Appunti di Intelligenza Artificiale – Silvio Moioli

- A1 Introduzione
  - Definizione di IA: studio delle macchine che...
    - fanno ciò che l'uomo sa far meglio
    - superino il test di Turing (originale, totale)
    - pensino intelligentemente (IA forte)
    - si comportino come se pensassero intelligentemente (IA debole)
    - migliorino i sistemi software (IA molto debole)
    - sono agenti razionali (Russel-Norvig)
- A2 Applicazioni
  - DARPA Grand Challenge: robotica, visione, pianificazione, apprendimento...
  - Deep Blue: scacchi
  - Mars Exploration Rovers: veicoli autonomi
  - Microsoft help, sistemi medicali: sistemi esperti
  - Google: ricerca, elaborazione del linguaggio naturale, traduzione automatica
  - Dragon: riconoscimento vocale
  - Reti neurali: simulazione, clustering...
  - Sistemi di interpretazione dati: analisi, spiegazione, valutazione empirica, analogia con casi passati...
- A3 Agenti
  - Visione unificante della materia
  - Funzione dell'agente: azione per ogni possibile sequenza di percezioni
  - Misura della prestazione: funzione che indica la “bontà” delle azioni intraprese dall'agente
  - Agente razionale: massimizza la prestazione o la prestazione attesa
  - Ambiente di lavoro: PEAS (Performance, Environment, Actuators, Sensors)
  - Tipologia di ambienti
    - Deterministico/stocastico
    - Episodico/sequenziale
    - Continuo/discreto
    - Completamente/parzialmente osservabile
      - i sensori percepiscono gli aspetti significativi dell'ambiente?
    - Statico/dinamico
      - l'ambiente cambia mentre l'agente decide?
      - Semi-staticità: la prestazione cambia mentre l'agente decide?
    - Singolo/Multiplo agente
  - Tipi di agente
    - Table-driven
    - Simple-reflex
    - Model based-reflex
      - può far fronte ad aspetti non osservabili dell'ambiente
    - Model based-goal based
      - può far fronte ad ambienti sequenziali;
      - introduce problemi di pianificazione e ricerca;
      - obiettivo univocamente determinato;
    - Model based-utility based
      - può far fronte ad ambienti stocastici, con più obiettivi, obiettivi irraggiungibili con certezza
      - obiettivi espressi da una funzione da massimizzare
    - Learning
      - agente in grado di imparare. Componenti:

        - Performance element: uno degli agenti visti finora
        - Critic: valutazione delle prestazioni
        - Learning element: modifica il Performance element in base al Critic
        - Problem generator: suggerisce azioni (sub-ottime) da compiere per l'apprendimento
- A5.1 Ricerca nello spazio degli stati
    - Agenti che operano in ambienti completamente osservabili, deterministici, discreti, statici, sequenziali, ad agente singolo
        - Funzionamento
            - Definizione del problema
                - Stato di partenza
                - Funzione successore (coppie azione-nuovo stato)
                - Obiettivo
                - Costo (tra due nodi adiacenti)
            - Definizione dell'obiettivo
            - Ricerca di una soluzione
                - Soluzione ottima: a costo minimo
            - Attuazione della soluzione
        - Valutazione
            - completezza
                - se c'è una soluzione, la trova?
            - ottimalità
                - la soluzione trovata ha costo minimo?
            - complessità nel tempo
            - complessità nello spazio
        - Come stabilire la complessità
            - benchmarking
            - analisi precisa (es. numero di istruzioni eseguite)
            - analisi asintotica (notazione O grande)
            - classe di complessità (P, NP, NP-completo)
- A5.2 Ricerca non informata
        - Algoritmi
            - General Tree Search
            - General Graph Search
            - Breadth-first search: il bordo è una coda
                - Completo (se b, branching factor, non è infinito)
                - Ottimo (se costo non decrescente in d, distanza soluzione)
                - Spazio: esponenziale in d
                - Tempo: esponenziale in d
            - Uniform-cost search: il bordo è una coda ordinata per costo crescente
                - Completo (se costo strettamente positivo)
                - Ottimo
                - Spazio: esponenziale nel rapporto fra costo della soluzione e minimo
                - Tempo: esponenziale nel rapporto fra costo della soluzione e minimo
            - Depth-first search: il bordo è una pila
                - Non completo
                - Non ottimo
                - Spazio: lineare in b*m (diametro spazio)
                - Tempo: esponenziale in m
            - Backtracking search: variante del depth-first che espande un nodo alla volta
                - Non completo
                - Non ottimo

        - Spazio: lineare in m
        - Tempo: esponenziale in m
      - Depth-limited search: variante del depth-first che espande al più l livelli
        - Non completo
        - Non ottimo
        - Spazio: lineare in b*l
        - Tempo: esponenziale in l
      - Iterative deepening depth-limited search: itera depth-limited search aumentando progressivamente l
        - Completo (se b non è infinito)
        - Ottimo (se costo non decrescente in d)
        - Spazio: lineare in d
        - Tempo: esponenziale in d
      - Bidirectional search: due ricerche una dalla sorgente e una dalla destinazione
        - Completo (se b non è infinito)
        - Ottimo (se costo non decrescente in d)
        - Spazio: lineare in d
        - Tempo: esponenziale in d/2
        - Attenzione: può non essere semplice calcolare il precedente
    - Ricerca in ambienti non osservabili (agente senza sensori)
      - stato creduto (belief state): insieme di stati in cui l'agente si potrebbe trovare
      - obiettivo: portarsi in uno stato creduto i cui stati reali corrispondenti sono tutti obiettivi
    - Ricerca in ambienti parzialmente osservabili
      - interfogliata con l'analisi dei nuovi dati
- A5.3 Ricerca informata
  - Funzioni euristiche
    - h(x): stima del minimo costo da x alla soluzione
    - ammissibilità: non sovrastima mai
    - consistenza: vale la disuguaglianza triangolare h(x)<=g(y)+h(y)
      - se h(x) è consistente allora f(x) è non decrescente
  - Algoritmi
    - Greedy best-first search: f(x), funzione di valutazione, è h(x)
      - Non completo
      - Non ottimo
      - Spazio: esponenziale in m, buoni coefficienti moltiplicativi se h è buona
      - Tempo: esponenziale in m, buoni coefficienti moltiplicativi se h è buona
    - A* search: f(x)=g(x)+h(x)
      - Completa (sugli alberi se h(x) è ammissibile, sui grafi se h(x) è consistente)
      - Ottima (sugli alberi se h(x) è ammissibile, sui grafi se h(x) è consistente)
      - Spazio: esponenziale in d, buoni coefficienti moltiplicativi se h è buona
      - Tempo: esponenziale in d, buoni coefficienti moltiplicativi se h è buona
- A6 Conoscenza e ragionamento
  - Agenti basati su conoscenza
    - Composti da
      - Base di conoscenza (KB, knowledge base)
        - Contiene
          - Dati iniziali
          - Percezioni
          - Azioni precedenti
        - Tipi di conoscenza
          - Empirica (funziona in pratica, non si sa perchè)

        - Causale (se ne conosce un modello)
        - Euristiche (si può dire in generale anche senza modello)
    - Motore di inferenza
        - Può essere usato per decidere azioni
        - Può essere usato per dedurre conoscenze non esplicite
        - Pattern di ragionamento
            - Deduttivo (conseguenze logiche dalle premesse)
            - Induttivo (generalizzazione)
            - Abduttivo (spiegazione)
- A7.1 Agenti logici, A7.2 Logica proposizionale, A7.3 Inferenza in PL
    - Logica: modo per rappresentare la conoscenza e "ragionare" su di essa
    - Logica Proposizionale
        - Frasi: entità definite secondo la sintassi che possono essere vere secondo la semantica
        - Sintassi (regole che definiscono le frasi ben formate)
            - Frasi atomiche: Vero, Falso, Simbolo
            - Frasi complesse: $\neg P$, $P \, connettivo \, Q$ dove $connettivo = \{ \wedge \vee \Rightarrow \Leftrightarrow \}$
        - Semantica (regole che definiscono la verità di una frase rispetto a un modello)
            - Modello: assegnamento di un valore di verità ad ogni simbolo
                - Spesso il modello è un'astrazione del mondo reale
            - Regole: tavole di verità
        - Proprietà delle frasi
            - Validità: la frase è verificata in ogni modello
            - Soddisfacibilità: la frase è verificata in almeno un modello
            - Implicazione logica: si dice che α è implicata in β e si scrive $\alpha \models \beta$ se e solo se β è vera in ogni modello in cui α è vera
            - Equivalenza logica: si dice che α è equivalente a β e si scrive $\alpha \equiv \beta$ se e solo se β assume gli stessi valori di α in ogni modello
            - Teorema di deduzione: $\alpha \models \beta$ se e solo se $\alpha \Rightarrow \beta$ è valida
            - Teorema della prova per refutazione: $\alpha \models \beta$ se e solo se $\alpha \wedge \neg \beta$ non è soddisfacibile
            - Letterale: una frase è un letterale se è atomica (letterale positivo) o la negazione di una frase atomica (letterale negativo)
            - Clausola: una frase è una clausola se è una disgiunzione di letterali
            - Clausola di Horn: una frase è clausola di Horn se al più uno dei suoi letterali è positivo
            - Clausola di Horn definita: una frase è clausola di Horn definita se esattamente uno dei suoi letterali è positivo
            - Forma normale congiuntiva (CNF): una frase è in CNF se è una congiunzione di disgiunzioni di letterali
        - Algoritmi di inferenza (ricerca di frasi α dalla base di conoscenza)
            - Caratteristiche
                - Complessità nello spazio
                - Complessità nel tempo
                - Correttezza: le frasi α sono implicate dalla base di conoscenza
                - Completezza: se esiste una frase α implicata, la trova
            - Model checking: applicazione della definizione
                - Corretto
                - Completo
                - Spazio: lineare in n, dimensione della base di conoscenza
                - Tempo: esponenziale in n
            - Ricerca nello spazio delle basi di conoscenza (per problemi monotoni)

    - Stati: basi di conoscenza
    - Funzioni successore: regole di inferenza
      - And elimination $\frac{\alpha \wedge \beta}{\alpha}$
      - Modus Ponens $\frac{\alpha \Rightarrow \beta, \beta}{\beta}$
      - Regole di equivalenza logica
    - Corretti
    - Completezza: dipende dall'algoritmo di ricerca usato
    - Complessità: il caso peggiore non è mai polinomiale, i casi pratici possono andar meglio
  - Risoluzione
    - Algoritmo di ricerca come i precedenti
    - Stato iniziale: $KB \wedge \neg \alpha$ (prova per refutazione)
    - Accetta solo basi in CNF
      - si dimostra che ogni frase può essere espressa in CNF
    - Usa una sola regola di inferenza (risoluzione):
      $\frac{l_1 \wedge l_2 \wedge ... \wedge l_n, m_{j_1} \wedge m_{j_2} \wedge ... m_{j_m}}{\bigwedge_{i \neq j_k} l_i \wedge \bigwedge_{j > n} m_j}$ dove $l_j = \neg m_j$
    - Corretto
    - Completo per refutazione (non può enumerare i casi)
    - Spazio: lineare in n
    - Tempo: esponenziale in n (caso generale)
    - Tempo: lineare in n (se la base è in clausole di Horn definite)
      - In quel caso il metodo prende il nome di Modus Ponens iterativo ed è completo
      - Esistono due varianti
        - forward chaining (deduci tutto il deducibile dalla base di conoscenza)
          - bisogna tener traccia dei nodi già processati
          - bisogna tener traccia del numero di condizioni mancanti
        - backward chaining (tenta di dimostrare o refutare una proposizione)
          - depth-first
          - bisogna tener traccia dei nodi già dimostrati/di cui si sa non esiste dimostrazione
- A7.4 Logica del primo ordine, A7.5 Basi di conoscenza nella logica del primo ordine, A7.6 Inferenza nella logica del primo ordine
  - Vantaggi del modello logico rispetto al modello procedurale
    - dichiaratività: il motore di inferenza è indipendente dal dominio, la conoscenza è più flessibile
    - permette di gestire fatti non completamente specificati
    - composizionalità: aggiungere fatti non invalida le conoscenze precedenti
  - Logica del primo ordine
    - Ontologia
      - Descrizione della natura della realtà (in una certa logica)
      - Oggetti, relazioni, funzioni
    - Epistemologia
      - Forme che la conoscenza può assumere (in una certa logica)
      - Come in PL: vero, falso, non decidibile
    - Sintassi
      - Costanti vero, falso

- Simboli
  - costanti per gli oggetti
  - predicati per le relazioni
  - funzioni per le funzioni
- Operatore di uguaglianza =, connettivi logici di PL
- Quantificatori: $\forall \exists$

■ Semantica
- Modello: insieme di
  - oggetti
  - relazioni n-arie: insiemi di n-uple di oggetti
    - le relazioni unarie si dicono proprietà
  - funzioni: relazioni in cui ogni argomento corrisponde a uno e un solo valore
- Interpretazione: assegnamento dei simboli agli oggetti, relazioni, funzioni del modello
- Una frase assume valore di verità fissato un modello e un'interpretazione (detta originale)
- Un predicato è vero se i suoi simboli corrispondono ad oggetti nella relazione corrispondente
- Una frase $\alpha = \beta$ è vera se α e β si riferiscono allo stesso oggetto
- Gli operatori logici hanno lo stesso significato di PL
- Una frase $\forall x \alpha$ è vera se α è vera in tutte le possibili interpretazioni estese
  - interpretazione estesa: assegnamento ad x di ogni possibile oggetto del modello
- Una frase $\exists x \alpha$ è vera se α è vera in almeno una interpretazione estesa
- Valgono le regole di De Morgan sui quantificatori

■ Algoritmi di inferenza
- Proposizionalizzazione: riscrivo la base di conoscenza in PL e uso i risultati noti
  - Quantificatore universale: ogni interpretazione estesa diventa una frase
    - Sostituzione: funzione $SUBST(\theta, \alpha)$ dove $\theta = \{x|y\}$ sostituisce tutte le occorrenze di x in α con y
    - Ground term: termine senza variabili
    - regola detta di istanziazione universale $\frac{\forall v \alpha}{SUBST(\{v|g\}, \alpha)}$ per tutti i ground term g
    - infinite proposizioni se si usano funzioni
    - si ottiene una base di conoscenza logicamente equivalente
  - Quantificatore esistenziale: una opportuna interpretazione estesa diventa una proposizione
    - regola detta di skolemizzazione: $\frac{\exists e \alpha}{SUBST(\{e|s\}, \alpha)}$ dove s può essere una costante o una funzione delle altre variabili, il cui simbolo non appare altrove nella base di conoscenza
    - si ottiene una base di conoscenza inferenzialmente equivalente: è soddisfacibile laddove l'originale era soddisfacibile
  - Ground term: variabili in PL
  - Corretto
  - Completezza: semidecidibile (se una frase non è implicata, l'algoritmo non termina, teorema di Herbrand)
  - Spazio: esponenziale in n, intrattabile
  - Tempo: esponenziale in n, intrattabile

- Modus Ponens generalizzato
  - Applicazione iterata della regola $\frac{p_1', p_2', \ldots, (p_1 \wedge p_2 \wedge \ldots \Rightarrow q)}{SUBST(\theta, q)}$ dove θ è tale che $SUBST(\theta, p_1') = SUBST(\theta, p_1)$, $SUBST(\theta, p_2') = SUBST(\theta, p_2)$ e così via.
  - Modo diverso di esprimere la condizione: $\theta = UNIFY(p', p)$
    - UNIFY è detto processo di unificazione, o pattern matching
    - E' un problema NP
  - Richiede che la base di conoscenza sia in clausole di Horn definite
  - Varianti
    - Forward chaining (sistemi a produzioni, Jess)
      - Due fatti si considerano uguali se differiscono solo nel nome delle variabili
      - Corretto e completo se non ci sono funzioni (la base di conoscenza è detta Datalog)
      - Variante: algoritmo RETE
        - ogni nuovo fatto deve essere conseguenza di almeno un fatto al tempo t-1, vengono memorizzati i match parziali
    - Backward chaining (Prolog)
      - basato su stack, inizialmente riempito con gli obiettivi
      - ad ogni passo cerco proposizioni che unificano con la testa dello stack
      - Corretto, incompleto
- Risoluzione (dimostratori di teoremi)
  - La base di conoscenza (qualsiasi) è convertita in CNF e skolemizzata
  - Si possono eliminare letterali in cui uno unifica con la negazione dell'altro: $\frac{l_1 \vee l_2 \vee \ldots, m_1 \vee m_2 \vee m_3 \ldots}{SUBST(\theta, l_1 \vee m_3 \vee \ldots)}$ dove $\theta = UNIFY(l_2, m_2)$
  - Completezza: refutation complete (se esiste la refutazione la trova, altrimenti può continuare la ricerca all'infinito)

- A8.1, A8.2, A8.3 Rappresentazione della conoscenza
  - Gli agenti basati sulla logica hanno bisogno di una rappresentazione del mondo nella base di conoscenza
    - diverse logiche risolvono problematiche diverse
    - qualunque sia la logica, esistono sempre problemi di fondo (es. abduzione)
  - Ontologia: problema di come rappresentare le parti interessanti del mondo
    - Ontologie generali: problema filosofico aperto
      - Problemi
        - Conoscenza di tipo diverso
        - Problemi di “integrabilità”
      - Tecniche generali
        - Tassonomie
          - Suddivido tutti gli oggetti in categorie con proprietà uniformi all'interno
          - Categorie in FOL
            - Predicati (es. $Basketball(b)$)
            - Oggetti (es. $Member(b, Basketballs)$)
          - Relazioni gerarchiche tra categorie: $\subset$, SubsetOf
          - Appartenenza a una categoria: $\in$, Member
          - Relazioni tra categorie in base agli oggetti
            - Disgiunte: non hanno oggetti in comune
            - Scomposizioni esaustive: ogni oggetto di una categoria è in almeno una delle sue scomposizioni esaustive

            - Partizioni: scomposizioni esaustive disgiunte di una categoria
        - Categorie tipiche: catturano gli aspetti comuni alla maggior parte degli oggetti
    - Composizione di oggetti
        - Relazioni PartOf()
            - transitive
            - riflessive
        - Proprietà intrinseche: che si mantengono dal tutto alla parte
        - Proprietà estrinseche: del tutto ma non della parte
        - Oggetti
            - Sostanze: con sole proprietà intrinseche
            - Cose: anche con proprietà estrinseche
    - Reti semantiche
        - Famiglia di linguaggi grafici riconducibili a FOL
            - generalmente meno potenti di FOL
            - generalmente più semplici e trasparenti da capire e usare
            - quelli più potenti hanno anche meccanismi extra-logici(procedurali)
        - Permettono di specificare relazioni binarie (n-arie con reificazione)
            - in particolare ereditarietà, sussunzione
            - in particolare ragionamento per default
        - Inferenza: visita appropriata del grafo
        - Esempi particolari
            - Frames: simili al paradigma OO, parti procedurali
            - Logiche descrittive (es. linguaggio CLASSIC) in alcuni casi più chiare e concise di FOL, riconducibili a FOL
- Ontologie del tempo: situation calculus
    - Formalizzazione del tempo in FOL
        - Permette
            - Proiezione: cosa succede se eseguo una sequenza di azioni?
            - Pianificazione: qual'è la sequenza di azioni (ottima) per ottenere un certo risultato?
    - Situazioni
        - istanti di tempo modellati da oggetti
        - situazione iniziale S0
        - funzione Result(a, s) ritorna la situazione che deriva dall'esecuzione dell'azione a alla situazione precedente s
    - Azioni
        - possono essere raggruppate in sequenza ed eseguite una a una: Result([a, r], s) = Result(r, Result(a,s))
    - Fluenti
        - predicati o funzioni che variano con le situazioni, situazione ultimo parametro
    - Predicati atemporali
        - non fluenti
    - Assiomi
        - Determinano come le azioni influenzano le situazioni e viceversa
        - Primo modello
            - Un assioma di possibilità per ogni azione, descrive le condizioni alla possibilità di effettuare l'azione: $condizioni \Rightarrow Poss(a, s)$
            - Un assioma degli effetti per ogni azione: $Poss(a, s) \Rightarrow postcondizioni$
            - Due assiomi dell'effetto per ogni coppia fluente-azione, uno positivo e uno negativo, che descrivono quando l'azione rende vero o falso il fluente

(cosa cambia)
        - Frame axioms che descrivono cosa non cambia
            - introducono inefficienza, O(AF) letterali, detto "representational frame problem"
    - Secondo modello: risolve il representational frame problem
        - Assiomi della possibilità e degli effetti come sopra
        - Assiomi dello stato successore, per ogni effetto e di ogni azione:
          $Poss(a, s) \Rightarrow$(F vero in s $\Leftrightarrow$(F è diventato vero in conseguenza di e $\vee$il fluente era ve
            - osservazione: generalmente il numero massimo di effetti E dovuti da ogni azione è inferiore a quello dei fluenti F, O(AE) è meglio di O(AF)
    - Altri problemi
        - Ramification problem: caratterizzazione dei cambiamenti implicati dalle azioni che sopravvivono al variare delle situazioni
        - Qualification problem: gli assiomi di effetto generalmente sono troppo poco specificati – si omettono molti fatti che generalmente falsi.
            - Possibili soluzioni
                - Completamento esaustivo delle condizioni (completamento di Clark)
                - Modifiche al motore di inferenza: si assume che ciò che non è detto esplicitamente è falso (closed world assumption)
        - Generazione di piani
            - inefficiente con il sistema descritto
            - sistema più usato derivato da A* (estende FOL): STRIPS
                - idea: una base di conoscenza modificabile
        - Casi non trattabili dal sitiuation calculus
            - tempo continuo
            - intervalli di tempo irregolari
            - condizioni che non siano precondizioni o postcondizioni di un'azione
            - concorrenza
            - eventi esterni...
- A9.1 Incertezza
    - Trattamento dell'incertezza dei dati: basi di conoscenza con elementi di probabilità
        - probabilità come modo per riassumere gli aspetti non modellati (laziness) o non noti (theoretical/practical ignorance) del problema
    - Trattamento dell'incertezza degli obiettivi: teoria dell'utilità combinata con la probabilità (teoria delle decisioni)
    - Teoria della probabilità
        - Ontologia
            - la stessa di PL, ogni fatto può essere vero o falso secondo un modello
        - Epistemologia
            - la probabilità esprime il grado di credenza dell'agente
        - Sintassi
            - Elemento atomico: variabile casuale
                - Possibili valori assunti dalla variabile: dominio
            - Connettivi logici come in PL
        - Assiomi della probabilità
            - $0 \leq P(a) \leq 1$
            - $P(vero) = 1 \quad , \quad P(falso) = 0$
            - $P(a \vee b) = P(a) + P(b) - P(a, b)$
        - Tabella completa delle congiunte: matrice della probabilità congiunta di tutte le variabili casuali del modello

- Probabilità condizionata: $P(a|b)=\frac{P(a,b)}{P(b)}$
- Regola del prodotto: $P(a,b)=P(a|b)P(b)$
  - Notazione compatta $\boldsymbol{P}(a,b)=\boldsymbol{P}(a|b)\boldsymbol{P}(b)$ per ogni valore assumibile da a e b
- Variabili indipendenti: $P(a,b)=P(a)P(b)$ oppure $P(a|b)=P(a)$
- Indipendenza condizionale: $P(X,Y|Z)=P(X|Z)P(Y|Z)$ (X ed Y sono indipendenti data Z)
- Teorema di Bayes: $P(b|a)=P(a|b)\frac{P(b)}{P(a)}$
  - Utilità: ottenere regole causali da regole diagnostiche; le ultime tendono ad essere più fragili
- Regola della catena: $P(X_1,X_2,...X_n)=\prod_{i=1}^{n}(X_i|X_{i-1},X_{i-2},...X_1)$
- Teorema delle probabilità totali $P(a)=\sum_i P(a,e_i)$ dove e_i formano una partizione di a

- Possibili richieste alla base di conoscenza nella forma $\boldsymbol{P}(\boldsymbol{X}|\boldsymbol{E})$, $\boldsymbol{Y}=[variabili \notin \boldsymbol{X}]$
  - **X**: query
  - **E**: evidence
  - **Y**: hidden variables
  - Algoritmo: somma sulle hidden variables, applicazione della regola della catena con evidence fissate
- Possibili basi di conoscenza
  - Tabella completa delle congiunte
    - utilizzo della memoria intrattabile: O(d^n), d massima arità delle variabili
    - difficile da costruire
  - Tabelle delle congiunte di gruppi di variabili scorrelate
    - Riduce le dimensioni, comunque difficilmente trattabile
    - vera indipendenza rara
  - Tabelle delle variabili condizionalmente scorrelate
    - condizione più debole della precedente: è possibile scomporre in più pezzi più piccoli
    - utilizzo della memoria O(n)
    - semplificazione del calcolo della regola di Bayes
    - condizione più facilmente ottenibile in pratica
    - Possibilità di implementare il "naive Bayes model"
      - una causa influenza direttamente più effetti condizionalmente indipendenti data la causa

- A9.2 Ragionamento probabilistico
  - Reti bayesiane
    - Notazione grafica sintetica per le asserzioni di indipendenza condizionale tra variabili
    - Sintassi
      - Grafo orientato aciclico (DAG)
        - nodi: variabili casuali
        - archi orientati: relazioni di dipendenza condizionale, orientati da X a Parent(X)
        - ad ogni arco è associato un insieme di probabilità condizionate $\boldsymbol{P}(X|Parents(X))$, per le variabili discrete organizzate in una tabella (CPT, Conditional Probability Table)

- Semantica
  - le CPT non riportano probabilità dipendenti
  - Dalle CPT si ricavano le congiunte: $P(x_{1,}x_{2,}...)=\prod_{i=1}^{n} P(X_i|Parents(X_i))$

    (deriva dalla chain rule tenendo conto dell'indipendenza tra le variabili)
    - Le CPT sono rappresentazioni compatte della tabella completa delle congiunte
    - Spazio: $O(n2^k)$ dove k è il massimo numero di $Parents(X_i)$
- Costruzione
  - Occorre stabilire la sequenza delle variabili casuali
  - Ogni variabile casuale può dipendere da alcune precedenti (genitori)
  - Diversi ordinamenti portano a diversi grafi con diverse prestazioni
    - reti ottime costruite con l'ordine causale
    - meno ottime costruite con l'ordine diagnostico
    - "saltare" peggio
    - comprensibilità vs. compattezza di rappresentazione
- Complessità
  - Caso generale: NP-difficile
  - Caso semplice, in cui il grafo è un polytree (c'è al più un cammino diretto tra ogni coppia di nodi): lineare
  - Esistono metodi approssimati