

# Minería de Datos

## IIC2433

Latent Semantic Indexing

Vicente Domínguez

# Recordando: Corpus

Un **corpus** es un conjunto de documentos.

Ejemplo:

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

# Vocabulario

Un **vocabulario** es una secuencia ordenada de **palabras** con un identificador único.

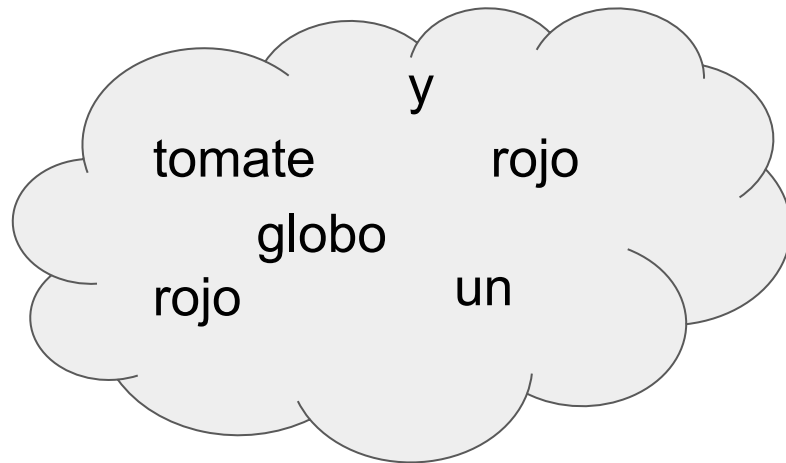
Ejemplo:

ID	palabra
1	amarillo
2	auto
3	globo
4	plátano
5	rojo
6	tomate
7	un
8	verde
9	y

# Bag of Words (*bolsa de palabras*)

Representamos un documento como una **bolsa de palabras**, sin considerar el orden de éstas.

Un tomate rojo y un  
globo rojo.



# Bag of Words (*bolsa de palabras*)

Podemos representar la bolsa de palabras de forma numérica, en una matriz

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

	1	2	3	4	5	6	7	8	9
Doc. 1									
Doc. 2									
Doc. 3									

ID	palabra
1	amarillo
2	auto
3	globo
4	plátano
5	rojo
6	tomate
7	un
8	verde
9	y

# Bag of Words (*bolsa de palabras*)

Podemos representar la bolsa de palabras de forma numérica, en una matriz

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

	1	2	3	4	5	6	7	8	9
Doc. 1	0	1	0	0	1	0	1	0	0
Doc. 2	0	0	1	0	2	1	2	0	1
Doc. 3	1	0	0	1	0	1	2	1	1

ID	palabra
1	amarillo
2	auto
3	globo
4	plátano
5	rojo
6	tomate
7	un
8	verde
9	y

# Tf-idf (*term frequency - inverse document frequency*)

$$\text{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

ID	palabra	idf
1	amarillo	0,48
2	auto	0,48
3	globo	0,48
4	plátano	0,48
5	rojo	0.17
6	tomate	0.17
7	un	0
8	verde	0,48
9	y	0.17

# Tf-idf (*term frequency - inverse document frequency*)

Para representar los documentos multiplicamos la frecuencia de cada palabra **tf** por el peso calculado **idf**

- **Documento 1:** Un auto rojo
- **Documento 2:** Un tomate rojo y un globo rojo.
- **Documento 3:** Un plátano amarillo y un tomate verde.

	1	2	3	4	5	6	7	8	9
Doc. 1	0	0,48	0	0	0.17	0	0	0	0
Doc. 2	0	0	0.48	0	0.34	0	0	0	0
Doc. 3	0,48	0	0	0,48	0	0.17	0	0,48	0.17

ID	palabra	idf
1	amarillo	0,48
2	auto	0,48
3	globo	0,48
4	plátano	0,48
5	rojo	0.17
6	tomate	0.17
7	un	0
8	verde	0,48
9	y	0.17



¿Habrá métodos especializados para la matriz de términos/documentos?

# Latent Semantic Indexing

- La matriz original es demasiado *sparse* para trabajarse computacionalmente.
- LSI busca una representación en baja dimensionalidad de la matriz de documentos.
- Trata de reducir el rango de la matriz original, tratando de armar una matriz aproximada lo más parecida posible.
- Pensado también para mitigar los problemas de **sinonimia** y **polisemia**.

# Latent Semantic Indexing

- Sea  $X$  una matriz donde el elemento  $(i,j)$  corresponde a la ocurrencia del término  $i$  en el documento  $j$ . La matriz  $X$  se ve de la siguiente forma

$$\mathbf{t}_i^T \rightarrow \begin{matrix} & \mathbf{d}_j \\ & \downarrow \\ \begin{bmatrix} x_{1,1} & \dots & x_{1,j} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \dots & x_{i,j} & \dots & x_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,j} & \dots & x_{m,n} \end{bmatrix} \end{matrix}$$

# Latent Semantic Indexing

$$\mathbf{t}_i^T = [x_{i,1} \quad \dots \quad x_{i,j} \quad \dots \quad x_{i,n}]$$

$$\mathbf{d}_j = \begin{bmatrix} x_{1,j} \\ \vdots \\ x_{i,j} \\ \vdots \\ x_{m,j} \end{bmatrix}$$

# Latent Semantic Indexing

- De esta representación, se pueden obtener varias cosas simplemente realizando multiplicaciones
- $\mathbf{t}_i^T \mathbf{t}_p$  : nos da la correlación que existe entre el término  $i$  y el  $p$ , a través de los documentos
- $\mathbf{X}\mathbf{X}^T$  : el producto de estas matrices nos da todas las relaciones de términos.
- $\mathbf{d}_j^T \mathbf{d}_q$  : nos da la correlación que existe entre el documento  $j$  y el  $q$ , a través de los términos
- $\mathbf{X}^T \mathbf{X}$  : el producto de estas matrices nos da todas las relaciones de documentos.

# Latent Semantic Indexing

- De álgebra lineal sabemos que, dada una matriz  $X$  de  $m \times n$ , sabemos que existe una descomposición:

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

- Donde  $U$  y  $V$  son matrices ortogonales y  $\Sigma$  es una matriz diagonal.
- Esta descomposición también es conocida como **SVD** o Singular Value Decomposition.

# Latent Semantic Indexing

- Dada esta descomposición, podemos reconstruir nuestras matrices de relaciones.

$$XX^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V^{TT} \Sigma^T U^T)$$

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

$$X^T X = (U\Sigma V^T)^T (U\Sigma V^T) = (V^{TT} \Sigma^T U^T)(U\Sigma V^T)$$

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

# Latent Semantic Indexing

- Dada esta descomposición, podemos reconstruir nuestras matrices de relaciones.

$$XX^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V^{TT} \Sigma^T U^T)$$

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

$$X^T X = (U\Sigma V^T)^T (U\Sigma V^T) = (V^{TT} \Sigma^T U^T)(U\Sigma V^T)$$

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



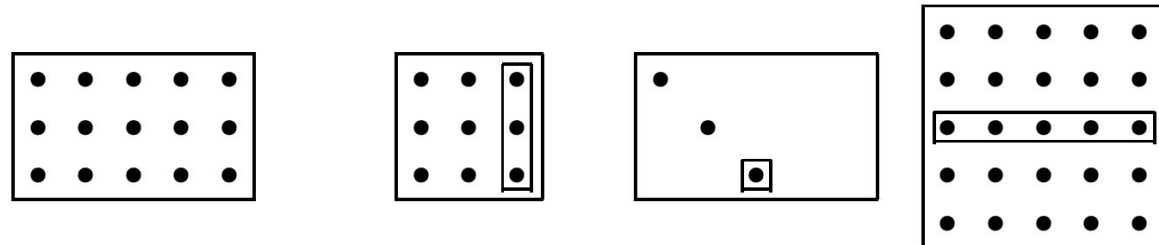
# Latent Semantic Indexing

- Ahora, dado que  $\Sigma\Sigma^T$  y  $\Sigma^T\Sigma$  son diagonales. Vemos que la matriz  $U$  debe contener los vectores propios de  $XX^T$  y la matriz de  $V$  contener los vectores propios de  $X^TX$  lo que esto. La descomposición puede verse de la siguiente forma.

$$\begin{array}{ccccccc}
 & & X & & U & & \Sigma & & V^T \\
 & & (\mathbf{d}_j) & & & & & & (\hat{\mathbf{d}}_j) \\
 & & \downarrow & & & & & & \downarrow \\
 (\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,j} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \dots & x_{i,j} & \dots & x_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,j} & \dots & x_{m,n} \end{bmatrix} & = & (\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \mathbf{u}_1 \end{bmatrix} & \dots & \begin{bmatrix} \mathbf{u}_l \end{bmatrix} \end{bmatrix} & \cdot & \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} & \cdot & \begin{bmatrix} \begin{bmatrix} \mathbf{v}_1 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \mathbf{v}_l \end{bmatrix} \end{bmatrix}
 \end{array}$$

# Latent Semantic Indexing

- ¿Qué podemos hacer con esto?
- Está demostrado que si elegimos los mayores valores singulares con sus correspondientes vectores singulares, obtendremos una matriz con el menor error posible utilizando la norma de Frobenius.

$$C_k = U \Sigma_k V^T$$


The diagram illustrates the matrix equation  $C_k = U \Sigma_k V^T$  using visual representations of matrices as grids of dots:

- $C_k$ : A 5x5 matrix represented by a grid of 25 dots.
- $U$ : A 5x5 matrix represented by a grid of 25 dots, with the last column highlighted by a vertical rectangle.
- $\Sigma_k$ : A 5x5 matrix represented by a grid of 25 dots, with the bottom-right element highlighted by a small square.
- $V^T$ : A 5x5 matrix represented by a grid of 25 dots, with the second row highlighted by a horizontal rectangle.

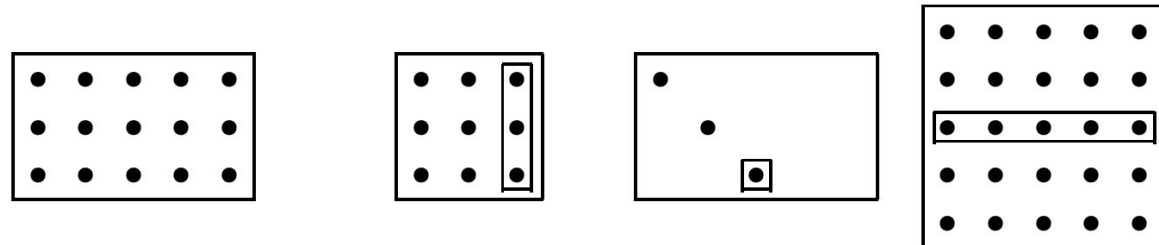
# Latent Semantic Indexing

- ¿Qué podemos hacer con esto?
- Está demostrado que si elegimos los mayores valores singulares con sus correspondientes vectores singulares, obtendremos una matriz con el menor error posible utilizando la norma de Frobenius.

$$X = C - C_k \quad \|X\|_F = \sqrt{\sum_{i=1}^M \sum_{j=1}^N X_{ij}^2}.$$

# Latent Semantic Indexing

- ¿Qué podemos hacer con esto?
- Está demostrado que si elegimos los mayores valores singulares con sus correspondientes vectores singulares, obtendremos una matriz con el menor error posible utilizando la norma de Frobenius.

$$C_k = U \Sigma_k V^T$$


The diagram illustrates the matrix equation  $C_k = U \Sigma_k V^T$  using visual representations of matrices as grids of dots:

- $C_k$ : A 5x5 matrix represented by a grid of 25 dots.
- $U$ : A 5x5 matrix represented by a grid of 25 dots, with the last column highlighted by a vertical rectangle.
- $\Sigma_k$ : A 5x5 matrix represented by a grid of 25 dots, with the bottom-right element highlighted by a small square.
- $V^T$ : A 5x5 matrix represented by a grid of 25 dots, with the second row highlighted by a horizontal rectangle.

# Latent Semantic Indexing

- Finalmente obtenemos esto:

$$X_k = U_k \Sigma_k V_k^T$$

- ¿Qué podemos hacer con esto?

# Latent Semantic Indexing

- Se pueden ver que tan relacionados estan dos documentos o dos términos en un espacio dimensional más pequeño.
- Se pueden clusterizar documentos.
- Se pueden realizar consultas de busqueda, transformando la consulta en un mini documento.
- Se ha mostrado que palabras con sinonimia si otorgan resultados similares en las consultas.