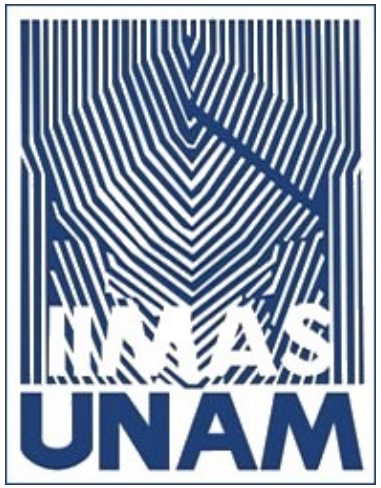
Procesamiento de Lenguaje Natural

Modelo de espacio vectorial – Vectores de Palabras



Dra. Helena Gómez Adorno

helena.gomez@iimas.unam.mx

Dra. Gemma Bel

gbele@iingen.unam.mx

Correo del curso:

pln.cienciadedatos@gmail.com

Asistente:

Luis Ramon Casillas

Parte de estas diapositivas están basadas en la especialización de PLN de Coursera

Contenido

- Modelos de espacio vectorial (VSM)
- Ventajas
- Aplicaciones
- Co-ocurrencia → Representación vectorial
- Distancia euclidiana
- Similitud coseno
- Uso de vectores de palabras
- Análisis de componentes principales (PCA)

Por qué usar el modelo espacio vectorial?

De donde eres?

De donde vienes?

↓

Significado diferente

Cuál es tu edad?

Cuántos años tienes?

↓

Mismo significado

Aplicaciones del VSM

Yo uso tenis para correr

Yo uso sandalias para caminar

Extracción de Información

Traducción automática

Chatbots

¿Qué *significa* una palabra?

- Obviamente, el significado de las palabras es realmente difícil de cuantificar incluso para los humanos.
- Entonces, ¿cómo podemos generar representaciones del significado de las palabras automáticamente? Lo abordamos de forma oblicua, utilizando lo que se conoce como hipótesis distribucional.

[1]https://aclweb.org/aclwiki/Distributional Hypothesis

La hipótesis distribucional

- Tomado de la lingüística: el significado de una palabra se puede determinar a partir de los contextos en los que aparece[1].
  - Las palabras con contextos similares tienen significados similares (Harris, 1954)
  - “Conocerás una palabra por la compañía que tiene” (Firth, 1957)

Firth, 1957

(Firth, J. R. 1957:11)

[1]https://aclweb.org/aclwiki/Distributional Hypothesis

Contexto?

- La mayoría de las representaciones vectoriales de palabras estáticas utilizan una noción simple de contexto: una palabra es un "contexto" para otra palabra cuando aparece lo suficientemente cerca de ella en el texto.
- Pero también podemos utilizar frases o documentos completos como contextos. En el caso más básico, fijamos un número de palabras como nuestra "ventana de contexto" y contamos todos los pares de palabras que están a menos de esa cantidad de palabras entre sí como co-ocurrencias.

Ejemplo: Co-ocurrencia de palabras

I like simple data

I prefer simple raw data

k=2

	simple	raw	like	I
data	2	1	1	0

Otras representaciones basadas en conteo

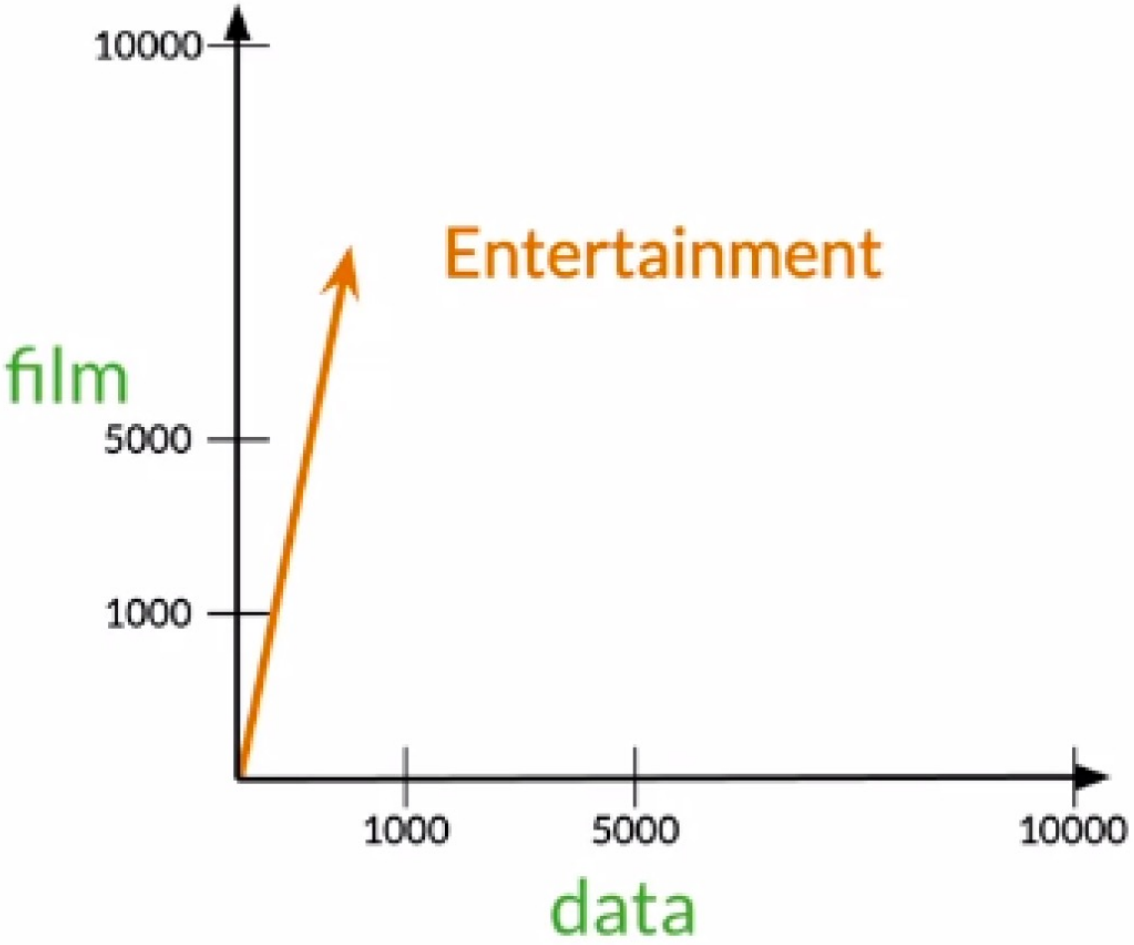
- Podemos generalizar un poco más este concepto.
  - Contextos más generales: oraciones o documentos
  - Ventanas de contexto ponderadas
  - Otras medidas de asociación palabra-contexto (por ejemplo, información mutua puntual)
- Esto nos da la noción más general de una matriz de contexto de palabras.

Ejemplo: palabra-documento

- Número de veces que una palabra ocurre dentro de cierta categoría

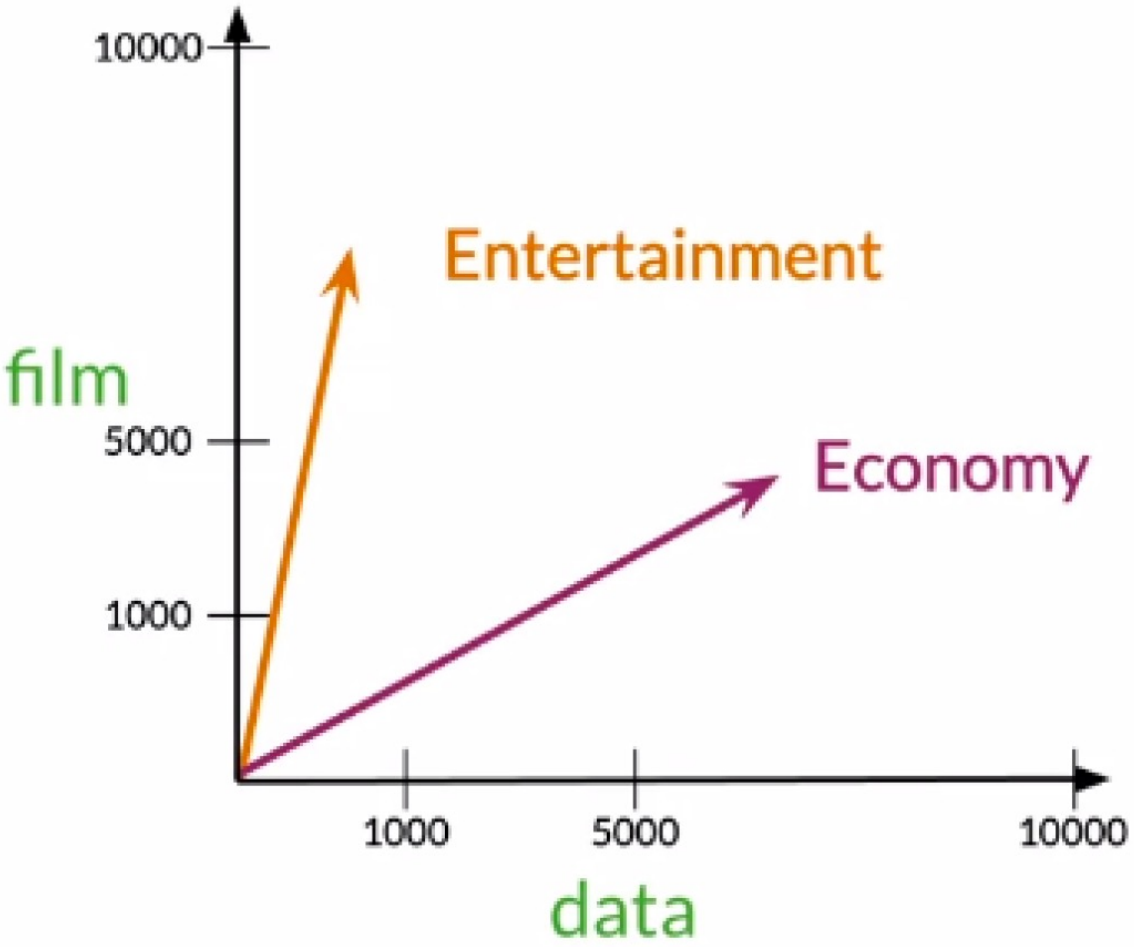
	Entertainment	Economy	Machine Learning
data	500	6620	9320
film	7000	4000	1000

Espacio vectorial



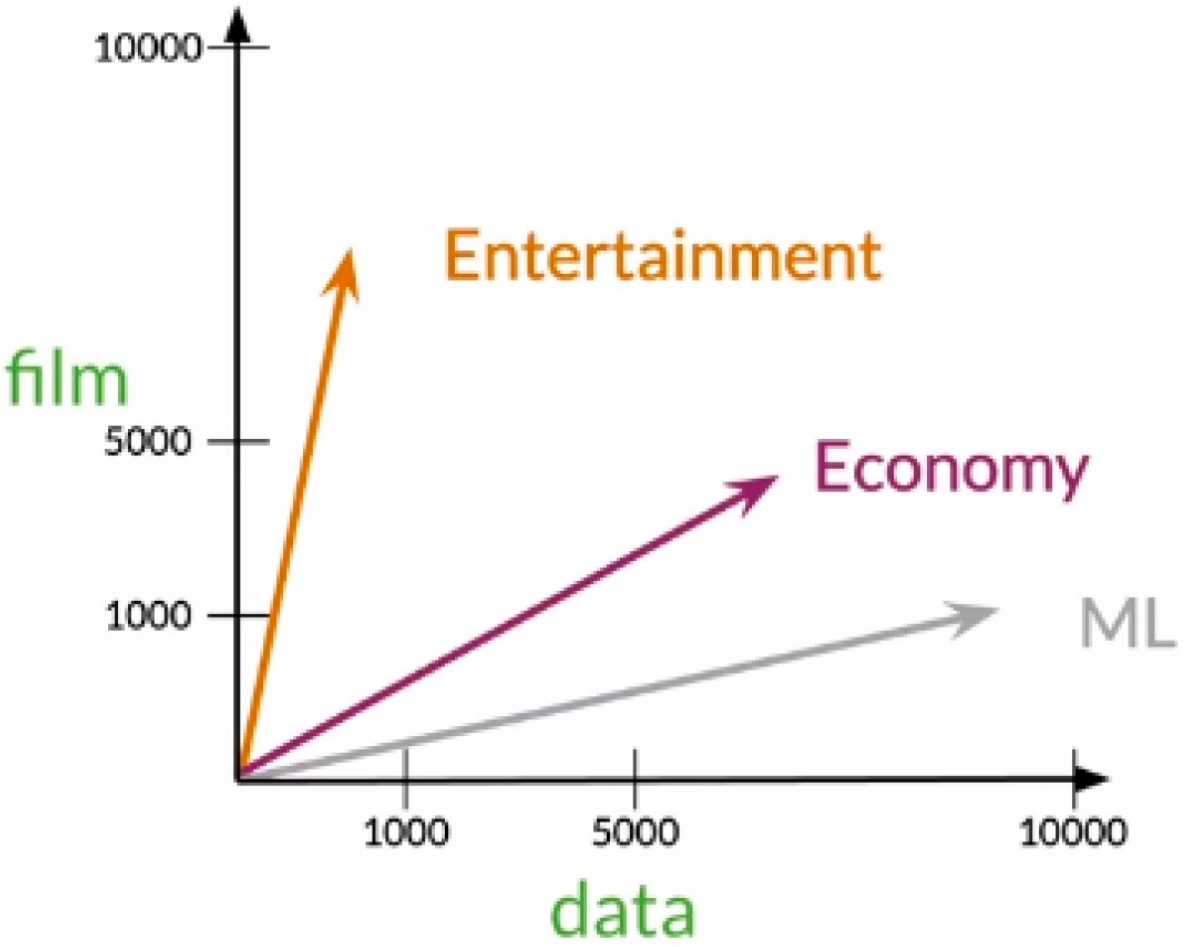
	Entertainment	Economy	ML
data	500	6620	9320
film	7000	4000	1000

Espacio vectorial



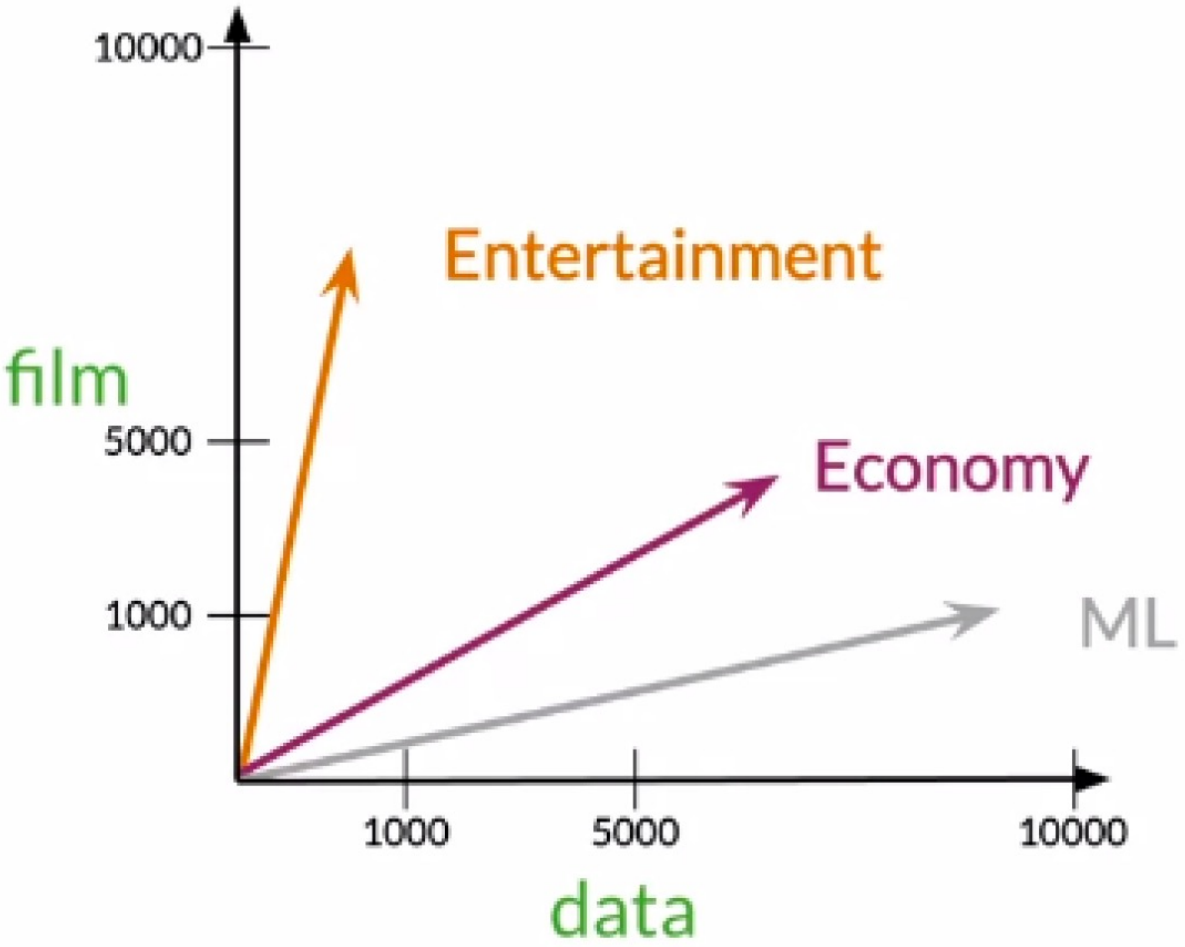
	Entertainment	Economy	ML
data	500	6620	9320
film	7000	4000	1000

Espacio vectorial



	Entertainment	Economy	ML
data	500	6620	9320
film	7000	4000	1000

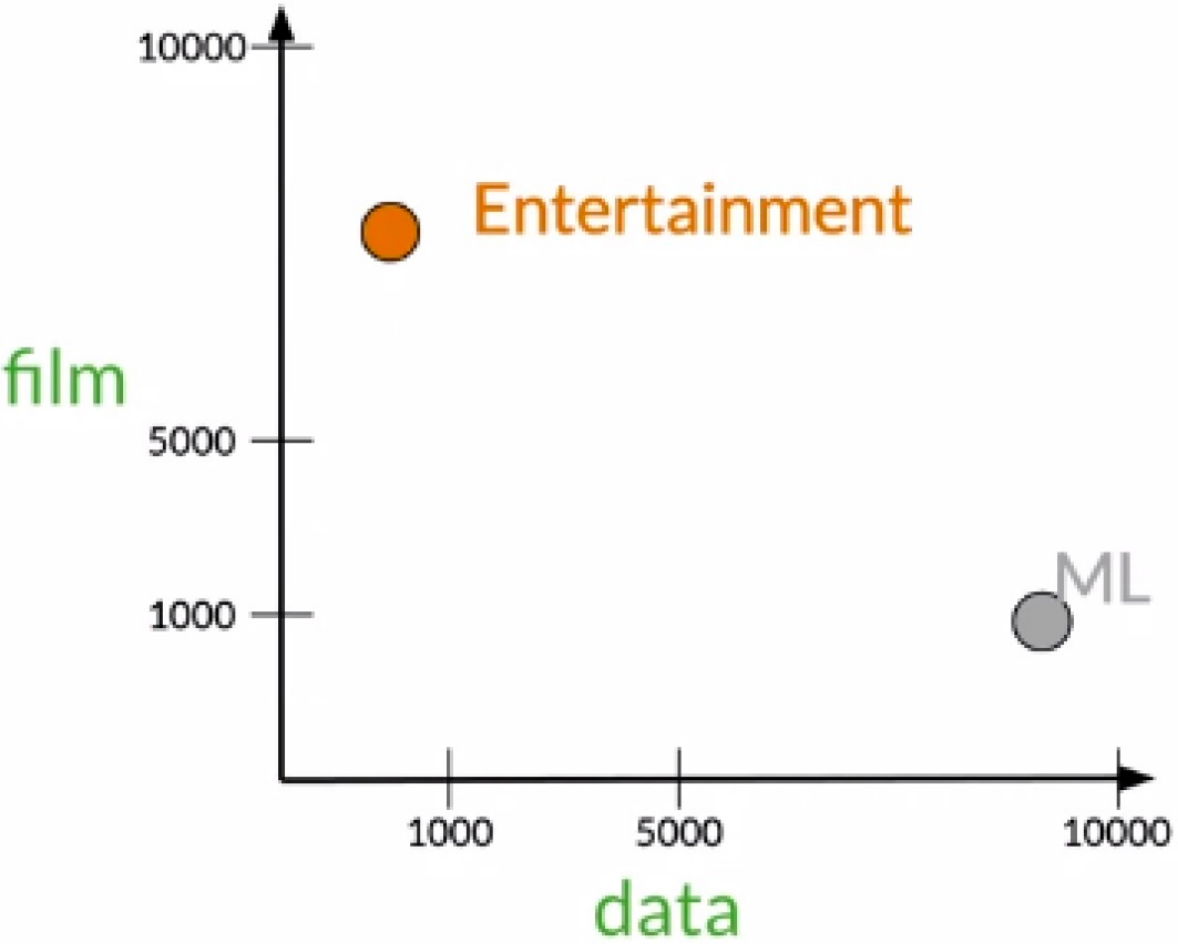
Espacio vectorial



	Entertainment	Economy	ML
data	500	6620	9320
film	7000	4000	1000

Medidas de “similitud”
-Ángulo
-Distancia

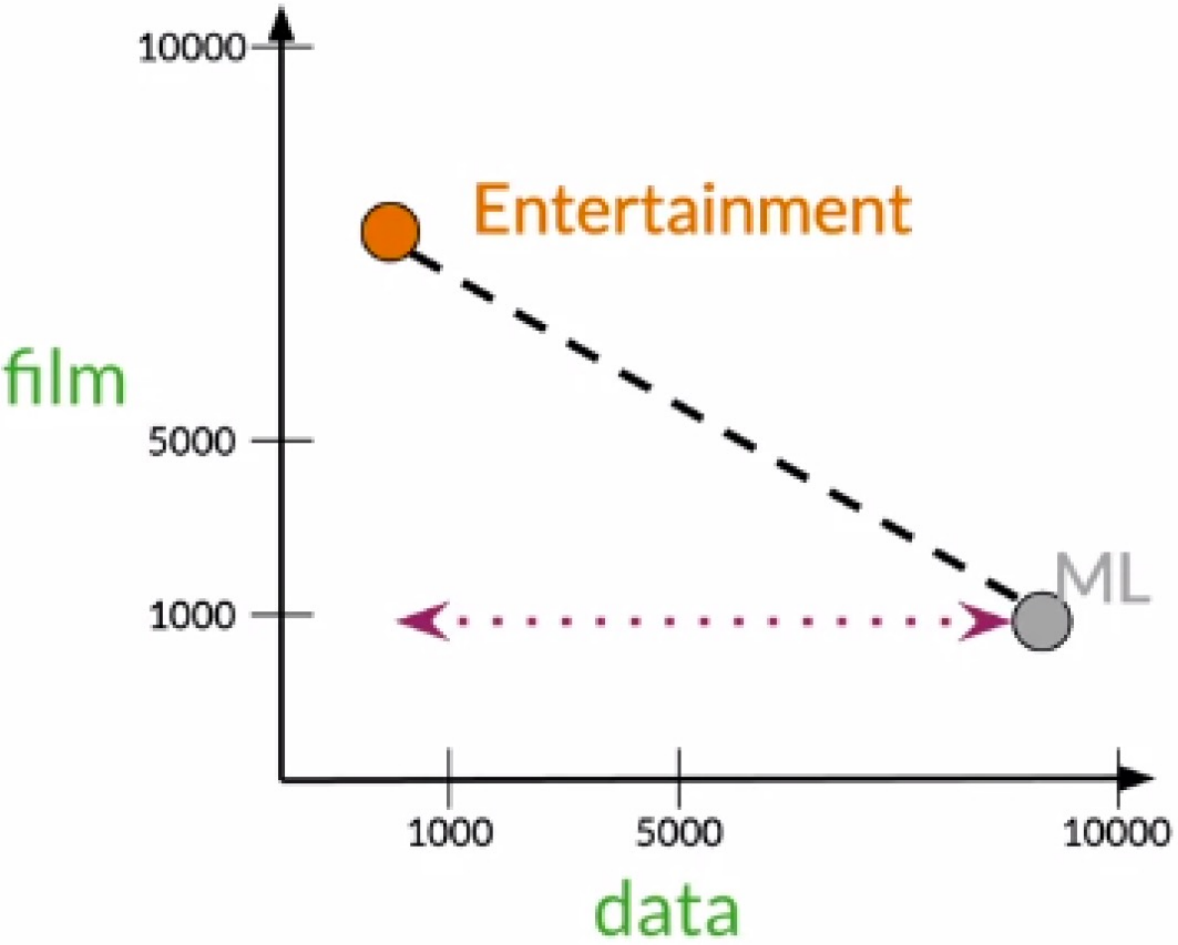
Distancia euclidiana



Corpus A: (500,7000)

Corpus B: (9320,1000)

Distancia euclidiana

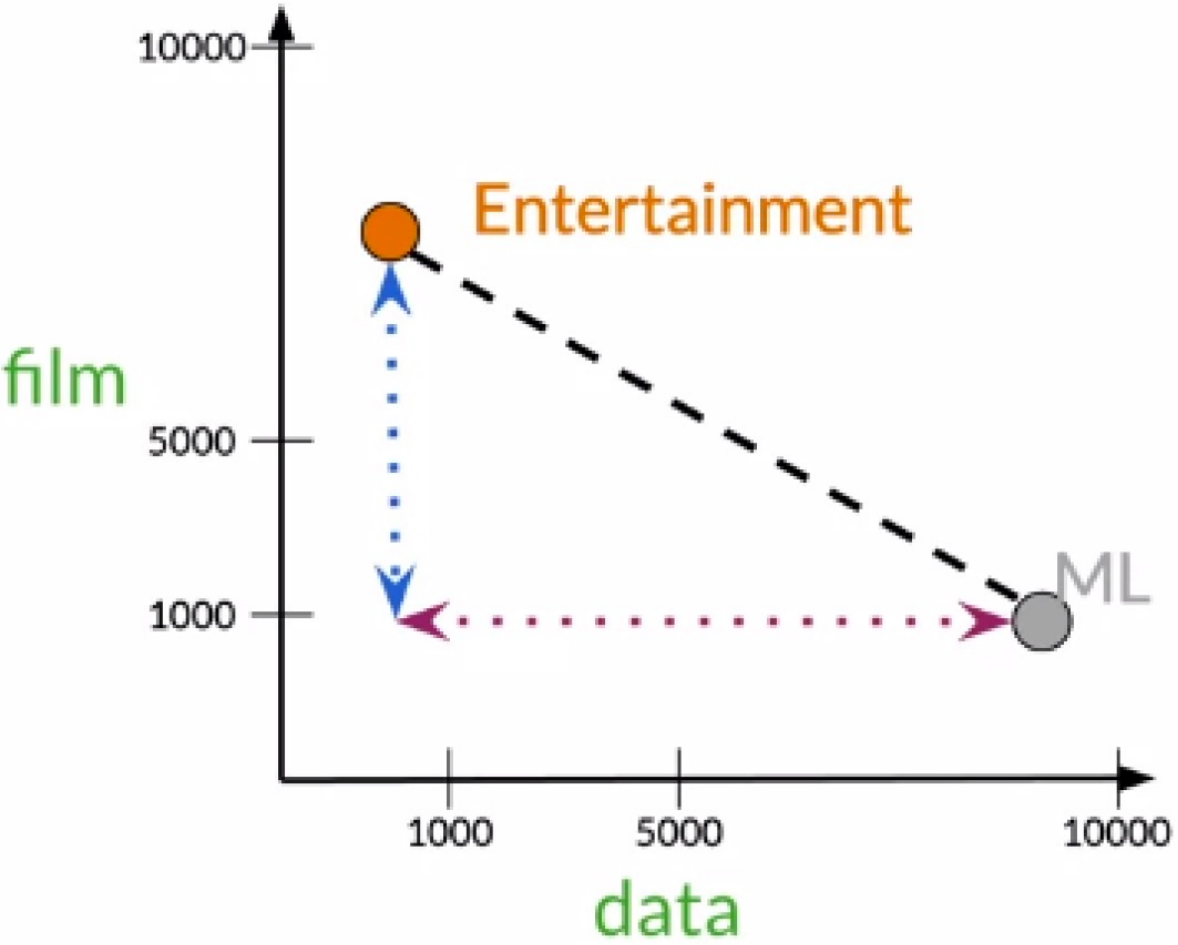


Corpus A: (500,7000)

Corpus B: (9320,1000)

$$d(B, A) = \sqrt{(B_1 - A_1)^2 + (B_2 - A_2)^2}$$

Distancia euclidiana

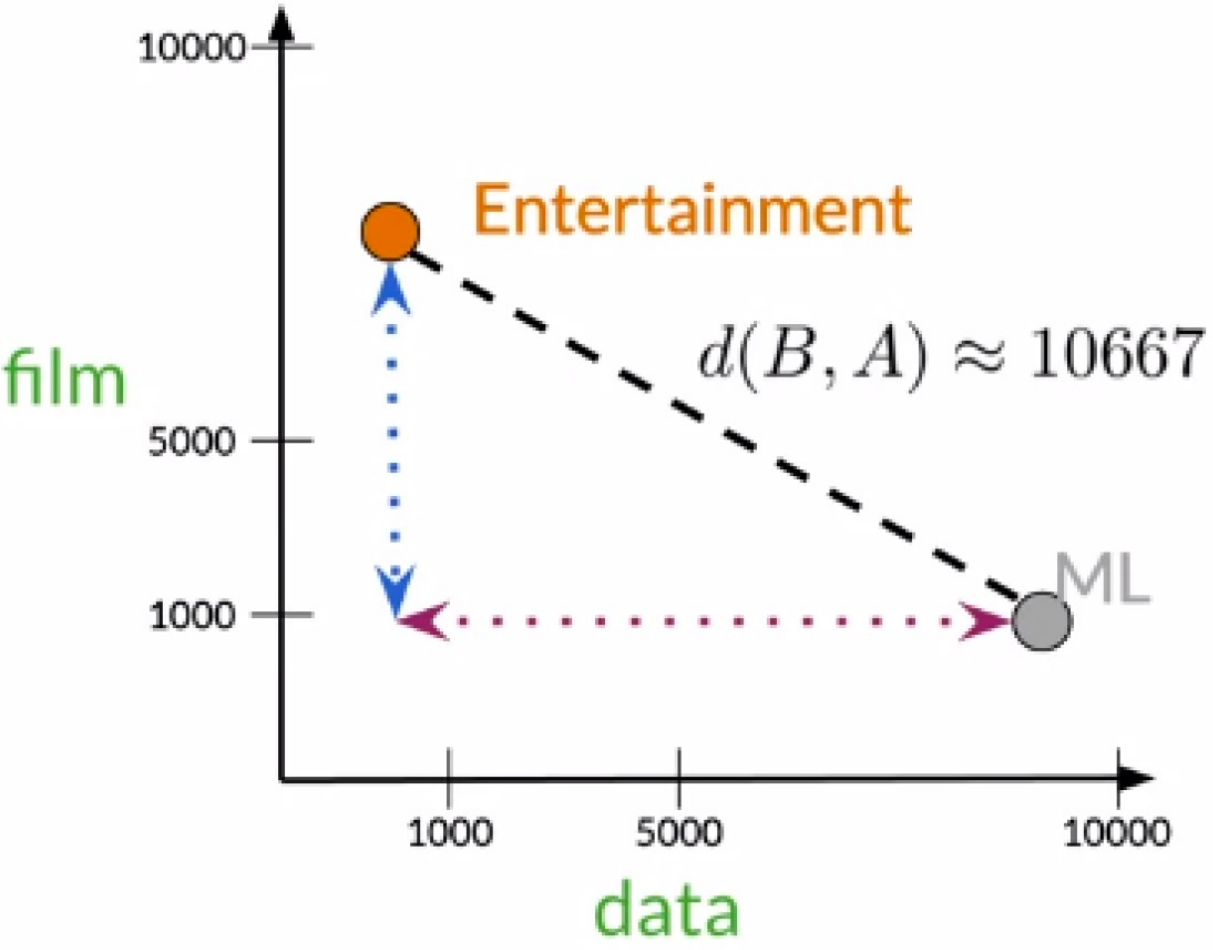


Corpus A: (500,7000)

Corpus B: (9320,1000)

$$d(B, A) = \sqrt{(B_1 - A_1)^2 + (B_2 - A_2)^2}$$

Distancia euclidiana



Corpus A: (500,7000)

Corpus B: (9320,1000)

$$d(B, A) = \sqrt{(B_1 - A_1)^2 + (B_2 - A_2)^2}$$

$$c^2 = a^2 + b^2$$

$$d(B, A) = \sqrt{(-8820)^2 + (6000)^2}$$

Distancia euclidiana para vectores n-dimensionales

	data	boba	$\vec{v}$ ice-cream
AI	6	0	1
drinks	0	4	6
food	0	6	8

Distancia euclidiana para vectores n-dimensionales

	data	boba ($\vec{w}$)	ice-cream ($\vec{v}$)
AI	6	0	1
drinks	0	4	6
food	0	6	8

Distancia euclidiana para vectores n-dimensionales

	data	$\vec{w}$ boba	$\vec{v}$ ice-cream
AI	6	0	1
drinks	0	4	6
food	0	6	8

$$= \sqrt{(1-0)^2 + (6-4)^2 + (8-6)^2}$$

$$= \sqrt{1+4+4} = \sqrt{9} = 3$$

$$d(\vec{v}, \vec{w}) = \sqrt{\sum_{i=1}^{n} (v_i - w_i)^2}$$

Distancia euclidiana para vectores n-dimensionales

	data	boba ($\vec{w}$)	ice-cream ($\vec{v}$)
AI	6	0	1
drinks	0	4	6
food	0	6	8

$$= \sqrt{(1-0)^2 + (6-4)^2 + (8-6)^2}$$

$$= \sqrt{1+4+4} = \sqrt{9} = 3$$

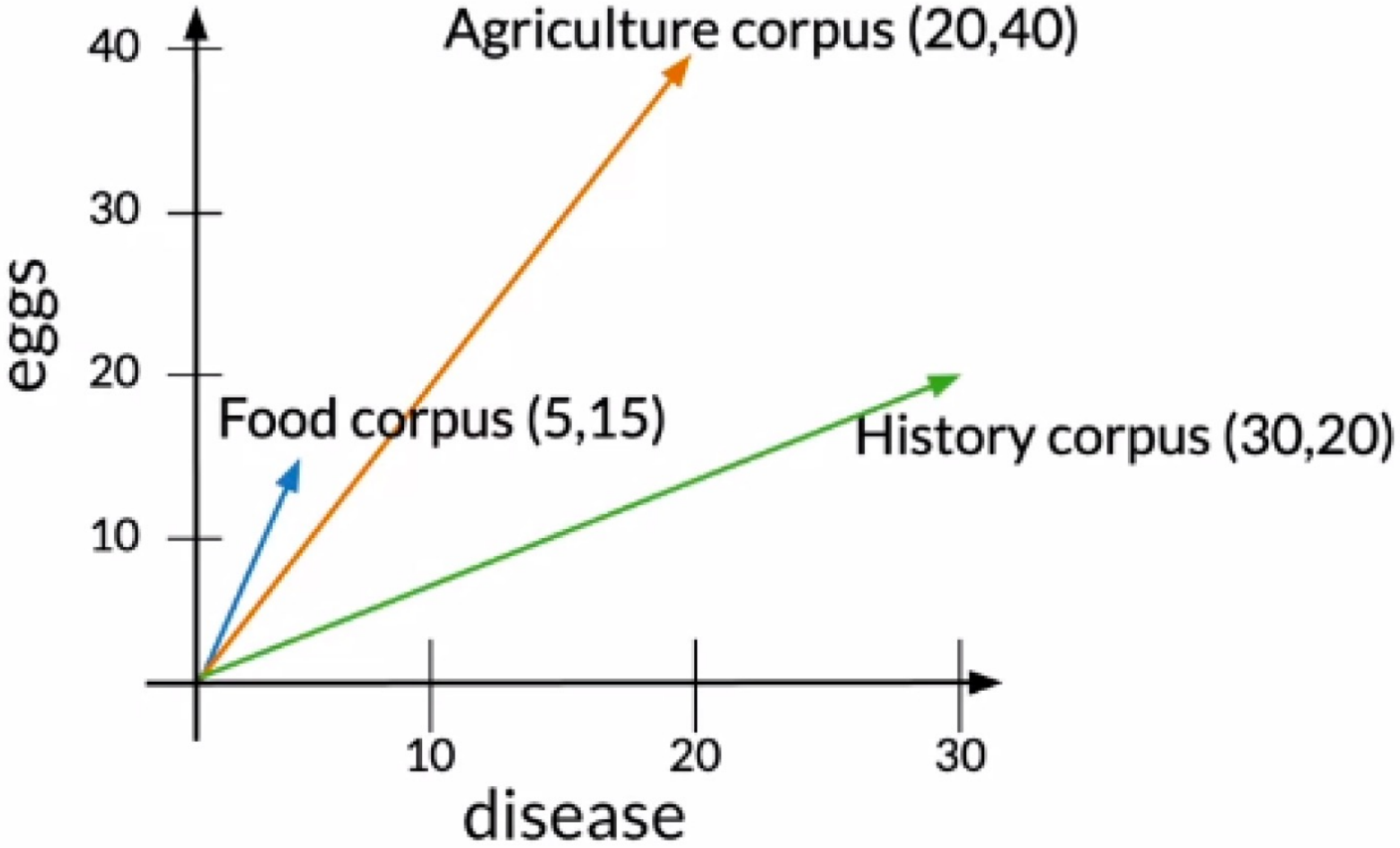
$$d(\vec{v}, \vec{w}) = \sqrt{\sum_{i=1}^{n} (v_i - w_i)^2} \longrightarrow \text{Norm of } (\vec{v} - \vec{w})$$

Distancia euclidiana en Python

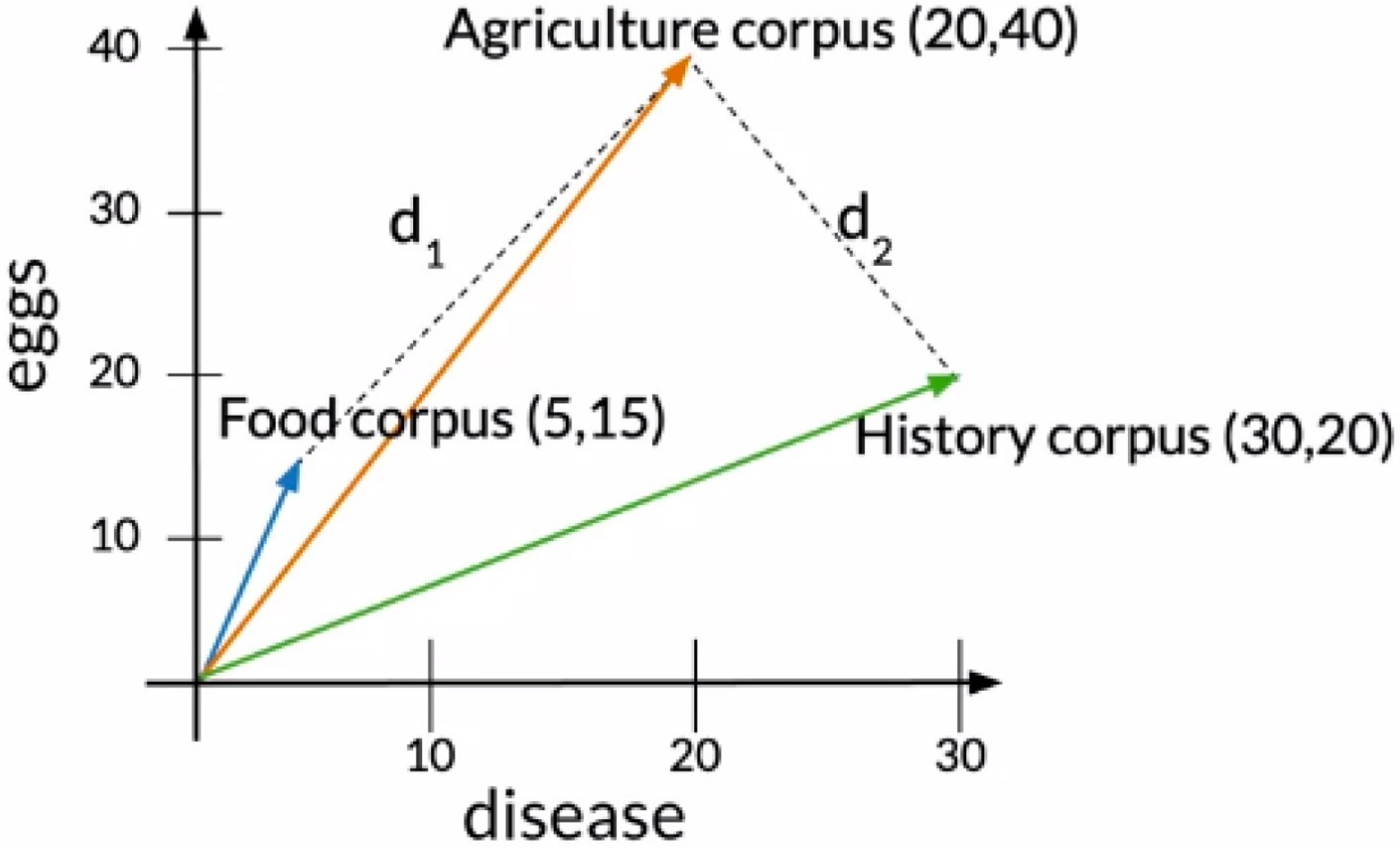
```
# Create numpy vectors v and w
v = np.array([1, 6, 8])
w = np.array([0, 4, 6])

# Calculate the Euclidean distance d
d = np.linalg.norm(v-w)
# Print the result
print("The Euclidean distance between v and w is: ", d)
```

```
The Euclidean distance between v and w is: 3
```

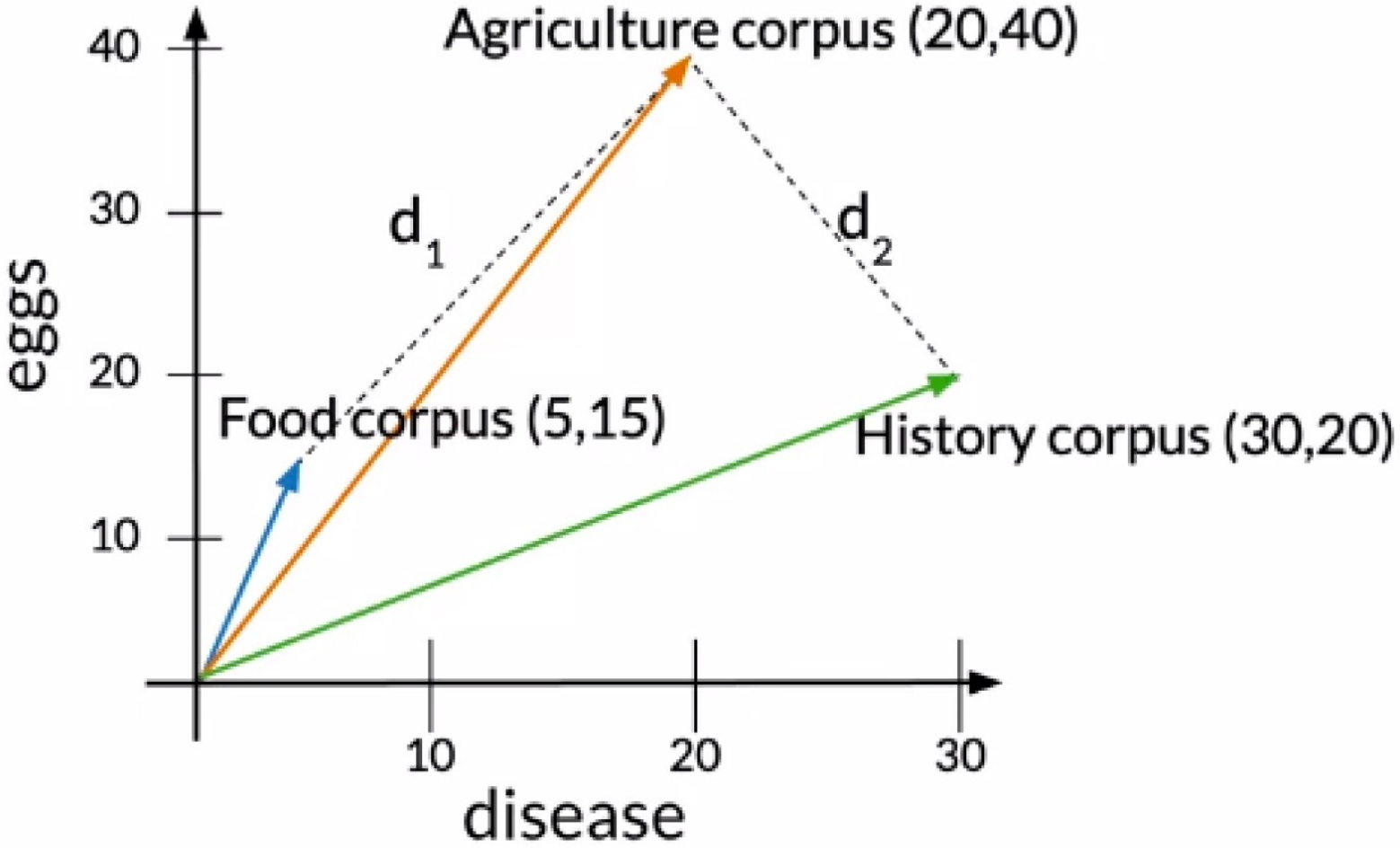
Distancia euclidiana vs. Similitud coseno



Distancia euclidiana vs. Similitud coseno

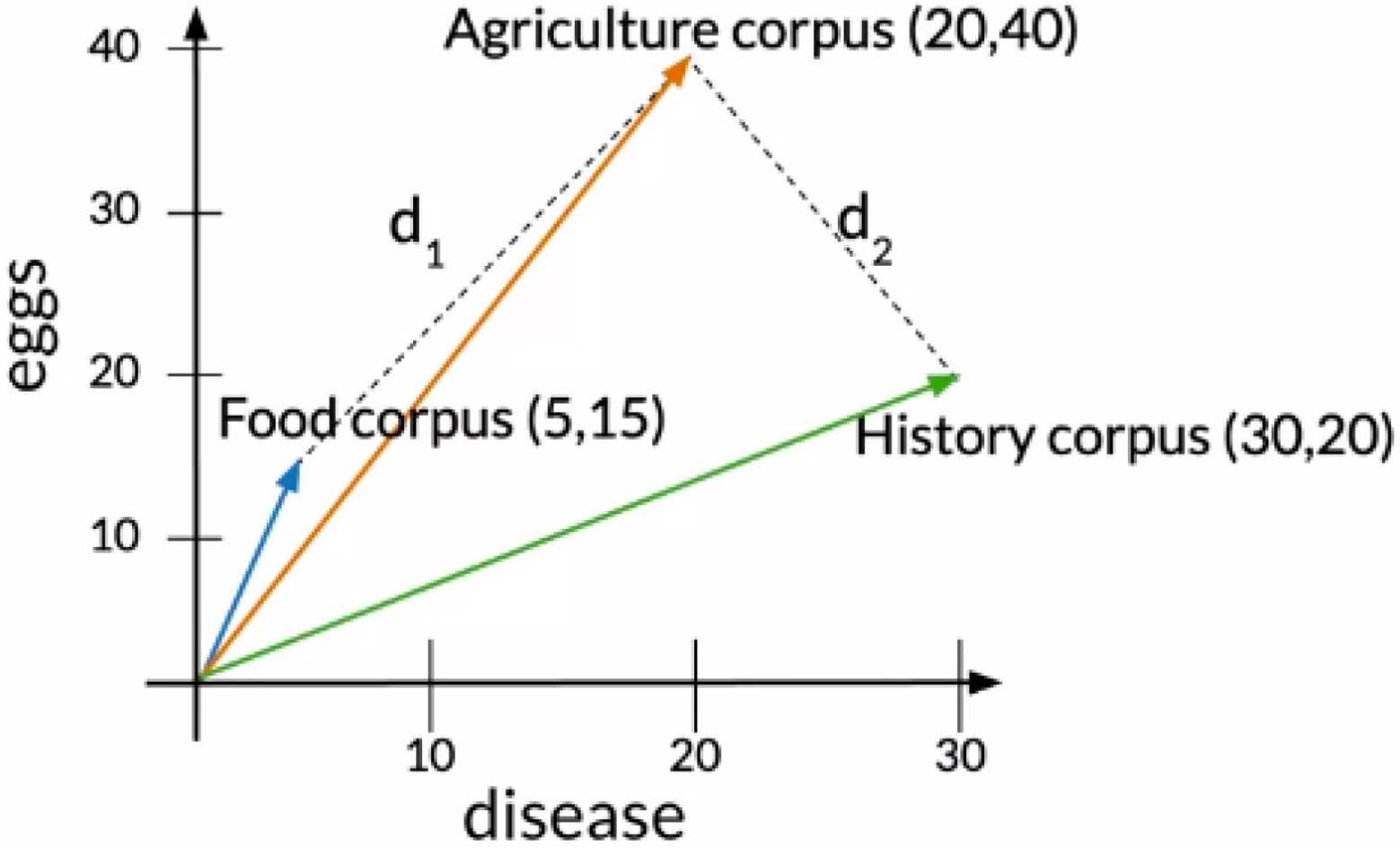


Distancia euclidiana vs. Similitud coseno



Euclidean distance: $d_2 < d_1$

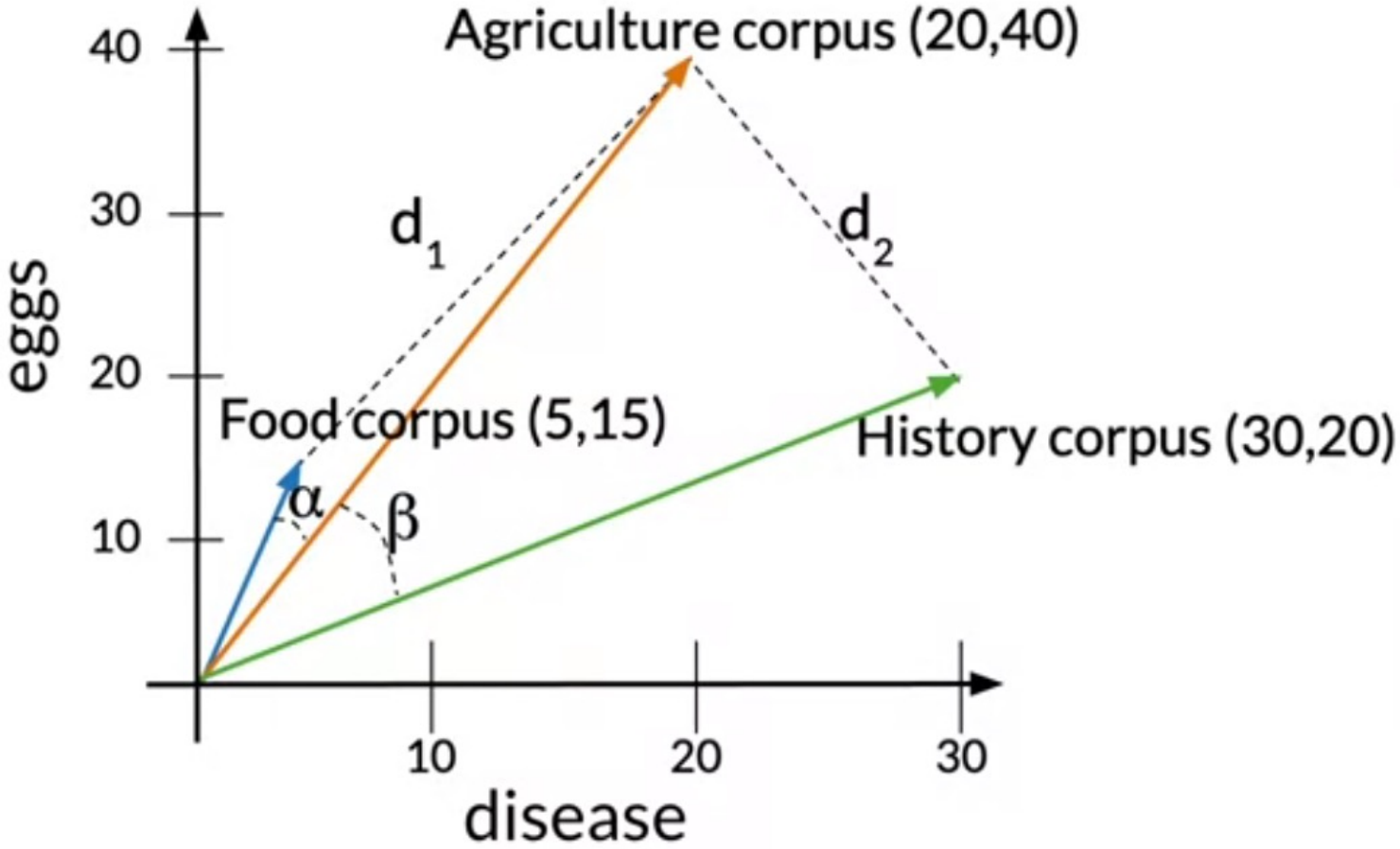
Distancia euclidiana vs. Similitud coseno



Euclidean distance: $d_2 < d_1$

El coseno del ángulo entre los vectores

Distancia euclidiana vs. Similitud coseno



Euclidean distance: $d_2 < d_1$

El coseno del ángulo entre los vectores

Definiciones previas

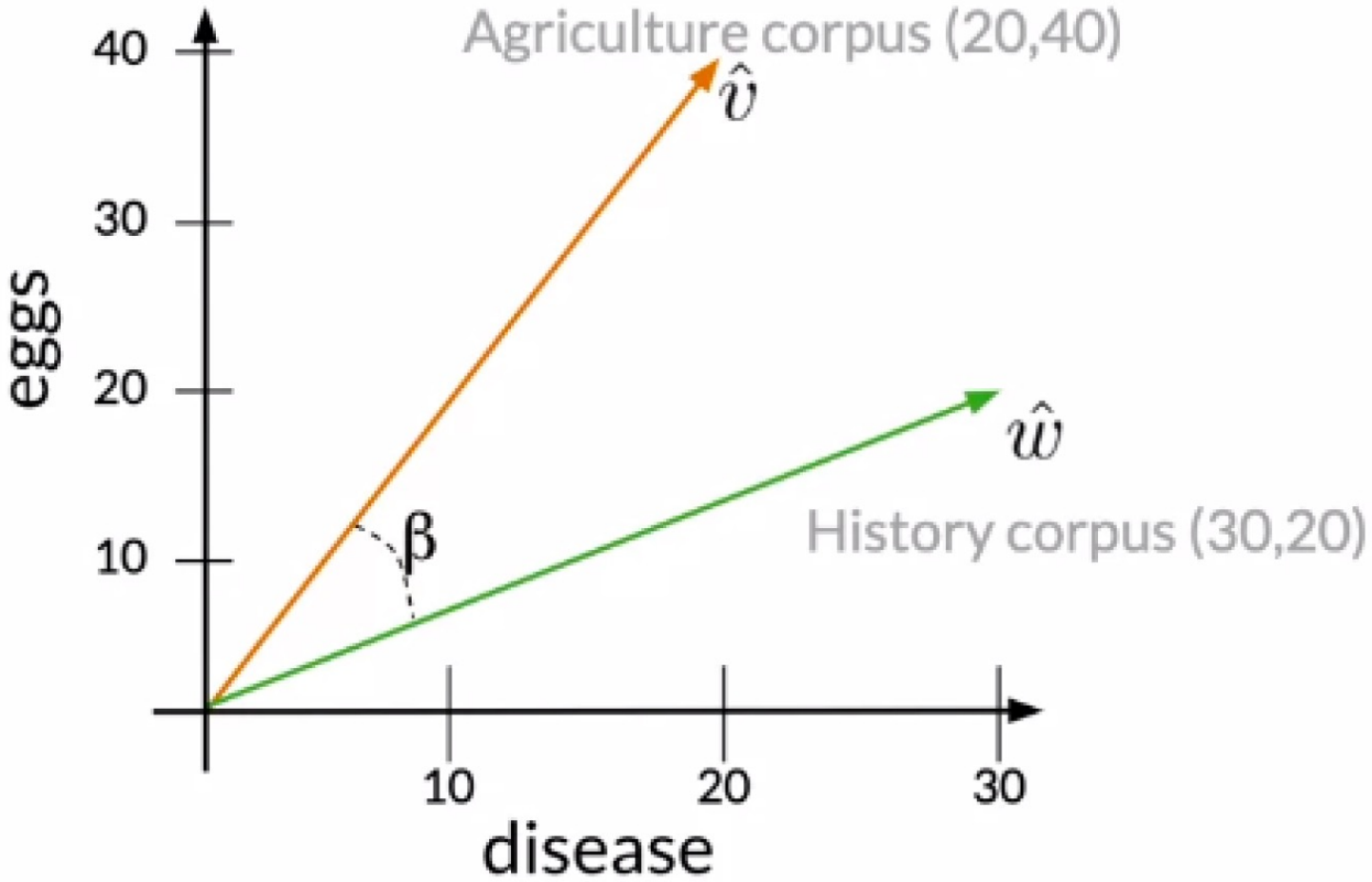
Norma del vector

$$\|\vec{v}\| = \sqrt{\sum_{i=1}^{n} v_i^2}$$

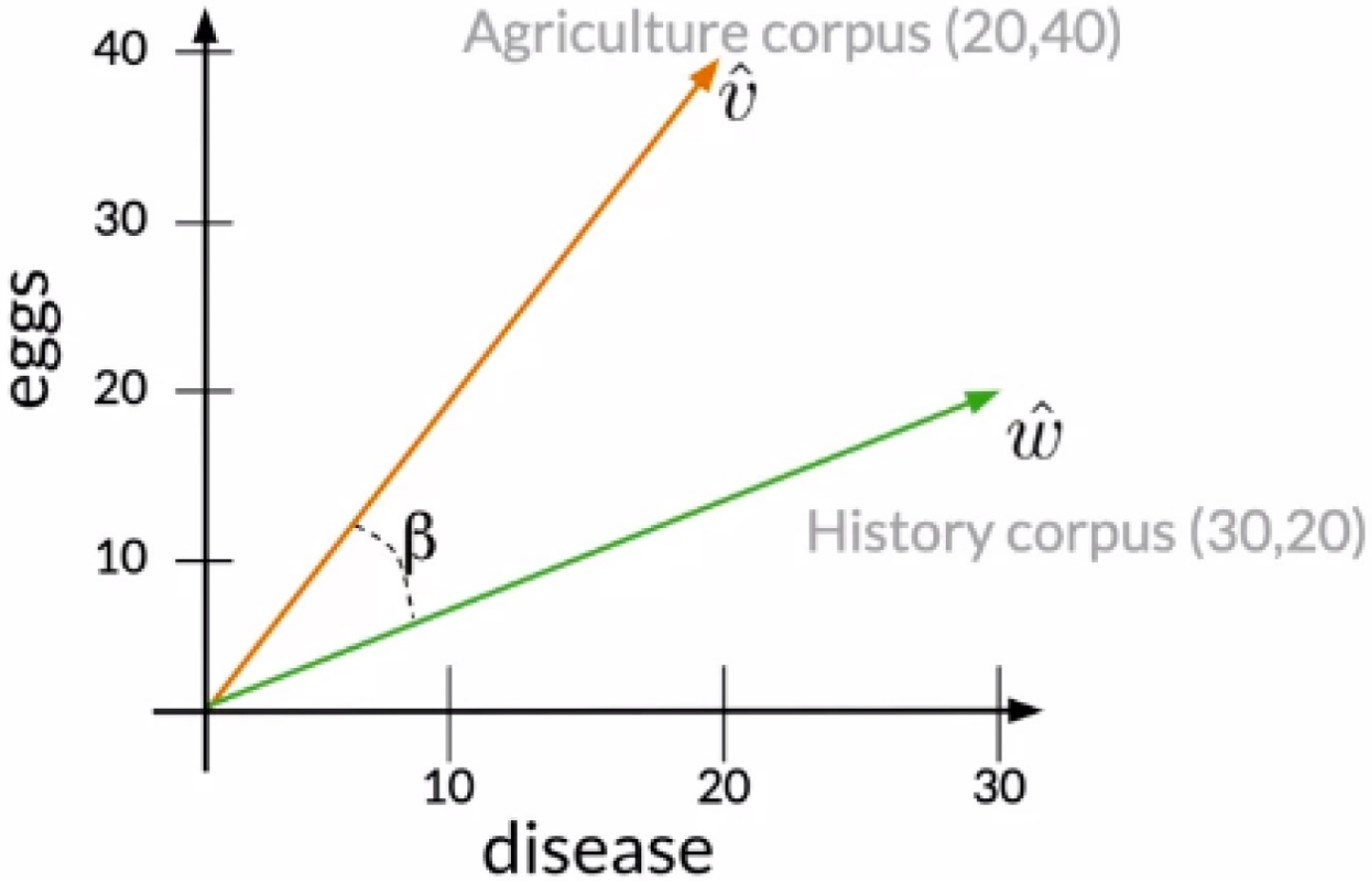
Producto punto

$$\vec{v}.\vec{w} = \sum_{i=1}^{n} v_i.w_i$$

Similitud coseno

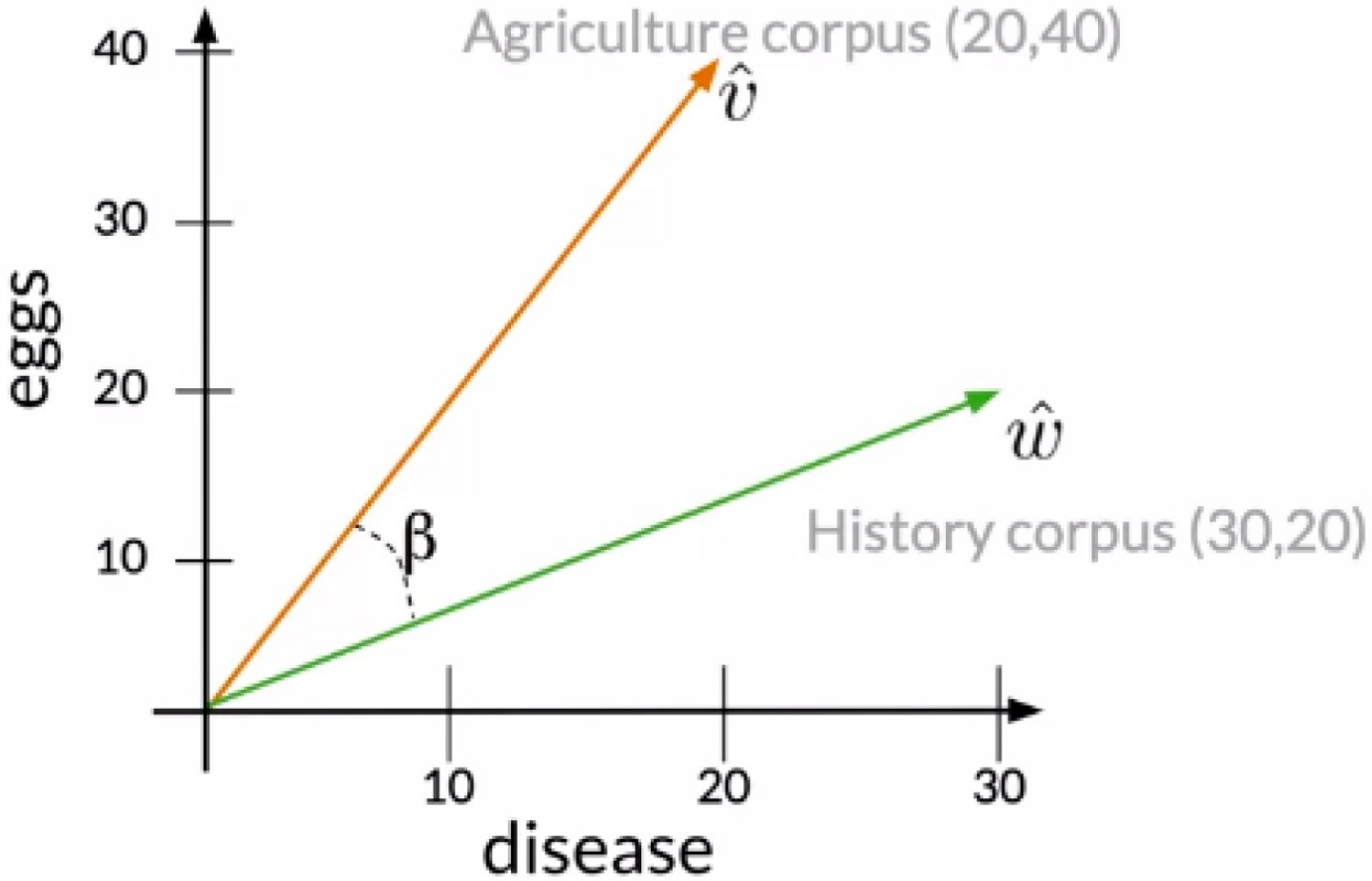


Similitud coseno



$$\hat{v} \cdot \hat{w} = \|\hat{v}\| \|\hat{w}\| \cos(\beta)$$

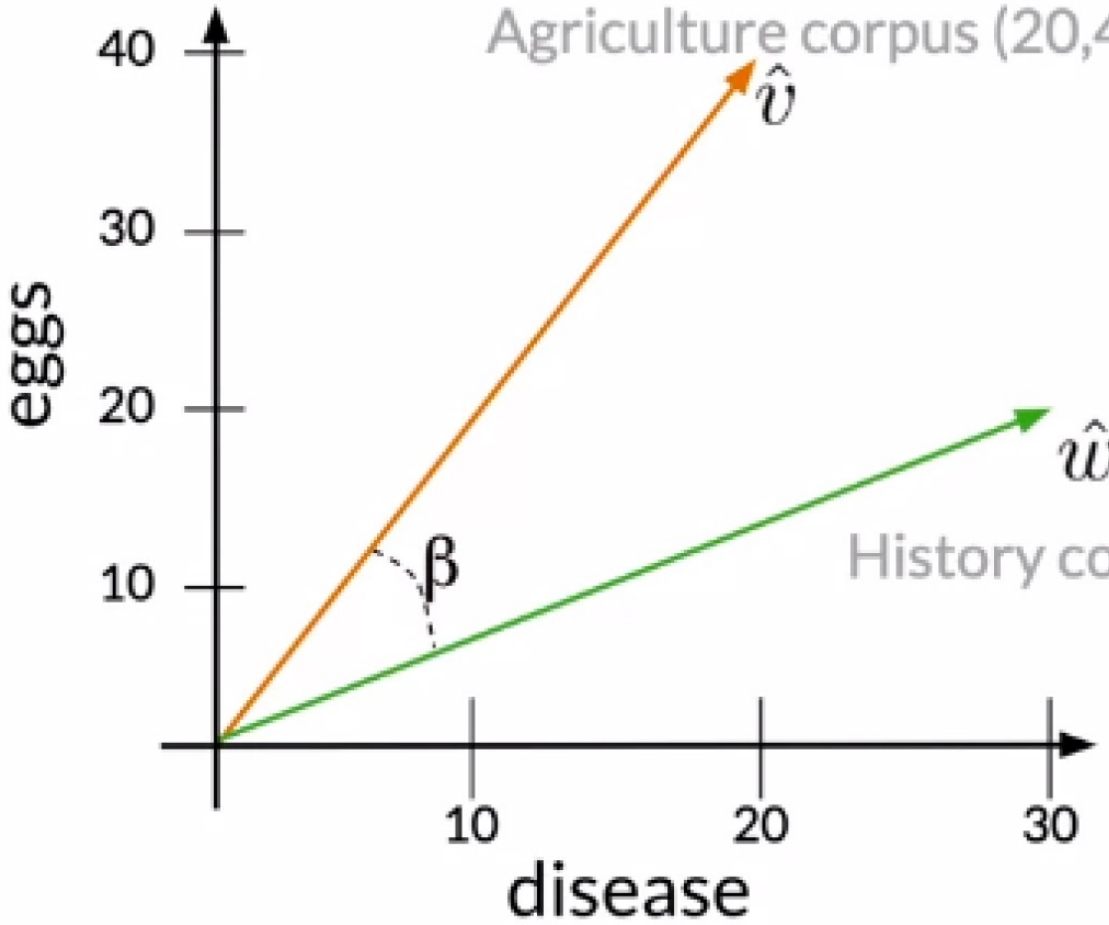
Similitud coseno



$$\hat{v} \cdot \hat{w} = \|\hat{v}\|\|\hat{w}\| \cos(\beta)$$

$$\cos(\beta) = \frac{\hat{v} \cdot \hat{w}}{\|\hat{v}\|\|\hat{w}\|}$$

Similitud coseno



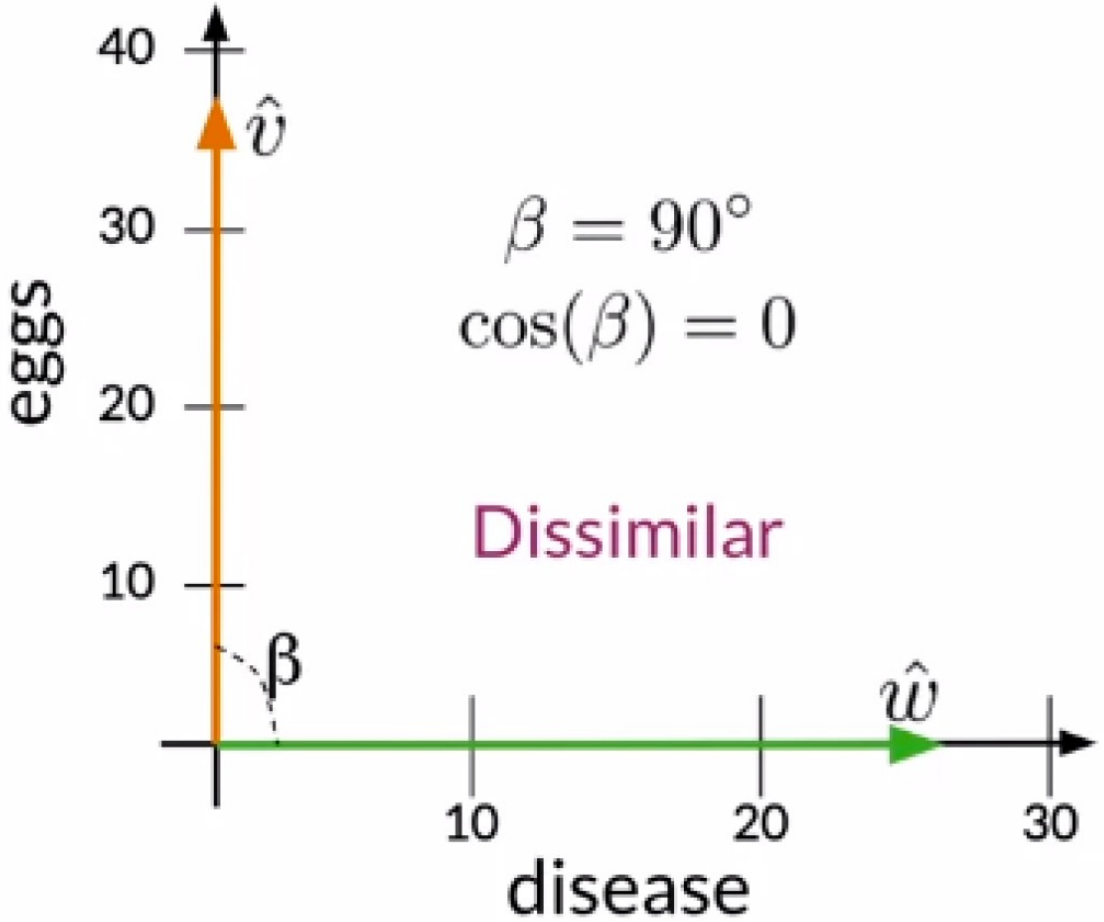
$$\hat{v} \cdot \hat{w} = \|\hat{v}\|\|\hat{w}\| \cos(\beta)$$

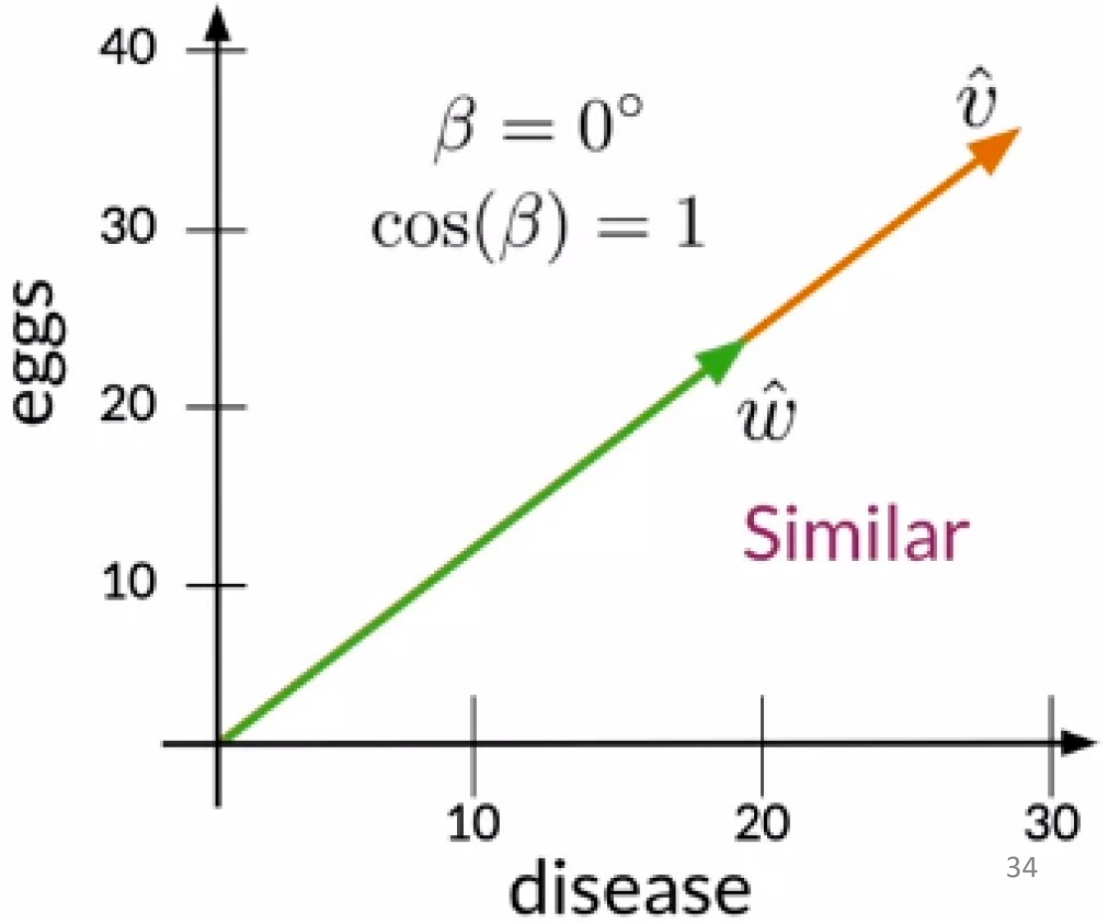
$$\cos(\beta) = \frac{\hat{v} \cdot \hat{w}}{\|\hat{v}\|\|\hat{w}\|}$$

$$= \frac{(20 \times 30) + (40 \times 20)}{\sqrt{20^2 + 40^2} \times \sqrt{30^2 + 20^2}}$$

$$= 0.87$$

Similitud coseno





Manipulando vectores de palabras

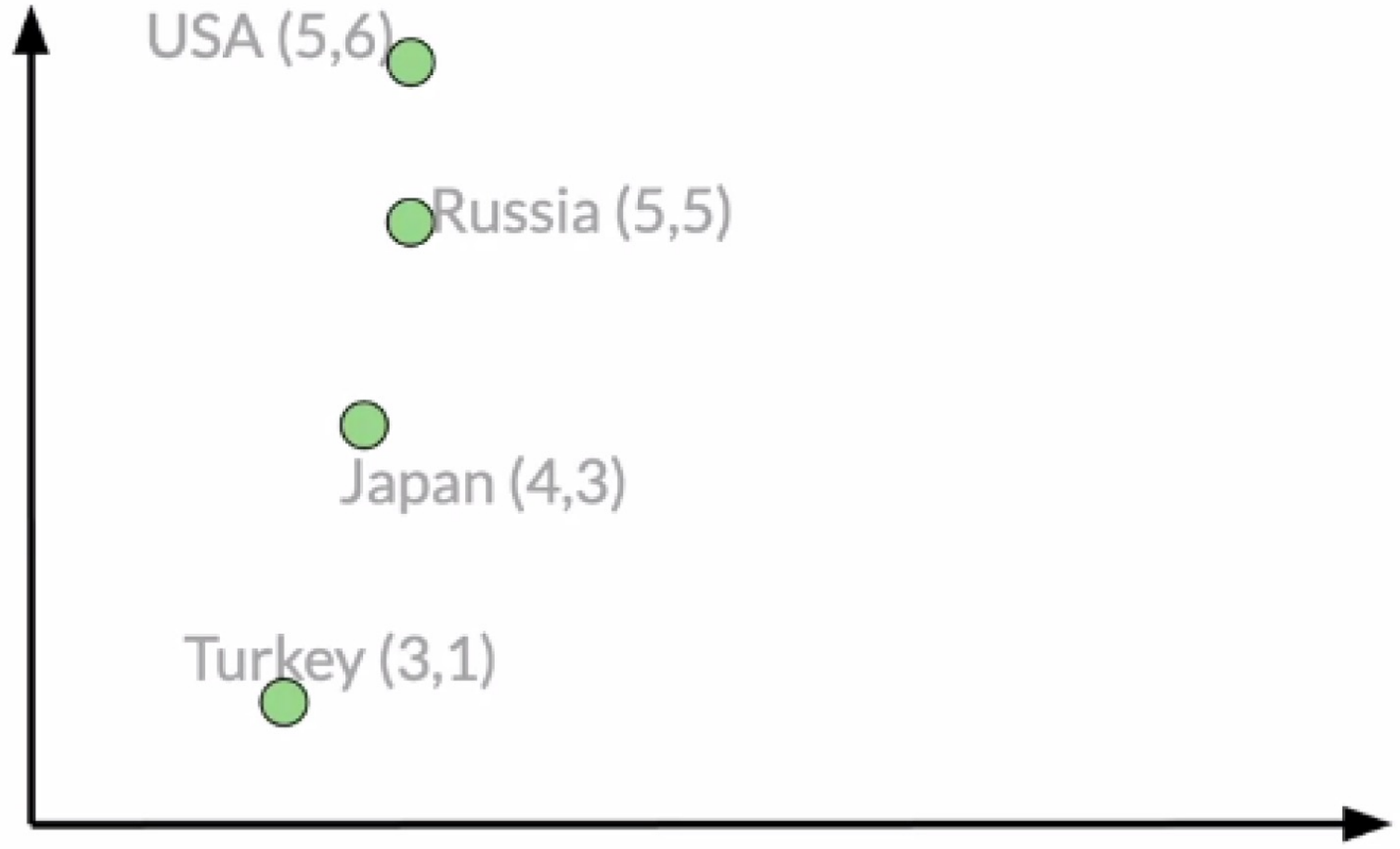
USA

Washington DC

Russia

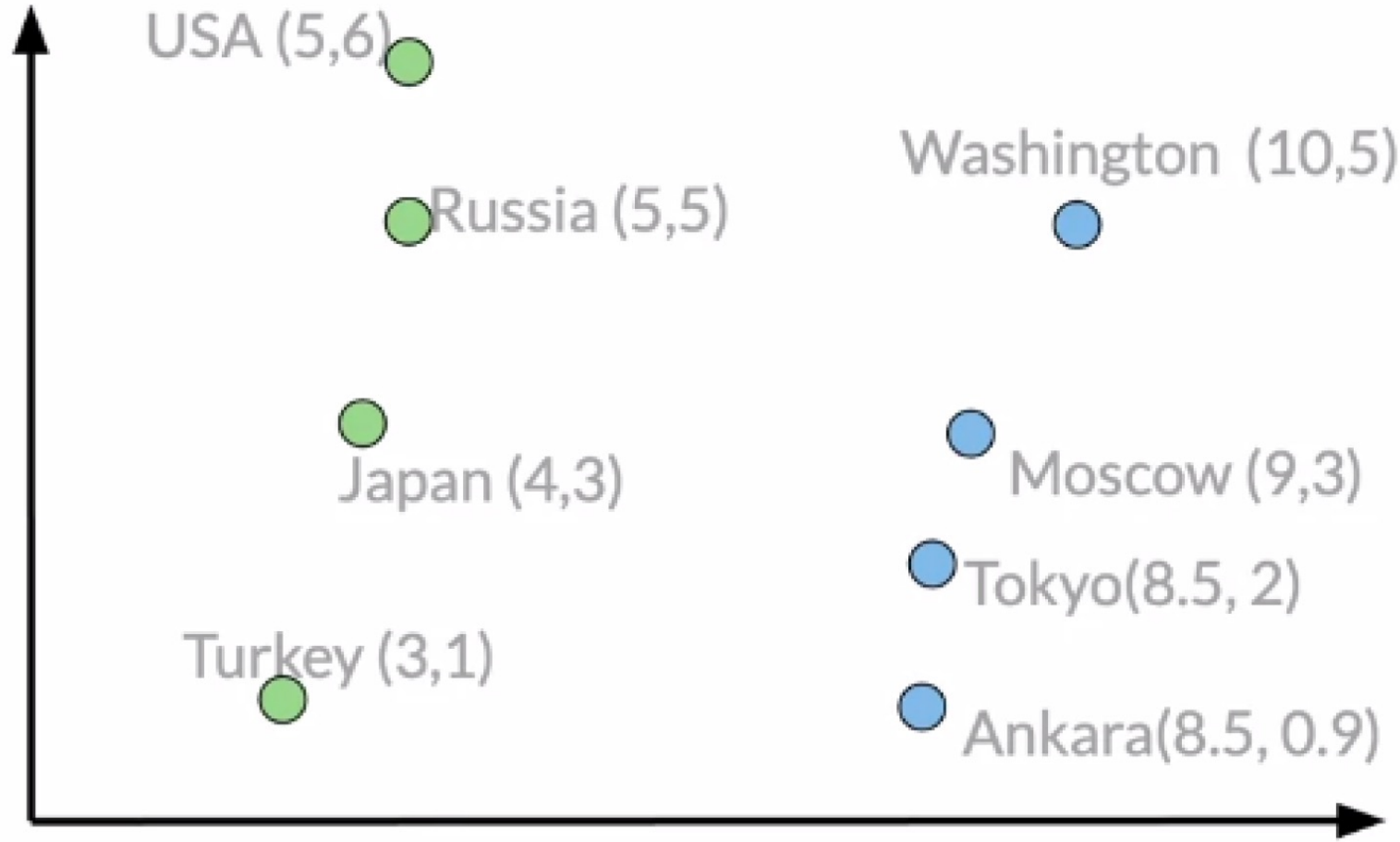
?

Manipulando vectores de palabras



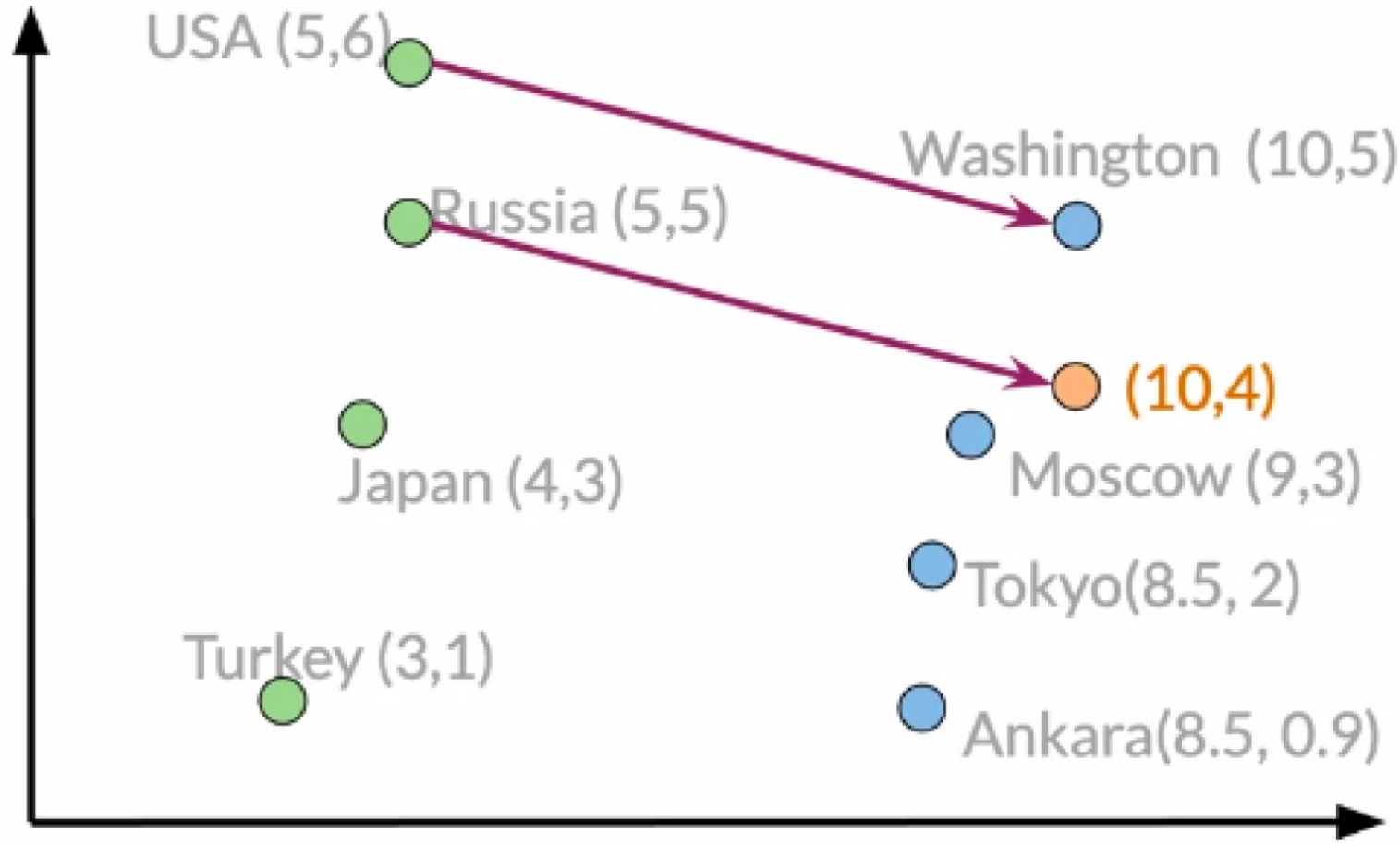
[Mikolov et al, 2013, Distributed Representations of Words and Phrases and their Compositionality]

Manipulando vectores de palabras



[Mikolov et al, 2013, Distributed Representations of Words and Phrases and their Compositionality]

Manipulando vectores de palabras

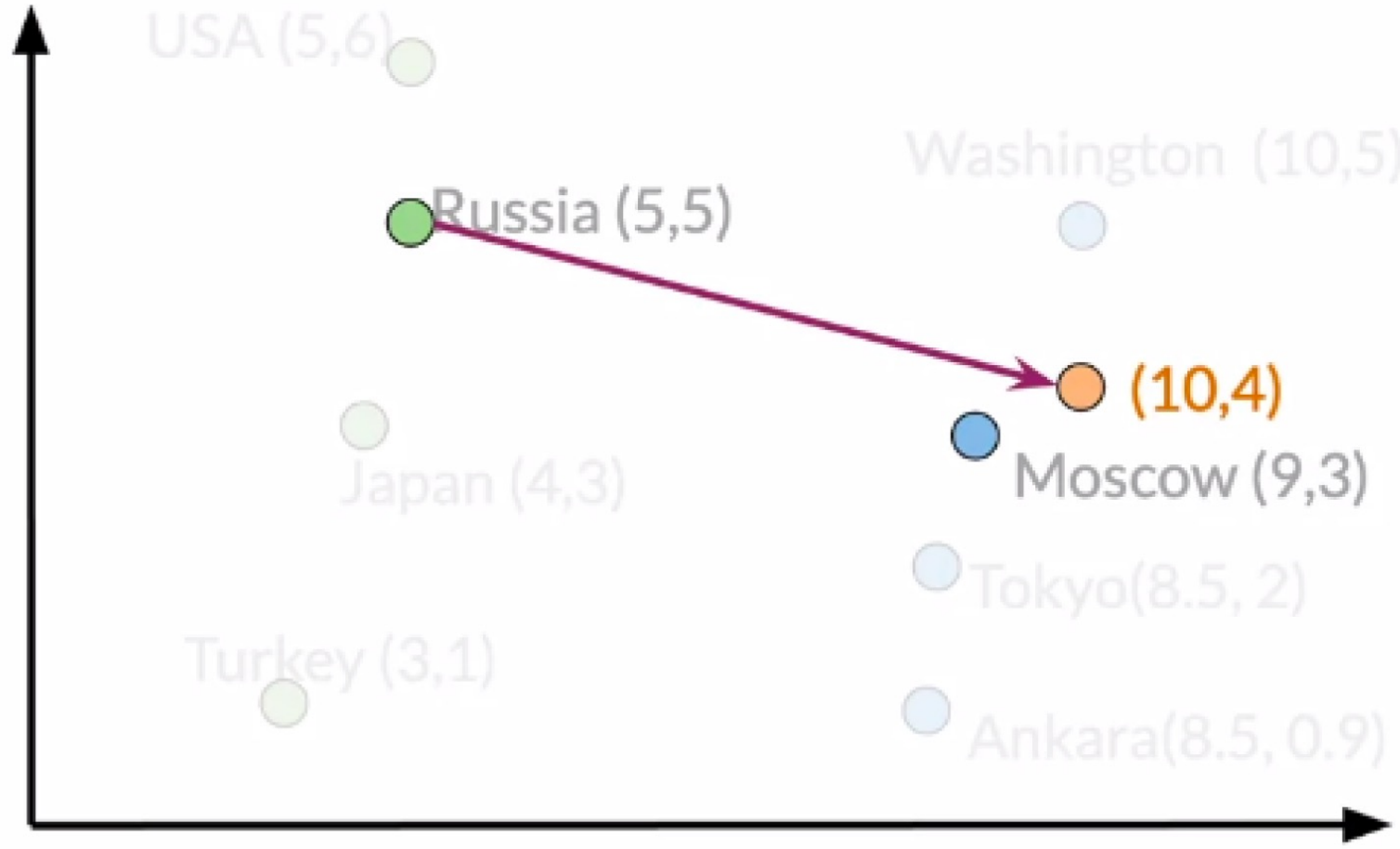


$$\text{Washington - USA} = \begin{bmatrix} 5 & -1 \end{bmatrix}$$

$$\text{Russia} + \begin{bmatrix} 5 & -1 \end{bmatrix} = \begin{bmatrix} 10 & 4 \end{bmatrix}$$

[Mikolov et al, 2013, Distributed Representations of Words and Phrases and their Compositionality]

Manipulando vectores de palabras



$$\text{Washington} - \text{USA} = \begin{bmatrix} 5 & -1 \end{bmatrix}$$

$$\text{Russia} + \begin{bmatrix} 5 & -1 \end{bmatrix} = \begin{bmatrix} 10 & 4 \end{bmatrix}$$

Moscow

[Mikolov et al, 2013, Distributed Representations of Words and Phrases and their Compositionality]

Visualización de vectores de palabras

$d > 2$

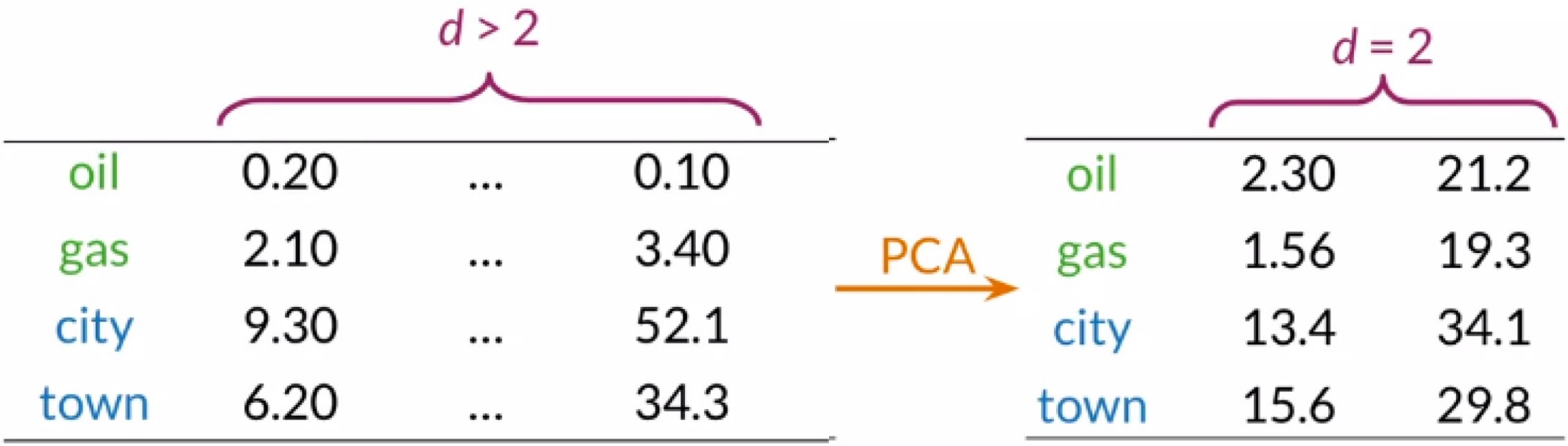
oil	0.20	...	0.10
gas	2.10	...	3.40
city	9.30	...	52.1
town	6.20	...	34.3

oil & gas

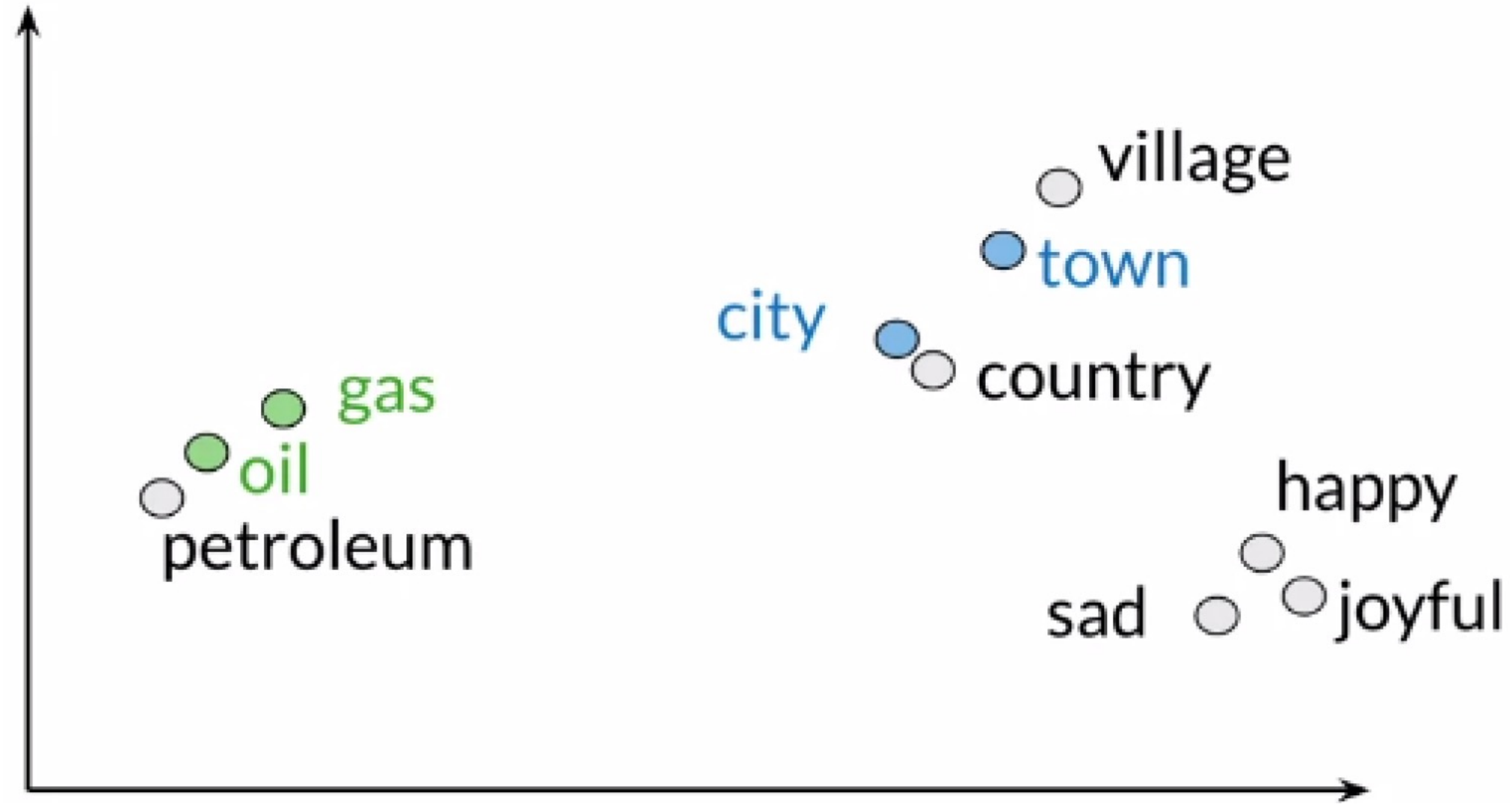
town & city

Cómo puedes visualizar si tu representación captura estas relaciones?

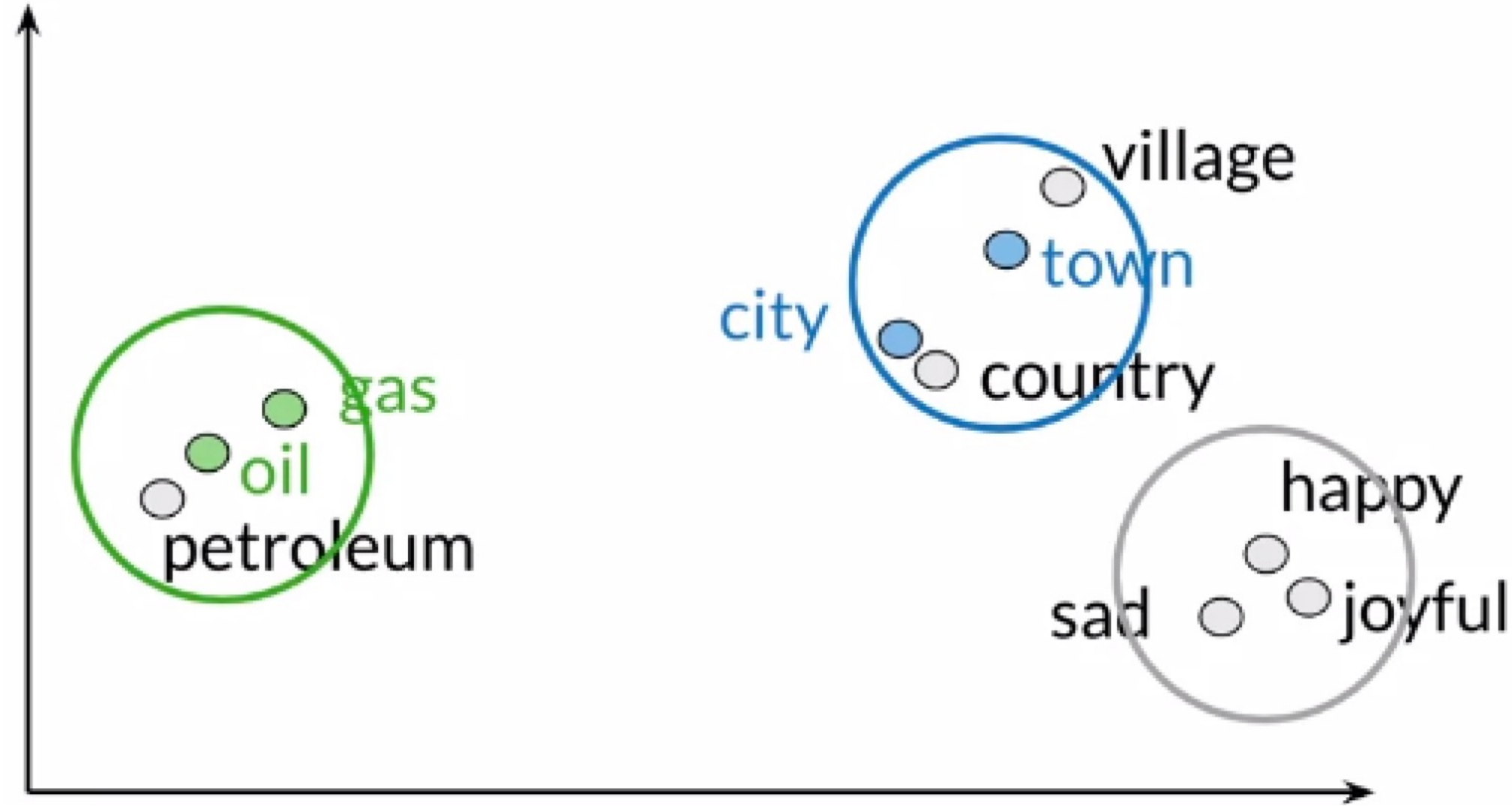
Visualización de vectores de palabras



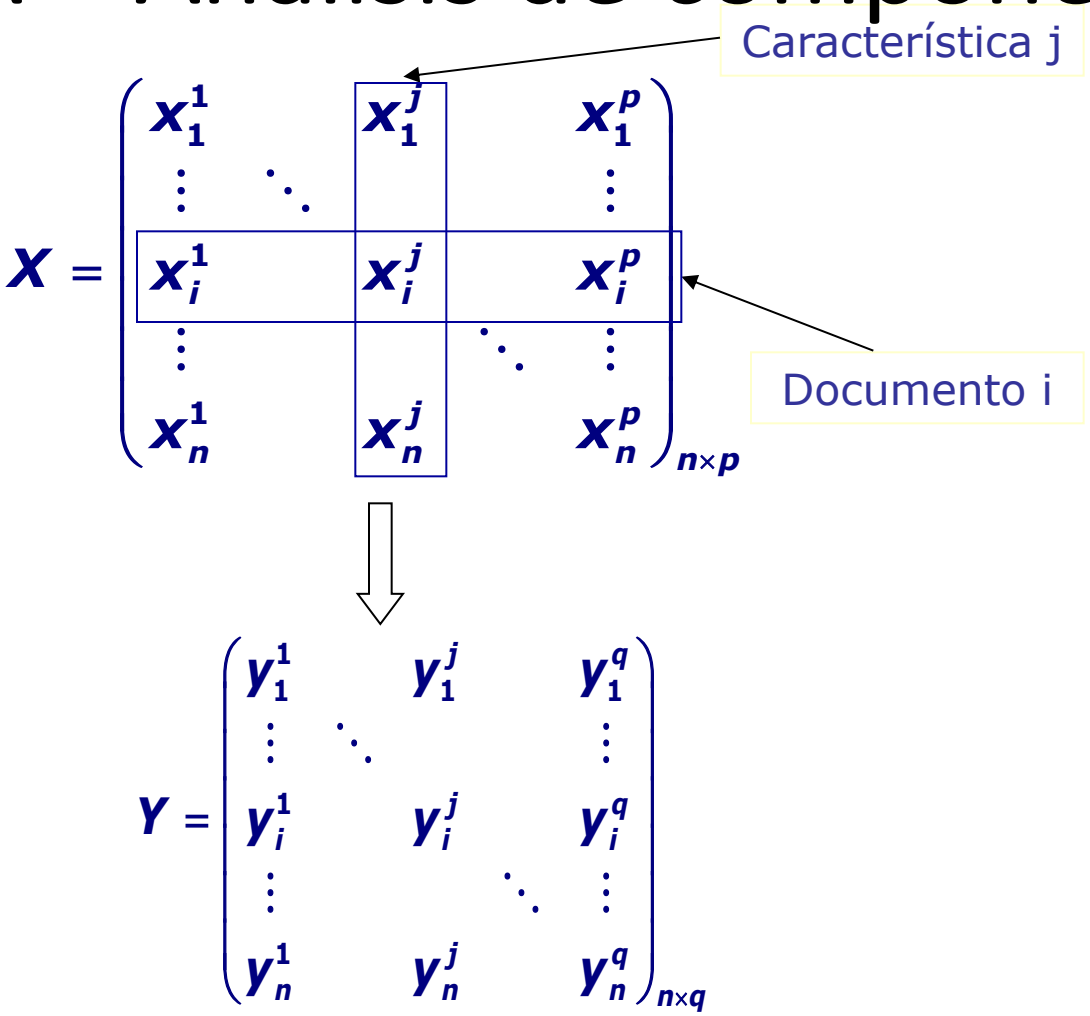
Visualización de vectores de palabras



Visualización de vectores de palabras



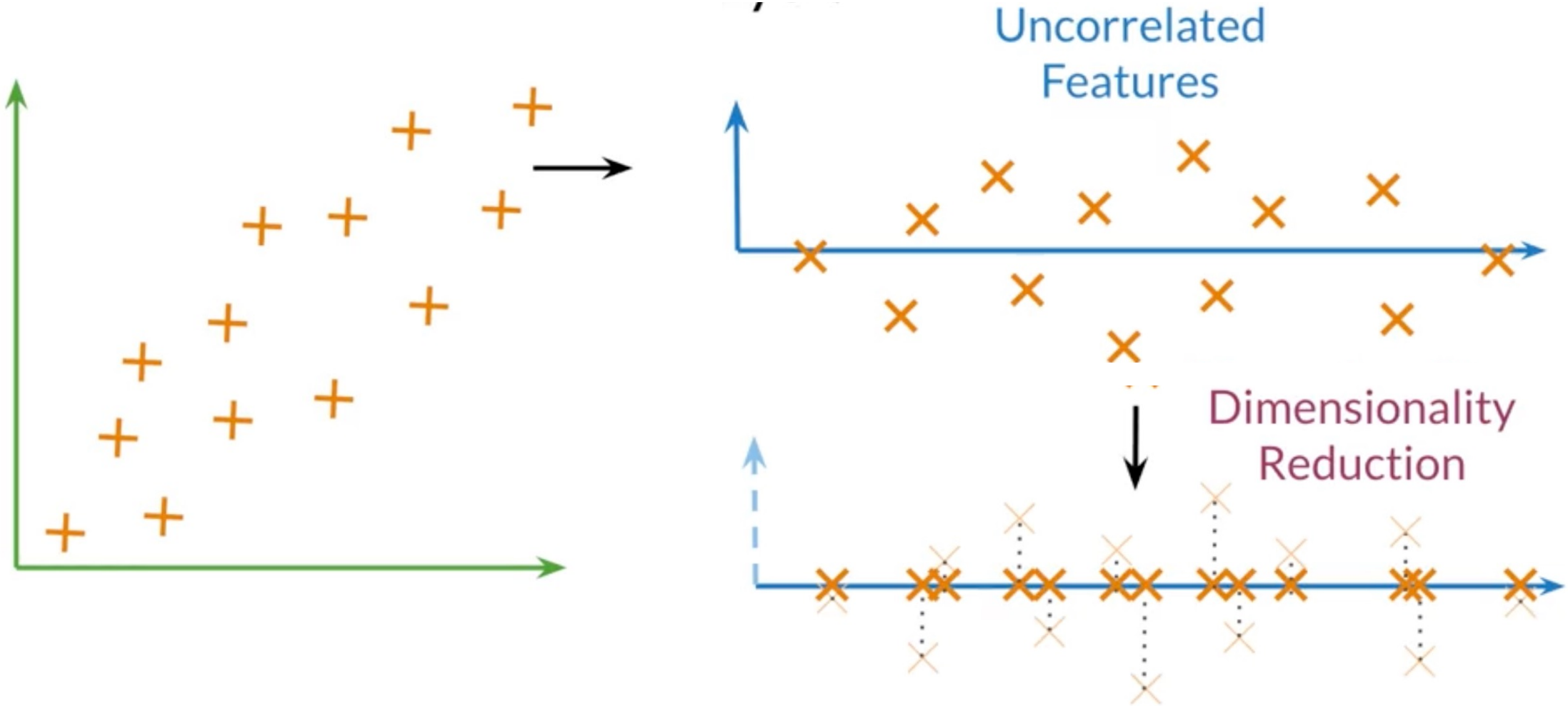
PCA – Análisis de componentes principales



Objetivo:

Definir q nuevas características, $q<p$, reteniendo el máximo de información de $\boldsymbol{X}$.

PCA – Análisis de componentes principales

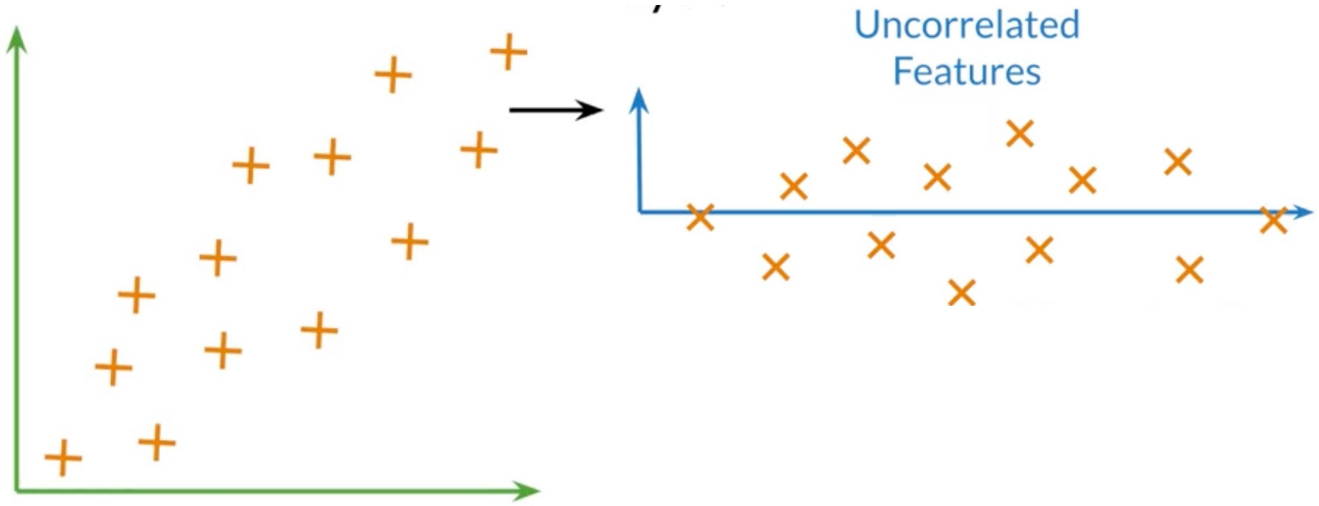


PCA – Algoritmo

- Eigenvector: Características no correlacionadas de tus datos

- Eigenvalue: La cantidad de información retenida por cada característica

PCA – Algoritmo



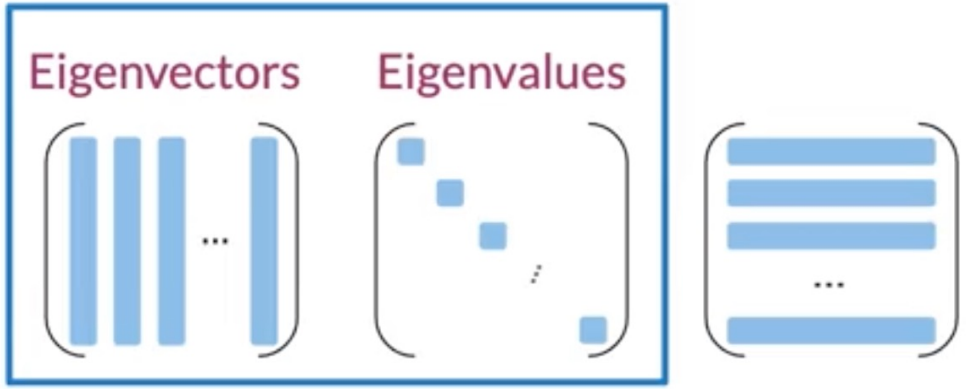
Mean Normalize Data $x_i = \frac{x_i - \mu_{x_i}}{\sigma_{x_i}}$

↓

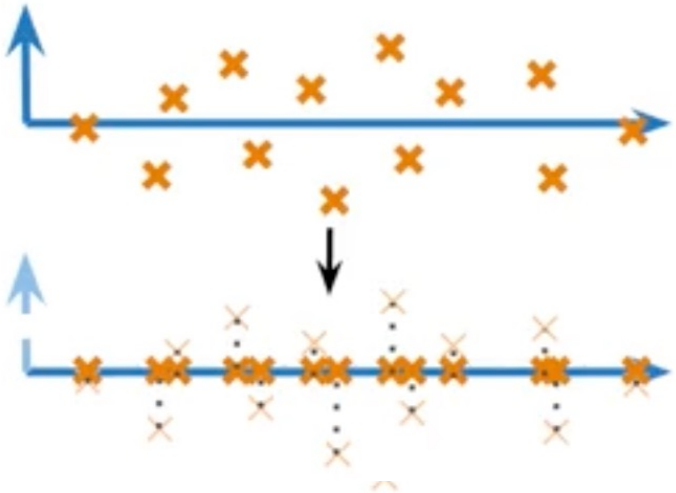
Get Covariance Matrix Σ

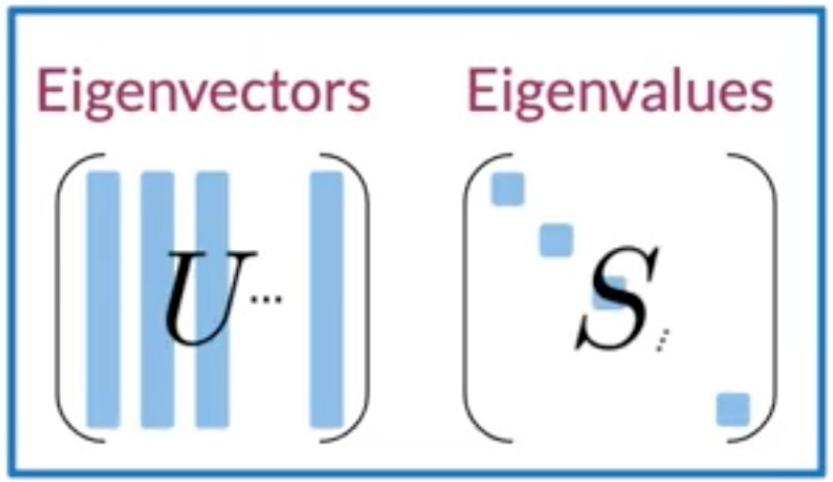
↓

Perform SVD $\mathrm{SVD}(\Sigma)$



PCA – Algoritmo





Dot Product to Project Data $X' = XU[:, 0:2]$

Percentage of Retained Variance $\frac{\sum_{i=0}^{1} S_{ii}}{\sum_{j=0}^{d} S_{jj}}$