Applicazioni della SVD

Gianna M. Del Corso

Dipartimento di Informatica, Università di Pisa, Italy

26 Aprile 2012

Le Applicazioni

- Algoritmo Eigenfaces per il riconoscimento automatico di volti umani
- Text Mining: Modello dello spazio vettoriale, LSI

Sono alcuni dei temi proposti nel corso di Laboratorio di Matematica Computazionale

Le Applicazioni

- Algoritmo Eigenfaces per il riconoscimento automatico di volti umani
- Text Mining: Modello dello spazio vettoriale, LSI

Sono alcuni dei temi proposti nel corso di Laboratorio di Matematica Computazionale

Eigenfaces

Problema: Riconoscimento automatico di volti umani

Uso: trovare un individuo in uno schedario, taggare foto come fa Picasa, Iphoto

Presenterò un semplice algoritmo chiamato Eigenfaces dovuto a Turk e Pentland che è del 1991. Successivamente le tecniche proposte sono state migliorate usando strumenti matematici più raffinati.

Eigenfaces

Problema: Riconoscimento automatico di volti umani

Uso: trovare un individuo in uno schedario, taggare foto come fa Picasa, Iphoto

Presenterò un semplice algoritmo chiamato Eigenfaces dovuto a Turk e Pentland che è del 1991. Successivamente le tecniche proposte sono state migliorate usando strumenti matematici più raffinati.

Eigenfaces

Problema: Riconoscimento automatico di volti umani

Uso: trovare un individuo in uno schedario, taggare foto come fa Picasa, Iphoto

Presenterò un semplice algoritmo chiamato Eigenfaces dovuto a Turk e Pentland che è del 1991. Successivamente le tecniche proposte sono state migliorate usando strumenti matematici più raffinati.

Eigenfaces

Obiettivo originario: proporre un metodo veloce, semplice e accurato

Gli algoritmi precedenti si concentravano su l'individuazione di caratteristiche quali la forma degli occhi, della bocca, del naso, confrontando le varie facce su queste caratteristiche fisionomiche.

Approccio banale, non funziona! Le teste possono essere piegate, ci possono essere luci, occhiali etc che rendono grande la norma della differenza...

Eigenfaces

Obiettivo originario: proporre un metodo veloce, semplice e accurato

Gli algoritmi precedenti si concentravano su l'individuazione di caratteristiche quali la forma degli occhi, della bocca, del naso, confrontando le varie facce su queste caratteristiche fisionomiche.

Approccio banale, non funziona! Le teste possono essere piegate, ci possono essere luci, occhiali etc che rendono grande la norma della differenza...

Eigenfaces

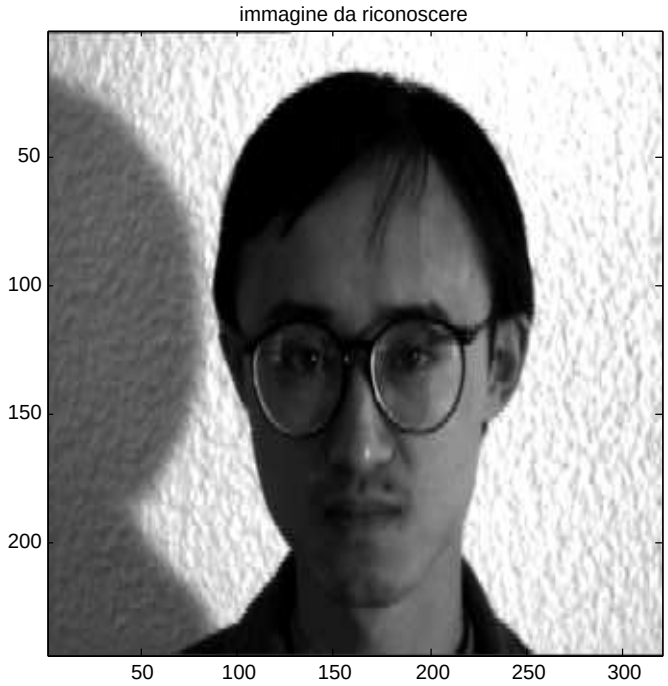
Obiettivo originario: proporre un metodo veloce, semplice e accurato

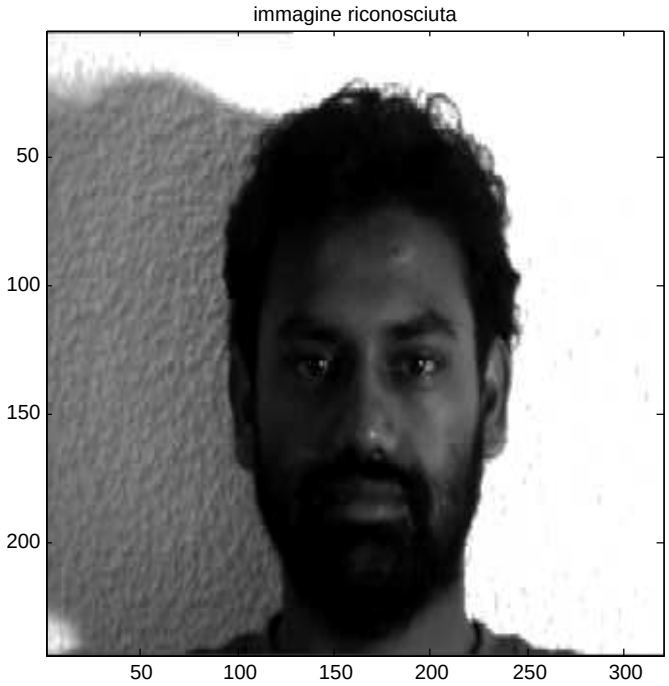
Gli algoritmi precedenti si concentravano su l'individuazione di caratteristiche quali la forma degli occhi, della bocca, del naso, confrontando le varie facce su queste caratteristiche fisionomiche.

Approccio banale, non funziona! Le teste possono essere piegate, ci possono essere luci, occhiali etc che rendono grande la norma della differenza...

Distanza tra immagini

Distanza tra immagini





Eigenfaces

Idea: tecniche di codifica e decodifica possono rivelare il contenuto informativo di immagini di volti enfatizzando le features locali e globali di un volto.

Queste features possono essere o non essere correlate con i nostri concetti fisionomici, tipo naso, labbra, occhi.

Scopo:

- Estrarre le informazioni rilevanti di un'immagine di un volto, codificarla in modo efficiente.
- Rappresentare le facce efficientemente

La SVD (imparentata con la PCA) sembra proprio la codifica che serve!

Eigenfaces

Idea: tecniche di codifica e decodifica possono rivelare il contenuto informativo di immagini di volti enfatizzando le features locali e globali di un volto.

Queste features possono essere o non essere correlate con i nostri concetti fisionomici, tipo naso, labbra, occhi.

Scopo:

- Estrarre le informazioni rilevanti di un'immagine di un volto, codificarla in modo efficiente.
- Rappresentare le facce efficientemente

La SVD (imparentata con la PCA) sembra proprio la codifica che serve!

Eigenfaces

Idea: tecniche di codifica e decodifica possono rivelare il contenuto informativo di immagini di volti enfatizzando le features locali e globali di un volto.

Queste features possono essere o non essere correlate con i nostri concetti fisionomici, tipo naso, labbra, occhi.

Scopo:

- Estrarre le informazioni rilevanti di un'immagine di un volto, codificarla in modo efficiente.
- Rappresentare le facce efficientemente

La SVD (imparentata con la PCA) sembra proprio la codifica che serve!

Eigenfaces

Idea: tecniche di codifica e decodifica possono rivelare il contenuto informativo di immagini di volti enfatizzando le features locali e globali di un volto.

Queste features possono essere o non essere correlate con i nostri concetti fisionomici, tipo naso, labbra, occhi.

Scopo:

- Estrarre le informazioni rilevanti di un'immagine di un volto, codificarla in modo efficiente.
- Rappresentare le facce efficientemente

La SVD (imparentata con la PCA) sembra proprio la codifica che serve!

Eigenfaces

... In termini matematici

- trovare i vettori singolari della distribuzione delle facce $\Longleftrightarrow$ autovettori della matrice di covarianza delle facce
- questi autovettori rappresentano un insieme di features che caratterizzano le differenze tra le varie facce
- ogni immagine contribuisce ad ogni autovetture, quindi un autovetture rappresenta una sorta di immagine "ghost" che chiamiamo eigenface

L'algoritmo

Inizializzazione: acquisizione del training set e calcolo delle eigenfaces Spazio delle facce

$$I_1, I_2, \cdots, I_M \quad \text{training set}$$

Il training set

L'algoritmo

$$\Psi = \frac{1}{M}\sum_{i=1}^{M} I_i \quad \text{faccia "media"}$$

$$\Phi_i = I_i - \Psi \quad \text{differenza con la faccia media}$$

Il training set

Togliendo la faccia media, il training set diviene

L'algoritmo

$$\Psi = \frac{1}{M}\sum_{i=1}^{M} I_i \quad \text{faccia "media"}$$

$$\Phi_i = I_i - \Psi \quad \text{differenza con la faccia media}$$

$$A = [\Phi_1(:), \Phi_2(:), \cdots, \Phi_M(:)]$$

L'algoritmo

Si cercano gli M vettori ortogonali $\mathbf{u}_k$ che descrivano al meglio i dati, massimizzino le quantità

$$\lambda_k = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{u}_k^T \Phi_i)^2$$

$$\mathbf{u}_j^T \mathbf{u}_k = \delta_{jk}.$$

$$\frac{\mathbf{u}_k^T \Phi_i}{\|\mathbf{u}_k\| \, \|\Phi_i\|} = \cos(\theta_k)$$

L'algoritmo

Si cercano gli M vettori ortogonali $\mathbf{u}_k$ che descrivano al meglio i dati, massimizzino le quantità

$$\lambda_k = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{u}_k^T \Phi_i)^2$$

$$\mathbf{u}_j^T \mathbf{u}_k = \delta_{jk}.$$

$$\frac{\mathbf{u}_k^T \Phi_i}{\|\mathbf{u}_k\| \, \|\Phi_i\|} = \cos(\theta_k)$$

L'algoritmo

$\mathbf{u}_k$ e λ_k sono gli autovettori e gli autovalori della matrice

$$C = \frac{1}{M} \sum_{i=1}^{M} \Phi_i \Phi_i^T = AA^T$$

C è la matrice di covarianza!

La matrice C ha dimensioni $N^2 \times N^2$, se le immagini sono $N \times N$.... è una matrice molto grande...

L'algoritmo

$\mathbf{u}_k$ e λ_k sono gli autovettori e gli autovalori della matrice

$$C = \frac{1}{M} \sum_{i=1}^{M} \Phi_i \Phi_i^T = AA^T$$

C è la matrice di covarianza!

La matrice C ha dimensioni $N^2 \times N^2$, se le immagini sono $N \times N$.... è una matrice molto grande...

L'algoritmo

Si può considerare la matrice $L = A^T A$ che è $M \times M$ e calcolare autovalori ed autovettori

$$(A^T A)\mathbf{v}_i = \mu_i \mathbf{v}_i,$$

$$A\,(A^T A)\,\mathbf{v}_i = \mu_i A\,\mathbf{v}_i,$$

$$(AA^T)A\mathbf{v}_i = \mu_i A\,\mathbf{v}_i,$$

sostituendo $A\mathbf{v}_i = \mathbf{u}_i$, abbiamo gli autovettori (non scalati) di C.
$\mathbf{u}_k = \sum_{i=1}^{M} v_{ki}\Phi_i$.

L'algoritmo

Quindi si calcolano M autovettori $\mathbf{v}_i$ di $L = A^T A$ e poi si calcolano le facce nello spazio degli autovettori di L.

In realtà ... trovo direttamente $\mathbf{u}_i$ e $\mathbf{v}_i$ con la SVD di A!!

$$A = U\Sigma V^H,$$

possiamo considerare la SVD ridotta,

$$U \in \mathbb{C}^{N^2 \times M}, \Sigma \in \mathbb{R}^{M \times M}, V \in \mathbb{C}^{M \times M}.$$

L'algoritmo

Quindi si calcolano M autovettori $\mathbf{v}_i$ di $L = A^T A$ e poi si calcolano le facce nello spazio degli autovettori di L.

In realtà ... trovo direttamente $\mathbf{u}_i$ e $\mathbf{v}_i$ con la SVD di A!!

$$A = U\Sigma V^H,$$

possiamo considerare la SVD ridotta,

$$U \in \mathbb{C}^{N^2 \times M}, \Sigma \in \mathbb{R}^{M \times M}, V \in \mathbb{C}^{M \times M}.$$

L'algoritmo

Quindi si calcolano M autovettori $\mathbf{v}_i$ di $L = A^T A$ e poi si calcolano le facce nello spazio degli autovettori di L.

In realtà ... trovo direttamente $\mathbf{u}_i$ e $\mathbf{v}_i$ con la SVD di A!!

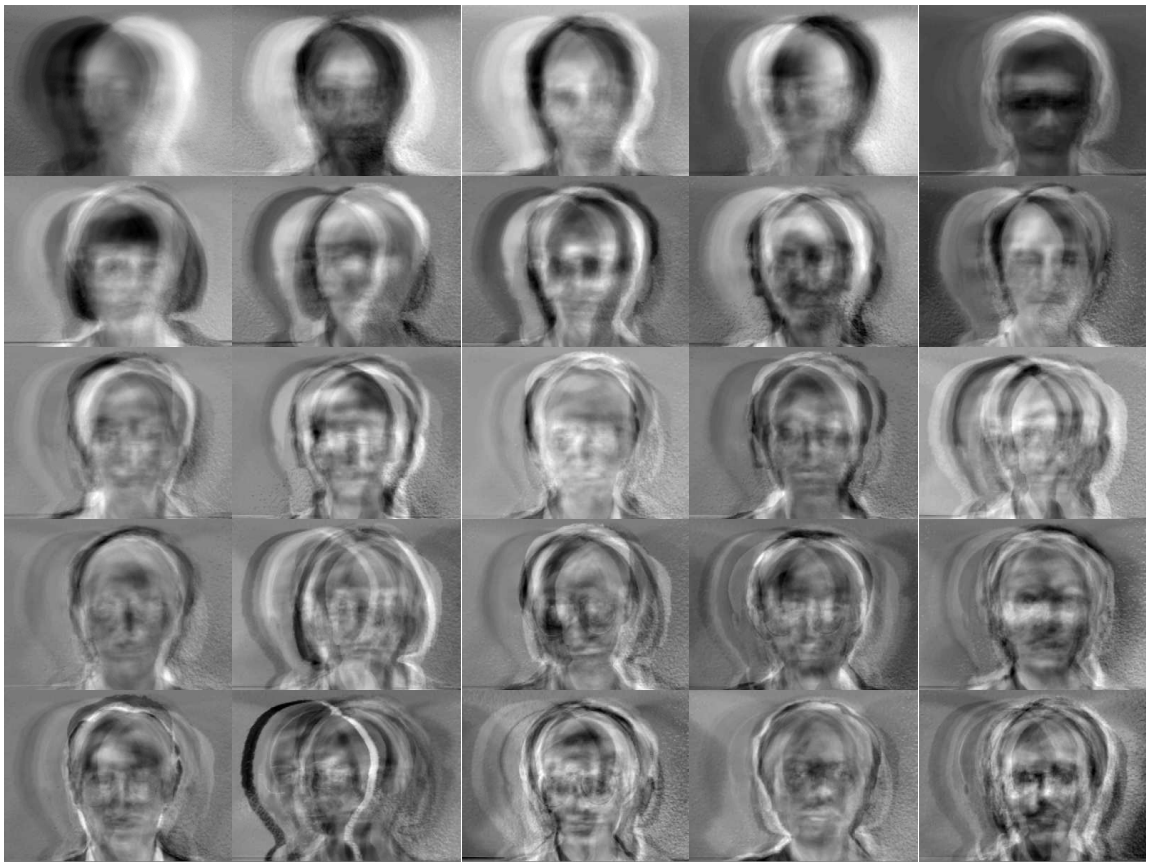
$$A = U\Sigma V^H,$$

possiamo considerare la SVD ridotta,

$U \in \mathbb{C}^{N^2 \times M}, \Sigma \in \mathbb{R}^{M \times M}, V \in \mathbb{C}^{M \times M}$.

L'algoritmo

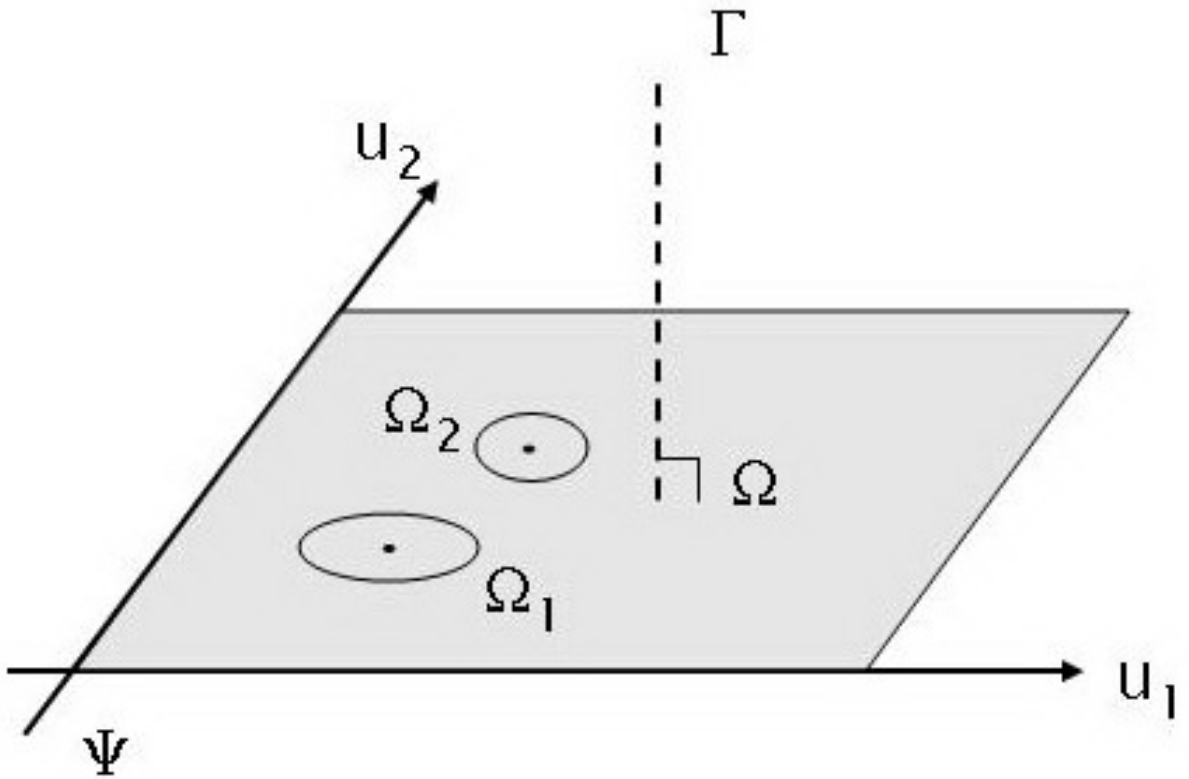
Le colonne di U sono le Eigenfaces



L'algoritmo

Si possono considerare solo $k << M$ eigenfaces corrispondenti ai σ_i più grandi!

span$\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ è lo spazio delle facce



Riconoscimento di facce

$\Omega_i = U(:, 1:k)^T \Phi_i$ è la proiezione della faccia $i-$esima sullo spazio ridotto delle facce.

Data immagine query I_Q che non è nel nostro training set,

- $\Phi = I_Q - \Psi$ sottraggo la faccia media
- $\Omega = U(:, 1:k)^T \Phi$ la proietto nello spazio delle facce

Riconoscimento di facce

$\Omega_i = U(:, 1:k)^T \Phi_i$ è la proiezione della faccia $i-$esima sullo spazio ridotto delle facce.

Data immagine query I_Q che non è nel nostro training set,

- $\Phi = I_Q - \Psi$ sottraggo la faccia media
- $\Omega = U(:, 1:k)^T \Phi$ la proietto nello spazio delle facce

Riconoscimento di facce

Si calcolano $\varepsilon_i = \|\Omega - \Omega_i\|^2$

- Se $\min_i \varepsilon_i < \theta_\varepsilon$ allora I_Q è riconosciuta come una foto di I_j, $j : \min_i \varepsilon_i = \varepsilon_j$
- Se per ogni i, $\varepsilon_i > \theta_\varepsilon$, allora è classificata come sconosciuta

L'algoritmo

Ricapitolando:

- Collezionare un set di M immagini di facce. Deve includere più immagini dello stesso individuo
- Costruire la matrice A disponendo le immagini vettorizzate per colonna
- Calcolare la decomposizione ai valori singolari di A troncata a $k << M$ termini.
- I k vettori singolari sinistri di A rappresentano le k eigenfaces
- Si proietta ogni immagine sullo spazio delle facce, si proietta anche l'immagine query
- Se $\min_i \|\Omega - \Omega_i\|^2$ è sufficientemente piccolo si "riconosce" l'immagine query.

L'implementazione

Training set: yalefaces 15 individual fotografati in 11 pose different:

"centerlight", "glasses", "happy", "leftlight", "noglasses", "normal", "rightlight", "sad", "sleepy", "surprised", "wink"

Si costruisce la matrice A di dimensioni $(320 * 243) \times (11 * 15)$.

Si toglie dal dataset una immagine query.

Immagini diverse

L'algoritmo come funziona se si chiede di riconoscere un immagine non umana?

Text Maning

Per Text Mining si intendono metodi per estrarre informazioni utili da grandi collezioni di documenti testuali.

Preprocessing dei documenti e delle query

- Eliminazione di *stop words*
- *Stemming*
- *Creazione di una lista di termini* $\rightarrow$ *indicizzazione dei documenti*

Il modello dello spazio vettoriale

Si crea una matrice termini-documenti A, in cui ogni documento è rappresentato da un vettore.

$A_{i,j} \neq 0$ se il documento $j-$esimo contiene il termine $i-$esimo

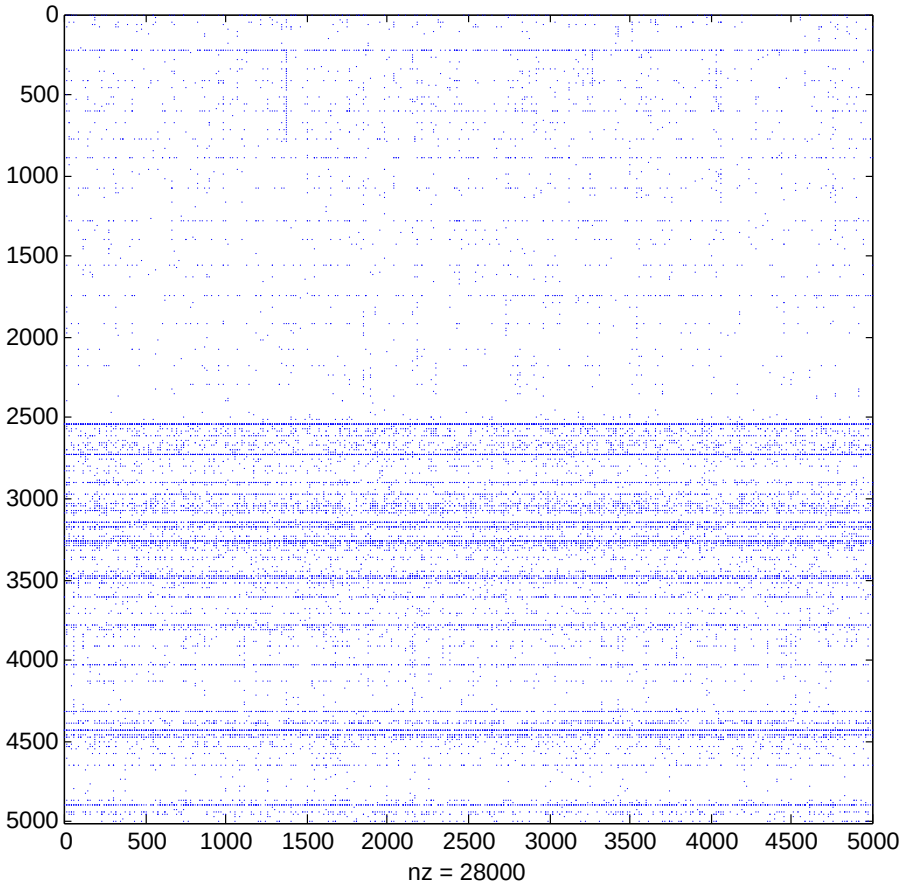
È comune applicare uno schema di "pesatura" dei termini ad esempio

$$A_{i,j} = f_{ij} \log(n/n_i)$$

dove f_{ij} è la frequenza del termine i nel doc j, e n_i è il numero di documenti in cui appare il termine i.
Se un termine è frequente solo in alcuni documenti, allora il termine discrimina bene tra differenti gruppi di documenti.

Il modello dello spazio vettoriale

La matrice è tipicamente molto sparsa



Il modello dello spazio vettoriale

Documenti

D1	Pronto soccorso per *neonati* e *divezzi*
D2	La cameretta di *neonati* e *bimbi* (per la vostra *casa*)
D3	La *sicurezza* dei *bimbi* a *casa*
D4	La *salute* del vostro *bimbo* e la sua *sicurezza*: dal *neonato* al *divezzo*
D5	*Nozioni* base di *sicurezza* per il *neonato*
D6	*Guida* con *nozioni* di base per la rimozione in *sicurezza* della ruggine
D7	*Guida* all'uso *salutare* del *Bimby*

Il modello dello spazio vettoriale

Ecco un esempio:

T1	neonat(o,i,a,e)
T2	bimb(o,a,i,e, y)
T3	guida
T4	salut(e, are)
T5	casa
T6	sicurezza
T7	divezz(o,i)
T8	nozion(e,i)

Il modello dello spazio vettoriale

Abbiamo lo seguente matrice termini documenti

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Il modello dello spazio vettoriale

Normalizzando per colonna- dividendo per $\|A(:,i)\|_2$ abbiamo

$$A = \begin{pmatrix} 0.7071 & 0.5774 & 0 & 0.4472 & 0.5774 & 0 & 0 \\ 0 & 0.5774 & 0.5774 & 0.4472 & 0 & 0 & 0.5774 \\ 0 & 0 & 0 & 0 & 0 & 0.5774 & 0.5774 \\ 0 & 0 & 0 & 0.4472 & 0 & 0 & 0.5774 \\ 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5774 & 0.4472 & 0.5774 & 0.5774 & 0 \\ 0.7071 & 0 & 0 & 0.4472 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5774 & 0.5774 & 0 \end{pmatrix}$$

Query matching

Ogni query può essere rappresentata con un vettore nello spazio vettoriale. In questo caso i vettori delle query sono lunghi 8 perchè abbiamo 8 termini.
Supponiamo la nostra query sia *La sicurezza del neonato in casa*, viene rappresentata come il vettore

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

.
Quali sono i documenti più vicini alla query?

Query matching

Una delle tecniche più semplici è la misura del coseno dell'angolo tra la query e i vettori dei documenti

Vengono fuori i seguenti valori:

$$cos(\theta) = (0.4082, 0.6667, 0.6667, 0.5164, 0.6667, 0.3333, 0)^T$$

Documenti restituiti come rilevanti la query *La sicurezza del neonato in casa* sono il 2, 3 e 5, cioè...

Query matching

D1 Pronto soccorso per *neonati* e *divezzi*
D2 La cameretta di *neonati* e *bimbi* (per la vostra *casa*)
D3 La *sicurezza* dei *bimbi* a *casa*
D4 La *salute* del vostro *bimbo* e la sua *sicurezza*: dal *neonato* al *divezzo*
D5 *Nozioni* base di *sicurezza* per il *neonato*
D6 *Guida* con *nozioni* di base per la rimozione in *sicurezza* della ruggine
D7 *Guida* all'uso *salutare* del *Bimby*

Che sono rilevanti!

Query matching

Se la query fosse stata Guida al bimbo, avremmo avuto

$$\cos(\theta) = (0, 0.4082, 0.4082, 0.3162, 0, 0.4082, 0.8165)^T$$

che restituiva come rilevante il documento 7: *Guida* all'uso *salutare* del *Bimby*!! Ci saremmo invece aspettati i documenti 1, 3, 4, 5....

Possibili miglioramenti

- criteri di pesatura dei termini
- approssimazioni di rango basse della matrice termini-documenti

Query matching

Se la query fosse stata Guida al bimbo, avremmo avuto

$$\cos(\theta) = (0, 0.4082, 0.4082, 0.3162, 0, 0.4082, 0.8165)^T$$

che restituiva come rilevante il documento 7: *Guida* all'uso *salutare* del *Bimby*!! Ci saremmo invece aspettati i documenti 1, 3, 4, 5....

Possibili miglioramenti

- criteri di pesatura dei termini
- approssimazioni di rango basse della matrice termini-documenti

Approssimazioni di rango basso:LSI

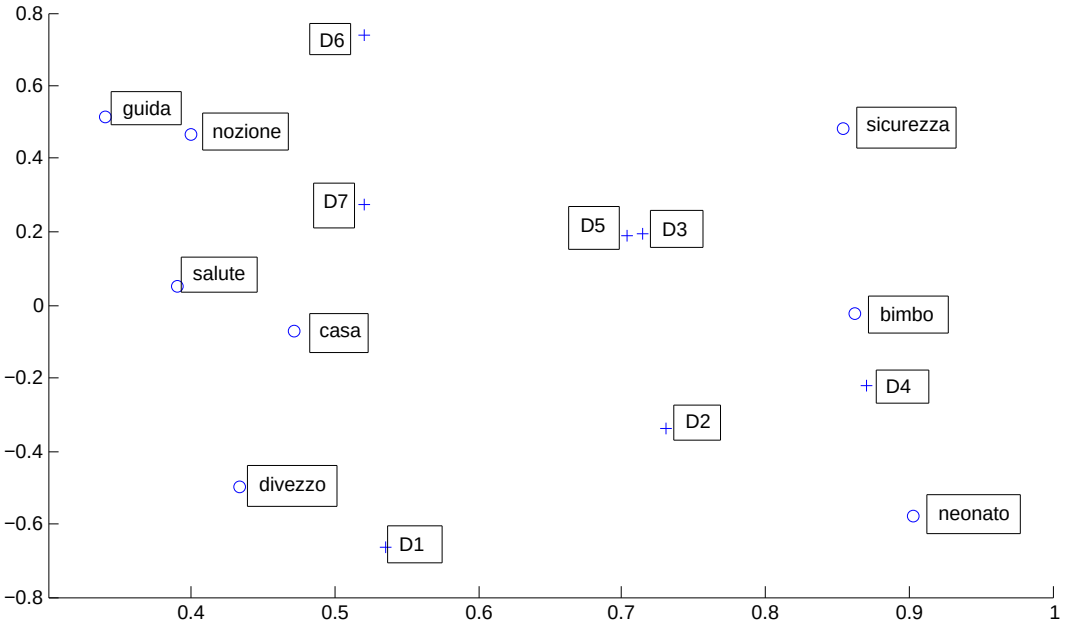
Parte del risultati negativi sembrano dovuti

- Scelta delle parole utilizzate
- differenza tra l'uso dei termini di diversi autori
- possibili errori

Sembra esserci una struttura sottostante latente nei dati che è distrutta dalla varietà delle parole usate nel testo.

Questa struttura può essere scoperta proiettando i dati in uno spazio di dimensione minore. Questo è il Latent semantic indexing

Approssimazioni di rango basso



Approssimazioni di rango basso

Usando la SVD possiamo proiettare sia i termini che i documenti in spazi di dimensione ridotta.
Sia A la matrice termini-documenti. $A = U\Sigma V^T$, approssimandola con una matrice di rango k,

$$A \approx U_k \Sigma_k V_k^T,$$

$U_k \approx$ lo spazio dei documenti, $V_k \approx$ lo spazio dei termini.

Query matching

In questo caso vogliamo confrontare una query $\mathbf{q}$ con una matrice di rango ridotta A_k

$$\cos(\theta_j) = \frac{(A_k \mathbf{e}_j)^T \mathbf{q}}{\|A_k \mathbf{e}_j\| \, \|\mathbf{q}\|} = \frac{(U_k \Sigma_k V_k^T \mathbf{e}_j)^T \mathbf{q}}{\|U_k \Sigma_k V_k^T \mathbf{e}_j\| \, \|\mathbf{q}\|}$$
$$= \frac{\mathbf{e}_j^T V_k \Sigma_k (U_k^T \mathbf{q})}{\|\Sigma_k V_k^T \mathbf{e}_j\| \, \|\mathbf{q}\|}$$

Query matching

Detto $\mathbf{s}_j = \Sigma_k V_k^T \mathbf{e}_j$, cioeè le colonne di A_k nella base di U_k, abbiamo

$$\cos(\theta_j) = \frac{\mathbf{s}_j^T (U_k^T \mathbf{q})}{\|\mathbf{s}_j\| \, \|\mathbf{q}\|}.$$

Non si deve esplicitamente calcolare A_k!
I vettori $\mathbf{s}_k$ possono essere calcolati una volta e usati per più query.

Query matching

Con la query La salute del bambino, utilizzando tutto lo spazio abbiamo

$$\cos(\theta) = (0, 0.4082, 0.4082, 0.6325, 0, 0, 0.8165),$$

proiettando sullo spazio di dimensione 2 otteniamo

$$\cos(\theta) = (0.4236, 0.6314, 0.6901, 0.6801, 0.6906, 0.4283, 0.6385)$$

D3 La *sicurezza* dei *bimbi* a *casa*
D5 *Nozioni* base di *sicurezza* per il *neonato*