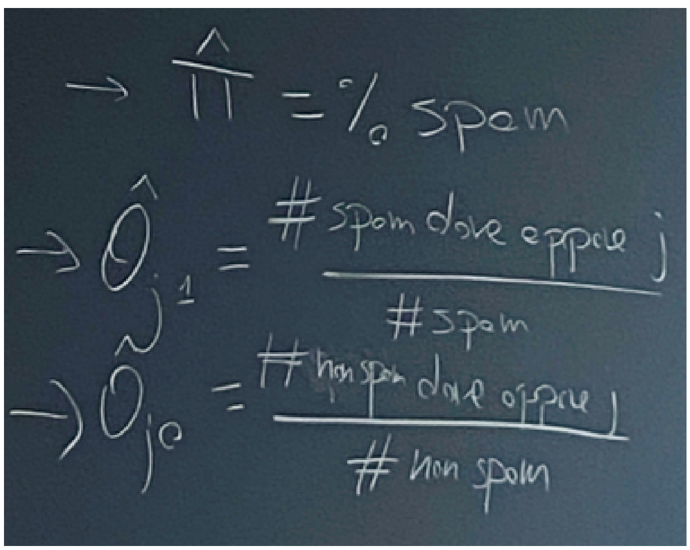
Lez20_ ProbabilisticModel2

December 6, 2023

1 Modelli probabilistici - parte 2

1.1 Breve recap

Nella scorsa lezione, abbiamo ottenuto, in riferimento alla parte **Learning**:



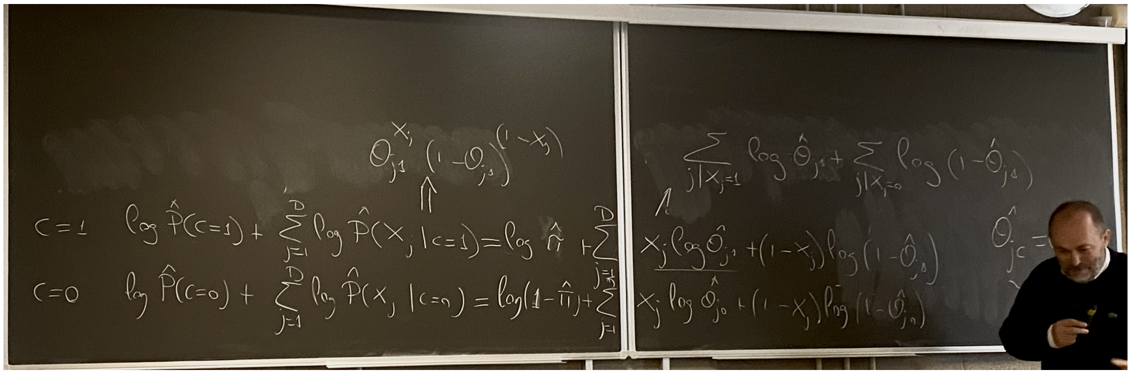
Mentre, lato **Inference**:

$h(x) = arg\ max_{c_k}\ log(\ P(c) \cdot \prod_{j=1}^{D} P(x_j|c=c_k)\ )$ (che possiamo scomporre in una somma, essendo un prodotto tra logaritmi!)

In questa formula, c può assumere due valori:

- $c = 1$, allora $h(x) = log(\bar{P}(c=1)) + \sum_{j=1}^{D} log(\bar{P}(x_j|c=1)) = log(\pi) + \sum_{j=1}^{D} \theta_{j_1}^{x_j} \cdot (1-\theta_{j_1})^{1-x_j}$, riscrivibile come: $log(\pi) + \sum_{j|x_j=1} log(\bar{\theta}_j) + \sum_{j|x_j=0} log(1-\bar{\theta}_{j_1})$
- $c = 0$ allora
  $h(x) = log(\bar{P}(c=0)) + \sum_{j=1}^{D} log(\bar{P}(x_j|c=0)) = log(1-\pi) + \sum_{j=1}^{D} \theta_{j_0}^{x_j} \cdot (1-\theta_{j_0})^{1-x_j}$, riscrivibile come: $log(\pi) + \sum_{j|x_j=1} log(\bar{\theta}_j) + \sum_{j|x_j=0} log(1-\bar{\theta}_{j_0})$

Riferimento (non spiegato benissimo dal professore):



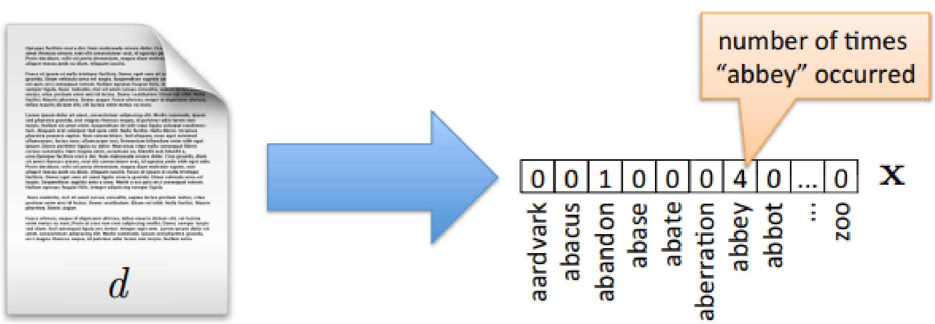
1.2 Document classification

Adesso ci chiediamo, come effetuiamo la classificazione? Qual è il miglior modo per rappresentare un documento? Lo vogliamo *semplice* e *utile* per i nostri scopi.

L'insieme di parole viste prima non ci dà tutte queste informazioni, ad esempio perdiamo informazioni sull'ordine di queste parole. Va bene per discriminare due argomenti (come spam e non spam).

1.2.1 Rappresentazione del documento

Rappresentiamo con x_j che ci dice *quante volte la parola occorre.*



1.2.2 Approccio con Naive Bayes

Prendiamo $\theta_c = \{\theta_{c_1}, ...\theta_{c_{|D|}}\}$ dove:

- D è il dizionario.
- θ_{c_j} è la probabilità che la parola j occorra nel documento di classe c. E' come se da un sacco estraessimo una certa parola, ma qui non stiamo assumendo un ordine, quindi sono equiprobabili. Ad esempio, in una mail, è più probabile iniziare con "*ciao*" rispetto ad "*addio*", però qui non considero questo aspetto. L'idea è che da qualche parola io riesca ad estrarre il contesto, ad esempio sentendo parlare di *Roma* e *Lazio* potrei associarle al calcio.
- $\sum_j \theta_{c_j} = 1$, e vale $\forall c$

La massima verosimiglianza del documento d caratterizzata dal vettore $\bar{x}$ è:

$$P(d|c) = \frac{(\sum_j x_j)!}{\prod_j x_j!} \prod_j (\theta_{cj})^{x_j}$$

Qui i valori di x non sono più 0 o 1, bensì le occorrenze. $P(d|c)$ è la distribuzione *multinomiale*, una generalizzazione della binomiale $P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$

(nb: d e $\bar{x}$ sarebbero la stessa cosa, cioè il documento è identificato da tale vettore. Per consistenza, il professore l'ha espressa come nella formula).

Sfruttando **Bayes**, e qualche step che il professore skippa, possiamo dire che:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \propto P(d|c)P(c) \propto \prod_j (\theta_{cj})^{x_j} P(c)$$

e successivamente (considerando anche l'operazione di *log*):

$$h(d) = \arg\max_c \left(\log P(c) + \sum_{j=1}^{|D|} x_j \log \theta_{jc} \right)$$

che è una funzione lineare di x, ovvero una generalizzazione delle formule viste ad inizio slide, in funzione di $c = 0$ e $c = 1$.

Dentro, abbiamo una sommatoria calcolata per ogni parola, ma molti hanno contatore pari a 0, e quindi tale operazione non è esosa.

1.2.3 Nella pratica

1. Creiamo il dizionario D.
2. Stimiamo la clase **prior**, ovvero $\bar{P}(c) = \frac{N_c}{N}$, dove N è il numero di documenti e N_c i documenti di classe c.
3. Stimiamo le probabilità condizionate, mediante **Laplace Smoothing** (per evitare di perdere parole che non compaiono nei nostri dataset), ovvero $\bar{\theta}_{c_j} = \frac{W_{c_j}+1}{W_c+|D|}$, dove W_c è il numero totale di parole in tutti i documenti di classe c e W_{c_j} è il numero delle volte che la parola j occorre nei documenti di classe c.

Allora, la classificazione **Naive bayes** per un nuovo documento d è:

$h(d) = \arg\max_c(\log \bar{P}(c) + \sum_j x_j \log \bar{\theta}_{j_c})$

La fase di training è molto semplice, basta fare una *passata* sui documenti, perchè richiede gli step *(1)* e *(2)*.

Chi cerca di evitare i filtri spam, lo fa cambiando le parole o come essa venga scritta (inserendo errori grammaticali, o uno smodato uso degli spazi o caratteri speciali).

1.3 Problematiche del Bag of Words

- Documenti presentano lunghezze diverse.
- Molte parole non sono utili per identificare un documento, basti pensare a tutti gli articoli. Tali parole sono dette *stop words*, e sono parole di uso comune nella grammatica di tutti i giorni (*about, by, did, do, he, if, in, is, me, was, your,...*).
- Non ci siamo mai chiesti se alcune parole possano essere più rilevanti di altre. Ad esempio, la parola *rinoceronte* è difficile immaginarla in un contesto diverso da quello degli *animali*.

1.3.1 Term Frequency

Identificata da $f_{t,d}$, misura l'importanza del termine t nel documento d.

Identifichiamo con $c_{t,d}$ le occorrenze di t in d.

Possiamo sfruttare diversi conteggi:

- **Boolean**: $f_{t,d} = \begin{cases} 1 & \text{if } c_{t,d} > 0, \\ 0 & \text{altrimenti} \end{cases}$
- **Raw Counts**: $f_{t,d} = c_{t,d}$
- **Log-Scaled Counts**: $f_{t,d} = \begin{cases} 1 + \log c_{t,d} & \text{if } c_{t,d} > 0, \\ 0 & \text{altrimenti} \end{cases}$

  Riduce l'impatto relativo del termine frequente.
- **Normalized Counts**: $f_{t,d} = \frac{c_{t,d}}{|d|}$ Normalizza i raw counts per la lunghezza del documento d.

1.3.2 Inverse Document Frequency

L'idea è l'inverso di prima, ovvero valorizzare le parole più rare, come *ippopotamo*, può portare più informazioni rispetto al parlare semplicemente di *acqua*.

La formula è: $\text{if}_{t,N} = \log \frac{|N|}{|N_t|+1}$, dove al numeratore abbiamo il set completo dei documenti, e al denominatore il subset dei documenti contenenti il termine t, che incrementiamo di $+1$.

1.3.3 TF-IDF Transform

Per compensare i problemi con i *raw counts* delle parole, utilizzare la trasformazione TF-IDF sui fattori con il Naive Bayes, usiamo $f_{\text{ift},d,x} = f_{t,d} \times \text{ift}_{t,x}$

Essa rappresenta i documenti come vettore x di feature *TF-IDF*, dove x_j rappresenta il TF-IDF della parola j nel documento.

Raccomandazioni:

- E' consigliato usare *raw counts* sui conteggi di $f_{t,d}$ log-scalati.

- E' consigliato normalizzare *ogni vettore x TF-IDF*, per omogenizzare la lunghezza. Tipicamente si usa $\bar{x} = \frac{x}{||x||_2}$, per poi usare tale valore come *unit vector* in Naive Bayes.

[ ]: