Apprendimento Bayesiano

[Capitolo 6, Mitchell]

- Teorema di Bayes
- Ipotesi MAP e ML
- algoritmi di apprendimento MAP
- Principio MDL (Minimum description length)
- Classificatore Ottimo di Bayes
- Apprendimento Ingenuo di Bayes (Naive Bayes)
- Richiamo di Reti Bayesiane
- Algoritmo EM (Expectation Maximization)

Metodi Bayesiani

Forniscono metodi computazionali di apprendimento:

- Apprendimento Naive Bayes
- Apprendimento di Reti Bayesiane
- Combinazione di conoscenza a priori (probabilità a priori) con dati osservati
- Richiedono probabilità a priori

Forniscono un framework concettuale utile

- Forniscono il “gold standard” per la valutazione di altri algoritmi di apprendimento
- Interpretazione del “Rasoio di Occam”

Teorema di Bayes

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

- $P(h)$ = probabilità a priori della ipotesi h
- $P(D)$ = probabilità a priori dei dati di apprendimento D
- $P(h|D)$ = probabilità di h dati D
- $P(D|h)$ = probabilità di D data h

Scelta Ipotesi

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

In generale si vuole selezionare l'ipotesi più probabile dati i dati di apprendimento

Ipotesi *"Maximum a posteriori"* h_{MAP}:

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(h|D) \\ &= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \arg\max_{h \in H} P(D|h)P(h) \end{aligned}$$

Se assumiamo $P(h_i) = P(h_j)$ allora si può ulteriormente semplificare, e scegliere la ipotesi *"Maximum likelihood"* (ML)

$$h_{ML} = \arg\max_{h_i \in H} P(D|h_i)$$

Apprendimento “Bruta Forza” dell’ipotesi MAP

1. Per ogni ipotesi h in H, calcola la probabilità a posteriori

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

2. Restituisci l’ipotesi h_{MAP} con la probabilità a posteriori più alta

$$h_{MAP} = \arg\max_{h \in H} P(h|D)$$

Interpretazione Find-S

Si consideri l'apprendimento di concetti (funzioni booleane)

- spazio delle istanze X, spazio delle ipotesi H, esempi di apprendimento D
- si consideri l'algoritmo FIND-S (restituisce l'ipotesi più specifica del version space $VS_{H,D}$)

Quale sarebbe l'ipotesi MAP ?

Corrisponde a quella restituita da FIND-S ?

Interpretazione Find-S

Assumiamo di fi ssare le istanze $\langle x_1, \ldots, x_m \rangle$

Assumiamo D essere l'insieme dei valori desiderati $D = \langle c(x_1), \ldots, c(x_m) \rangle$

Scegliamo $P(D|h)$:

Interpretazione Find-S

Assumiamo di fissare le istanze $\langle x_1, \ldots, x_m \rangle$

Assumiamo D essere l'insieme dei valori desiderati $D = \langle c(x_1), \ldots, c(x_m) \rangle$

Scegliamo $P(D|h)$:

- $P(D|h) = 1$ **se** h è consistente con D, **altrimenti** $P(D|h) = 0$

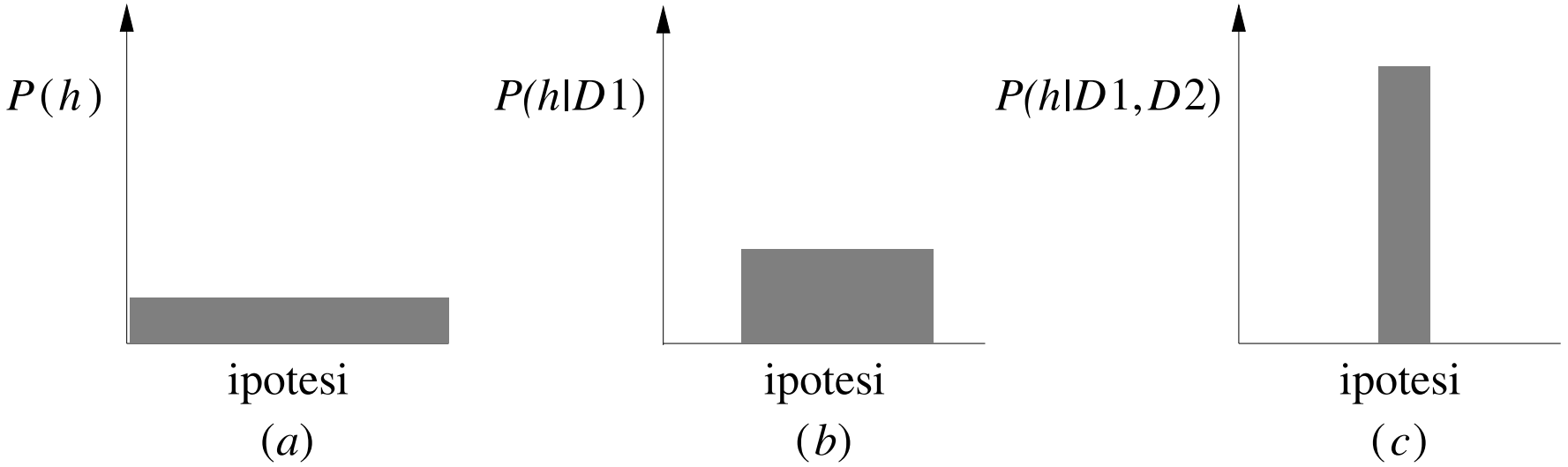
Scegliamo $P(h)$ essere la distribuzione *uniforme*

- $P(h) = \frac{1}{|H|}$ per tutte le h in H (per FIND-S, definire $P(h_i) < P(h_j)$ se $h_i >_g h_j$)

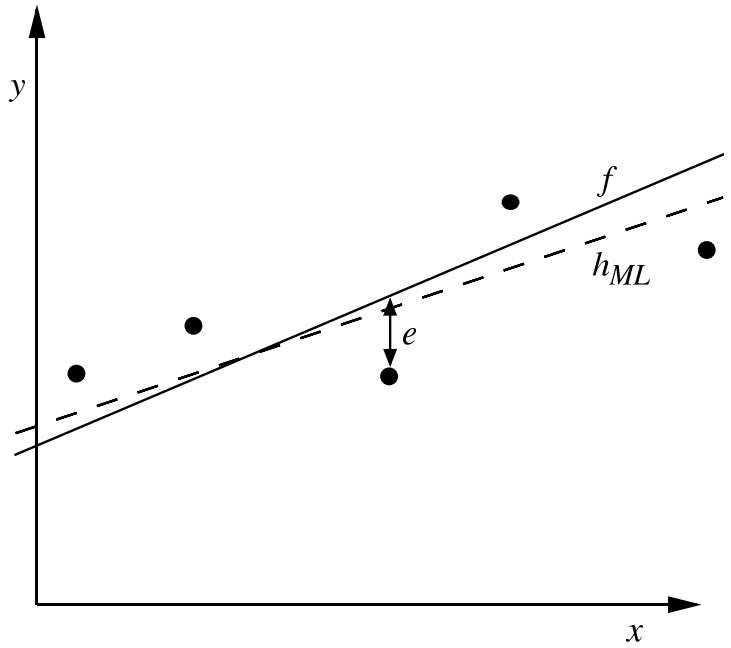
Allora,

$$P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|} & \text{se } h \text{ è consistente con } D \\ 0 & \text{altrimenti} \end{cases}$$

Evoluzione delle probabilità a posteriori



Apprendimento di una Funzione a Valori Reali



Si consideri una qualunque funzione target f a valori reali, esempi di apprendimento $\langle x_i, d_i \rangle$, dove d_i presenta del rumore

- $d_i = f(x_i) + e_i$
- e_i è una variabile random (rumore) estratta indipendente per ogni x_i secondo una distribuzione Gaussiana con media 0

Allora l'ipotesi h_{ML} è quella che minimizza:

$$h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

Apprendimento di una Funzione a Valori Reali

$$
\begin{aligned}
h_{ML} &= \arg\max_{h\in H} p(D|h) \\
&= \arg\max_{h\in H} \prod_{i=1}^{m} p(d_i|h) \\
&= \arg\max_{h\in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2}
\end{aligned}
$$

che si tratta meglio massimizzando il logaritmo naturale...

$$
\begin{aligned}
h_{ML} &= \arg\max_{h\in H} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\
&= \arg\max_{h\in H} \sum_{i=1}^{m} -\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 \\
&= \arg\max_{h\in H} \sum_{i=1}^{m} -\left(d_i - h(x_i)\right)^2 \\
&= \arg\min_{h\in H} \sum_{i=1}^{m} \left(d_i - h(x_i)\right)^2
\end{aligned}
$$

Apprendere Probabilità

Si consideri il problema di predire la probabilità di sopravvivenza di un paziente malato

Esempi di apprendimento $\langle x_i, d_i \rangle$, dove d_i è 1 o 0

Si vuole allenare una rete neurale a restituire in output la *probabilità* dato x_i (non uno 0 o 1)

In questo caso si può mostrare che

$$h_{ML} = \arg\max_{h \in H} \sum_{i=1}^{m} d_i \ln h(x_i) + (1 - d_i) \ln(1 - h(x_i))$$

Regola di apprendimento per i pesi di una unità sigmoidale:

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk}$$

dove

$$\Delta w_{jk} = \eta \sum_{i=1}^{m} (d_i - h(x_i))\, x_{ijk}$$

Principio MDL (Minimum Description Length)

Rasoio di Occam: preferire l'ipotesi più semplice

MDL: preferire l'ipotesi h che minimizza

$$h_{MDL} = \arg\min_{h \in H} L_{C_1}(h) + L_{C_2}(D|h)$$

dove $L_C(x)$ è la lunghezza di descrizione di x sotto la codifica C

Esempio: H = alberi di decisione, D = etichette di allenamento

- $L_{C_1}(h)$ è # bit per descrivere l'albero h
- $L_{C_2}(D|h)$ è # bit per descrivere D dato h
  - Notare che $L_{C_2}(D|h) = 0$ se gli esempi sono classifi cati perfettamente da h. Basta descrivere solo le eccezioni
- Quindi h_{MDL} cerca un compromesso fra dimensione albero e numero errori

Principio MDL (Minimum Description Length)

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(D|h)P(h) \\ &= \arg\max_{h \in H} \log_2 P(D|h) + \log_2 P(h) \\ &= \arg\min_{h \in H} -\log_2 P(D|h) - \log_2 P(h) \end{aligned}$$

Dalla Teoria dell'Informazione:

> Il codice ottimo (il codice con lunghezza aspettata pi`u corta) per un evento con probabilità p è $-\log_2 p$ bit.

Pertanto:

- $-\log_2 P(h)$ è la lunghezza di h usando un codice ottimo
- $-\log_2 P(D|h)$ è la lunghezza di D dato h usando un codice ottimo

$\Rightarrow$ preferire l'ipotesi che minimizza

$$lunghezza(h) + lunghezza(errori)$$

Classifi cazione pù probabile di nuove istanze

Finora abbiamo cercato l'*ipotesi* più probabile dati i dati D (cioè, h_{MAP})

Data una nuova istanza x, qual' è la *classificazione* più probabile ?

- $h_{MAP}(x)$ non è la calssificazione più probabile!

Consideriamo:

- tre possibili ipotesi:

$$P(h_1|D) = .4,\ P(h_2|D) = .3,\ P(h_3|D) = .3$$

- data una nuova istanza x,

$$h_1(x) = +,\ h_2(x) = -,\ h_3(x) = -$$

- qual' è la classificazione più probabile per x?

Classifi catore Ottimo di Bayes

Classificazione Ottima di Bayes:

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Esempio:

$$P(h_1|D) = .4, \quad P(-|h_1) = 0, \quad P(+|h_1) = 1$$
$$P(h_2|D) = .3, \quad P(-|h_2) = 1, \quad P(+|h_2) = 0$$
$$P(h_3|D) = .3, \quad P(-|h_3) = 1, \quad P(+|h_3) = 0$$

pertanto

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) = .4 \qquad \sum_{h_i \in H} P(-|h_i)P(h_i|D) = .6$$

e

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) \quad = \quad -$$

Classifi catore di Gibbs

Il classificatore ottimo di Bayes può essere molto costoso da calcolare se ci sono molte ipotesi

Algoritmo di Gibbs:

1. Scegliere una ipotesi a caso, secondo $P(h|D)$
2. Usarla per classificare la nuova istanza

Fatto sorprendente: assumiamo che i concetti target siano estratti casualmente da H secondo una probabilità a priori su H. Allora:

$$E[errore_{Gibbs}] \leq 2E[errore_{BayesOttimo}]$$

Supponendo distribuzione a priori uniforme su ipotesi corrette in H,

- Seleziona una qualunque ipotesi da VS, con probabilità uniforme
- Il suo errore aspettato non è peggiore del doppio dell'errore ottimo di Bayes

Classifi catore Naive di Bayes

Una delle tecniche più semplici e popolari

Quando usarlo:

- insiemi di dati di dimensione abbastanza grande
- gli attributi che descrivono le istanze sono condizionalmente indipendenti data la classificazione

Applicazioni su cui ha avuto successo:

- Diagnosi
- Classificazione di documenti testuali

Classifi catore Naive di Bayes

Funzione target $f : X \to V$, con istanze x descritte da attributi $\langle a_1, a_2 \ldots a_n \rangle$.

Il valore pi`u probabile di $f(x)$è:

$$\begin{aligned} v_{MAP} &= \arg\max_{v_j \in V} P(v_j | a_1, a_2 \ldots a_n) \\ v_{MAP} &= \arg\max_{v_j \in V} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)} \\ &= \arg\max_{v_j \in V} P(a_1, a_2 \ldots a_n | v_j) P(v_j) \end{aligned}$$

assunzione Naive di Bayes:

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

che defi nisce il

Classificatore Naive di Bayes: $v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$

Algoritmo Naive di Bayes

Naive_Bayes_Learn($esempi$)

For each valore target v_j

$\hat{P}(v_j) \leftarrow$ stima $P(v_j)$

For each valore di attributo a_i di ogni attributo a

$\hat{P}(a_i|v_j) \leftarrow$ stima $P(a_i|v_j)$

Classify_New_Instance(x)

$$v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

Naive Bayes: Esempio

Consideriamo il problema *Giocare a Tennis*, e la nuova istanza

$$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong \rangle$$

Vogliamo calcolare:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$$

Giocare a Tennis!!

E' la giornata ideale per giocare a Tennis ?

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

Naive Bayes: Esempio

Consideriamo il problema *Giocare a Tennis*, e la nuova istanza

$$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong \rangle$$

Vogliamo calcolare:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

$$P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y) = .005$$

$$P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n) = .021$$

$$\rightarrow v_{NB} = n$$

Naive Bayes: Considerazioni Aggiuntive

1. L'assunzione di indipendenza condizionale è spesso violata

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

- ...ma sembra funzionare comunque. Notare che non è necessario stimare correttamente la probabilità a posteriori $\hat{P}(v_j|x)$; è sufficiente che

$$\arg\max_{v_j \in V} \hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = \arg\max_{v_j \in V} P(v_j) P(a_1 \ldots, a_n | v_j)$$

- la probabilità a posteriori calcolata da Naive Bayes è spesso vicina a 1 o 0 anche se non dovrebbe

Naive Bayes: Considerazioni Aggiuntive

2. cosa succede se nessun esempio di apprendimento con valore di target v_j possiede valore di attributo a_i? In tal caso

$$\hat{P}(a_i|v_j) = 0, \text{ e... } \hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = 0$$

Una soluzione tipica è la stima Bayesiana per $\hat{P}(a_i|v_j)$

$$\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$$

dove

- n è il numero di esempi di apprendimento per cui $v = v_j$,
- n_c numero di esempi per cui $v = v_j$ e $a = a_i$
- p è la stima a priori per $\hat{P}(a_i|v_j)$
- m è il peso dato a priori (cioè il numero di esempi "virtuali")

Esempio di applicazione: classifi cazione di documenti testuali

- apprendere quali documenti sono di interesse
- apprendere a classificare pagine web per argomento
- ...

Il classificatore Naive di Bayes costituisce una delle tecniche più utilizzate in questi contesti

Quali attributi usare per rappresentare documenti testuali ?

Classifi cazione di documenti testuali

Concetto target $Interessante? : Documento \rightarrow \{+, -\}$

1. Rappresentare ogni documento tramite un vettore di parole
   - Un attributo per posizione di parola nel documento
2. Apprendimento: usare gli esempi di apprendimento per stimare
   - $P(+), P(-), P(doc|+), P(doc|-)$

Assunzione di indipendenza condizionale di Naive Bayes

$$P(doc|v_j) = \prod_{i=1}^{lunghezza(doc)} P(a_i = w_k|v_j)$$

dove $P(a_i = w_k|v_j)$ è la probabilità che la parola in posizione i sia w_k, dato v_j

Una assunzione addizionale: $P(a_i = w_k|v_j) = P(a_m = w_k|v_j), \forall i, m$

LEARN_NAIVE_BAYES_TEXT($Esempi, V$)

1. collezionare tutte le parole ed altri token che occorrono in $Esempi$

- $Vocabolario \leftarrow$ tutte le parole distinte ed altri token in $Esempi$

2. calcolare (stimare) i termini $P(v_j)$ e $P(w_k|v_j)$

- **for each** valore di target v_j in V **do**
  - $doc_j \leftarrow$ sottoinsieme di $Esempi$ per cui il valore di target è v_j
  - $P(v_j) \leftarrow \frac{|doc_j|}{|Esempi|}$
  - $Text_j \leftarrow$ un unico documento creato concatenando tutti i documenti in doc_j
  - $n \leftarrow$ numero di parole e token totali in $Text_j$ (contando parole e token duplicati più volte)
  - **for each** parola e token w_k in $Vocabolario$
    - $n_k \leftarrow$ numero di volte che w_k occorre in $Text_j$
    - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabolario|}$

CLASSIFY_NAIVE_BAYES_TEXT(doc)

- $posizioni \leftarrow$ tutte le posizioni in doc che contengono token trovati nel $Vocabolario$
- Restituisci v_{NB}, dove

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i \in posizioni} P(a_i|v_j)$$

Reti Bayesiane: richiamo

Perchè sono interessanti:

- Naive Bayes usa una assunzione di indipendenza condizionale troppo restrittiva
- ...ma se non si usa una assunzione di tale tipo il problema è intrattabile...
- le Reti Bayesiane descrivono l'indipendenza condizionale tra *sottoinsiemi* di variabili

$\rightarrow$ permettono di combinare conoscenza a priori sulla (in)dipendenza fra variabili con dati osservati (esempi di apprendimento)

Indipendenza Condizionale: richiamo

Definizione: X è *condizionalmente indipendente* da Y dato Z se la distribuzione di probabilità che governa X è indipendente dal valore di Y dato il valore di Z; cioè, se

$$(\forall x_i, y_j, z_k)\ P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

in modo compatto, scriveremo $P(X|Y, Z) = P(X|Z)$

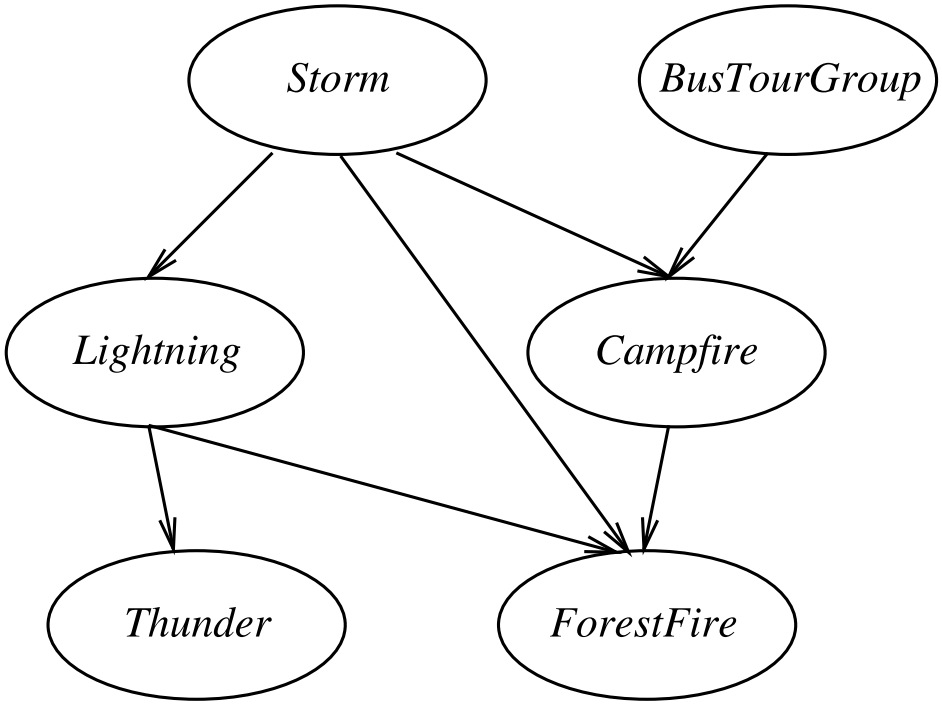
Esempio: $Fulmine$ è condizionalmente indipendente da $Pioggia$, dato $Lampo$

$$P(Fulmine|Pioggia, Lampo) = P(Fulmine|Lampo)$$

Naive Bayes usa l'indipendenza condizionale per giustifi care

$$\begin{aligned} P(X, Y|Z) &= P(X|Y, Z)P(Y|Z) \\ &= P(X|Z)P(Y|Z) \end{aligned}$$

Esempio di Rete Bayesiana

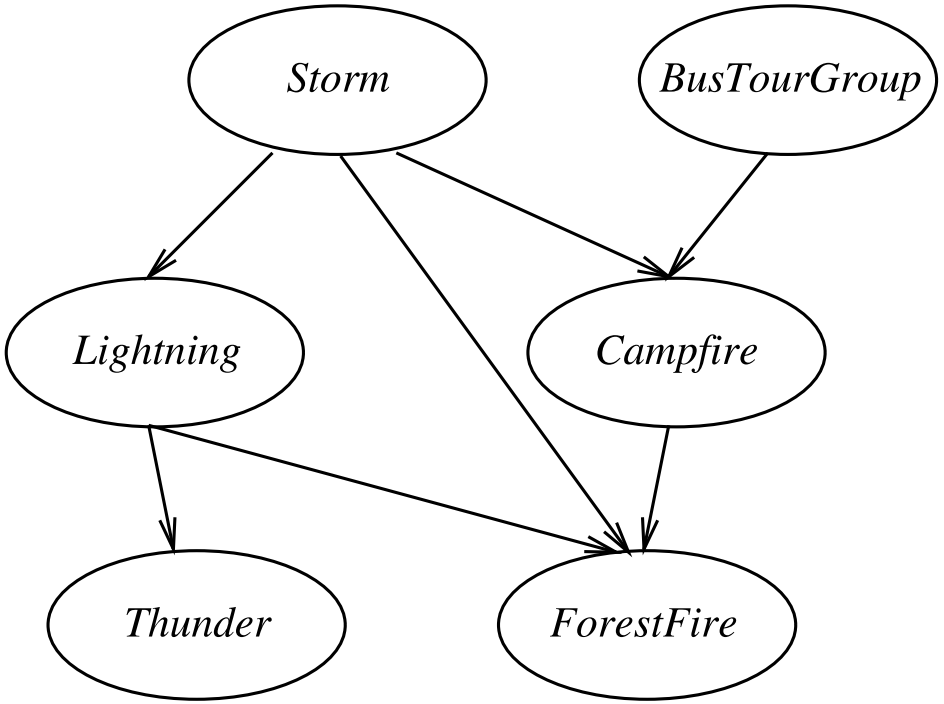


	S,B	$S,\neg B$	$\neg S,B$	$\neg S,\neg B$
C	0.4	0.1	0.8	0.2
$\neg C$	0.6	0.9	0.2	0.8

la rete rappresenta un insieme di asserzioni di indipendenza condizionale:

- ogni nodo è asserito essere condizionalmente indipendente dai suoi non-discendenti, dati i suoi genitori
- grafo diretto aciclico

Reti Bayesiane



	S,B	$S,\neg B$	$\neg S,B$	$\neg S,\neg B$
C	0.4	0.1	0.8	0.2
$\neg C$	0.6	0.9	0.2	0.8

rappresenta $P(Storm, BusTourGroup, \ldots, ForestFire)$

- in generale, $P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i | Genitori(Y_i))$
- quindi, la distribuzione congiunta è totalmente specifi cata dal grafo e dalle $P(y_i | Genitori(Y_i))$

Inferenza nelle Reti Bayesiane

Come inferire la distribuzione di probabilità sui valori che una o più variabili possono assumere, dati alcuni valori osservati per altre variabili ?

- Reti Bayesiane contengono tutta l'informazione per esguire tale inferenza (rappresentano la probabilità congiunta)
- Se si ha una sola variabile con valore sconosciuto, è facile rispondere
- Nel caso più generale, l'inferenza è un problema NP arduo

In pratica

- metodi esatti di inferenza polinomiali se la rete è un poli-albero
- metodi Monte Carlo "simulano" stocasticamente la rete per calcolare soluzioni approssimate