Ragionamento Bayesiano

Caratteristiche dell'apprendimento bayesiano

- Ogni esempio osservato nel training incrementa o decrementa la fiducia di ipotesi corretta (non si eliminano completamente le ipotesi inconsistenti con gli esempi, maggiore flessibilità).
- Combinazione di osservazioni e conoscenza a priori.
- Possibilità di ottenere predizioni probabilistiche.
- Classificazione combinando predizioni di più ipotesi pesate con le rispettive probabilità.
- Anche quando computazionalmente intrattabili, i metodi bayesiani forniscono un riferimento di decisione ottima con cui confrontare altri metodi pratici.
- Bisogna conoscere molte probabilità, spesso stimate a partire da conoscenza di fondo, dati già disponibili, assunzioni sulla forma delle distribuzioni.

Cenni di teoria della probabilità

- Probabilità di un evento A:

$$0 \leq P(A) \leq 1$$

- Regola del prodotto:

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

- Probabilità condizionata:

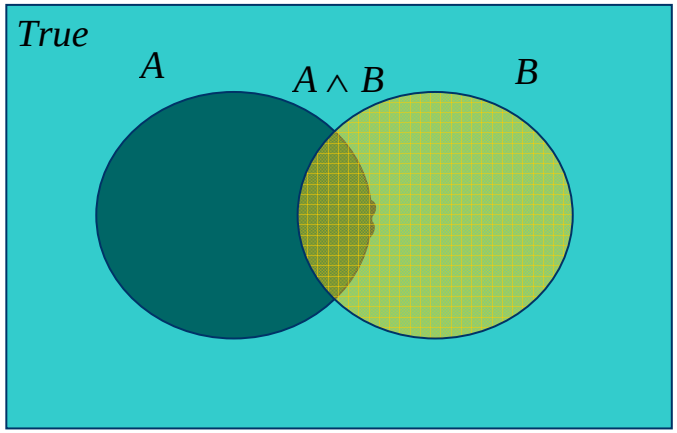
$$P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

- Regola della somma:

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

Cenni di teoria della probabilità (2)

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$



altrimenti sarebbe:

$$P(A \vee B) = P(A \wedge \neg B) + P(\neg A \wedge B) + P(A \wedge B)$$

Cenni di teoria della probabilità (3)

- Indipendenza tra eventi:

$$P(A \wedge B) = P(A)P(B)$$

- Indipendenza condizionale:

$$P(A \wedge B \mid C) = P(A \mid C)P(B \mid C)$$

altrimenti per la regola del prodotto sarebbe:

$$P(A \wedge B \mid C) = P(A \mid C)P(B \mid A \wedge C)$$

Cenni di teoria della probabilità (4)

- Teorema della probabilità totale: se A_1, …, A_n sono mutuamente esclusivi con $\sum_{i=1}^{n} P(A_i) = 1$ allora:

$$P(B) = \sum_{i=1}^{n} P(B \mid A_i) P(A_i)$$

Teorema di Bayes

- La migliore ipotesi h appartenente allo spazio H, date le osservazioni D e una conoscenza a priori circa la probabilità delle singole h, può essere considerata come l'ipotesi più probabile, ovvero quella che massimizza la quantità P(h|D).
- Il teorema di Bayes ci permette di calcolare tale probabilità:

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

- P(h) probabilità a priori dell'ipotesi h.
- P(D|h) probabilità di osservare i dati D quando l'ipotesi corretta è h.
- P(D) probabilità di osservare i dati D.

Ipotesi maximun a posteriori (MAP)

- Date le ipotesi h∈H e i dati osservati D, usando il teorema di Bayes si trova l'ipotesi che massimizza la probabilità a posteriori:

$$h_{MAP} \equiv \arg\max_{h \in H} P(h \mid D)$$

$$= \arg\max_{h \in H} \frac{P(D \mid h) P(h)}{P(D)}$$

$$= \arg\max_{h \in H} P(D \mid h) P(h)$$

dove si è omesso il termine P(D) perché è una costante indipendente da h.

Ipotesi maximum likelihood (ML)

- Nei casi in cui si può assumere che tutte le h sono equiprobabili, si massimizza P(D|h) (*likelihood* o verosimiglianza) e si trova l'ipotesi h_{ML} a massima verosimiglianza:

$$h_{ML} = \arg\max_{h \in H} P(D \mid h)$$

se $P(h_i) = P(h_j)$ per tutti le h_i e h_j in H.

Esempio: diagnosi medica

- Due ipotesi:
  - il paziente ha una particolare forma di cancro
  - il paziente non ha il cancro
- Evidenze (dati) fornite da un test di laboratorio (imperfetto):
  - $\oplus$ (positivo)
  - $\varnothing$ (negativo)
- Conoscenza a priori: nell'intera popolazione, 8 persone su 1000 hanno quel tipo di cancro
- Caratteristiche del test:
  - fornisce un corretto positivo solo nel 98% dei casi in cui la malattia è presente
  - fornisce un corretto negativo solo nel 97% dei casi in cui il male è assente

Esempio: diagnosi medica (2)

- La situazione è riassunta dalle seguenti probabilità:

  $P(\text{cancro}) = 0.008$ $\quad P(\neg\text{cancro}) = 0.992$

  $P(\oplus|\text{cancro}) = 0.98$ $\quad P(\varnothing|\text{cancro}) = 0.02$

  $P(\oplus|\neg\text{cancro}) = 0.03$ $\quad P(\varnothing|\neg\text{cancro}) = 0.97$

- Data la positività di un paziente al test, quale diagnosi?

Esempio: diagnosi medica (3)

- Applico l'algoritmo di maximum a posteriori hypothesis, calcolando:
  - $P(cancro|\oplus) = P(\oplus|cancro)\cdot P(cancro) = 0.98 \cdot 0.008 = 0.0078$
  - $P(\neg cancro|\oplus) = P(\oplus|\neg\, cancro)\cdot P(\neg\, cancro) = 0.03 \cdot 0.992 = 0.0298$
- Pertanto, $h_{MAP} = \neg cancro$.
- L'esatta probabilità a posteriori si determina normalizzando le due quantità precedenti in modo che la somma dia 1:

$$P(\neg cancro \mid \oplus) = \frac{0.0298}{0.0078 + 0.0298} = 0.79$$

Apprendimento di concetti a forza bruta

- Spazio delle ipotesi H definito su X.
- Il task è imparare un concetto obiettivo c : X $\rightarrow$ 0, 1.
- Insieme di addestramento <<x_1, d_1>, … <x_m, d_m>>, in cui x_i è un'istanza di X e $d_i = c(x_i)$.
- Algoritmo di apprendimento MAP a forza bruta:
  - per ogni h $\in$H calcolo $P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$
  - restituisco l'ipotesi h_{MAP}.
- Computazionalmente dispendioso: teorema di Bayes applicato per ogni h, ma fornisce uno standard con cui confrontarsi.

Apprendimento di concetti a forza bruta (2)

- Per specificare il problema di apprendimento dell'algoritmo BRUTE-FORCE MAP LEARNING, occorre specificare i valori usati per P(h) e P(D|h) (da cui si può calcolare P(D)).
- Facciamo tre assunzioni:
  - D è senza rumore, ovvero $d_i = c(x_i)$;
  - il concetto target c è contenuto nello spazio delle ipotesi H;
  - non abbiamo una ragione a priori per considerare un'ipotesi più probabile di un'altra.
- La 2ª e 3ª assunzione comportano che $P(h) = \frac{1}{|H|}$ per tutti gli h $\in$ H.
- La 1ª assunzione (corrispondente a "è dato un mondo in cui h è la descrizione corretta del concetto target c") comporta che:

$$P(D|h) = \begin{cases} 1 & \text{se } d_i = h(x_i) \text{ per tutti i } d_i \in D \\ 0 & \text{altrimenti} \end{cases}$$

Apprendimento di concetti a forza bruta (3)

- Applicando il teorema di Bayes, si desume che:
  - se h è inconsistente con D,

$$P(h|D) = \frac{0 \cdot P(h)}{P(D)} = 0$$

  - se h è consistente con D,

$$P(h|D) = \frac{1 \cdot \frac{1}{|H|}}{P(D)} = \frac{1 \cdot \frac{1}{|H|}}{\frac{|VS_{H,D}|}{|H|}} = \frac{1}{|VS_{H,D}|}$$

dove $VS_{H,D}$ è il sottoinsieme di ipotesi di H che è consistente con D.

Apprendimento di concetti a forza bruta (4)

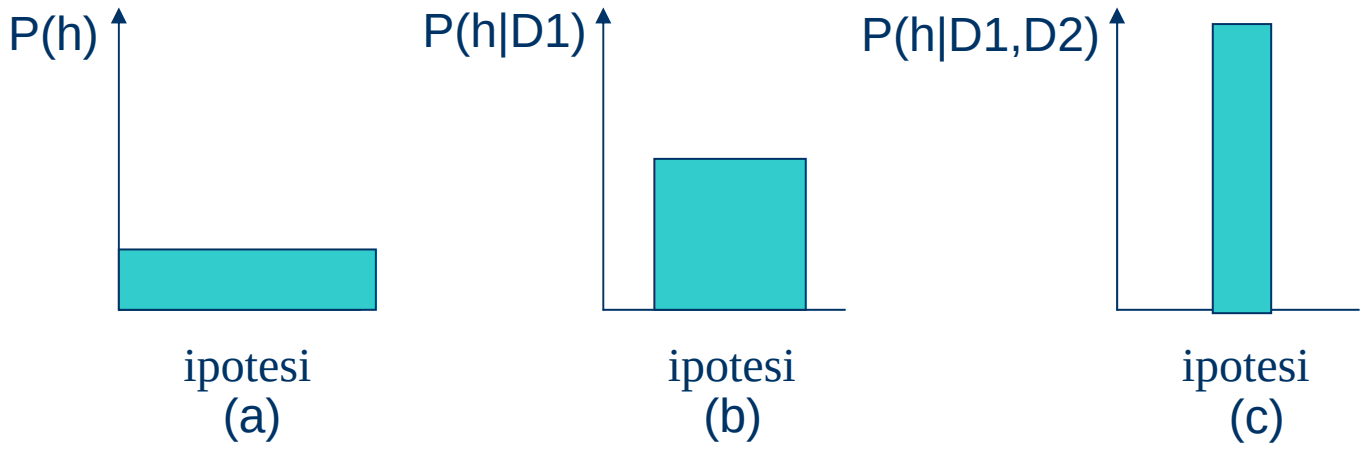
- Nota: $P(D) = \frac{|VS_{H,D}|}{|H|}$ perché la somma su tutte le ipotesi di P(D) deve essere 1 e perché il numero di ipotesi di H consistenti con D è per definizione $|VS_{H,D}|$, ovvero si può calcolare, tenendo conto che le ipotesi sono mutuamente esclusive (cioè, $(\forall i \neq j)(P(h_i \wedge h_j)=0)$), così:

$$P(D) = \sum_{h_i \in H} P(D \mid h_i) P(h_i) = \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|} + \sum_{h_i \notin VS_{H,D}} 0 \cdot \frac{1}{|H|} =$$

$$= \sum_{h_i \in VS_{H,D}} 1 \cdot \frac{1}{|H|} = \frac{|VS_{H,D}|}{|H|}$$

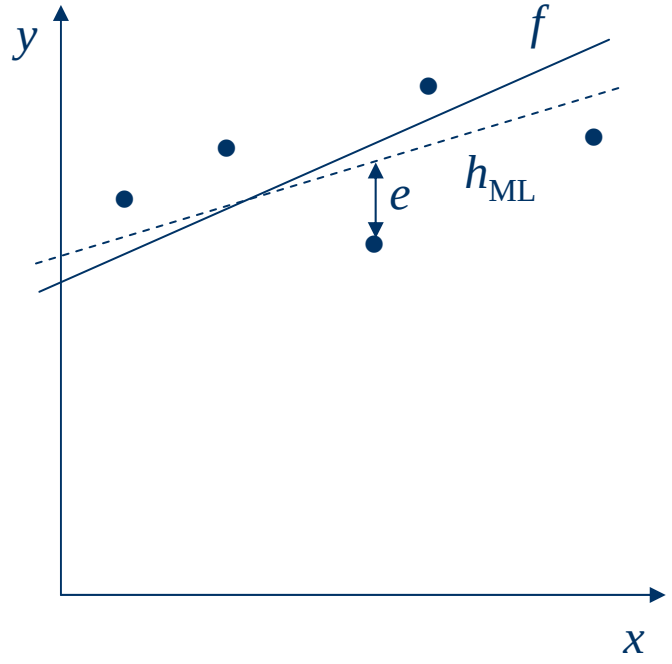
Esempio dell'evoluzione delle probabilità associate a delle ipotesi

- Inizialmente tutte le ipotesi hanno la stessa probabilità (fig. a).
- Man mano si accumulano i dati di training (fig. b e fig. c), la probabilità a posteriori per le ipotesi inconsistenti diventa zero mentre la probabilità totale (la cui somma = 1) si suddivide sulle restanti ipotesi consistenti.



Apprendimento di una Funzione a Valori Reali

Sotto certe assunzioni un algoritmo di learning che minimizza l'errore quadratico tra l'output delle predizioni delle ipotesi e i dati di training produce un'ipotesi a massima verosimiglianza. Il caso di una funzione a valori reali:



Apprendimento di una Funzione a Valori Reali (2)

- Si consideri una qualunque funzione target f a valori reali, ed esempi di apprendimento $\langle x_i, d_i \rangle$, dove d_i è affetto da rumore:
  - $d_i = f(x_i) + e_i$
  - e_i è una variabile random (rumore) estratta indipendentemente 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 (3)

- Essendo i nostri valori nel continuo, occorre utilizzare, invece della probabilità, la funzione densità di probabilità, definita come:

$$p(x_0) = \lim_{\varepsilon \to 0} \frac{1}{\varepsilon} P(x_0 \leq x \leq x_0 + \varepsilon)$$

- La definizione di ipotesi a massima verosimiglianza

$$h_{ML} = \arg\max_{h \in H} P(D \mid h)$$

qui diventa

$$h_{ML} = \arg\max_{h \in H} p(D \mid h)$$

Apprendimento di una Funzione a Valori Reali (4)

- Assumendo che gli esempi di training siano indipendenti data h, possiamo scrivere p(D|h) come il prodotto dei vari $p(d_i|h)$

$$h_{ML} = \arg\max_{h \in H} p(D \mid h) =$$

$$= \arg\max_{h \in H} \prod_{i=1}^{m} p(d_i \mid h)$$

Apprendimento di una Funzione a Valori Reali (5)

- Dato che il rumore e_i obbedisce a una distribuzione Normale con media zero e varianza σ^2, ciascun d_i deve anche obbedire ad una distribuzione Normale con varianza σ^2 centrata intorno al valore target $f(x_i)$.
- Pertanto $p(d_i|h)$ può essere scritta come una distribuzione Normale con varianza σ^2 e media $\mu = f(x_i)$.
- Poiché stiamo scrivendo l'espressione per la probabilità di d_i dato che h è la corretta descrizione della funzione target f, possiamo effettuare la sostituzione $\mu = f(x_i) = h(x_i)$.

Apprendimento di una Funzione a Valori Reali (6)

$$h_{ML} = \arg\max_{h \in H} p(D \mid h) =$$

$$= \arg\max_{h \in H} \prod_{i=1}^{m} p(d_i \mid h) =$$

$$= \arg\max_{h \in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2}$$

che si tratta meglio massimizzando il logaritmo naturale…

Apprendimento di una Funzione a Valori Reali (7)

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

$$= \arg\max_{h \in H} \sum_{i=1}^{m} -\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2 =$$

$$= \arg\max_{h \in H} \sum_{i=1}^{m} -\left(d_i - h(x_i)\right)^2 =$$

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

Classificatore Bayesiano Ottimo

- Invece di chiederci qual è l'*ipotesi* più probabile dato l'insieme di addestramento, ci chiediamo qual è la più probabile *classificazione* di una nuova istanza dato l'insieme di addestramento.
- È possibile fare meglio che semplicemente applicare l'ipotesi h_{MAP} alla nuova istanza.
- Esempio:
  - consideriamo uno spazio delle ipotesi che contenga tre ipotesi h_1, h_2 e h_3;
  - supponiamo che $P(h_1|D) = 0.4$, $P(h_2|D) = 0.3$ e $P(h_3|D) = 0.3$ (pertanto, h_1 è l'ipotesi MAP);
  - ora una nuova istanza x sia classificata positiva da h_1 e negativa da h_2 e h_3;
  - considerando tutte le ipotesi, la probabilità che x sia positiva è 0.4 (probabilità associata ad h_1), e quindi la probabilità che sia negativa è 0.6.
  - la classificazione più probabile (X negativa) risulta diversa da quella prodotta da h_{MAP}.

Classificatore Bayesiano Ottimo (2)

- In generale, la *classificazione più probabile* è ottenuta combinando le predizioni di tutte le ipotesi pesate con le loro probabilità a posteriori.
- Se la possibile classificazione di un nuovo esempio può assumere un qualche valore v_j da un qualche insieme V, allora la probabilità $P(v_j|D)$ che la corretta classificazione per la nuova istanza sia v_j è appunto

$$P(v_j \mid D) = \sum_{h_i \in H} P(v_j \mid h_i) P(h_i \mid D)$$

- L'ottima classificazione della nuova istanza è il valore v_j per cui $P(v_j|D)$ è massimo.
- La Classificazione Bayesiana Ottima risulta quindi essere:

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

Classificatore Bayesiano ottimo: esempio

$V = \{\oplus, \otimes\}$

$P(h_1 | D) = 0.4,\ P(\otimes | h_1) = 0,\ P(\oplus | h_1) = 1$

$P(h_2 | D) = 0.3,\ P(\otimes | h_2) = 1,\ P(\oplus | h_2) = 0$

$P(h_3 | D) = 0.4,\ P(\otimes | h_3) = 1,\ P(\oplus | h_3) = 0$

$$\sum_{h_i \in H} P(\oplus | h_i) P(h_i | D) = 0.4$$

$$\sum_{h_i \in H} P(\otimes | h_i) P(h_i | D) = 0.6$$

$$\arg\max_{v_j \in \{\oplus, \otimes\}} \sum_{h_j \in H} P(v_j | h_i) P(h_i | D) = \otimes$$

Algoritmo di Gibbs

- Limite del classificatore bayesiano ottimo: computazionalmente oneroso. Richiede di calcolare la probabilità a posteriori per ogni ipotesi in H e di combinare le predizioni di ciascuna ipotesi per classificare ciascuna nuova istanza.
- Alternativa: algoritmo di Gibbs, definito come segue:
  1. Scegliere un'ipotesi h da H in modo *random*, in accordo con la distribuzione di probabilità a posteriori su H;
  2. Usare h per predire la classificazione della prossima istanza x.

  Sorprendentemente, si dimostra che sotto certe condizioni l'errore atteso di misclassificazione è al più il doppio dell'errore atteso del classificatore ottimo bayesiano.

Classificatore Naive Bayes

- Si applica quando ciascuna istanza x è descritta da una congiunzione di valori attributo $\langle a_1, a_2,\ldots,a_n\rangle$ e quando la funzione target f(x) può assumere un qualsiasi valore da un insieme finito V.
- È disponibile un set di esempi di training della funzione target ed viene presentata una nuova istanza, descritta dalla tupla di valori attributo $\langle a_1, a_2,\ldots,a_n\rangle$. Occorre predire il valore target, ovvero la classificazione, per questa nuova istanza.
- L'approccio Bayesiano di classificare la nuova istanza è assegnare il valore target più probabile, v_{MAP}, dati i valori attributo che descrivono l'istanza:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2, \ldots, a_n)$$

$$= \arg\max_{v_j \in V} \frac{P(a_1, a_2, \ldots, a_n \mid v_j) P(v_j)}{P(a_1, a_2, \ldots, a_n)}$$

$$= \arg\max_{v_j \in V} P(a_1, a_2, \ldots, a_n \mid v_j) P(v_j)$$

Classificatore Naive Bayes (2)

- Nota: in $v_{MAP} = \arg\max_{v_j \in V} P(a_1, a_2, ..., a_n \mid v_j) P(v_j)$

- è facile stimare $P(v_i)$: si contano le frequenze con cui appare ogni valore target v_i nel training set.
- È complicato valutare $P(a_1, a_2, \ldots, a_n \mid v_i)$, a meno di disporre di un data set estremamente ampio per il training: il numero di termini è uguale al numero di possibili istanze moltiplicato il numero di possibili valori obiettivo!

Classificatore Naive Bayes (3)

- Il Classificatore Naive Bayes introduce una semplificazione assumendo che, dato il valore obiettivo, i valori dei singoli attributi siano condizionalmente indipendenti. Ciò comporta che la probabilità

$$P(a_1, a_2, \ldots, a_n \mid v_j) = P(a_1 \mid v_j) P(a_2 \mid v_j, a_1) \ldots P(a_n \mid v_j, a_1, a_2, \ldots, a_{n-1})$$

venga calcolata nel seguente modo

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

Il classificatore così ottenuto è il seguente:

$$v_{NB} = \arg\max_{v_j} P(v_j) \prod_i P(a_i \mid v_j)$$

Esempio illustrativo

- Classificazione dei giorni in base a quando qualcuno gioca a tennis.
- Data training: tabella che segue.
- Qui vogliamo usare un classificatore naive Bayes e la tabella di data training per classificare la seguente nuova istanza:

  (*Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong*)

data training

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

Esempio illustrativo (2)

- Il task consiste nel predire il valore target (*yes* or *no*) del concetto target *PlayTennis* per la nuova istanza. Applicando la formula precedente:

$$v_{NB} = \arg\max_{v_j \in \{yes,no\}} P(v_j) \prod_i P(a_i \mid v_j)$$

$$= \arg\max_{v_j \in \{yes,no\}} P(v_j) \cdot P(Outlook = sunny \mid v_j) \cdot P(Temperature = cool \mid v_j) \cdot$$

$$\cdot P(Humidity = high \mid v_j) \cdot P(Wind = strong \mid v_j)$$

i valori di probabilità si desumono dalla tabella.

Esempio illustrativo (3)

- Dai 14 esempi si desume che:

  P(*PlayTennis* = *yes*) = 9/14 = 0.64

  P(*PlayTennis* = *no*) = 5/14 = 0.36

  e così:

  P(*Wind* = *strong*|*PlayTennis* = *yes*) = 3/9 = 0.33

  P(*Wind* = *strong*|*PlayTennis* = *no*) = 3/5 = 0.60

  ecc.

Esempio illustrativo (4)

- Infine:

  P(*yes*)·P(*sunny*|*yes*)·P(*cool*|*yes*)·P(*high*|*yes*)·P(*strong*|*yes*) = 0.0053

  P(*no*)·P(*sunny*|*no*)·P(*cool*|*no*)·P(*high*|*no*)·P(*strong*|*no*) = 0.0206

  si conclude *PlayTennis* = *no.*

  La probabilità condizionata che la il valore target sia *no* si ottiene per normalizzazione: 0.0206/(0.0206+0.0053)=0.795

Naive Bayes: considerazioni aggiuntive

- L'assunzione di indipendenza condizionata è spesso violata

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

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

$$\arg\max_{v_j \in V} \hat{P}(v_j) \prod \hat{P}(a_i \mid v_j) = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i, a_2, \ldots a_n \mid v_j)$$

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

La stima delle probabilità: lo smoothing

- Usualmente si valuta la probabilità di un evento come n_c/n, dove n_c è il numero di occorrenze dell'evento, n il numero totale di prove.
- Se la base di dati di training è scarsa, n_c può essere piccolo o addirittura nullo, portando a stime errate nelle formule precedenti.
- In questi casi si usa la seguente definizione di probabilità (m-estimate of probability):

$$\frac{n_c + m \cdot p}{n + m}$$

La stima delle probabilità (2)

- n_c e n sono quelli definiti prima, p è la nostra stima a priori della probabilità che vogliamo stimare, m è una costante detta *equivalent sample size*, stabilisce quanto fortemente pesare p in relazione ai dati osservati.

- Un metodo per scegliere p è, in mancanza di informazioni, assumere la distribuzione uniforme. Ad esempio, se un attributo può assumere k valori, si pone p=1/k. Nell'esempio precedente, *Wind* assume due valori, si può porre p=0.5.

- La costante m "allarga" il numero di prove, immaginando campioni virtuali. Se m=0, si ricade nella definizione canonica.

La stima delle probabilità (3)

- Solitamente, si pone m pari al numero di valori assunti (k nel caso precedente: è come assumere che ogni valore è apparso 1 volta prima di cominciare a contare). In questo caso, m·p=1 (detto anche *add-one smoothing*, caso particolare del *Laplace smoothing*).

- Alternativa: Linear interpolation:
  - Si stima P(X) dai dati
  - Ci si assicura che la stima di P(X|Y) non sia troppo differente da P(X):

$$P_{LIN}(x \mid y) = \alpha \hat{P}(x \mid y) + (1.0 - \alpha)\hat{P}(x)$$

Cosa succede se α vale 0? E se vale 1?

Applicazione: classificazione di documenti testuali

- Scopo:
  - apprendere concetti target come “documenti di interesse su un certo argomento”
  - apprendere a classificare pagine web per argomento
  - …
- Il classificatore Naive Bayes costituisce una delle tecniche più utilizzate in questi contesti.

- Bisogna stabilire il setting del problema.

Classificazione di documenti testuali (2)

- Lo spazio delle istanze X consiste in tutti i possibili documenti di testo (presi così come sono, con punteggiatura, differente lunghezza, ecc.).
- Sono dati degli esempi di training per una funzione target incognita f(x), che può assumere valori in un insieme finito V.
- Il task è apprendere da questi esempi di training per predire il valore target per successivi documenti di testo.
- Qui la funzione tagert è la definizione di un documento come *interessante* o *non-interessante* per un soggetto, e i valori target sono *like* e *dislike*.

Classificazione di documenti testuali (3)

- Problemi connessi:
  - come rappresentare un arbitrario documento di testo in termini di valori attributo;
  - come stimare le probabilità richieste dal classificatore Naïve Bayes.
- Riguardo al primo, si sceglie di rappresentare ogni documento come un vettore di parole, definendo
  - un attributo per ogni posizione di parola nel documento,
  - come valore di quell'attributo la parola (ad es. in Inglese) trovata in quella posizione.

Classificazione di documenti testuali (4)

- Riguardo al secondo, bisogna disporre di un sufficiente numero di testi di training classificati come *like* o come *dislike* (ad es., 300 delle prime e 700 delle seconde).
- Se occorre classificare un nuovo documento, si applica la classificazione naïve Bayes, che massimizza la probabilità di osservarele le parole effettivamente trovate nel documento, con la solita assunzione di indipendenza condizionale.

Classificazione di documenti testuali (5)

- Se ad esempio il documento di testo è la slide precedente:

$$v_{NB} = \underset{v_j \in \{like, dislike\}}{arg\,max}\ P(v_j) \prod_{i=1}^{59} P(a_i \mid v_j) =$$

$$= \underset{v_j \in \{like, dislike\}}{arg\,max}\ P(v_j) P(a_1 = "riguardo" \mid v_j) P(a_2 = "al" \mid v_j) \ldots$$

$$\ldots P(a_{59} = "condizionale" \mid v_j)$$

Classificazione di documenti testuali (6)

- Nota: in realtà l'indipendenza delle parole non è consistente (es.: dopo naïve c'è sempre Bayes), ma il classificatore sorprendentemente funziona!
- Per il calcolo di v_{NB}, occorre stimare $P(v_j)$ e $P(a_i=w_k|v_j)$.
- La prima si stima facilemente dalla classificazione dei testi di training: nel nostro esempio, P(*like*) = 0.3, P(*dislike*) = 0.7.
- La seconda è problematica: nel nostro esempio, ci sono 2 valori target, 59 posizioni, il numero di parole si può stimare dell'ordine di 50.000. Bisognerebbe stimare $2 \cdot 59 \cdot 50000 \approx 5$ milione di termini!

Classificazione di documenti testuali (7)

- Si semplifica il problema assumendo che $P(a_i = w_k|v_j) = P(a_m = w_k|v_j)$ per tutti gli *i*, *j*, *k*, *m*. Il numero di termini da stimare, nella forma di $P(w_k|v_j)$, si riduce a $2 \cdot 50000$: ancora grande ma trattabile.

- Per il calcolo delle probabilità $P(w_k|v_j)$, conviene utilizzare lo smoothing:

$$\frac{n_k + 1}{n + |Vocabolary|}$$

dove n è il numero totale di posizione parole in tutti gli esempi di training il cui valore target è v_j, n_k è il numero di volte che si trova la parola w_k nelle n posizioni parola, |Vocabolary| è il numero totale di parole differenti (e altri token) nei dati di training.

Classificazione di documenti testuali: l'algoritmo

LEARN_NAIVE_BAYES_TEXT(Examples, V)

Examples is a set of text documents along with their target values. V is the set of all possible target values. This function learns the probability terms $P(w_k|v_j)$, describing the probability that a randomly drawn word from a document in class v_j will be the English word w_k. It also learns the class prior probabilities $P(v_j)$.

1. *Collect all words, punctuation, and other tokens that occur in Examples*
   - *Vocabolary* ← the set of all distinct words and other tokens occurring in any text document from *Examples*

Classificazione di documenti testuali: l'algoritmo (2)

2. *Calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms*
   - For each target value v_j in V do
     - $docs_j \leftarrow$ the subset of documents from *Examples* for wich the target value is v_j
     - $P(v_j) \leftarrow |docs_j| / |Examples|$
     - $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$
     - $n \leftarrow$ total number of distinct word position in $Text_j$
     - For each word w_k occurs in *Vocabulary*
       - $n_k \leftarrow$ number of times word w_k occurs in $Text_j$
       - $P(w_k|v_j) \leftarrow (n_k+1) / (n+|Vocabulary|)$

Classificazione di documenti testuali: l'algoritmo (3)

CLASSIFY_NAIVE_BAYES_TEXT(*Doc*)

Return the estimated target value for the document Doc. a_i *denotes the word found in the ith position within Doc.*

- *Positions* ← all word positions in *Doc* that contain tokens found in *Vocabulary*
- Return v_{BN}, where

$$v_{NB} = \underset{v_j \in V}{arg\,max}\, P(v_j) \prod_{i \in positions} P(a_i \mid v_j)$$

Junk Mail Filtering

- Le cosiddette *junk mail*, dette anche *spam*, hanno sempre creato numerosi problemi:
  - perdita di tempo da parte degli utenti;
  - contenuti sconvenienti;
  - occupazione di spazio su disco nei server;
  - ecc.
- Metodi per filtrare automaticamente questo tipo di mail sono presto diventati necessari.
- Le prime tecniche erano *rule-based*:
  - rigide e poco robuste;
  - non in grado di fornire un livello di confidenza;
  - non prevedono un modello di utilità (il costo di misclassificare un messaggio legittimo come spam è molto più alto del costo di classificare come legittimo un messaggio spazzatura).

Junk Mail Filtering (2)

- Un approccio di tipo probabilistico permette di superare questi limiti.
- Il classificatore Naive Bayes fornisce un buon compromesso tra complessità ed efficacia.
- L’estensibilità del modello Bayesiano permette inoltre di utilizzare elementi specifici del dominio in questione, andando oltre la categorizzazione del testo classica.

Junk Mail Filtering: l'approccio Bayesiano

- Il modello Bayesiano può essere applicato facilmente al problema dei messaggi email rappresentando questi ultimi mediante dei *vettori di attributi* o *feature vector*.
- Le singole dimensioni di tale vettore sono le parole osservate nel *corpus* di email di addestramento.
- Ogni email è quindi rappresentata da un *vettore binario* che indica la presenza o meno nel messaggio di ogni parola.
- È possibile aggiungere ulteriori attributi relativi al problema, quali la presenza di frasi come "FREE!", "only $", "be over 21", "$$$ BIG MONEY $$$", il dominio dell'indirizzo del mittente, la percentuale di caratteri non alfanumerici, ecc.

Junk Mail Filtering: selezione degli attributi

- Si desidera selezionare gli attributi *più significativi* per la classificazione:
  - riduzione della dimensionalità del modello;
  - attenuazione degli effetti dell'assunzione *naive*.
- Eliminazione delle parole con frequenza più alta (articoli, preposizioni, ecc.)
- Eliminazione delle parole che compaiono meno di tre volte.
- Calcolo della mutua informazione tra gli attributi A_i e la classe V:

$$I(A_i, V) = \sum_{a_j \in A_i, v_j \in V} P(A_i, V) \log \frac{P(A_i, V)}{P(A_i) P(V)}$$

- Scelta dei 500 attributi con mutua informazione più alta.

Junk Mail Filtering: classificazione

- V = {*junk*, ¬*junk*}
- Il classificatore è dunque il seguente:

$$\arg\max_{v_j \in V} P(a_1, a_2, ..., a_n \mid v_j) P(v_j)$$

dove

$$P(a_1, a_2, ..., a_n \mid v_j) = P(a_1 \mid v_j) P(a_2 \mid v_j) ... P(a_n \mid v_j) = \prod_i P(a_i \mid v_j)$$

quindi

$$v_{NB} = \arg\max_{v_j} P(v_j) \prod_i P(a_i \mid v_j)$$

Junk Mail Filtering: risultati

- Come già detto, si ha l'opportunità di fare una classificazione *cost sensitive*, per questo una email viene classificata *junk* solo se tale probabilità è maggiore del 99.9%

Attributi considerati	Junk		Legittime	
	Precision	Recall	Precision	Recall
Solo parole	97.1%	94.3%	87.7%	93.4%
Parole+ Frasi	97.6%	94.3%	87.8%	94.7%
Parole + Frasi + Spec.dom.	100.0%	98.3%	96.2%	100.0%

Definizione

$$\mathrm{Pr}\,ecisione = \frac{|\{Documenti\_attinenti\} \cap \{Documenti\_recuperati\}|}{|\{Documenti\_recuperati\}|}$$

$$\mathrm{Re}\,call = \frac{|\{Documenti\_attinenti\} \cap \{Documenti\_recuperati\}|}{|\{Documenti\_attinenti\}|}$$