Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

Recuperación de Información Inteligente

Análisis de Semántica Latente (LSI/LSA)

Minería de la Web

Marcelo Paulo Amaolo

Presentación

- Marcelo Paulo Amaolo
- Docente
  - Ingeniería de Software y Fundamentos Teóricos del Departamento Ciencias de la Computación de la Universidad Nacional del Comahue
  - Investigador del Grupo de Investigación de Ingeniería de Software (GIISCO)
- **Dir. Gral. Digesto Jurídico de la Provincia de Neuquén**
  - **Análisis Normativa y Resguardo de Normativas emanadas del PEP**
- **mamaolo@uncoma.edu.ar**
  mamaolo@neuquen.gov.ar

Presentación

Esquema

- Introducción
- IR: Algunos problemas - Ejemplo
- Definición (LSI/LSA): Motivación, principios construcción
- Bases Matemáticas: SVD
- Un ejemplo de juguete: Cálculo de Matrices, Visualización, aplicación de conceptos
- Otras Comparaciones
- Crecimiento del corpus: Costo y Limitaciones
- Areas de Aplicación y Aplicaciones Concretas

Bibliografía Básica Utilizada

- "Using linear algebra for Intelligent Information Retrieval", Berry M.W., Dumais S.T., O'Brien G.W., 1995.
- "Indexing by Latent Semantic Analysis", Deerwester S., Dumais S.T., Harshman R., 1997.
- "An Introduction to Latent Semantic Analysis", Lander T.K., Foltz P.W., Laham, D., 1998.
- Información de la Web

Introducción

- Análisis de Semántica Latente (LSA)
  - teoría y método para extraer y representar el significado del uso contextual de palabras
  - determinación de la similaridad del significado de palabras y pasajes de palabras analizando un corpus de texto
  - la agregación de todas las palabas de un contexto en el cual una palabra puede o no aparecer, provee un conjunto de restricciones mutuas que determinan la similaridad de significado de las palabras o conjunto de palabras

IR: Algunos problemas

- Analogía
  - usuario buscando datos en la web
  - proceso de memoria semántica de las personas
- El usuario tiene una “idea”
- Debe expresar esas ideas en palabras
- El sistema trata de buscar el texto con el mismo significado
- Exito si el texto representa la idea

Algunos problemas

- ¿Y si las palabras utilizadas no son las "apropiadas" para el corpus?
  - Padre, papá, progenitor y elefante
  - Padre, papá y progenitor son "sinónimos"
  - Buscar por palabras "padre" tiene la misma distancia con "progenitor" o "papá" que con elefante

Algunos problemas

- Sinonimia
  - enorme ocurrencia de sinónimos
  - disminuye la “completitud” (recall)
- Polisemia
  - recuperación de documentos irrelevantes
  - disminuye la “sanidad” (precision)
- Ruido
  - búsqueda booleana de palabras específicas
  - contenido de documentos no relacionado

LSI: Motivación

- Forma útil de establecer relaciones entre palabras y documentos.
- Descubrir palabras que “realmente” estén relacionados (implicados) por la consulta.
- LSI permite realizar la búsqueda de “conceptos” y no de palabras
- LSI puede recuperar documentos relacionados a la búsqueda del usuario, aunque la consulta y los documentos no compartan palabras

LSI: Motivación

- LSI asume que existe una estructura LATENTE en el uso de las palabras – oculta por la variabilidad de la elección de palabras
- Análogo
  - Modelo Señal + Ruido del Procesamiento de Señales

Ejemplo simple

- Documentos:
  - Doc 1: “Indexación de base de datos para recuperación y acceso de documentos”
  - Doc 2: “Teoría de Información de Computadora”
  - Doc 3: “Recuperación de Información por Computadora”
- Consulta:
  - Búsqueda de Información por Computadora

Algunos problemas: ejemplo

	Acceso	Documento	Recuperación	Información	Teoría	Base de Datos	Indexación	Computadora
Doc 1								
Doc 2								
Doc 3								

Algunos problemas: ejemplo

	Acceso	Documento	Recuperación	Información	Teoría	Base de Datos	Indexación	Computadora	RELEVANTE	COINCIDE
Doc 1										
Doc 2										
Doc 3										

Algunos problemas: ejemplo

	Acceso	Documento	Recuperación	Información	Teoría	Base de Datos	Indexación	Computadora	RELEVANTE	COINCIDE
Doc 1										
Doc 2										
Doc 3										

Algunos problemas: ejemplo

	Acceso	Documento	Recuperación	Información	Teoría	Base de Datos	Indexación	Computadora	RELEVANTE	COINCIDE
Doc 1										
Doc 2										
Doc 3										

Consulta: Búsqueda de Información por Computadora

Algunos problemas: ejemplo

	Acceso	Documentos	Recuperación	Información	Teoría	Base de Datos	Indexación	Computadora	RELEVANTE	COINCIDE
Doc 1										
Doc 2										
Doc 3										

Consulta: Búsqueda de Información por Computadora

LSI: Principios

- Mapea los documentos y las palabras a un Espacio Vectorial Multidimensional.
- Cada dimensión del espacio corresponde a un concepto de la colección de documentos.
- Así, los tópicos subyacentes se codifican con un vector.
- Las palabras relacionadas en un documento y una consulta se mapean a vectores cercanos.

LSI: Principios

- Basado en una técnica estadísco-algebraica (SVD) que extrae e infiere las relaciones esperadas del uso contextual de palabras en documentos
- No utiliza construcciones manuales, diccionarios, bases de conocimiento, redes semánticas, gramáticas, ontologías, corpus paralelos, etc.
- Entrada: sólo texto crudo

LSI: Principios / Construcción

- Se utiliza un corpus de entrenamiento de un dominio de interés
- Naturaleza de los documentos
  - Una oración, un párrafo, un capítulo, etc.
- Vocabulario de palabras
  - Tamaño dado por el corpus
  - Se eliminan palabras no conceptuales (stopwords)
  - Pueden utilizarse “giros” (+ de 1 palabra)

Bases Matemáticas: SVD

- Descomposición de valores singulares (Singular Valued Decomposition – SVD)
- Recordemos
  - Autovectores extendidos a matrices (> a <)
  - Valor indica:
    - "Cantidad" del vector presente en la matriz
    - Impacto de las direcciones en el comportamiento de la matriz

SVD

- Con los N valores más grandes, mostramos un error de aproximación por mínimos cuadrados a la matriz original usando el menor conjunto de números (sacamos aquellos con menor impacto)
- Matriz Reducida:
  - Compresión de la original
  - “Sacar detalle” actúa como un “reductor de ruido” o “reductor de pormenores poco válidos”
  - Puede mejorar la performance (depende del contexto)
- Esto hace LSI posible.

Un ejemplo de juguete

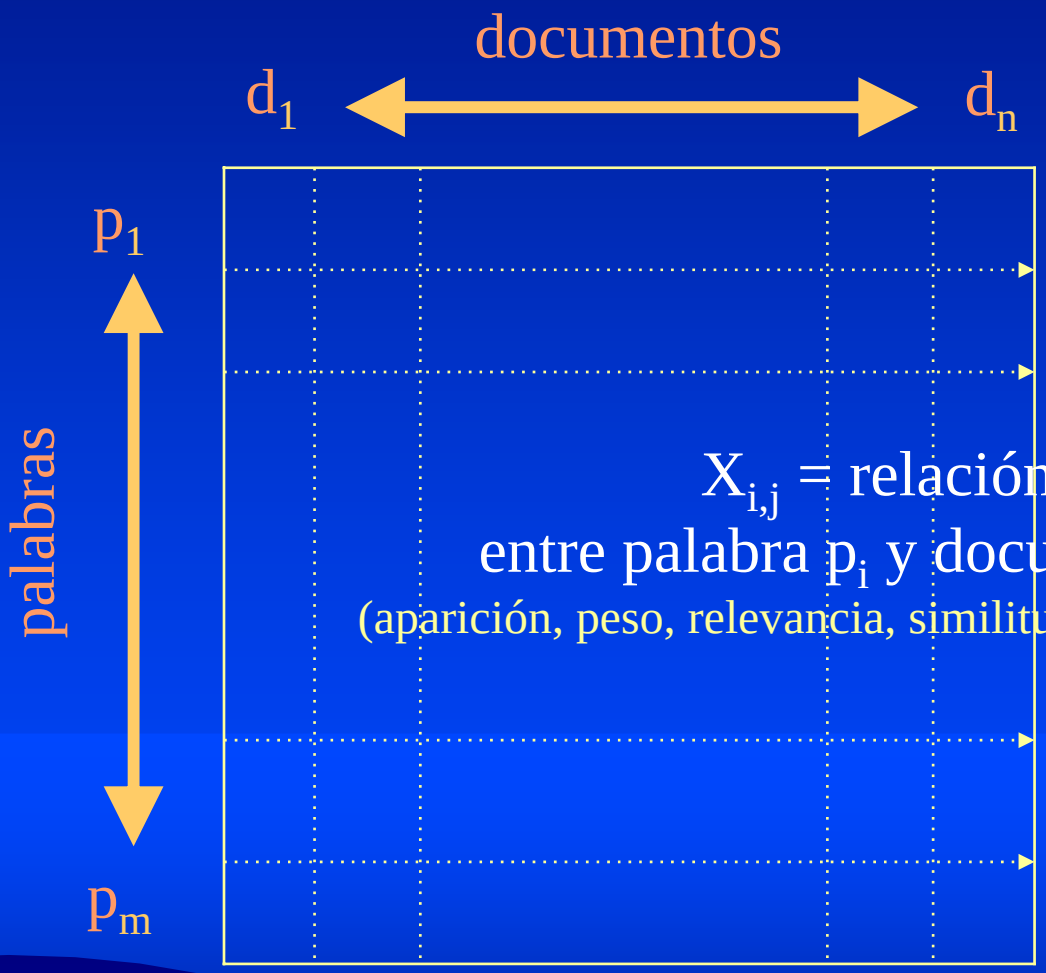
B1 A Course on Integral Equations
B2 Attractors for Semigroups and Evolution Equations
B3 Automatic Differentiation of Algorithms: Theory, Implementation, and Application
B4 Geometrical Aspects of Partial Differential Equations
B5 Ideals, Varieties, and Algorithms - An Introduction to Computational Algebraic Geometry and Commutative Algebra
B6 Introduction to Hamiltonian Dynamical Systems and the N-Body Problem
B7 Knapsack Problems: Algorithms and Computer Implementations
B8 Methods of Solving Singular Systems of Ordinary Differential Equations
B9 Nonlinear Systems
B10 Ordinary Differential Equations
B11 Oscillation Theory for Neutral Differential Equations with Delay
B12 Oscillation Theory of Delay Differential Equations
B13 Pseudodifferential Operators and Nonlinear Partial Differential Equations
B14 Sine Methods for Quadrature and Differential Equations
B15 Stability of Stochastic Differential Equations with Respect to Semi-Martingales
B16 The Boundary Integral Approach to Static and Dynamic Contact Problems
B17 The Double Mellin-Barnes Type Integrals and Their Applications to Convolution Theory

Un ejemplo de juguete

B1 A Course on Integral Equations
B2 Attractors for Semigroups and Evolution Equations
B3 Automatic Differentiation of Algorithms: Theory, Implementation, and Application
B4 Geometrical Aspects of Partial Differential Equations
B5 Ideals, Varieties, and Algorithms - An Introduction to Computational Algebraic Geometry and Commutative Algebra
B6 Introduction to Hamiltonian Dynamical Systems and the N-Body Problem
B7 Knapsack Problems: Algorithms and Computer Implementations
B8 Methods of Solving Singular Systems of Ordinary Differential Equations
B9 Nonlinear Systems
B10 Ordinary Differential Equations
B11 Oscillation Theory for Neutral Differential Equations with Delay
B12 Oscillation Theory of Delay Differential Equations
B13 Pseudodifferential Operators and Nonlinear Partial Differential Equations
B14 Sine Methods for Quadrature and Differential Equations
B15 Stability of Stochastic Differential Equations with Respect to Semi-Martingales
B16 The Boundary Integral Approach to Static and Dynamic Contact Problems
B17 The Double Mellin-Barnes Type Integrals and Their Applications to Convolution Theory

Un ejemplo de juguete

Matriz X



$X_{i,j}$ = relación
entre palabra p_i y documento d_j
(aparición, peso, relevancia, similitud, esquema tf-idf)

Un ejemplo de juguete

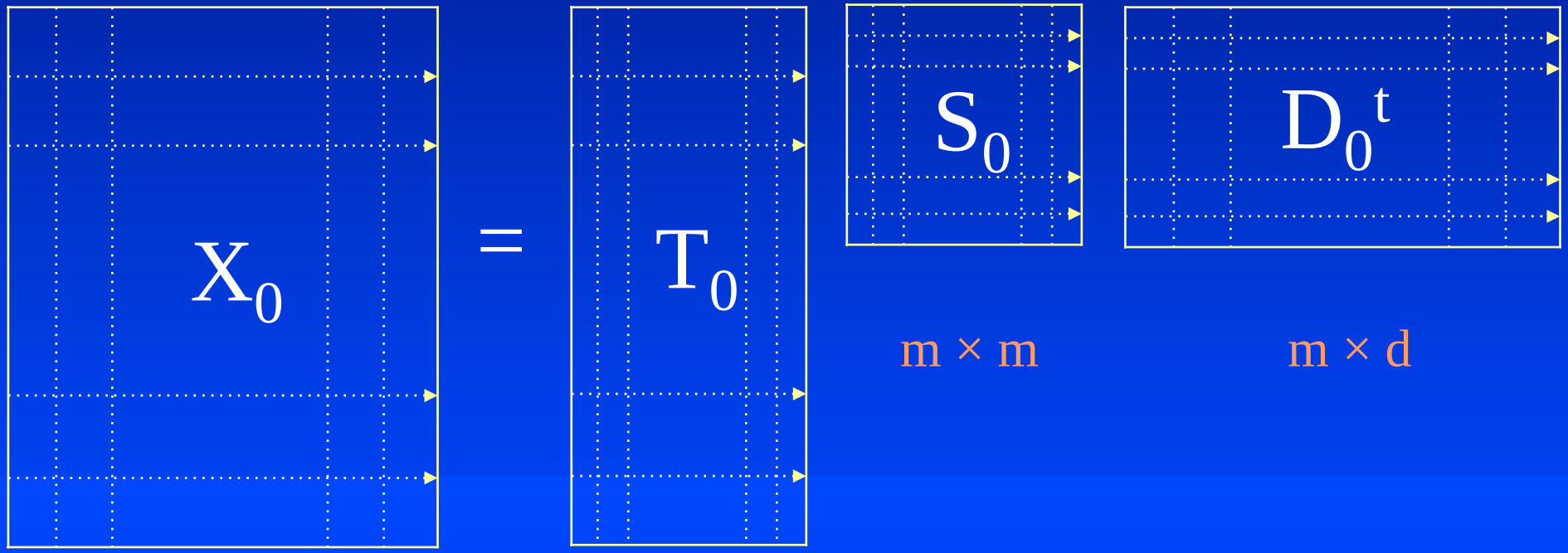
Palabras	Documentos (rala = 19,12%)																
	b1	b2	b3	b4	b5	b6	b7	b8	b9	b10	b11	b12	b13	b14	b15	b16	b17
algorithms	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
application	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
delay	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
differential	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0
equations	1	1	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0
implementation	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
integral	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
introduction	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
methods	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
nonlinear	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
ordinary	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
oscillation	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
partial	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
problem	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
systems	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
theory	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1

SVD

documentos

conceptos

palabras



p × d

p × m

m × m

m × d

Un ejemplo de juguete: SVD

documentos

conceptos

palabras

X_0 = T_0 S_0 D_0^t

16 × 17

16 × 14

14 × 14

m × 17

Un ejemplo de juguete: SVD

- $X_0 = T_0 S_0 D_0^t$
  - T_0 y D_0 ortonormales ($T_0 T_0^t = I$, $D_0 D_0^t = I$)
  - S_0 diagonal
  - T_0 es la matriz de autovectores de XX^t
  - D_0 es la matriz de autovectores de X^tX
  - S_0^2 es la matriz de autovalores
    - $s_{i,i}$ (raíces cuadradas de autovalores de XX^t y X^tX)

Un ejemplo de juguete: SVD

$T_0^t =$

0,016	0,027	0,178	0,601	0,669	0,015	0,052	0,007	0,150	0,081	0,150	0,178	0,141	0,011	0,095	0,205
-0,432	-0,376	-0,169	0,119	0,121	-0,360	-0,225	-0,112	0,113	0,067	0,113	-0,169	0,097	-0,236	0,040	-0,545
0,289	-0,018	-0,284	0,009	0,029	0,186	0,059	0,315	0,190	0,146	0,190	-0,284	0,057	0,460	0,468	-0,302
0,380	-0,162	0,122	0,159	-0,142	0,244	-0,807	0,114	0,003	0,006	0,003	0,122	0,024	-0,162	-0,033	-0,040
-0,279	-0,222	0,368	-0,054	-0,176	-0,275	-0,080	0,267	0,035	-0,057	0,035	0,368	-0,336	0,210	0,480	0,146
-0,007	-0,258	0,136	0,004	0,054	-0,063	0,044	0,177	-0,454	0,339	-0,454	0,136	0,434	0,328	-0,136	-0,122
0,126	-0,324	0,068	0,021	0,192	0,052	0,087	0,078	0,028	-0,608	0,028	0,068	-0,240	0,366	-0,433	-0,256
-0,194	-0,128	0,095	0,077	-0,157	0,375	-0,067	-0,732	0,075	0,175	0,075	0,095	-0,042	0,410	0,056	-0,033
0,261	-0,239	0,090	-0,446	0,465	0,146	0,099	-0,131	-0,082	0,323	-0,082	0,090	-0,416	-0,262	0,115	-0,149
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,707	0,000	-0,707	0,000	0,000	0,000	0,000	0,000
-0,351	0,231	-0,140	-0,255	0,396	0,213	-0,351	-0,032	-0,249	-0,395	-0,249	-0,140	0,085	0,096	0,296	0,091
0,153	-0,216	0,211	-0,368	-0,008	0,054	0,154	-0,116	0,209	-0,294	0,209	0,211	0,647	-0,173	0,207	-0,005
0,214	-0,133	-0,035	0,429	-0,194	0,087	0,277	-0,192	-0,316	-0,312	-0,316	-0,035	-0,087	-0,286	0,420	-0,168
-0,440	-0,140	0,056	0,074	-0,100	0,690	0,194	0,402	0,051	0,068	0,051	0,056	-0,008	-0,252	-0,099	-0,084
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000

Un ejemplo de juguete: SVD

$S_0 =$

4,531													
	2,758												
		2,421											
			1,904										
				1,881									
					1,745								
						1,660							
							1,276						
								1,063					
									1,000				
										0,823			
											0,616		
												0,423	
													0,183

Un ejemplo de juguete: SVD

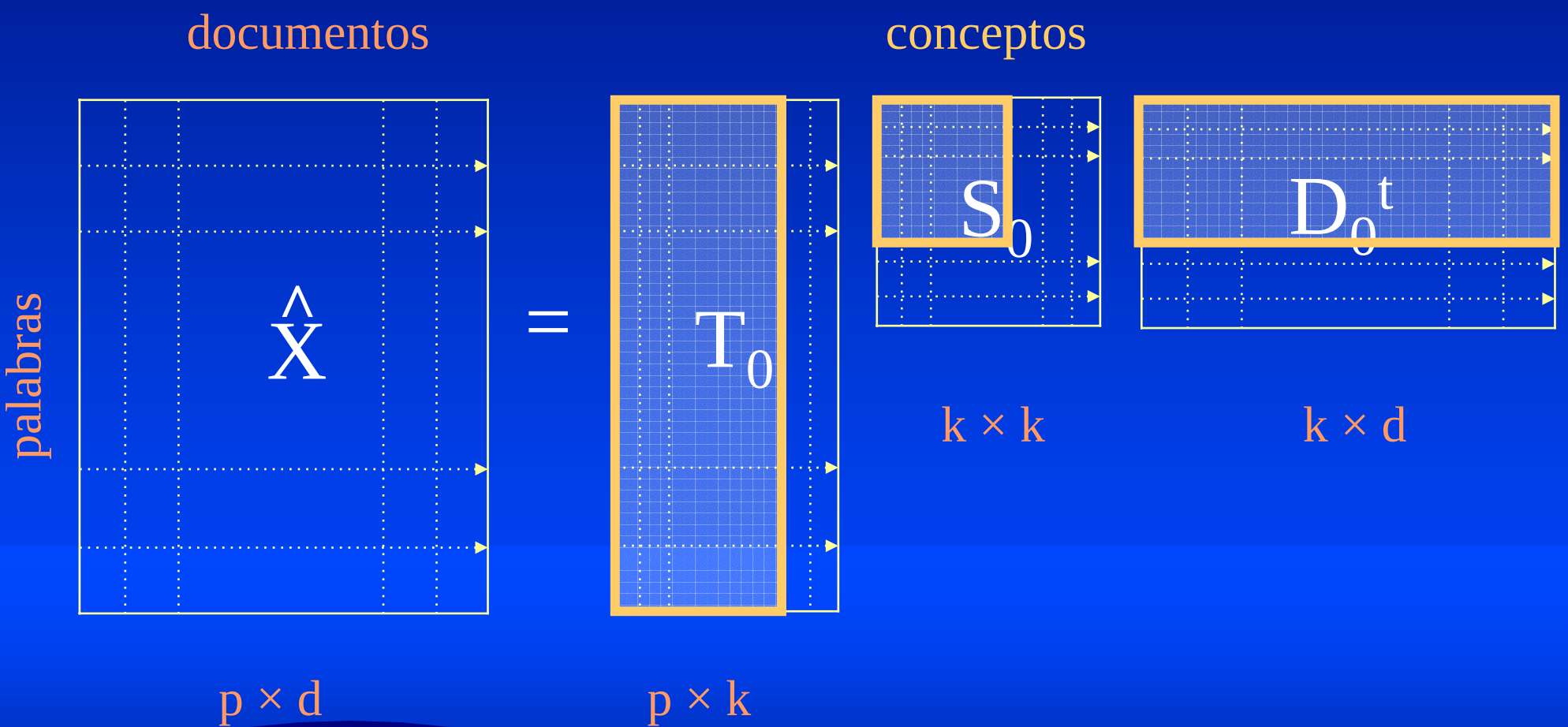
$D_0 =$

0,159	-0,038	0,036	-0,499	-0,136	0,056	0,168	-0,175	0,531	0,000	0,055	0,236	0,195	0,514	0,034	0,000
0,148	0,044	0,012	-0,075	-0,094	0,031	0,116	-0,123	0,438	0,000	0,481	-0,014	-0,459	-0,546	0,179	0,000
0,058	-0,621	0,064	0,222	-0,335	-0,258	-0,241	0,015	0,018	0,000	0,225	-0,023	0,001	0,138	0,380	0,000
0,312	0,122	0,039	0,022	-0,301	0,281	-0,016	-0,095	-0,373	0,000	0,275	0,438	0,351	-0,185	0,678	0,000
0,005	-0,197	0,249	0,260	-0,006	0,097	0,123	-0,726	0,122	0,000	-0,464	0,061	0,054	-0,207	1,000	0,000
0,025	-0,112	0,513	-0,042	0,509	0,212	0,006	-0,209	-0,262	0,000	0,438	-0,132	-0,135	0,281	1,129	0,000
0,009	-0,373	0,386	0,243	-0,183	0,148	0,328	0,463	0,137	0,000	-0,049	0,057	0,036	-0,012	1,628	0,000
0,368	0,183	0,366	-0,005	0,169	-0,565	-0,099	0,099	-0,028	0,000	-0,074	0,402	0,057	-0,127	2,757	0,000
0,039	0,039	0,253	-0,014	0,225	0,116	-0,627	0,181	0,412	0,000	-0,120	-0,142	0,256	-0,172	3,047	0,000
0,314	0,128	0,094	0,010	-0,104	-0,227	0,145	-0,004	-0,059	-0,707	-0,131	-0,273	-0,190	0,136	3,539	0,000
0,404	-0,233	-0,343	0,116	0,346	0,118	0,057	0,061	0,048	0,000	-0,058	0,065	-0,007	0,005	3,626	0,000
0,404	-0,233	-0,343	0,116	0,346	0,118	0,057	0,061	0,048	0,000	-0,058	0,065	-0,007	0,005	5,862	0,000
0,330	0,147	0,100	0,025	-0,331	0,476	-0,383	0,042	-0,069	0,000	-0,205	-0,039	-0,388	0,184	7,608	0,000
0,314	0,128	0,094	0,010	-0,104	-0,227	0,145	-0,004	-0,059	0,707	-0,131	-0,273	-0,190	0,136	20,534	0,000
0,280	0,087	0,016	0,009	-0,123	0,033	0,128	-0,062	0,018	0,000	0,172	-0,612	0,557	-0,141	0,000	0,000
0,014	-0,167	0,214	-0,509	0,069	0,214	0,273	0,269	-0,154	0,000	-0,309	-0,031	-0,022	-0,315	0,000	0,000
0,063	-0,415	-0,108	-0,530	-0,083	-0,192	-0,296	-0,179	-0,272	0,000	-0,035	-0,110	-0,057	-0,164	0,000	0,000

Un ejemplo de juguete: SVD

- Reducción de las dimensiones de X
- $\hat{X} = T\ S\ D^t$
  - Se reordena S_0 de mayor a menor
  - Se seleccionan los k primeros términos
  - Se reduce la dimensión eliminando el resto de términos (ruido)
  - La elección de k es clave: eliminar ruido pero no perder demasiada información

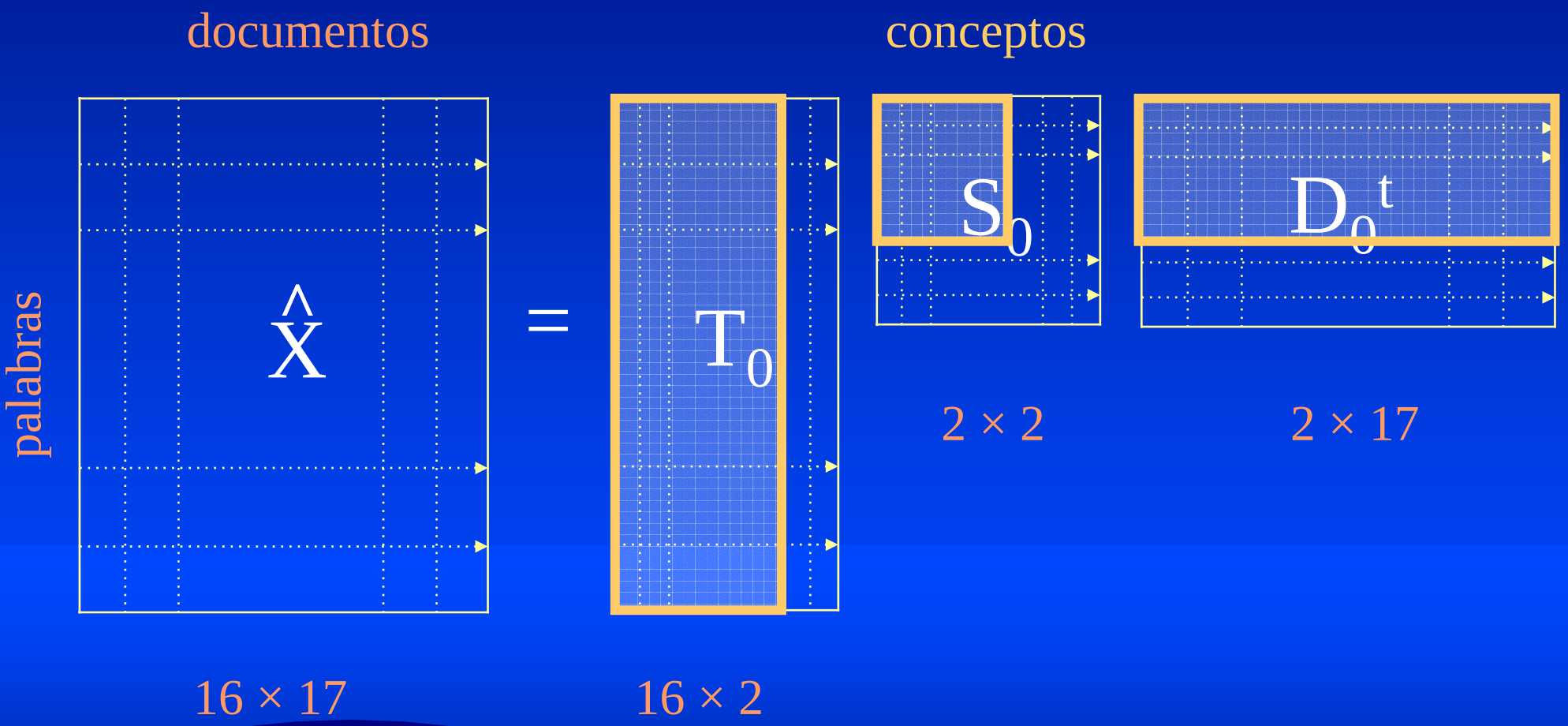
Un ejemplo de juguete: SVD



Un ejemplo de juguete: SVD

- $\hat{X} = T\ S\ D^t$
  - $\hat{X}$ = aproximación X_0
  - Para nuestro ejemplo k = 2
    - Expresarlo en un plano
  - Utilizando la matriz "truncada" generada por SVD, la estructura "latente" subyacente se represente en el espacio dimensional k-reducido.
  - El "ruido" del uso de las palabras se ha eliminado

Un ejemplo de juguete: SVD



Un ejemplo de juguete: SVD

$T_0^t =$

0,016	0,027	0,178	0,601	0,669	0,015	0,052	0,007	0,150	0,081	0,150	0,178	0,141	0,011	0,095	0,205
-0,432	-0,376	-0,169	0,119	0,121	-0,360	-0,225	-0,112	0,113	0,067	0,113	-0,169	0,097	-0,236	0,040	-0,545
0,289	-0,018	-0,284	0,009	0,029	0,186	0,059	0,315	0,190	0,146	0,190	-0,284	0,057	0,460	0,468	-0,302
0,380	-0,162	0,122	0,159	-0,142	0,244	-0,807	0,114	0,003	0,006	0,003	0,122	0,024	-0,162	-0,033	-0,040
-0,279	-0,222	0,368	-0,054	-0,176	-0,275	-0,080	0,267	0,035	-0,057	0,035	0,368	-0,336	0,210	0,480	0,146
-0,007	-0,258	0,136	0,004	0,054	-0,063	0,044	0,177	-0,454	0,339	-0,454	0,136	0,434	0,328	-0,136	-0,122
0,126	-0,324	0,068	0,021	0,192	0,052	0,087	0,078	0,028	-0,608	0,028	0,068	-0,240	0,366	-0,433	-0,256
-0,194	-0,128	0,095	0,077	-0,157	0,375	-0,067	-0,732	0,075	0,175	0,075	0,095	-0,042	0,410	0,056	-0,033
0,261	-0,239	0,090	-0,446	0,465	0,146	0,099	-0,131	-0,082	0,323	-0,082	0,090	-0,416	-0,262	0,115	-0,149
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,707	0,000	-0,707	0,000	0,000	0,000	0,000	0,000
-0,351	0,231	-0,140	-0,255	0,396	0,213	-0,351	-0,032	-0,249	-0,395	-0,249	-0,140	0,085	0,096	0,296	0,091
0,153	-0,216	0,211	-0,368	-0,008	0,054	0,154	-0,116	0,209	-0,294	0,209	0,211	0,647	-0,173	0,207	-0,005
0,214	-0,133	-0,035	0,429	-0,194	0,087	0,277	-0,192	-0,316	-0,312	-0,316	-0,035	-0,087	-0,286	0,420	-0,168
-0,440	-0,140	0,056	0,074	-0,100	0,690	0,194	0,402	0,051	0,068	0,051	0,056	-0,008	-0,252	-0,099	-0,084
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000

Un ejemplo de juguete: SVD

$S_0 =$

4,531													
	2,758												
		2,421											
			1,904										
				1,881									
					1,745								
						1,660							
							1,276						
								1,063					
									1,000				
										0,823			
											0,616		
												0,423	
													0,183

Un ejemplo de juguete: SVD

$D_0 =$

0,159	-0,038	0,036	-0,499	-0,136	0,056	0,168	-0,175	0,531	0,000	0,055	0,236	0,195	0,514	0,034	0,000
0,148	0,044	0,012	-0,075	-0,094	0,031	0,116	-0,123	0,438	0,000	0,481	-0,014	-0,459	-0,546	0,179	0,000
0,058	-0,621	0,064	0,222	-0,335	-0,258	-0,241	0,015	0,018	0,000	0,225	-0,023	0,001	0,138	0,380	0,000
0,312	0,122	0,039	0,022	-0,301	0,281	-0,016	-0,095	-0,373	0,000	0,275	0,438	0,351	-0,185	0,678	0,000
0,005	-0,197	0,249	0,260	-0,006	0,097	0,123	-0,726	0,122	0,000	-0,464	0,061	0,054	-0,207	1,000	0,000
0,025	-0,112	0,513	-0,042	0,509	0,212	0,006	-0,209	-0,262	0,000	0,438	-0,132	-0,135	0,281	1,129	0,000
0,009	-0,373	0,386	0,243	-0,183	0,148	0,328	0,463	0,137	0,000	-0,049	0,057	0,036	-0,012	1,628	0,000
0,368	0,183	0,366	-0,005	0,169	-0,565	-0,099	0,099	-0,028	0,000	-0,074	0,402	0,057	-0,127	2,757	0,000
0,039	0,039	0,253	-0,014	0,225	0,116	-0,627	0,181	0,412	0,000	-0,120	-0,142	0,256	-0,172	3,047	0,000
0,314	0,128	0,094	0,010	-0,104	-0,227	0,145	-0,004	-0,059	-0,707	-0,131	-0,273	-0,190	0,136	3,539	0,000
0,404	-0,233	-0,343	0,116	0,346	0,118	0,057	0,061	0,048	0,000	-0,058	0,065	-0,007	0,005	3,626	0,000
0,404	-0,233	-0,343	0,116	0,346	0,118	0,057	0,061	0,048	0,000	-0,058	0,065	-0,007	0,005	5,862	0,000
0,330	0,147	0,100	0,025	-0,331	0,476	-0,383	0,042	-0,069	0,000	-0,205	-0,039	-0,388	0,184	7,608	0,000
0,314	0,128	0,094	0,010	-0,104	-0,227	0,145	-0,004	-0,059	0,707	-0,131	-0,273	-0,190	0,136	20,534	0,000
0,280	0,087	0,016	0,009	-0,123	0,033	0,128	-0,062	0,018	0,000	0,172	-0,612	0,557	-0,141	0,000	0,000
0,014	-0,167	0,214	-0,509	0,069	0,214	0,273	0,269	-0,154	0,000	-0,309	-0,031	-0,022	-0,315	0,000	0,000
0,063	-0,415	-0,108	-0,530	-0,083	-0,192	-0,296	-0,179	-0,272	0,000	-0,035	-0,110	-0,057	-0,164	0,000	0,000

Un ejemplo de juguete: SVD

T

algorithms	0,016	-0,432
application	0,027	-0,376
delay	0,178	-0,169
differential	0,601	0,119
equations	0,669	0,121
implementation	0,015	-0,360
integral	0,052	-0,225
introduction	0,007	-0,112
methods	0,150	0,113
nonlinear	0,081	0,067
ordinary	0,150	0,113
oscillation	0,178	-0,169
partial	0,141	0,097
problem	0,011	-0,236
systems	0,095	0,040
theory	0,205	-0,545

S

4,531	
	2,758

D

0,159	-0,038
0,148	0,044
0,058	-0,621
0,312	0,122
0,005	-0,197
0,025	-0,112
0,009	-0,373
0,368	0,183
0,039	0,039
0,314	0,128
0,404	-0,233
0,404	-0,233
0,330	0,147
0,314	0,128
0,280	0,087
0,014	-0,167
0,063	-0,415

Un ejemplo de juguete: SVD

$$\hat{X} = T\ S\ D^t$$

algorithms	0,056	-0,042	0,743	-0,123	0,235	0,135	0,445	-0,191	-0,043	-0,130	0,307	0,307	-0,151	-0,130	-0,083	0,200	0,499
application	0,058	-0,028	0,650	-0,089	0,205	0,119	0,387	-0,145	-0,036	-0,095	0,290	0,290	-0,112	-0,095	-0,056	0,175	0,438
delay	0,146	0,099	0,337	0,195	0,096	0,072	0,181	0,212	0,013	0,194	0,436	0,436	0,198	0,194	0,186	0,089	0,244
differential	0,421	0,417	-0,045	0,889	-0,051	0,031	-0,097	1,062	0,119	0,896	1,026	1,026	0,946	0,896	0,792	-0,017	0,035
equations	0,470	0,462	-0,032	0,985	-0,051	0,038	-0,097	1,176	0,131	0,993	1,148	1,148	1,048	0,993	0,879	-0,014	0,051
implementation	0,048	-0,034	0,621	-0,101	0,196	0,113	0,371	-0,157	-0,036	-0,106	0,259	0,259	-0,124	-0,106	-0,068	0,167	0,417
integral	0,061	0,008	0,399	-0,002	0,123	0,075	0,233	-0,027	-0,015	-0,005	0,240	0,240	-0,013	-0,005	0,012	0,107	0,272
introduction	0,016	-0,009	0,194	-0,028	0,061	0,035	0,115	-0,046	-0,011	-0,030	0,084	0,084	-0,035	-0,030	-0,019	0,052	0,130
methods	0,097	0,114	-0,153	0,250	-0,058	-0,018	-0,110	0,307	0,039	0,253	0,203	0,203	0,270	0,253	0,218	-0,043	-0,086
nonlinear	0,052	0,063	-0,094	0,137	-0,035	-0,012	-0,066	0,169	0,022	0,139	0,106	0,106	0,149	0,139	0,119	-0,026	-0,054
ordinary	0,097	0,114	-0,153	0,250	-0,058	-0,018	-0,110	0,307	0,039	0,253	0,203	0,203	0,270	0,253	0,218	-0,043	-0,086
oscillation	0,146	0,099	0,337	0,195	0,096	0,072	0,181	0,212	0,013	0,194	0,436	0,436	0,198	0,194	0,186	0,089	0,244
partial	0,092	0,106	-0,130	0,233	-0,050	-0,014	-0,094	0,285	0,035	0,235	0,197	0,197	0,251	0,235	0,203	-0,036	-0,071
problem	0,032	-0,022	0,407	-0,065	0,129	0,074	0,243	-0,102	-0,023	-0,068	0,171	0,171	-0,080	-0,068	-0,043	0,110	0,274
systems	0,065	0,069	-0,043	0,148	-0,020	-0,002	-0,037	0,179	0,021	0,149	0,149	0,149	0,158	0,149	0,131	-0,012	-0,019
theory	0,204	0,071	0,987	0,106	0,301	0,191	0,569	0,067	-0,022	0,099	0,726	0,726	0,086	0,099	0,130	0,264	0,682

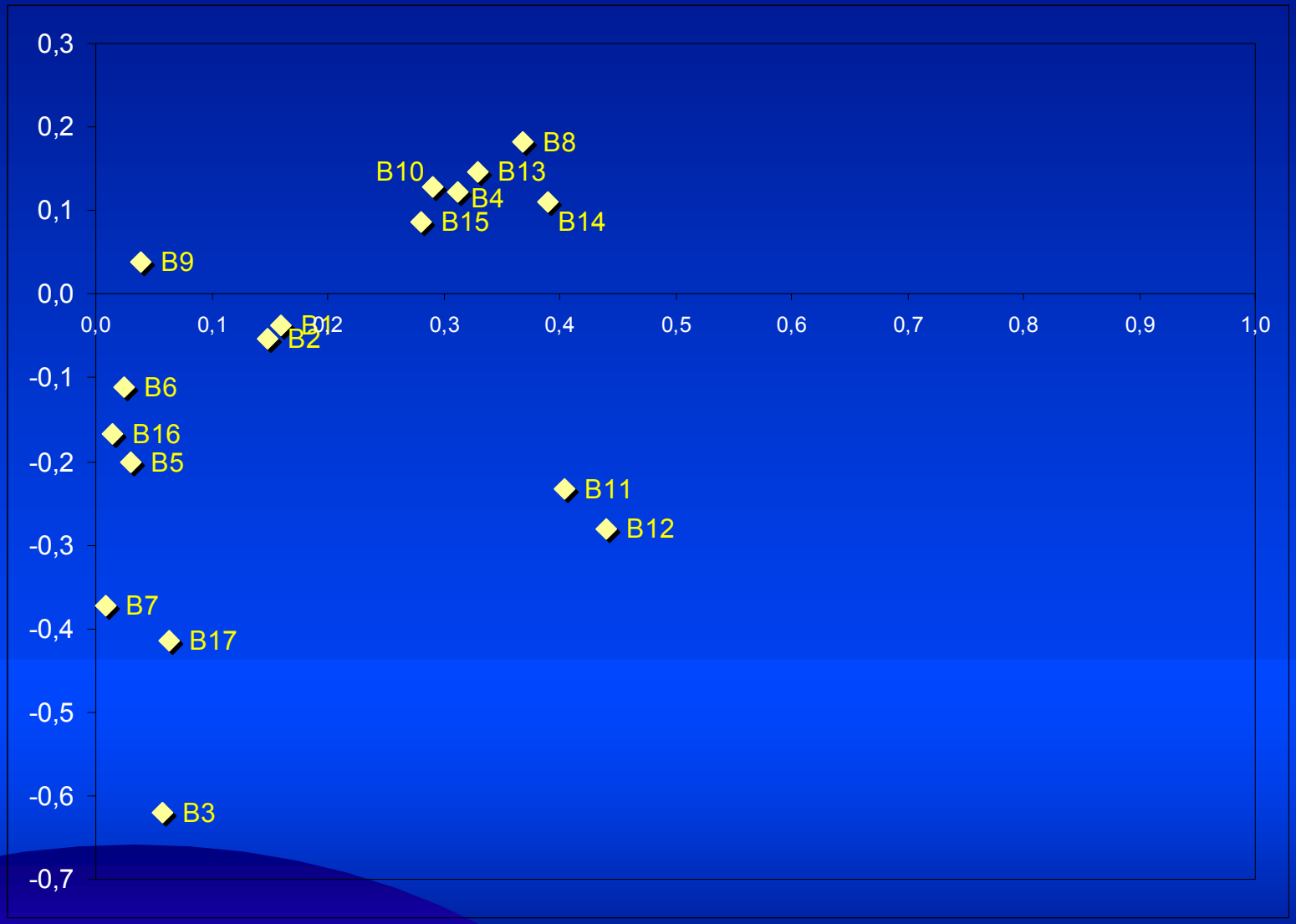
Recuperación de Información

- Consulta:
  - k-“vector” Q de palabras
  - Se “ubica” en el espacio k-dimensional

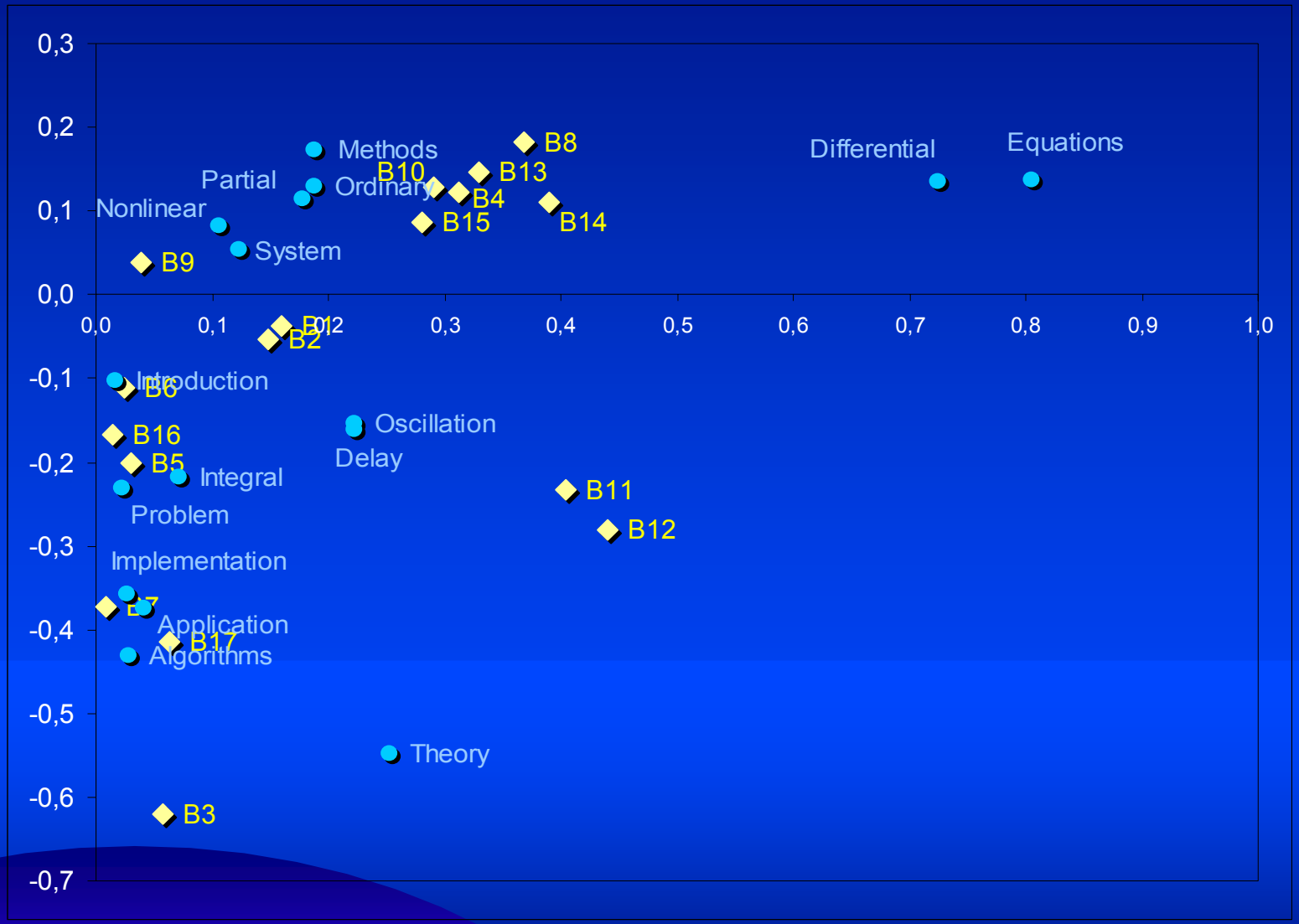
$$Q = Q^t T S^{-1}$$

  - $Q^t T$ = consulta “mapeada” al espacio de palabras
  - S^{-1} aporta los “pesos” de cada dimensión
  - Luego Q se compara con el espacio de todos los vectores de documentos (similaridad)
  - Medida: Coseno entre los ángulos de d_i y Q

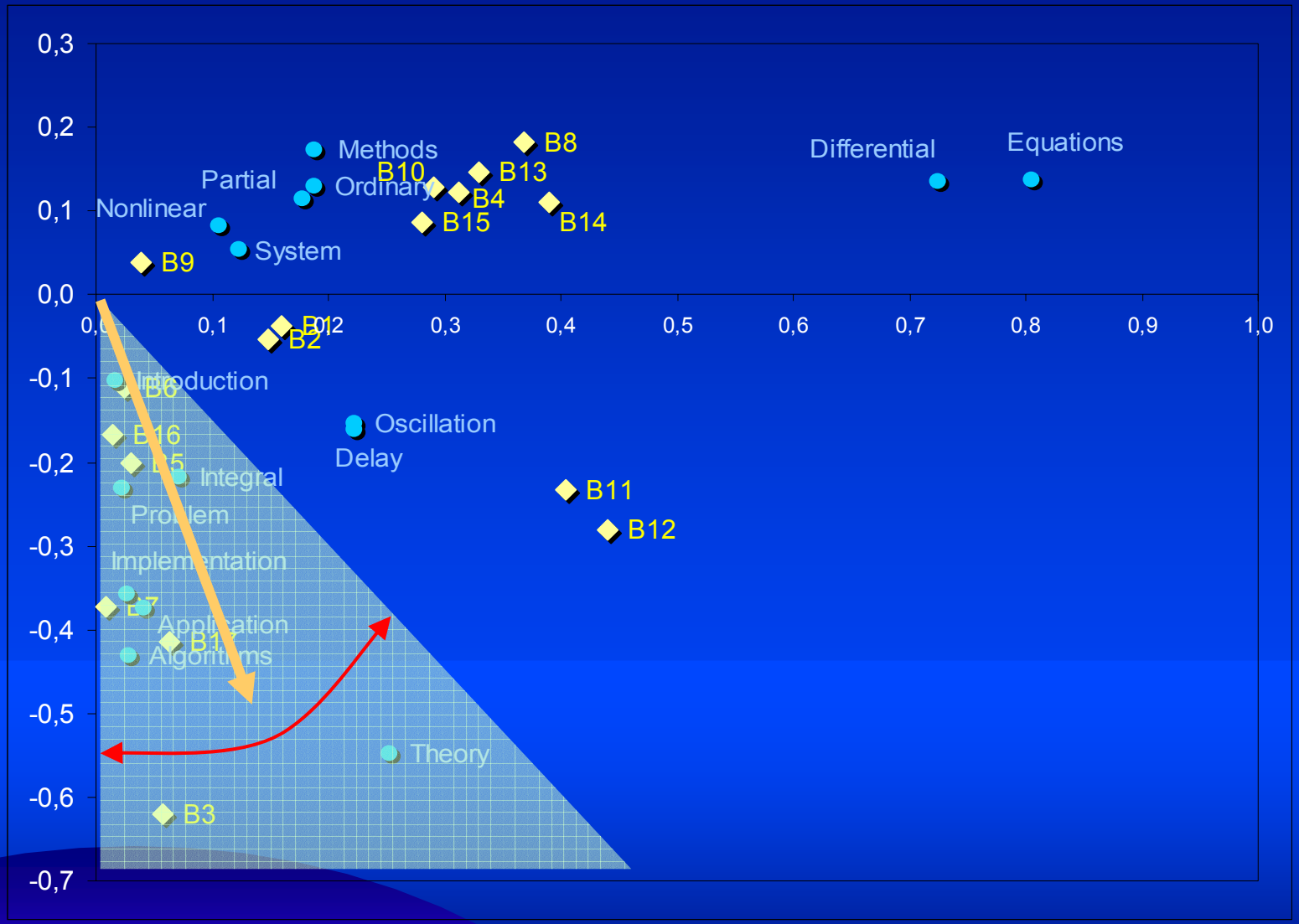
Mapeo de Documentos



Mapeo de Documentos



Mapeo de Documentos



Consulta

- Consulta = “application and theory”
  - AND : palabra no conceptual
  - Consulta = “application theory”

algorithms	application	delay	differential	equations	implementation	integral	introduction	methods	nonlinear	ordinary	oscillation	partial	problem	systems	theory
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Consulta: Comparación

Número de factores de la matriz S					
K = 2		K = 4		K = 8	
B17	0.99	B17	0.87	B17	0.88
B3	0.99	B3	0.82	B3	0.78
B6	0.99	B12	0.57	B12	0.37
B16	0.99	B11	0.57	B11	0.37
B5	0.98	B16	0.38		
B7	0.98	B7	0.38		
B12	0.55	B1	0.35		
B11	0.55	B5	0.22		
B1	0.38				

Matcheo de palabras	
B3	SI
B11	SI
B12	SI
B17	SI

Otras comparaciones

- Documento / documento: (clustering)
  - $C^d = X^t X = D S^2 D^t$
  - $C^d_{i,j}$ = similitud entre documentos d_i y d_j
  - Producto escalar (coseno) entre columnas de X
  - (i.e. entre filas de DS)
- Palabra / palabra:
  - $C^t = X X^t = T S^2 T^t$
  - $C^t_{i,j}$ = similitud entre palabras k_i y k_j
  - Producto escalar (coseno) entre filas de X

Otras comparaciones

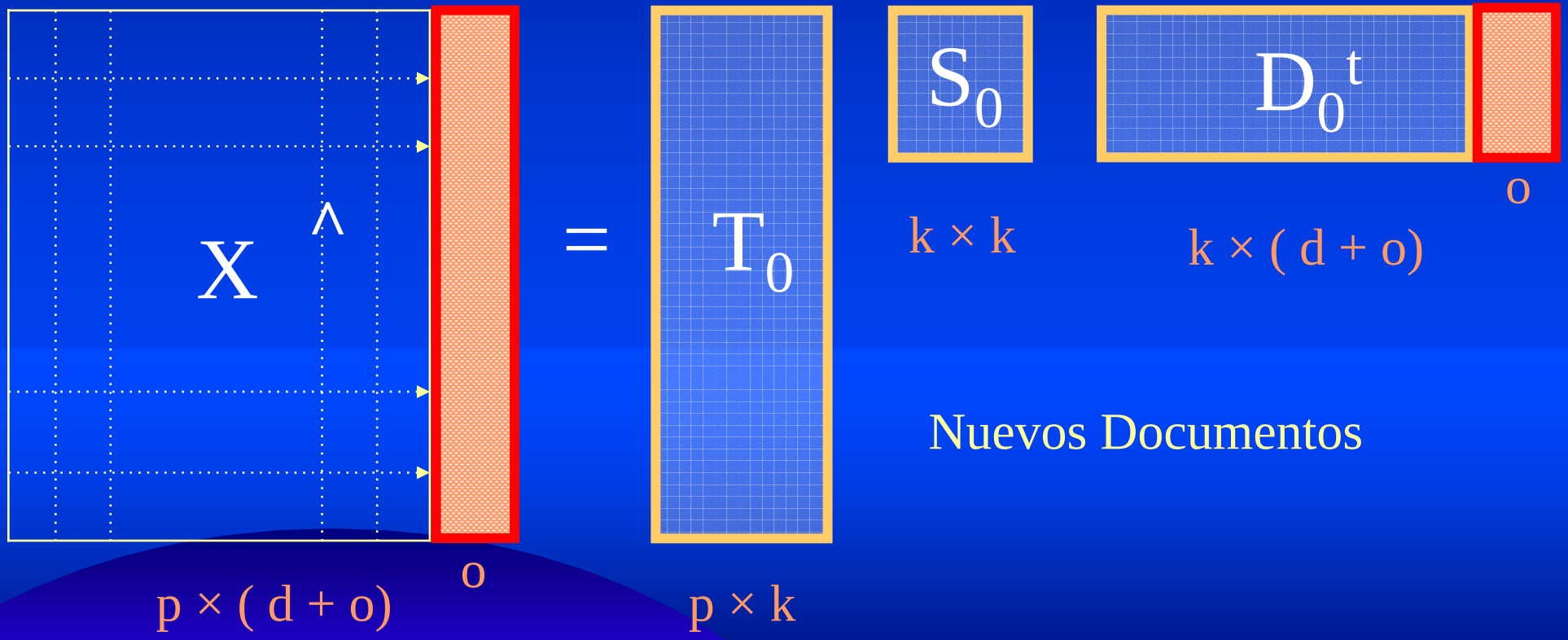
- Término / documento
  - $X_{i,j}$ = similitud entre término k_i y documento d_j
  - Producto escalar entre filas de $TS^{½}$ (términos) y filas de $DS^{½}$ (documentos)
- Consulta / documento
  - q = vector de términos de la query
  - q^t T = vector comparable con documentos (filas de DS) por coseno

Crecimiento del Corpus

- Estrategias:
  - Incorporación de palabras y documentos
  - Recálculo completo por SVD
  - Actualización de SVD
- Depende
  - Cantidad de documentos / palabras
  - Naturaleza de los nuevos documentos

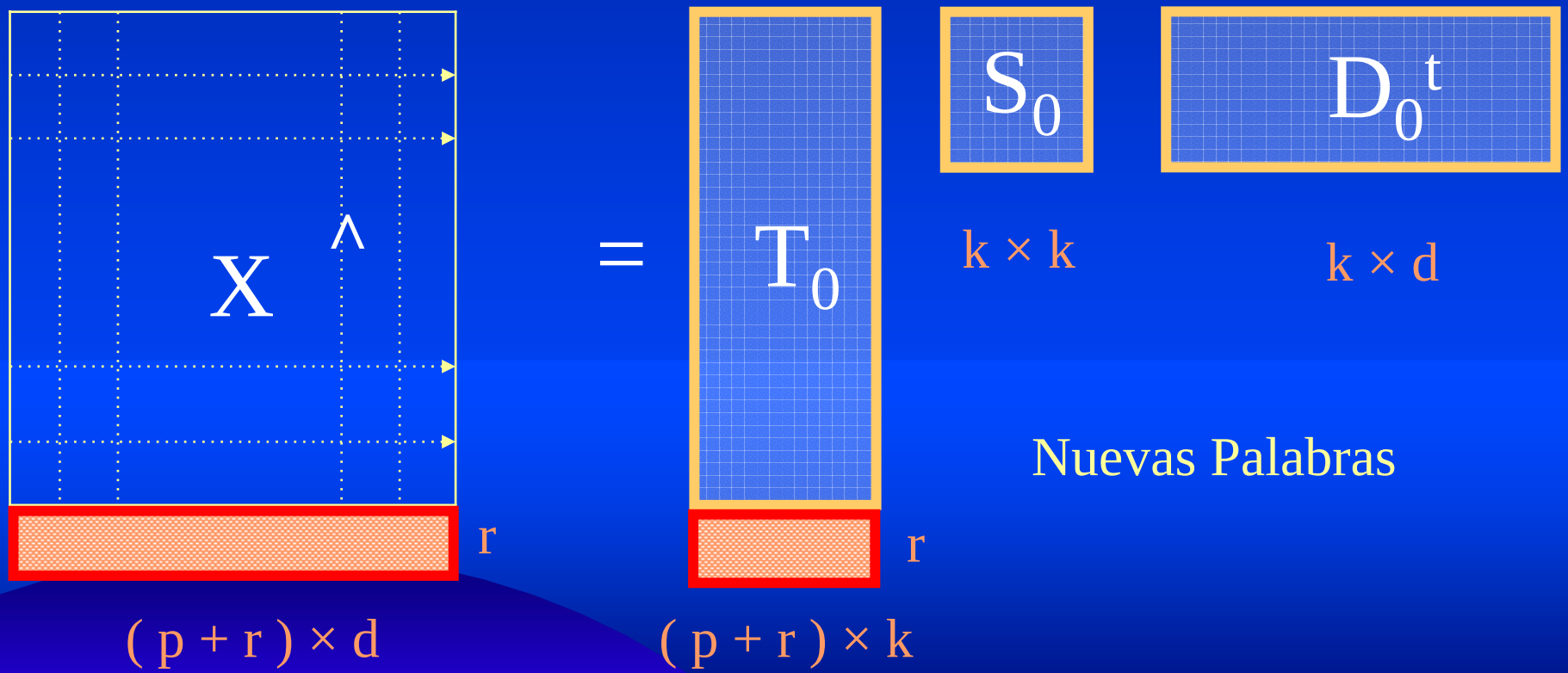
Crecimiento del Corpus

- Incorporación de palabras y documentos
  - $\boldsymbol{d'=d^T U_k S^{-1}_k}$ similar a la proyección de una consulta



Crecimiento del Corpus

- Incorporación de palabras y documentos
  - $\boldsymbol{d'=d^T U_k S^{-1}_k}$ similar a la proyección de una consulta



Crecimiento del Corpus

- Actualización de SVD
  - Aprovecha los anteriores autovectores y autovalores calculados
  - “Pega” la nueva matriz (o × d) a la matriz X
  - Continua iterando para “corregir” lo calculado (pesos de la matriz original)
  - Recupera semántica añadida

Crecimiento del Corpus

- Recálculo completo por SVD
  - Afecta la naturaleza semántica de las matrices
  - Redefine la estructura latente subyacente
  - Requiere más costo computacional
    - ¿cuánto más costoso?

Costo de Cálculo de SVD

- ***Dumais 1995**: "SVD toma solamente 2 minutos en un Sparc10 para una matriz de 2.000 x 5.000, pero el tiempo crece a entre 18 y 20 horas para matrices de 60.000 x 80.000"*
- ***Hong 2000**: "El Algoritmo SVD es O(N2 k3), con N número de palabras + documentos, y k el número de dimensiones en el espacio conceptual". "Sin embargo, si la colección es estable, SVD se calcula sólo una vez, lo que significa un costo aceptable"*
- ***Leif**: Si hoy tenemos computadoras 100 veces más rápidas que Dumais en 1995, conjuntos de datos 20 veces más grandes y funciones SVD optimizadas (en vez de prototipos de investigación), debería tomar alrededor de unas 20 horas*

Limitaciones

- LSI es una técnica “bolsa de palabras”
- No considera ordenamiento de palabras, postagging, sintaxis
- Algunas consideraciones futuras
  - ¿Añadir información sintáctica a LSA?
  - Integrar sintaxis, semántica LSA, análisis contextual

Moraleja

Si está planeando utilizar LSI, úselo para aquello que realmente sirve…

Algunas Áreas de aplicación

- Comprensión de Lenguaje Natural
  - Evaluación automática de respuesta de estudiantes
- Ciencia Cognitiva
  - Representación y adquisición del conocimiento
  - Test sinonimia (TOEFL)
- Reconocimiento y comprensión de la lengua hablada
  - Clasificación Semántica
  - Modelización semántica

Del sitio oficial Google…

http://www.google.com.pr/intl/es/management.html

- Craig Silverstein, Director de tecnología

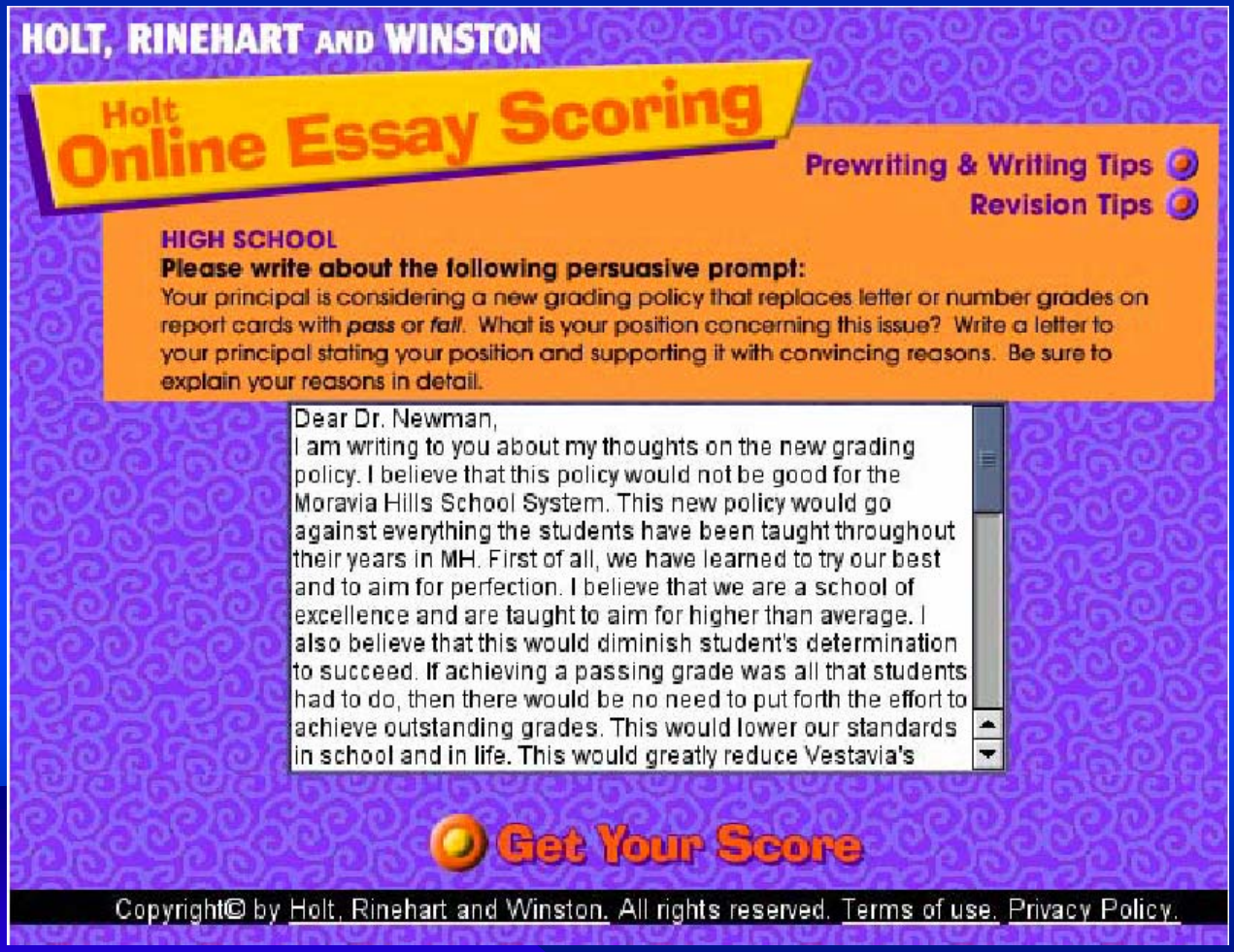
Craig Silverstein fue el primer empleado contratado por los fundadores de Google y creó muchos de los componentes de IT originales que apoyaron el desarrollo y crecimiento de Google. Craig Silverstein está actualmente con licencia de la Universidad de Standford, donde cursa un doctorado en Ciencias de la Computación, enfocado a la recopilación de información y data mining. Silverstein otorgó a Google sus conocimientos en algoritmos de compresión, mientras todavía era un proyecto de investigación en Stanford. Sus otros intereses académicos incluyen versiones muy eficientes de estructuras de datos básicas, como las tablas hash, así como el clustering eficiente de grandes volúmenes de datos usando la Distribución/Recopilación y el indexado de semántica latente cuando se relaciona con clustering, temas que exploró en el Laboratorio Xerox PARC.

Aplicaciones concretas

- Generación automática de tesauros de dominios específicos
- Extracción de voces claves de corpus y documentos
- Búsqueda de documentos similares
- Hallazgo de documentos relacionados con otros documentos, palabras, etc.
- Recuperación de información en otros idiomas

Aplicaciones concretas

- Control de ensayos y escritos, con devolución sustantiva



Aplicaciones concretas

- Detección de plagios en obligaciones estudiantiles

An example of plagiarism

MAINFRAMES

Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high demand by its users (clients). Examples of such organizations and enterprises using mainframes are online shopping websites such as Ebay, Amazon, and computing-giant

MAINFRAMES

Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand by its users (clients). Examples of these include the large online shopping websites -i.e. : Ebay, Amazon, Microsoft, etc.

Dudas

- Espacio para dudas