Apprendimento di Reti Bayesiane

Massera Gianluca

Università degli studi di Roma “La Sapienza”
facoltà di S.M.F.N. corso di laurea in Informatica

Reti Bayesiane: introduzione

- Le **Rete Bayesiane** forniscono un modello per le distribuzioni di probabilità

Reti Bayesiane: introduzione

- Le **Rete Bayesiane** forniscono un modello per le distribuzioni di probabilità
- Le caratteristiche che le rendono un modello utile per agenti intelligenti sono:

Reti Bayesiane: introduzione

- Le **Rete Bayesiane** forniscono un modello per le distribuzioni di probabilità
- Le caratteristiche che le rendono un modello utile per agenti intelligenti sono:
  - **Compattezza della rappresentazione**

Reti Bayesiane: introduzione

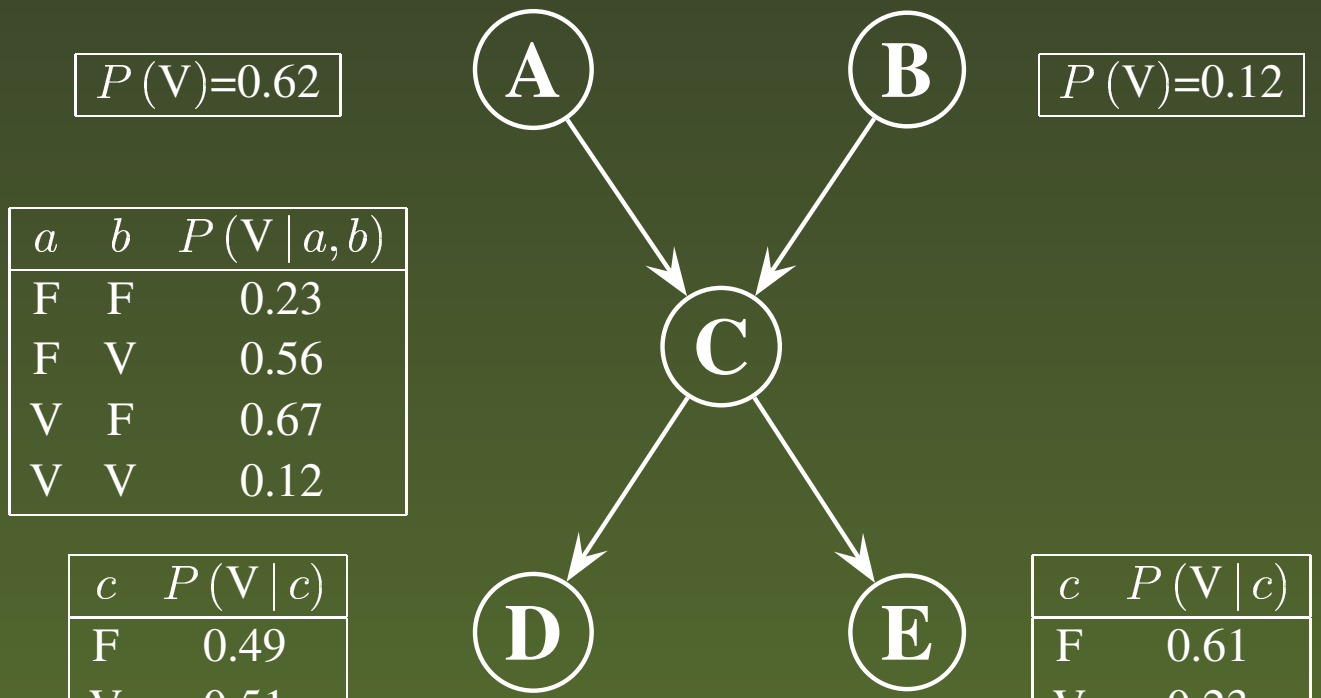
- Le **Rete Bayesiane** forniscono un modello per le distribuzioni di probabilità
- Le caratteristiche che le rendono un modello utile per agenti intelligenti sono:
  - **Compattezza della rappresentazione**
  - **Causalità in primo piano**

Reti Bayesiane: introduzione

- Le **Rete Bayesiane** forniscono un modello per le distribuzioni di probabilità
- Le caratteristiche che le rendono un modello utile per agenti intelligenti sono:
  - **Compattezza della rappresentazione**
  - **Causalità in primo piano**
  - **Dotato di una Semantica**

Rete Bayesiana: definizione

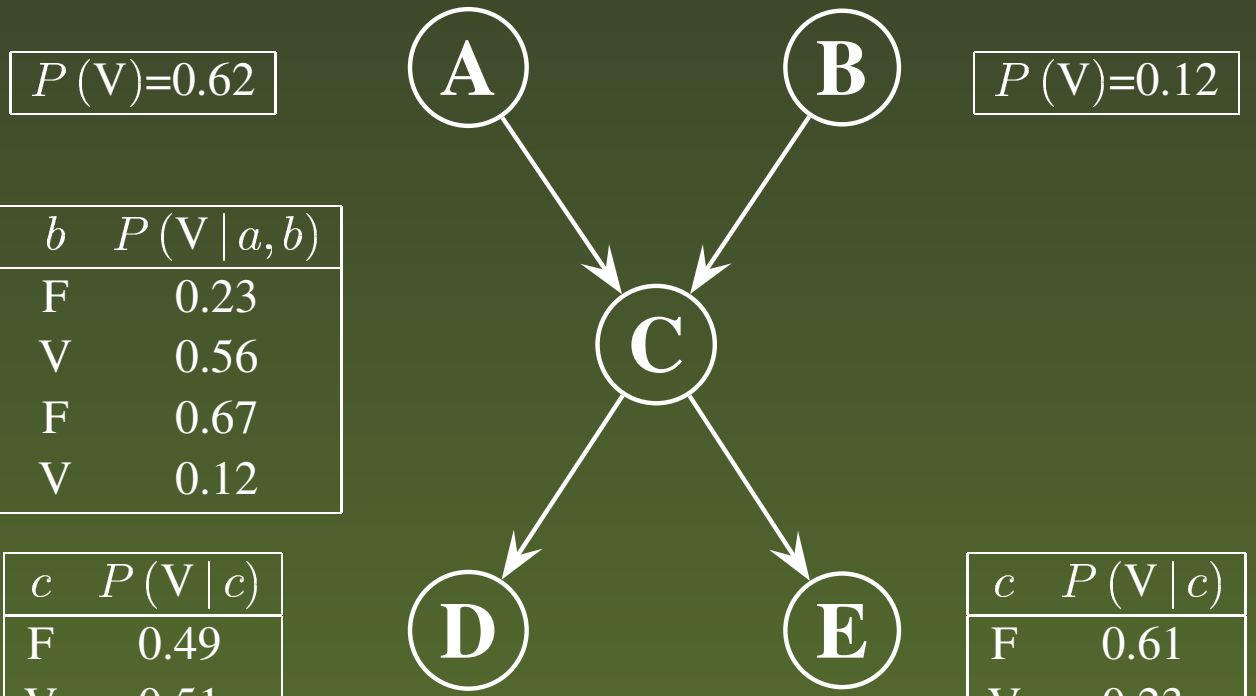
Una Rete Bayesiana è una tripla $\langle V, A, \Theta \rangle$ dove:



Rete Bayesiana: definizione

Una Rete Bayesiana è una tripla $\langle V, A, \Theta \rangle$ dove:

- $(V, A) = G$ è il grafo diretto aciclico che rappresenta le relazioni di causalità fra i nodi



$P(\mathrm{V})$=0.62

$P(\mathrm{V})$=0.12

a	b	$P(\mathrm{V} \mid a,b)$
F	F	0.23
F	V	0.56
V	F	0.67
V	V	0.12

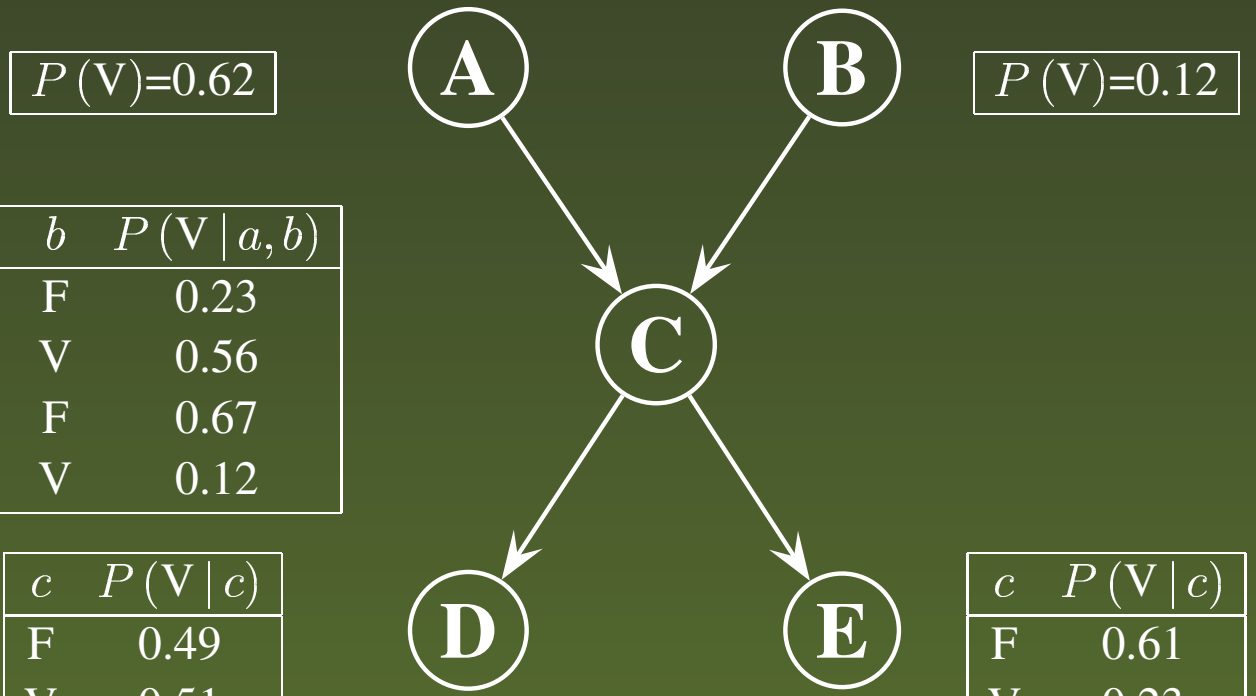
c	$P(\mathrm{V} \mid c)$
F	0.49
V	0.51

c	$P(\mathrm{V} \mid c)$
F	0.61
V	0.23

Rete Bayesiana: definizione

Una Rete Bayesiana è una tripla $\langle V, A, \Theta \rangle$ dove:

- $\Theta = \{\theta_i\}_{i=1}^{|V|}$ è un insieme di distribuzioni di probabilità condizionate: $\theta_i = P(X_i \,|\, \mathbf{Pa}_i)$



$P(\mathrm{V})$=0.62

$P(\mathrm{V})$=0.12

a	b	$P(\mathrm{V} \,\|\, a,b)$
F	F	0.23
F	V	0.56
V	F	0.67
V	V	0.12

c	$P(\mathrm{V} \,\|\, c)$
F	0.49
V	0.51

c	$P(\mathrm{V} \,\|\, c)$
F	0.61
V	0.23

Reti Bayesiane & Agenti

- Per gli **agenti intelligenti**, la rete bayesiana rappresenta la loro **conoscenza**

- In base agli stimoli dell'ambiente, l'agente interroga la rete bayesiana per dedurre qual'è l'azione più idonea

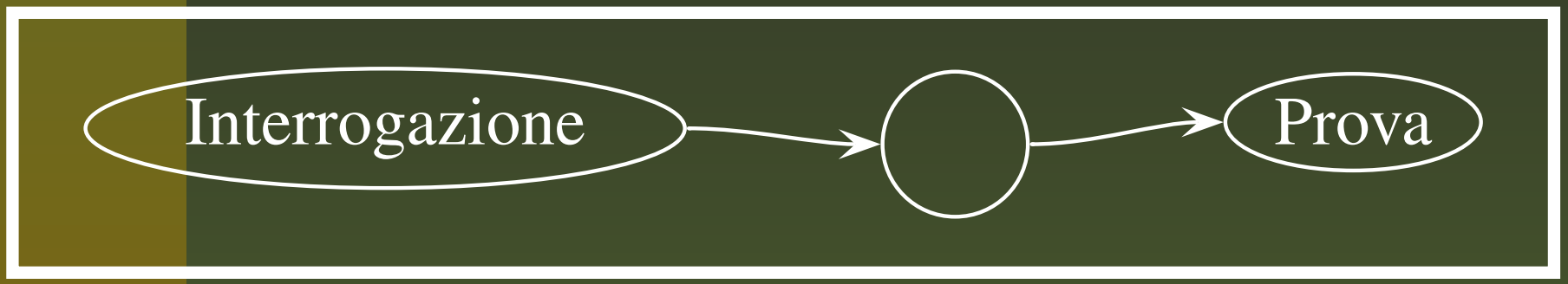
Reti Bayesiane: interrogazione

Esistono quattro tipi di interrogazione

Reti Bayesiane: interrogazione

Esistono quattro tipi di interrogazione

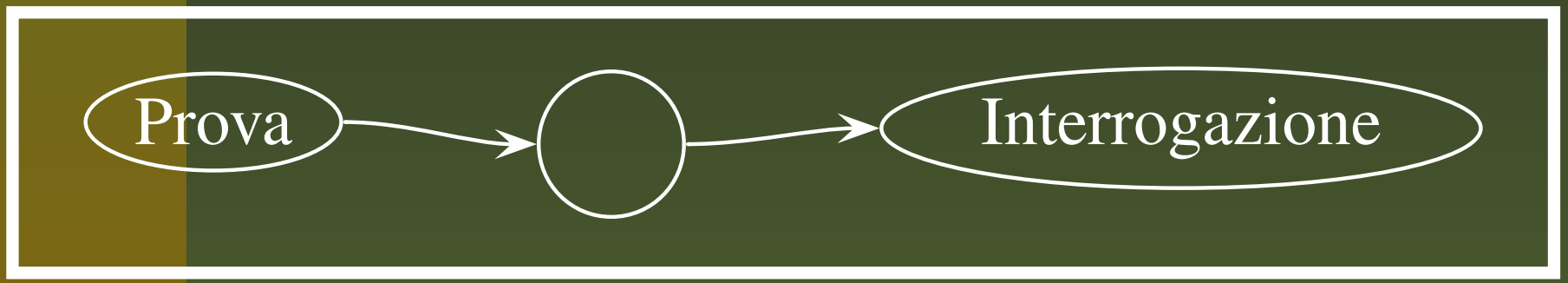
- **Diagnostica**: dagli effetti alle cause



Reti Bayesiane: interrogazione

Esistono quattro tipi di interrogazione

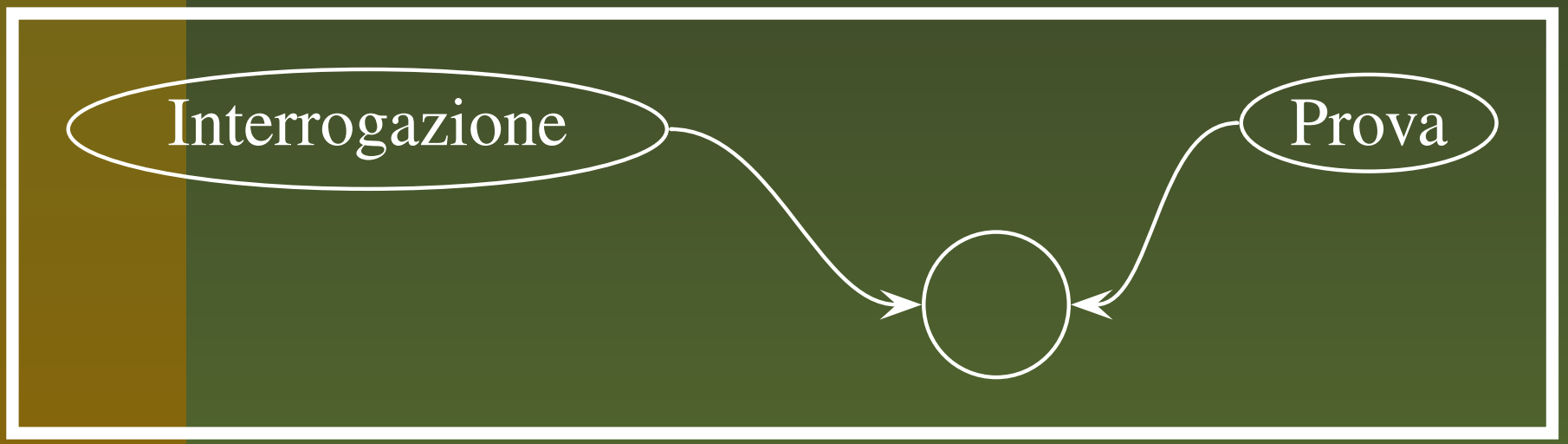
- **Diagnostica**: dagli effetti alle cause
- **Causale**: dalle cause agli effetti



Reti Bayesiane: interrogazione

Esistono quattro tipi di interrogazione

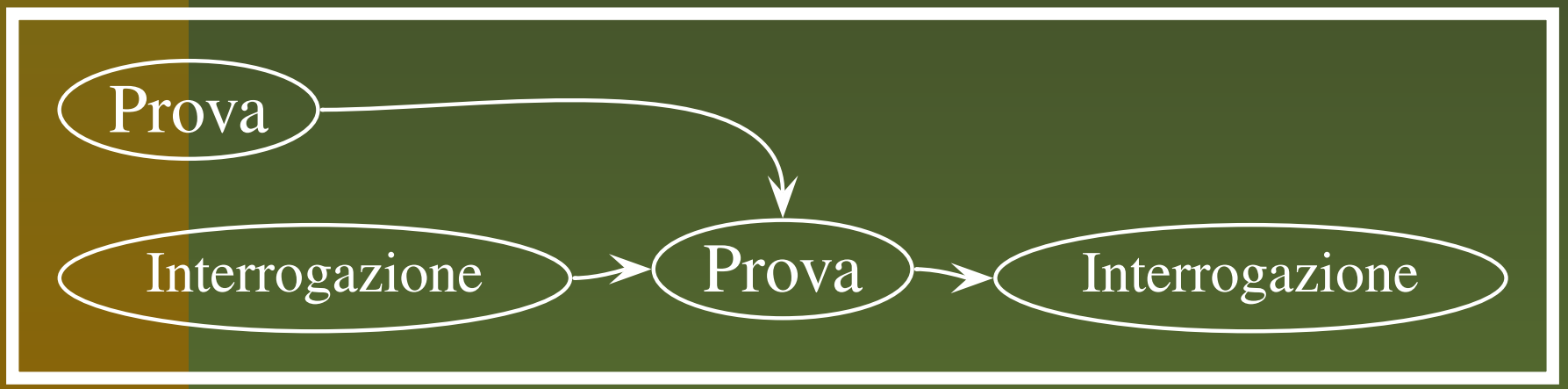
- **Diagnostica**: dagli effetti alle cause
- **Causale**: dalle cause agli effetti
- **Intercausale**: tra cause di un effetto comune



Reti Bayesiane: interrogazione

Esistono quattro tipi di interrogazione

- **Diagnostica**: dagli effetti alle cause
- **Causale**: dalle cause agli effetti
- **Intercausale**: tra cause di un effetto comune
- **Mista**: una combinazione delle precedenti



Reti Bayesiane: classificatori

- Tramite una rete bayesiana si rappresenta la distribuzione

$$P(C \mid \mathbf{A})$$

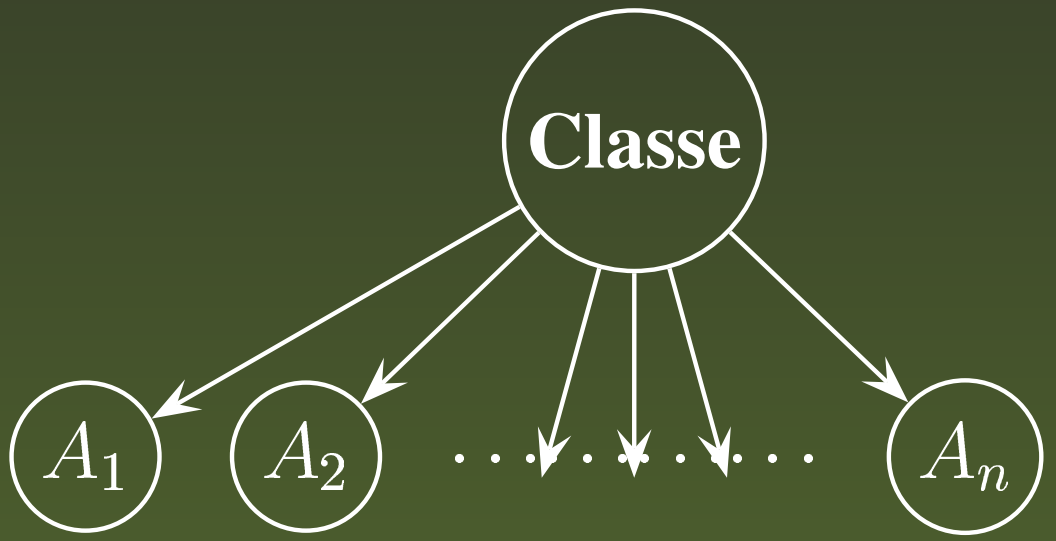
dove C è l'insieme delle etichette, ed $\mathbf{A}$ l'insieme degli attributi degli esempi

- A questo punto, un semplice modo per classificare un esempio ($\mathbf{a}$) consiste nel calcolare

$$F(\mathbf{a}) = \arg\max_{c \in C} P(c \mid \mathbf{a})$$

Reti Bayesiane: struttura Naive

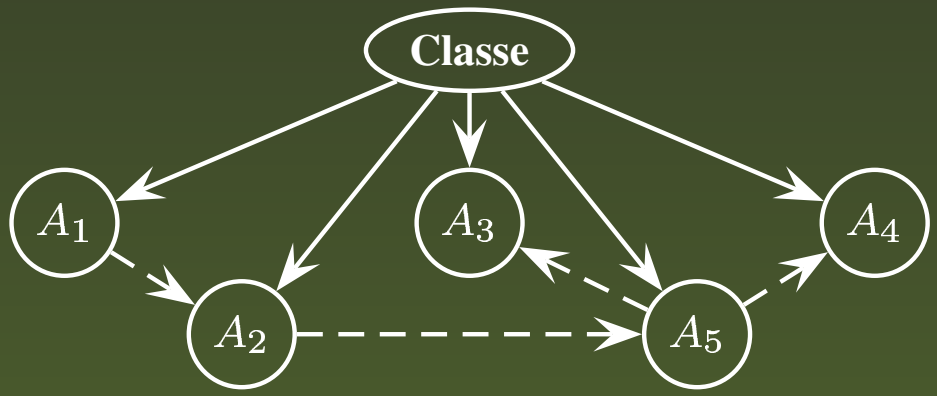
- Si tratta di una semplice struttura grafica che consente di realizzare facilmente dei classificatori
- Si assume che A_i è **indipendente** da A_j per $i \neq j$



- Il grafo è **fissato**; le distribuzioni di probabilità condizionate devono essere **apprese**

Reti Bayesiane: Naive aumentata

- Generalizza la Naive togliendo l'assunzione di indipendenza fra gli attributi
- Infatti, si assume che le dipendenze fra gli attributi sono rappresentabili con un **albero diretto**



- Il grafo è **parzialmente fissato**, le linee trattegiate vanno **apprese**; anche le distribuzioni di probabilità condizionate vanno **apprese**

Apprendimento: introduzione

Tutti i metodi di apprendimento finora sviluppati possono essere divisi in due categorie

Apprendimento: introduzione

Tutti i metodi di apprendimento finora sviluppati possono essere divisi in due categorie

- **Metodi basati sul punteggio**

  cercano la rete bayesiana con il punteggio massimo,

  assegnatogli secondo un qualche criterio

Apprendimento: introduzione

Tutti i metodi di apprendimento finora sviluppati possono essere divisi in due categorie

- **Metodi basati sul punteggio**

  cercano la rete bayesiana con il punteggio massimo, assegnatogli secondo un qualche criterio

- **Metodi basati sull'indipendenza**

  ricavano la struttura grafica analizzando le relazioni di indipendenza fra gli attributi. Eseguono due passi principali:

  1. **apprendono** la struttura grafica della rete
  2. **apprendono** le distribuzioni di probabilità condizionate

Apprendimento: Il punteggio MDL

- Il ***M**inimal **D**escription **L**enght* è un esempio di punteggio basato sulla teoria dell'informazione
  - Volendo comprire l'insieme di apprendimento D con un codice, la teoria dell'informazione garantisce l'ottimo se si usa la distribuzione degli esempi in D
  - Allora se, per codificare D, si usa la distribuzione P_B data dalla rete B, si può affermare che
    - Il codice ottimale si ottiene quanto P_B replica la distribuzione degli esempi in D
  - Quindi, apprendere una rete bayesiana vuol dire eseguire la ricerca di quella che minimizza la lunghezza del codice

Apprendimento: MDL in dettaglio

- Con l'MDL si cerca un mimino
  sia per la dimensione della rete (**compattezza**),
  che per la codifica di D (**accuratezza**)
- Dato un insieme di apprendimento D, si cerca la rete bayesiana B tale da minimizzare la quantità

$$DL_g(V, A) + \sum_i DL_\theta(X_i, \mathbf{Pa}_i) + DL_d(D \mid B)$$

Apprendimento: MDL in dettaglio

- Con l'MDL si cerca un mimino
  sia per la dimensione della rete (**compattezza**),
  che per la codifica di D (**accuratezza**)
- Dato un insieme di apprendimento D, si cerca la rete bayesiana B tale da minimizzare la quantità

$$DL_g(V, A) + \sum_i DL_\theta(X_i, \mathbf{Pa}_i) + DL_d(D \mid B)$$

- **Lunghezza della descrizione di** A

Apprendimento: MDL in dettaglio

- Con l'MDL si cerca un mimino
  sia per la dimensione della rete (**compattezza**),
  che per la codifica di D (**accuratezza**)
- Dato un insieme di apprendimento D, si cerca la rete bayesiana B tale da minimizzare la quantità

$$DL_g(V, A) + \sum_i DL_\theta(X_i, \mathbf{Pa}_i) + DL_d(D \mid B)$$

- **Lunghezza della descrizione di Θ**

Apprendimento: MDL in dettaglio

- Con l'MDL si cerca un mimino
  sia per la dimensione della rete (**compattezza**),
  che per la codifica di D (**accuratezza**)
- Dato un insieme di apprendimento D, si cerca la rete bayesiana B tale da minimizzare la quantità

$$DL_g(V, A) + \sum_i DL_\theta(X_i, \mathbf{Pa}_i) + DL_d(D \mid B)$$

- **Lunghezza della descrizione di** D

Apprendimento: con DAG fissato (1)

- In alcuni casi la struttura grafica è **completamente fissata**:
  - nelle reti bayesiane Naive
  - nella seconda fase dei metodi basati sull'indipendenza
  - quando la rete viene completamente determinata da un esperto del dominio
  - e altro ancora
- In questi casi l'apprendimento consiste soltanto nel trovare i giusti valori da inserire all'interno delle tabelle θ_i con $0 < i \leq |V|$

Apprendimento: con DAG fissato (2)

- Un risultato basato sull'MDL consente di apprendere **efficientemente** e **velocemente** i valori **ottimi** per le tabelle $\{\theta_i\}$, vediamo come:

Apprendimento: con DAG fissato (2)

- Un risultato basato sull'MDL consente di apprendere **efficientemente** e **velocemente** i valori **ottimi** per le tabelle $\{\theta_i\}$, vediamo come:
- Rivediamo la funzione da minimizzare vista per il punteggio MDL:

$$DL_g(V, A) + \sum_i DL_\theta(X_i, \mathbf{Pa}_i) + DL_d(D \mid B)$$

Apprendimento: con DAG fissato (2)

- Un risultato basato sull'MDL consente di apprendere **efficientemente** e **velocemente** i valori **ottimi** per le tabelle $\{\theta_i\}$, vediamo come:
- Il primo termine è una **costante** perchè il grafo è **fissato**

$$DL_g(V, A) + \sum_i DL_\theta(X_i, \mathbf{Pa}_i) + DL_d(D \mid B)$$

Apprendimento: con DAG fissato (2)

- Un risultato basato sull'MDL consente di apprendere **efficientemente** e **velocemente** i valori **ottimi** per le tabelle $\{\theta_i\}$, vediamo come:
- Anche questo termine è una **costante** perchè una volta fissato il grafo sono fissate anche il numero e la dimensione delle tabelle θ

$$DL_g(V, A) + \sum_i DL_\theta(X_i, \mathbf{Pa}_i) + DL_d(D \mid B)$$

Apprendimento: con DAG fissato (2)

- Un risultato basato sull'MDL consente di apprendere **efficientemente** e **velocemente** i valori **ottimi** per le tabelle $\{\theta_i\}$, vediamo come:
- Questo è l'unico termine **variabile** in funzione dei valori inseriti nelle tabelle θ

$$DL_g(V, A) + \sum_i DL_\theta(X_i, \mathbf{Pa}_i) + DL_d(D \mid B)$$

Apprendimento: con DAG fissato (2)

- Un risultato basato sull'MDL consente di apprendere **efficientemente** e **velocemente** i valori **ottimi** per le tabelle $\{\theta_i\}$, vediamo come:
- Si dimostra che il minimo si raggiunge quando le probabilità sono ottenute dalle frequenze degli esempi in D

$$DL_g(V, A) + \sum_i DL_\theta(X_i, \mathbf{Pa}_i) + DL_d(D \mid B)$$

Apprendimento: TPDA

- ***T**hree-**P**hase **D**ependency **A**nalysis*, è un algoritmo basato sull'indipendenza
- Attraverso tre fasi costruisce il grafo della rete:
  1. **Bozza**: costruisce un'albero iniziale che farà da base per le successive fasi
  2. **Crescita**: inserisce, man mano, nuovi archi fino a realizzare un grafo completo di tutte le dipendenze fra gli attributi
  3. **Potatura**: serve ad eliminare archi ridondanti al fine di rendere la rete il più compatta possibile
- Al termine di queste, apprende le tabelle delle probabilità condizionate come già visto

Apprendimento: TPDA (test CI)

Test di Indipendenza Condizionata

- È indispensabile per gli algoritmi basati sull'indipendenza, infatti la loro complessità è il numero di test CI effettuati
- Inoltre, gli algoritmi vengono confrontati anche in base all'implementazione del test CI
- Nel caso di TPDA si suppone che:

X **e** Y **sono indipendenti dato** C **se e solo se** $I(X, Y \mid C) < \varepsilon$

- Si tratta di un'indipendenza condizionata più debole rispetto a quella usuale tra variabili aleatorie

Apprendimento: TPDA (inizio)

- gli **Input** dell'algoritmo sono:
  - D, insieme di apprendimento
  - ε, soglia del test CI
- Il primo passo dell'algoritmo inizializza alcune strutture per le fasi successive:

1) $V \leftarrow$ insieme degli attributi di D
   $E \leftarrow \emptyset$; insieme di archi non orientati
   $L \leftarrow \{(X,Y) \mid I(X,Y) > \varepsilon \, e \, X \neq Y\}$

Apprendimento: TPDA (bozza)

- Questa fase genera un albero secondo l'algoritmo d'apprendimento dato da Chow e Liu nel 1968

2) **Fase di bozza**

ordina L in modo decrescente rispetto a $I(X,Y)$

per ogni $(X,Y) \in L$

Se $\nexists$ cammino tra X e Y nel grafo (V,E) allora

aggiungi $\{X,Y\}$ ad E,

rimuovi (X,Y) da L

Apprendimento: TPDA (crescita)

- Durante questa fase, si considerano le coppie di nodi rimaste dopo la fase di bozza
- Per ognuna di esse, la funzione *EdgeNeeded_H* determina la necessità di inserire un arco tra i nodi della coppia.
- Spesso la rete generata, fino a questo punto, presenta svariati archi ridondanti.

```
3) Fase di crescita
   per ogni (X,Y) ∈ L
     Se EdgeNeeded_H(V, E, X, Y, D, ε) allora
       aggiungi {X,Y} ad E
```

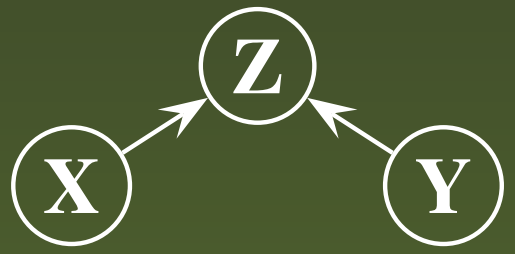
Apprendimento: TPDA (potatura)

- Questa fase individua ed elimina gli archi ridondanti

4) **Fase di potatura**
per ogni $\{X,Y\} \in E$
Se $\exists$ cammino tra X e Y diverso dal solo arco $\{X,Y\}$ allora
$E' \leftarrow E \setminus \{X,Y\}$
Se $\neg$*EdgeNeeded_H*$(V, E', X, Y, D, \varepsilon)$ allora
$E \leftarrow E'$
per ogni $\{X,Y\} \in E$
Se X ha almeno tre vicini, o Y ha almeno tre vicini allora
$E' \leftarrow E \setminus \{X,Y\}$
Se $\neg$*EdgeNeeded*$(V, E', X, Y, D, \varepsilon)$ allora
$E \leftarrow E'$

Apprendimento: TPDA (fine)

- Le fasi precedenti costruiscono un grafo non-diretto
- L'ultima operazione consiste nel trovare la corretta orientazione agli archi, ed ottenere finalmente il DAG della rete bayesiana
- Per orientare gli archi si cercano le **v-strutture** del grafo tramite appositi test CI:



- Una volta determinate le v-strutture, gli archi rimanenti vengono orientati di conseguenza

Apprendimento: TPDA (EdgeNeeded)

- Nei passaggi dell'algoritmo visti, viene richiamata in molti punti una funzione che effettua test CI per valutare se un certo arco è necessario oppure no.
- Ci sono tre versioni differenti:
  - *EdgeNeeded**: complessità esponenziale
  - *EdgeNeeded_H*: molto veloce grazie ad un'euristica, ma esistono dei casi in cui fallisce; $O\left(k^2\right)$
  - *EdgeNeeded*: aggiunge all'euristica dei controlli per evitare il fallimento, quindi più lenta rispetto a *EdgeNeeded_H*; $O\left(k^2\right)$

Apprendimento: TPDA (complessità)

La complessità dell'algoritmo è $O\left(n^4\right)$

Apprendimento: TPDA (complessità)

La complessità dell'algoritmo è $O\left(n^4\right)$

- **Fase di bozza**
  richiede un test CI per ogni coppia di nodi; $O\left(n^2\right)$

Apprendimento: TPDA (complessità)

La complessità dell'algoritmo è $O\left(n^4\right)$

- **Fase di bozza**
  richiede un test CI per ogni coppia di nodi; $O\left(n^2\right)$
- **Fase di crescita**
  richiede $O\left(n^2\right)$ test CI per ogni coppia di nodi; $O\left(n^4\right)$

Apprendimento: TPDA (complessità)

La complessità dell'algoritmo è $O\left(n^4\right)$

- **Fase di bozza**
  richiede un test CI per ogni coppia di nodi; $O\left(n^2\right)$
- **Fase di crescita**
  richiede $O\left(n^2\right)$ test CI per ogni coppia di nodi; $O\left(n^4\right)$
- **Fase di potatura**
  richiede $O\left(n^2\right)$ test CI per ogni arco presente; $O\left(n^4\right)$

Conoscenza a priori: introduzione

- Per conoscenza a priori si intende tutto il bagaglio di esperienza che può fornire una persona esperta all'agente prima di effettuare l'apprendimento
- Tuttavia, a seconda del modello usato per rappresentare la conoscenza, può risultare molto difficile sfrutturla, come nel caso di:
  - Reti Neurali
  - Alberi Decisionali
- Invece, la Rete Bayesiana è dotata di una semantica che permette facilmente di sfrutture la conoscenza a priori fornita da un esperto

Conoscenza a priori & TPDA

- L'algoritmo TPDA permette l'inserimento di conoscenza attraverso diversi meccanismi:
  - si può sostituire la fase di bozza fornendo un grafo iniziale
  - si può fornire un ordinamento (anche parziale) dei nodi della rete,
    ovvero relazioni di causa-effetto indirette
  - specificando alcuni archi del grafo diretto,
    ovvero relazioni di causa-effetto dirette

Conoscenza a priori: acquisizione

- Il processo di acquisizione della conoscenza di un esperto richiede due passaggi:
  1. L'esperto comunica, con il **linguaggio naturale**, la sua conoscenza ad un intermediario in grado di comprenderlo
  2. Il compito dell'intermediario è quello di **tradurre** la conoscenza nel formalismo matematico necessario per descrivere la rete

Conoscenza: interfaccia d'acquisizione

- Nell'ambito dello sviluppo, da parte dell'HP, di un agente intelligente, per aiutare i suoi clienti, a risolvere i malfunzionamenti delle stampanti è nata la necessità di fondere la conoscenza maturata da tante persone
- Far svolgere questo compito ai pochi intermediari disponibili sarebbe stato uno sforzo titanico
- La soluzione è stata quella di creare un'interfaccia grafica ad hoc in modo da sostituire la persona con un'applicazione

Conoscenza: interfaccia d'acquisizione

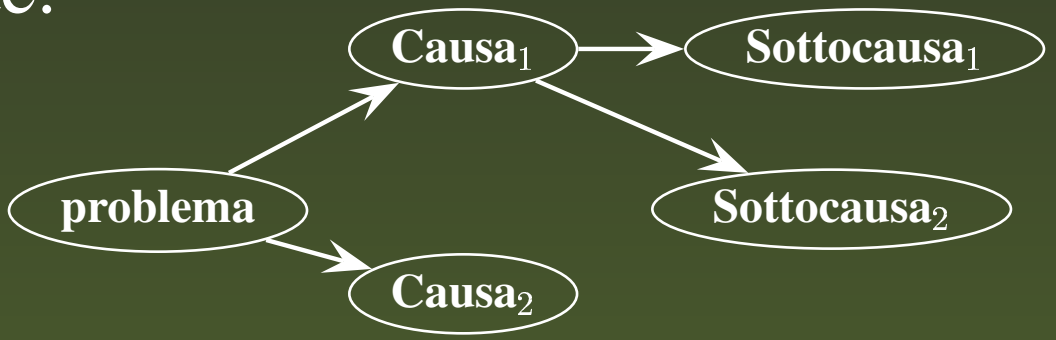
- I requisiti fondamentali per una tale interfaccia sono:
  - guidare l'esperto, attraverso passi intuitivi e naturali, lungo il processo di costruzione della rete
  - poter specificare probabilità di cause o effetti anche in modo non-causale, a seconda delle preferenze

Conoscenza: interfaccia d'acquisizione

- Nell'ambito della risoluzione dei problemi si possono suddividere i nodi in tre categorie:
  - di **Causa**: è il nodo che rappresenta le possibili cause del malfunzionamento
  - di **Azione**: questi nodi rappresentano azioni che possono risolvere il problema a seconda della causa
  - di **Domanda**: servono all'agente per raccogliere informazioni dall'utente circa la natura del malfunzionamento
- L'interfaccia grafica presenta all'esperto varie finestre che consentono di specificare l'intera rete bayesiana

Interfaccia d'acquisizione: Cause

- Si chiede all'esperto di specificare tutte le cause possibili di malfunzionamento e le loro probabilità
- Per semplificare la specifica, è possibile strutturare le cause in sottocause come mostra l'albero seguente:



- Le probabilità possono essere specificate in modo causale, ad esempio: $P\left(Sottocausa_1 \mid Causa_1\right)$
- Dal punto di vista dell'agente **solo le foglie** dell'albero sono importanti, il resto verrà ignorato

Interfaccia d'acquisizione: Azioni

- Per ogni causa dichiarata, si richiede all'esperto di fornire un elenco delle possibili azioni per risolvere il problema
- Inoltre, si richiede anche:
  - la probabilità di risolvere il problema supponendo che l'azione sia stata svolta correttamente:
    $P\left(Azione = yes \,|\, I = f, corretta, requisiti\right)$
  - le probabilità di eseguire tutto correttamente:
    $P\left(corretta\right)$ e $P\left(requisiti\right)$
- Infine, si specifica il costo dell'azione, che racchiude difficoltà, tempo e energie necessarie

Interfaccia d'acquisizione: Domande

- si dividono in due categorie
  - *diagnostiche*: sono poste per individuare gli effetti del malfunzionamento; ad esempio: "la stampa è troppo chiara?"
  - *generali*: sono poste per avvalorare una possibile causa; ad esempio "da quanto tempo non si cambia il toner?"
- È più intuitivo specificare la probabilità di una causa data la domanda che il viceversa, quindi:
  - si selezionano la domanda e l'elenco delle cause associate
  - si specifica $P(I = f_i \,|\, Q)$ $1 \leq i \leq k$ e $P(Q)$

Conoscenza: interfaccia d'acquisizione

- Alla fine del processo di acquisizione, le informazioni fornite dall'esperto sono sufficienti a realizzare una rete bayesiana naive pronta per essere usata da un agente, e perfezionata man mano che viene utilizzata