

L'idea dell'information retrieval è rappresentare in maniera vettoriale le informazioni, tipo un documento di una collisione viene rappresentato con un vettore. Ogni componente del vettore è legata ad un termine (parola chiave) associato al termine (contiamo quante volte una certa parola chiave compare nel documento, quindi parliamo di frequenza.)

Linear Algebra and information retrieval

vector representation of information

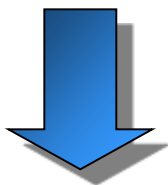
a **document** (web site, book,...) of a
collection (Internet, library,...) is
represented as vector

each **component** of the vector is tied
to a **term** (key-word) associated to
the document

Qui fa un esempio con il documento a e i tre termini, allora il valore associato a una componente riflette l'importanza del termine nella semantica del documento. Come detto prima in pratica stiamo definendo una funzione di frequenza dell'occorrenza del termine nel documento.

example:

document *A*



vector *a*

terms (key-words):

informatics, applied,
geomatics



three components

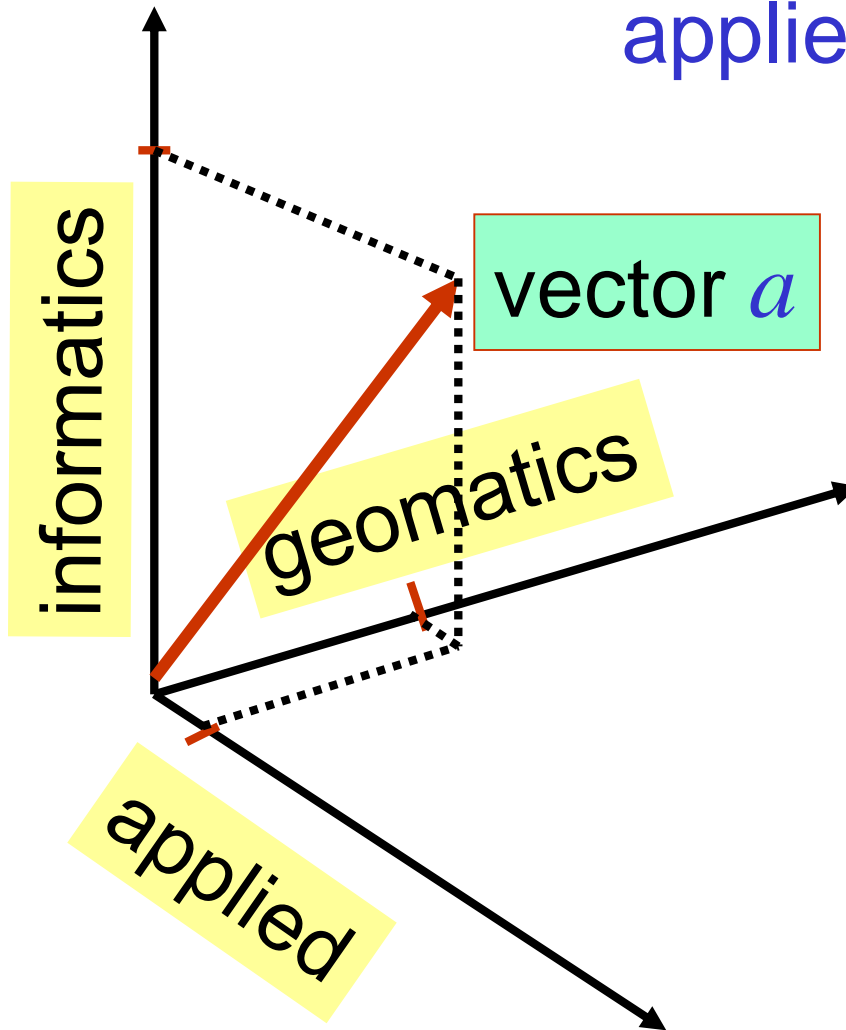
the **value** of a component depends on the
importance of the term in the
semantics of the document

function of the frequency of occurrence of
the term inside the document

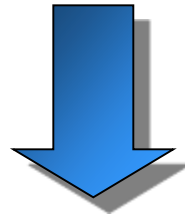
Queste sono tipo le misure allora il documento A è un vettore a tre dimensioni, chiaramente avremo tanti vettori quanti sono i documenti

document *A*

informatics (5.0),
geomatics (2.5)
applied (0.5),



- a **database** contains d **documents**
- each document is described by t **terms**



B term-documents matrix

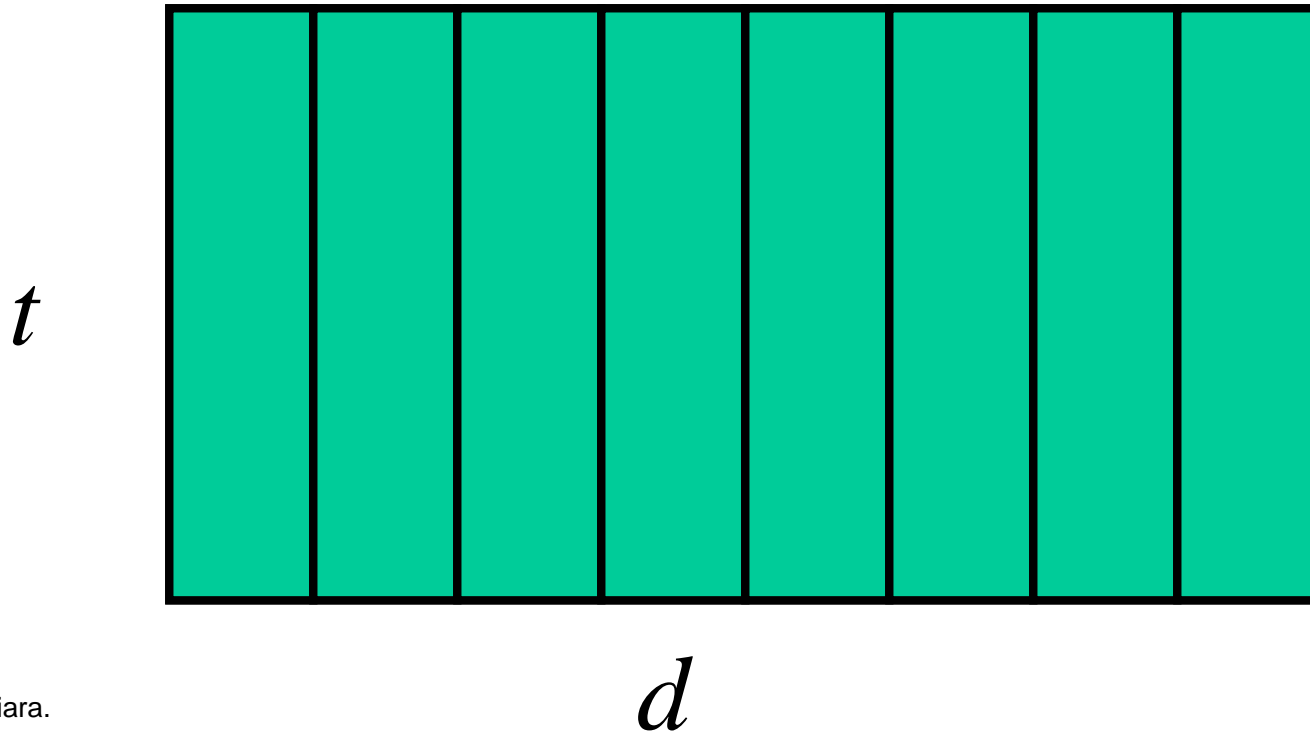
B is a $t \times d$ matrix

documents = columns of the matrix B
terms = rows of the matrix B

B is $t \times d$

B term-documents matrix

documents = columns of B



Questa frase è chiara.

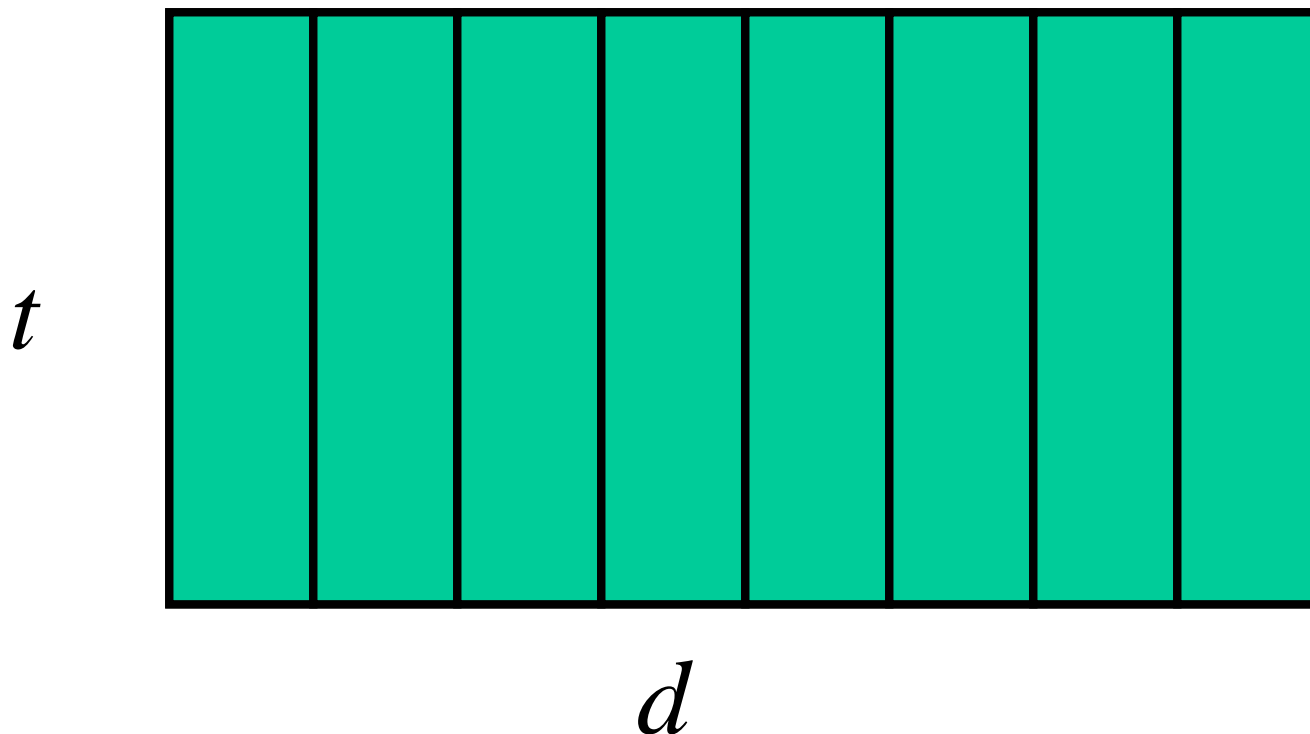
b_{ij} is the frequency of occurrence of the i -th **term** in the j -th **document**

B è sparsa perché in generale ci sono tanti termine.

B term-documents matrix

B is $t \times d$

documents = columns of B



B is sparse

$$B = (b_1, b_2, \dots, b_d)$$

Internet (2020):

B order of 900000 x 50E9 (terms x web pages)

Questo vettore riga t_i^T ci dice il termine i -esimo compare nei documenti.

Facendo il prod scalare della riga i -esima e la riga p di fatto è la correlazione tra due termini del documento. $B \cdot B^T$ contiene tutti i prodotto scalari e la matrice ottenuta è la matrice di covarianza dei termini.

B term-documents matrix

B is $t \times d$

t_i^T							
t_p^T							

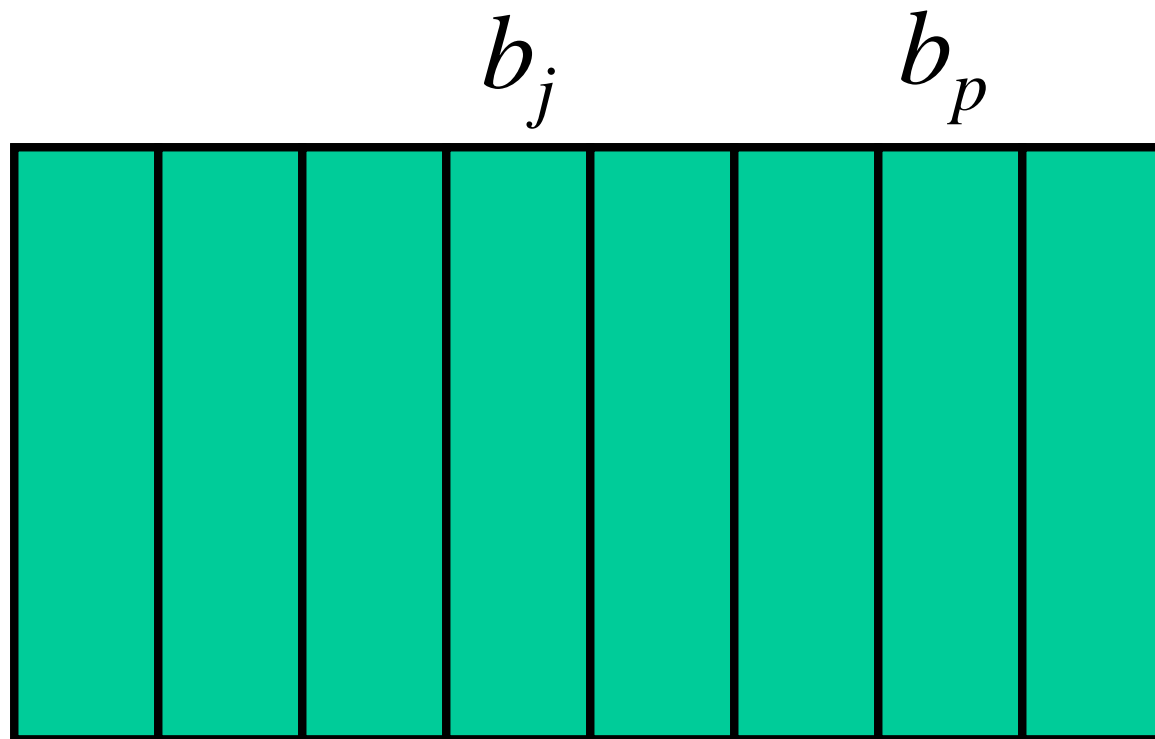
$t_i^T t_p^T$ **correlation** between 2 terms

BB^T contains all such scalar products
(**covariance** matrix of terms)

Se invece facciamo il prod scalare tra due colonne otteniamo la correlazione tra 2 documenti. Il prod da la trasposta di B e V contiene tutti i prodotti scalari infatti è detta covariance matrix of documents.

B term-documents matrix

B is $t \times d$



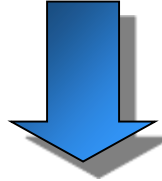
$b_j^T b_p$ **correlation** between 2 documents

$B^T B$ contains all such scalar products
(**covariance matrix of documents**)

Lo spazio delle colonne di B descrive tutto il db e sfruttando le relazioni geometriche tra i vettori ovvero i documenti, possiamo individuare somiglianze e differenze di contenuto.

the **column space** of B describes the whole **database**

Fare una query vuol dire richiedere tramite un insieme di termini di trovare documenti del DB che sono rilevanti rispetto a quei termini.



Exploit the **geometrical relations** among **vectors** (documents) to detect **similarities** and **differences** of **content**

making a query means:
request, through a set of **terms**, to find database **documents** that are relevant to those terms

In ogni caso l'idea è fare le query per trovare/recuperare i documenti e quindi le informazioni, ma chiaramente anche la query va espressa in un modo che il pc possa capirlo.

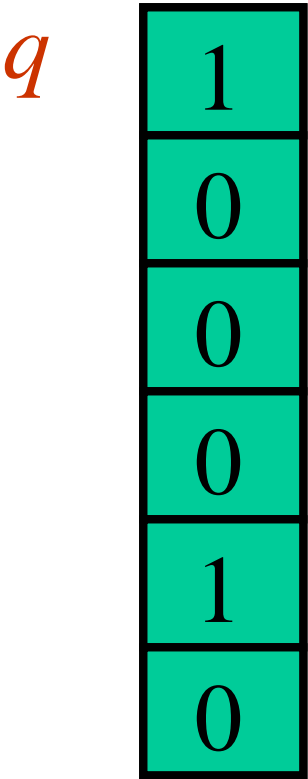
L'idea è che poiché la query è un insieme di termini di fatto è anch'essa un vettore all'interno di questo spazio, vettore q di t componenti. Settiamo 0 in corrispondenza dei termini che non ci interessano. E' quindi una matrice sparsa perché ha molti 0

a **query** may be represented as a **vector** q
of t components

basic idea: represent a **query**
as a **document**

Qui per esempio stiamo richiedendo i documenti che contengono il primo e il quinto termine.

is a sparse
vector



query that requires
documents that
contain the **first** and
fifth terms (of all
terms)

Fare il matching vuol dire trovare i documenti più simili alla query, quindi trovare i vettori-documenti geometricamente più vicino al vettore-query. Quindi stiamo parlando di trovare una misura di somiglianza matematica/geometrica, tipo abbiamo visto il coseno dell'angolo tra il vettore query e i vettori documenti. L'output è 1 se ha somiglianza massima, 0 somiglianza minima cioè ortogonale.

query matching = find the documents more similar to the query

find the document-vectors geometrically closest to the query-vector (respect to a chosen measure)

measures of similarity

cosine of the angle between the query-vector and the document-vectors

1 = maximum similarity

0 = minimum similarity (orthogonality)

Questo è quello che si dovrebbe fare. b_j è la colonna j -esima quindi $\cos j$ è la misura di somiglianza della query q con il j -esimo documento cioè colonna b_j .

for each **query** q compute d cosines

$$\cos \theta_j = \frac{b_j^T q}{\|b_j\|_2 \|q\|_2} = \frac{b_j^T q}{\sqrt{b_j^T b_j} \sqrt{q^T q}}$$

questa è la norma 2, detta anche euclidea.

$$\|v\|_2 = \sqrt{v^T v} = \sqrt{\sum_{i=1}^t v_i^2}$$

2-norm
(length)

6 terms ($t = 6$):	
t1: baked t2: recipe t3: bread	t4: cake t5: pastry t6: dessert

- 5 documents ($d = 5$):
- d1: Baked bread without recipe
 - d2: The art of classical Viennese pastry
 - d3: Numerical recipes: the art of scientific calculation
 - d4: Bread, pastry, desserts and cakes: precise recipes for baking
 - d5: Pastry: the book of the best French recipes

term-documents matrix is 6x5

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

baked
appears in
documents **d1**
and **d4**

document **d4** contains
all the terms

normalize the term-documents matrix

$$B = \begin{pmatrix} 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0.5774 & 0 & 1 & 0.4082 & 0.7071 \\ 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0 & 0 & 0 & 0.4082 & 0 \\ 0 & 1 & 0 & 0.4082 & 0.7071 \\ 0 & 0 & 0 & 0.4082 & 0 \end{pmatrix}$$



each column has
length 1

Exemple of query matching:

query: **baked bread**

$$q^{(1)} = (1, 0, 1, 0, 0, 0)^T$$

↑ ↑
baked **bread**

Data la query con la sua rappresentazione, devo calcolare i coseni degli angoli e selezionare i vettori documento con coseno dell'angolo maggiore di una certa soglia.

search of relevant documents:

- compute the **cosine** of the angle among the **query-vector** and the **document-vectors**
- select the document-vectors with cosine **greater than 0.5**

Anche il vett query andrebbe normalizzato, ma in questa fase non lo fa in quanto non siamo interessati ai termini ma solo ai documenti.

$$q^{(1)} = (1, 0, 1, 0, 0, 0)^T$$

$\cos \theta_1 = 0.8165$	similarity query and first document
$\cos \theta_2 = 0$	similarity query and second document
$\cos \theta_3 = 0$	similarity query and third document
$\cos \theta_4 = 0.5774$	similarity query and fourth document
$\cos \theta_5 = 0$	similarity query and fifth document

query: **baked bread**

$$\cos \theta_1 = 0.8165$$

d1: Baked bread without recipe

$$\cos \theta_4 = 0.5774$$

d4: Bread, pastry, desserts and cakes: precise recipes for baking

document **d1** is more relevant than document **d4**; documents **d2**, **d3**, **d4** **are not** relevant

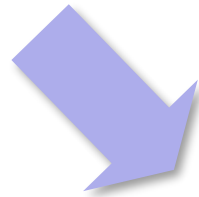
query: **baked**

$\cos \theta_1 = 0.5774$	similarity query and first document
$\cos \theta_2 = 0$	similarity query and second document
$\cos \theta_3 = 0$	similarity query and third document
$\cos \theta_4 = 0.4082$	similarity query and fourth document
$\cos \theta_5 = 0$	similarity query and fifth document

Noi però dovremmo ridurre alcuni problemi di questi DB. Quindi dovremmo ridurre il rango della matrice B , in modo da eliminare questi problemi che non sono geometrici. L'unico modo per fare questo che noi conosciamo è fare la fattorizzazione QR della matrice B

reducing the
uncertainty in the
database of
documents

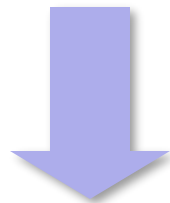
reduce the size of
the matrix that
represents the data-
base of documents



rank reduction
of the term-documents matrix B

QR factorization of the matrix B

rank reduction of the term-documents matrix B



detect and eliminate
redundant information in the
representation of the database

example of source of redundancy:

- ❖ mirror in Internet
- ❖ different editions of the same book in a library

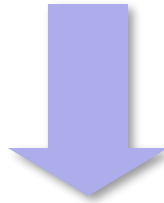
edizioni diverse di uno stesso libro in biblioteca.

La prima cosa da fare è individuare le dipendenze tra le varie colonne di B , cioè individuare una base ortogonale per lo spazio delle colonne di B , cioè se B ha rango r_B allora bastano r_B vettori per poter rappresentare tutto il range quindi tutte le colonne della matrice B cioè i vettori documenti.

first step:

detect dependency among columns of B

find an **orthogonal basis** of the **column space** of B ($range(B)$)



if B has rank r_B , then r_B **basis-vectors** of the column space of B may substitute the d column vectors for representing such space

Il modo per trovare una base per lo spazio delle colonne di B è calcolare la sua fattorizzazione QR.

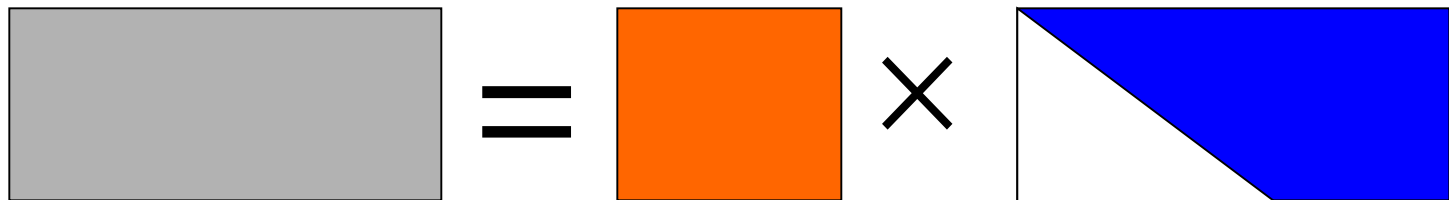
a **basis** of the column space of B is provided by the QR factorization of B

$$B = QR$$

R è triangolare superiore $t \times d$

Q è ortogonale $t \times t$

- R is $t \times d$ upper triangular
- Q is $t \times t$ orthogonal ($Q^T Q = I$)



$$r_B=4$$

Facendo la fattorizzazione della matrice b di prima con $r_b = 4$ otteniamo queste due matrici.

$$Q = \begin{array}{ccccc|cc} -0.577 & 0 & -0.408 & 0 & 0.707 & 0 \\ -0.577 & 0 & 0.816 & 0 & 0 & 0 \\ -0.577 & 0 & -0.408 & 0 & -0.707 & 0 \\ 0 & 0 & 0 & -0.707 & 0 & -0.707 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.707 & -0.0000 & 0.707 \end{array}$$

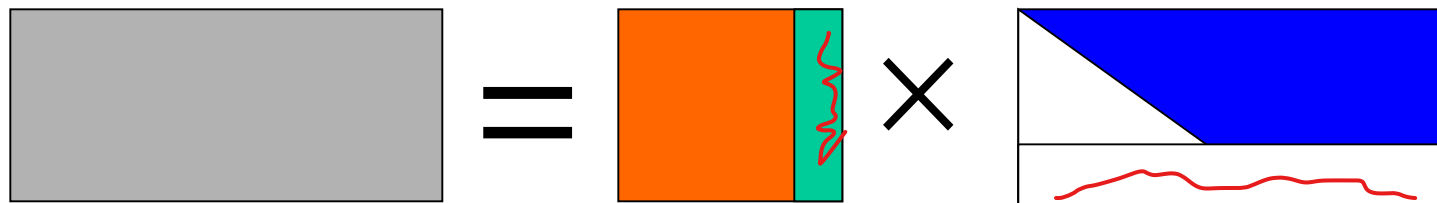
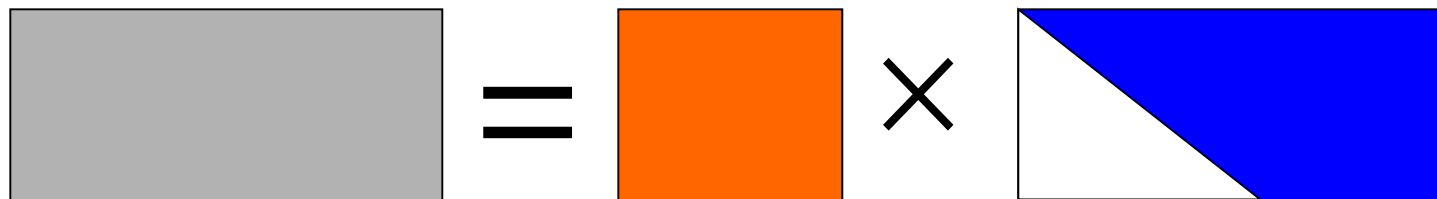
$$R = \begin{array}{ccccc} -1.0000 & 0 & -0.5774 & -0.7071 & -0.4082 \\ 0 & -1 & 0 & -0.4082 & -0.7071 \\ 0 & 0 & 0.8165 & -0.0000 & 0.5774 \\ 0 & 0 & 0 & -0.5774 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}$$

queste sotto alla linea nero annullano le componenti evidenziate dalla barra verticale nera.. gli elementi a dx della barra.

Quella in rosso è una base per il range.

Il significato di quella roba in rosso è che poichè quelli bianchi sono nulli allora anche la parte verde diventerà nulla quindi non la consideriamo.

$$B = QR$$



$$B = \begin{pmatrix} Q_B & Q_B^\perp \end{pmatrix} \begin{pmatrix} R_B \\ 0 \end{pmatrix}$$

questa è la base del complemento ortogonale

$$r_B=4$$

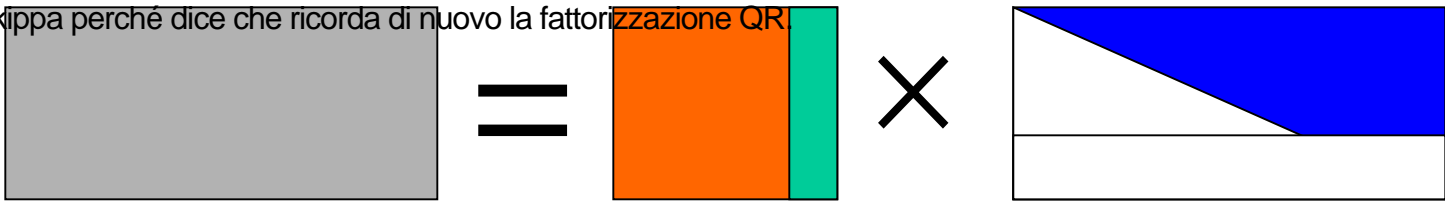
Quindi queste sono le Qb e Rb cioè le matrici "ridotte". Se cambio base e avevo il terzo documento e terza colonna di B ora avendo cambiato base per il range quali sono le componenti della terza colonna di B espressa in termini di base diversa, espressa dalle colonne di Qb, o se vogliamo in altri termini chi è in di Qb e Rb la j-esima colonna di b? $B = \begin{pmatrix} 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0.5774 & 0 & 1 & 0.4082 & 0.7071 \\ 0.5774 & 0 & 0 & 0.4082 & 0 \\ 0 & 1 & 0 & 0.4082 & 0.7071 \\ 0 & 0 & 0 & 0.4082 & 0 \end{pmatrix}$ bj sarà = Qb * j-esima colonna di R, quindi rj sono le componenti di bj sulla Qb. Quindi sono in un altro spazio dove lo stesso vettore ha però un'altra rappresentazione.

$$Q_B = \begin{pmatrix} -0.577 & 0 & -0.408 & 0 \\ -0.577 & 0 & 0.816 & 0 \\ -0.577 & 0 & -0.408 & 0 \\ 0 & 0 & 0 & -0.707 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -0.707 \end{pmatrix}$$

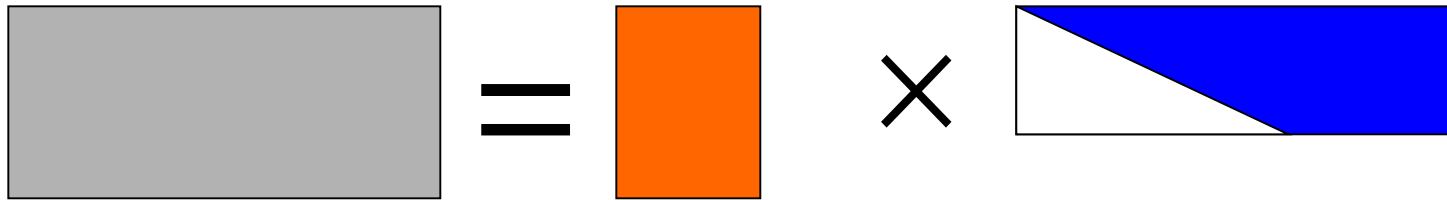
$$R_B = \begin{pmatrix} -1.0000 & 0 & -0.5774 & -0.7071 & -0.4082 \\ 0 & -1 & 0 & -0.4082 & -0.7071 \\ 0 & 0 & 0.8165 & -0.0000 & 0.5774 \\ 0 & 0 & 0 & -0.5774 & 0 \end{pmatrix}$$

document 1 document 2 document 3 document 4 document 5

Questa la skipa perché dice che ricorda di nuovo la fattorizzazione QR.



$$B = \begin{pmatrix} Q_B & Q_B^\perp \end{pmatrix} \begin{pmatrix} R_B \\ 0 \end{pmatrix}$$

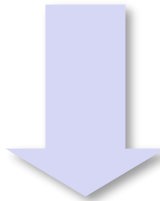


$$\begin{aligned} B &= Q_B R_B + Q_B^\perp 0 \\ &= Q_B R_B \end{aligned}$$

Qui quindi ripete che Q_B sono una base dello spazio delle colonne di B , quindi il contenuto semantico del db è completamente descritto da una base dello spazio delle colonne della matrice termini-documenti che lo rappresenta. Le colonne di R_B sono le coordinate dei documenti nella nuova base Q_B

$$B = Q_B R_B$$

the columns of Q_B are a **basis**
of the **column space** of B



the **semantic content** of the database is fully
described by the **basis** Q_B of the **column space**
of the term-documents matrix

the columns of R_B are the **coordinates** of the
documents in the basis Q_B

Quindi nel processo di query matching dobbiamo usare Q_B e R_B invece di B e quindi il coseno di della j -esima colonna è dato da questa formula.

in the query matching process use

Q_B and R_B instead of B

Semplicemente sostituiamo con Q_B che è la base e R_j che rappresenta il documento nella nuova base.

questo prod è lo si può scrivere anche come r_j trasposto * q_B trasposto * q , ma Q_B traspo per q è la lunghezza della proiezione ortogonale di q sulla prima colonna di Q_B , la seconda componente di $Q_B^T q$ è la proiezione ort di q sulla seconda col di Q_B etc.

$\cos \theta_j$

$= \frac{b_j^T q}{\|b_j\|_2 \cdot \|q\|_2} = \frac{(Q_B r_j)^T q}{\|Q_B r_j\|_2 \cdot \|q\|_2} =$

$= \frac{r_j^T (Q_B^T q)}{\|r_j\|_2 \cdot \|q\|_2}$

quindi questi prod scalari che stanno qui dentro (nelle parentesi) è il vettore q rappresentato sulla base delle colonne di Q_B

Il succo del discorso è che nell'analisi dei dati spesso si cambiamo le basi per poter evidenziare degli aspetti in maniera più semplice rispetto a quella standard con cui essi vengono, per cui in questa lezione abbiamo fatto questo.
N.b i dati non vengono spostati, abbiamo cambiato la base e non lo spazio che è al momento il rango di B , vedremo che si può anche diminuire.

in the **query matching** process use
 Q_B and R_B instead of B

query: **baked bread**

$$q^{(1)} = (1, 0, 1, 0, 0, 0)^T$$

$$\cos \theta_1 = 0.8165$$

$$\cos \theta_2 = 0$$

$$\cos \theta_3 = 0$$

$$\cos \theta_4 = 0.5774$$

$$\cos \theta_5 = 0$$

same result

Ora quello che voglio fare è approssimare B con una matrice C di rango inferiore quindi appunto diminuire lo spazio in cui rappresentiamo i nostri vettori. L'idea che C sia di rango minore ma non deve essere troppo diversa da B quindi deve rappresentare in modo soddisfacente il contenuto semantico del db.

second step:

approximate matrix B by a matrix C of
minor rank


matrix C must represent in a suitable way the semantic content of the database

assumption (reasonable):
choosing **terms** (key-words) for the database (database indexing) introduces uncertainty in the matrix B

Come procedo? Lavoro su R, andando ad azzerare la componente che mi permette di annullare un'ulteriore porzione della matrice, in particolare le colonne di Q. Avendo tre righe a 0 possiamo eliminare 3 colonne di , quindi in questo modo possiamo tagliare di più le matrici.

find a matrix C of **rank less** than B which is
a reasonable approximation of B

approximate R

$$R = \begin{pmatrix} -1.0000 & 0 & -0.5774 & -0.7071 & -0.4082 \\ 0 & -1 & 0 & -0.4082 & -0.7071 \\ 0 & 0 & 0.8165 & -0.0000 & 0.5774 \\ \hline 0 & 0 & 0 & -0.5774 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$


find a matrix C of **rank less** then B which is
a reasonable approximation of B

approximate R

$$\tilde{R} \begin{pmatrix} -1.0000 & 0 & -0.5774 & -0.7071 & -0.4082 \\ 0 & -1 & 0 & -0.4082 & -0.7071 \\ 0 & 0 & 0.8165 & -0.0000 & 0.5774 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

find a matrix C of **rank less** than B which is
a **reasonable approximation** of B

In conseguenza a ciò devo eliminare le ultime tre colonne di Q .

delete the last three columns
of Q

$$\tilde{Q} \begin{pmatrix} -0.577 & 0 & -0.408 \\ -0.577 & 0 & 0.816 \\ -0.577 & 0 & -0.408 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Ora determino C di rango minore a B facendo $Q\tilde{t}ilde * R\tilde{t}ilde$ e devo capire quanto C è lontana da B facendo quindi la norma.
C ha rango 3, B 4.

find a matrix C of **rank less** then B which is
a reasonable approximation of B

$$C = \tilde{Q}\tilde{R}$$

$$r_C=3$$

0.5774	0	0.0000	0.4082	0.0000
0.5774	0	1.0000	0.4082	0.7071
0.5774	0	-0.0000	0.4082	-0.0000
0	0	0	0	0
0	1.0000	0	0.4082	0.7071
0	0	0	0	0

Questa differenza ci dice quanto sono distanti tra loro, si parla di errore di apporssimazione.

$$B - \tilde{Q}\tilde{R}$$

approximation error matrix

0.0000	0	-0.0000	-0.0000	-0.0000
0	0	0	-0.0000	0
0	0	0.0000	-0.0000	0.0000
0	0	0	0.4082	0
0	0	0	0	0.0000
0	0	0	0.4082	0

Qui capiamo l'effetto dell'aver trovato una matrice C di rango inferiore, ovviamente ora i coseni li calcolo rispetto alle colonne di R tilde.

documents in the new reference system

$B \approx \tilde{Q}\tilde{R}$

$\tilde{Q} \begin{pmatrix} -0.577 & 0 & -0.408 \\ -0.577 & 0 & 0.816 \\ -0.577 & 0 & -0.408 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

basis

$\tilde{R} \begin{pmatrix} -1.0000 & 0 & -0.5774 & -0.7071 & -0.4082 \\ 0 & -1 & 0 & -0.4082 & -0.7071 \\ 0 & 0 & 0.8165 & -0.0000 & 0.5774 \end{pmatrix}$

components

document 1

document 2

document 3

document 4

document 5

results of the query in the new reference system

Otteniamo quindi un risultato abbastanza simile.

$$q^{(1)} = (1, 0, 1, 0, 0, 0)^T$$

baked bread

$\cos \theta_1 = 0.8165$	similarity query and first document
$\cos \theta_2 = 0$	similarity query and second document
$\cos \theta_3 = 0$	similarity query and third document
$\cos \theta_4 = 0.7071$	similarity query and fourth document
$\cos \theta_5 = 0$	similarity query and fifth document

results of the query in the new reference system

stessa cosa ma con una query diversa.

query: **baked**

$\cos \theta_1 = 0.5774$	similarity query and first document
$\cos \theta_2 = 0$	similarity query and second document
$\cos \theta_3 = 0$	similarity query and third document
$\cos \theta_4 = 0.5000$	similarity query and fourth document
$\cos \theta_5 = 0$	similarity query and fifth document

LSA = Latent Semantic Analysis

<http://lsa.colorado.edu>
<http://knowledgesearch.org>

- ✓ documents comparison (clustering, classification)
- ✓ search engines
- ✓ search for relationship between terms (synonymy, polysemy)
- ✓ applications to natural languages
- ✓

Invece di fare la fattorizzazione QR possiamo usare la fattorizzazione SVD della matrice termini documenti che consente di agire anche sullo spazio delle righe della matrice.

instead of **QR factorization**
you can use
SVD factorization
of the term-documents matrix
(which also allows you to act on the
space of the rows of the matrix)