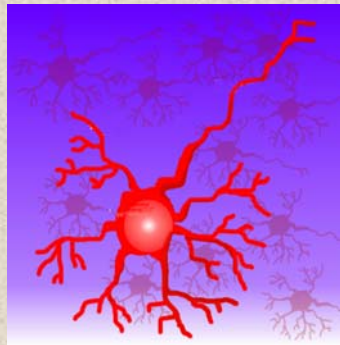


# Text Categorization

metodologia di apprendimento mediante approcci statistici e  
reti neurali artificiali



Claudio Biancalana

1

## Contatti

---

**Claudio Biancalana**

**[c.biancalana@openinformatica.org](mailto:c.biancalana@openinformatica.org)**

2

# Information Retrieval (IR)

---

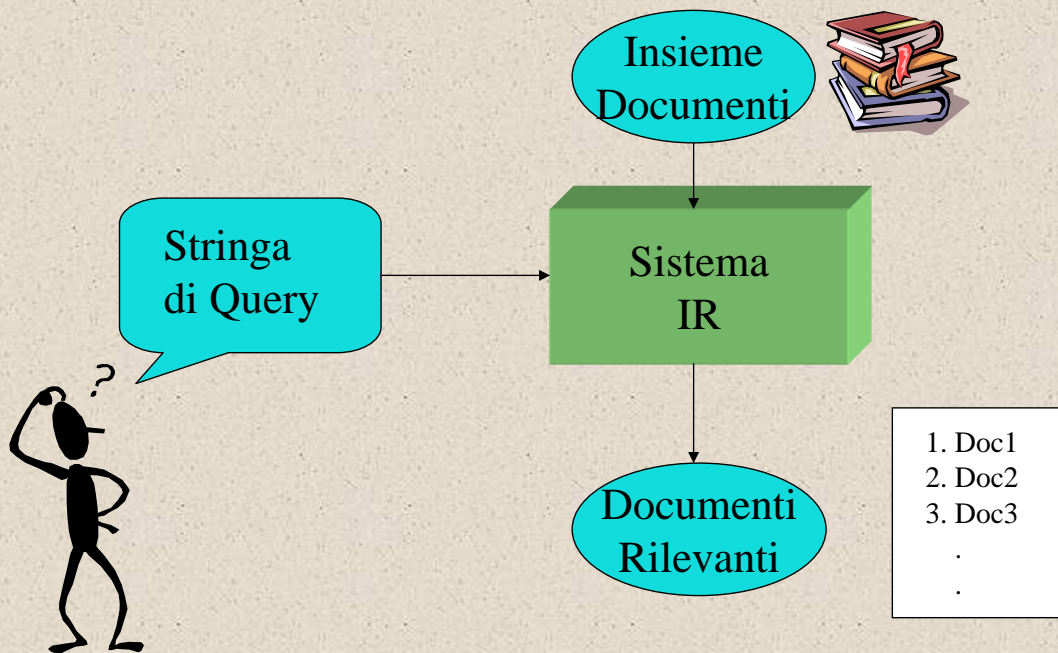
- Indicizzazione e ricerca di documenti testuali
- Ricerca di pagine sul web
- Obiettivo principale è la ricerca **efficace** di documenti rilevanti
- Obiettivo secondario è la ricerca **efficiente** tra un insieme vasto di documenti

## Obiettivi IR

---

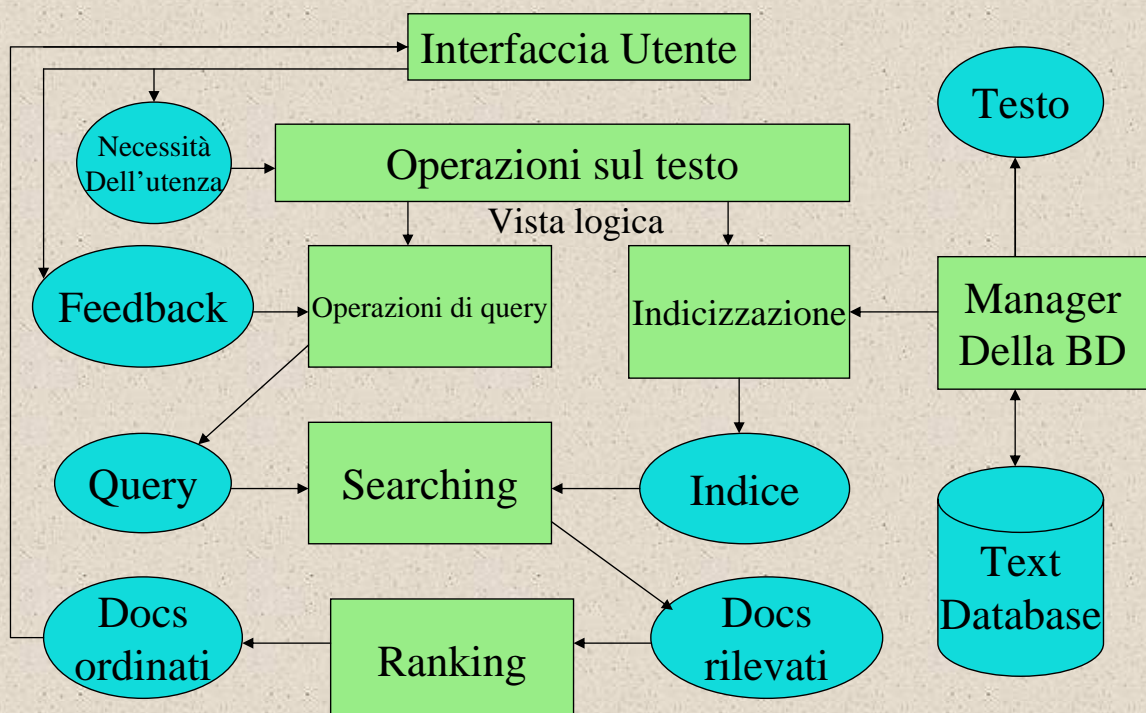
- **Input:**
  - Un insieme di documenti testuali scritti in linguaggio naturale.
  - Una stringa rappresentante una query.
- **Output:**
  - Un insieme di documenti ordinati in base alla loro rilevanza alla query.

# Sistema IR



5

# Architettura di un sistema IR



6

# Categorizzazione

---

- Input:
  - Una descrizione di una istanza,  $x \in X$ , dove  $X$  è l'istanza linguaggio o spazio dell'istanza.
  - Un numero fissato di categorie:  
 $C = \{c_1, c_2, \dots, c_n\}$
- Output:
  - La categoria di  $x$ :  $c(x) \in C$ , dove  $c(x)$  è una funzione di categorizzazione che ha come dominio  $X$  e come codominio  $C$ .

7

# Imparare per categorizzare

---

- Un esempio di apprendimento per una istanza  $x \in X$ , accoppiata con la sua categoria  $c(x)$ :  
 $\langle x, c(x) \rangle$  con una sconosciuta funzione di categorizzazione,  $c$ .
- Sia dato un insieme di esempi di apprendimento,  $D$ .
- Trovare una ipotizzata funzione di categorizzazione,  $h(x)$ , tale che:

$$\forall \langle x, c(x) \rangle \in D : h(x) = c(x)$$

*Consistenza*

8



## Category Learning Problem (esempio)

- Istanza di linguaggio:  $\langle \text{grandezza, colore, forma} \rangle$ 
  - grandezza  $\in \{\text{small, medium, large}\}$
  - colore  $\in \{\text{rosso, blu, verde}\}$
  - forma  $\in \{\text{quadrato, cerchio, triangolo}\}$
- $C = \{\text{positivo, negativo}\}$

•  $D$ :

Esempio	Grand.	Colore	Forma	Categoria
1	small	rosso	cerchio	positivo
2	large	rosso	cerchio	positivo
3	small	rosso	triangolo	negativo
4	large	blu	cerchio	negativo

9

## Text Categorization

- Assegnare documenti ad un insieme fissato di categorie.
- Applicazioni:
  - Pagine web
    - Recommending
    - Classificazioni (Yahoo!)
  - Messaggi su Newsgroup
    - Recommending
    - Filtraggio spam
  - Articoli
    - Giornali personalizzati
  - Email
    - Routing
    - Prioritizing
    - Folderizing
    - Filtraggio spam

10

## Algoritmi di apprendimento

- Lo sviluppo “manuale” di una funzione di text categorization è difficile.
- Algoritmi per l'apprendimento:
  - **Bayesian (naïve)**
  - Reti neurali
  - **Relevance Feedback (Rocchio)**
  - Nearest Neighbor (case based)
  - **Support Vector Machines (SVM)**
  - Basato su regole

11

## Rappresentazione dei documenti

- Una collezione di  $n$  documenti può essere rappresentata nel modello come una matrice dei termini del documento.
- Un valore nella matrice corrisponde al “**peso**” di **un termine nel documento**; zero significa che il termine non è significativo nel documento o più semplicemente non è presente nel documento.

$$\begin{pmatrix} & T_1 & T_2 & \dots & T_t \\ D_1 & w_{11} & w_{21} & \dots & w_{t1} \\ D_2 & w_{12} & w_{22} & \dots & w_{t2} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{1n} & w_{2n} & \dots & w_{tn} \end{pmatrix}$$

12

## TF\*IDF

- Metodologia di rappresentazione dei pesi del vettore per il pattern recognition

TF = frequenza del termine nel documento

$$\text{IDF} = \log (N/n)$$

N= numero totale di documenti nella collezione di training

n = numero di documenti che contengono il termine nella collezione di training

Per ogni termine abbiamo **TF\*IDF**

13

## Rappresentazione grafica

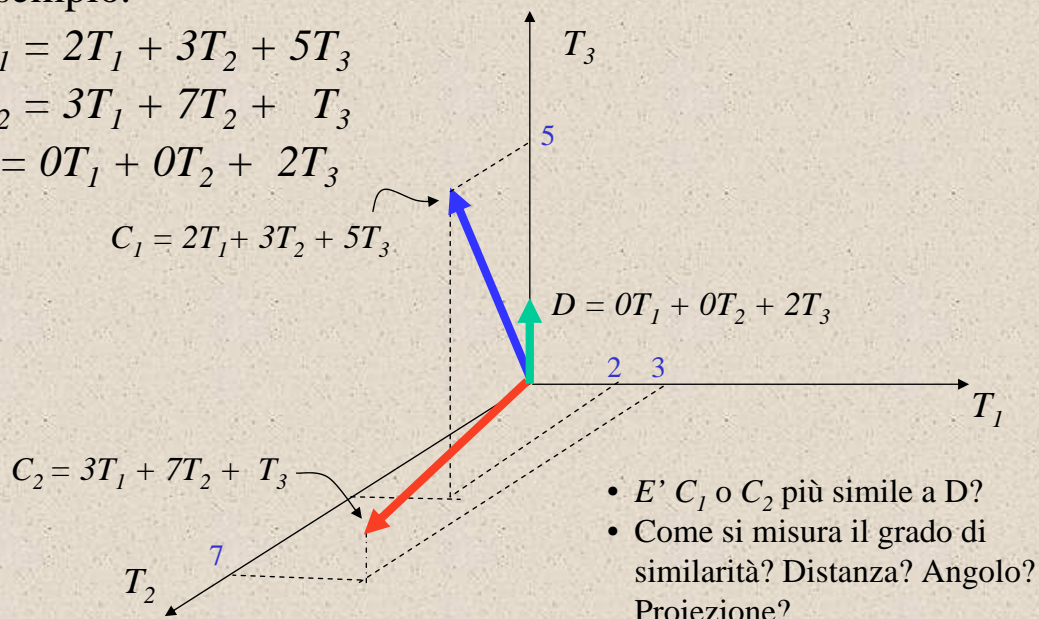
Esempio:

$$C_1 = 2T_1 + 3T_2 + 5T_3$$

$$C_2 = 3T_1 + 7T_2 + T_3$$

$$D = 0T_1 + 0T_2 + 2T_3$$

$$C_1 = 2T_1 + 3T_2 + 5T_3$$



14

## Algoritmo rocchio

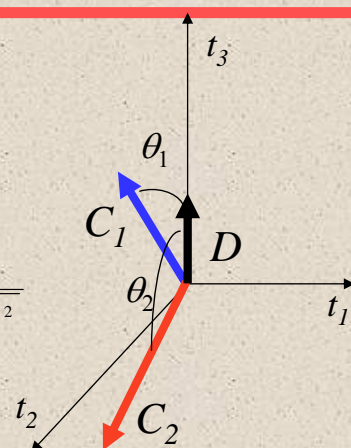
- Usiamo lo standard di indicizzazione TF/IDF per rappresentare in forma vettoriale i documenti di testo (normalizzati secondo la frequenza massima di un termine)
- Per ogni categoria, viene elaborato un vettore “Prototipo” dalla somma dei vettori di training nella categoria
- Assegnamo il documento di test alla categoria col vettore “prototipo” più vicino mediante la regola di similarità del coseno.

15

## Regole di similarità del coseno

- Si misura il coseno dell'angolo fra due vettori...

$$\text{CosSim}(c_j, d) = \frac{c_j \cdot d}{|c_j| \cdot |d|} = \frac{\sum_{i=1}^t (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^t w_{ij}^2} \cdot \sqrt{\sum_{i=1}^t w_{iq}^2}}$$



$$\begin{aligned} C_1 &= 2T_1 + 3T_2 + 5T_3 & \text{CosSim}(C_1, D) &= 10 / \sqrt{(4+9+25)(0+0+4)} = 0.81 \\ C_2 &= 3T_1 + 7T_2 + 1T_3 & \text{CosSim}(C_2, D) &= 2 / \sqrt{(9+49+1)(0+0+4)} = 0.13 \\ D &= 0T_1 + 0T_2 + 2T_3 \end{aligned}$$

16



## Algoritmo Rocchio (apprendimento)

---

Sia l'insieme delle categorie  $\{c_1, c_2, \dots, c_n\}$   
For  $i$  from 1 to  $n$  let  $\mathbf{p}_i = \langle 0, 0, \dots, 0 \rangle$  (*inizializzazione*)  
For each esempio di training  $\langle x, c(x) \rangle \in D$   
    Let  $\mathbf{d}$  = vettore TF/IDF per il doc  $x$   
    Let  $i = j: (c_j = c(x))$   
    (*somma di tutti i vettori in  $c_i$  per ottenere  $\mathbf{p}_i$* )  
    Let  $\mathbf{p}_i = \mathbf{p}_i + \mathbf{d}$

17

## Algoritmo Rocchio (Test)

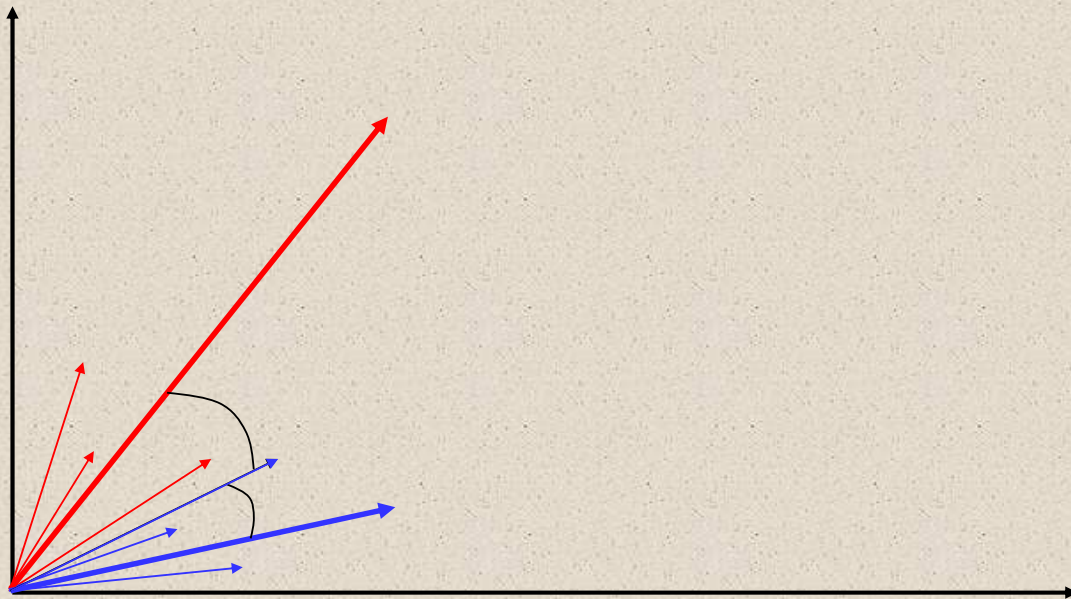
---

Dato un documento di test  $x$   
Let  $\mathbf{d}$  = vettore TF/IDF per  $x$   
Let  $m = -2$  (*inizializzazione*)  
For  $i$  from 1 to  $n$ :  
    (*calcola la similarità col vettore prototipo*)  
    Let  $s = \text{cosSim}(\mathbf{d}, \mathbf{p}_i)$   
    if  $s > m$   
        let  $m = s$   
        let  $r = c_i$  (*aggiorna il più simile*)  
Return class  $r$

18

## Text categorization con Rocchio

---



19

## Algoritmo di apprendimento Nearest-Neighbor

---

- L'apprendimento si riduce al modo di immagazzinare le rappresentazioni degli esempi di training in  $D$ .
- Test dell'istanza  $x$ :
  - Elabora la similarità tra  $x$  e tutti gli esempi in  $D$ .
  - Assegna ad  $x$  la categoria del più simile in  $D$ .
- Non si calcolano esplicitamente i prototipi delle categorie.
- Conosciuto anche sotto il nome di:
  - Case-based
  - Memory-based
  - Lazy learning

20

## Metrica per la similarità

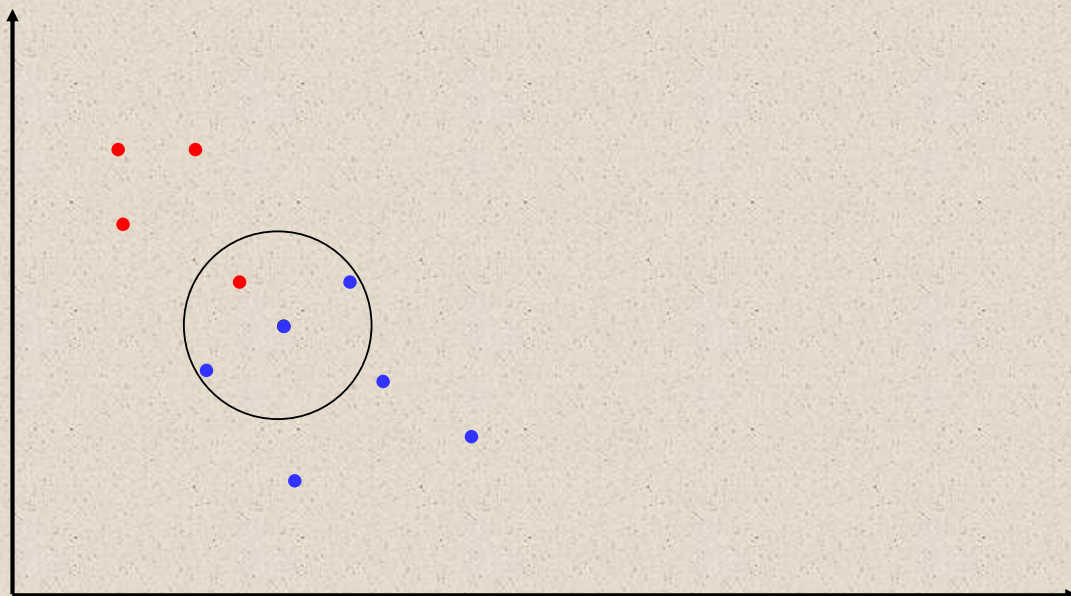
---

- Nearest neighbor si basa su una metrica di similarità (o distanza)
- La più semplice per uno spazio continuo è la distanza euclidea.
- La più semplice per spazi d'istanza m-dimensionali binari è la distanza di Hamming
- Per i testi, la similarità basata sul coseno, per i vettori costruiti mediante indicizzazione TF-IDF, è tipicamente la più efficiente.

21

## 3 Nearest Neighbor Illustration (Euclidian Distance)

---



22

## K Nearest Neighbor per testi

---

### Training:

For each each esempio di training  $\langle x, c(x) \rangle \in D$

Calcola il corrispondente vettore TF-IDF,  $\mathbf{d}_x$ , per il doc  $x$

### Test dell'istanza $y$ :

Calcola il vettore TF-IDF  $\mathbf{d}$  per il doc  $y$

For each  $\langle x, c(x) \rangle \in D$

Let  $s_x = \text{cosSim}(\mathbf{d}, \mathbf{d}_x)$

Ordina gli esempi,  $x$ , in  $D$  al decrescere di  $s_x$

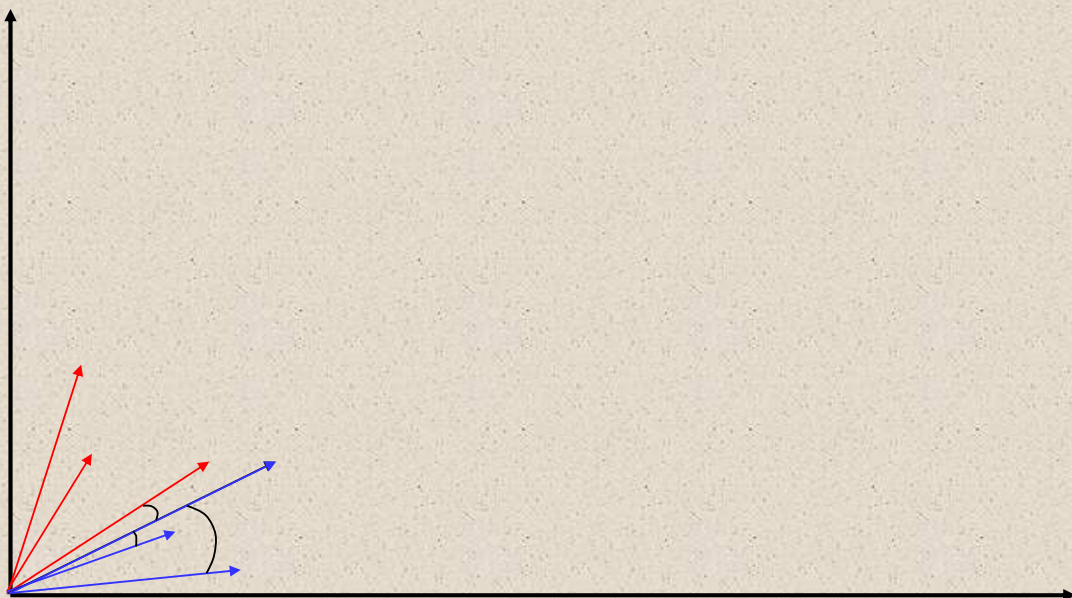
Let  $N = I$  primi  $k$  esempi in  $D$ . *(ottiene così i vicini più simili)*

Return la classe con più esempi in  $N$

23

## Nearest Neighbor per testi

---

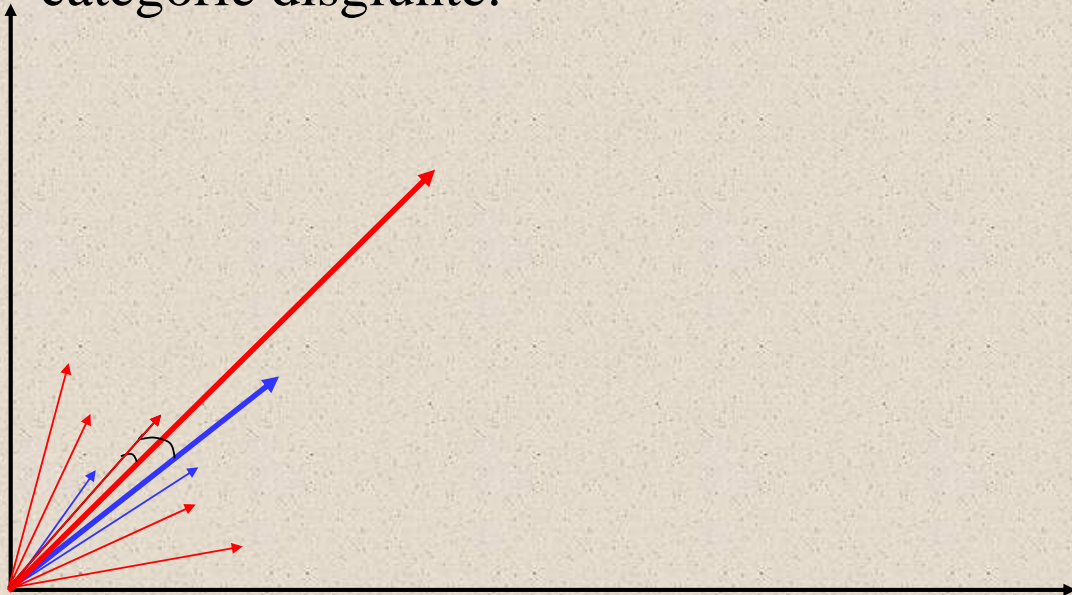


24



## Rocchio: anomalia

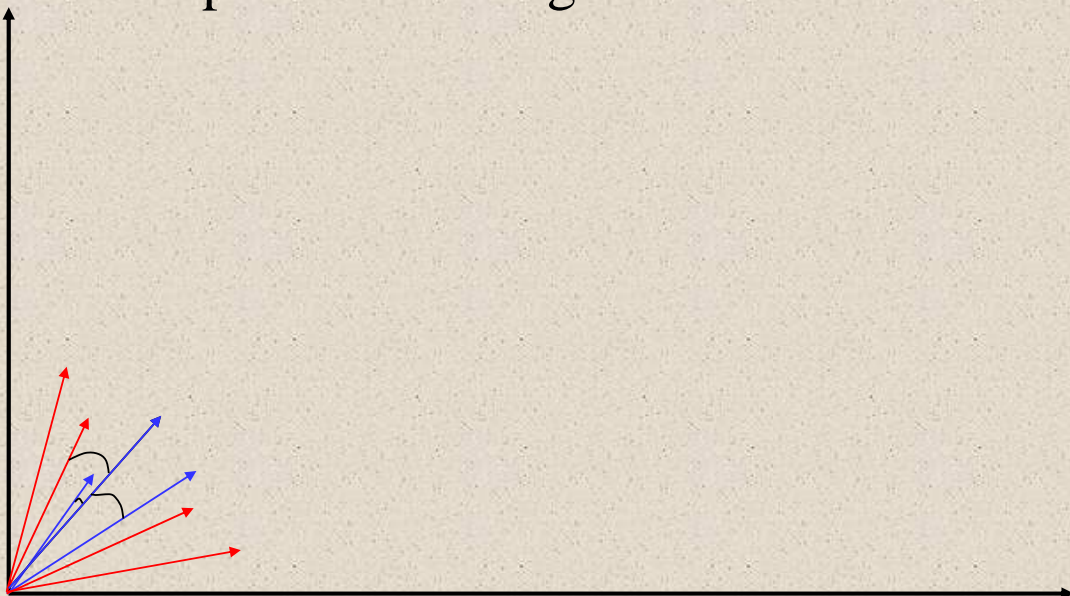
- I prototipi possono avere problemi con categorie disgiunte.



25

## Nearest Neighbor

- Nearest Neighbor tende ad avere in tal caso un comportamento migliore.



26

## Metodi bayesiani

---

- Apprendere e classificare mediante approcci probabilistici
- Il teorema di bayes gioca un ruolo critico nell'apprendimento e classificazione.

27

## Categorizzazione Bayesiana

---

- Sia l'insieme delle categorie  $\{c_1, c_2, \dots, c_n\}$
- Sia  $E$  una descrizione di un'istanza
- Determinare il grado di appartenenza di  $E$  per ogni  $c_i$
- $P(E)$  può essere determinata solo se le categorie sono complete e disgiunte.

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

$$P(E) = \sum_{i=1}^n P(c_i)P(E | c_i)$$

$$\sum_{i=1}^n P(c_i | E) = \sum_{i=1}^n \frac{P(c_i)P(E | c_i)}{P(E)} = 1$$

28

## Categorizzazione Bayesiana

- E' necessario sapere:
  - $P(c_i)$
  - $P(E | c_i)$
- $P(c_i)$  sono facilmente stimati dai dati.
  - se  $n_i$  degli esempi in  $D$  sono in  $c_i$ , allora  $P(c_i) = n_i / |D|$
- Se si assume che le caratteristiche di una istanza siano indipendenti data la categoria  $c_i$

$$P(E | c_i) = P(e_1 \wedge e_2 \wedge \dots \wedge e_m | c_i) = \prod_{j=1}^m P(e_j | c_i)$$

29

## Esempio

- $C = \{\text{allergia, raffreddore, salute}\}$
- $e_1 = \text{starnuto}; e_2 = \text{tosse}; e_3 = \text{febbre}$
- $E = \{\text{starnuto, tosse, } \neg \text{febbre}\}$

Prob	salute	raffreddore	allergia
$P(c_i)$	0.9	0.05	0.05
$P(\text{starnuto} c_i)$	0.1	0.9	0.9
$P(\text{tosse} c_i)$	0.1	0.8	0.7
$P(\text{febbre} c_i)$	0.01	0.7	0.4

30

## Esempio (continua)

Probabilità	Salute	Raffreddore	Allergia
$P(c_i)$	0.9	0.05	0.05
$P(\text{starnuto}   c_i)$	0.1	0.9	0.9
$P(\text{tosse}   c_i)$	0.1	0.8	0.7
$P(\text{febbre}   c_i)$	0.01	0.7	0.4

$E = \{\text{starnuto, tosse, } \neg \text{febbre}\}$

$$P(\text{salute} | E) = (0.9)(0.1)(0.1)(0.99)/P(E) = 0.0089/P(E)$$

$$P(\text{raffreddore} | E) = (0.05)(0.9)(0.8)(0.3)/P(E) = 0.01/P(E)$$

$$P(\text{allergia} | E) = (0.05)(0.9)(0.7)(0.6)/P(E) = 0.019/P(E)$$

Categoria più probabile: allergia

$$P(E) = 0.089 + 0.01 + 0.019 = 0.0379$$

$$P(\text{salute} | E) = 0.23$$

$$P(\text{raffreddore} | E) = 0.26$$

$$P(\text{allergia} | E) = 0.50$$

31

## Algoritmo di apprendimento Naïve Bayes

Let  $V$  = il vocabolario di tutte le parole nei documenti in  $D$

For each categoria  $c_i \in C$

Let  $D_i$  = il sottoinsieme di documenti in  $D$   
nella categoria  $c_i$

$$P(c_i) = |D_i| / |D|$$

Let  $T_i$  = la concatenazione di tutti i documenti in  $D_i$

Let  $n_i$  = il numero totale delle occorrenze in  $T_i$

For each parola  $w_j \in V$

Let  $n_{ij}$  = il numero di occorrenze di  $w_j$  in  $T_i$

$$\text{Let } P(w_i | c_i) = (n_{ij} + 1) / (n_i + |V|)$$

32



## Algoritmo di test Naïve Bayes

---

Dato un documento di test  $X$

Let  $n$  = numero di occorrenze in  $X$

Return categoria:

$$\operatorname{argmax}_{c_i \in C} P(c_i) \prod_{i=1}^n P(a_i | c_i)$$

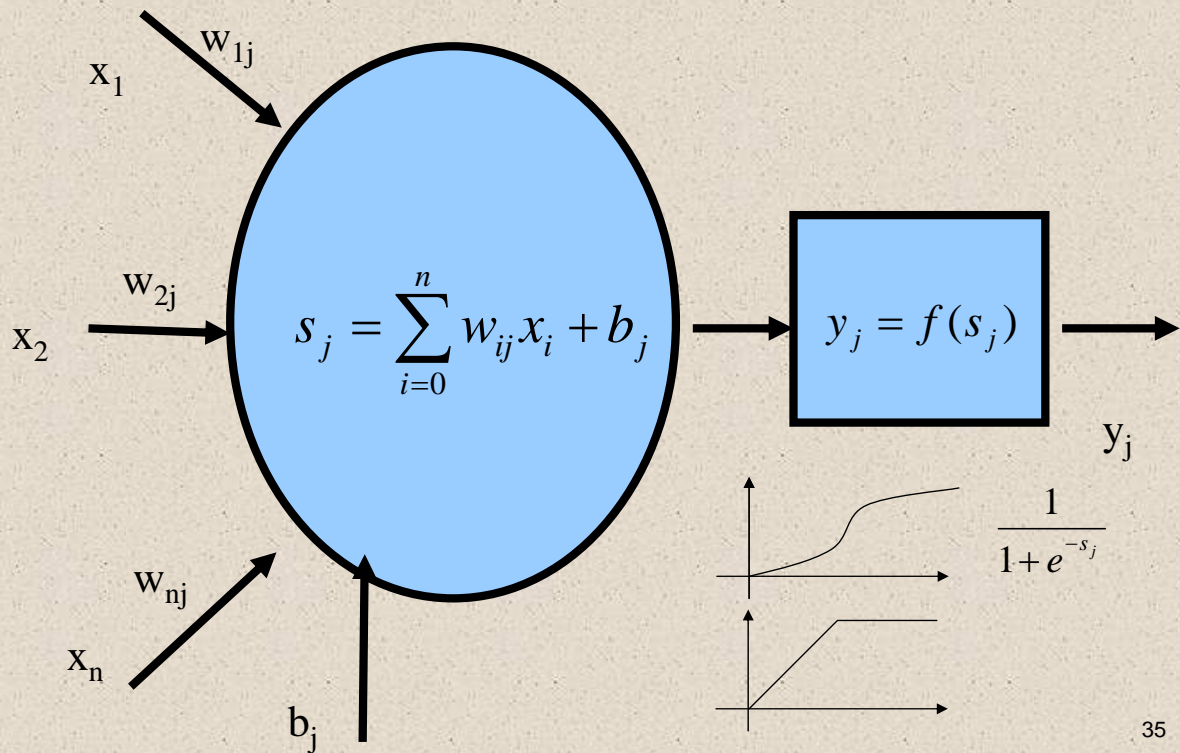
dove  $a_i$  l'occorrenza della parola nella  $i$ -esima  
posizione in  $X$

## Reti neurali

---

- Una rete neurale consiste in un pool di semplici processi elementari che comunicano fra loro spedendosi segnali attraverso numerose connessioni pesate

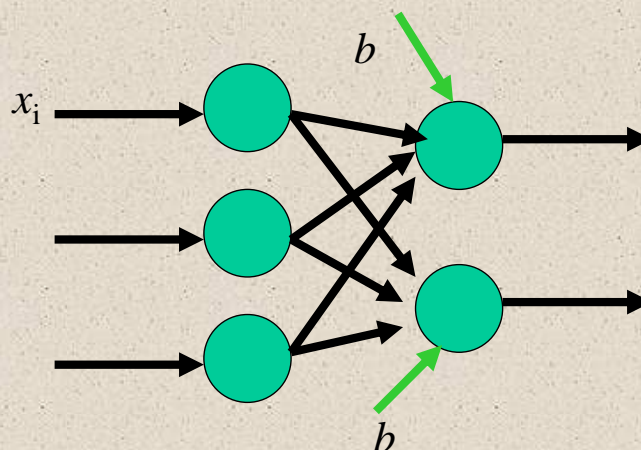
## Neurone artificiale



35

## Percettrone

- Una rete a strato singolo consiste in uno o più neuroni di output, ognuno dei quali è connesso con un fattore peso  $w_{ij}$  a tutti gli input  $x_i$ .

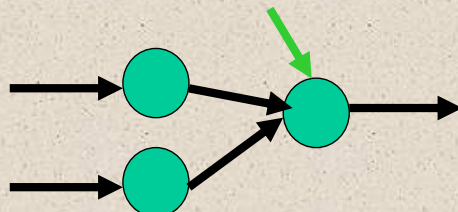


36

## Percettrone

---

- Nel caso più semplice abbiamo solo due input e un solo output. L'output del neurone è:

$$y = f\left(\sum_{i=1}^2 w_i x_i + b\right)$$


- Supponiamo la seguente funzione di attivazione

$$f = \begin{cases} 1 & \text{if } s > 0 \\ -1 & \text{if } s \leq 0 \end{cases}$$

37

## Percettrone

---

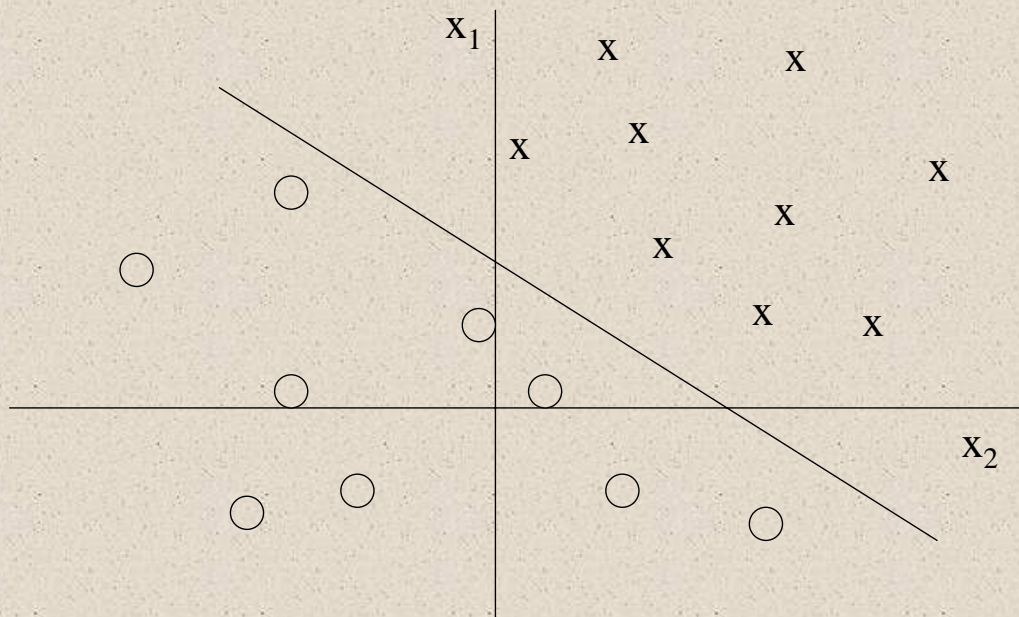
- In questa semplice rete (il neurone) può essere usato per separare gli input in due classi.
- La separazione nelle due classi è data da

$$w_1 x_1 + w_2 x_2 + b = 0$$

38

## Percettrone

---



39

## Learning

---

- I pesi della rete neurale sono modificati durante la fase di learning

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}$$

$$b_{ij}(t+1) = b_{ij}(t) + \Delta b_{ij}$$

40



## Learning

---

- Si parte con pesi casuali
- Dalla coppia di input  $(\mathbf{x}, d(\mathbf{x}))$ :  
se  $y \neq d(\mathbf{x})$  allora modifica il peso mediante la formula:

$$\Delta w_{ij} = d(\mathbf{x})x_i$$

NB Regola del gradiente discendente

$$\Delta w_{ij} = -\gamma \frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial w_{ij}} \quad \frac{\partial y_i}{\partial w_{ij}} = x_j \quad \frac{\partial E}{\partial y_i} = -(d_i - y_i) = \delta_i \quad \Delta w_{ij} = \gamma \delta_i x_j$$

←-----→   ←-----→   ←-----→

41

## Categorizzazione e perceptron

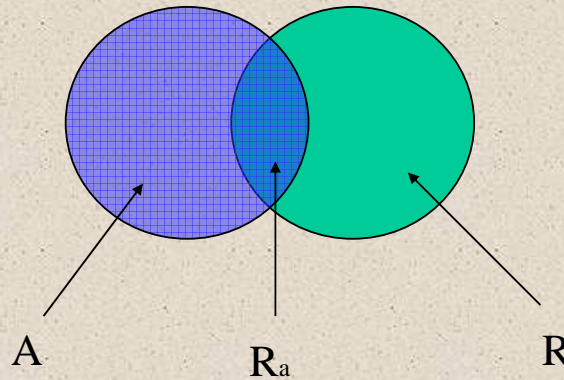
---

- Un perceptrone per ogni categoria
- Learning sui documenti di training della sua categoria
- Durante la fase di test, il perceptrone fornisce un valore VERO/FALSO sull'appartenenza del vettore rappresentativo il documento alla categoria

42

## Valutare la categorizzazione

- Esistono due parametri accettati dalla comunità IR



RECALL:  $|R_a|/|R|$     PRECISION:  $|R_a|/|A|$

43

## Valutare la categorizzazione

- PRECISION
  - L'abilità nel restituire i documenti che sono più rilevanti.
- RECALL
  - L'abilità nel restituire tutti i documenti rilevanti dell'intero dominio.
- F-Measure

$$F = \frac{2PR}{P + R} = \frac{2}{\frac{1}{R} + \frac{1}{P}}$$

44