

LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA (D.M. 270)

UNIVERSITÀ DEGLI STUDI DI ROMA TRE

Catalogazione open data

SIMONE GASPERONI

L^AT_EX
Roma, Italia

Contents

1	Introduzione	2
1.1	La filosofia open-data	2
1.2	Interoperabilità	2
1.3	EUROVOC	3
1.4	DCAT e DCAT-AP-it	3
2	Elaborazione dei corpus	4
2.1	Lemmatizzazione	4
2.2	Disambiguazione	5
3	Vector space model	7
3.1	Il classificatore JEX	7
4	Bayesian learning	8
4.1	Formula di Bayes	8
4.2	Classificatori bayesiani naïve	10
4.3	Stima delle probabilità (m-estimate)	11
4.4	Classificazione bayesiana di testi	12
4.5	Rappresentazione testi e stima delle probabilità	12
5	Aspetti software	14
5.1	Architettura	14
5.2	Deployment	14

1 Introduzione

I dati sono aperti se chiunque è libero di accederli, usarli, modificarli, e condividerli

1.1 La filosofia open-data

Sulla pagina Open Definition 2.1¹ - ultima definizione rilasciata - è possibile prendere nota degli aspetti più importanti che caratterizzano la filosofia open-data gli aspetti più importanti sono:

- Disponibilità e accesso.
- Riutilizzo e redistribuzione: i dati devono essere forniti a condizioni tali da permetterne il riutilizzo e la redistribuzione. Ciò comprende la possibilità di combinarli con altre basi di dati.
- Partecipazione universale: tutti devono essere in grado di usare, riutilizzare e redistribuire i dati.

Il movimento open-data si diffonde notevolmente a seguito della promulgazione della Direttiva sull'Open government del 2009 negli Stati Uniti d'America: *Fin dove possibile e sottostando alle sole restrizioni valide, le agenzie devono pubblicare le informazioni on line utilizzando un formato aperto (open) che possa cioè essere recuperato, soggetto ad azioni di download, indicizzato e ricercato attraverso le applicazioni di ricerca web più comunemente utilizzate. Per formato open si intende un formato indipendente rispetto alla piattaforma, leggibile dall'elaboratore e reso disponibile al pubblico senza che sia impedito il riuso dell'informazione veicolata.*

In questo contesto è nato il portale Data.gov, creato con l'obiettivo di raccogliere in un unico portale tutte le informazioni rese disponibili dagli enti statunitensi in formato aperto.

In Italia il portale che raccoglie i dati aperti della pubblica amministrazione è Dati.gov.it

1.2 Interoperabilità

Tra le ragioni più importanti della filosofia open-data abbiamo l'interoperabilità, vale a dire, la capacità di diversi sistemi e organizzazioni di lavorare insieme. Nel caso specifico degli open data - ovviamente - ci riferiamo alla capacità di combinare una grande quantità di dati da diverse fonti. L'interoperabilità è importante perché permette a componenti diverse di lavorare insieme. L'abilità di rendere ciascun dato un componente e di combinare insieme vari componenti è essenziale per la costruzione di sistemi sofisticati. In assenza di interoperabilità ciò diventa quasi impossibile - come nel mito della Torre di Babele, in cui l'impossibilità di comunicare (e quindi di Inter-operare) dà luogo a un fallimento

¹<http://opendefinition.org/od/2.1/en>

sistemico della costruzione della torre. Una chiara definizione di "apertura" assicura che sia possibile combinare dataset aperti provenienti da fonti diverse in modo adeguato, evitando così la "Torre di Babele".

Il punto cruciale di un bacino di dati (o linee di codice) accessibili e utilizzabili in modo condiviso è il fatto che potenzialmente possono essere liberamente "mescolati" con dati provenienti da fonti anch'esse aperte. L'interoperabilità è la chiave per realizzare il principale vantaggio pratico dell'apertura: aumenta in modo esponenziale la possibilità di combinare diverse basi di dati, e quindi sviluppare nuovi e migliori prodotti e servizi.²

1.3 EUROVOC

1.4 DCAT e DCAT-AP-it

²<http://opendatahandbook.org/guide/it/what-is-open-data>

2 Elaborazione dei corpus

L'Elaborazione del linguaggio naturale, detta anche NLP (dall'inglese *Natural Language Processing*), è il processo di trattamento automatico mediante un calcolatore elettronico delle informazioni scritte o parlate in una lingua naturale.

Questo processo è reso particolarmente difficile e complesso a causa delle caratteristiche intrinseche di ambiguità del linguaggio umano. Per questo motivo il processo di elaborazione viene suddiviso in fasi diverse, tuttavia simili a quelle che si possono incontrare nel processo di elaborazione di un linguaggio di programmazione:

- **Analisi lessicale:** scomposizione di un'espressione linguistica in token (in questo caso le parole).
- **Analisi grammaticale:** associazione delle parti del discorso a ciascuna parola nel testo.
- **Analisi sintattica:** arrangiamento dei token in una struttura sintattica (ad albero: parse tree).
- **Analisi semantica:** assegnazione di un significato (semantica) alla struttura sintattica e, di conseguenza, all'espressione linguistica.

Nell'analisi semantica la procedura automatica che attribuisce all'espressione linguistica un significato tra i diversi possibili è detta disambiguazione. La comprensione del linguaggio naturale è spesso considerata un problema IA-completo, poiché si pensa che il riconoscimento del linguaggio richieda una conoscenza estesa del mondo e una grande capacità di manipolarlo. Per questa ragione, la definizione di "comprensione" è uno dei maggiori problemi dell'elaborazione del linguaggio naturale.³

2.1 Lemmatizzazione

La lemmatizzazione è "quel complesso di operazioni che conducono a riunire tutte le forme sotto il rispettivo lemma", intendendo per lemma "ciascuna parola-titolo o parola-chiave di un dizionario" e per forma ogni possibile diversa realizzazione grafica di un lemma.⁴

Sono esempi di forme "mangerò", "mangeremmo", "mangiaste", "mangi": il lemma cui ciascuna di queste forme è riducibile è "mangiare" (ovvero, la parola che appare come entrata sui dizionari), in quanto è convenzione italiana che il lemma verbale sia rappresentato dalla forma coniugata all'infinito presente attivo.

Esistono dunque delle convenzioni di lemmatizzazione proprie di ciascuna lingua: come visto, in italiano è uso convenzionale che il lemma verbale sia la forma coniugata all'infinito presente attivo.

³I. Chiari, Introduzione alla linguistica computazionale. Laterza, 2007

⁴R. Busa, Fondamenti di informatica linguistica. Istituto Geografico De Agostini, 1987

Il lemma è anche una forma: “mangiare” è, quindi, sia un lemma (che, da solo, rappresenta tutte le proprie possibili forme), sia una forma. Il contrario non vale: una forma non è necessariamente lemma. Infatti, mangiasse è una forma, ma non un lemma.

La lemmatizzazione è, dunque, una pratica apparentemente facile, se non, addirittura, ovvia ed intuitiva: più, o meno tutti, infatti, esercitiamo quotidianamente la conoscenza della differenza tra lemma e forma. Tuttavia, alla prova dei fatti, la lemmatizzazione rivela una serie di problemi, il più delle volte non immaginabili prima di averne fatto esperienza diretta: queste difficoltà rendono il lemmatizzare un esercizio interessante, in quanto costringe chi lo esercita a riflettere su:

- Quanti e quali automatismi l'uomo metta inconsciamente in atto ogni volta che parla, o scrive.
- Quanto sia difficile formalizzare questi automatismi.

Esistono due tipi di lemmatizzazione:

- **Lemmatizzazione morfologica:** analizza le forme di parole in isolamento, ovvero fuori dal contesto sintattico, fornendone tutti i valori che sono possibili in un dato sistema linguistico.
- **Lemmatizzazione morfo-sintattica:** analizza le forme di parole entro il contesto sintattico. Non è mai ambigua, ma sempre univoca, in quanto l'immersione della forma nella frase ne precisa il valore. Quindi, mentre la lemmatizzazione morfologica è indipendente dal testo, la lemmatizzazione sintattica è, invece, legata al contesto.

Il lessico (o vocabolario) è il complesso delle parole e delle locuzioni di una lingua (ad esempio, il lessico italiano, il lessico greco). Un lessico in genere ha una forma molto strutturata, memorizza i significati e i possibili usi di ogni parola, codifica le relazioni fra le parole e i significati. Un dizionario è una forma di lessico in cui i significati sono espressi tramite definizioni ed esempi.

Un lessema è l'unità minima che viene rappresentata nel lessico, associa il lemma (la forma ortografica scelta per rappresentare la parola) con una forma simbolica di rappresentazione del significato (il senso).

2.2 Disambiguazione

Molte parole nel linguaggio naturale sono delle polisemie, vale a dire, possono avere più significati. Le tecniche finalizzate alla disambiguazione sono conosciute in letteratura come *Word Sense Disambiguation* (WSD).

La WSD può essere affrontata come un problema di classificazione, il senso corretto è la classe da predire la parola è rappresentata con un insieme (vettore) di feature in ingresso al classificatore, l'ingresso è di solito costituito da una rappresentazione della parola da disambiguare (target) e del contesto in cui si

trova (un certo numero di parole a sinistra e destra della parola target). Il classificatore può essere stimato con tecniche di apprendimento automatico a partire da un dataset etichettato. Si possono usare diversi modelli per costruire il classificatore (Naïve Bayes, reti neurali, alberi di decisione...).

Il problema dell'approccio appena descritto è che il modello di apprendimento supervisionato che scegliamo di adottare potrebbe richiedere un training set troppo grande per essere sufficientemente preciso, per questo motivo esistono tecniche alternative come i metodi .

Tra i metodi *Dictionary-based* abbiamo l'algoritmo di Lesk (1986) che è un metodo molto semplice, basato sull'intuizione secondo cui un dizionario può fornire informazioni sul contesto legato ai sensi delle parole (le glosse).

3 Vector space model

3.1 Il classificatore JEX

4 Bayesian learning

L'apprendimento bayesiano è un metodo computazionale di apprendimento automatico 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 bayesiani sono metodi statistici di classificazione, predicono la probabilità che una data istanza appartenga ad una certa classe.

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 esempio in Mozilla o SpamAssassin per riconoscere le mail spam dalle mail ham.⁵

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

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

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

$$P(E = e_1 \cup E = e_2) = P(E = e_1) + P(E = e_2)^6$$

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:

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

Definizione 2. A e B sono condizionalmente indipendenti se

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

o il suo equivalente

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

4.1 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

⁵slide del corso di "Intelligenza artificiale", Università di Roma Tre, AA 2015-2016

⁶gli eventi e1 ed e2 sono disgiunti

della probabilità assoluta. Viene impiegato per calcolare la probabilità di una causa che ha scatenato l'evento verificato.⁷

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.⁸ Dalla teoria delle probabilità condizionali abbiamo:

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

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

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

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

infine abbiamo che

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

essendo gli eventi A incompatibili abbiamo

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

⁷wikipedia.it

⁸Introduzione alla probabilità, Enzo Orsingher, Luisa Beghin, Carocci editore

4.2 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 C_1, \dots, C_n 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

$$\text{Max}_i[P(C_i|X)]$$

$P(X)$ è uguale per tutte le classi per cui non occorre calcolarla, $P(C_i)$ si può calcolare facilmente sull'insieme dei dati di addestramento, si conta la percentuale di istanze di classe C_i 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 pertanto:

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

dove v è la categoria predetta.

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"
}

```

4.3 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.

4.4 Classificazione bayesiana di testi

Il classificatore bayesiano 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)⁹

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 persona, 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.

4.5 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:

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

dove le categorie sono due:

$$C_1 : \textit{like}, C_2 : \textit{dislike}$$

avremo dunque

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

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

$$m = p = |\textit{Vocabolario}|$$
$$\frac{w + 1}{n + |\textit{Vocabolario}|}$$

⁹Machine learning, Mc Graw Hill, 1997, Tom M. Mitchell

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 pseudocodice 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(parola|v[j])=(w+1)/(n+|Vocabolario|)
    }
  }
}

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

5 Aspetti software

5.1 Architettura

5.2 Deployment