi da fare sono:

1) Calcolare il gradiente:

$$\frac{\partial \ln(P_w(D))}{\partial w_{ijk}} = \sum_{d=1}^{|D|} \frac{P(Y_i = y_{ij}, U_i = u_{ik} \mid X_d)}{w_{ijk}}$$

2) Aggiornare i pesi lungo il gradiente

$$w_{ijk} = w_{ijk} + \ell \cdot \frac{\partial \ln(P_w(D))}{\partial w_{ijk}}$$

Dove con ℓ indichiamo la dimensione del passo.

3) Rinormalizzare i pesi:

Essendo probabilità devono stare fra $\emptyset$ e 1 e la $\sum_j w_{ijk} == 1 \quad \forall i,k$.