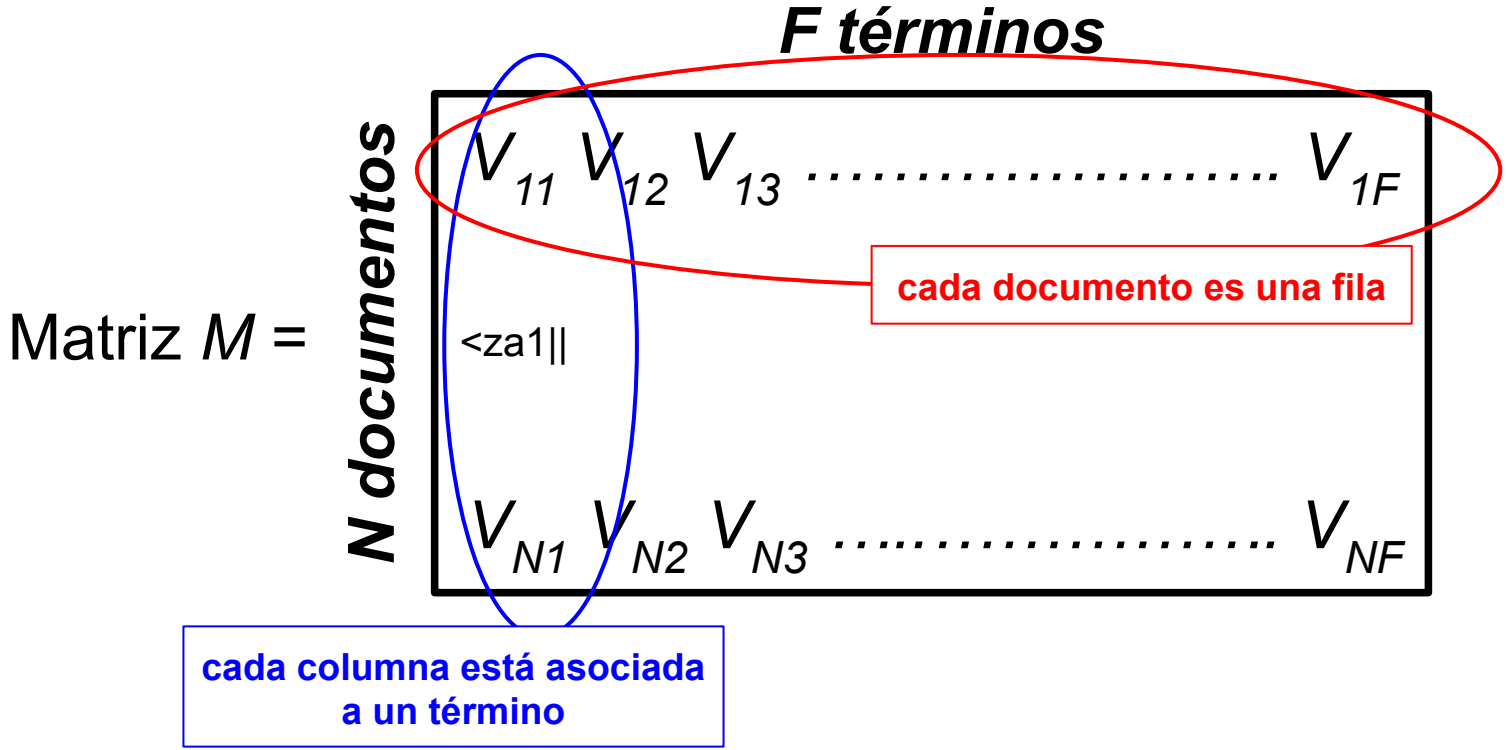
Reducción dimensional en textos y descomposición en tópicos

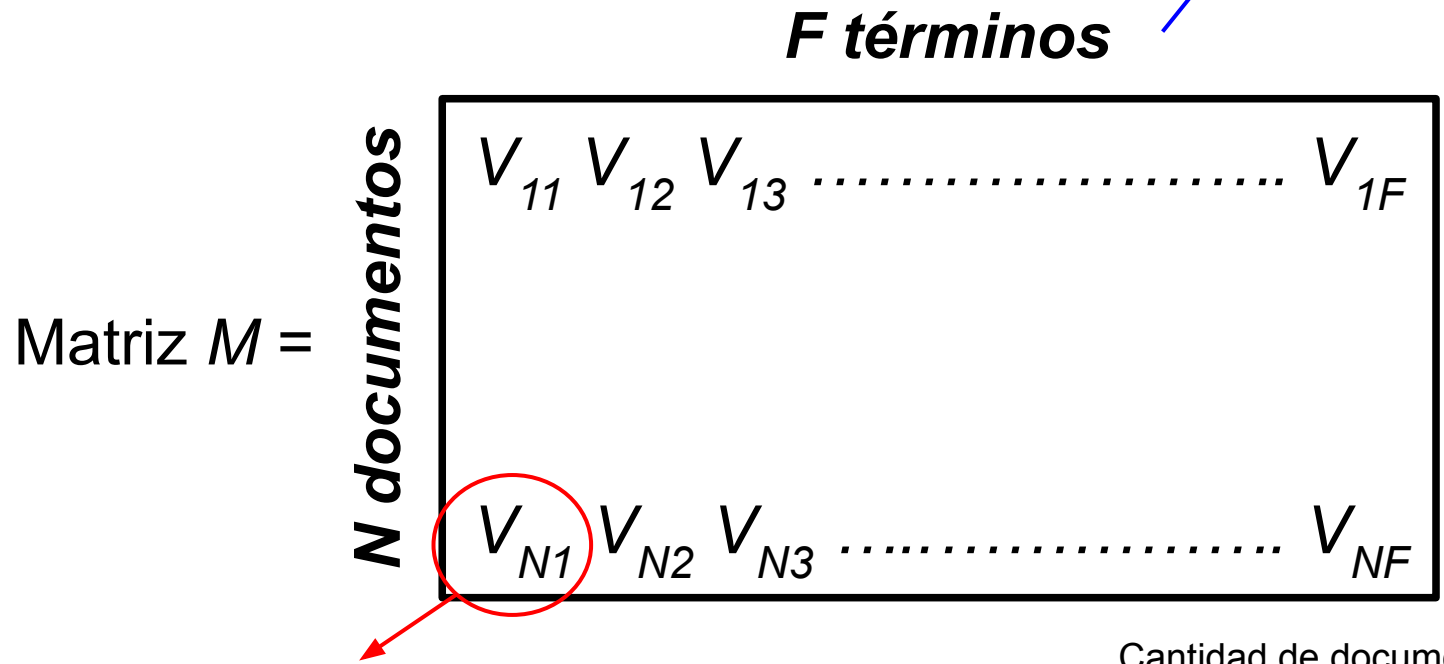
Laboratorio de Datos - 1°C 2021

(Repaso) Representación de los textos



(Repaso) Representación de los textos

Palabras, bigramas, trigramas, lemas, solo la raíz de la palabra...



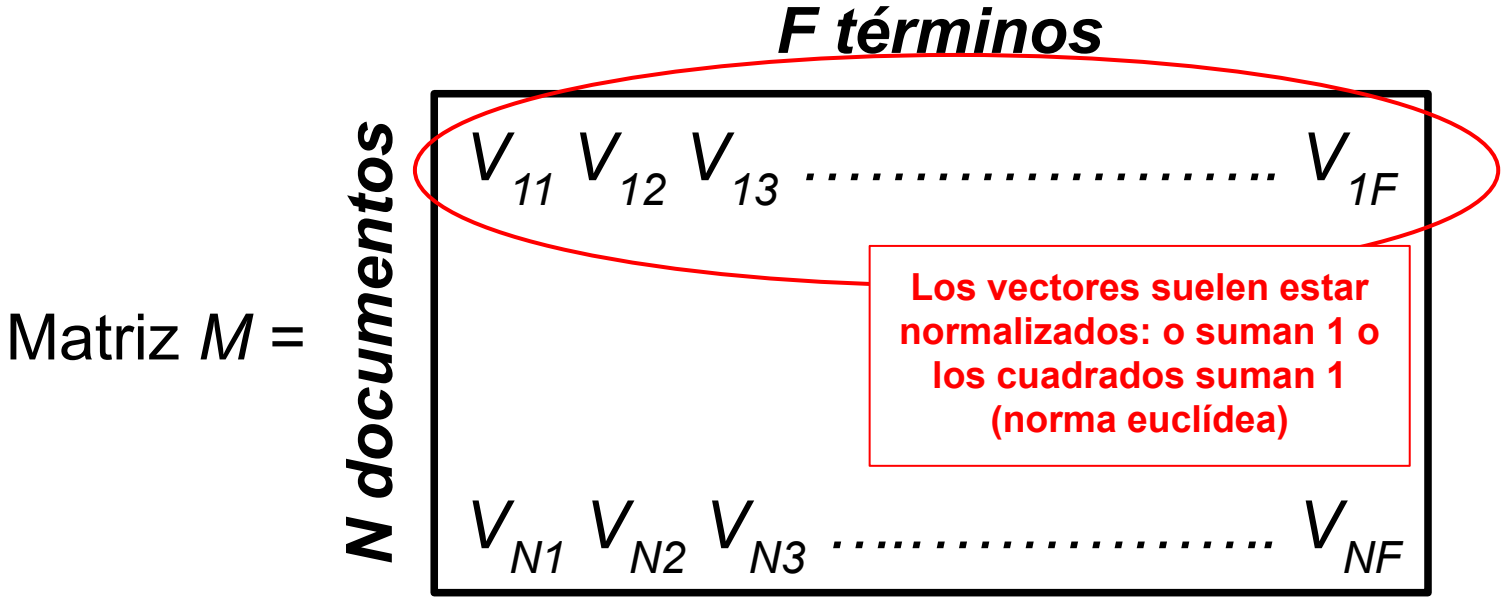
Frecuencia del término (tf) o frecuencia x especificidad (idf)

$$idf_t = \log\left(\frac{N}{n_t}\right)$$

Cantidad de documentos en el corpus

Cantidad de documentos donde aparece el término *t*

(Repaso) Representación de los textos



Clase de hoy

- **Reducción dimensional** con Latent Semantic Analysis (LSA): descripción de los textos en un espacio de combinaciones lineales de los vectores de los términos. Relación con PCA.

- **Descomposición en tópicos**: detección de grupos de textos con una temática similar. Algoritmos basados en descomposición de matrices (NMF: Non-Negative Factorization) y modelos probabilísticos (LDA: Latent Dirichlet Allocation).

Clase de hoy

- **Reducción dimensional** con Latent Semantic Analysis (LSA): descripción de los textos en un espacio de combinaciones lineales de los vectores de los términos. Relación con PCA.

- **Descomposición en tópicos**: detección de grupos de textos con una temática similar. Algoritmos basados en descomposición de matrices (NMF: Non-Negative Factorization) y modelos probabilísticos (LDA: Latent Dirichlet Allocation).

Latent Semantic Analysis

Problema ya planteado en clases anteriores:

la matriz de documentos-términos suele tener muchos ceros, lo cual esconde un poco la relación entre los distintos documentos o términos.

Solución: **reducción dimensional!** (pérdida de información = abstracción)

	Palabra 1	Palabra 2	Palabra 3	Palabra 4	Palabra 5	
Relato 1	0	0.12	0.01	0	0	
Relato 2	0	0	0.44	0.15	0.65	
Relato 3	0.11	0.31	0.28	0	0	(···)
Relato 4	0	0	0.05	0.21	0	
Relato 5	0	0.13	0	0.07	0	
		(···)				

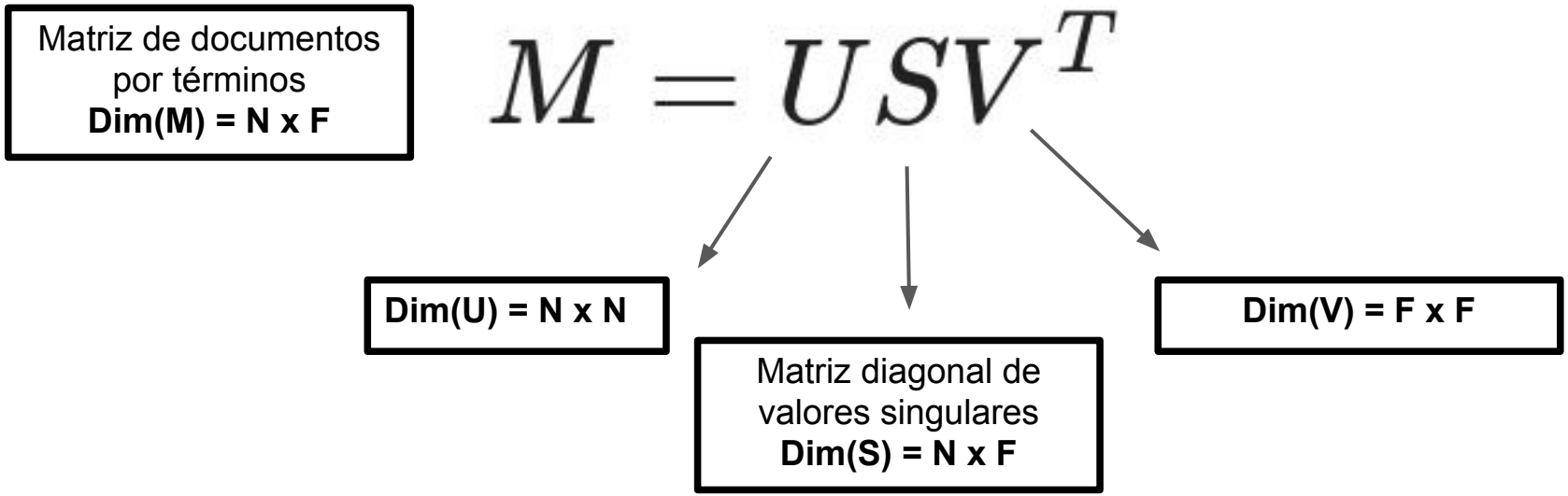
La correlación lineal entre filas nos da una idea de la similitud del significado entre relatos

La correlación lineal entre columnas nos da una idea de la similitud del significado entre palabras

Pero hay un problema: la mayor parte de los valores son 0

Latent Semantic Analysis

El método de LSA consiste en descomponer en valores singulares la matriz M. Los valores singulares son una generalización del concepto de autovalores para matrices no cuadradas:



Ejemplo (del paper original de LSA)

Documentos

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Matriz con frecuencia de términos.

Atención! En la implementación de LSA se suele poner la matriz M traspuesta: los términos como filas y los documentos en las columnas. Es todo lo mismo salvo una trasposición.

r (human.user) = -.38

r (human.minors) = -.29

Correlaciones entre términos

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259-284.

Ejemplo (del paper original de LSA)

U =

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

S =

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

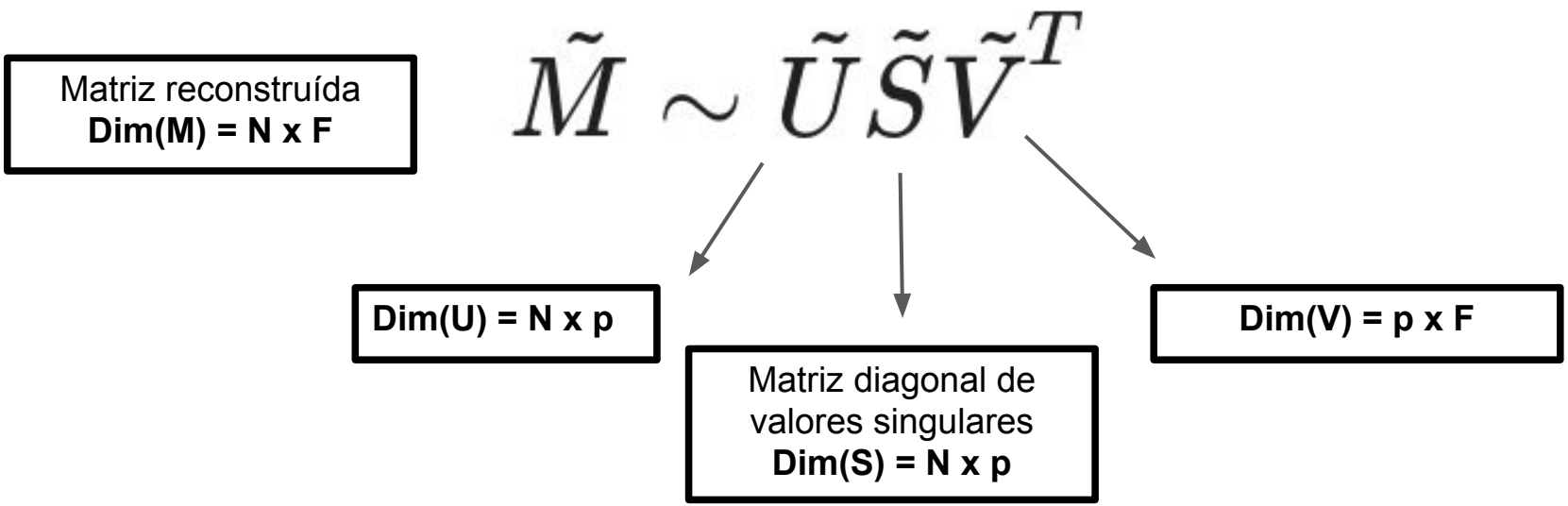
V =

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259-284.

Latent Semantic Analysis

Aproximación: reconstrucción de la matriz quedándonos con unos pocos valores singulares. Recortamos las matrices hasta una dimensión interna *p* (cantidad de valores singulares con los que me quedo).



Ejemplo (del paper original de LSA)

Aproximación: reconstrucción de la matriz quedándonos con unos pocos valores singulares.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), 259-284.

Ejemplo: reconstrucción

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

r (human.user) = .94

r (human.minors) = -.83

→ Correlaciones

Relación con PCA

Las componentes principales son los autovectores de la matriz de covarianza **S**:

$$\bar{\bar{S}}\bar{u} = \lambda\bar{u}$$

Componentes principales

Autovalores: varianza en la dirección de la componente correspondiente.

Donde: $$S_{ij} = \frac{1}{N}\sum_{n}^{N}(x_{ni} - \bar{x_i})(x_{nj} - \bar{x_j})$$

Valor medio del feature j

Valor del feature j la instancia n

Pidiendo además: $$\sum_i u_i^2 = 1$$

Relación con PCA

Hacer SVD es prácticamente igual a PCA:

Las columnas de V tienen los autovectores de $M^T * M$

$$M = USV^T$$

Las columnas de U tienen los autovectores de $M * M^T$

Si los datos están centrados, $M * M^T$ es una matriz de covarianza. Se demuestra que los valores singulares al cuadrado son los autovalores de la matriz de covarianza.

Ventajas de reducir la dimensionalidad con LSA

- Detecta mejor documentos o términos similares (recordar que perder información es en parte abstraer). Lo que está detectando son conjuntos de términos con contextos similares.

- Útil para sistemas de recomendación: por ejemplo, dado un documento, cuáles son los documentos más parecidos.

Observaciones

- ¿Sirve para tópicos? Los signos negativos en la reconstrucción puede resultar un tanto difícil de interpretar...

- Luego de ver word2vec en la materia, LSA puede parecer un poco obsoleto...

Scikit-learn

sklearn.decomposition.TruncatedSVD

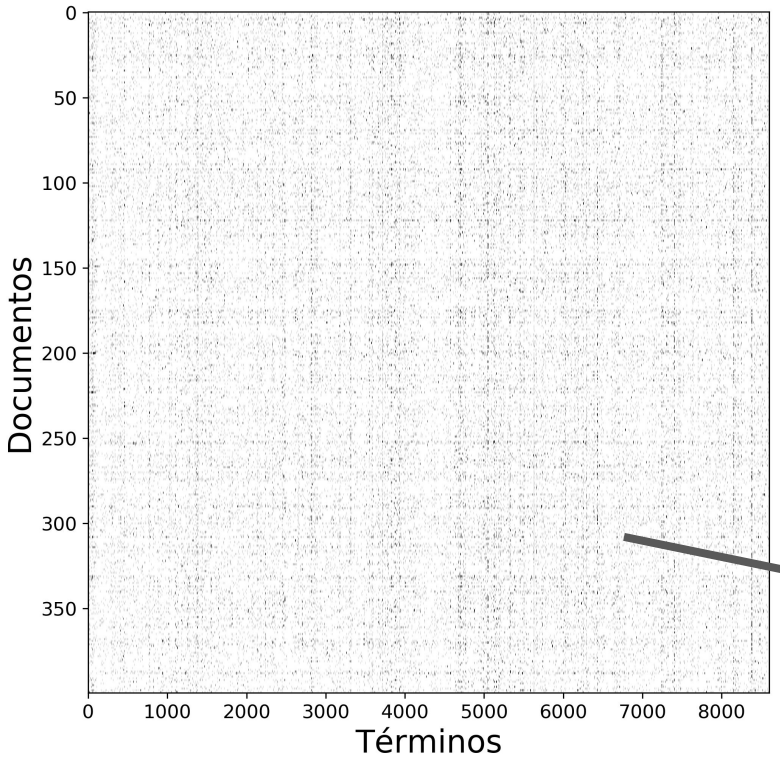
class sklearn.decomposition.**TruncatedSVD**(*n_components=2, *, algorithm='randomized', n_iter=5, random_state=None, tol=0.0*) [source]

Clase de hoy

- **Reducción dimensional** con Latent Semantic Analysis (LSA): descripción de los textos en un espacio de combinaciones lineales de los vectores de los términos. Relación con PCA.

- **Descomposición en tópicos**: detección de grupos de textos con una temática similar. Algoritmos basados en descomposición de matrices (NMF: Non-Negative Factorization) y modelos probabilísticos (LDA: Latent Dirichlet Allocation).

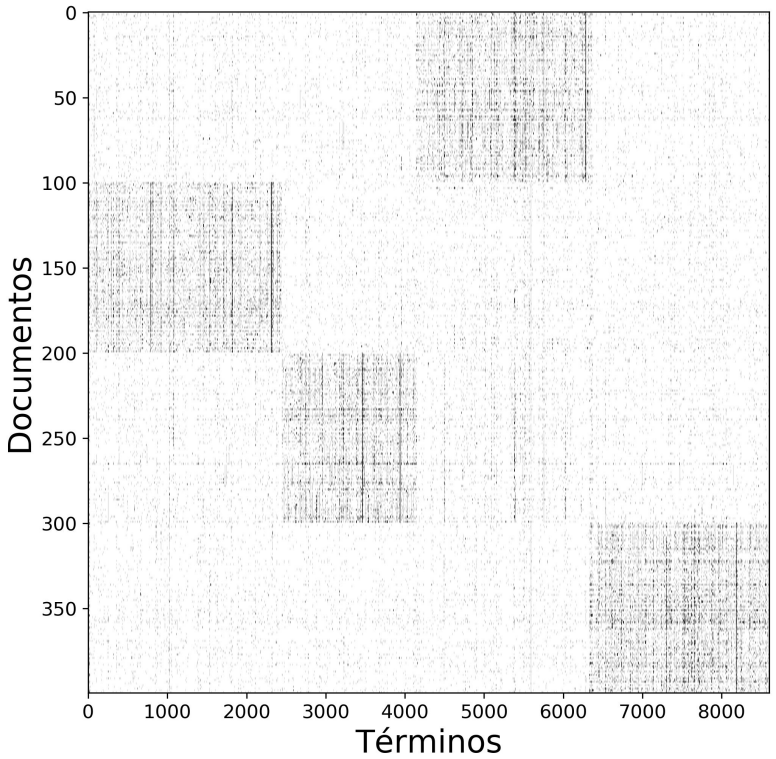
Tópicos



¿Cómo se ve una matriz de documentos por términos real?

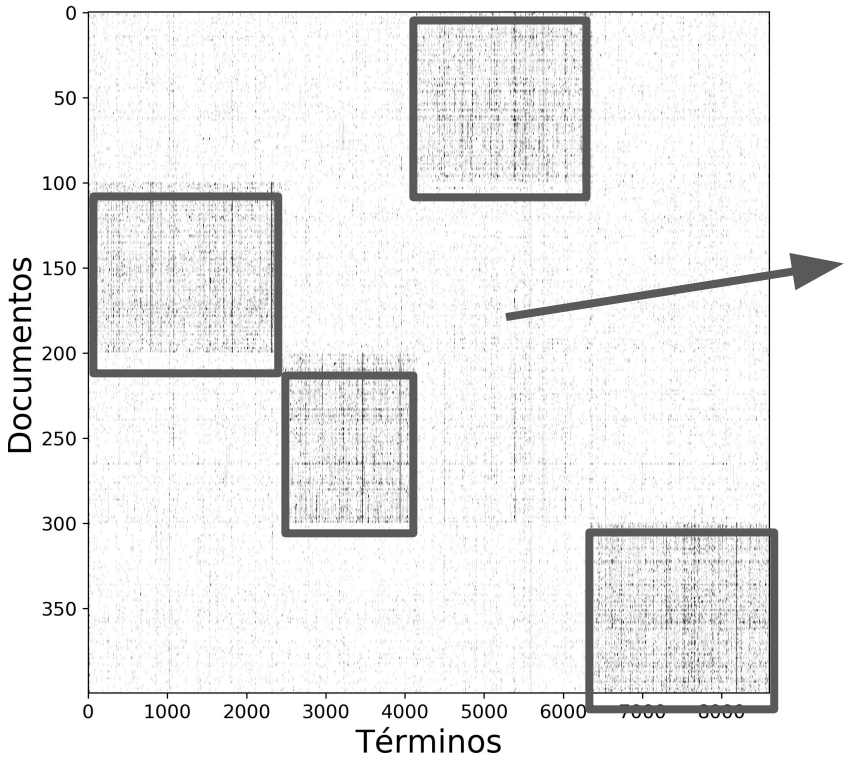
En blanco las componentes igual a cero; en negro las componentes distintas de cero.

Tópicos



Ordenando la matriz, tanto en filas como en columnas...

Tópicos



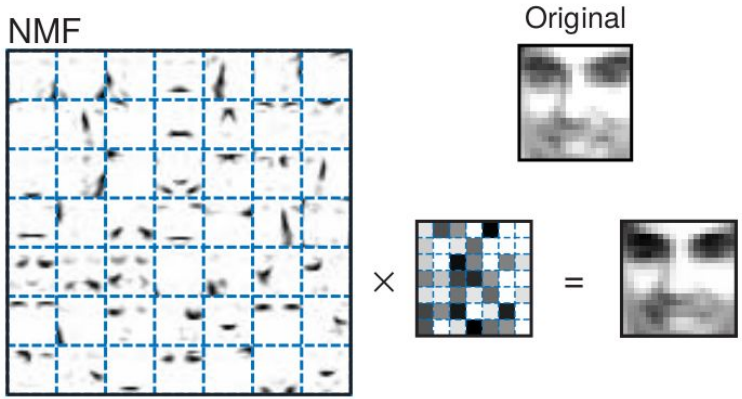
Emergencia de bloques: **Conjunto de documentos que usan términos similares.** Estos bloques **emergen naturalmente** del “ordenamiento” de la matriz de documentos por términos.

A los bloques los identificamos como **tópicos o ejes temáticos.**

¿Cómo “ordenamos”? Con algoritmos de identificación de tópicos (NMF, LDA, etc…)

Non-negative factorization (NMF)

Es una descomposición matricial de los datos, pidiendo la no-negatividad de cada una de las componentes.

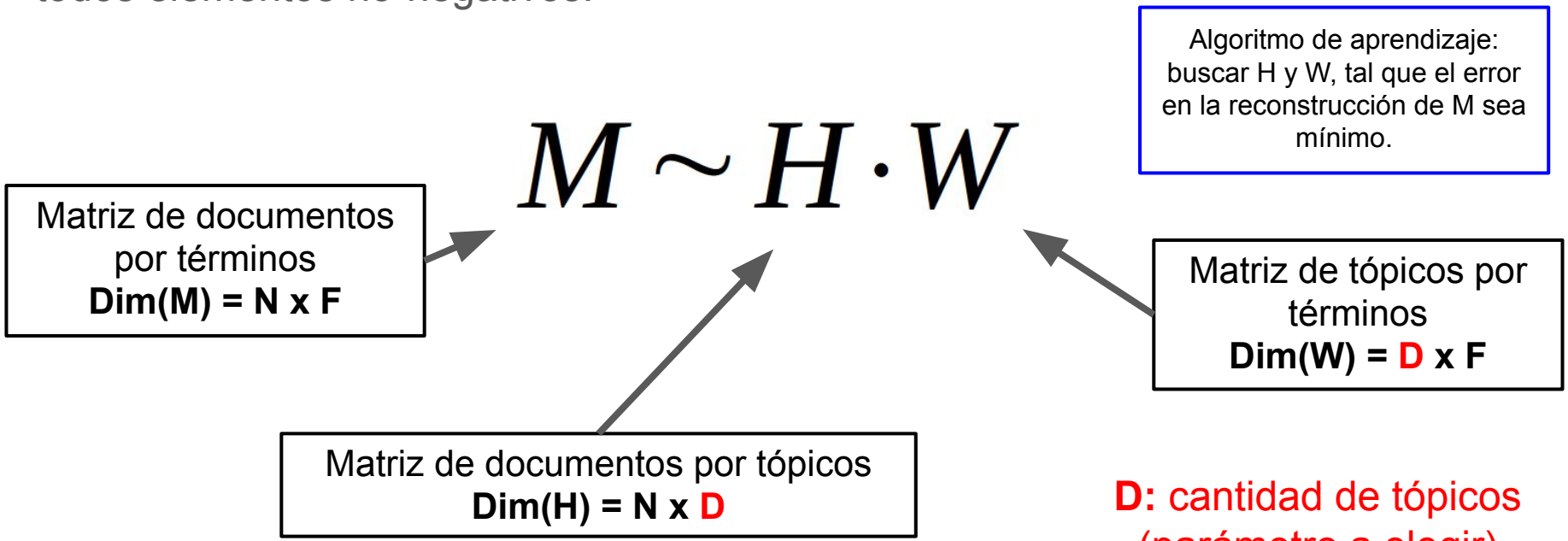


Pedir no negatividad es forzar la descripción de una señal como la suma de otras señales (fuentes).

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788-791.

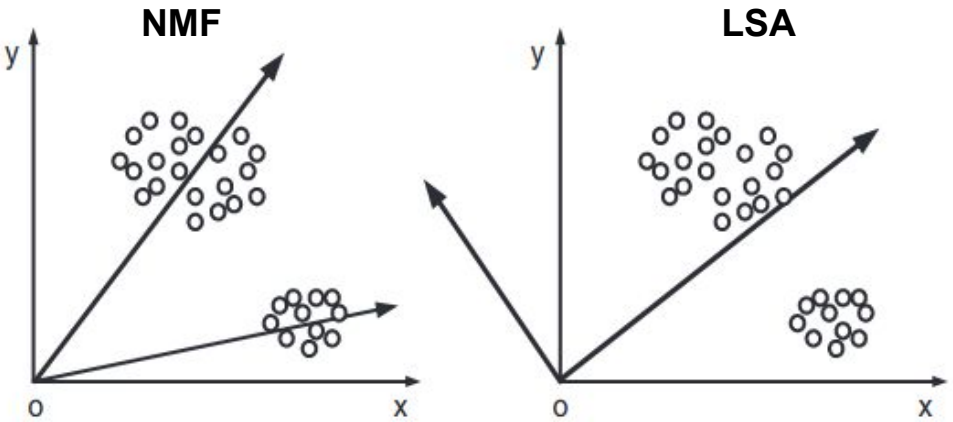
Non-negative factorization (NMF)

Para textos, describimos la matriz *M* como la multiplicación de dos matrices con todos elementos no-negativos.

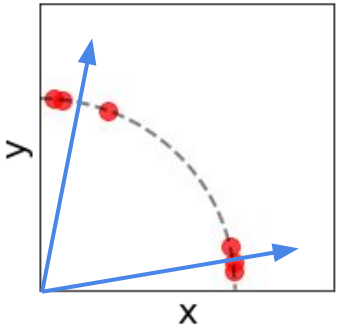


Non-negative factorization (NMF)

La no-negatividad lleva a tópicos no necesariamente ortogonales:



NMF

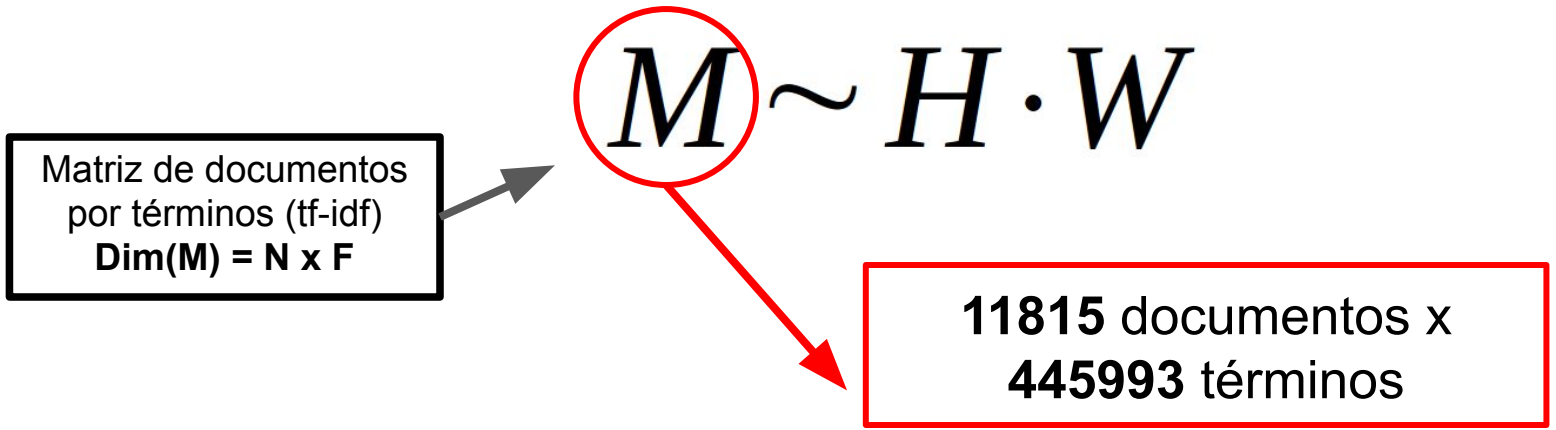


Más aún, si los vectores documento tienen norma euclídea igual a 1, los documentos son puntos en una esfera de radio 1.

Xu, W., Liu, X., & Gong, Y. (2003, July). Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 267-273).

Ejemplo (autobombo)

Notas de las secciones políticas de distintos medios, publicadas en el período del **31 de Julio al 5 de Noviembre del 2017:**



Pinto, S., Albanese, F., Dorso, C. O., & Balenzuela, P. (2019). Quantifying time-dependent Media Agenda and public opinion by topic modeling. *Physica A: Statistical Mechanics and its Applications*, *524*, 614-624.

Interpretación de la matriz W

$$M \sim H \cdot W$$

Matriz de tópicos por términos
Dim(W) = D x F

	Maldonado	Cristina	Kirchner	Santiago	Cambiemos	...
Tópico 1	1.1	**9.6**	**7.2**	1.4	**8.5**	
Tópico 2	**9.4**	0.4	1.2	**8.3**	1.3	

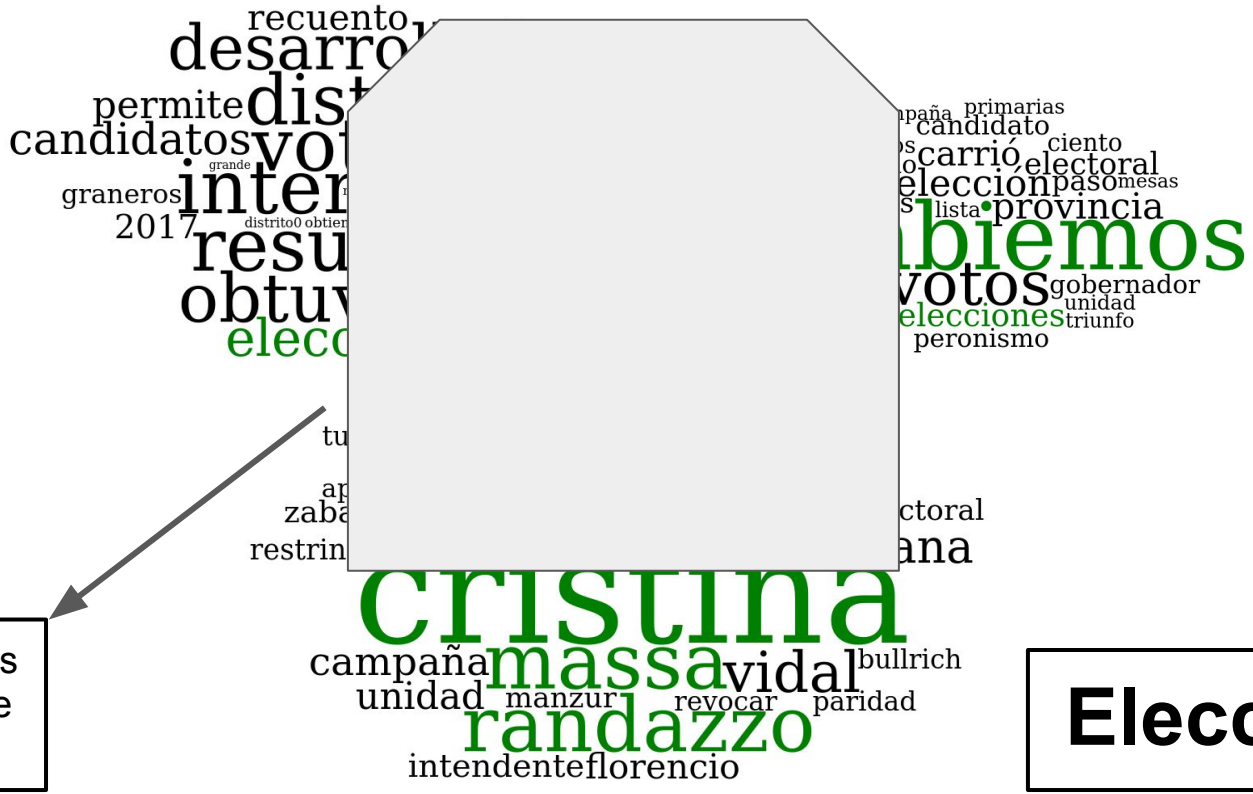
Más todos los tópicos que haya

Más todos los términos que haya

Podemos **ordenar los términos de mayor a menor** peso para cada tópico e **interpretar**.

Los tópicos pueden tener peso distinto de cero en todos los términos.

Tópicos en el corpus de noticias (D = 10)



Wordclouds o nubes de palabras

Elecciones

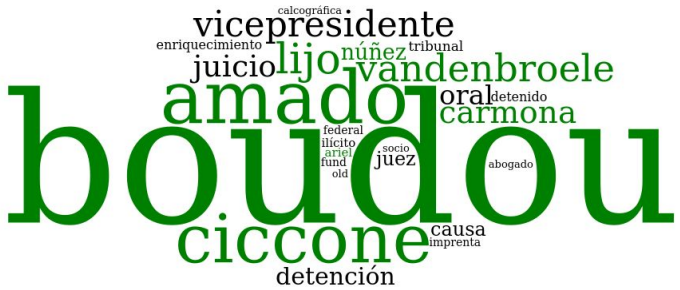
Tópicos en el corpus de noticias (D = 10)

Maldonado

Tópicos en el corpus de noticias (D = 10)

De Vido

Boudou



Milagro Sala

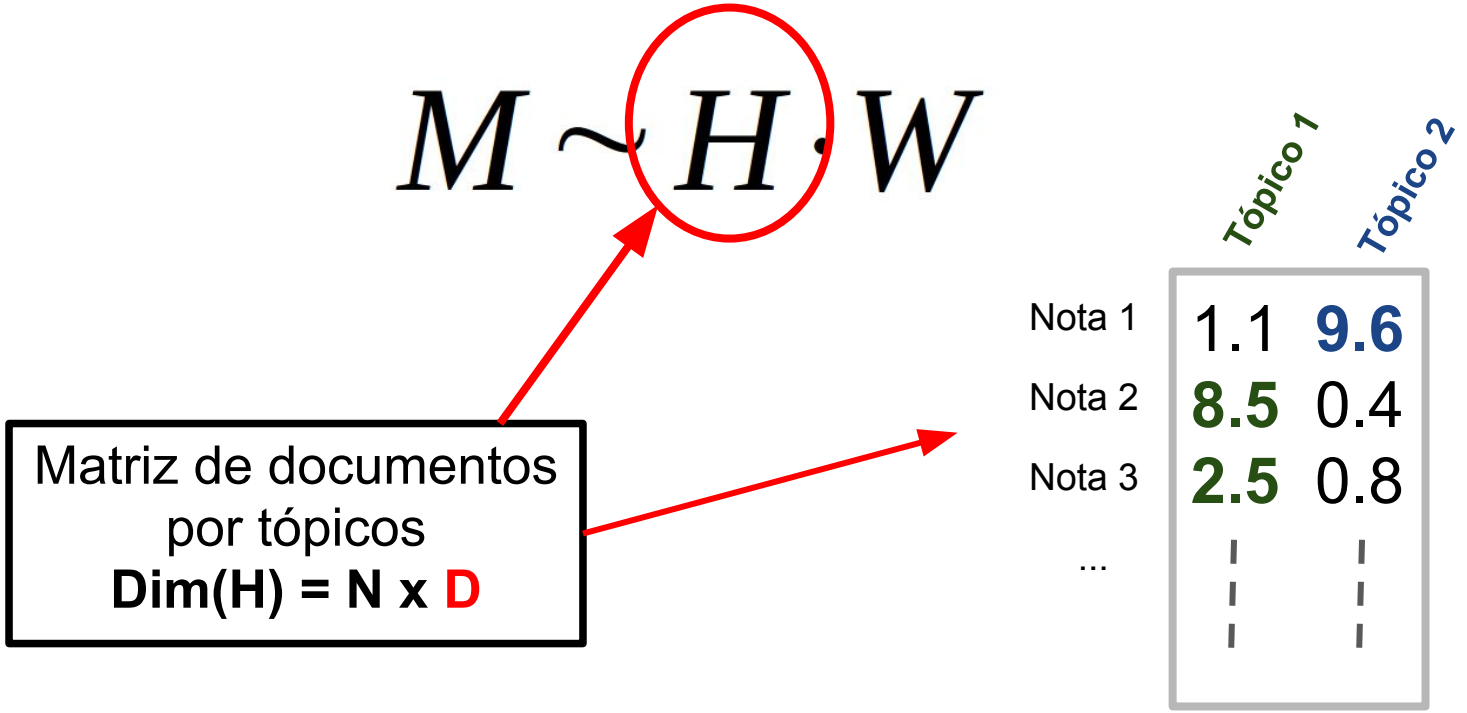
Tópicos en el corpus de noticias (D = 10)

Macri

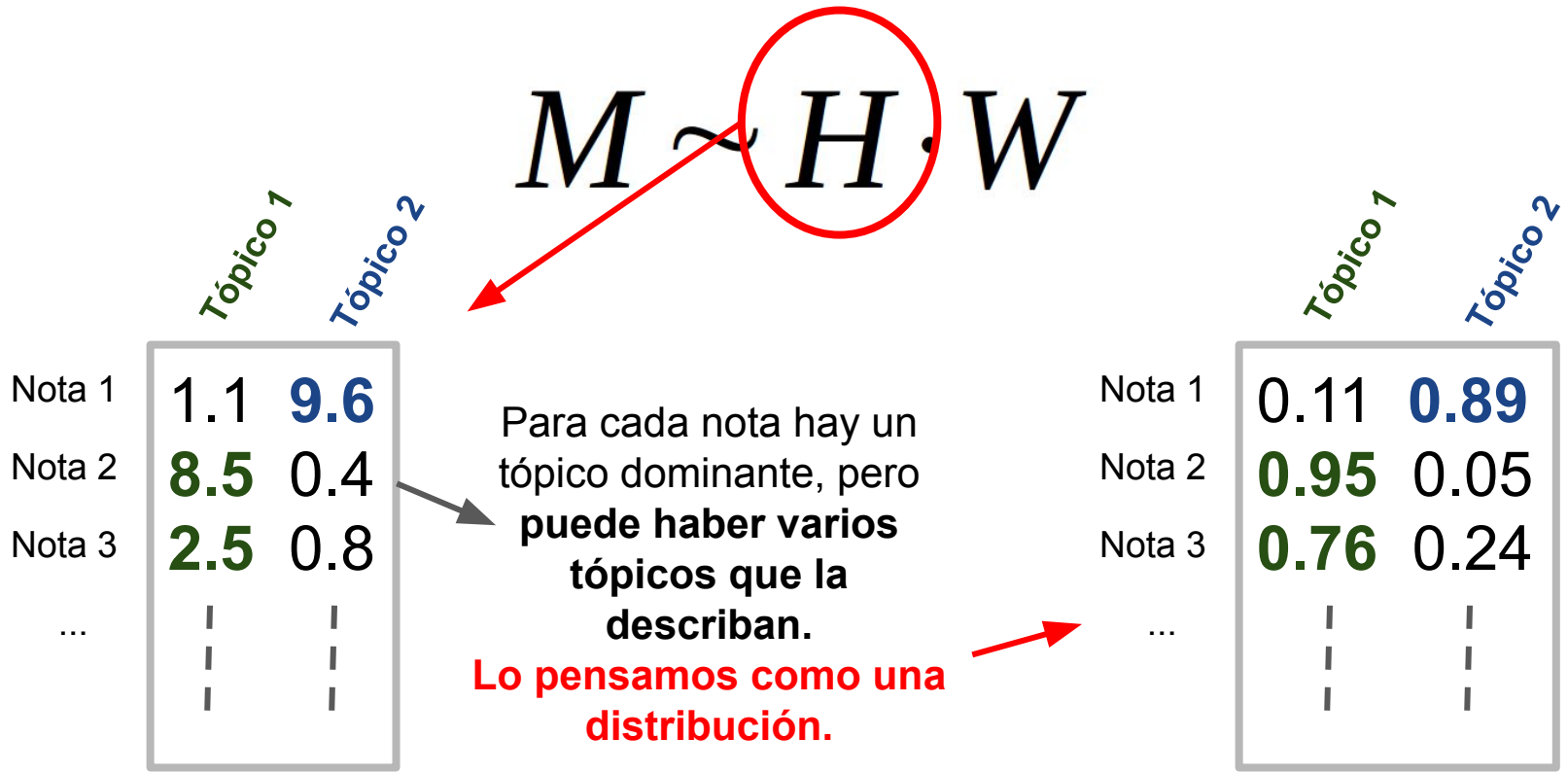
Nisman

Notar que nuestra descripción es finalmente con **7 tópicos**

Interpretación de la matriz H



Interpretación de la matriz H



Interpretación de la matriz H

$$M \sim H \cdot W$$

	Tópico 1	Tópico 2
Nota 1	0.11	**0.89**
Nota 2	**0.95**	0.05
Nota 3	**0.76**	0.24
...		

Recomendaciones para NMF

- Suele andar mejor con la matriz de documentos por términos pesada por **idf** (*inverse document frequency*), es decir, la matriz **tf-idf**.
- Los vectores documentos se suelen entrar normalizados (por ejemplo, a norma euclídea igual a 1).
- Eliminar stopwords es siempre útil: si no lo hiciéramos es muy probable la emergencia de un tópico compuesto por sólo éstas.

Características de NMF

- Los tópicos resultantes son no-ortogonales: esto suele ser dar una interpretación más natural de los tópicos, dado que hay temas que suelen tener overlap.
- Al normalizar los vectores documentos en el espacio de tópicos podemos interpretar dichos vectores como distribuciones (en el espacio de tópicos).
- Desventaja: la cantidad de tópicos D fija además un nivel de resolución. El algoritmo suele encontrar D tópicos de tamaño similar (los tópicos chicos suelen ser absorbidos por los grandes, o bien, uno grande suele partirse en varios).

Latent Dirichlet Allocation (LDA)

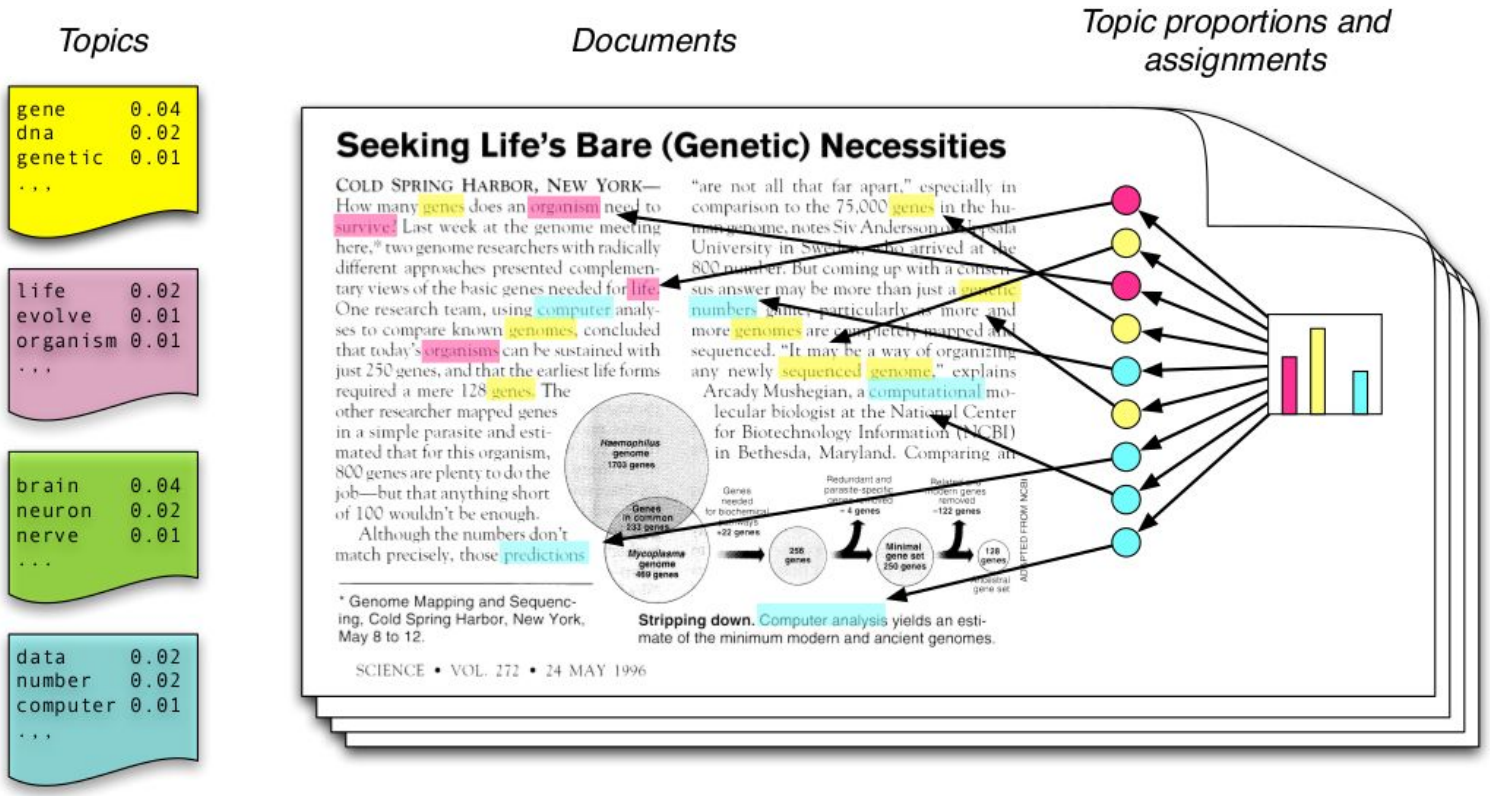
Es un modelo probabilístico generativo, es decir, trata de proponer un modelo para describir cómo se generó la matriz de documentos por términos.

Las hipótesis son:

- un tópico es una distribución en el espacio de términos;

- un documento es una distribución en el espacio de tópicos (es una mixtura de tópicos).

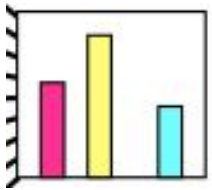
Latent Dirichlet Allocation (LDA)



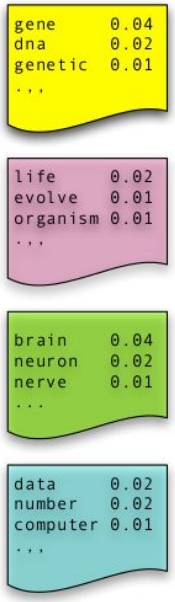
Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.

Latent Dirichlet Allocation (LDA)

¿Cuál es el modelo generativo? La idea es ir construyendo término a término un documento. Supongamos que ya conocemos todas las distribuciones:



Elijo un tópico de la distribución del documento en el espacio de tópicos

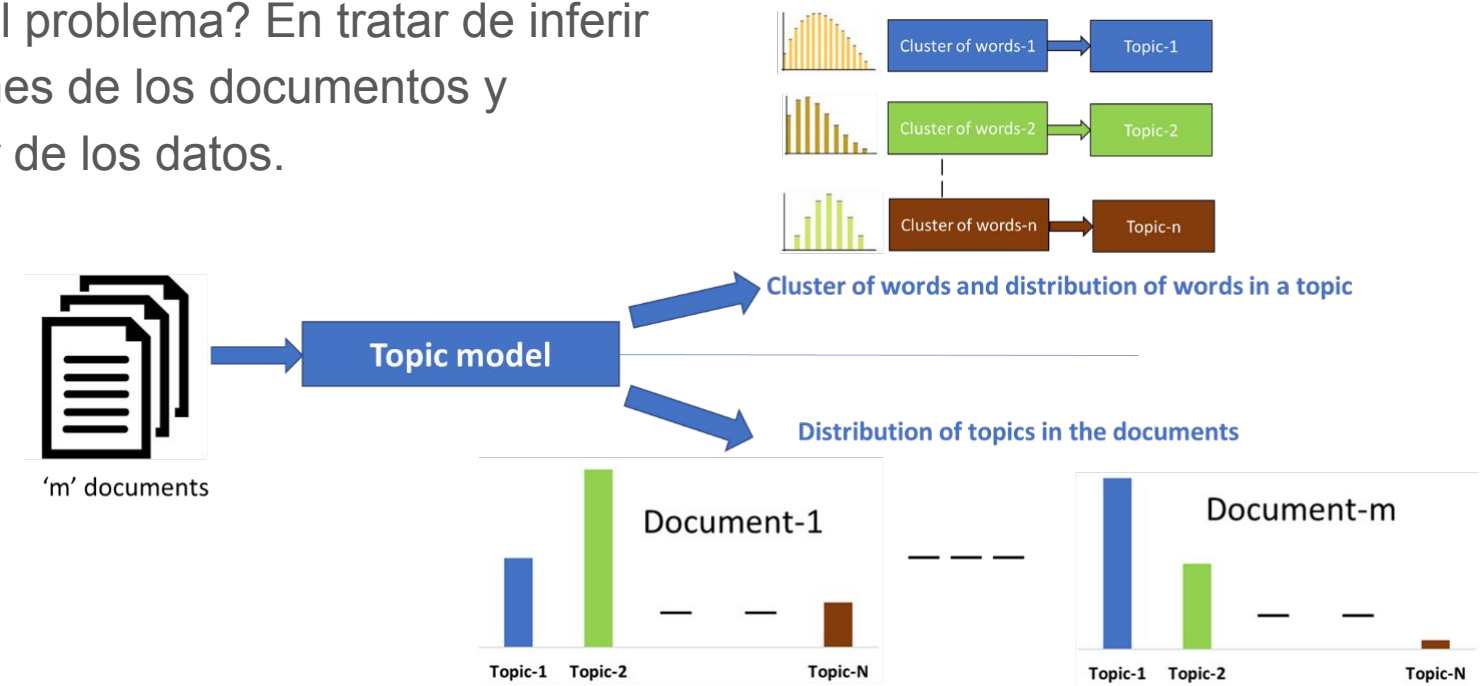


Elijo un término de la distribución del tópico elegido en el espacio de términos

El término elegido forma parte del documento e itero hasta completar los N términos del documentos

Latent Dirichlet Allocation (LDA)

¿Dónde está el problema? En tratar de inferir las distribuciones de los documentos y tópicos a partir de los datos.



https://www.mygreatlearning.com/blog/understanding-latent-dirichlet-allocation/

Latent Dirichlet Allocation (LDA)

La matemática se nos escapa pero el lenguaje es completamente probabilístico: probabilidades condicionales, teorema de Bayes, priors, marginalización, etc...

LDA assumes the following generative process for each document $\mathbf{w}$ in a corpus $\mathcal{D}$:

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
   (b) Choose a word w_n from $p(w_n \mid z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, *3*, 993-1022.

Latent Dirichlet Allocation (LDA)

La matemática se nos escapa pero el lenguaje es completamente probabilístico: probabilidades condicionales, teorema de Bayes, priors, marginalización, etc...

$$p(\mathbf{w}\,|\,\alpha,\beta) = \int p(\theta\,|\,\alpha)\left(\prod_{n=1}^{N}\sum_{z_n} p(z_n\,|\,\theta)\,p(w_n\,|\,z_n,\beta)\right) d\theta.$$

Proba de obtener un documento dado parámetros del modelo

Probabilidad de elegir el tópico del documento

Proba de elegir un término dado un tópico

Objetivo: inferir estos objetos (a través de inferir los parámetros de las distintas distribuciones).

Latent Dirichlet Allocation (LDA)

Ventajas de los modelos generativos:

- Las hipótesis del modelo están explícitas: si el modelo falla (por ejemplo, no encuentra los tópicos correctos en un corpus bien definido) se puede chequear si es porque los datos no cumplen alguna. De variar las hipótesis vienen las extensiones de LDA.
- Generación de datos sintéticos y autoconsistencia: podemos inicializar el modelo con ciertos parámetros, generar datos sintéticos y ver si recuperamos los parámetros originales.

Observaciones

- Si bien vienen de conceptos diferentes, la salida de NMF y LDA es muy similar: objetos que podemos interpretar como distribuciones (en el espacio de tópicos o en el espacio de términos).

- Al menos en su formulación original, en LDA aún no podemos esquivarle a fijarle el número de tópicos antes… es decir, no se infieren con el modelo.

Scikit-learn

sklearn.decomposition.NMF

class sklearn.decomposition.**NMF**(*n_components=None, *, init='warn', solver='cd', beta_loss='frobenius', tol=0.0001, max_iter=200, random_state=None, alpha=0.0, l1_ratio=0.0, verbose=0, shuffle=False, regularization='both'*) [source]

sklearn.decomposition.LatentDirichletAllocation

class sklearn.decomposition.**LatentDirichletAllocation**(*n_components=10, *, doc_topic_prior=None, topic_word_prior=None, learning_method='batch', learning_decay=0.7, learning_offset=10.0, max_iter=10, batch_size=128, evaluate_every=- 1, total_samples=1000000.0, perp_tol=0.1, mean_change_tol=0.001, max_doc_update_iter=100, n_jobs=None, verbose=0, random_state=None*) [source]

Pero no todo termina en scikit-learn...

TextBlob: Simplified Text Processing

spaCy

Entre otras...