Text Mining

Modulo dell'Insegnamento di Data Mining

Riduzione della Dimensionalità
Selezione di Feature con Mutual Information e Test Chiquadro
Latent Semantic Analysis

Laurea Magistrale in Ingegneria e Scienze Informatiche
DISI – Università di Bologna

Gianluca Moro
Dipartimento di Informatica – Scienza e Ingegneria
Università di Bologna
Via Venezia, 52 – I-47521 Cesena (FC)
Gianluca.Moro@Unibo.It

(draft)

Riduzione della Dimensionalità (i)

- Repository di doc generano un num. di feature molto superiori a problemi con dati strutturati
  - 10,000 – 1,000,000 parole distinte … (uni-gram, ngram..)
- Ridurre il numero di feature rende applicabili un numero maggiore di tool e classificatori
  - alcuni non sono in grado di elaborare milioni di feature
- Meno feature riducono il tempo di training
  - il tempo di training per alcuni metodi è quadratico rispetto al numero di feature
- I modelli di classificazione prodotti sono più snelli e veloci e possono essere più generali
  - assenza di feature che aumentano rumore e overfitting

Riduzione della Dimensionalità (ii)

- L'assunzione è che i dati, in genere, contengano variabili ridondanti/irrilevanti nella costruzione di modelli di mining
- Area di ricerca e tecnologica molto vasta
  - selezione di feature:
    selezionano un sottoinsieme di feature lasciando invariata la rappresentazione dei dati
  - estrazione di feature:
    trasformazione della rappresentazione dei dati secondo un nuovo insieme di feature ridotto
- la selezione di feature più semplice nel text mining:
  - usare i termini più frequenti **Term Frequency,** o rilevanti **TF-IDF** etc., *e.g. i primi k termini con migliore TF-IDF*
  - nella pratica, nel 90% dei casi, se si utilizzano migliaia di feature, sono efficaci quanto i metodi più avanzati

Feature Controproducenti: Esempio

- Riprendiamo la classificazione per la classe *China*
- Supponiamo che un termine raro, come ***arachnocentric,*** non induca informazioni per la classe *China* ...
- ... ma tutte le istanze di ***arachnocentric*** sono nei documenti del training set della classe *China*
- un classificatore apprenderebbe incorrettamente che ***arachnocentric*** è un'evidenza per la classe *China*
- questa informazione accidentale del training set produce una generalizzazione scorretta
  - i.e. il modello è affetto da *overfitting*

Selezione di feature: 2 metodi

- l'obiettivo è stimare, dal training set, l'utilità delle feature nella classificazione
- primo metodo: Mutual Information
  - misura la reciproca dipendenza di due variabili, i.e. *quanta informazione hanno in comune*
  - nella classificazione di documenti si misura la dipendenza tra termini e classi/categorie
  - per ogni classe si selezionano i termini con la mutual information maggiore
- secondo metodo: Test χ^2 (chiquadro)
  - test statistico di verifica d'ipotesi
  - fissato un livello di confidenza, indica se la differenza tra dati osservati e attesi è significativa o frutto del caso

Mutual information

- Calcola "quanta informazione" condividono un termine *t* ed una classe (misurata in num. di bit con $\log_2$)
- ad esempio, se il termine *t* è indipendente dalla classe allora la mutual information è 0
  - questo accade quando la distribuzione del termine *t* è la stessa nella classe e nell'intero corpus di documenti
- Definizione:
  - *U* è la variabile aleatoria relativa al documento, se vale $e_t = 1$, il termine *t* è nel doc, altrimenti $e_t = 0$
  - *C* è la variabile aleatoria relativa alla classe, se $e_c = 1$ il doc è nella classe *c*, altrimenti $e_c = 0$

$$I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) \log_2 \frac{P(U = e_t, C = e_c)}{P(U = e_t) P(C = e_c)}$$

Mutual Information: come calcolarla

- si calcola la stima della massima verosimiglianza
- N_{10}: numero di doc che contengono t ($e_t = 1$) e che non sono nella classe c ($e_c = 0$);
- N_{11}: numero di doc che contengono t ($e_t = 1$) e che sono nella classe c ($e_c = 1$);
- N_{01}: numero di doc che non contengono t ($e_t = 1$) e che sono nella classe c ($e_c = 1$);
- N_{00}: numero di doc che non contengono t ($e_t = 1$) e che non sono nella classe c ($e_c = 1$);
- $N = N_{00} + N_{01} + N_{10} + N_{11}$

$$I(U;C) = \frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_{0.}N_{.0}}$$

Esempio di MI su Reuters: termine EXPORT e classe *poultry*

	$e_c = e_{poultry} = 1$	$e_c = e_{poultry} = 0$
$e_t = e_{\text{EXPORT}} = 1$	$N_{11} = 49$	$N_{10} = 27{,}652$
$e_t = e_{\text{EXPORT}} = 0$	$N_{01} = 141$	$N_{00} = 774{,}106$

sostituzione di questi valori nella formula: **N = num doc totali 801.948**

$$\begin{aligned} I(U;C) &= \frac{49}{801{,}948}\log_2\frac{801{,}948\cdot 49}{(49+27{,}652)(49+141)} \\ &+ \frac{141}{801{,}948}\log_2\frac{801{,}948\cdot 141}{(141+774{,}106)(49+141)} \\ &+ \frac{27{,}652}{801{,}948}\log_2\frac{801{,}948\cdot 27{,}652}{(49+27{,}652)(27{,}652+774{,}106)} \\ &+ \frac{774{,}106}{801{,}948}\log_2\frac{801{,}948\cdot 774{,}106}{(141+774{,}106)(27{,}652+774{,}106)} \\ &\approx 0.000105 \end{aligned}$$

Selezione di Feature con Mutual Information su Reuters

Class: *coffee*

term	MI
COFFEE	0.0111
BAGS	0.0042
GROWERS	0.0025
KG	0.0019
COLOMBIA	0.0018
BRAZIL	0.0016
EXPORT	0.0014
EXPORTERS	0.0013
EXPORTS	0.0013
CROP	0.0012

Class: *sports*

term	MI
SOCCER	0.0681
CUP	0.0515
MATCH	0.0441
MATCHES	0.0408
PLAYED	0.0388
LEAGUE	0.0386
BEAT	0.0301
GAME	0.0299
GAMES	0.0284
TEAM	0.0264

SELEZIONE DI FEATURE CON TEST CHIQUADRO

Selezione di Feature con Test Statistico χ^2 (chiquadro)

- test per verificare l'indipendenza di due eventi A e B
  - A e B sono indipendenti se P(AB)=P(A)P(B) oppure …
  - … se P(A|B) = P(A) e P(B|A) = P(B)
- Nella selezione di feature i due eventi sono:
  - A = occorrenza del termine *t*
  - B = occorrenza della classe *c*
- Siano *N* le frequenze osservate ed *E* le freq. attese, ipotizzando i due eventi indipendenti *(ipotesi nulla)*
- Siano $e_c, e_t \in \{0, 1\}$ col medesimo significato in MI
- La formula seguente stima la differenza tra i valori osservati e quelli attesi per *t* e *c* in D

$$X^2(\mathbb{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}$$

Test Statistico χ^2 applicato a doc (i)

$$X^2(\mathbb{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}$$

- e.g. E_{11}=num. atteso di doc in cui il termine *t* e la classe *c* sono presenti insieme assumendo che *t* e *c* siano indipendenti

$$E_{11} = |D| \cdot P(t) \cdot P(c) = |D| \cdot \frac{N_{11} + N_{10}}{|D|} \cdot \frac{N_{11} + N_{01}}{|D|}$$

- applicato all'esempio precedente sul data set Reuter con *t* = export e *c*=poultry

$$E_{11} = |D| \cdot \frac{49 + 141}{|D|} \cdot \frac{49 + 27652}{|D|} \approx 6.6$$

- Calcoliamo gli altri valori $E_{e_t e_c}$ nella stessa maniera

Test Statistico χ^2 (chiquadro) (ii)

	$e_{poultry} = 1$	$e_{poultry} = 0$
$e_{export} = 1$	$N_{11} = 49$ $E_{11} \approx 6.6$	$N_{10} = 27{,}652$ $E_{10} \approx 27{,}694.4$
$e_{export} = 0$	$N_{01} = 141$ $E_{01} \approx 183.4$	$N_{00} = 774{,}106$ $E_{00} \approx 774{,}063.6$

- la formula misura quanto deviano tra loro i valori attesi e quelli osservati, in questo esempio tra il termine **EXPORT** e la classe ***poultry***

$$X^2(\mathbb{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \approx 284$$

- maggiore è χ^2, minore è la prob. che valga l'ipotesi di indipendenza tra termine e classe -> **serve una soglia**
  - si può fare questa inferenza perché se 2 eventi sono indipendenti, allora $X^2 \sim \chi^2$, dove χ^2 è la distribuzione omonima

Test Statistico χ^2 (chiquadro) (iii)

$$X^2(\mathbb{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \approx 284$$

- l'ipotesi (nulla) di indipendenza tra *t* e *c* è rigettata ?
- tabella distribuzione χ^2 *(con 1 grado di libertà)*
  - *gradi di libertà*=(i-1)x(j-1) <-*matrice precedente*
  - i = num. di colonne
  - j = num. di righe

p	χ^2
0.1	2.71
0.05	3.84
0.01	6.63
0.005	7.88
0.001	10.83

- e.g. due eventi producono $X^2 = 5$
  - allora 5 > 3.84 => l'ipotesi di indipendenza è rigettata con confidenza (i.e. certezza) del 95%
  - ma se occorre maggiore certezza, e.g. confidenza 99% allora l'ipotesi è accettata: 5 < 6.63
    -> *le differenze tra i valori attesi e quelli osservati risultano casuali*

Test Statistico χ^2 (chiquadro) (iv)

$$X^2(\mathbb{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \approx 284$$

- l'ipotesi (nulla) di indipendenza tra *t* e *c* è rigettata ?
- calcoliamo i gradi di liberta
  - la matrice delle osservazioni è 2 x 2
  - gradi di libertà: (2-1)x(2-1) = 1
  - -> selezioniamo la relativa distribuzione χ^2
- X^2 = 284 > 10.83 =>
  - l'ipotesi è rigettata
  - con (almeno) 99.9% di confidenza
  - i.e. c'è (molto) meno di 1 possibilità su 1000 che *t* e *c* siano indipendenti -> ***t* è una feature valida per *c***

p	χ^2
0.1	2.71
0.05	3.84
0.01	6.63
0.005	7.88
0.001	10.83

Test χ^2: formula più semplice

Nel caso 2x2 il valore χ^2 si può ottenere come segue:

$$\chi^2(t, c) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)}$$

A = #(t,c)	C = #(¬t,c)
B = #(t,¬c)	D = #(¬t, ¬c)

N = A + B + C + D

Test Statistico χ^2: Ranking dei Termini

- I valori di X^2 calcolati per ogni termine sono normalizzati, quindi sono confrontabili
- Il rank di ogni termine si calcola come la media pesata dei relativi valori X^2 rispetto alle classi

$$\chi^2_{avg}(t) = \sum_{i=1}^{|C|} P(c_i)\chi^2(D,t,c_i)$$

- Oppure il rank di ciascun termine è il proprio valore massimo

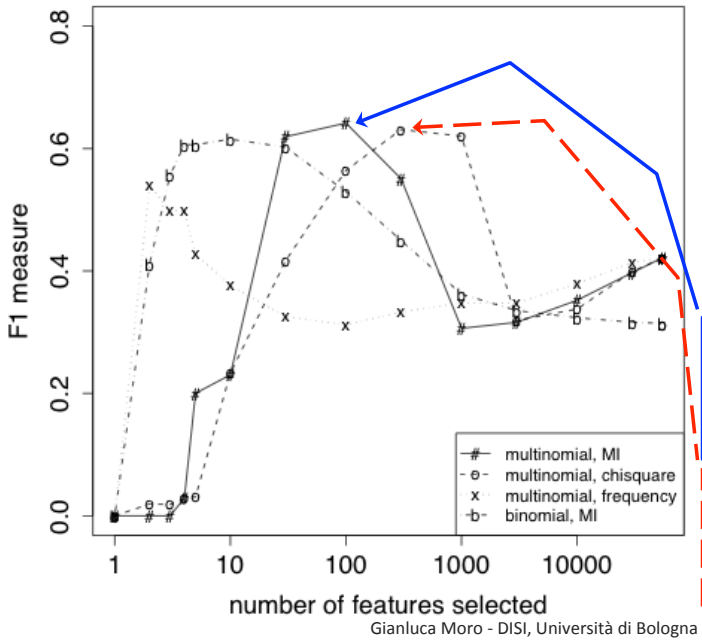
$$\chi^2_{\max}(t) = \max_{i=1}^{|C|}\{\chi^2(D,t,c_i)\}$$

- si selezionano i primi k termini (i.e. feature) con valori maggiori

Test Statistico χ^2: Limiti

- il test χ^2 è inadeguato con valori osservati/attesi bassi
  - non applicabile se dati osservati < 30 o la loro somma < 200
  - metodi di correzione quando i dati osservati sono inferiori a 50 e quelli attesi < 5 (per tabelle 2x2, *correzione di Yates*)
  - altri test statistici adatti con pochi dati (*Kolmogorov-Smirnov*)
- all'aumentare del numero di test eseguiti su un problema, aumenta la probabilità di errore totale
  - in 1000 ipotesi rigettate ciascuna con il 5% di errore, in media 50 test sono sbagliati => *nella classificazione di testo alcuni termini sono erroneamente selezionati o non selezionati,* ***ma ciò raramente influenza in modo significativo il risultato***
- E' più importante il **ranking χ^2 relativo tra termini**
  - non richiede stretta aderenza teorica come i test di indipendenza

Modelli Naive Bayes: Con e Senza Selezione di Feature



100K doc Training e 100K doc Test

binomial = Bernoulli Naive Bayes

feature selezionate con Mutual Information (MI) e chiquadro

multinomial: feature selezionate con MI; il max a 100 feature

multinomial: feature selezionate con χ^2; il max a 300 feature

Valutazioni dei Risultati (i)

- Naive bayes binomiale raggiunge il suo massimo con 10 feature scelte con mutual inform.
  - è il migliore in assoluto con solo 10 feature -> la rappresentazione a termini senza frequenza con poche feature è più robusta
  - all'aumentare delle feature (i.e. dimensioni) l'efficacia diminuisce poiché i doc rappresentati con termini senza frequenza diventano sempre meno distinguibili
- Il Multinomiale raggiunge il max con più feature
  - **Mutual Information**: il max con circa **100 feature**
  - χ^2**:** raggiunge lo stesso valore max ma con **300 feature**
  - *χ^2 rispetto a MI da più importanza a termini rari perciò richiede più termini di MI per arrivare al max*
  - nella pratica si osserva che χ^2 seleziona feature migliori

Valutazioni dei Risultati (ii)

- Il modello multinomiale con termini più frequenti, selezionati per ogni categoria, è il peggiore
  - termini molto frequenti condivisi tra tutte le classi contribuiscono a renderle indistinguibili
    - e.g. i giorni della settimana in un data set di notizie sono tra i più frequenti su tutte le categorie
  - categorie molto sbilanciate nel num. doc -> la frequenza dei termini non normalizzata non le caratterizza
- Due modalità per calcolare i termini più frequenti:
  - frequenza di documento: (*più adatto al modello binomiale*)
    num. dei doc nella classe ***c*** che contengono il termine ***t***
  - frequenza di categoria: (*più adatto al modello multinomiale*)
    num. di occorrenze del termine ***t*** nei doc della classe ***c***

Selezione di Feature con MI su 6 categorie di Reuters: Limiti

UK	
london	0.1925
uk	0.0755
british	0.0596
stg	0.0555
britain	0.0469
plc	0.0357
england	0.0238
pence	0.0212
pounds	0.0149
english	0.0126

China	
china	0.0997
chinese	0.0523
beijing	0.0444
yuan	0.0344
shanghai	0.0292
hong	0.0198
kong	0.0195
xinhua	0.0155
province	0.0117
taiwan	0.0108

poultry	
poultry	0.0013
meat	0.0008
chicken	0.0006
agriculture	0.0005
avian	0.0004
broiler	0.0003
veterinary	0.0003
birds	0.0003
inspection	0.0003
pathogenic	0.0003

coffee	
coffee	0.0111
bags	0.0042
growers	0.0025
kg	0.0019
colombia	0.0018
brazil	0.0016
export	0.0014
exporters	0.0013
exports	0.0013
crop	0.0012

elections	
election	0.0519
elections	0.0342
polls	0.0339
voters	0.0315
party	0.0303
vote	0.0299
poll	0.0225
candidate	0.0202
campaign	0.0202
democratic	0.0198

sports	
soccer	0.0681
cup	0.0515
match	0.0441
matches	0.0408
played	0.0388
league	0.0386
beat	0.0301
game	0.0299
games	0.0284
team	0.0264

per la classificaz. il termine **kong** è ridondante essendo legato solo ad **hong**

MI, χ^2 , TF etc. sono metodi greedy

possono selezionare **feature che non incrementano l'informazione utile** rispetto alle feature già selezionate

Esercizi

- Calcolare MI e X^2 per il termine KYOTO e la classe *Japan* in base a questo data set
- Creare una tabella per cui MI = 0 e il test X^2 accetti l'ipotesi nulla, i.e. termine e classe sono indipendenti

	$e_c = e_{japan} = 1$	$e_c = e_{japan} = 0$
$e_t = e_{KYOTO} = 1$		
$e_t = e_{KYOTO} = 0$		

	docID	words in document	in c = Japan?
training set	1	Kyoto Osaka Taiwan	yes
	2	Japan Kyoto	yes
	3	Taipei Taiwan	no
	4	Macao Taiwan Shanghai	no
	5	London	no

Esperimento con il data set WebKB

- Data set pubblico contenente pagine web di 4 Università:
  - Cornell, Washington, U.Texas, Wisconsin
  - ~8000 istanze pre-classificate in 7 categorie
    student, faculty, person, project, course, depart., *(other)*
  - http://www.cs.cmu.edu/~webkb/
- Esperimento con subset di ~600 istanze con 6 cat.
- Risultati con NB binomiale senza selezione di feature

	Student	Faculty	Person	Project	Course	Departmt
Extracted	180	66	246	99	28	1
Correct	130	28	194	72	25	1
Accuracy:	72%	42%	79%	73%	89%	100%

WebKB: selezione di feature con MI

Faculty	
associate	0.00417
chair	0.00303
member	0.00288
ph	0.00287
director	0.00282
fax	0.00279
journal	0.00271
recent	0.00260
received	0.00258
award	0.00250

Students	
resume	0.00516
advisor	0.00456
student	0.00387
working	0.00361
stuff	0.00359
links	0.00355
homepage	0.00345
interests	0.00332
personal	0.00332
favorite	0.00310

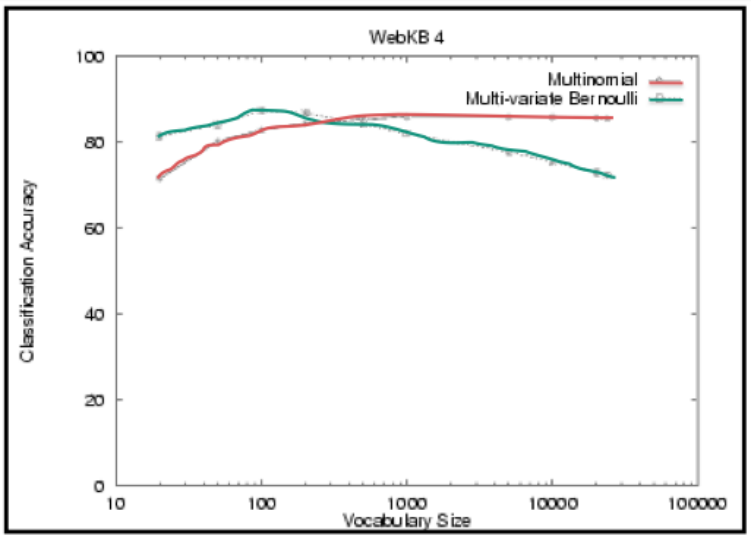
Courses	
homework	0.00413
syllabus	0.00399
assignments	0.00388
exam	0.00385
grading	0.00381
midterm	0.00374
pm	0.00371
instructor	0.00370
due	0.00364
final	0.00355

Departments	
departmental	0.01246
colloquia	0.01076
epartment	0.01045
seminars	0.00997
schedules	0.00879
webmaster	0.00879
events	0.00826
facilities	0.00807
eople	0.00772
postgraduate	0.00764

Research Projects	
investigators	0.00256
group	0.00250
members	0.00242
researchers	0.00241
laboratory	0.00238
develop	0.00201
related	0.00200
arpa	0.00187
affiliated	0.00184
project	0.00183

Others	
type	0.00164
jan	0.00148
enter	0.00145
random	0.00142
program	0.00136
net	0.00128
time	0.00128
format	0.00124
access	0.00117
begin	0.00116

Confronto dei modelli NB: WebKB



all'aumentare dei termini selezionati con MI, il mod. binomiale, a differenza del multinomiale, peggiora la sua efficacia

Naive Bayes vs. altri metodi

(a)

	NB	Rocchio	kNN	SVM
micro-avg-L (90 classes)	80	85	86	89
macro-avg (90 classes)	47	59	60	60

(b)

	NB	Rocchio	kNN	trees	SVM
earn	96	93	97	98	98
acq	88	65	92	90	94
money-fx	57	47	78	66	75
grain	79	68	82	85	95
crude	80	70	86	85	89
trade	64	65	77	73	76
interest	65	63	74	67	78
ship	85	49	79	74	86
wheat	70	69	77	93	92
corn	65	48	78	92	90
micro-avg (top 10)	82	65	82	88	92
micro-avg-D (118 classes)	75	62	n/a	n/a	87

Evaluation measure: F_1

Naive Bayes does pretty well, but some methods beat it consistently (e.g., SVM).

Naive Bayes & Selezione di Feature: Conclusioni

- efficiente sia nella fase di learning che in quella di test (lineare rispetto alla quantità di testo)
- Bassa quantità di dati utilizzati, soprattutto il modello binomiale con selezione di feature
- più efficace in domini con numerose feature ugualmente importanti (non è il migliore in generale nella text classification)
- Più robusto di altri metodi con feature irrilevanti
- Resistente al concept drift (variazioni nelle classi nel tempo)
- Competizione KDD-CUP 97 vinta con soluzioni basate su Naive Bayes: 1° e 2° posto su 16 partecipanti

  Dominio: risposta a email pubblicitarie in ambito finanziario
  Obiettivo: predire se il destinatario risponderà (750.000 record)
- Con il modello binomiale la selezione di feature è necessaria

LATENT SEMANTIC ANALYSIS

Spazio dei doc a vettori: Punti di Forza

- Matching lessicale tra termini
  - matching parziale tra ricerche e documenti, quando nessun documento contiene esattamente tutti i termini ricercati
- Ranking dei risultati della ricerca in base a misure di similarità (e.g. similarità coseno)
  - il ranking consente di gestire risultati con ampio numero di documenti
- Rappresentazione di termini e doc con diverse modalità di term weighting (frequenza binaria, occorrenze, tf-idf, ...)
- Supporta
  - clustering di documenti
  - rilevanza con feedback degli utenti
- Fondato sulla geometria

Spazio dei doc a vettori: Limiti (i)

- Problemi semantici nei linguaggi naturali
  - *Jaguar:* car ? football team ? animal ?
- **Polisemia**:
  - parole con **diversi significati** che dipendono dal contesto
  - maggiore è l'eterogeneità semantica dei documenti, più evidente diventa il problema
- La modellazione di doc con lo spazio a vettori non riconosce i diversi significati di una stessa parola
  - **problema:** la similarità coseno, in caso di polisemia, è maggiore della similarità reale
  - → restituzione di doc irrilevanti → diminuzione della precision

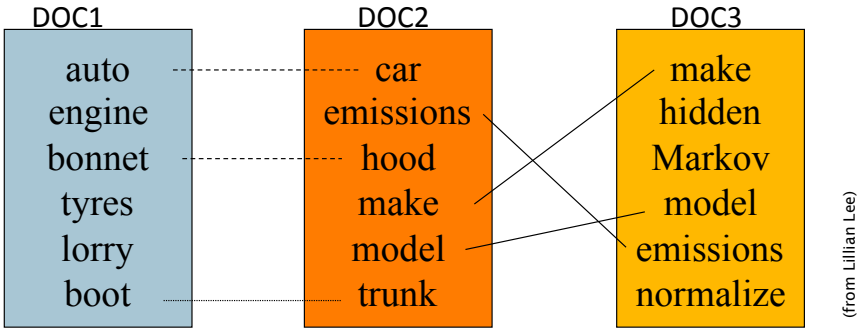
$$\text{sim}_{\text{true}}(d, q) < \cos(\angle(\vec{d}, \vec{q}))$$

Spazio dei doc a vettori: Limiti (ii)

- Sinonimia:
  - termini lessicalmente distinti con **significato identico o simile**
  - *Ship, boat*: hanno analogo significato ?
- la rappresentazione dei doc nel modello a vettori non contempla *associazioni* semantiche tra termini
  - e.g. nessuna associazione tra *tree* e *wood* poiché lessicalmente distinti
- problema: con termini distinti ma *sinonimi* o *associati* tra query e doc, la similarità coseno è inferiore a quella reale
  - → doc rilevanti non sono recuperati → diminuzione della recall

$$\text{sim}_{\text{true}}(d, q) > \cos(\angle(\vec{d}, \vec{q}))$$

Esempio del Problema



(from Lillian Lee)

Sinonimia

similarità coseno nulla
tra DOC1 e DOC2
benché siano
semanticamente simili

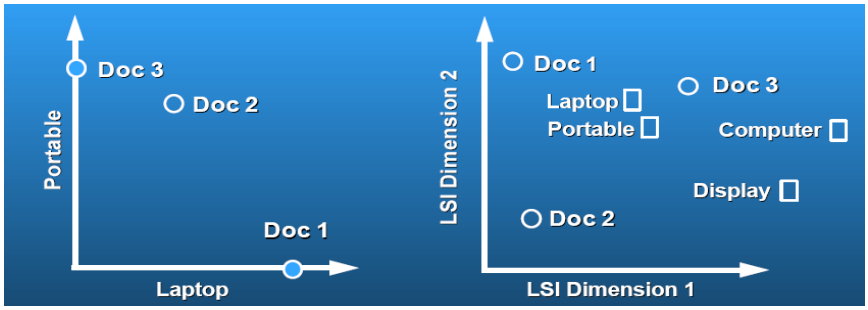
Polisemia

similarità coseno >0
tra DOC2 e DOC3
ma semanticamente
del tutto dissimili

Latent Semantic Analysis/Indexing

- Trasforma la matrice termini doc facendo emergere associazioni semantiche latenti tra termini e doc
  - mapping della matrice in un spazio a vettori ridotto che approssima quello originale trascurando dettagli, compreso l'eventuale "rumore"
  - **associazioni semantiche latenti**: associazione di ordine superiore non basate sul match lessicale tra termini
- Nello spazio trasformato, termini semanticamente simili o associati sono in posizioni limitrofe
  - la similarità semantica tra termini lessicalmente distinti emerge grazie alla co-occorrenza dei termini in doc diversi
- *Latent Semantic Indexing:* lo spazio trasformato è usato nell'IR come indice per ricerche per similarità

Latent Semantic Analysis: Intuizione



courtesy of Susan Dumais

- Spazio originale termini-doc (fig. a sinistra)
  - Doc1 contiene il termine **laptop**, ma non **portable** e viceversa nel Doc3 → ciò contribuisce a rendere i 2 doc dissimili
  - Doc2 contiene entrambi i termini
- Spazio trasformato con nuove dimensioni: (fig. a destra)
  - grazie a **Doc2,** emerge la similarità semantica latente tra **laptop** e **portable** e tra **doc1** e **doc3** → più vicini con similarità coseno

Trasformazione con Singular Value Decomposition (SVD): Richiami di Algebra

- il rango *r* di una matrice è il numero di vettori righe o colonne linearmente indipendenti
- autovettori di una matrice *A* quadrata MxM:
  - *v* è autovettore (destro) di *A* se $Av = \lambda v$ dove λ è l'autovalore di *v*
  - *A* ha al più M autovalori, i.e. le soluzioni del polinomio di grado M in λ, $(A-\lambda I)v = 0$, $v \neq 0$ → $A-\lambda I = 0$ (Soluzione con moltiplicatori di Lagrange)
- la matrice *A* con rango *r* ha *r* autovettori *ortonormali*
  - vettori con norma unitaria e ortogonali tra loro
- gli autovettori ortonormali sono una base per i vettori di *A*
  - autovettori destri: sono una base per i vettori colonne di *A*
  - autovettori sinistri: base per i vettori righe di *A*
- una base ortonormale consente di def. un sistema di riferimento

Trasformazione con Singular Value Decomposition (SVD): Fattorizzazione

- Per una matrice *C* MxN di rango *r* esiste una fattorizzazione SVD come segue:

$$\underset{MxN}{C} = \underset{MxM}{U}\ \underset{MxN}{\Sigma}\ \underset{NxN}{V^T}$$ con Σ matrice diagonale

- Esaminiamo la matrice MxM $CC^T=(U\Sigma V^T)\,(V\Sigma^T U^T)=U\Sigma V^T V\Sigma^T U^T$ $=U\Sigma\Sigma^T U^T=U\Sigma^2 U^T \rightarrow U\Sigma^2 U^T U=CC^T U \rightarrow U\Sigma^2 U^T U=CC^T U \rightarrow CC^T U=\Sigma^2 U$ → U contiene gli autovettori destri di CC^T, i.e. le colonne di U sono gli autovettori di CC^T e Σ^2 contiene i relativi autovalori λ
- Analogamente le colonne di V sono gli autovettori di C^TC
- il prodotto scalare di 2 vettori è una misura di similarità tra essi → CC^T è una matrice quadrata di similarità
- → U e V basi ortonormali di un nuovo sistema di riferim. e Σ i cui valori $\sigma = \sqrt{\lambda}$ *indicano la variabilità dei dati per ogni dimen.*

Esempio di Fattorizzazione SVD

$$C=\begin{pmatrix}1&-1\\0&1\\1&0\\-1&1\end{pmatrix} \qquad CC^T=\begin{pmatrix}2&-1&1&-2\\-1&1&0&1\\1&0&1&-1\\-2&1&-1&2\end{pmatrix} \qquad C^TC=\begin{pmatrix}3&-2\\-2&3\end{pmatrix}$$

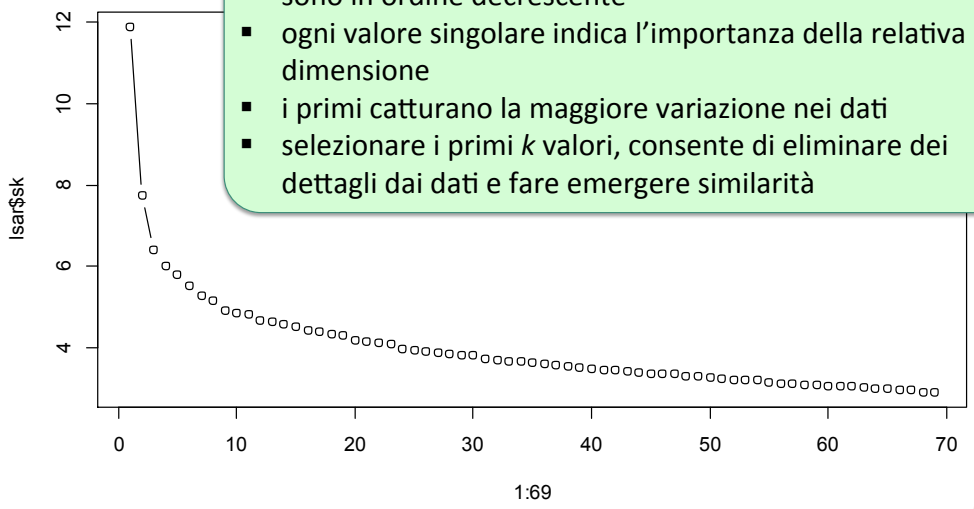
$$\Sigma^2=\begin{pmatrix}5&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix} \qquad U=\begin{pmatrix}-.632&0&-.774&.023\\.316&.707&-.276&-.569\\-.316&.707&.276&.569\\.632&0&-.498&.593\end{pmatrix} \qquad V=\begin{pmatrix}-.707&.707\\.707&.707\end{pmatrix}$$

autovalori di CC^T e C^TC = varianza dei dati per ogni asse del nuovo sistema di riferimento

autovettori di CC^T e C^TC

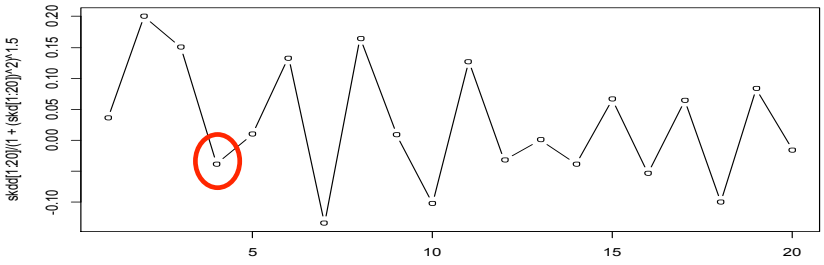
Valori Singolari della Matrice Σ

- i valori singolari in Σ, componenti delle nuove dimensioni, sono in ordine decrescente
- ogni valore singolare indica l'importanza della relativa dimensione
- i primi catturano la maggiore variazione nei dati
- selezionare i primi *k* valori, consente di eliminare dei dettagli dai dati e fare emergere similarità

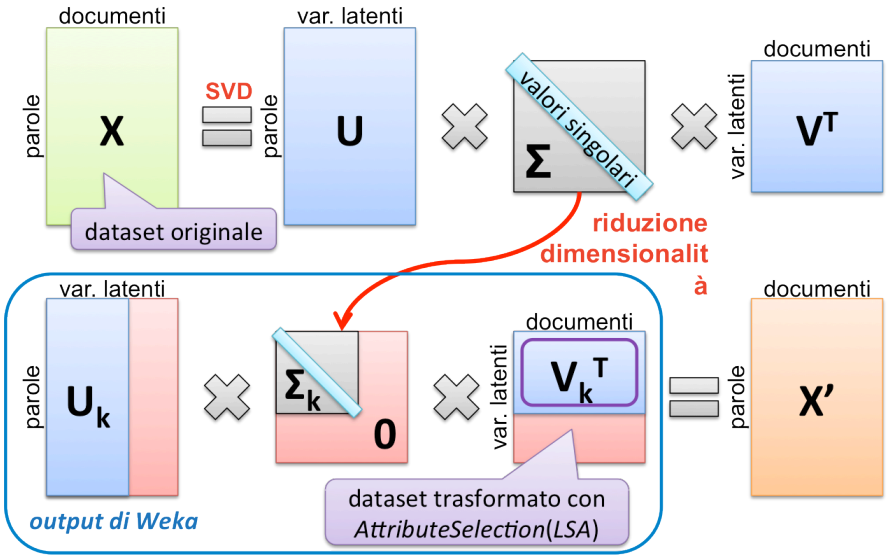


Quanti Valori Singolari Selezionare ?

- Un punto di knee è dove il raggio di curvatura della funzione che interpola l'iperbole (e.g. vedi slide prec.) è minimo locale
  - La curvatura di una funzione y = f(x) è k = y''/(1+(y')^2)^(3/2)
  - e.g. il primo minimo locale è 4 → selezioniamo i primi 4 valori singolari, i.e. dimensioni (i.e. rango), con cui approssimare la matrice originale → riduzione della dimensionalità
  - Il num. migliore di dimensioni dipende dai dati e dal num. di variabili dello spazio originale



SVD applicata alla Matrice Termini-Doc: Riduzione della Dimensionalità



Approssimazione Low-rank

- data una matrice *C* MxN e un intero positivo *k*, determinare la matrice C_k MxN di rango al più *k* che minimizzi la differenza X=C-C_k secondo la norma di Frobenius seguente

$$\|X\|_F = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} X_{ij}^2}.$$

- se *r* è il rango di *C*, allora C_r=*C* e la norma di Frobenius della matrice differenza è zero.
- quando *k* è inferiore ad *r*, C_k è l'approssimazione low-rank ottimale
  - i.e. non esiste C_k' t.c. $||C - C_k'||_F < ||C - C_k||_F$

LSA: Similarità tra Termini, Doc e Query-Doc

- sia $X=U\Sigma V^T$ la matrice originale termini-doc fattorizzata in SVD
- $X_k = U\Sigma_k V^T$ è la matrice termini-doc ricostruita con rango k
- La similarità tra coppie di doc o termini si misura con il coseno del prodotto scalare come segue:
  - similarità tra doc $X_k^T X_k = (U\Sigma_k V^T)^T(U\Sigma_k V^T) = V\Sigma_k^T U^T U\Sigma_k V^T = V\Sigma_k^2 V^T = (V\Sigma_k)(V\Sigma_k)^T$ → similarità tra doc v_i e v_j = coseno$(v_i\Sigma_k, v_j\Sigma_k) = (v_i\Sigma_k)(v_j\Sigma_k)^T / (\|v_i\Sigma_k\| \|v_i\Sigma_k\|)$
  - idem similarità tra termini $X_k X_k^T = (U\Sigma_k V^T)(U\Sigma_k V^T)^T = U\Sigma_k V^T V\Sigma_k^T U^T = U\Sigma_k^2 U^T$
- La similarità tra una query q (i.e. vettore di termini in X) e un doc si calcola trasformando q in un nuovo doc q_k $q_k = \Sigma_k^{-1} U_k^T q$
  - la trasformazione di q nel doc q_k segue da $U^T X_k = U^T U\Sigma_k V^T$ → $U^T X_k = \Sigma_k V^T$ → $\Sigma_k^{-1} U^T X_k = \Sigma_k^{-1}\Sigma_k V^T$ → $\Sigma_k^{-1} U^T X = V_k^T$
- similarità tra q_k e un doc v_i: coseno$(q_k\Sigma_k, v_i\Sigma_k)$=coseno$(qU_k, v_i\Sigma_k)$

LSA: Similarità tra un Termine ed un Doc

- sia $X_k=U\Sigma_k V^T$ la matrice X termini-doc fattorizzata con SVD a rango ridotto k
- ogni $x_{ij} \in X_k$ è l'associazione tra il termine u_i e il doc v_j è dato dal prodotto scalare $x_{ij} = u_i \Sigma_k v_j = (u_i \Sigma_k^{½})(\Sigma_k^{½} v_j)$
- perciò la **similarità tra il termine u_i e il doc v_j** è data da coseno$(u_i \Sigma_k^{½}, \Sigma_k^{½} v_j)$
- Analogamente la **similarità tra un termine u_i ed una query q** di termini corrisponde alla **similarità tra il termine u_i e il doc q_k** ottenuto trasformando q come nella slide precedente:
  - q è trasformata nel doc $q_k = \Sigma_k^{-1} U_k^T q = v_j$ → $u_i \Sigma_k^{½}\Sigma_k^{½} v_j = u_i \Sigma_k^{½}\Sigma_k^{-½} U_k^T q$
  - la similarità tra q e il termine u_i è coseno$(u_i \Sigma_k^{½}, \Sigma_k^{-½} U_k^T q)$

LSA: Esempio

- Matrice termini-documenti (binaria)

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

LSA Esempio: Matrice U dei termini

- $C = U\Sigma V^T$: matrice U

U	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

LSA Esempio: matrice Σ

- C = UΣV^T: matrice Σ

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

LSA Esempio: Matrice V

- C = UΣV^T: matrice V^T

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

LSA Esempio: Riduzione delle dim. a 2

U	1	2	3	4	5
ship	−0.44	−0.30	0.00	0.00	0.00
boat	−0.13	−0.33	0.00	0.00	0.00
ocean	−0.48	−0.51	0.00	0.00	0.00
wood	−0.70	0.35	0.00	0.00	0.00
tree	−0.26	0.65	0.00	0.00	0.00

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

Matrice Originale C e Ridotta C_2: Similarità tra Termini

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

C: similarità ship e boat = **0.0**
C_2: similarità=0.52

ship•boat = 0.85 * 0.36 + 0.52 * 0.36 + 0.28 * 0.16 + 0.13 * -.20 + 0.21 * −.02 + -.08 * -.18 ≈ 0.52

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

Matrice *C* originale e Ridotta C_2: Similarità tra Documenti

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

C: similarità tra d2 e d3 = **0.0**
C_2: similarità = 0.52

d2 •d3 = 0.52 ∗ 0.28 + 0.36 ∗ 0.16 + 0.72 ∗ 0.36 + 0.12 ∗ 0.20 + −0.39 ∗ −0.08 ≈ 0.52

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

Esempio di Query Senza e con LSA

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

$$U_2=\begin{pmatrix}-0.44 & -0.3\\ -0.13 & -0.33\\ -0.48 & -0.51\\ -0.7 & 0.35\\ -0.26 & 0.65\end{pmatrix} \qquad \Sigma_2=\begin{pmatrix}2.16 & 0\\ 0 & 1.59\end{pmatrix}$$

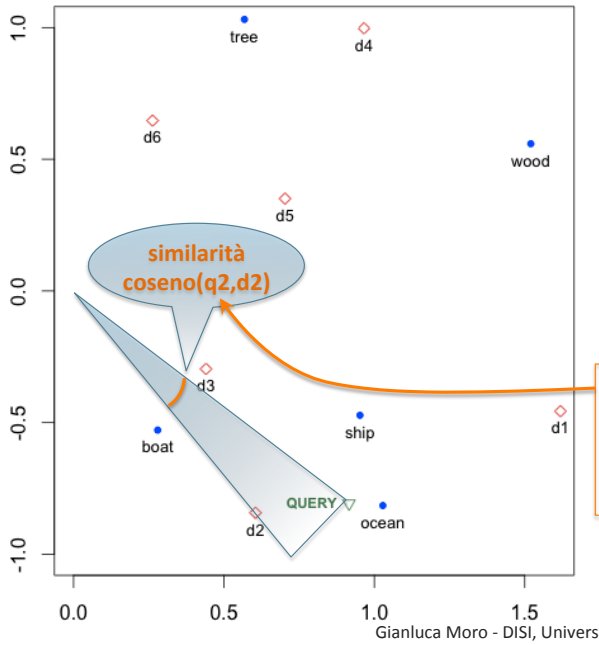
V_2 =

	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41

$$\Sigma_2^{-1}=\begin{pmatrix}2.16^{-1} & 0\\ 0 & 1.59^{-1}\end{pmatrix}$$

- Query = "ship ocean" → q = [1 0 1 0 0]^T
- trasformazione di q in $\vec{q}_k = \Sigma_k^{-1} U_k^T \vec{q}.$ → q_2 = [-0.43 -0.51]^T
- **Similarità tra q e d_2 in *C*: coseno(q, d_2) = 0.5**
- **Similarità con LSA tra q_2 e d_2 di V_2: coseno($q_2\Sigma_2$, $d_2\Sigma_2$) = 0.97**

Esempio: Termini-Doc, Query e Similarità



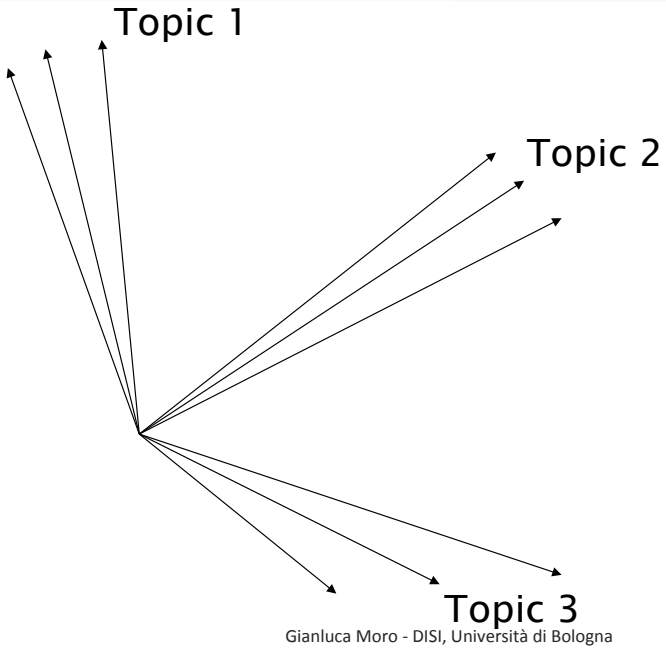
q = "**ship ocean**"

	C	d_1	d_2	d_3	d_4	d_5	d_6
1	ship	1	0	1	0	0	0
0	boat	0	1	0	0	0	0
1	ocean	1	1	0	0	0	0
0	wood	1	0	0	1	1	0
0	tree	0	0	0	1	0	1

q2="ship ocean"

	C_2	d_1	d_2	d_3	d_4	d_5	d_6
0.64	ship	0.85	0.52	0.28	0.13	0.21	−0.08
0.39	boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
0.85	ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
0.36	wood	0.97	0.12	0.20	1.03	0.62	0.41
-0.28	tree	0.12	−0.39	−0.08	0.90	0.41	0.49

LSA: Effetto della Fattorizzazione SVD



LSA: Applicazioni

- Collaborative Filtering e Recommendation System
  - comprendere e predire gli interessi di un insieme di utenti a partire da una massa di dati delle loro scelte (prodotti, servizi etc.)
  - sistemi intelligenti che propongono prodotti, servizi etc. ad utenti (e.g. amazon, findory,)
  - metodo utilizzato nella soluzione che ha vinto il milione di dollari per predire quali film vedranno un insieme di persone a partire da un insieme di film già visti (www.netflix.com)
- Opinion Mining & Sentiment Analysis
  - e.g. se la matrice fattorizzata con LSA contiene opinioni e utenti è possibile raggruppare gli utenti in base alle opinioni
- Sistemi di **comprensione** del linguaggio naturale
- Analogo, in parte, a metodi di clustering

LSA: Limiti

- Non cattura la polisemia, i.e. più significati di una parola
  - ogni occorrenza di una parola è trattata sempre con lo stesso significato poiché si rappresenta come un singolo punto nello spazio
  - Esempio: l'occorrenza di chair in un doc contenente "*The Chair of the Board*" e in un altro doc contenente "*the chair maker*" sono trattate con lo stesso significato
  - Il risultato è un vettore che rappresenta la media dei diversi significati della parola nel corpus
  - questo effetto negativo spesso è alleviato perché in un corpus per ogni parola predomina un significato rispetto ad altri
- Il costo computazionale è quello di SVD su una matrice m x n
  - $O(\min\{mn^2, m^2n\})$: 2 metodi, se $n<<m \rightarrow O(mn^2)$ altrimenti $O(m^2n)$
- Espressività limitata dalla rappresentazione bag of words

Esempio Precedente in **R** (i)

```
install.packages("lsa"); library(lsa) # installa e importa LSA

# creazione della matrice termini documenti tdm_ship_boat
tdm_ship_boat = matrix(c(1,0,1,0,0,0,  0,1,0,0,0,0,  1,1,0,0,0,0, 1,0,0,1,1,0,
0,0,0,1,0,1), nrow=5, ncol=6, byrow = TRUE, dimnames = list(c("ship", "boat",
"ocean", "wood", "tree"), c("d1", "d2", "d3","d4","d5","d6")))

# applicazione LSA con rango ridotto 2 a tdm_ship_boat:
# genera 3 matrici: termini tk, autovalori sk, documenti dk
tdm_ship_boat_lsa = lsa(tdm_ship_boat,2)

# matrice dei termini tk scalata rispetto agli autovalori in sk per
# posizionare i termini nel grafico termini e doc; %*% prodotto matriciale
tdm_ship_boat_lsa_terms = tdm_ship_boat_lsa$tk %*% diag(tdm_ship_boat_lsa$sk)

# matrice dei doc dk scalata rispetto agli autovalori in sk
# per posizionare i doc nel grafico termini e doc
tdm_ship_boat_lsa_docs = tdm_ship_boat_lsa$dk %*% diag(tdm_ship_boat_lsa$sk)

# estrazione dei termini per aggiungerli nel grafico termini e doc in
# corrispondenza alla loro posizione
termNames_ship_boat = rownames(tdm_ship_boat)

# estrazione dei nomi dei doc per aggiungere nel grafico in corrispondenza
# alla posizione di ciascun doc
docsNames_ship_boat = colnames(tdm_ship_boat)
```


Esempio Precedente in **R** (ii)

```
# grafico delle posizioni dei termini; pch indica la forma di ciascun punto,
# xlim e ylim fissano l'intervallo degli assi
plot(tdm_ship_boat_lsa_terms,pch=20,col="blue",xlim=c(0.0, 1.6),ylim=c(-1,1))

# aggiunge nel grafico i termini, cex dimensione del font, pos=1
# posiziona il nome sotto il punto
text(tdm_ship_boat_lsa_terms, labels=termNames_ship_boat, cex=0.8, pos=1)

points(tdm_ship_boat_lsa_docs, pch=23, col="red") # aggiunge le posiz. dei doc

# aggiunge al grafico i nomi dei doc
text(tdm_ship_boat_lsa_docs, labels=docsNames_ship_boat, cex=0.8, pos=1

# QUERY "ship ocean" nello spazio originale, i.e. matrice termini doc iniziale
# generazione del vettore query q dallo spazio originale
q = query("ship ocean", rownames(tdm_ship_boat))

# similarità coseno tra q e il doc d2 nello spazio orignale; [,2]→colonna 2
#     as.vector(q) converte q in vettore, il tipo dato richiesto da cosine()
cosine(as.vector(q), as.vector(tdm_ship_boat[,2])) # restituisce 0.5

# Stessa query, ma nella matrice termini doc ricostruita con LSA
tdm_ship_boat_ricostruita = as.textmatrix(tdm_ship_boat_lsa)

# trasforma q in un doc nella matrice ricostruita e ricalcola la similarità
ship_ocean = fold_in(q,tdm_ship_boat_lsa) # cosine() seguente restituisce 0.97
cosine(as.vector(ship_ocean), as.vector(as.vector(tdm_ship_boat_ricostruita[,2])))
```


Esempio Precedente in **R** (iii)

```
# trasforma la query q nel doc q2 appartenente a V = dk (q2 = Σk^-1 U^T q)
q2 = diag(tdm_ship_boat_lsa$sk^-1) %*% t(tdm_ship_boat_lsa$tk) %*% q

# estrazione del vettore del doc d2 da LSA (seconda riga matrice dk)
d2=tdm_ship_boat_lsa$dk[2,]

# VISUALIZZAZIONE della posizione della query q2 e della relativa etichetta
# nel grafico LSA dei termini e doc; font=2 per il grassetto
points(t(q2) %*% diag(tdm_ship_boat_lsa$sk), pch=25, col="seagreen")
text(t(q2) %*% diag(tdm_ship_boat_lsa$sk),labels="QUERY",font=2,cex=0.7,pos=2, col="green")

# Calcolo similarità coseno tra q e d2 con LSA usando cos(q2Σ2, d2Σ2) invece che
# fold_in()
cosine(as.vector(t(q2)%*%diag(tdm_ship_boat_lsa$sk)),as.vector(d2%*%diag(tdm_ship_boat_lsa
$sk)))

# Restituisce termini simili a "boat" in ordine decrescente di similarità
# coseno fino a 0.5 → ocean 0.91, ship 0.81 anche se nessun doc contiene
# entrambi i termini boat e ship
associate(tdm_ship_boat_lsa_terms, "boat", threshold = 0.5)

# Nello matrice originale termini-doc lo stesso comando restituisce solo
# ocean 0.71
associate(tdm_ship_boat_lsa_terms, "boat", threshold = 0.5)

# Come sopra ma con "tree"→ wood 0.75, ma nessun risultato nella matrice orig.
associate(tdm_ship_boat_lsa_terms, "tree", threshold = 0.5)
```