

# Apprendimento Automatico

## Metodi Bayesiani - Naive Bayes



Fabio Aioli

10 Dicembre 2018



Una delle tecniche più semplici e popolari

Quando usarlo:

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

Applicazione su cui ha avuto successo:

- Diagnosi
- Classificazione di documenti testuali



# Classificatore Naive Bayes

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

Il valore più probabile di  $f(x)$  è:

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

Assunzione Naive Bayes:

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



Mettendo tutto assieme:

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

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

Classificatore Naive Bayes:

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



# Algoritmo Naive Bayes

## Naive\_Bayes\_Learn(esempi)

- **For each** valore target  $v_j$ 
  - $\hat{P}(v_j) \leftarrow$  stima di  $P(v_j)$  su esempi
  - **For each** valore di attributo  $a_i$  di ogni attributo  $a$ ,  
 $\hat{P}(a_i|v_j) \leftarrow$  stima di  $P(a_i|v_j)$  su esempi
- **return**  $\hat{P}(v_j), \hat{P}(a_i|v_j) \forall i, j$

## Classify\_New\_Instance(x)

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

# Ri-giochiamo a tennis?



È una giornata adatta per una partita di 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



## Ri-giochiamo a tennis?

Consideriamo di nuovo il problema Giocare a Tennis già incontrato in precedenza

Consideriamo una nuova istanza:

$\langle 0 = \text{sunny}, T = \text{cool}, H = \text{high}, W = \text{strong} \rangle$

Vogliamo calcolare:

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

$$P(\text{yes})P(\text{sunny}|\text{y})P(\text{cool}|\text{yes})P(\text{high}|\text{yes})P(\text{strong}|\text{yes}) = 0.005$$

$$P(\text{no})P(\text{sunny}|\text{no})P(\text{cool}|\text{no})P(\text{high}|\text{no})P(\text{strong}|\text{no}) = 0.021$$

$$\rightarrow v_{NB} = \text{no}$$



## Naive Bayes: considerazioni aggiuntive

L'assunzione di indipendenza condizionale è spesso violata

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

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

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

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



## Naive Bayes: considerazioni aggiuntive

Cosa succede se nessun esempio di apprendimento con valore target  $v_j$  possiede attributo  $a_k$ ? In tal caso:

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

Una soluzione tipica è la *m-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  $v = v_j$
- $n_c$  è il numero di esempi di apprendimento per cui  $v = v_j$  e  $a = a_i$
- $p$  è la stima a priori per  $\hat{P}(a_i|v_j)$ , di solito  $1/|a_i|$
- $m$  è il peso dato a priori (cioè il numero di esempi "virtuali")

# Esempio di applicazione: Classificazione di documenti testuali



- apprendere quali documenti sono di interesse
- apprendere a classificare pagine web per argomento
- spam / no spam
- ...

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

Quali attributi usare per rappresentare documenti testuali?

# Esempio di applicazione: Classificazione di documenti testuali



Concetto target: *Interessante?* : *Documento*  $\rightarrow \{\oplus, \ominus\}$

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

Assunzione di indipendenza condizionale di Naive Bayes:

$$P(doc|v_j) = \prod_i^{lunghezza(doc)} P(a_i = w_k|v_j)$$

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

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

# Esempio di applicazione: Classificazione di documenti testuali



Occorre quindi stimare "solo" le  $P(v_j) \forall j$  e le  $P(w_k|v_j) \forall k, j$

Possiamo utilizzare una  $m$ -stima con priori uniforme e  $m$  uguale alla dimensione del vocabolario.

$$\hat{P}(w_k|v_j) = \frac{n_k + 1}{n + |\text{vocabolario}|}$$

dove

- $n$  è il numero totale di posizioni di parole in tutti i documenti di training aventi di classe  $v_j$
- $n_k$  è il numero di volte che la parola  $w_k$  si trova in queste posizioni
- $|\text{Vocabolario}|$  è il numero totale di parole distinte trovate nel training set

# Esempio di applicazione: Classificazione di documenti testuali



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*
  - *Vocabulary*  $\leftarrow$  the set of all distinct words and other tokens occurring in any text document from *Examples*
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<sub>j</sub>*  $\leftarrow$  the subset of documents from *Examples* for which the target value is  $v_j$
    - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
    - *Text<sub>j</sub>*  $\leftarrow$  a single document created by concatenating all members of *docs<sub>j</sub>*
    - $n \leftarrow$  total number of distinct word positions in *Text<sub>j</sub>*
    - for each word  $w_k$  in *Vocabulary*
      - $n_k \leftarrow$  number of times word  $w_k$  occurs in *Text<sub>j</sub>*
      - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

CLASSIFY\_NAIVE\_BAYES\_TEXT(*Doc*)

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

- *positions*  $\leftarrow$  all word positions in *Doc* that contain tokens found in *Vocabulary*
- Return  $v_{NB}$ , where

$$v_{NB} = \operatorname{argmax}_{v_j \in V} P(v_j) \prod_{i \in \text{positions}} P(a_i|v_j)$$

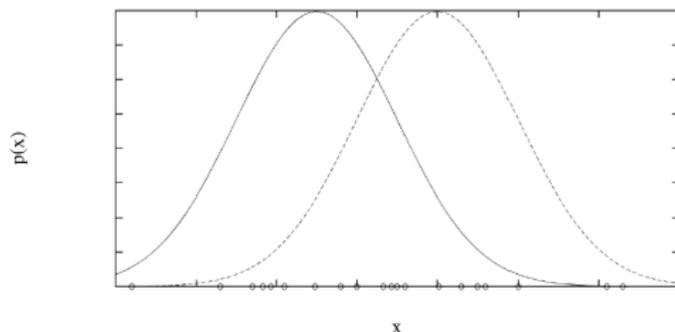


Quando utilizzarlo:

- dati solo parzialmente osservabili
- clustering non-supervisionato (valore target non osservabile)
- apprendimento supervisionato (alcuni attributi con valori mancanti)

Alcuni esempi:

- apprendimento Reti Bayesiane
- apprendimento di modelli di Markov nascosti (Hidden Markov Models)



Assumiamo che ogni istanza  $x$  sia generata:

- scegliendo una delle Gaussiane con probabilità uniforme
- generando una istanza a caso secondo la Gaussianina scelta



## EM per stimare k-medie

Date:

- istanza da  $X$  generate da una mistura di  $k$  Gaussiane
- medie  $\langle \mu_1, \dots, \mu_k \rangle$  sconosciute delle  $k$  Gaussiane (assumiamo  $\sigma^2$  conosciuto e uguale per tutte le Gaussiane).
- non si sa quale istanza  $x_i$  è stata generata da quale Gaussianiana

Determinare:

- stime maximum-likelihood di  $\langle \mu_1, \dots, \mu_k \rangle$

Ogni istanza può essere pensata nella forma  $y_i = \langle x, z_{i1}, z_{i2} \rangle$  (caso  $k = 2$ ), dove:

- $z_{ij}$  è 1 se l'esempio  $i$  è stato generato dalla Gaussianiana  $j$ , 0 altrimenti
- $x_i$  osservabile
- $z_{ij}$  non osservabili



## EM per stimare k-medie ( $k = 2$ )

Algoritmo EM: scegliere a caso l'ipotesi  $h = \langle \mu_1, \mu_2 \rangle$ , poi ripetere:

- **Passo E:** Calcola il valore atteso  $E[z_{ij}]$  di ogni variabile non osservabile  $z_{ij}$ , assumendo valga l'ipotesi corrente  $h$

$$\begin{aligned} E[z_{ij}] &= \frac{p(x = x_i | \mu = \mu_j)}{\sum_{n=1}^2 p(x = x_i | \mu = \mu_n)} \\ &= \frac{e^{-\frac{1}{\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^2 e^{-\frac{1}{\sigma^2}(x_i - \mu_n)^2}} \end{aligned}$$

- **Passo M:** Calcola la nuova ipotesi maximum-likelihood  $h$ , assumendo che il valore preso da ogni variabile non osservabile  $z_{ij}$  sia il suo valore atteso (calcolato sopra)

$$\pi_{ij} = \frac{E[z_{ij}]}{\sum_{i=1}^m E[z_{ij}]} \quad \text{and} \quad \mu_j \leftarrow \sum_{i=1}^m \pi_{ij} x_i$$



## Argomenti:

- Teorema di Bayes
- Ipotesi MAP, ML
- Classificazione più probabile
- Intro Naive Bayes
- Naive Bayes per documenti testuali
- Algoritmo EM

## Esercizi:

- Implementare NB su documenti testuali (per esempio "Twenty User Newsgroups", [http://scikit-learn.org/stable/datasets/twenty\\_newsgroups.html](http://scikit-learn.org/stable/datasets/twenty_newsgroups.html))
- Confrontarlo con altri classificatori