Apprendimento bayesiano per la categorizzazione dei testi

Simone Gasperoni

October 22, 2016

L'apprendimento bayesiano è un metodo computazionale di apprendimento basato sul calcolo delle probabilità, che può fornire predizioni probabilistiche sfruttando i principi del teorema di Bayes per realizzare un apprendimento non supervisionato (mediante le Reti Bayesiane e i classificatori bayesiani). I classificatori bayesani sono metodi statistici di classificazione, predicono la probabilità che una data istanza appartenga ad una certa classe.

1 Introduzione

I classificatori bayesiani sono metodi incrementali: ogni istanza dell'insieme di addestramento modifica in maniera incrementale la probabilità che una ipotesi sia corretta. La conoscenza già acquisita può essere combinata facilmente con le nuove osservazioni basta aggiornare i conteggi. Questi metodi sono utilizzati ad esmpio in Mozilla o SpamAssassin per riconoscere le mail spam dalle mail ham.[1]

Una probabilità è una misura su un insieme di eventi che soddisfa tre assiomi:

$$0 \leq P(E = e_i) \leq 1$$

$$\sum_{n=1}^{n} P(E = e_i) = 1$$

$$P(E = e_1 \cup E = e_2) = P(E = e_1) + P(E = e_2)^2$$

Un modello probabilistico consiste in uno spazio di possibili esiti mutualmente esclusivi insieme alla misura di probabilità associata ad ogni esito. Che tempo fa domani? esiti: {SOLE, NUVOLE, PIOGGIA, NEVE}, l'evento corrispondente ad una precipitazione è il sottoinsieme {PIOGGIA, NEVE}.

Definizione 1. La probabilità condizionale è definita come:

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

[1] slide del corso di "Intelligenza artificiale", Università di Roma Tre, AA 2015-2016

[2] gli eventi e1 ed e2 sono disgiunti

Definizione 2. A e B sono condizionalmente indipendenti se

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

o il suo equivalente

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

2 Formula di Bayes

Il teorema di Bayes (conosciuto anche come formula di Bayes o teorema della probabilità delle cause), proposto da Thomas Bayes, deriva da due teoremi fondamentali delle probabilità: il teorema della probabilità composta e il teorema della probabilità assoluta. Viene impiegato per calcolare la probabilità di una causa che ha scatenato l'evento verificato.[3]

Teorema 1. *gli eventi A (per ogni i) sono stocasticamente indipendenti e sono spesso chiamati cause di E, vale a dire:*

$$E \subset \bigcup_{j=1}^{n} A_j$$

abbiamo

$$P(A_i|E) = \frac{P(E|A_i)P(A_i)}{P(E)} = \frac{P(E|A_i)P(A_i)}{\sum_{j=1}^{n} P(E|A_j)P(A_j)}$$

Segue la dimostrazione.[4] *Dalla teoria delle probabilità condizionali abbiamo:*

$$(1)\ \ P(A_i|E) = \frac{P(A_i \cap E)}{P(E)}$$

$$(2)\ \ P(E|A_i) = \frac{P(A_i \cap E)}{P(A_i)} \Rightarrow P(A_i \cap E) = P(E|A_i)P(A_i)$$

andiamo a sostituire la (2) a numeratore della (1) ottenendo:

$$P(A_i|E) = \frac{P(E|A_i)P(A_i)}{P(E)}$$

dato che

$$E \subset \bigcup_{j=1}^{n} A_j$$

abbiamo che

$$P(E) = \sum_{j=1}^{n} P(E|A_j)P(A_j)$$

perché

$$E = E \cap (\bigcup_{j=1}^{n} A_j) \Rightarrow E = \bigcup_{j=1}^{n} E \cap A_j$$

[3] wikipedia.it

infine abbiamo che

$$P(E) = P(\bigcup_{j=1}^{n} E \cap A_j)$$

essendo gli eventi A incompatibili abbiamo

$$P(E) = \sum_{j=1}^{n} P(E \cap A_j) \Rightarrow \sum_{j=1}^{n} P(E|A_j)P(A_j)$$

3 Classificatori bayesiani naïve

L'assunzione di indipendenza rende i calcoli possibili consente di ottenere classificatori ottimali quando è soddisfatta ma è raramente soddisfatta in pratica. Questa assunzione consentono di considerare le relazioni causali tra gli attributi, in realtà, si è visto che anche quando l'ipotesi di indipendenza non è soddisfatta, il classificatore naïve Bayes spesso fornisce ottimi risultati.

Sia X una istanza da classificare, e C1,...,Cn le possibili classi. I classificatori Bayesiani calcolano

$$P(C_i|X)$$

come

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

Andiamo a cercare l'indice i per massimizzare la probabilità condizionata, ottenendo

$$Max_i[P(C_i|X)]$$

P(X) è uguale per tutte le classi per cui non occorre calcolarla, P(Ci) si può calcolare facilmente sull'insieme dei dati di addestramento, si conta la percentuale di istanze di classe Ci sul totale. Assunzione dei classificatori naïve: indipendenza degli attributi. Se X è composta dagli attributi 'a' con indice da 1 a m, otteniamo

$$P(X|C_i) = \prod_{j=1}^{m} P(A_j = a_j|C_i)$$

per il calcolo di

$$P(A_j = a_j|C_i)$$

La formula che verrà utilizzata nella fase di predizione dal classificatore sarà pertato:

$$v = Max_i[P(C_i) \prod_{j=1}^{m} P(A_j = a_j|C_i)]$$

dove v è la categoria predetta.

[4]Introduzione alla probabilità, Enzo Orsingher, Luisa Beghin, Carocci editore

Se gli A sono categorici, viene stimato come la frequenza relativa delle istanze che hanno quel determinato valore a con indice j tra tutte le istanze di C. Se A è continuo, si assume che la probabilità segue una distribuzione Gaussiana, con media e varianza stimata a partire dalle istanze di classe C.

```
#pseudo codice classificatore naif
#vettore dei target v[j]
def v[]
#vettore degli attributi a[i]
def a[]
#p(a[i]|v[j])
#probabilita' che occorra a[i] quando il documento etichettato v[j]
#p(v[j]) probab. che occorra v[j]

def double p(e){
   "ritorna la stima probabilistica per l'evento e"
}

def void naif_bayes_learner(){
   for each j do {
      stima p(v[j])
      for each i do {
         stima p(a[i]|v[j])
      }
   }
}

#entry da classificare: x
def v nuova_classificazione(x){
      "ritorna v[j] tale che
      p(v[j])*p(a[1]|v[j])*..*p(a[i]|v[j])*..*p(a[n]|v[j])
      sia il massimo possibile"
}
```

3.1 Stima delle probabilità (m-estimate)

Quando calcoliamo le probabilità dobbiamo avere alcune accortezze, infatti, se un certo valore di un attributo non si verifica mai per una data classe quando arriva una nuova istanza X la probabilità sarà sempre nulla indipendentemente da quanto siano probabili i valori per gli altri attributi. Il problema delle frequenze nulle non è il solo nella stima delle probabilità in un classificatore bayesiano. Le probabilità tendono ad essere sottostimate in alcune circostanze, ad esempio:

$$P(wind = strong|playTennis = no)$$

stimato come

$$\frac{n_C}{n}$$

Questa stima è buona in molti casi ma se abbiamo pochi esempi con playTennis=no la stima tenderà a zero. Un modo per far fronte a tutte queste problematiche è mediante l'uso di una stima chiamata "m-estimate":

$$\frac{n_C + mp}{n + m}$$

p è la probabilità a priori, solitamente si assume p come il reciproco di k dove k è il numero di valori diversi per attributo

$$\frac{n_C + m\frac{1}{k}}{n + m}$$

m è la *equivalent sample size*, come si può vedere dalla formula serve a determinare il peso di incidenza di p sui dati osservati.

4 Classificazione bayesiana di testi

Il classificatore byaesiano sopradescritto trova applicazione nel campo della categorizzazione dei testi essendo - ad oggi - uno dei metodi più efficaci conosciuti. Segue una trattazione sull'algoritmo che sfrutta le intuizioni probabilistiche bayesiane, altri esempi sono descritti da Lewis (1991), Lang (1995), Joachims (1996)[5]

Le istanze X che abbiamo considerato fin'ora possono ora essere considerati documenti testuali. Il training set da considerare è una collezione di documenti etichettati (classificati), su questa base di conoscenza si dovrà costruire un sistema di predizione per le entry dello spazio X.

Prendiamo in considerazione una collezione di testi, ad esempio 1000, dei quali 300 interessano ad una certa persone, mentre invece, gli altri 700 sono considerati non interessanti, questo può essere considerato un dataset per addestrare il nostro classificatore dove le categorie sono "like" e "dislike".

Due problemi fondamentali nella progettazione del classificatore bayesiano sono: la rappresentazione dei documenti, e la modalità di stima delle probabilità.

4.1 Rappresentazione testi e stima delle probabilità

Il modo più semplice di rappresentare i testi è mediante una raccolta - senza considerare l'ordine - di parole. Testi lunghi daranno luogo ad insiemi di attributi molto grandi, come vedremo questo non è un problema. Questo tipo di approccio è personalizzabile introducendo n-grammi di parole o la lemmatizzazione. Ricollegandoci all'esempio di prima abbiamo:

$$v = Max_i[P(C_i) \prod_{j=1}^{m} P(A_j = a_j | C_i)]$$

[5]Machine learning, Mc Graw Hill, 1997, Tom M. Mitchell

dove le categorie sono due:

$$C_1 : like, C_2 : dislike$$

avremo dunque

$$P(C_1) = \frac{300}{1000}, \;\; P(C_2) = \frac{700}{1000}$$

Le probabilità condizionate sono semplicemente proporzionali alle frequenze con cui occorre un parola dentro tutti i documenti di una categoria. La stima delle probabilità è una m-estimate nella quale consideriamo

$$m = p = |Vocabolario|$$

$$\frac{w+1}{n+|Vocabolario|}$$

con n numero di parole di tutti i documenti di una determinata categoria, w numero di occorrenze di una data parola nell'insieme di parole di una data categoria.

In sintesi l'algoritmo di classificazione dei testi usa un classificatore bayesiano naïve con l'assunzione che la probabilità di occorrenza della parola è indipendente dalla posizione dentro i documenti.

Segue lo psudocodice di un approccio minimale:

```
#pseudo codice classificatore naif per la categorizzazione testi
#vettore dei target v[j]
def v[]

#inizializzo vocabolario con tutti i vocaboli
def vocabolario=init_vocabolario()

def void naif_bayes_TEXT_learner(){
   #per ogni target v[j]
   for each j do {
      docs[j]="insieme dei documenti etichettati con v[j]"
      p(v[j])=|docs[j]|/|Esempi|
      text[j]="concateno tutti i docs[j]"
      n="numero di parole distinte dentro text[j]"

      #qui si calcolano i pesi per le parole
      for each parola in Vocabolario do{
         w="numero di volte che la parola occorre in text[j]"
         p(w|v[j])=(w+1)/(n+|Vocabolario|)
      }
   }
}

#entry da classificare: x
nuova_classificazione(x)
```