Modelli Probabilistici per le Decisioni

UniShare

Davide Cozzi
@dlcgold

Indice

Capitolo 1

Introduzione

Questi appunti sono presi a lezione. Per quanto sia stata fatta una revisione è altamente probabile (praticamente certo) che possano contenere errori, sia di stampa che di vero e proprio contenuto. Per eventuali proposte di correzione effettuare una pull request. Link: `https://github.com/dlcgold/Appunti`.

Capitolo 2

Ripasso di Probabilità

Riprendiamo qualche definizione.

Definizione 1. *Definiamo **variabile casuale** come un'osservazione, un esito o un evento il cui valore è incerto.*

Definizione 2. *Definiamo **dominio o spazio degli eventi** come l'insieme dei possibili valore che può assumere una variabile casuale.*

Definizione 3. *Definiamo **spazio di probabilità o modello di probabilità** come uno spazio degli eventi corredato da un assegnamento:*

$$P(\omega), \omega \in \Omega$$

tale che:

- $0 \leq P(\omega) \leq 1$
- $\sum_{\omega} p(\omega) = 1$

con omega evento e Ω spazio degli eventi.

Definizione 4. *Definiamo **evento atomico o campione** una specificazione completa del calore delle variabili casuali di interesse.*
L'insieme di tutti i possibili eventi atomici è:

- *mutualmente esaustivo (non potendo accadere altro)*
- *mutualmente esclusivo (può accadere solo un evento atomico di quelli possibili)*

Definizione 5. *Definiamo un **evento** (non atomico) A può essere un qualunque sottoinsieme di Ω tale che:*

$$P(A) = \sum_{\omega \in A} P(\omega)$$

Definizione 6. *Definiamo una* ***variabile aleatoria*** *è una variabile che può assumere diversi valori in corrispondenza di altrettanti eventi che costituiscono una partizione dello spazio delle probabilità.*

Si ricorda che, per una variabile a e una b:

- $0 \leq P(a) \leq 1$
- $P(\top) = 1$ e $P(\bot) = 0$
- $P(a \vee b) = P(a) + p(b) - p(a \wedge b)$

Definizione 7. *Definiamo una* ***probabilità condizionata*** *rappresenta la verosimiglianza che un evento a si verifichi se b si verifica e si denota con:*

$$P(a|b)$$

Si ha quindi la specifica che alcuni eventi rendono altri eventi più o meno verosimili.
Si parla quindi di eventi ***dipendenti****.*

Definizione 8. *Due eventi sono* ***indipendenti*** *se un evento non influisce sulla realizzazione dell'altro:*

$$P(a|b) = P(a)$$

Si ha quindi la seguente regola.

Teorema 1 (Regola del prodotto)**.** *Possiamo calcolare che due eventi si verifichino contemporaneamente tramite la probabilità condizionata e quella dei singoli eventi:*

$$P(a, b) = P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$$

Con $P(a, b) = P(a \wedge b)$ è detta ***probabilità congiunta*** *("=" perché sono due modi per scriverla).*

Posso fare la tabella dei vari eventi condizionati.

Teorema 2 (Regola della somma)**.** *Si ha che, avendo la tabella degli eventi:*

$$P(x) = \sum_{y} P(x, y)$$

con $P(x)$ detta ***probabilità marginale****.*

La somma di tutte le possibili combinazioni di eventi, quindi dei valori della tabella, deve dare 1.
Su slide esempio di uso di quanto detto, dove si arriva al teorema di Bayes.
Si vuole infatti passare dal conoscere $P(a|b)$ al consocere $P(b|a)$.

Teorema 3 (Teorema di Bayes)**.** *Il teorema enuncia che:*

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Avendo:

- $P(h)$ *che è la probabilità conosciuta a priori di* h*. Tale probabilità riflette qualsiasi conoscenza di base sulla possibilità che* h *sia corretta*
- $P(D)$ *che è la probabilità conosciuta a priori di* D*, ovvero la probabilità che* D *sia osservato*
- $P(D|h)$ *che è la probabilità di osservare* D *in presenza dell'ipotesi* h
- $P(D|h)$ *che è la probabilità a posteriori di* h*. Tale probabilità riflette la "confidenza" di avere* h *dopo che* D *è stato osservato*

In altri termini, avendo:

$$P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$$

ho che:

$$P(a|b)P(b) = P(b|a)P(a)$$

arrivando a dire che:

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)}$$

notando la correlazione tra probabilità congiunta e Bayes.
Che è il punto fondamentale della moderna teoria dell'intelligenza artificiale in quanto permette di raccogliere l'evidenza senza poi usare le tabelle delle probabilità congiunte, che sarebbero difficilissimi da osservare. Se pensiamo ad alcuni eventi come cause "nascoste" non necessariamente osservabili se modelliamo la verosimiglianza degli eventi osservabili date le cause nascoste si ha:

$$P(causa|effetto) = \frac{P(effetto|causa)P(causa)}{P(effetto)}$$

Si ha quindi un modello per inferire/derivare la verosimiglianza della causa nascosta e quindi rispondendo a:

$$P(causa|effetto)$$

Avendo quindi la probabilità di una causa dato un effetto. *Dato l'effetto modello la causa.*
Se non si ha una delle due probabilità a priori posso stimare per poi normalizzare. In altri termini il denominatore $P(D)$ è spesso solo una quantità di normalizzazione, essendo spesso difficile da stimare.
Le probabilità possono essere definite su due approcci:

- **approccio frequentista o oggettivista** che considera la probabilità come un'entità misurabile legata alla frequenza di accadimento, come si fa ne machine learning
- **approccio soggettivista** che considera la probabilità come una misura del grado di attesa soggettivo del verificarsi di un evento, come si fa nel corso di statistica

Capitolo 3

Modelli Probabilistici

Nel passato si sono usati **sistemi a regole**, dove codificando tutto quello che può succedere si cercava di giungere ad una decisione. Questo però era molto dispendioso, si arrivava o a vero a falso, senza via di mezzo, e si dovevano avere dati ipoteticamente completi e sicuri in partenza. Si parla in questo caso di **modelli logici**.
Viviamo in un'era dove si hanno molti dati, sia in ambito sociale, che di business che scientifico. Questi dati devono essere analizzati al fine di poter prendere **decisioni** e per farlo si deve per capire la situazione in cui ci si trova e spesso posso capirlo solo osservando i dati, non osservando la variabile specifica. Dalle osservazioni dobbiamo inferire il valore di variabili "nascoste". Spesso si ha a che fare con dati non completi, non consistenti, spesso errati, con rumore di trasmissione etc...
Tali dati sono comunque evidenze per percepire la situazione in cui ci si trova. L'obiettivo del corso è fornire strumenti modellistici per rappresentare l'incertezza nel modello, incertezza per struttura e parametri, e per rappresentare in termini probabilistici gli errori nei dati. Si vuole quindi implementare algoritmi di "ragionamento", automatizzati e adattivi, oltre che robusti e scalabili.
I **modelli probabilistici** sono anche detti **modelli generativi**. Si usa la teoria delle probabilità per esprimere incertezza e rumore associati al modello e ai dati, soprattutto usando la teoria Bayesana, per fare previsione e adattare i modelli. Questi modelli permettono di partire da una "credenza" iniziale, anche soggettiva, per poi raccogliere evidenze aggiustando tale modello.
I modelli probabilistici sono anche modelli di machine learning, in quanto apprendono.
Bisogna quindi partire dalle osservazioni generate rispetto ad un valore di variabile per poi inferire tale variabile (ad esempio parto dai risultati di un gioco per capire quando è bravo il giocatore, che non è una variabile che posso

sapere a priori). Si parte dai dati e si arriva al valore della variabile che ha generato questi dati (per questo *modello generativo*). Man mano che raccolgo informazioni raffino il modello, più o meno come fa un essere umano ("più rispondi alle domande all'orale e più il docente capisce il tuo voto, anche se alla fine non si ha la certezza che il voto rispecchi la preparazione"). I dati possono non portare alla certezza, ma più dati si hanno e più ci si avvicina, riducendo l'incertezza.
Un esempio pratico è il modello **Elo** (nato per gli scacchi) da cui deriva quello usato da *Xbox* per capire come appaiare giocatori online in base alle skill. Il valore di bravura viene rappresentato come una distribuzione, in primis con una Gaussiana, con una certa media e una certa varianza/deviazione standard, quindi solo due numeri. Cambiare il modello significa solo cambiare quei due valori. Per confrontare due giocatori capisco la distribuzione a partire dai dati del giocatore che si hanno, diminuendo l'incertezza all'aumentare dei dati. Con il modello probabilistico poi, a partire dal risultato modificherò le distribuzioni di partenza, cambiando la percezione su essi. Nel tempo posso tenere aggiornato i modelli probabilistici che rappresentano una certa variabile e usarli per fare confronti (ad esempio confrontando due giocatori per poi fare l'appaiamento).
Con i modelli probabilistici si ha una capacità espressiva maggiore di quella di un modello logico, avendo le distribuzioni di probabilità e potendo anche usare varie soglie.
Un *modello generativo* parte dalle probabilità a priori e può "generare" possibili eventi, generando campioni verosimili con una certa distribuzione statistica.
Nella vita reale si osservano degli accadimenti e studiandoli si può risalire alla probabilità degli eventi, tramite l'approccio frequentista.

3.1 Incertezza

Si introduce quindi l'**incertezza**. Non sempre si ha a che fare con dati "certi" e precisi, che possono portare con più facilità ad una certa decisione, potendo giungere ad una decisione **ottimale** senza alcun dubbio su quale essa sia.
Con l'**incertezza** sui dati bisogna modificare l'idea di **soluzione ottimale**. Si arriva a dover capire quale sia la **soluzione ottima** in un contesto dove "non si sa cosa succederà", partendo da dati incerti.
Si ha che:

- un evento osservato può avere molte cause

- la verosimiglianza di un'ipotesi sulla causa cambia man mano che si raccolgono pezzi di evidenza
- usando modelli probabilistici di ragionamento possiamo calcolare quanto probabile è una certa ipotesi. Si ipotizza che le fonti di incertezza siano quantificabili

Vari esempi di vita in merito sulle slide.
Spesso si ha un approccio "frequentista", valutando la frequenza di un evento per capire la probabilità che tale evento accada, inferendo così una distribuzione di probabilità dalla frequenza con la quale si osservano i dati. Questo è piuù o meno come funziona il cervello umano ma bisogna fare la stessa cosa con un calcolatore e per questo ci verrà incontro il **teorema di Bayes**.
Si ha inoltre che un sistema che considera anche l'incertezza, che è presente in moltissime situazioni, dovrebbe funzionare meglio di uno che non lo fa ma ci serve in primis un modo per rappresentare l'incertezza stessa.
Più parametri ha il modello e più è difficile rappresentarlo.
Si ha il **Degree of Belief** che è una probabilità a priori sono ricavate da:

- osservazioni statistiche
- regole generali e conosciute
- combinazioni di sorgenti di evidenza

In ogni caso si hanno quindi evidenze empiriche.
Vediamo ora il rapporto tra i **modelli causali** e la **regola di Bayes**.
Ricordiamo che per Bayes si ha:

$$P(causa|effetto) = \frac{P(effetto|causa)P(causa)}{P(effetto)}$$

Con le reti causali vorremmo risalire dall'effetto alla causa ma normalmente si hanno più informazioni su $P(causa|effetto)$ che su $P(effetto|causa)$.
Conoscendo $P(effetto|causa)$ per ogni causa posso evitare di calcolare $P(effetto)$, infatti, dato $c = causa$ ed $e = effeto$:

$$P(c|e) = \frac{P(e|c)P(c)}{P(e)} = \frac{P(e|c)P(c)}{\sum_{\forall h \in causa} P(e|h)P(h)}$$

Vediamo un po' di notazione:

- con $< \top, \bot >$ indichiamo una distribuzione di probabilità

- α costante di normalizzazione per trascurare il denominatore di Bayes (lo sostituisce). È detto **fattore di normalizzazione**

Esempio 1. *Vediamo un esempio:*

$$P(meningite =< \top, \bot > | s = \top) = \alpha < P(s|m)P(m), P(s|\neg m)P(\neg m) >$$

SI assume che l'effetto deve essere scaturito a causa di una delle cause ipotizzate e non altre. A volte è più difficile calcolare $P(effeto|causa)$ per tutte le cause indipendentemente che calcolare direttamente $P(effetto)$.
Dato:

$$P(A|B) = \alpha P(B|A)P(A)$$

si ha che:

- $P(A)$ è la probabilità a priori
- $P(B|A)$ probabilità a posteriori
- $P(B|A)$ verosimiglianza

Se la probabilità a priori è nulla si assegna una probabilità ε (anche solo per un'osservazione) a tutti gli eventi che riteniamo possibili, anche se ancora non sono accaduti. Se un evento può realizzarsi deve avere una probabilità a priori, anche se molto piccola. Bisogna poi riscalare la probabilità di tutti per poter includere anche questi eventi rari. **Esempi su slide**.
Vediamo quindi come si **combinano le evidenze**. Qualora si abbiano più effetti il modello diventa più complesso. Per n effetti arei 2^n possibili combinazioni di evidenze da modellate. Si utilizza quindi la **catena di probabilità condizionali**, che, per esempio, per 4 eventi è:

$$P(A, B, C, D) = P(A|B, C, D)P(B|C, D)P(C|D)P(D)$$

ottenuta sfruttando la regola del prodotto:

$$P(A, B) = P(A|B)P(B)$$

La catena di probabilità condizionali è utile per rappresentare la probabilità congiunta in quanto permette una rappresentazione più compatta (potendo mettere anche insieme diverse fonti).

Definizione 9. *Considerato un insieme di eventi $E_1, \dots E_n$ e tutte le possibili combinazioni dei loro valori $\top$ e $\bot$. Supponiamo di conoscere tutti i valori $P(E_1, \dots, E_n)$. Supponiamo che un sottoinsieme di questi presenti un*

valore definito, ovvero $E_j = e = \top$ allora chiamo ***inferenza probabilistica*** *il processo di calcolo del valore:*

$$P(E_i = \top | E_i = e)$$

In generale l'inferenza probabilistica non è trattabile con questo metodo avendo una lista 2^n probabilità congiunte $P(E_1, \dots E_n)$ (lista che per di più spesso non abbiamo).
Si ragiona quindi spesso tramite ***metodi approssimati/qualitativi****, avendo magari centinaia di evidenze.*

Esempio su slide.

Per risolvere il problema viene anche incontro l'**indipendenza condizionata**.
Due eventi possono diventare indipendenti data la presenza di un altro evento, che è causa comune di entrambi. Si passa da una dipendenza causale diretta alla dipendenza dovuta ad un effetto causale indiretto (???). Se si conosce la causa i due eventi sono indipendenti se non la si conosce potrebbero essere dipendenti.

Definizione 10. *Definiamo la* ***regola di marginalizzazione*** *per due insiemi di variabili Y e Z come:*

$$P(Y) = \sum_{z \in Z} P(Y, z)$$

In alternativa uso le probabilità condizionate usando la ***regola del condizionamento****:*

$$P(Y) = \sum_{z \in Z} P(Y|z)P(z)$$

*Potrei anche usare l'****inferenza per enumerazione*** *dove semplicemente sommo i valori della tabella rispetto a ciò che mi interessa (se voglio $P(c = \top, m = \top)$ sommo tutti i valori con almeno uno dei due nella tabella).*

Su slide esempio di conto per tutti con anche conto per α.
Un principio generale di computazione è:

- specificare la variabile oggetto della "query"
- fissare lo stato delle variabili per le quali è disponibile l'evidenza
- calcolare la probabilità a posteriori sommando rispetto alle variabili sulle quali non è disponibile evidenza

Quindi indiciamo con x tale variabile oggetto di query. Data la realizzazione congiunta $e(evidenza)$ per un sottoinsieme E di variabili dette **variabili con evidenza** si indica con Y l'insieme restanti variabili. Y è detto insieme delle variabili senza evidenza. L'intero insieme delle variabili del problema è quindi:

$$\{X\} \cup E \cup Y$$

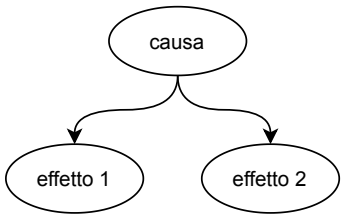
La distribuzione marginale a posteriori di X è ottenuto per marginalizzazione rispetto a Y:

$$P(X|E = e) = \alpha P(X, E = e) = \alpha \sum P(X, E = e, Y = y)$$

Posso quindi fare query per qualsiasi variabili avendo la tabella delle probabilità congiunte ma tale metodo non è efficiente.

3.1.1 Reti Bayesiane

Le relazioni di indipendenza condizionata può essere illustrata da un grafo, dove un nodo è collegato ad un altro con arco diretto sse il primo è causa dell'altro:



Esempio 2. *Vediamo un esempio:*

$$P(carie|mal\_di\_denti \wedge incastro) = \alpha P(|mal\_di\_denti \wedge incastro|carie)P(carie)$$

$$= \alpha P(|mal\_di\_denti|carie)P(incastro|carie)P(carie)$$

Ulteriori esempi su slide.
Le probabilità congiunte le posso rappresentare in una tabella.
Le **reti Bayesiane** sfruttano grafi diretti aciclici per rappresentare le assunzioni di indipendenza condizionale tra variabili in modo chiaro ed efficiente. Un arco diretto tra A e B rappresenta una relazione di causalità: A influenza B. Si ha quindi che "pattern" di ragionamenti sono un cammino tra un nodo e un altro.
Si passa da $O(2^n)$ a $O(n)$, per n numero di effetti.

Definizione 11. *Si ha che l'evento A è condizionalmente indipendente dall'evento B se, dato l'evento C:*

$$P(A|B,C) = P(A|C)$$

ovvero la conoscenza di B non porta a nessuna ulteriore variazione della probabilità di A rispetto a quella dell'avverarsi di C.
Dall'indipendenza di A e B dato C si ha che:

$$P(A,B|C) = P(A|C)P(B|C) = P(A,B|C)$$

Se C è un insieme vuoto ho, non avendo correlazione:

$$P(A,B) = P(A)P(B)$$

Le reti Bayesiane quindi analizzano le cause dirette e indirette permettendo di rappresentare in modo efficiente la distribuzione congiunta di probabilità, tramite dipendenza e indipendenza condizionale.
L'inferenza basata su enumerazione è in $O(d^n)$ sia in spazio che tempo con:

- d massima cardinalità del supporto (se binario $d = 2$)
- n numero di variabili

è questo non va bene.
Una distribuzione congiunta può essere rappresentata come produttoria di n valori di probabilità di **eventi indipendenti**, passando da un arrivando a $O(n)$, avendo:

$$P(C_1, \dots C_n) = P(C_1) \cdot \dots \cdot \dots P(C_n)$$

Rappresentando quindi, tramite l'indipendenza delle variabili, in modo compatto una distribuzione congiunta.
Nel mondo reale però non si ha indipendenza assoluta tra le variabili e spesso anche il gran numero di variabili rende difficile la specifica di una distribuzione congiunta.
Si usa quindi l'indipendenza condizionale, sfruttando le variabili condizionalmente indipendenti.
Per scrivere la definizione congiunta usiamo la **chain rule**. Dato $P(X,Y,Z)$ si ha che, avendo X e Y indipendenti:

$$P(X,Y,Z) = P(X|Y,Z)P(Y,Z) = P(Z|Y,Z)P(Y|Z)P(Z) = P(X|Z)P(Y|Z)P(Z)$$

riducendo quindi il numero di valori di probabilità necessari al conto.
Le asserzioni di indipendenza condizionate si basano sul dominio in analisi e

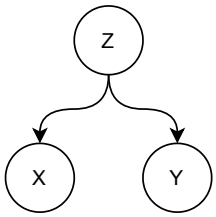
consente di limitare le complessità del modello. Il caso in cui tutte le variabili sono indipendenti ho la fattorizzazione delle probabilità delle singole variabili ed è un caso specifico di questo discorso.
Ricordando che per Bayes:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} = \alpha P(X|Y)P(Y)$$

calcolando la probabilità della causa data la conoscenza dello stato degli effetti. Quindi, avendo X e Y indipendenti:

$$P(X|Y,Z) = \alpha P(X,Y|Z)P(Z) = \alpha P(X|Z)P(Y|Z)P(Z)$$

Avendo graficamente $P(X,Y,Z) = \alpha P(X|Z)P(Y|Z)P(Z)$:



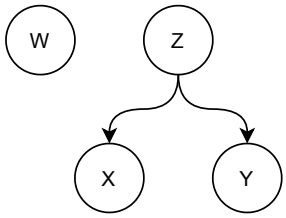
Possiamo quindi parlare meglio delle **reti Bayesiane**, spesso indicata con:

- **Bayesian Belief Network, BBN**
- **Probabilistic Network, PN**
- **Causal Network, CN**

Tali reti appartengono alla classe dei **modelli grafico-probabilistici**.

Definizione 12. *Una **rete Bayesiana** è un grafo cui nodi sono annotati da una informazione quantitativa, tramite tabelle di probabilità condizionata, e i cui archi definiscono dipendenza e indipendenza tra le variabili dei nodi. Si hanno solo archi orientati. Se X è causa diretta di Y ho un arco tra X e Y. X è detto genitore e Y figlio. Le variabili possono essere sia continue che discrete.*
La topologia della rete e le probabilità condizionate dei nodi dati genitori sono sufficienti a specificare (implicitamente) la distribuzione congiunta di tutte le variabili.
Uno o più nodi isolati segnalano indipendenza assoluta. Due figli di uno stesso

genitore segnalano che sono condizionalmente indipendenti. Nell'immagine X e Y sono condizionalmente indipendenti, W è indipendente dalle altre 3 variabili, Z è causa diretta di X e Y mentre tra queste ultime non esiste una relazione diretta di causalità:



Tale grafo non contiene cicli e quindi si parla di ***Directed Acyclic Graph (DAG)****, non essendo possibile che una variabile causi se stessa.*

Come detto la componente quantitativa è costruita da un insieme di tabella di probabilità condizionale. Ogni nodo ha associata quindi una *Conditioned Probability Table (CPT)* che traduce l'impatto dei genitori sulla variabile stessa.
Su slide esempio esteso.
Ricapitolando:

- ogni nodo ha CPT
- ogni riga della CPT somma ad uno (e se ho un solo valore non scrivo sia il vero che il falso visto che posso fare $1-$)
- la CPT di una variabile booleana con K variabili genitori contiene 2^K valori che possono essere specificati indipendentemente
- una variabile senza genitori ha una sola riga con i valori di probabilità a priori per ogni possibile valore che la variabile può assumere

Possiamo dire che si hanno due chiavi di lettura dal punto di vista della semantica, semanticamente equivalenti:

- la rete rappresenta una **distribuzione congiunta di probabilità**. Questa lettura è utile per progettare e implementare procedure di inferenza. Per questa lettura dice che ogni rete costituisce una descrizione completa del dominio che rappresenta e pertanto ogni elemento della distribuzione di probabilità congiunta può essere calcolato a partire dall'informazione contenuta nella rete. Un generico elemento della distribuzione di probabilità congiunta

è associato ad una realizzazione congiunta delle variabili (nodi) presenti nella rete:

$$P(X_1 = x_1 \wedge \cdots \wedge X_n = x_n) = P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

che è detta **formula di fattorizzazione**.
Avendo che con le maiuscole abbiamo le variabili (anche per $Parents$) e con le minuscole le realizzazioni (anche per $parents$). Si ha quindi che ogni elemento della distribuzione congiunta è rappresentato come prodotto di opportune componenti delle CPT che costituiscono quindi una rappresentazione decomposta della distribuzione di probabilità congiunta. Per questa rappresentazione posso usare le reti per rispondere a qualsiasi query relative al dominio che descrive tramite marginalizzazioni.
Esempio su slide.

- la rete codifica un **insieme di relazioni di indipendenza condizionale**. Questa lettura è utile per costruire un modello di rete Bayesiana.

Sfrutto la formula di fattorizzazione per determinate la componente topologica della rete. Si ricorda che per la **cchin rule**:

$$P(x_1, \ldots, x_n) = P(x_n \mid x_{n-1}, \ldots, x_1) \cdot P(x_{n-1} \mid x_{n-2}, \ldots, x_1) \cdot \ldots \cdot P(x_2 \mid x_1) \cdot P(x_1)$$

$$= \prod_{i=1}^{n} P(x_i \mid x_{n-1}, \ldots, x_1)$$

Noto che tale formula è confrontabile con la formula di fattorizzazione e ho che, a patto che $Parents(X_1) \subseteq \{x_{i-1}, \ldots, x_1\}$:

$$\mathbf{P}(x_i \mid x_{i-1}, \ldots, x_1) = \mathbf{P}(x_i \mid \text{ parents } (X_i))$$

Quindi una Rete Bayesiana rappresenta correttamente un dominio solo a condizione che ogni nodo risulti condizionalmente indipendente dai suoi predecessori, per un dato ordinamento, dati i suoi genitori. Pertanto, per costruire una Rete Bayesiana che abbia la corretta struttura del dominio da modellare è necessario scegliere, per ogni nodo, i nodi genitore in modo che tale proprietà risulti verificata. Quindi i genitori di X_i devono essere scelti da $\{X_1, \ldots, X_{i-1}\}$.
Si ha quindi la seguente procedura di costruzione incrementale della componente topologica;

1. Si seleziona un insieme di variabili $\{X_1, \ldots, X_n\}$ per descrivere il modello

2. scelgo un ordinamento per le variabili $\{X_{(1)}, \ldots, X_{(n)}\}$. Con un ordinamento sbagliato arrivo a definire reti sbagliate o reti più complesse del necessario, magari con informazioni ridondanti

3. inizializzo i nodi aggiunti alla rate partendo da $i = 1$

4. si seleziona la variabile $X_{(i)}$ alla rete, si pone $Parents(X_{(i)})$ uguale all'insieme minimale di nodi, attualmente appartenenti alla rete $X_{(1)}, \ldots, X_{(i-1)}$, che soddisfa la proprietà di indipendenza condizionale:
$$\mathbf{P}\left(X_{(i)} \mid X_{(i-1)}, \ldots, X\right) = \mathbf{P}\left(X_{(i)} \mid \text{ Parents } \left(X_{(i)}\right)\right.$$
per poi calcolare la CPT di $X_{(i)}$

5. tengo conto del nodo aggiunto nel conto totale dei nodi e faccio i++. Se ho finito le variabili termino altrimenti torno a 4)

Una rete è più compatta, solitamente, dell'intera distribuzione di probabilità congiunta e tale compattezza è un esempio della proprietà dei **sistemi strutturati localmente o sparsi**, dove ogni sotto-componente interagisce solo con un numero limitato di altre componenti, indipendentemente dal numero totale di componenti del sistema. La strutturazione locale normalmente ha crescita lineare e non esponenziale (mediamente con una rete Bayesiana è ragionevole pensare che ogni variabile sia influenzata da al massimo k, con k costante, variabili).
Nel caso in cui si consideri una Rete Bayesiana costituita da n variabili (nodi) booleane. Si ha che la quantità di informazione per specificare una CPT è limitata superiormente da 2^k per cui la rete completa richiede $n \cdot 2^k$ cifre contro i 2^n dell'intera distribuzione di probabilità congiunta.
Si hanno quindi vari tipi di rete, relativi a situazioni diametralmente opposte:

- **rete completamente connessa (*fully connected network*)**, dove ogni variabile può essere potenzialmente influenzata da tutte le altre e quindi specificare una CPT richiede le stesse informazioni della distribuzione congiunta

- **rete con relazioni di causalità tenui** dove se aggiungo delle relazioni, degli archi aumento la complessità della rete (???). Si valuta il guagno in accuratezza contro quelle in complessità

Anche in un dominio strutturato localmente la costruzione di una Rete Bayesiana non è semplice, volendo un numero limitato di influenze per ogni variabile e che la topologia della rete rifletta le relazioni di influenza diretta. La procedura di costruzione di una Rete Bayesiana funziona in modo tale che quando si giunge ad aggiungere un nodo, i nodi candidati ad essere suoi genitori, ovvero i nodi che lo influenzano direttamente o indirettamente, siano già stati aggiunti alla corrente struttura della rete. Il corretto ordine comporta che si devono prima stabilire le cause radice per poi aggiungere quelle variabili che da loro vengono influenzate,

procedendo fino alle foglie che non sono causa di nulla.
lato **semantica topologica** abbiamo due specificazioni equivalenti:

- un nodo è condizionalmente indipendente dai suoi non-discendenti dati suoi genitori
- un nodo è condizionalmente indipendente da tutti i nodi restanti della rete, data la conoscenza dello stato dei suoi genitori, dei suoi figli e dei genitori dei suoi figli. Tale insieme di nodi è detto **Markov Blanket**

Vediamo come capire se in una rete Bayesiana due variabili sono condizionalmente indipendenti.

Definizione 13. *Definiamo che X e X sono **d-separati** da un insieme E di variabili con evidenza (ovvero osservazioni) sse ogni cammino non orientato da X a Z è **bloccato** ovvero sse:*

- *lungo il cammino si ha una variabile $V \in E$ (quindi di cui conosco il valore) e gli archi che collegano V al cammino sono solo **tail to tail** (ovvero da V escono i due archi: $\rightarrow V \leftarrow$) e quindi il cammino è bloccato da V*
- *lungo il cammino si ha una variabile $V \in E$ (quindi di cui conosco il valore) e gli archi che collegano V al cammino sono solo **tail to head** (ovvero un arco entra in V ed uno esce: $\rightarrow V \rightarrow$) e quindi il cammino è bloccato da V*
- *lungo il cammino si ha una variabile $V \notin E$ (quindi di cui NON conosco il valore) tale che nessuno dei suoi discenti appartiene all'insieme E e gli archi che collegano V al cammino sono solo **head to head***

Esempi su slide.

Teorema 4 (teorema di Verma e Pearl)**.** *Se in una rete Bayesiana un insieme E di variabili con evidenza d-separa X e Y allora X e Y sono indipendenti.*

La d-separazione si calcola in tempo lineare avendo quindi un algoritmo efficiente per inferire automaticamente se apprendere il valore di una variabile può fornire informazioni aggiuntivi su altre variabili date le informazioni a disposizione quindi può essere utile per capire se apprendere il valore di una delle due variabili può aiutarci ad avere informazione aggiuntiva rispetto all'altra.
Si hanno algoritmi di propagazione dell'informazione che portano a calcolare la distribuzione di probabilità, incondizionata o marginale, su ogni nodo. Tale distribuzione è detta anche **belief function** del nodo.
L'algoritmo di propagazione aggiorna le belief di ogni nodo della rete ogni volta che si ha una nuova evidenza (???).

Esempi su slide. Su elearning file sulla *Moralizzazione*, argomento extra.
Anche avendo massimo k genitori per ogni variabile si ha che servono $O(2^k)$ parametri per ogni CPT.
Si ha una distribuzione canonica per rappresentare pattern standard per avere un numero limitato di parametri per compilare la CPT. Uno di questi pattern è detto **modi deterministici**. Un nodo deterministico ha che il suo valore è determinato unicamente da quello dei genitori, senza incertezza, avendo quindi una relazione ad esempio logica (*and, or,* etc...) o numerica (*max, min,* etc...). Ovviamente non tutta la rete è fatta da questi nodi altrimenti non userei una rete Bayesiana ma in una rete complessa si possono avere tali pattern. Tramite tali relazioni calcolo direttamente i valori delle CPT.
Un altro pattern, in presenza di incertezza, è quello **noisy logical relationship**, con generalizzazioni degli operatori logici. Un esempio è il **noisy-OR**, che introduce incertezza circa la capacità di causare $\top$ nella variabile figlio da parte del nodo genitore. Si ha che la relazione di causalità tra genitore e figlio potrebbe essere inibita. Con noisy-OR si hanno due ipotesi:

- tutte le possibili cause sono note (eventualmente si possono aggiungere *leak node*, un nodo fittizio per rendere esaustive la cause)
- l'inibizione di un genitore è indipendente da quella di altri genitori per il nodo considerato

L'inibizione è legata ad una certa probabilità, dette *probabilità di inibizione*, ovvero **noise parameters**.
La probabilità di un evento viene ottenuta tramite il prodotto delle probabilità di inibizione di ogni nodo genitore. Quindi per certe combinazioni di or non ho il valore preciso ma una certa probabilità. Se esattamente un genitore è $\top$ allora il figlio è $\top$ con probabilità pari al noise parameter di tale genitore. Se nessun genitore è nello stato $\top$ allora certamente il figlio è nello stato $\bot$. Negli altri casi appunto moltiplico i noisy parameter dei genitori con valore $\top$. Ho quindi limitato il numero di parametri. Quindi se una variabile dipende da k genitori di norma ho la noisy-OR con $O(k)$ parametri e non $O(2^k)$, per descrivere l'intera distribuzione di probabilità congiunta.
Esempi su slide.
Potrei avere **reti con nodi continui** e potendo assumere un numero infinito di valori si rende impossibile specificare esplicitamente i valori della distribuzione condizionale e di quella a priori. Si procede quindi **discetizzando** i valori in un numero finito di intervalli, perdendo comunque di informazione, soprattutto con grandi CPT. Una alternativa sono le **funzioni di densità di probabilità**, trascritte un numero finito e, solitamente, contenuto di parametri.
Un esempio di tale funzione è la **gaussiana univariata**:

$$N(\mu, \sigma^2)$$

che ha due parametri: *media* e *varianza*.
Una rete con nodi discreti e continui insieme è detta **rete Bayesiana ibrida** e per la sua definizione servono due tipi di distribuzione:

- la distribuzione condizionale di una variabile continua dati i genitori discreti o continui
- la distribuzione condizionale di una variabile discreta dati i genitori continui

Usando quindi questa *gaussiana lineare* (comoda perché facendo operazioni tra gaussiane si ottiene comunque una gaussiana) perché il nodo figlio è distribuito secondo una distribuzione gaussiana la cui media μ varia linearmente con il valore del genitore mentre la deviazione standard σ è fissata. In pratica si fa una sorta di regressione lineare con rumore gaussiano. Si ha che la gaussiana lineare ha 2 proprietà:

1. una rete che contiene solo nodi continui con distribuzione Gaussiana Lineare ha una distribuzione di probabilità congiunta **Gaussiana multivariata**
2. una rete che contiene nodi continui con distribuzione Gaussiana Lineare e nodi discreti, nessuno dei quali sia figlio di nodi continui, definisce una distribuzione Gaussiana Condizionale per ogni assegnamento di valori per le variabili discrete; la distribuzione sulle variabili continue è **Gaussiana multivariata**

Avendo una cosa del tipo $P(c|r, s = \top)$ si avrebbe:

$$N(a_\top \cdot r + b_\top, \sigma_\top^2)(c) = \frac{1}{\sqrt{2\pi} \cdot \sigma_\top} e^{-\frac{1}{2}\left(\frac{c-(a_\top \cdot r + b_\top)}{\sigma_\top}\right)^2}$$

dovendo poi fare un discorso analogo per $\perp$. Si ha che a e b sono il legame con la regressione lineare.
Esempio su slide.
Anche usando modelli semplici si possono ottenere distribuzioni flessibili e interessanti.
Passiamo ora alla **distribuzione associata a variabili discrete con genitori continui**, tramite una **funzione a soglia soft**.
Un modo di ottenere soglie soft è usare l'integrale della normale standard:

$$\phi(x) = \int_{-\infty}^{x} N(0,1)(x)\,\mathrm{d}x$$

usando la distribuzione **probit**:

$$P(X|Y = c)\phi(\frac{-c+\mu}{\sigma})$$

il che significa che la soglia di costo si verifica intorno al valore , lo spessore della regione soglia è proporzionale al valore σ, e la probabilità di acquistare diminuisce all'aumentare del costo.
In alternativa si ha la funzione **probit**, che utilizza la **funzione sigmoid** per ottenere una soglia soft:

$$P(X|Y = c)\phi(\frac{1}{1 + e^{-2 \cdot \frac{-c+\mu}{\sigma}}})$$

3.1.2 Inferenza nelle Reti Bayesiana

Lo scopo di un modello probabilistico è quello di ottenere la distribuzione a posteriori, per un certo insieme di variabili query, in presenza di un evento, ovvero un assegnamento congiunto di valori ad un insieme di variabili con evidenza. Sono dati:

- X variabile query
- E insieme delle evidenze E_i
- e evento specifico
- Y insieme variabili non evidenziate Y_i

L'insieme completo delle variabili è:

$$\mathbf{X} = \{X\} \cup E \cup Y$$

e si hanno query del tipo:

$$P(X|E = e)$$

Dato un modello di rete Bayesiana si hanno 4 tipologie di inerenza;

- **diagnostica**, avendo l'effetto si vuole capire la causa
- **causale**, avendo la causa voglio stimare la probabilità degli effetti
- **intercausale**, avendo una causa, capire la probabilità un'altra causa di un effetto comune
- **mista**, avendo sia diagnostica che causale o sia diagnostica che intercausale

Avere osservato che uno degli eventi fa divenire meno probabile che un altro si sia verificato è il meccanismo di **explaining away**.
Con una rete Bayesiana quindi posso cercare:

- probabilità condizionata $P(X|e)$

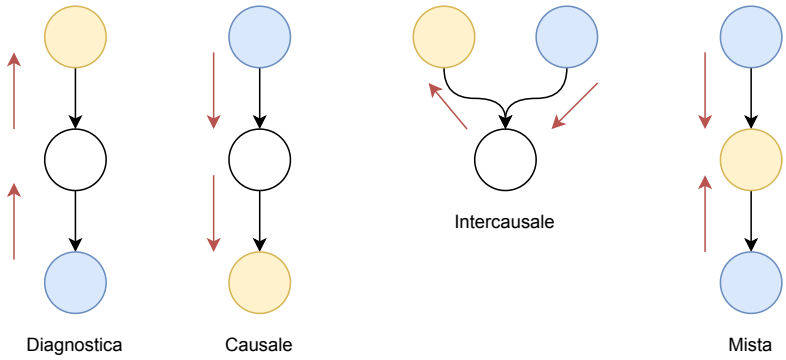


Figura 3.1: Tipi di inferenza, in blu l'evidenza e in giallo la variabile X

- quale valore massimizza $P(X|e)$, cercando la massima probabilità a posteriori
- quale è la distribuzione di probabilità di X dato e (che è il caso generale)

Ogni distribuzione condizionale può essere ottenuta tramite un procedimento di somma di opportuni elementi dell'intera distribuzione di probabilità congiunta (marginalizzazione). Si ha quindi che:

$$P(X|E=e) = \alpha \cdot P(X, E=e) = \alpha \cdot \sum_{y} P(X, E=e, Y=y)$$

e con la regola di fattorizzazione delle reti Bayesiane:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

e quindi $P(X, E=e, Y=y)$ nella distribuzione congiunta, possono essere scritti sotto forma di prodotti di probabilità condizionali della rete. In definitiva concludiamo che:

Ad una query è possibile rispondere utilizzando una Rete Bayesiana tramite la computazione di somme di prodotti di probabilità condizionali della rete

Vedremo sia algoritmi esatti che approssimati, tramite campionamento, per l'inferenza.

3.2 Inferenza esatta

Per fare inferenza esatta sfruttiamo l'equazione:

$$P(X|E=e) = \alpha \cdot P(X, E=e) = \alpha \cdot \sum_{y} P(X, E=e, Y=y)$$

sfruttando la fattorizzazione, che fornisce un'espressione in termini di CPT per effettuare il calcolo.
Esempio su slide.
Nel caso peggiore, dovendo marginalizzare tutte le variabili, ho un algoritmo, per una rete di n variabili booleane, di complessità:

$$O(n \cdot 2^n)$$

ma a seconda dei casi posso effettuare delle ottimizzazioni, estraendo elementi costanti dalle sommatorie generate usando la chain rule (**guardare sempre esempio su slide**).
Per ogni sommatoria che si crea poi eseguo l'**algoritmo di enumerazione**, vedendo l'algoritmo come una sorta di struttura ad albero. Si richiede quindi di iterare su tutte le possibili realizzazioni congiunte delle variabili coinvolte.
Con queste ottimizzazioni si arriva a:

$$O(2^n)$$

Un miglioramento comunque non sufficiente.
Studiando l'albero di computazione si notano ripetizioni di computazione quando si effettua l'algoritmo di enumerazione, che consiste nella valutazione ricorsiva di tale albero. Si memorizzano quindi le operazioni già fatte per non doverle ripetere, ad esempio tramite **l'algoritmo di eliminazione variabili**. Questo algoritmo valuta da destra a sinistra, ovvero bottom-up, le espressioni e memorizza i risultati intermedi mentre la marginalizzazione viene effettuata solo per le porzioni di espressione che dipendono dalla variabile stessa.
Tale algoritmo si basa sul fatto che ogni variabile che non sia predecessore di una variabile in studio o di una variabile con evidenza è irrilevante nello studio della query. Si rimuovono quindi dall'albero le variabili inutili, ovvero quelle oggetto della query e quelle con evidenza.
L'ordine di eliminazione influisce sulla complessità temporale complessiva. Con reti molto complesse magari non binarie, il problema resta di elevata complessità e non ci si può aspettare di poter fare inferenza esatta su tali reti.
Con le reti Bayesiane si nota che è possibile fare, a seconda della query, calcolo parallelo. Si ragiona comunque solitamente per campionamenti, generando configurazioni possibili della rete tramite le distribuzioni della CPT e calcolano la frequenza in cui una certa variabile assume un certo valore date le evidenze. Per farlo bisogna generare numeri pseudocasuali.

3.3 Numeri Pseudocasuali

Innanzitutto su un numero singolo non possiamo dire nulla in merito alla sua casualità. Se uno dice 3 non posso inferire nulla sulla sua casualità.
Una lista di numeri può apparire casuale ma potrebbe non esserlo.

Definizione 14. *una* ***sequenza di numeri pseudo casuali*** *è una sequenza di realizzazioni di variabili aleatorie indipendenti e identicamente distribuite.*
Tali numeri sembrano impredicibili e senza regolarità. "Sembrano" perché comunque dietro si ha un meccanismo di calcolo.

Il lancio dei dadi è un metodo manuale valido, completamente causale al più di trucchi, ma richiede troppo tempo. Esistono anche device specifici per farlo ma si hanno altri problemi, si preferisce quindi un calcolatore che genera le stesse sequenze, in modo pseudocasuale.
Serve un **seme di generazione**, avendo che con un seme genero sempre la stessa sequenza e questo è importante per poter valutare due sistemi in base agli stessi scenari, rigenerando sempre la stessa sequenza. Questa è la **proprietà di riproducibilità**.
I numeri causali generati devono avere una serie di proprietà statistiche, devono sembrare impredicibili ma:

- la distribuzione deve essere, o perlomeno simulare, una distribuzione uniforme, verificabile all'aumentare dei numeri generati tramite il calcolo della media (che deve essere a metà del range nel quale si hanno numeri generati, se range $[0, 9]$ voglio media 4.5)
- devo avere indipendenza statistica
- devo avere riproducibilità di valori
- non devo avere ripetitività su un prefissato periodo abbastanza lungo, se avessi periodi troppo brevi potrei fare predizione

Il seme, se non specificato, viene scelto via hardware, tipo il clock attuale della cpu etc...
La routine che genera numeri pseudocasuali di default genera una distribuzione uniforme su $[0, 1]$ deve:

- essere veloce
- avere periodi di ripetizioni lunghe
- non presentare lunghi gap tra la generazione di due numeri, nella discretizzazione di variabili continue, avendo uniformità
- essere replicabile
- generare sequenze con proprietà statistiche simili a quelle reali

Uno dei primi algoritmi è stato quello di Von Neumann, detto **middle square**, che genera un numero con 10 cifre. Si parte da un numero, il seed, si eleva al quadrato e si prendono le 10 cifre centrali, che saranno il prossimo numero. In realtà è molto pseudocasuale avendo che ogni numero dipende dal precedente ma

sembra causale e si hanno casi di numeri che comportano facilmente la riconoscibilità di tale pattern o che addirittura generano un loop, tornando al seme dopo poche generazioni.
Nel tempo si sono quindi di studiati vari parametri iniziali dell'algoritmo.
Un primo metodo, detto **moltiplicative congruential method (*MCM*)**, prevede:

$$x_n = a \cdot x_{n-1}, \;\; 0 \leq x_n \leq m$$

con a e m interi scelti e un x_0 seed. La scelta di m deve essere tale per avere cicli molto ampi e quindi m è solitamente molto grande, nonché primo (cosa capita empiristicamente). Sono state studiate varie combinazioni di m e a, come $m = 2^{31} - 1$ e $a = 7^5$ o $m = 2^{35} - 31$ e $a = 5^5$. Siu ha che m deve portare ad avere pochi gap.
Si ha poi il **linear conrguential method (*LCM*)**, che prevede:

$$x_n = (a \cdot x_{n-1} + c) \bmod m$$

con anche c scelto opportunamente. Si ha che:

- $a, c \geq 0$
- $m > x_0, c, a$

Però ci servono variabili aleatorie uniformi in $[0, 1]$ quindi devo fare, per LCM:

$$u_n = \frac{x_n}{m}$$

Con LCM si ha ciclo pari a m e si generano numeri discretizzati e quindi serve un m molto grande (ma anche qui si hanno scelte di parametri "sfortunate", che comportano cicli brevi).
Un ciclo normalmente deve tendere al numero intero massimo rappresentabile, ma dipende dai casi.
Aumentare m riduce periodicità e generazione di numeri razionali. Solitamente $m \geq 10^9$, per ottenere un sottoinsieme denso in $[0, 1]$. Solitamente m è quindi il massimo intero rappresentabile.
Il **generatore di Learmouth-Lewis** prevede:

- $c = 0$
- $a = 75$
- $m = 2^{31} - 1$

Che è uno dei più usati ma si hanno molti altri generatori.
A partire dai numeri tra 0 e 1 posso passare a distribuzioni arbitrarie.

3.3.1 Generazione di Distribuzione Generica

Partiamo ora da una sequenza di numeri pseudocasuali $U(0,1)$. Si hanno varie tecniche per la generazione di una distribuzione generica:

- **tecnica di trasformazione inversa**
- **metodo di accettazione rifiuto**
- **metodo di composizione**

In generale si vuole utilizzare poche volta la routine di generazione di numeri random.
Ricordiamo due concetti:

- **funzione di densità** che prevede:

  $$P(x = x_1) = 0$$

  e:

  $$P(x_1 - \varepsilon \leq X \leq x_1 + \varepsilon) = \int_{x_1-\varepsilon}^{x_1+\varepsilon} f_X(x)\,\mathrm{d}x$$

- **distribuzione cumulativa**, che prevede;

  $$F_X(x) = p(X \leq x) = \int_{-\infty}^{x} f_X(y)\,\mathrm{d}y$$

Trasformazione Inversa

Vogliamo ora generare una variabile aleatoria X con funzione di densità di probabilità:

$$f_X(x)$$

Si hanno quindi vari step:

1. si calcola la funzione di distribuzione di probabilità o funzione cumulativa di probabilità:

   $$F_X(x) = \int_{-\infty}^{X} f_X(\tau)\,\mathrm{d}\tau$$

   che, qualora sia possibile calcolarla in forma chiusa, è continua, monotona, crescente e compresa tra 0 e 1, avendo $F_X(x) = p(X \leq x)$

2. si pone:

   $$u = F_X(x)$$

   con u numero random ($u - U(0,1)$

3. si risolve:
$$X = F_X^{-1}(u)$$
ottenendo che la variabile aleatoria X è distribuita secondo $f_X(x)(X \sim f_X(x))$

Vediamo quindi qualche caso particolare.
Supponiamo di voler costruire una successione di numeri pseudocasuali come osservazioni dalla **distribuzione esponenziale** ovvero con funzione di distribuzione:

$$F(x) = 1 - e^{\lambda \cdot x}$$

Ho quindi:
$$U = F(x) = 1 - e^{\lambda \cdot x} \implies 1 - U = e^{\lambda \cdot x}$$

e quindi:
$$\ln(1 - U) = \ln\left(e^{\lambda \cdot x}\right) \implies \ln(1 - U) = -\lambda \cdot x$$

e quindi:
$$x = -\frac{\ln(1 - U)}{\lambda}$$

e quindi:
$$x = F^{-1}(U) = -\frac{\ln(1 - U)}{\lambda}$$

e se U è una variabile aleatoria con distribuzione uniforme su $[0, 1)$ ho che x è una variabile aleatoria con distribuzione esponenziale con media $\frac{1}{\lambda}$. Da una successione di numeri pseudocasuali uniformi ottengo una successione di numeri pseudocasuali con distribuzione esponenziale.
Inoltre se U è in [0,1) lo è anche $1 - U$ e quindi sostituisco $(1 - U)$ con U nella formula, anche se questo cambiamento potrebbe indurre un cambiamento nella correlazione delle variabili X generate.
Quindi la routine di generazione di X con funzione di densità di probabilità esponenziale chiama una sola volta per ogni X la routine di generazione di numeri casuali R. Si ha, per la routine di X, quindi lo stesso ciclo di tale routine R e gap crescenti per X crescente. Se la routine è replicabile lo è anche quella per X. Se la routine di generazione è ideale lo è anche quella di X.
Posso estendere il metodo a distribuzioni discrete, con X **variabile aleatoria discreta**.
Suppongo X con valori $x_1, x_2, \ldots$ con:

$$x_1 < x_2 < \ldots$$

Data U variabile uniformemente distribuita in $[0, 1)$ si ha che:

$$X(U) = \max\{x_i | \, U \in [F(x_i - 1) - F(x_i)]\}$$

Si ha quindi, avendo $P(X = i) = p_i$:

- **funzione di densità**:

$$\sum_{i=\infty}^{\infty} P(X = i) = 1$$

- **Distribuzione cumulativa**:

$$F_X(x) = P(X \leq x) = \sum_{i=-\infty}^{x} P(X = i)$$

In generale si ha che:

$$P(X = x_j) = p_j, \;\; j = 0, 1, \ldots, n$$

con:

$$\sum_{j=0}^{n} p_j = 1$$

e ho:

- x_0 se $u \leq p_0$
- x_1 se $p_0 \leq\leq p_0 + p_1$
- $\cdots$
- x_j se $\sum_{i=0}^{j-1} p_i \leq u \leq \sum_{i=0}^{j}$
- $\cdots$

In base a questo si possono avere procedure più efficienti di altre (???), rendendo la ricerca di appartenenza dell'intervallo il più efficiente possibile.
Ad esempio, volendo generare una distribuzione discreta uniforme con:

- $X \varepsilon \{0, 1, \ldots, n\}$ (???)
- $p_0 = p_1 = \cdots p_n = \frac{1}{n+1}$

Ottenendo:

$$X = \lfloor (n + 1) \cdot u \rfloor$$

3.3.2 Metodo Acceptance-Rejection

Il metodo precedente si basava sul calcolo di F^{-1} ma non sempre questo è possibile in modo efficiente, a seconda della funzione. Si sono stati quindi creati altri metodi, più generali, come il **metodo Acceptance-Rejection**, detto anche **metodo del rigetto**.
Suppongo una distribuzione definita su $[a, b]$. Suppongo di conoscere la densità di

probabilità della variabile aleatoria X tale che $f_X(x)$ è su $[a, b]$ e la sua immagine è definita sul codominio $[0, c]$. In pratica tale funzione è in un rettangolo $[a, b] \times [0, c]$. Si genera quindi un numero in questo rettangolo e se cade fuori dalla funzione lo rifiuto, altrimenti lo accetto.
Si generano due sequenze pseudocasuali uniformi tra $[0, 1]$ e le chiamo U_1 e U_2. Si generano quindi altre due sequenze uniformi ytramite:

$$\begin{cases} X = a + (b - a) \cdot U_1 \\ Y = c \cdot U_2 \end{cases}$$

rimappando $[0, 1]$ in $[a, b]$ e in $[0, c]$. Quindi ad ogni (U_1, U_2) si ha (X, Y) che è nel rettangolo. Se quest'ultima cade nell'area della funzione viene accettata e in caso contrario rifiutata, provando una nuova coppia. La sequenza di valori ottenuta segue quindi la distribuzione data dalla funzione.
Il metodo è più efficiente quando l'area della funzione copre quasi interamente il rettangolo, avendo lo scarto di poche coppie. Si hanno quindi i seguenti step per generare una variabile aleatoria X, con funzione di densità di probabilità $f_X(x)$ su $[a, b]$:

- si genera una variabile R distribuita in modo uniforme su $[a, b]$, ovvero $U(a, b)$
- si accetta tale valore con probabilità pari a:

  $$\frac{F_X(R)}{\max F_X(x)}$$

  in quanto il rettangolo è "aderente" alla funzione e la c tende a coincidere con il massimo della funzione. Tale numero è rifiutato con probabilità:

  $$1 - \frac{F_X(R)}{\max F_X(x)}$$

Dove la funzione è "bassa" si scartano più valori.
La routine di generazione di tale variabile aleatoria X quindi:

- chiama due volte la routine che genera un numero random, una per R e una per capire se accettare e rifiutare
- ciclo e gap dipendono dal prodotto tra questi due numeri casuali ed p replicabile se lo è la routine di generazione del numero casuale
- è da usare solo se non si hanno altri metodi

3.4 Inferenza Approssimata

Il poter generare numeri causali serve al campionamento. Se le reti hanno una certa struttura posso usare l'inferenza esatta ma in generale è un problema NP-hard. SI usa quindi solo su reti singolarmente connesse, dove si ha al più un cammino orientato tra coppie di nodi. Tali reti sono dette **polytree** e permettono complessità spaziale e temporale lineare nella dimensione della rete (ovvero nel numero di elementi della CPT). Addirittura se il numero di genitori di ogni nodo è limitato da una costante si ha tempo lineare nel numero di nodi.
Per rispondere a query univariate posso usare la **eliminazione delle variabili**.
La computazione della probabilità a posteriori per tutte le variabili può essere meno efficiente, ad esempio in un polytree si arriva a $O(n^2)$. Per restare in $O(n)$ si usano algoritmi di clustering, dove si uniscono nodi per ottenere un polytree, dove si può poi usare un metodo efficiente. Purtroppo anche gli algoritmi di clustering presentano inferenze NP-hard, per la costruzione delle CPT dei cluster.
Per reti multiplamente connesse si usano quindi algoritmi di inferenza approssimata, tramite algoritmi **Monte Carlo**, la cui accuratezza dipende dal numero di campioni generati.
Vedremo due metodi:

- **direct sampling**
- **Markov chain sampling**

3.4.1 Direct Sampling

In primis bisogna generare campioni secondo una distribuzione di probabilità, tramite generazione di numeri pseudocasuali, che è l'idea di base di ogni algoritmo di campionamento.
Partiamo dl caso semplice, dove si ha una rete Bayesiana senza variabili con evidenza. Si campiona quindi la rete, seguendo l'ordine topologico partendo dalla radice e campionando le variabili. La scelta della distribuzione per il campionamento è quella che risulta dal condizionamento sui valori del nodo genitore. Tale algoritmo è detto **campiona-Priori**, che ha in input la rete e genera/campiona le variabili della rete. Tali campioni simulano la distribuzione congiunta della rete. Genero in pratica possibili configurazioni della rete.
Esempio su slide.
L'algoritmo genera quindi campioni della distribuzione di probabilità congiunta a priori. Chiamo:

$$S_{CP}(x_1, \ldots, x_n)$$

come la probabilità che uno specifico evento sia generato tramite l'algoritmo Campiona-Priori. Sappiamo poi che:

$$S_{CP}(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(x_i))$$

in quanto ogni campione dipende solo dai genitori. Questo ricordiamo rappresenta la probabilità dell'evento considerato secondo la distribuzione di probabilità con giunta della rete, la fattorizzazione. Si ha quindi che:

$$S_{CP}(x_1, \ldots, x_n) = P(x_1, \ldots, x_n)$$

rendendo semplice rispondere alla query.
Le risposte, dati N campioni, vengono date tramite conteggio dei campioni. SI ha che:

$$N_{cp}(x_1, \ldots, x_n)$$

è la frequenza per l'evento $(x_1, \ldots x_n)$ negli N campioni.
Ogni campione da un contributo paritario e ci si aspetta che la frequenza convenga al valore atteso in accordo con la distribuzione da cui si estraggono gli N campioni:

$$\lim_{N\to\infty} \frac{N_{cp}(x_1, \ldots, x_n)}{N} = S_{CP}(x_1, \ldots, x_n) = P(x_1, \ldots, x_n)$$

Campionando in modo indipendente generiamo campioni che rappresentano la distribuzione della rete, quindi all'aumentare di N ci si aspetta che un evento compaia con la giusta probabilità, anche se comunque è un "tende", per cui si parla di inferenza approssimata.
Con $\approx$ si indica la probabilità stimata, che diviene esatta per $N = \infty$. Tale stima è detta **consistente**, avendo, per $m \leq n$:

$$P(x_1, \ldots, x_m) = \frac{N_{PS}(x_1, \ldots, x_m)}{N}$$

La probabilità di un evento parzialmente specificato viene stimata come la frazione dei casi compatibili con l'evento parzialmente specificato sul numero di tutti i casi generati tramite campionamento.
Passiamo ora all'avere evidenze nella rete, volendo computare una probabilità condizionata del tipo:

$$P(X|E = e)$$

Si usa quindi l'algoritmo di **campionamento con rigetto**, che usa quello **campionapriori** per generare campioni sulla distribuzione della rete per poi rifiutare i campioni non conformi con l'evidenza e. Si ottiene la distribuzione relativa solo alla configurazione della rete dove si ha una certa evidenza.
La stima a posteriori viene ottenuta contando la frequenza per $X = x$ sull'insieme dei campioni rigettati.
Sia:

$$\hat{P}(X|E = e)$$

la stima della distribuzione a posteriori fatta con il campionamento con rigetto.
Si ha quindi che:

$$\hat{P}(X|E = e) = \alpha N_{CP}(X, E = e) = \frac{N_{CP}(X, E = e)}{N_{CP}(E = e)}$$

Sapendo che:

$$P(x_1, \ldots, x_n) \approx \frac{N_{CP}(x_1, \ldots, x_n)}{N}$$

Ho che:

$$\hat{P}(X|E = e) \approx \frac{P(X, E = e)}{P(E = e)} = P(X|E = e)$$

Avendo che l'algoritmo produce stime consistenti della distribuzione vera, tendendo ad essa all'aumentare del numero di campioni non rigettati. La deviazione standard dell'errore è proporzionale a, con N_v numero di campioni non rigettati:

$$\frac{1}{\sqrt{N_v}}$$

Lo svantaggio dell'algoritmo è che butta via troppi campioni, con un tasso di rigetto che cresce esponenzialmente con il numero di variabili con evidenza. Questo dettaglio rende inutilizzabile l'algoritmo con reti Bayesiane reali, molto grosse.
Un'alternativa è l'algoritmo di **likelihood weighting** che evita di generare campioni inutili che verrebbero poi scartati. Tale algoritmo fissa il valore dei nodi delle variabili con evidenza, in accordo all'evidenza stessa, generando poi i campioni solo per i nodi restanti. Tale algoritmo pesa diversamente i vari eventi, col peso che è il likelihood dell'evento associato all'evidenza. Questo succede perché alcuni campioni sono più verosimili e altri meno e quindi pesano di più nella stima delle probabilità a posteriori. Si generano quindi le singole variabili come nel campionamento a priori saltando quelle con evidenza.
Si ha quindi che:

- l'algoritmo usa tutti i campioni generati, essendo quindi più efficiente del campionamento con rigetto
- all'aumentare del numero di nodi con evidenza la performance degrada in quanto molti campioni estratti hanno peso infinitesimale e quindi la stima sarà dipendente da pochissimi campioni con peso comunque piccolo. La stima dipende quindi da pochissimi campioni

Esempio su slide.

3.4.2 Markov Chain Monte Carlo

Passiamo ora all'algoritmo **Markov Chain Monte Carlo (*MCMC*)**.
I metodi precedenti usavano il campionamento diretto mentre altri si basano appunto su MCMC, generando una sequenza di campioni uno dipendente dall'altro. MCMC non genera ogni veneto partendo da 0 ma tramite modifiche causali di un evento che lo precede.
Pensiamo alla rete in un certo stato, identificato tramite l'assegnamento di un valore ad ogni nodo, avendo una configurazione. Lo stato successivo è ottenuto

tramite campionamento di una delle variabili senza evidenza, X_i, condizionalmente ai valori correnti assunti dalle variabili della Markov Blanket di X_i.
MCMC si muove casualmente nello spazio degli stati della rete campionando una variabile alla volta, escluse le variabili con evidenza che rimangono fissate al valore di evidenza.
Esempio su slide.
La giustificazione teorica di MCMC non viene approfondita ma si noti che è possibile estrarre un campione dalla distribuzione di una variabile X_i, noto lo stato delle variabili casuali della Markov Blanket di tale variabile, $mb(X_i)$. Il campionamento si basa sul fatto che:

$$P(x_i|MB(X_I)) = \alpha \cdot P(x_i|parents(X_i)) \cdot \prod_{Y_i \in children(X_I)} P(y_i|parents(Y_i))$$

avendo che la probabilità di una variabile casuale dato il suo Markov Blanket è proporzionale alla probabilità della variabile dati i genitori moltiplicata la probabilità di ogni suo figlio, dati i rispettivi genitori.

Capitolo 4

Markov Chains

Aggiungiamo la componente temporale al ragionamento probabilistico. Si aggiunge quindi incertezza sui cambiamenti dell'ambiente nel corso del tempo, assente nello studio delle reti Bayesiane. Si vuole prendere decisioni in un ambiente che evolve nel tempo, avendo un **ambiente dinamico**.
Per descrivere questo mondo mutevole si usano variabili casuali, descritte da uno stato in ogni istante temporale. La relazione tra variabili casuali in istanti diversi di tempo descrivono l'evoluzione dello stato. Uno stato al tempo t dipende da quello al tempo $t - x$.
Si passa quindi da **modelli statici**, come le reti Bayesiane, dove il valore delle variabili non cambia nel tempo, a **modelli dinamici** dove il valore delle variabili cambia nel tempo e lo stato corrente dipende da quelli passati, dipende dalla *storia*.
Il **processo di cambiamento** è descritto da una serie di *time slice*, ciascuno con un insieme di variabili causali.

Definizione 15. *Definiamo un* ***processo stocastico*** *come:*

$$\{X(t), t \in T\}$$

come un insieme di variabili casuali $X(t), \forall t$ dove ogni variabile evolve nel tempo. L'insieme T di indici e lo spazio X possono essere continui o discreti.
Si hanno:

- *processi stocastici a tempo continuo, $X(t), t > 0$*
- *processi stocastici a tempo discreto, $X(t), t = 0, 1, \ldots, n$*
- *processi stocastici a stati continui*
- *processi stocastici a stati discreti*

$X(t)$ è quindi il valore dello stato del sistema al tempo t, essendo il valore di una variabile causale che descrive tale stato.

Il processo stocastico più semplice è detto **random walk**, usate anche per campionare grafi complessi. In questo caso si ha tempo discreto e spazio degli stati discreto. Si ha:

$$X_t = X_{t-i} + \varepsilon_t, \;\; t = 1, 2, 3, \ldots$$

con ε_t elemento casuale. In questo caso i è fisso e solitamente vale 1. Ad esempio potrei avere:

$$\varepsilon_t = \{-1, 1\}$$

con:

$$P(\varepsilon_t = -1) = P(\varepsilon_t = 1) = \frac{1}{2}$$

Usando, ad esempio, $P > \frac{1}{2}$ ottengo una **random walk con drift**, avendo scelte spesso "in salita".
Potrei avere ε_t con valori continui, magari in una normale standard, $\varepsilon_t \approx N(0, 1)$. In tal caso lo spazio degli stati diventa continuo anche se il tempo resta discreto.
Un altro processo stocastico è il **processo autoregressivo del primo ordine (*first order autoregressive process*)**, dato dall'equazione:

$$X_t = aX_{t-i} + b + \varepsilon_t$$

con a e b constanti e $-1 < a < 1$. In questo caso i è fisso e solitamente vale 1. Si ha che $\varepsilon_t \approx N(0, 1)$.
Esempi su slide.

4.1 Processi Markoviani

Una proprietà importante dei processi stocastici è la **proprietà Markoviana**.

Definizione 16. *La **proprietà Markoviana** assicura che la distribuzione di probabilità per tutti i possibili valori futuri del processo dipende solo dal valore corrente e non dai valori passati o da altre informazioni correnti. Ovvero:*

$$P(x_{t+1} = i_{t+1} | X_t = i_i, X_{t-1} = i_{t-1}, \ldots, X_0 = i_0) = P(X_{t+1} = i_{t+1} | X_t = i_t)$$

Definizione 17. *I processi stocastici che soddisfano la **proprietà Markoviana** sono detti, equivalentemente, **processi di Markov**, **processi Markoviani**, o **processi con assenza di memoria**.*

Definizione 18. *Un processo stocastico a tempi discreti è detto **catena di Markov** se, per $t = 1, 2, 3, \ldots$ e per tutti gli stati vale:*

$$P(x_{t+1} = j | X_t = i, X_{t-1} = i_{t-1}, \ldots, X_0 = i_0) = P(X_{t+1} = j | X_t = i)$$

Se:

$$P(X_0 = i) = q_i$$

si ha che la distribuzione di probabilità iniziale è:

$$q = [q_1, \ldots, q_i, \ldots, q_n]$$

che dice come è distribuito lo stato iniziale della carena di Markov.

Definizione 19. *Se la probabilità di un certo evento è indipendente dal tempo t si ha una* ***catena di Markov stazionaria****, avendo:*

$$P(X_{t+1} = j | X_t = i) = p_{ij}$$

Con p_{ij} che è la probabilità che al tempo $t+1$ il sistema sarà nello stato j essendo nello stato i al tempo t.
Questa è la ***proprietà di stazionarietà*** *e semplifica lo studio dei modelli.* ***Stazionario quindi non significa statico, avendo comunque un'evoluzione basata sul tempo****.*
Un modello non stazionario, preso in parti più piccole, per diversi periodi di tempo, può essere studiato come una serie di modelli stazionari, semplificando lo studio.
Si ha quindi che p_{ij} rappresenta la probabilità di raggiungere uno stato j partendo da uno stato i della catena, avendo quindi una matrice di probabilità P, con valori positivi e righe che sommano a 1:

$$p_{ij} \geq 0$$

$$\sum_{j=0}^{n} p_{ij} = 1$$

La markov chain è quindi rappresentata da tale matrice P, detta ***matrice di transazione (a un passo)****.*

Ovviamente la matrice di rappresentazione posso rappresentarla come grafo. In tale grafo ogni nodo rappresenta uno stato e l'arco (i, j), con quindi i e j stati, rappresenta la probabilità di transizione p_{ij}.

Definizione 20. *Per una markov chain si hanno, per descrivere una variabile casuale:*

- *insieme degli stati $\{S_1, \ldots S_n\}$ che sono i valori che può assumere la variabile casuali*
- *probabilità di transizione di stati dallo stato i a quello j, data da $p_{ij} = P(X_{t+1} = S_j | X_t = S_i)$*
- *la distribuzione iniziale degli stati, avendo stato iniziale 0, data da $\pi_i = P[X_0 = S_i]$*

Esempio su slide.
Se una catena di Markov in uno stato i al tempo m si può calcolare la probabilità che dopo n passi sia nello stato j tramite:

$$P(X_{m+1} = j|X_m = i) = P(X_n = j|X_0 = i) = p_{ij}(n)$$

Con $p_{ij}(n)$ che dice la probabilità di passare da i a j in n passi. Ne segue che, per due passi:

$$p_{ij}(2) = \sum_{k=1}^{n} p_{ik} \cdot p_{ki}$$

avendo il prodotto scalare tra la riga i e la colonna j.
Il caso generale è che:

$$p_{ij}(n) = \text{ij-esimo elemento di } P^n$$

Esempio su slide.

Teorema 5. *La risoluzione della probabilità di transizione a n passi:*

$$P_{ij}^n = P\{X_{n+k} = j|X_k = i\}, n, i, j \geq 0$$

si risolve con le ***equazioni di Chapman-Kolmogorov****:*

$$P_{ij}^{n+m} = \sum_{k=0}^{\infty} P_{ik}^n \cdot P_{kj}^m, \forall n, m \geq 0 \ e \ i, j \geq 0$$

Avendo che:

$$P(n + m) = P(n) \cdot P(m)$$

Dimostrazione. Si ha che:

$$P_{ij}^{n+m} = P\{X_{n+m} = j|X_0 = i\}$$

ma allora:

$$P_{ij}^{n+m} = \sum_{k=0}^{\infty} P\{X_{n+m} = j, X_n = k|X_0 = i\}$$

e quindi:

$$P_{ij}^{n+m} = \sum_{k=0}^{\infty} P\{X_{n+m} = j|X_n = k, X_0 = i\} \cdot P\{X_n = k|X_0 = i\}$$

concludendo che:

$$P_{ij}^{n+m} = \sum_{k=0}^{\infty} P_{ik}^n \cdot P_{kj}^m$$

□

Teorema 6. *La probabilità di essere in uno stato j al tempo n, non conoscendo la catena di Markov al tempo 0 è:*

$$\sum_i q_i \cdot p_{ij}(n) \cdot (\textit{colonna } j \textit{ di } p^n)$$

Definizione 21. *Uno stato j è **raggiungibile** dallo stato i se per qualche n si ha un cammino da i a j:*

$$P_{ij} = n > 0$$

Definizione 22. *Due stati **comunicano** se uno stato è raggiungibile dall'altro e viceversa. Ogni stato comunica con se stesso e si ha la proprietà transitiva.*

Definizione 23. *Una catena di Markov è detta **irriducibile** se tutti i suoi stati sono comunicanti fra loro*

Definizione 24. *Un insieme di stati S in una catena di Markov è un insiemechiuso se nessuno stato fuori S è raggiungibile dagli stati in S.*

Definizione 25. *Uno stato i **assorbente** se $P_{ii} = 1$. Tale stato rappresenta un **deadlock**.*

Definizione 26. *Uno stato i è **transiente** se esiste j raggiungibile da i ma non vale il viceversa. Si ha che:*

$$\sum_{n=1}^{\infty} P_{ii}^n < \infty$$

*Uno stato non transiente è detto **ricorrente**, avendo:*

$$\sum_{n=1}^{\infty} P_{ii}^n = \infty$$

Teorema 7. *Se lo stato i è ricorrente e j comunica con i allora j è ricorrente, in quanto la ricorrenza è una **proprietà di classe**.*
Anche essere transiente è una proprietà di classe,

Definizione 27. *Una catena di Markov è **finita** sse il numero di stati è finito.*

Teorema 8. *Tutti gli stati di una catena di Markov finita irriducibile sono ricorrenti.*

Definizione 28. *Uno stato i è **periodico**, con periodo $k > 1$ se k è il più piccolo numero tale che tutti i cammini che dallo stato i tornano ad i hanno una lunghezza multipla di k.*
*Uno stato non periodico è detto **aperiodico**.*

Definizione 29. *Si definisce catena di Markov **ergodica** se tutti gli stati in una catena sono ricorrenti, aperiodici e comunicano l'uno con l'altro.*
*Una catena di Markov ergodica permette di raggiunger una distribuzione di equilibrio ed è usata anche negli algoritmi di **page rank**.*

Teorema 9. *Data una matrice di probabilità P per una catena ergodica di N stati. Si ha che:*

$$\lim_{t \to +\infty} P_{ij}(t) = \pi_j$$

dipendendo quindi solo dallo stato d'arrivo. Per t abbastanza grande si arriva quindi ad uno stato di equilibrio, avendo che:

$$\pi = \pi \cdot P, \;\; \pi = [\pi_1, \ldots, \pi_n]$$

avendo stazionarietà, avendo un vettore di distribuzione di equilibrio.
Questo è alla base dell'algoritmo di page rank.
In altri termini aumentando n si ha che $P_{ij}(n)$ non cambia da un certo punto in poi, $\forall i, j$.
Per trovare il vettore di probabilità stazionarie risolvo quindi il sistema dato dai vari π_i (prendo le colonne di P come coefficienti e p_i come variabili) aggiungendo anche la condizione che $\sum_i \pi_i = 1$ (cosa che permette di usare un'equazione in meno). Il vettore dei π_i risultante è il vettore di probabilità stazionarie.

Esempio su slide.
Il comportamento di una catena di Markov prima di raggiungere una distribuzione di equilibrio è detto **transitorio**, transitando da alcuni stati prima dell'equilibrio.
Il numero di transizioni attese prima di raggiungere lo stato j, essendo nello stato i, in una catena ergodica è:

$$m_{ij} = p_{ij} \cdot (1) + \sum_{k \neq j} p_{ik} \cdot (1 + m_{kj}) = 1 + \sum_{k \neq j} p_{ik} \cdot m_{kj} = \frac{1}{\pi_i}$$

Esempio su slide.

Definizione 30. *Le catene assorbenti sono catene di Markov nelle quali alcuni stati sono assorbenti, mentre tutti gli altri sono stati transienti.*

Si ha interesse a chiedersi quanti passi intercorrono da una stato transiente ad uno assorbente e la probabilità di iniziare con uno stato transiente e terminare in uno stato assorbente (da cui si ricorda "non si esce più").

Definizione 31. *Bisogna considerare la **matrice di transizione** scritta come:*

$$\begin{bmatrix} Q & R \\ 0 & I \end{bmatrix}$$

con:

- Q *matrice che rappresenta le relazioni tra stati transienti*
- R *matrice che rappresenta le transizioni da statitransienti a stati assorbenti*
- 0 *matrice di soli 0*
- I *matrice identità*

Definizione 32. *Se si è in uno stato transiente i il numero di periodi che si trascorrono in j transiente prima dell'assorbimento nello stato è il ij-esimo elemento della matrice $(I - Q)^{-1}$.*

Esempio su slide.

Definizione 33. *Se si è in uno stato transiente i la probabilità di arrivare ad uno stato assorbente j è il ij-esimo elemento della matrice $(I - Q)^{-1} \cdot R$.*

Vari esempi su slide, tra cui pagerank.

Teorema 10. *Sotto certe condizioni, ovvero:*

- *esiste una distribuzione stazionaria unica q con $q_i > 0, \forall i$*
- *dato $N(i, t)$ il numero di volte che la catena di Markov visita lo stato i in t step*

si ha che:

$$\lim_{t \to \infty} \frac{N(i,t)}{t} = \pi_i$$

Questo teorema è usato nel pagerank. Quindi data una pagina u il pagerank è la probabilità per la catena di Markov:

$$pagerank(u) = \frac{p}{n + (1-p)} \sum_{(v,u) \in E} \frac{pagerank(v)}{outdegree(v)}$$

con:

- n *numero totale di nodi del grafo*
- p *probabilità di fare un "random jump", ovvero di capitare per caso nella pagina*

Questa regola è indipendente dalla query e tiene conto del "web opinion", ovvero l'importanza della pagina.

4.2 Hidden Markov Models

Le catene di Markov sono un modello importante e molto utilizzato.
Un altro modello essenziale è quello **hidden Markov models (*HMM*)**, dove non si possono osservare gli stati.
Le reti Bayesiane possono usare le catene di Markov con la peculiarità che alcune variabili sono nascoste, per trattare informazioni parziali con incertezza. L'uso quindi copre varie tematiche, come *serie storiche*, *speech to text*, *analisi musicale*, etc...
Si ha quindi un modello dinamico per cui servono:

- un insieme X_t di variabili di stato non osservabili al tempo t, come per una rete bayesiana
- un insieme E_t di variabili osservabili al tempo t
- la dipendenza tra le variabili
- l'ipotesi che i cambiamenti siano regolati da un processo stazionario, la relazione per passare tra x_t e x_{t+1} non cambia nel tempo
- l'ipotesi di Markov del primo ordine, ovvero tutta l'informazione del passato è racchiusa nello stato corrente. In questo modo le CPT restano semplici

Si uniscono quindi reti bayesiane che evolvono nel tempo e processo Markoviano, si ha la componente dinamica.
Si ha quindi, per un processo Markoviano del primo ordine:

$$P(X_t|X_{0:t-1}) = P(X_t|X_{t-1})$$

Posso anche andare al secondo ordine, dicendo che la storia passata è racchiusa negli ultimi due stati e non sono nell'ultimo:

$$P(X_t|X_{0:t-1}) = P(X_t|X_{t-2:t-1})$$

Definizione 34. *Una **catena di Markov a stati nascosti** è un processo stocastico a tempo continuo o discreto (faremo solo discreto) caratterizzato da:*

- *un insieme di stati X_t*
- *un insieme di osservazioni E_t*
- *una matrice delle probabilità di transizione con $P(X_t|X_{0:t-1}) = P(X_t|X_{t-1})$, avendo un **modello di transizione**, equivalente della matrice di transizione per le normali catene di Markov*

- *una matrice delle probabilità di emissione delle osservazioni con* $P(E_t|X_{0:t-1}, E_{0:t-1}) = P(E_T|X_t)$*, avendo un* ***modello sensoriale****, che da la probabilità di osservare un evento in un certo stato essendo in quel preciso stato*
- *una matrice delle probabilità iniziali degli strati al tempo 0, ovvero* $P(X_0)$

Si ha quindi, $\forall t$ *finito, la seguente distribuzione congiunta:*

$$P(X_0, X_1, \ldots, X_t, E_1, \ldots, E_t) = P(X_0) \prod_{i=1}^{t} P(X_i|X_{i-1})P(E_i|X_i)$$

Avendo un elemento aggiuntivo (l'ultimo) rispetto alle semplici catene di Markov.

Bisogna fare inferenza sullo stato in cui ci si trova in base alle evidenze.
L'ipotesi di Markov del prim'ordine è comunque tanto utile per contenere la complessità quanto forte, in quanto suppone che le variabili di stato contengano tutte le informazioni necessarie per caratterizzare la distribuzione dell'istante successivo. Per risolvere il problema posso aumentare l'ordine del modello (ma anche la complessità, come si è visto sopra per il secondo ordine etc...) o posso aumentare il numero delle variabili di stato.
Si hanno vari task di inferenza:

- **filtraggio**, data una sequenza di osservazioni $E_1, \ldots E_t$ mi da la distribuzione di probabilità di X_t
- **previsione**, data una sequenza di osservazioni $E_1, \ldots E_t$ probabilità di avere un evento al tempo $t + n$, con n arbitrario
- **smoothing**, ovvero fare un inferenza su uno stato passato basandosi su tutta la storia fino al tempo attuale
- **ricerca della spiegazione più probabile**, cercando la sequenza di stati più probabili che porta ad un certo evento

Si può anche fare **apprendimento** del modello di transizione o di quello sensoriale come sottoprodotto dell'inferenza, tramite **expectation-maximation**. In altri termini fare apprendimento è fare uno smoothing completo.

4.2.1 Filtraggio

Il filtraggio è quindi il calcolo della distribuzione a posteriori dello stato corrente date tutte le osservazioni $P(X_t|e_{1:t})$, facendo quindi una stima ricorsiva basata sulle distribuzioni fino al tempo $t - 1$:

$$P(X_t|e_{1:t}) = f(e_t, P(X_{t-1}|e_{1:t-1}))$$

Se il modello sensoriale fosse deterministico si avrebbe una semplice catena di Markov
Il filtraggio è composto da:

- **proiezione**
- **aggiornamento**

Sapendo il filtraggio al tempo t si cerca quello a $t+1$ e si fa aggiornando in base alla nuova prova e_{t+1}:

$$P(X_{t+1}|e_{1:t+1})$$

divido quindi l'evidenza:

$$= P(X_{t+1}|e_{1:t}, e_{t+1})$$

applico Bayes:

$$= \alpha P(e_{t+1}|X_{t+1}, e_{1:t})P(X_{t+1}|e_{1:t})$$

e infine uso la proprietà di Markov dell'evidenza:

$$= \alpha P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t})$$

Il secondo termine rappresenta la predizione del prossimo step e il primo aggiorna la previsione sulla nuova evidenza. Si condiziona quindi lo stato corrente X_t per ottenere il nuovo stato:

$$P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(X_{t+1}|x_t, e_{1:t})P(e_t|e_{1:t})$$

Ma usando la proprietà di Markov si ha:

$$\alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(X_{t+1}|x_t)P(e_t|e_{1:t})$$

Avendo che:

$$f_{1:t+1} = \alpha forward(f_{1:t}; e_{t+1})$$

4.2.2 Predizione

La **predizione** è una sorta di filtraggio senza aggiunta di nuove informazioni/osservazioni, prevedendo gli stati futuri:

$$P(X_{t+k}|e_{1:t})$$

Maggiore è l'incertezza del tempo di transizione e prima, in termini di tempo di mixing/tempo transiente, raggiungerò un punto fisso per la previsione, raggiungendo una distribuzione stazionaria, e più ignoto sarà il futuro.
Esempio su slide.

Questo algoritmo possiamo pensarlo come un **mini-forward**, essendo un filtraggio senza aggiustamento. Trovare $P(X_t)$ dati $P(X_0)$ e $P(X_t|X_{t-1})$ si fa quindi:

$$P(X_t) = \sum_{x_{t-1}} P(X_t, x_{t-1}) = \sum_{x_{t-1}} P(X_t|x_{t-1})P(x_{t-1})$$

dove:

- $P(X_t|x_{t-1})$ è conosciuta dal modello di transizione
- $P(x_{t-1})$ è conosciuto da $P(X_0)$ o da simulazioni precedenti

Per raggiungere lo stato stazionario si fa come con le catene di Markov.

4.2.3 Calcolo della Verosimiglianza

La **verosimiglianza** è la probabilità che un modello abbia generato una sequenza di osservazioni data. Questo è utile per confrontare diverse sequenze di prove da diversi modelli. È come se avessi un modello di classificazione. Indico la sequenza di prove con:

$$P(e_{1:t})$$

Per confrontare i modelli calcolo quindi:

$$l_{1:t} = P(X_t, e_{1:t})$$

e:

$$l_{1:t+1} = forward(l_{1:t}; e_{t+1})$$

avendo anche la probabilità di una sequenza di prove, calcolata in modo ricorsivo:

$$L_{1:t} = P(e_{1:t}) = \sum_{x_t} l_{1:t}(x_t)$$

4.2.4 Smoothing

Lo **smoothing**, detto anche **regolarizzazione**, è il processo di calcolo della distribuzione di stati passati date le osservazioni fino allo stato corrente. È il ragionamento del tipo "col senno di poi". Si vuole calcolare quindi:

$$P(X_k|e_{1:t}) \text{ per } 1 \leq k < t$$

Considero quindi separatamente:

- le osservazioni fino a k
- le osservazioni da $k+1$ a t, con ovviamente $t > k$

Ho quindi:

$$P(X_k|e_{1:t}) = P(X_k|e_{1:k}, e_{k+1:t})$$

Applico quindi Bayes:

$$P(X_k|e_{1:t}) = \alpha P(X_k|e_{1:k})P(, e_{k+1:t}|X_k, e_{1:k})$$

e per l'indipendenza condizionata:

$$P(X_k|e_{1:t}) = \alpha P(X_k|e_{1:k})P(, e_{k+1:t}|X_k)$$

ottenendo, con $f_{1:k}$ che consiste nel filtrare in avanti da 1 a k, in modo quindi forward:

$$P(X_k|e_{1:t}) = \alpha f_{1:k} b_{k+1:t}$$

Il "pezzo" di raccolta prove $b_{k+1:t}$ è detto backward e si calcola con:

$$P(e_{k+1:t}|X_k) = \sum_{x_{k+1}} P(e_{k+1:t}|X_k, x_{k+1})P(x_{k+1}|X_k) = \sum_{x_{k+1}} P(e_{k+1:t}|x_{k+1})P(x_{k+1}|X_k)$$

$$\sum_{x_{k+1}} P(e_{k+1}, e_{k+2:t}|x_{k+1})P(x_{k+1}|X_k) = \sum_{x_{k+1}} P(e_{k+1}|x_{k+1})P(e_{k+2:t}|x_{k+1})P(x_{k+1}|X_k)$$

Dove il primo e il terzo termine sono tratte dal modello e quindi li conosco e il termine mediano la chiamata ricorsiva.
Nell'esecuzione quindi si memorizzano i risultati del filtraggio forward su tutta la sequenza e si esegue la ricorsione all'indietro:

---

function FORWARDBACKWARD($ev, priori$)
  $fv[0] \leftarrow priori$
  for $i \leftarrow 1$ **to** n **do**
    $fv[i] \leftarrow Forward(fv[i-1], ev[i])$
  $b \leftarrow \vec{1}$
  for $i \leftarrow t$ **downto** 1 **do**
    $sv[i] \leftarrow Normalize(fv[i] \cdot b)$
    $b \leftarrow Backward(b, ev[i])$
  return sv

---

con:

- ev un vettore di dimensione t di valori di prova
- $priori$ distribuzione a priori dello stato iniziale
- fv vettore di messaggi in avanti per i passi da 0 a t

- b rappresentazione del messaggio all'indietro, inizialmente $\vec{1}$
- st vettore di stime regolarizzate dei passi da 1 a t

Si ha quindi che lo smoothing ha una complessità spaziale elevata a causa dei molti stati e delle lunghe sequenza. Permette però di usare tutta l'informazione disponibile e va usato solo quando interessa avere una stima precisa del passato.
Esempio su slide.
Una soluzione è il cosiddetto **smoothing a ritardo fisso** (se ha detto bene cosa fosse me lo sono perso).

4.2.5 Sequenza più Probabile

Vediamo quindi all'ultimo task, quello della **sequenza più probabile**, ovvero data una sequenza di osservazioni vogliamo trovare la sequenza di stati che più probabilmente ha generato il mio set di osservazioni:

$$\underset{x_{1:t}}{\operatorname{argmax}} P(x_{1:t}|e_{1:t})$$

Supposto che ogni stato possa assumere solo valori binari e che la mia sequenza di osservazioni sia lunga n allora si avranno 2^n possibili sequenze di stati, avendo qualcosa di computazionalmente troppo costoso.
Si ragiona quindi in ottica di algoritmo di ottimizzazione dinamica, considerando ogni sequenza come un cammino lungo un grafo, avendo che la probabilità di ogni cammino è il prodotto delle probabilità di transizione per le probabilità delle osservazioni rilevate ad ogni stato. Si usa l'**algoritmo di Viterbi** che si basa su un'assunzione:

esiste una relazione ricorsiva fra i cammini più probabili verso ogni stato x_{t+1} e i cammini più probabili verso ogni stato x_t.

Questa relazione ricorsiva è del tipo:

$$\max_{x_1 \dots x_t} P(x_1, \dots x_t, X_{t+1}|e_{1:t+1})$$

$$= \alpha P(e_{t+1}) \max_{x_t}(P(X_{t+1}|x_t) \max_{x_1,\dots x_{t-1}} P(x_1, \dots x_{t-1}, x_t|e_{1:t})$$

Al termine sarà disponibile la probabilità della sequenza più probabile che raggiunge ogni stato finale. Si cerca quindi il cammino più lungo sul grafo.
Vediamo quindi come ottenere la sequenza più probabile al tempo t: m_t:

$$m_t[x] = \max_{x_{1:t-1}} P(x_{1:t-1}, x) = \max_{x_{1:t-1}} P(x_{1:t-1})P(x|x_{t-1})$$

$$= \max_{x_{t-1}} P(x_t|x_{t-1}) \max_{x_{1:t-2}} P(x_{1:t-1}) = \max_{x_{t-1}} P(x_t|x_{t-1})m_{t-1}[x]$$

Avendo:

$$m_1[x] = P(x_1)$$

Per vedere invece la sequenza più probabile di stati date le osservazioni:

$$x^*_{1:T} = \underset{x_{1:T}}{\text{argmax}}\, P(x_{1:T}|e_{1:T})$$

Avendo quindi:

$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t}) = \max_{x_{1:t-1}} P(x_{1:t-1}, e_{1:t-1})P(x_t|x_{t-1})P(e_t|x_t)$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) \max_{x_{t-2}} P(x_{1:t-1}|e_{1:t-1})$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_t - 1]$$

Esempi su slide.

Capitolo 5

Filtri di Kalman

Sentire primi 5 minuti.
Si vuole stimare lo stato i un sistema dinamico partendo da una sequenza di osservazioni rumorose. Si hanno:

- un **modello di transizione** che descrive la fisica di moto. È la previsione
- un **modello sensoriale** che descrive il processo di misurazione. È la correzione della previsione

Inoltre si il punto chiave di differenza rispetto ad HMM, ovvero **l sistema è descritto da un insieme di variabili continue**, come posizione e velocità (di cui osserviamo solo una certa misurazione). Il modello di transizione quindi non è una catena di Markov.
Si chiama *filtro* per i due passaggi detti sopra, previsione e correzione. Posso usare una rete Bayesiana per rappresentarlo.
Supponiamo che l'intervallo di tempo tra due osservazioni sia Δ e che la velocità tra due istanti temporali sia costante, allora l'aggiornamento della posizione avviene tramite la formula:

$$X_{t+\Delta} = X_t + \dot{X}\Delta$$

Aggiungendo del rumore gaussiano otteniamo un modello di transizione gaussiano lineare, del rumore bianco, si ha:

$$P(X_{t+\Delta} = x_{t+\Delta}|X_t = x_t, \dot{X}_t = \dot{x}_t) = N(x_t + \dot{x}_t\Delta, \sigma^2)(x_{t+\Delta})$$

Si ricorda che la gaussiana è unimodale, avendo un solo massimo, avendo che a probabilità a posteriori si focalizza attorno al vero stato con poca incertezza.
L'idea di base è quindi che le belief siano rappresentate da distribuzioni normali multivariate.
Se la distribuzione corrente $P(X_t|e_{1:t})$ è Gaussiana e il modello di transizione

$P(X_{t+1}|x_t)$ è lineare, allora la predizione ad un passo sarà anch'essa Gaussiana:

$$P(X_{t+1}|e_{1:t}) = \int_{x_t} P(X_{t+1}|x_t)P(x_t|e_{1:t})\,\mathrm{d}x_t$$

Si ha quindi l'assunzione che il modello di moto sia lineare gaussiano e che il rumore sia anch'esso un modello gaussiano.
Se la predizione $P(X_{t+1}|e_{1:t})$ è Gaussiana e il modello sensoriale $P(e_{t+1}|X_{t+1})$ è Gaussiano lineare allora dopo aver aggiornato il modello rispetto alla nuova evidenza anche la distribuzione aggiornata, ovvero:

$$P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t})$$

è gaussiana lineare.
Ad ogni iterazione garantisco il mantenimento della gaussiana aumentando l'efficienza dell'algoritmo, si parla quindi, con queste assunzioni, che il filtro di Kalman è un **filtro ottimo** (anche se nella realtà non si possono avere tali assunzioni se non approssimando).
Il filtro si riduce quindi a calcolare ad ogni iterazione media e varianza della gaussiana.
L'operatore *forward* per il filtro di Kalman accetta un messaggio in avanti gaussiano $f_{1:t}$, specificato da una media μ e matrice di covarianza Σ_{t+1}. Iniziando da una distribuzione gaussiana iniziale $P(X_0) = N(\mu_0, \Sigma_0)$ filtrando con un modello gaussiano lineare otterremo ancora una distribuzione statale gaussiana, descrivendo la distribuzione a posteriori come una gaussiana con media e varianza. Infatti, se la distribuzione non fosse Gaussiana il filtraggio nel continuo o ibrido (discreto e continuo) genera distribuzioni dello stato la cui rappresentazione cresce senza limiti col tempo (assume forme sempre più complesse).
Non si vede la teoria matematica dietro.
Esempi su slide con conti non richiesti in esame. Si parla di **filtro di bayes**, avendo che si calcola la probabilità a posteriori di X_t condizionato alle misure e ai "controlli" (roba di robotica) fino al tempo t. Si richiedono 3 distribuzioni di probabilità, indicando con KF il filtro di Kalman:

1. la belief iniziale $P(X_0)$ con KF che assume la distribuzione dello stato iniziale Gaussiana

2. la probabilità della misura $P(Z_t|Z_t)$, con KF che assume la distribuzione di misura gaussiana lineare

3. la probabilità di transizione $P(X_t|X_{t-1})$, con KF che assume il sistema dinamico lineare

In questo caso come detto, il filtro di Kalman è un **filtro ottimo** ma con assunzioni quasi irreali (**bozza di algoritmo su slide**).
Non sempre si hanno queste assunzioni lineari (rasformazioni tra stati e misure

raramente funzioni lineari) e in tal caso, con funzioni non lineari, si parla di **Extended Kalman Filter** e in questo caso si calcola un'approssimazione gaussiana per la vera belief. Si usano approssimazioni di Taylor al primo ordine. **Materiale su slide super accennato in aula**.
Si hanno anche vari **filtri non parametrici**, come *histogram filter* (con istogrammi che rassegano a ogni regione che decompone lo spazio degli stati una singola probabilità cumulata) e *particles filter* (probabilità a posteriori tramite sottoinsiemi), dove non si hanno assunzioni parametriche rigide sulla densità di probabilità a posteriori e vanno bene per rappresentare belief multimodali.
L'idea chiave dei particles filter è quello di rappresentare la probabilità a posteriori tramite un insieme di campionamenti random dello stato disegnati da tale probabilità. Una particle è quindi un'ipotesi su come potrebbe essere lo stato al tempo t e quindi l'insieme dei particles altro non è che formato dai campioni di probabilità a posteriori, avendo:

$$x_t = x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[M]}$$

Quindi la likelihood dell'ipotesi di stato x_t per essere inclusa tra i particles deve essere idealmente proporzionale alla probabilità a posteriori del suo filtro di Bayes, avendo quindi:

$$x_t^{[m]} \sim P(x_t | z_{1:t}, u1 : t)$$

Avendo che più una regione nello spazio degli stati è densa di campioni tanto più probabile è che il vero stato sia in quella regione.
Su slide accenni dell'algoritmo.
In questo caso si usa anche il concetto di **resampling**, avendo che ciascun particle viene campionato con probabilità pari al suo peso, passando dal set temporaneo $\overline{X}_t$ a quello finale X_t (**qualche nota extra su slide**).