Modelli Geometrici per la Semantica Lessicale, l'Apprendimento Automatico e l'Information Retrieval

R. Basili

Corso di *Web Mining* e *Retrieval*
a.a. 2015-16

May 11, 2016

Change of Basis

Change of Basis

Given two alternative basis $B = \{\underline{b}_1, ..., \underline{b}_n\}$ and $B' = \{\underline{b}'_1, ..., \underline{b}'_n\}$, such that the square matrix $\mathbf{C} = (c_i k)$ describe the change of the basis, i.e.

$$\underline{b}'_k = c_{1k}\underline{b}_1 + c_{2k}\underline{b}_2 + ...c_{nk}\underline{b}_n \qquad \forall k = 1, ..., n$$

Matrix and Change of Basis

Matrix and Change of Basis

The effect of the matrix $\mathbf{C}$ on a generic vector $\underline{x}$ allows to compute the change of basis according only to the involved basis B and B'. For every $\underline{x} = \sum_{k=1}^{n} x_k \underline{b}_k$ such that in the new basis B', $\underline{x}$ can be expressed by $\underline{x} = \sum_{k=1}^{n} x'_k \underline{b}'_k$, then it follows that:

$$\underline{x} = \sum_{k=1}^{n} x'_k \underline{b}'_k = \sum_{k} x'_k \left(\sum_{i} c_{ik} \underline{b}_i \right) = \sum_{i,k=1}^{n} x'_k c_{ik} \underline{b}_i$$

from which it follows that:

$$x_i = \sum_{k=1}^{n} x'_k c_{ik} \qquad \forall i = 1, \ldots, n$$

Matrix and Change of Basis

Matrix and Change of Basis

The effect of the matrix $\mathbf{C}$ on a generic vector $\underline{x}$ allows to compute the change of basis according only to the involved basis B and B'. For every $\underline{x} = \sum_{k=1}^{n} x_k \underline{b}_k$ such that in the new basis B', $\underline{x}$ can be expressed by $\underline{x} = \sum_{k=1}^{n} x'_k \underline{b}'_k$, then it follows that:

$$\underline{x} = \sum_{k=1}^{n} x'_k \underline{b}'_k = \sum_{k} x'_k \left(\sum_{i} c_{ik} \underline{b}_i \right) = \sum_{i,k=1}^{n} x'_k c_{ik} \underline{b}_i$$

from which it follows that:

$$x_i = \sum_{k=1}^{n} x'_k c_{ik} \qquad \forall i = 1, \ldots, n$$

The above condition suggests that $\mathbf{C}$ is sufficient to describe any change of basis through the matrix vector mutliplication operations:

$$\underline{x} = \mathbf{C}\underline{x}'$$

Matrix and Change of Basis

Matrix and Change of Basis

The effect of the matrix $\mathbf{C}$ on a matrix $\mathbf{A}$ can be seen by studying the case where $\underline{x}, \underline{y}$ are the expression of two vectors in a base B while their counterpart on B' are $\underline{x}', \underline{y}'$, respectively. Now if $\mathbf{A}$ and $\mathbf{B}$ are such that $\underline{y} = \mathbf{A}\underline{x}$ and $\underline{y}' = \mathbf{B}\underline{x}'$, then it follows that:

$$\underline{y} = \mathbf{C}\underline{y}' = \mathbf{A}\underline{x} = \mathbf{A}(\mathbf{C}\underline{x}') = \mathbf{AC}\underline{x}'$$

(this means that)

$$\underline{y}' = \mathbf{C}^{-1}\mathbf{AC}\underline{x}'$$

from which it follows that:

$$\mathbf{B} = \mathbf{C}^{-1}\mathbf{AC}$$

The transformation of basis $\mathbf{C}$ is a *similarity transformation* and matrices $\mathbf{A}$ and $\mathbf{C}$ are said *similar*.

Matrix and Change of Basis

Matrix and Change of Basis

The effect of the matrix $\mathbf{C}$ on a matrix $\mathbf{A}$ can be seen by studying the case where $\underline{x}, \underline{y}$ are the expression of two vectors in a base B while their counterpart on B' are $\underline{x}', \underline{y}'$, respectively. Now if $\mathbf{A}$ and $\mathbf{B}$ are such that $\underline{y} = \mathbf{A}\underline{x}$ and $\underline{y}' = \mathbf{B}\underline{x}'$, then it follows that:

$$\underline{y} = \mathbf{C}\underline{y}' = \mathbf{A}\underline{x} = \mathbf{A}(\mathbf{C}\underline{x}') = \mathbf{AC}\underline{x}'$$

(this means that)

$$\underline{y}' = \mathbf{C}^{-1}\mathbf{AC}\underline{x}'$$

from which it follows that:

$$\mathbf{B} = \mathbf{C}^{-1}\mathbf{AC}$$

The transformation of basis $\mathbf{C}$ is a *similarity transformation* and matrices $\mathbf{A}$ and $\mathbf{C}$ are said *similar*.

From EigenVectors and Matrix Decomposition to Topic Models

Matrix Eigendecomposition

Let us create a matrix $\mathbf{S}$ with columns the n eigenvectors of a matrix $\mathbf{A}$. We have that

$$\mathbf{AS} = \mathbf{A}[\underline{x}_1, ..., \underline{x}_n] =$$

$$= \mathbf{A}\underline{x}_1 + ... + \mathbf{A}\underline{x}_n =$$

$$= \lambda_1 \underline{x}_1 + ... + \lambda_n \underline{x}_n = [\underline{x}_1, ..., \underline{x}_n]\Lambda$$

where Λ is the diagonal matrix with the eigenvalues of $\mathbf{A}$ along its diagonal:

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \dots & \lambda_n \end{pmatrix}$$

EigenVectors of symmetric matrices

Now suppose that the above n eigenvectors are linearly independent. This is true when the matrix has n distinct eigenvalues. Then matrix $\mathbf{S}$ is invertible and it holds: $\mathbf{AS} = \mathbf{S}\Lambda$ so that

$$\mathbf{A} = \mathbf{S}\Lambda\mathbf{S}^{-1}$$

Towards SVD

EigenDecomposition of Symmetric matrices

Now let $\mathbf{A}$ be an $m \times n$ matrix with entries being real numbers and $m > n$.
Let us consider the $n \times n$ square matrix $\mathbf{B} = \mathbf{A}^T\mathbf{A}$.
It is easy to verify that B is symmetric, as $\mathbf{B}^T = (\mathbf{A}^T\mathbf{A})^T =$
$\mathbf{A}^T(\mathbf{A}^T)^T = \mathbf{A}^T\mathbf{A} = \mathbf{B}$.
It has been shown that the eigenvalues of such matrices $(\mathbf{A}^T\mathbf{A})$ are real non-negative numbers. Since they are non-negative we can write them in decreasing order as squares of non-negative real numbers:

$$\sigma_1^2 \geq \sigma_2^2 \geq \ldots \geq \sigma_n^2.$$

For some index r (possibly n) the first r numbers $\sigma_1, \ldots, \sigma_r$ are positive whereas the rest are zero. For the above eigenvalues, we know that the corresponding eigenvectors $\underline{x}_1, \ldots, \underline{x}_r$ are perpendicular. Furthermore, we normalize them to have length 1. Let

$$\mathbf{S}_1 = [\underline{x}_1, \ldots, \underline{x}_r]$$

Towards SVD (2)

From the set of r orthonormal eigenvectors we can create the following vectors

$$\underline{y}_1 = \frac{1}{\sigma_1}\mathbf{A}\underline{x}_1, \ldots, \underline{y}_r = \frac{1}{\sigma_r}\mathbf{A}\underline{x}_r$$

These are perpendicular m-dimensional vectors of length 1 (orthonormal vectors) as:

$\underline{y}_i^T \underline{y}_j = \left(\frac{1}{\sigma_i}\mathbf{A}\underline{x}_i\right)^T \frac{1}{\sigma_j}\mathbf{A}\underline{x}_j = \frac{1}{\sigma_i\sigma_j}\underline{x}_i^T\mathbf{A}^T\mathbf{A}\underline{x}_j = \frac{1}{\sigma_i\sigma_j}\underline{x}_i^T\mathbf{B}\underline{x}_j = \frac{1}{\sigma_i\sigma_j}\underline{x}_i^T\sigma_j^2\underline{x}_j = \frac{\sigma_j}{\sigma_i}\underline{x}_i^T\underline{x}_j$

Now this is 0 when $i \neq j$ and 1 when $i = j$ (as $\underline{x}_i^T\underline{x}_j = 0$ when $i \neq j$ and $\underline{x}_i^T\underline{x}_i$=1 $\forall i$)

Towards SVD (3)

Moreover, given

$$\mathbf{S}_2 = [\underline{y}_1, ..., \underline{y}_r]$$

we have $\underline{y}_j^T \mathbf{A} \underline{x}_i = \underline{y}_j^T (\sigma_i \underline{x}_i) = \sigma_i \underline{y}_j^T \underline{x}_i$ which is 0 if $i \neq j$, and σ_i if $i = j$.
It follows thus that:

$$\mathbf{S}_2^T \mathbf{A} \mathbf{S}_1 = \Sigma$$

where Σ is the diagonal $r \times r$ matrix with $\sigma_1, ..., \sigma_r$ along the diagonal.

The SVD

Observe that $\mathbf{S}_2^T$ is $r \times m$, $\mathbf{A}$ is $m \times n$, and $\mathbf{S}_1$ is $n \times r$, and thus the above matrix multiplication is well defined.
Since $\mathbf{S}_2$ and $\mathbf{S}_1$ have orthonormal columns, $\mathbf{S}_2\mathbf{S}_2^T = \mathbf{I}_{m\times m}$ and $\mathbf{S}_1\mathbf{S}_1^T = \mathbf{I}_{n\times n}$ (where $\mathbf{I}_{m\times m}$ and $\mathbf{I}_{n\times n}$ are the $m \times m$ and $n \times n$ identity matrices.
Thus, by multiplying the equality

$$\mathbf{S}_2^T\mathbf{A}\mathbf{S}_1 = \Sigma$$

by $\mathbf{S}_2$ on the left and $\mathbf{S}_1^T$ on the right, we have

$$\mathbf{A} = \mathbf{S}_2\Sigma\mathbf{S}_1^T$$

Summing-up the SVD definition

Reiterating, matrix Σ is diagonal and the values along the diagonal are $\sigma_1, \ldots, \sigma_r$ which are called *singular values*.
They are the square roots of the eigenvalues of $\mathbf{A}^T\mathbf{A}$ and thus completely determined by $\mathbf{A}$.

SVD

The above decomposition of $\mathbf{A}$ into

$$\mathbf{S}_2\Sigma\mathbf{S}_1^T$$

is called *singular value decomposition*.
For the ease of notation, let us denote $\mathbf{S}_2$ by $\mathbf{V}$ and $\mathbf{S}_1$ by $\mathbf{U}$ (getting thus rid of the subscripts). Then

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$$

Overview

- Da SVD agli spazi di documenti
- La *Singular Value Decomposition*
  - Definizione
  - Esempi
  - Riduzione della dimensionalita' del problema
- *Latent Semantic Analysis* e SVD
  - Interpretazione
  - *Latent Semantic Indexing*
- Applicazioni di LSA: *term* e *document clustering*
- *Latent Semantic kernels*

Analisi delle componenti principali

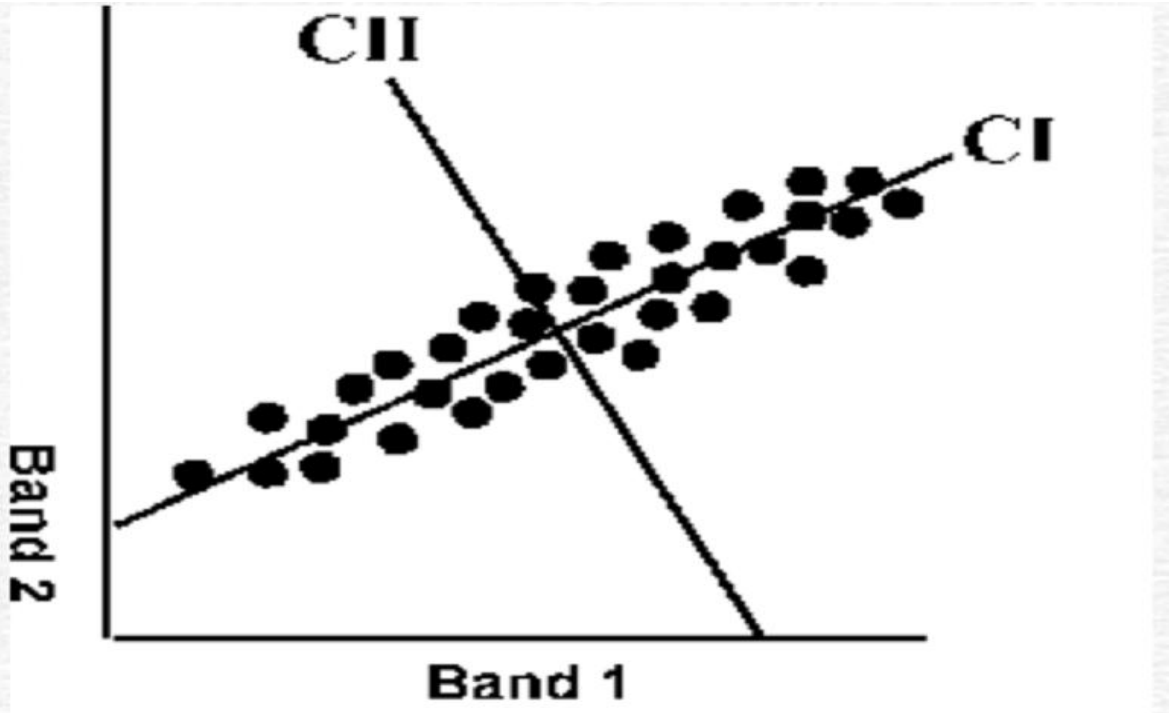


Figure 1 : The Principal Components Transform
(after Lillesand and Kiefer, 1994, 573)

LSA e Singular Value Decomposition

Dati $\mathscr{V}$ il vocabolario dei termini ($|\mathscr{V}| = m$) e
$\mathscr{T}$ l'insieme dei testi di una collezione ($|\mathscr{T}| = n$), applico alla matrice W degli m termini (righe) per gli n documenti (colonne) la *decomposizione in valori singolari* vista nella sezione precedente, (Golub and Kahan, 1965):

$$W = U\Sigma V^T$$

dove:

- U $(m \times r)$ con m vettori riga u_i e' singolare (i.e. $UU^T = I$)
- Σ $(r \times r)$ e' diagonale, con s_{ij} tc. $s_{ij} = 0 \quad \forall i = 1, \ldots, r$ ed i valori singolari nella diagonale principali tali che $s_i = s_{ii}$ e $s_1 \geq s_2 \geq \ldots \geq s_r > 0$
- V $(n \times r)$ con n vettori riga v_i e' singolare ($VV^T = I$)

(OSS: r e' il *rango* di W).

Le proprieta' di SVD

La decomposizione in valori singolari traduce la matrice W di partenza in:

$$W = U\Sigma V^T$$

dove:

- U e V sono le matrici dei *vettori singolari sinistri* e *destri* di W (cioe' gli *autovettori* o *eigenvectors* di WW^T e W^TW, rispettivamente)

Le proprieta' di SVD

La decomposizione in valori singolari traduce la matrice W di partenza in:

$$W = U\Sigma V^T$$

dove:

- U e V sono le matrici dei *vettori singolari sinistri* e *destri* di W (cioe' gli *autovettori* o *eigenvectors* di WW^T e W^TW, rispettivamente)
- le colonne di U e le righe di V definiscono uno spazio *ortonormale*, cioe' $UU^T = I$ e $VV^T = I$

Le proprieta' di SVD

La decomposizione in valori singolari traduce la matrice W di partenza in:

$$W = U\Sigma V^T$$

dove:

- U e V sono le matrici dei *vettori singolari sinistri* e *destri* di W (cioe' gli *autovettori* o *eigenvectors* di WW^T e W^TW, rispettivamente)
- le colonne di U e le righe di V definiscono uno spazio *ortonormale*, cioe' $UU^T = I$ e $VV^T = I$
- Σ e' la matrice diagonale dei valori singolari di W. I valori singolari s_i sono le radici degli autovalori λ_i di WW^T (o W^TW poichÃ¨ coincidono)

Le trasformazioni lineari ottenute ora sono due (come vedremo tra poco): WV e W^TU.

LSA: proprieta' di SVD

La decomposizione in valori singolari

$$W = U\Sigma V^T$$

si puo' approssimare con

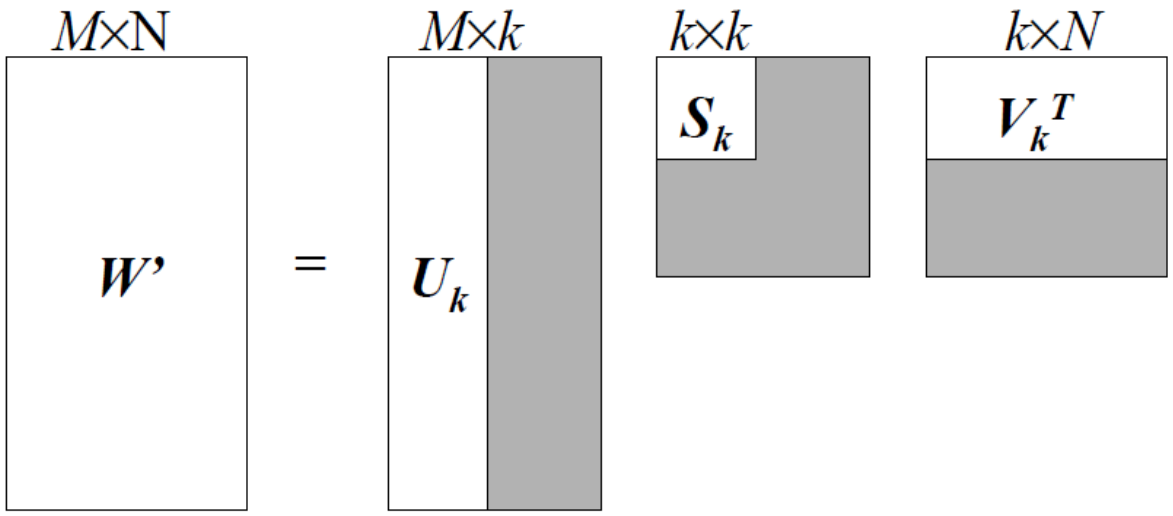
$$W \sim W' = U_k \Sigma_k V_k^T$$

trascurando le trasformazioni lineari di ordine superiore a $k << r$ in modo che:

- U_k $(M \times k)$ con M vettori riga u_i e' singolare ed ortonormale $(U_k U_k^T = I)$
- Σ_k $(k \times k)$ e' diagonale, con s_{ij} tc. $s_{ij} = 0 \quad \forall i = 1, ..., k$ ed i valori singolari nella diagonale principali tali che $s_i = s_{ii}$ e $s_1 \geq s_2 \geq ... \geq s_k > 0$
- V_k $(N \times k)$ con N vettori riga v_i e' singolare ed ortonormale $(V_k V_k^T = I)$

Riduzione del rango

Riduzione del rango



k e' il numero di valori singolari scelti per rappresentare i concetti nell'insieme dei documenti.
In genere, $k << r$ (rango iniziale).

LSA: proprieta' di SVD

La decomposizione in valori singolari

$$W \sim W' = U_k \Sigma_k V_k^T$$

ha un certo insieme di proprieta'

- la matrice Σ_k e' unica (anche se U e V non lo sono)
- per costruzione W' e' la matrice che soddisfa la decomposizione di ordine k **piu' vicina** a W (in norma euclidea)
- gli s_i sono le radici $s_i = \sqrt{\lambda_i}$ degli autovalori λ_i di WW^T
- le *componenti principali* del problema sono in relazione a U_k e V_k

LSA: interpretazione semantica

Si dice che in $W = U\Sigma V^T$, Σ cattura la *struttura semantica latente* dello spazio di partenza di W, $\mathscr{V} \times \mathscr{T}$ e che la riduzione a W' non ha effetto significativo su questa proprieta'.
... Vediamo perche':

LSA: interpretazione semantica

Si dice che in $W = U\Sigma V^T$, Σ cattura la *struttura semantica latente* dello spazio di partenza di W, $\mathscr{V} \times \mathscr{T}$ e che la riduzione a W' non ha effetto significativo su questa proprieta'.
... Vediamo perche':

- gli autovalori ed autovettori sono direzioni particolari dello spazio di partenza $\mathscr{V} \times \mathscr{T}$, determinate dalla trasformazione (termini in documenti) W.

LSA: interpretazione semantica

Si dice che in $W = U\Sigma V^T$, Σ cattura la *struttura semantica latente* dello spazio di partenza di W, $\mathscr{V} \times \mathscr{T}$ e che la riduzione a W' non ha effetto significativo su questa proprieta'.
... Vediamo perche':

- gli autovalori ed autovettori sono direzioni particolari dello spazio di partenza $\mathscr{V} \times \mathscr{T}$, determinate dalla trasformazione (termini in documenti) W.
- US si ottiene da $W = U\Sigma V^T$: infatti $WV = U\Sigma V^T V = U\Sigma$, cioe' per ogni riga (termine) i-esima di W (o U), $u_i\Sigma = w_i V$.

LSA: interpretazione semantica

Si dice che in $W = U\Sigma V^T$, Σ cattura la *struttura semantica latente* dello spazio di partenza di W, $\mathscr{V} \times \mathscr{T}$ e che la riduzione a W' non ha effetto significativo su questa proprieta'.
... Vediamo perche':

- gli autovalori ed autovettori sono direzioni particolari dello spazio di partenza $\mathscr{V} \times \mathscr{T}$, determinate dalla trasformazione (termini in documenti) W.
- US si ottiene da $W = U\Sigma V^T$: infatti $WV = U\Sigma V^T V = U\Sigma$, cioe' per ogni riga (termine) i-esima di W (o U), $u_i\Sigma = w_i V$.
- QUINDI: rappresentare i vettori dei termini (righe w_i della matrice W) mediante $u_i\Sigma$, SIGNIFICA: combinare linearmente gli elementi (correlazioni con tutti i documenti, v_j) della base ortonormale data da V (dopo il troncamento a k)

LSA: interpretazione semantica (cont'd)

Inoltre (per i **documenti**):

- VS si ottiene da $W = (U\Sigma V^T)$: infatti $W^T = (U\Sigma V^T)^T = V\Sigma U^T$, da cui $W^T U = V\Sigma$. Le colonne (documenti) w_j di W (o righe di V) sono tali che $v_j\Sigma = w_j U$.

LSA: interpretazione semantica (cont'd)

Inoltre (per i **documenti**):

- VS si ottiene da $W = (U\Sigma V^T)$: infatti $W^T = (U\Sigma V^T)^T = V\Sigma U^T$, da cui $W^T U = V\Sigma$. Le colonne (documenti) w_j di W (o righe di V) sono tali che $v_j\Sigma = w_j U$.
- QUINDI: rappresentare i vettori dei testi (colonne della matrice W) mediante $v_j\Sigma$, significa combinare linearmente (mediante Σ) le righe (correlazioni con i termini u_i) della base ortonormale data da U

LSA: interpretazione semantica (cont'd)

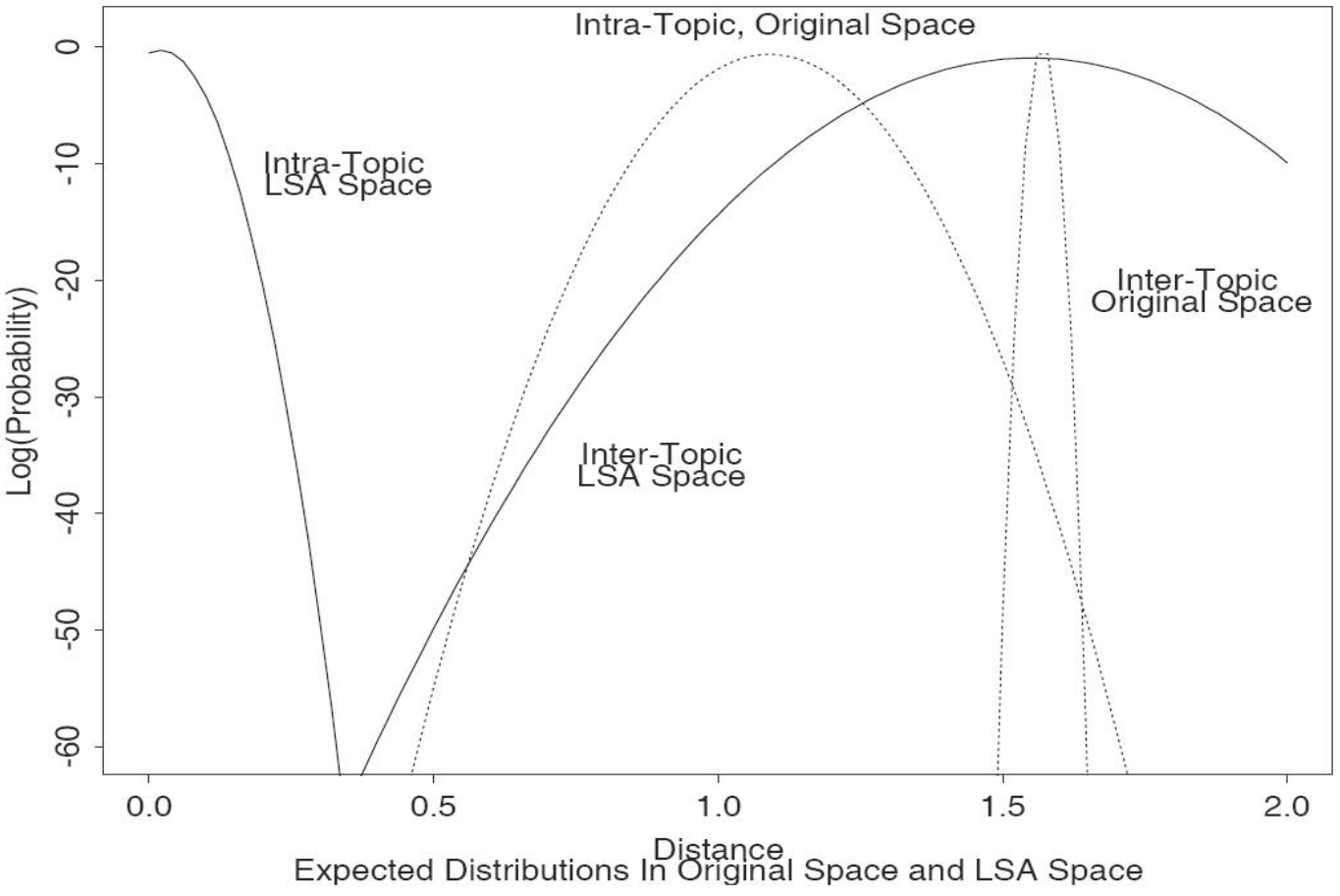
Inoltre (per i **documenti**):

- VS si ottiene da $W = (U\Sigma V^T)$: infatti $W^T = (U\Sigma V^T)^T = V\Sigma U^T$, da cui $W^T U = V\Sigma$. Le colonne (documenti) w_j di W (o righe di V) sono tali che $v_j\Sigma = w_j U$.
- QUINDI: rappresentare i vettori dei testi (colonne della matrice W) mediante $v_j\Sigma$, significa combinare linearmente (mediante Σ) le righe (correlazioni con i termini u_i) della base ortonormale data da U
- $U\Sigma$ e $V\Sigma$ rappresentano le mappature di termini in $\mathscr{V}$ e documenti in $\mathscr{T}$ nello spazio di dimensione k generato dalla SVD

LSA: interpretazione semantica (cont'd)

- Le trasformazioni sono date dalle combinazioni lineari sulle k dimensioni che corrispondono a concetti (temi). Infatti:
- Le matrici U e V sono basi ortonormali per lo spazio LS di dimensione k generato. Esso sono le direzioni privilegiate della trasformazione iniziale e quindi sono combinazioni lineari in WW^T (o W^TW) cioe' concetti determinati dal ricorrere degli stessi termini nei documenti (e viceversa).
- Ogni vettore di termine w_i dunque e' rappresentato in tale spazio tramite WV cioe' come combinazione dei documenti inziali, il che equivale a calcolare $U\Sigma$
- Analogamente per i documenti w_j mediante W^TU=$V\Sigma$

LSA: un esempio da una distribuzione artificiale



LSA: un esempio calcolato

Terms		d1	d2	d3		q
a		1	1	1		0
arrived		0	1	1		0
damaged		1	0	0		0
delivery		0	1	0		0
fire		1	0	0		0
gold	**W =**	1	0	1	**q =**	1
in		1	1	1		0
of		1	1	1		0
shipment		1	0	1		0
silver		0	2	0		1
truck		0	1	1		1

LSA: Calcolo $U\Sigma V^T$

$$U = \begin{bmatrix} -0.4201 & 0.0748 & -0.0460 \\ -0.2995 & -0.2001 & 0.4078 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.1576 & -0.3046 & -0.2006 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.3151 & -0.6093 & -0.4013 \\ -0.2995 & -0.2001 & 0.4078 \end{bmatrix} \quad S = \begin{bmatrix} 4.0989 & 0.0000 & 0.0000 \\ 0.0000 & 2.3616 & 0.0000 \\ 0.0000 & 0.0000 & 1.2737 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.4945 & 0.6492 & -0.5780 \\ -0.6458 & -0.7194 & -0.2556 \\ -0.5817 & 0.2469 & 0.7750 \end{bmatrix} \quad V^T = \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \\ -0.5780 & -0.2556 & 0.7750 \end{bmatrix}$$

LSA: Approssimazione del rango, $k = 2$

$$U = \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} \qquad S = \begin{bmatrix} 4.0989 & 0.0000 \\ 0.0000 & 2.3616 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.4945 & 0.6492 \\ -0.6458 & -0.7194 \\ -0.5817 & 0.2469 \end{bmatrix} \qquad V^T = \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \end{bmatrix}$$

LSA

Calcolo della similarita' Query-Doc

- Per n documenti, la matrice V contiene n righe, ognuna delle quali rappresenta le coordinate del documento d_i proiettato nello spazio LSA
- Una query q puo' essere trattata come uno pseudo-documento e proiettata anch'essa nello spazio LSA

LSA: $W = U\Sigma V^T$

Uso della trasformazione SVD

Se $W = U\Sigma V^T$ si ha anche che

- $V = W^T U \Sigma^{-1}$

LSA: $W = U\Sigma V^T$

Uso della trasformazione SVD

Se $W = U\Sigma V^T$ si ha anche che

- $V = W^T U\Sigma^{-1}$
- $d = d^T U\Sigma^{-1}$

LSA: $W = U\Sigma V^T$

Uso della trasformazione SVD

Se $W = U\Sigma V^T$ si ha anche che

- $V = W^T U\Sigma^{-1}$
- $d = d^T U\Sigma^{-1}$
- $q = q^T U\Sigma^{-1}$ (pseudo documento)

LSA: $W = U\Sigma V^T$

Uso della trasformazione SVD

Se $W = U\Sigma V^T$ si ha anche che

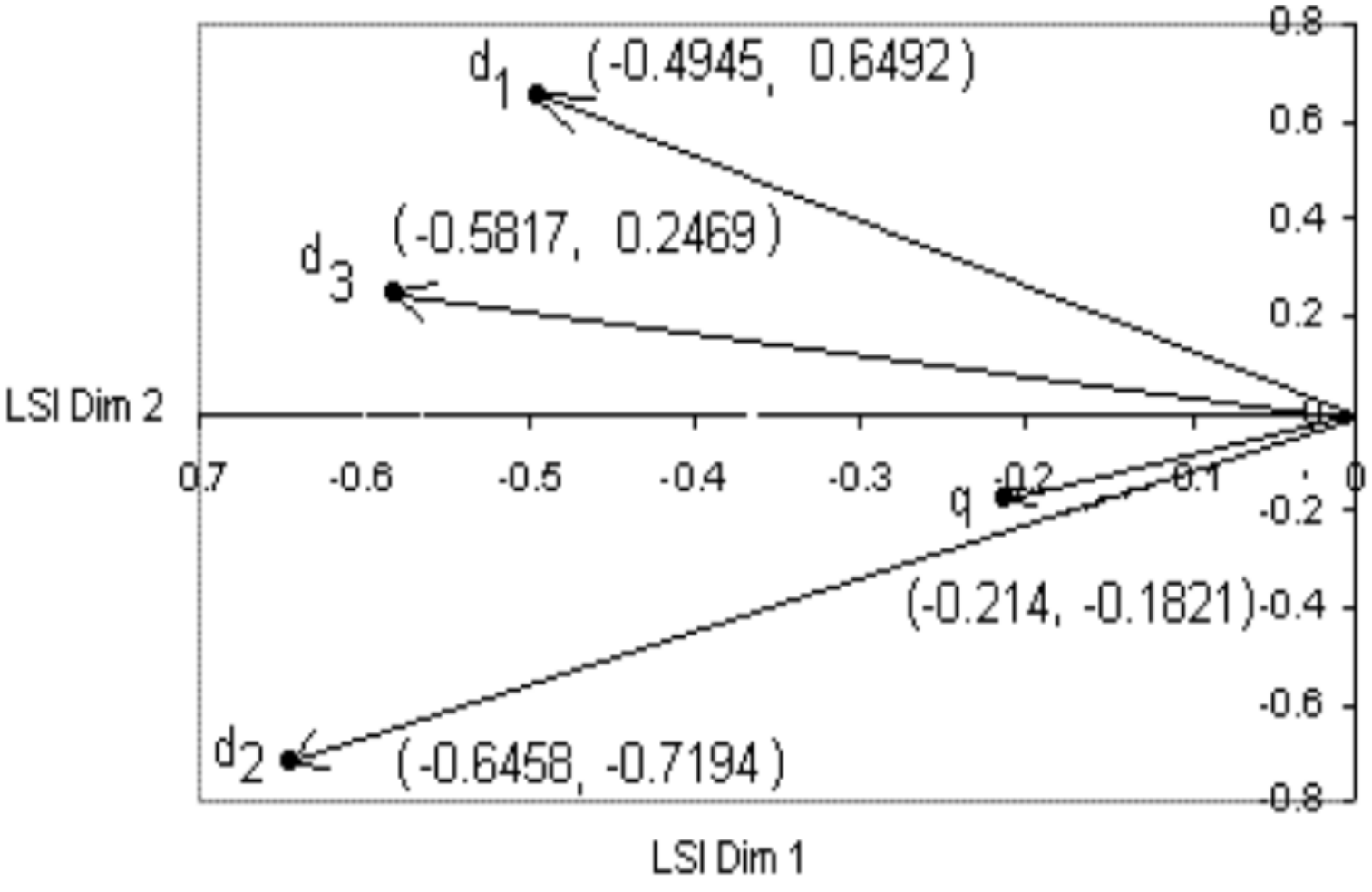
- $V = W^T U\Sigma^{-1}$
- $d = d^T U\Sigma^{-1}$
- $q = q^T U\Sigma^{-1}$ (pseudo documento)

A valle della riduzione dimensionale di ordine k

- $d = d^T U_k \Sigma_k^{-1}$
- $q = q^T U_k \Sigma_k^{-1}$ (pseudo documento)

Ne segue che: $sim(q,d) = sim(q^T U_k \Sigma_k^{-1}, d^T U_k \Sigma_k^{-1})$

LSA: Calcolo del query vector



LSA: Vettori della query e dei documenti

$$\mathbf{q} = \mathbf{q}^T \mathbf{U} \mathbf{S}^{-1}$$

$$\mathbf{q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} \begin{bmatrix} \frac{1}{4.0989} & 0.0000 \\ 0.0000 & \frac{1}{2.3616} \end{bmatrix}$$

$$\mathbf{q} = \begin{bmatrix} -0.2140 & -0.1821 \end{bmatrix}$$

LSA: Relazioni del secondo ordine

LSA e significato

- La rappresentazione in LSA dei termini coglie un numero maggiore di aspetti del significato dei termini, poiche' contribuiscono alla nuova rappresentazione tutte le co-occorrenze nei diversi contesti descritti dalla matrice W iniziale

- La rappresentazione di un termine t nel nuovo spazio non e' piu' il versore unitario $\vec{t}$ ortogonale a (e quindi indipendente da) tutti gli altri

LSA: Relazioni del secondo ordine

LSA e significato

- La rappresentazione in LSA dei termini coglie un numero maggiore di aspetti del significato dei termini, poiche' contribuiscono alla nuova rappresentazione tutte le co-occorrenze nei diversi contesti descritti dalla matrice W iniziale
- La rappresentazione di un termine t nel nuovo spazio non e' piu' il versore unitario $\vec{t}$ ortogonale a (e quindi indipendente da) tutti gli altri
- La similitudine tra due termini t_i e t_j dipende dalla trasformazione $U\Sigma$ e dipende quindi da tutte le co-occorrenze comuni con altri termini t_k (con $t_k \neq t_i, t_j$). Queste dipendenze costituiscono relazioni del *secondo ordine*.

LSA: SVD e term clustering

$$\mathbf{M} = \mathbf{K}_{t \times s} \mathbf{S}_{s \times s} \mathbf{D}^{T}_{s \times N}$$

M =

	d_1	d_2	d_3	d_4	d_5	d_6
shuttle	1	0	1	0	0	0
astronaut	0	1	0	0	0	0
moon	1	1	0	0	0	0
car	1	0	0	1	1	0
truck	0	0	0	1	0	1

= txs X sxs X sxN

K =

	dim_1	dim_2	dim_3	dim_4	dim_5
shuttle	-0.44	-0.30	0.57	0.58	0.25
astronaut	-0.13	-0.33	-0.59	0.00	0.73
moon	-0.48	-0.51	-0.37	0.00	-0.61
car	-0.70	0.35	0.15	-0.58	0.16
truck	-.0.26	0.65	-0.41	0.58	-0.09

S =

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	1.28	0.00	0.00
0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.39

LSA: SVD e term clustering

	d_1	d_2	d_3	d_4	d_5	d_6
shuttle	1	0	1	0	0	0
astronaut	0	1	0	0	0	0
moon	1	1	0	0	0	0
car	1	0	0	1	1	0
truck	0	0	0	1	0	1

M =

$$\mathbf{M} = \mathbf{K}_{t\times s}\mathbf{S}_{s\times s}\mathbf{D}^T_{s\times N}$$

= txs X sxs X sxN

K =

	dim_1	dim_2	dim_3	dim_4	dim_5
shuttle	-0.44	-0.30	0.57	0.58	0.25
astronaut	-0.13	-0.33	-0.59	0.00	0.73
moon	-0.48	-0.51	-0.37	0.00	-0.61
car	-0.70	0.35	0.15	-0.58	0.16
truck	-.0.26	0.65	-0.41	0.58	-0.09

S =

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	1.28	0.00	0.00
0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.39

LSA: SVD e term clustering

	d_1	d_2	d_3	d_4	d_5	d_6
shuttle	1	0	1	0	0	0
astronaut	0	1	0	0	0	0
moon	1	1	0	0	0	0
car	1	0	0	1	1	0
truck	0	0	0	1	0	1

M =

$$\mathbf{M} = \mathbf{K}_{t\times s}\mathbf{S}_{s\times s}\mathbf{D}^{T}_{s\times N}$$

= txs X sxs X sxN

K =

	dim_1	dim_2	dim_3	dim_4	dim_5
shuttle	-0.44	-0.30	0.57	0.58	0.25
astronaut	-0.13	-0.33	-0.59	0.00	0.73
moon	-0.48	-0.51	-0.37	0.00	-0.61
car	-0.70	0.35	0.15	-0.58	0.16
truck	-.0.26	0.65	-0.41	0.58	-0.09

S =

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	1.28	0.00	0.00
0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.39

LSA: Metriche di pesatura

In LSA i modelli di pesatura possono variare per migliorare la ricerca nello spazio delle trasformazioni lineari possibili

- Frequenza. c_{ij} (o le sue varianti normalizzate $\frac{c_{ij}}{|d_j|}$, $\frac{c_{ij}}{max_{lk}c_{lk}}$)
- (Landauer) $w_{ij} = \frac{log(c_{ij}+1)}{\frac{1+\sum_{j=1}^{N}\frac{c_{ij}}{t_i}log\frac{c_{ij}}{t_i}}{logN}} = \frac{log(c_{ij}+1)}{\frac{1+\sum_{j=1}^{N}P_{ij}logP_{ij}}{logN}}$
- (Bellegarda, LM modeling) $w_{ij} = (1-\varepsilon_i)\frac{c_{ij}}{n_j}$ con

$$\varepsilon_i = -\frac{1}{log_2 N}\sum_{j=1}^{N}\frac{c_{ij}}{t_i}log\frac{c_{ij}}{t_i}$$

LSA: Metriche tra termini

In LSA la similarita' tra i termini e' descritta dal prodotto

$$WW^T$$

e puo' essere calcolata come

$$U\Sigma V^T(U\Sigma V^T)^T = (U\Sigma V^T)(V\Sigma^T U^T) = U\Sigma\Sigma^T U^T = U\Sigma(U\Sigma)^T$$

cioe' moltiplicando la matrice delle (u_i) per Σ e poi applicando il prodotto scalare.

Applicazioni: *Indexing* (ass dei termini ai docs), *word/term clustering* (raggruppamento di termini).

LSA: Word Clustering

Cluster 1
Andy, antique, antiques, art, artist, artist's, artists, artworks, auctioneers, Christie's, collector, drawings, gallery, Gogh, fetched, hysteria, masterpiece, museums, painter, painting, paintings, Picasso, Pollock, reproduction, Sotheby's, van, Vincent, Warhol
Cluster 2
appeal, appeals, attorney, attorney's, counts, court, court's, courts, condemned, convictions, criminal, decision, defend, defendant, dismisses, dismissed, hearing, here, indicted, indictment, indictments, judge, judicial, judiciary, jury, juries, lawsuit, leniency, overturned, plaintiffs, prosecute, prosecution, prosecutions, prosecutors, ruled, ruling, sentenced, sentencing, suing, suit, suits, witness

LSA: Metriche tra documenti

In LSA la similarita' tra i documenti e' descritta dal prodotto

$$W^T W$$

e puo' essere calcolata come

$$(U\Sigma V^T)^T U\Sigma V^T = (V\Sigma^T U^T)(U\Sigma V^T) = V\Sigma\Sigma V^T = V\Sigma(V\Sigma)^T$$

cioe' moltiplicando la matrice delle (v_i) per Σ e poi applicando il prodotto scalare.

Applicazioni: *document clustering* (raggruppamento di docs) e *categorizzazione dei testi* (associazione di categorie ai documenti).

LSA: Osservazioni

- LSA e' una tecnica per ottenere sistematicamente le rotazioni necessarie nello spazio $\mathscr{V} \times \mathscr{T}$ ad allineare i nuovi assi con le dimensioni lungo le quali sussite la piu' ampia variazione tra i documenti
- Il primo asse (s_1) fornisce la variazione massima, il secondo quella massima tra le rimanenti e cosi' via
- Vengono trascurate (scelta di k) le dimensioni le cui variazioni sono trascurabili

LSA: Osservazioni (2)

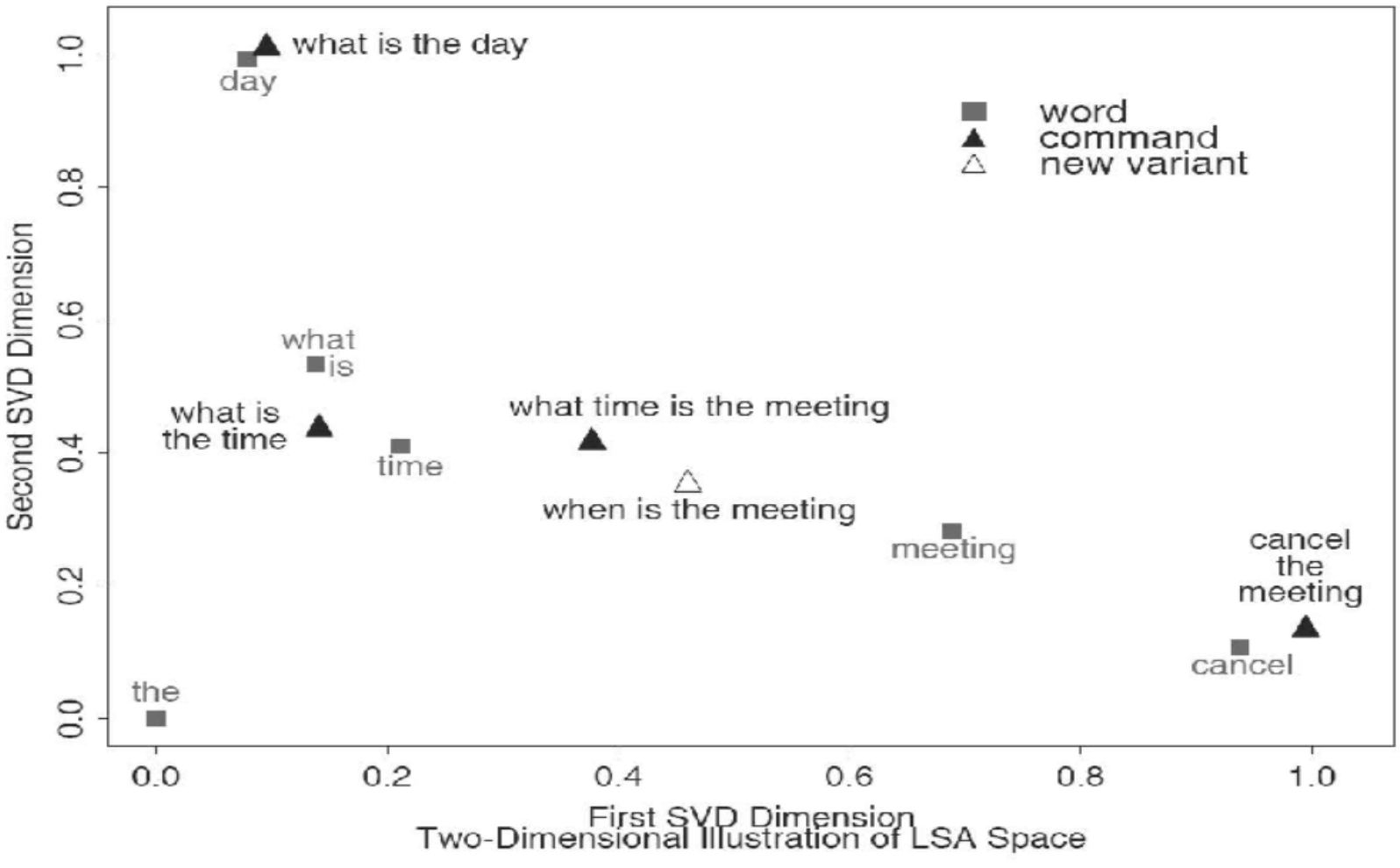
- LSA e' una tecnica per ottenere sistematicamente le rotazioni necessarie nello spazio $\mathscr{V} \times \mathscr{T}$ ad allineare i nuovi assi con le dimensioni lungo le quali sussite la piu' ampia variazione tra i documenti
- Ipotesi sottostanti:
  - misura la variazione attraverso la norma euclidea (minimizza lo scarto quadratico)
  - la distribuzione dei pesi legati ai termini nei documenti e' *normale* (i sistemi di pesatura tendono a consolidare questa proprieta')

LSA: Altre Applicazioni

Inferenza Semantica nel *Automatic Call Routing*

- Il *task* e' quello di mappare domande (per un *Call Center*) in una procedura (azione) di *reply*.
- Domande di training:
  (T1) *What is the time*, (T2) *What is the day*, (T3) *What time is the meeting*, (T4) *Cancel the meeting*
- *the* e' irrilevante, *time* e' ambiguo (tra 1 e 3)
- Domanda di ingresso: *when is the meeting* (classe corretta **T3**)

Automatic Call Routing su uno spazio bidimensionale



LSA vs. Machine Learning

Esiste una relazione tra LSA ed i modelli di *learning* ?

Induction in LSA

- LSA fornisce un metodo generale per la *stima di similarita'* quindi e' utilizzabile in tutti gli algoritmi induttivi per guidare la generalizzazione (ad es. iperpiano di separazione)

LSA vs. Machine Learning

Esiste una relazione tra LSA ed i modelli di *learning* ?

Induction in LSA

- LSA fornisce un metodo generale per la *stima di similarita'* quindi e' utilizzabile in tutti gli algoritmi induttivi per guidare la generalizzazione (ad es. iperpiano di separazione)
- LSA fornisce una trasformazione lineare dei vettori di attributi originali ispirata dalle caratteristiche del data set, quindi produce una nuova *metrica guidata dai dati* stessi. Questo meccanismo e' particolarmente importante quando non esistono modelli analitici (generativi) precisi delle distribuzioni dei dati iniziali (ad es. dati linguistici)

LSA vs. Machine Learning

Esiste una relazione tra LSA ed i modelli di *learning* ?

Induction in LSA

- LSA fornisce un metodo generale per la *stima di similarita'* quindi e' utilizzabile in tutti gli algoritmi induttivi per guidare la generalizzazione (ad es. iperpiano di separazione)
- LSA fornisce una trasformazione lineare dei vettori di attributi originali ispirata dalle caratteristiche del data set, quindi produce una nuova *metrica guidata dai dati* stessi. Questo meccanismo e' particolarmente importante quando non esistono modelli analitici (generativi) precisi delle distribuzioni dei dati iniziali (ad es. dati linguistici)
- LSA puo' essere applicato a dati non controllati (per esempio collezioni di documenti tematiche non annotate) e quindi estende il potere di generalizzazione dei metodi *supervised* sfruttando informazioni esterne al task. Questa proprieta' consente di indurre una conoscenza complementare al task stesso

LSA: Machine Learning tasks

Applicazioni di LSA in Relevance Feedback

- Automatic Global Analysis
- Stimatore di pseudo-rilevanza *prima* della espansione.

LSA: Machine Learning tasks

Applicazioni di LSA in Relevance Feedback

- Automatic Global Analysis
- Stimatore di pseudo-rilevanza *prima* della espansione.

LSA and language semantics

- SVD in distributional analysis: semantic spaces (see (Pado and Lapata, 2007))
- Word Sense Discrimination as clustering in LSA-like spaces (see Schutze, 1998)
- Word Sense Disambiguation in LSA spaces (see (Gliozzo et al., 2005), (Basili et al., 2006)
- Framenet predicate induction (see (Basili et al., 2008), (Pennacchiotti et al., 2008))

Frames as Conceptual Patterns

An example: the KILLING *frame*

Frame: KILLING	
A KILLER or CAUSE causes the death of the VICTIM.	
Frame Elements	
CAUSE	**The rockslide** killed nearly half of the climbers.
INSTRUMENT	It's difficult to suicide **with only a pocketknife**.
KILLER	**John** drowned Martha.
MEANS	The flood exterminated the rats **by cutting off access to food**.
VICTIM	John drowned **Martha**.
Predicates	annihilate.v, annihilation.n, asphyxiate.v,assassin.n, assassinate.v, assassination.n, behead.v, beheading.n, blood-bath.n, butcher.v, butchery.n, carnage.n, crucifixion.n, crucify.v, deadly.a, decapitate.v, decapitation.n, destroy.v, dispatch.v, drown.v, eliminate.v, euthanasia.n, euthanize.v, ...

Harvesting frames in semantic fields

Semantic Spaces (Pado and Lapata, 2007)

A **Semantic Space** for a set of N targets is 4-tuple $< B, A, S, V >$ where:

- B is the set of basic features (e.g. words co-occurring with the targets)
- A is a lexical association function that weights the correlations between $b \in B$ and the targets
- S is a similarity function between targets (i.e. in $\Re^{|B|} \times \Re^{|B|}$)
- V is a linear transformation over the original $N \times |B|$ matrix

Harvesting frames in semantic fields

Semantic Spaces: a definition

A Semantic Space for a set of N targets is 4-tuple $< B, A, S, V >$ where:

- B is the set of basic features (e.g. words co-occurring with the targets)
- A is a lexical association function that weights the correlations between $b \in B$ and the targets
- S is a similarity function between targets (i.e. in $\Re^{|B|} \times \Re^{|B|}$)
- V is a linear transformation over the original $N \times |B|$ matrix

Examples

- In IR systems targets are documents, B is the term vocabulary, A is the $tf \cdot idf$ score. The S function is usually the cosine similarity, i.e. $sim(\vec{t_1}, \vec{t_2}) = \frac{\sum_i t_{1i} \cdot t_{2i}}{||\vec{t_1}|| \cdot ||\vec{t_2}||}$

Harvesting frames in semantic fields

Semantic Spaces: a definition

A Semantic Space for a set of N targets is 4-tuple $< B, A, S, V >$ where:

- B is the set of basic features (e.g. words co-occurring with the targets)
- A is a lexical association function that weights the correlations between $b \in B$ and the targets
- S is a similarity function between targets (i.e. in $\Re^{|B|} \times \Re^{|B|}$)
- V is a linear transformation over the original $N \times |B|$ matrix

Examples

- In IR systems targets are documents, B is the term vocabulary, A is the $tf \cdot idf$ score. The S function is usually the cosine similarity, i.e. $sim(\vec{t_1}, \vec{t_2}) = \frac{\sum_i t_{1i} \cdot t_{2i}}{||\vec{t_1}|| \cdot ||\vec{t_2}||}$
- In Latent Semantic Analysis (Berry et al. 94) targets are documents (or dually words), and the SVD transformation is used as V

Harvesting ontologies in semantic fields

Semantic Spaces and Frame semantics

These lexicalized models corresponds to useful generalizations regarding *sinonimy*, *class membership* or *topical similarity*

- As frames are rich linguistic structures it is clear that more than one of such properties hold among members (i.e. LUs) of the same frame

Harvesting ontologies in semantic fields

Semantic Spaces and Frame semantics

These lexicalized models corresponds to useful generalizations regarding *sinonimy*, *class membership* or *topical similarity*

- As frames are rich linguistic structures it is clear that more than one of such properties hold among members (i.e. LUs) of the same frame
- *Topical similarity* plays a role as frames evoke events in very similar topical situations (e.g. KILLING vs. ARREST)

Harvesting ontologies in semantic fields

Semantic Spaces and Frame semantics

These lexicalized models corresponds to useful generalizations regarding *sinonimy*, *class membership* or *topical similarity*

- As frames are rich linguistic structures it is clear that more than one of such properties hold among members (i.e. LUs) of the same frame
- *Topical similarity* plays a role as frames evoke events in very similar topical situations (e.g. KILLING vs. ARREST)
- *Sinonimy* is also informative as LU's in a frame can be synonyms (such as *kid*, *child*), quasi-sinonyms (such as *mother* vs. *father*) and co-hyponims

Harvesting ontologies in semantic fields

Semantic Spaces and Frame semantics

These lexicalized models corresponds to useful generalizations regarding *sinonimy*, *class membership* or *topical similarity*

- As frames are rich linguistic structures it is clear that more than one of such properties hold among members (i.e. LUs) of the same frame
- *Topical similarity* plays a role as frames evoke events in very similar topical situations (e.g. KILLING vs. ARREST)
- *Sinonimy* is also informative as LU's in a frame can be synonyms (such as *kid*, *child*), quasi-sinonyms (such as *mother* vs. *father*) and co-hyponims

Which feature models and metrics correspond to a suitable geometrical notion of frame*hood*?

Latent Semantic Spaces

LSA and Frame semantics

In our approach SVD is applied to source co-occurrence matrices in order to

- Reduce the original dimensionality
- Capture *topical similarity* latent in the original documents, i.e. second order relations among targets

$$M = K_{txs} S_{sxs} D^T_{sxN}$$

M =	d_1	d_2	d_3	d_4	d_5	d_6
shuttle	1	0	1	0	0	0
astronaut	0	1	0	0	0	0
moon	1	1	0	0	0	0
car	1	0	0	1	1	0
truck	0	0	0	1	0	1

= txs X sxs X sxN

K =	dim_1	dim_2	dim_3	dim_4	dim_5
shuttle	-0.44	-0.30	0.57	0.58	0.25
astronaut	-0.13	-0.33	-0.59	0.00	0.73
moon	-0.48	-0.51	-0.37	0.00	-0.61
car	-0.70	0.35	0.15	-0.58	0.16
truck	-.0.26	0.65	-0.41	0.58	-0.09

S =

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	1.28	0.00	0.00
0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.39

Harvesting ontologies in semantic fields

Framehood in a semantic space

- Frames are rich polymorphic classes and clustering is applied for detecting multiple centroids

Harvesting ontologies in semantic fields

Framehood in a semantic space

- Frames are rich polymorphic classes and clustering is applied for detecting multiple centroids
- Regions of the space where LU's manifest are also useful for detecting the sentences that express the intended predicate semantics

Harvesting ontologies in semantic fields

Framehood in a semantic space

- Frames are rich polymorphic classes and clustering is applied for detecting multiple centroids
- Regions of the space where LU's manifest are also useful for detecting the sentences that express the intended predicate semantics

The clustering Algorithm

k-means

- Hard clustering algorithm fed with a fixed number of k randomly chosen seeds (centroids)
- Sensitive to the choice of k, and the seeding

Quality Threshold clustering ((Heyer et al., 1999))

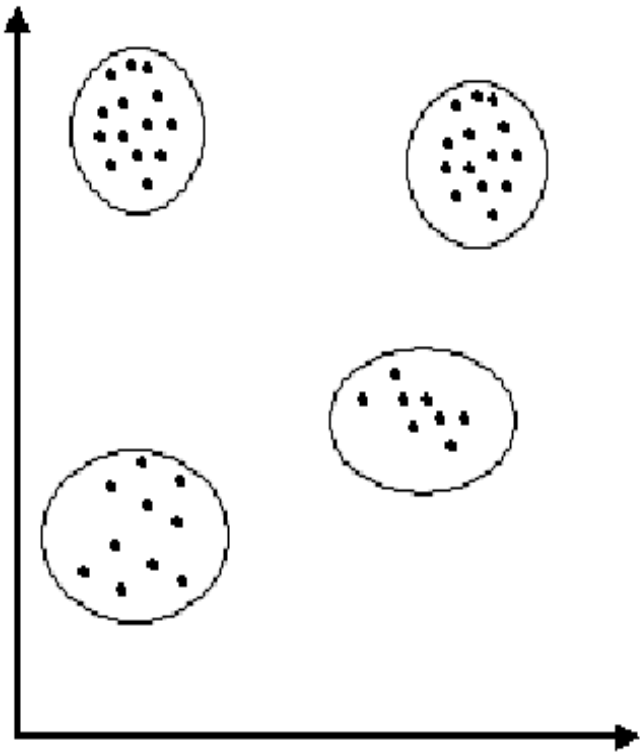
- Aggregative clustering similar to k-means with thresholds to increase flexibility
- Minimal infracluster similarity (activate *new seeds*)
- Minimal intra-cluster dissimilarity (activate *merge*)
- Maximal number of cluster members (activate *splits*)

The clustering Algorithm

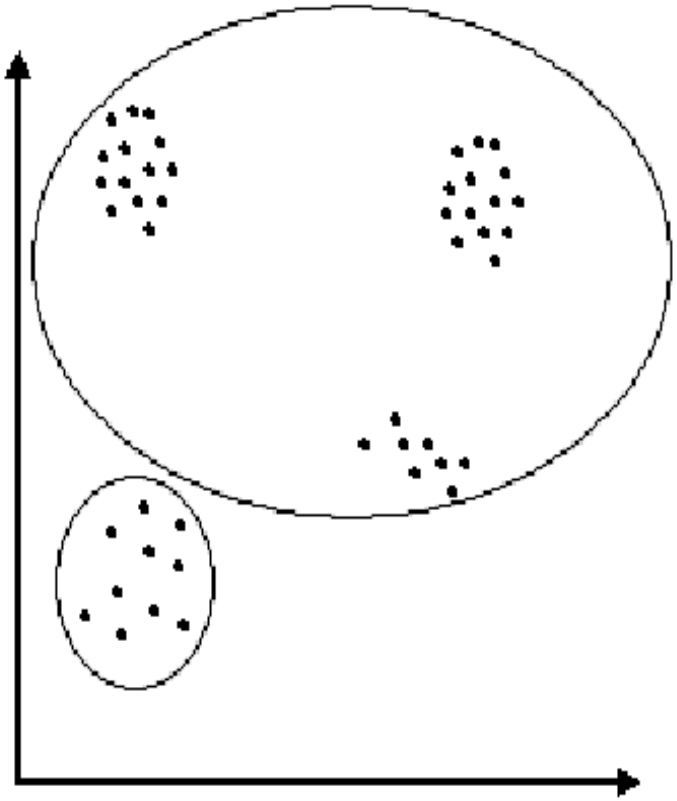
QT-clustering

Require: QT {Quality Threshold for clusters}
  repeat
    for all slot fillers w **do**
      Select C_x as the best cluster for w.
      if $sim(w, c_x) > QT$ **then**
        Generate new cluster C_w for w
      else
        Accept w into cluster C_x
      end if
    end for
  until No other shift is necessary or the maximum number of iteration is reached

The clustering Algorithm: Different settings



Clustering 1

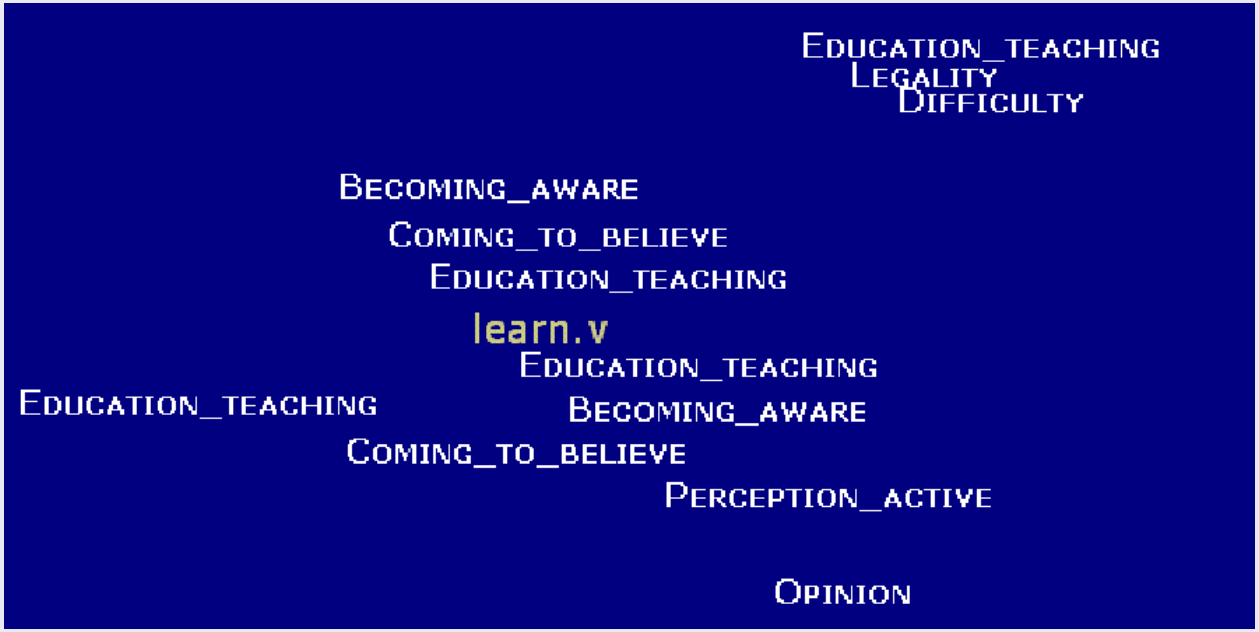


Clustering 2

Harvesting ontologies in semantic fields

Framehood in a semantic space

- Frames are rich polymorphic classes and clustering is applied for detecting multiple centroids
- Regions of the space where LU's manifest are also useful for detecting the sentences that express the intended predicate semantics



Evaluation: Current experimental set-up

The Corpus

- TREC 2005 vol. 2
- # of docs: about 230,000
- # of tokens: about 110,000,000 (more than 70,000 *types*)
- Source Dimensionality: 230,000 $\times$ 49,000
- LSA Dimensionality reduction: 7,700 $\times$ 100

Evaluation: Current experimental set-up

The Corpus

- TREC 2005 vol. 2
- # of docs: about 230,000
- # of tokens: about 110,000,000 (more than 70,000 *types*)
- Source Dimensionality: 230,000 $\times$ 49,000
- LSA Dimensionality reduction: 7,700 $\times$ 100

Syntactic and Semantic Analysis

- Parsing: Minpar (Lin,1998)
- Synonimy, hyponimy info: Wordnet 1.7
- Semantic Similarity Estimation: CD library (Basili et al,, 2004)
- Framenet: 2.0 version

Evaluating the LU Classification in English

English	Number of frames: 220 Number of LUs: 5042
	Most likely frames: *Self_Motion* (p=0.015), *Clothing* (p=0.014)

Table: The Gold Standard for the test over English

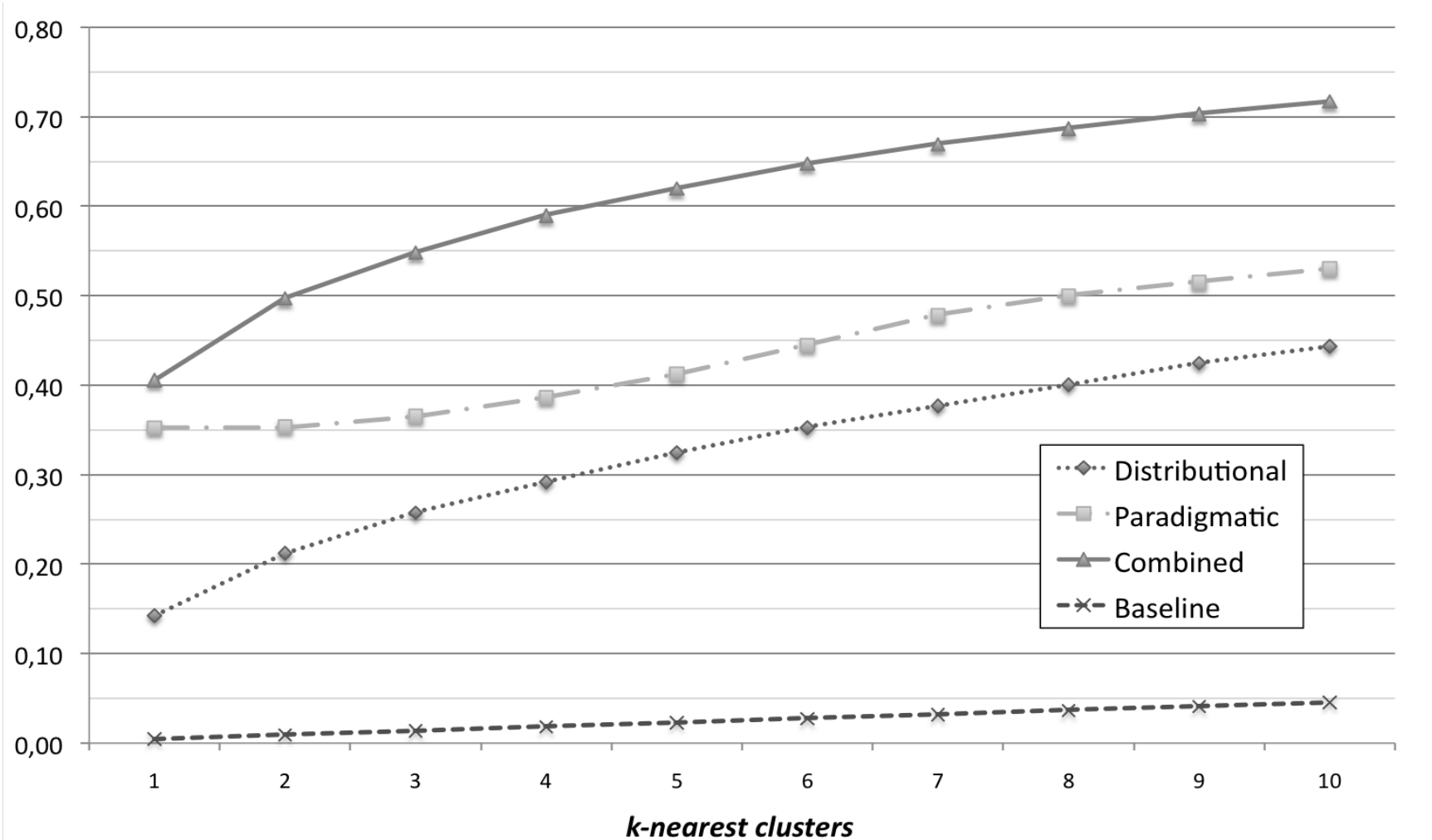
The measure is the **Accuracy**: the percentage of correctly classified *lu*'s by at least one of the proposed k choices.

The impact of LU Classification on the English data set

	Nouns	Verbs
Targeted Frames	220	220
Involved LUs	2,200	2,180
Average LUs per frame	10.0	9.91
LUs covered by WordNet	2,187	2,169
Number of Evoked Senses	7,443	11,489
Average Polysemy	3.62	5.97
Represented words (i.e. $\sum_F W_F$)	2,145	1,270
Average represented LUs per frame	9.94	9.85
Active Lexical Senses (L_F)	3,095	2,282
Average Active Lexical Senses ($\lvert L_F \rvert / \lvert W_F \rvert$) per word over frames	1.27	1.79
Active synsets (S_F)	3,512	2,718
Average Active synsets ($\lvert S_F \rvert / \lvert W_F \rvert$) per word over frames	1.51	2.19

Table: Statistics on nominal and verbal senses in the paradigmatic model of the English FrameNet

Comparative evaluation for English: accuracy



LSA e metodi Kernel

Le funzioni kernel $K(\vec{o}_i, \vec{o})$ possono essere usate per la stima (implicita) della similarita' tra due termini o documenti, o_i e o, in spazi complessi al fine di addestrare una SVM secondo l'equazione:

$$h(\vec{x}) = sgn\left(\sum_{i=1}^{l} \alpha_i K(o_i, o) + b\right)$$

dove $\vec{x} = \phi(o)$, ed l dipende dal learning set.
Alcuni modelli che utilizzano LSA per la definizione di $K(o_i, o)$ sono stati definiti per problemi di

- Word Sense Disambiguation (classificazione occorrenze di una parola in sensi)
- Text Categorization (categorizzazione testi)

LSA-based Domain Kernels: Applicazione ai Lessici

Assunzioni:

- o_i rappresenta un "*termine*" $\vec{t_i}$
- Lo spazio di rappresentazione di partenza per i $\vec{t_i}$ e' quello di un VSM tradizionale (pesatura dei termini tramite la loro pesatura $TF \times IDF$ nei documenti)
- La matrice T di partenza e' quindi termini per doc
- I vettori risultanti rappresentano le espressioni lessicali nello spazio (dei concetti) LSA

LSA-based Domain Kernels (2)

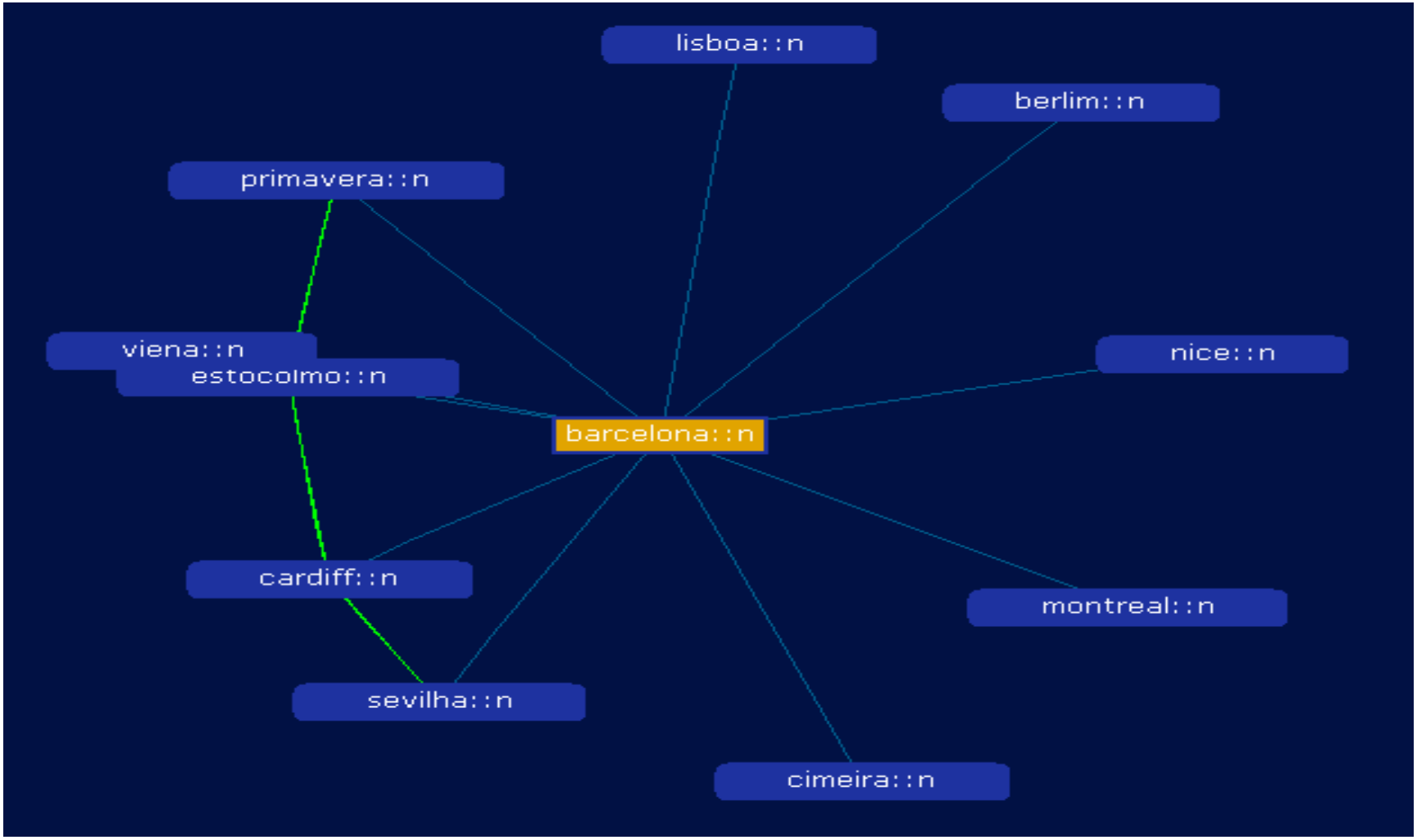
Processo:

- Applico la trasformazione LSA di dimensione k, $o_i \leftarrow \vec{t_i}$ (OSS: gli o_i sono vettori in LSA)
- Assumo la metrica tra termini $\vec{t_i}$ come la similarita' tra oggetti di tipo o_i (cioe' nel *Latent Semantic Space*)
- Addestro un categorizzatore (per esempio per riconoscere la categoria semantica dei termini) definendo una funzione kernel $K(.,.)$ nel seguente modo:
  $K(\vec{t_i},\vec{t}) = K(\phi^{-1}(o_i),\phi^{-1}(o)) \doteq K_{LSA}(o,o_i)$ dove:

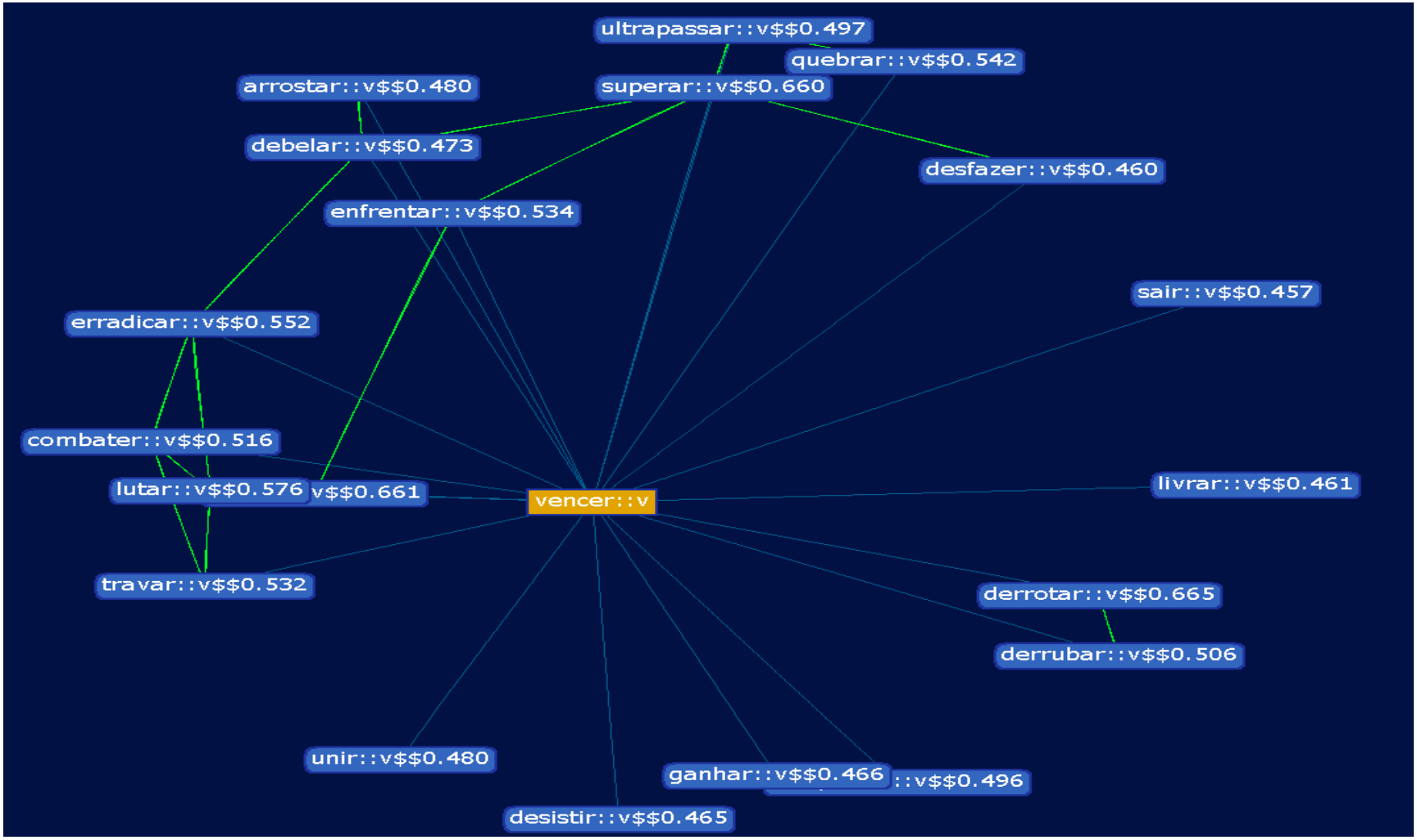
$$K_{LSA}(o,o_i) = o_i \otimes_{LSA} o$$

OSS: $\otimes_{LSA}$ indica il prodotto scalare tra gli oggetti o_i (i termini) nello spazio LSA.

LSA space for terms in a foreign language (portuguese)



LSA space for terms in a foreign language (portuguese)



LSA-based Domain Kernels (3)

LSA determina la trasformazione SVD e la approssimazione di ordine k.
k e' la dimensione nello spazio trasformato, cioe' il numero di componenti principali del problema inziale (VSM semplice)
Interpretando tali componenti come *domini*:

- o_i risulta la descrizione di un termine $\vec{t_i}$ nei diversi domini
- termini simili secondo $K_{LSA}(o_i, o)$ condividono il maggior numero di domini
- il kernel risultante $K_{LSA}(o_i, o)$ e' detto *Latent Semantic Kernel* (Cristianini&Shawe-Taylor,2004) o *domain kernel* (Gliozzo & Strapparava,2005).

LSA-based Domain Kernels:

Applicazione alla *Text Categorization*

- un documento x_i (come un termine) puo' essere mappato in uno spazio di tipo LSA, $o_i \leftarrow \vec{x}_i$
- le k componenti di o_i rappresentano la descrizione di x_i nel nuovo spazio
- il kernel $K_{LSA}(o_i, o)$ risultante cattura le similarita' mediante una descrizione di dominio di $\vec{x}_i$
- l'addestramento della SVM genera l'equazione dell'iperpiano secondo LSA:

$$f(\vec{x}) = \left(\sum_{i=1}^{l} \alpha_i K_{LSA}(o_i, o_j) + b \right)$$

OSS: LSA puo' essere computato in una collezione *esterna* al data set di training.

References

SVD and LSA

- Susan T. Dumais, Michael Berry, Using Linear Algebra for Intelligent Information Retrieval, SIAM Review, 1995, 37, 573–595
- G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, K. E. Lochbaum, Information retrieval using a singular value decomposition model of latent semantic structure, SIGIR '88: Proc. of the ACM SIGIR conference on Research and development in Information Retrieval, 1988

Non linear embeddings

- S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 2000.
- B. J. Tenenbaum, V. Silva, and J. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science, pp.2319-2323, 2000
- Xiaofei He, Partha Niyogi, Locality Preserving Projections. Proceedings of Advances in Neural Information Processing Systems, Vancouver, Canada, 2003.

References

LSA in language learning

- Hinrich Schutze, Automatic word sense discrimination, Computational Linguistics, 24(1), 1998.
- Beate Dorow and Dominic Widdows, Discovering Corpus-Specific Word Senses. EACL 2003, Budapest, Hungary. Conference, pages 79-82
- Basili Roberto, Marco Cammisa, Alfio Gliozzo, Integrating Domain and Paradigmatic Similarity for Unsupervised Sense Tagging, 17th European Conference on Artificial Intelligence (ECAI06), Riva del Garda, Italy, 2006.
- Sebastian Pado and Mirella Lapata. Dependency-based Construction of Semantic Space Models. In Computational Linquistics, Vol. 33(2):161-199, 2007.
- Basili R. Pennacchiotti M., Proceedings of GEMS "*Geometrical Models of Natural Language Semantics*", 2009 (http://www.aclweb.org/anthology/W/W09/#0200), 2010 (http://aclweb.org/anthology-new/W/W10/#2800)