



Detección de tópicos en un conjunto de notas del Diario La Nación.



Detección de tópicos

Objetivos: detectar de forma semi-automática los tópicos en un conjunto de notas surgidas del diario.

Presentación: análisis de métodos sobre un corpus conocido:

- Descripción vectorial de las notas (tf-idf).

Métodos de detección:

- K-means + PCA.
- Detección de comunidades en redes complejas.

Detección de tópicos

Objetivos: detectar de forma semi-automática los tópicos en un conjunto de notas surgidas del diario.

Presentación: análisis de métodos sobre un conjunto de notas conocido:

- Descripción vectorial de las notas (tf-idf).

Métodos de detección:

- K-means + PCA.
- Detección de comunidades en redes complejas.

Detección de tópicos

Objetivos: detectar de forma semi-automática los tópicos en un conjunto de notas surgidas del diario.

Presentación: análisis de métodos sobre un conjunto de notas conocido:

- **Descripción vectorial de las notas (tf-idf).**

Métodos de detección:

- **K-means + PCA.**
- **Detección de comunidades en redes complejas.**

Detección de tópicos

Objetivos: detectar de forma semi-automática los tópicos en un conjunto de notas surgidas del diario.

Presentación: análisis de métodos sobre un conjunto de notas conocido:

- **Descripción vectorial de las notas (tf-idf).**

Métodos de detección:

- K-means + PCA.
- Detección de comunidades en redes complejas.

Term frequency - Inverse document frequency **(Tf-idf)**

- **Describo las notas como vectores en un espacio multidimensional.**
- El espacio vectorial son, en principio, **todos los términos de los documentos (notas)**. Además de palabras, se pueden incluso incorporar **n-gramas** (conjunto de palabras. Ej: “casa rosada”).
- Cada documento es descripto por un vector, cuyas componentes son la **multiplicación** entre la cantidad de ocurrencias en el documento de un dado término (**tf**), multiplicado por una valorización (**idf**).
Idf(t) cuantifica qué tan específico es un término en un conjunto de documentos.

Term frequency - Inverse document frequency (Tf-idf)

Vector
documento

$$v = [\dots, tf(t) \cdot idf(t), \dots]$$

Cantidad de veces que aparece el término t en el documento.

$$tf(t) = 1 + \log \left(\frac{1+N}{1+n_t} \right)$$

Para n_t más grandes, $idf(t)$ es más chico → términos más específicos de un documento tienen mayor valorización.

N : # total de documentos.

n_t : # documentos donde aparece el término t .

Ejemplo

Documentos:

- 1) La casa de Julia
- 2) La casa de Cristian
- 3) De la casa de Seba

Espacio de 6 dimensiones.

julia

seba

cristian

la casa

de

$$idf = 1 + \log\left(\frac{1+3}{1+1}\right) = 1$$

$$idf = 1 + \log\left(\frac{1+3}{1+1}\right) = 1.69$$

$$v_3 = [1, 1, 2, 1.69, 0, 0]$$

“de” es importante porque **es más frecuente** en el documento.
“seba” es importante por **específico** del documento.

Conjunto de notas →
Matriz documentos x términos

F términos

$$M = \begin{matrix} & \text{\color{blue} N documentos} \\ \text{\color{black} N} & \end{matrix}$$

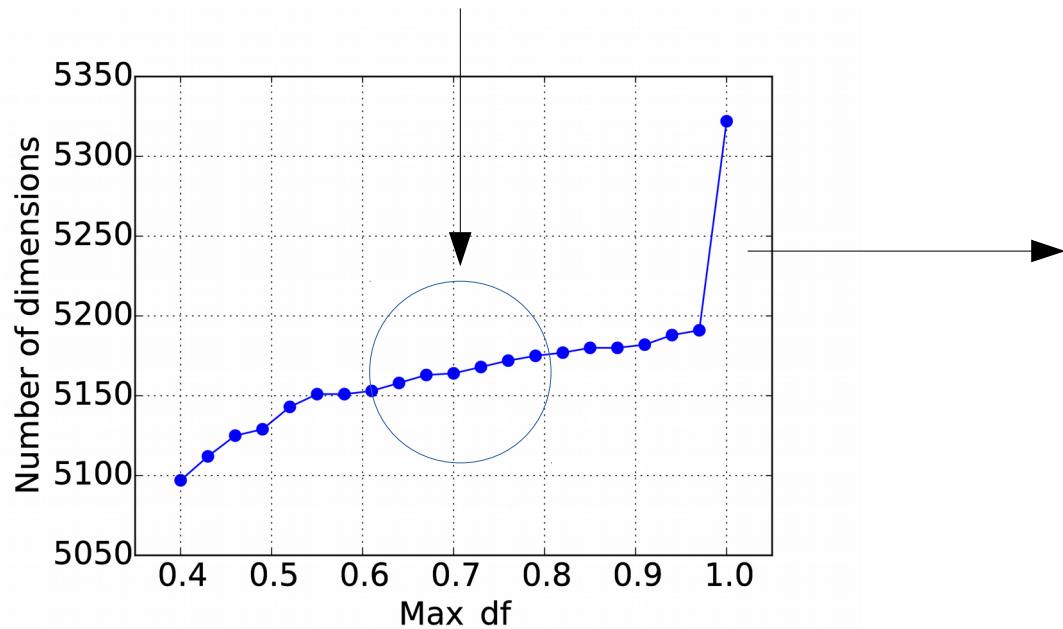
V_{11}	V_{12}	V_{13}	...	V_{1F}
...
...
...
V_{N1}	V_{N2}	V_{N3}	...	V_{NF}

$$\text{Dim}(M) = N \times F$$

Term frequency - Inverse document frequency (Tf-idf)

Parámetros que se pueden regular y valores escogidos:

- **Max_df: descartar términos por muy occurrentes:** Podemos fijar un límite superior en la cual si un término aparece en más documentos que ese límite lo descartamos. Tomo los **términos que aparezcan en menos del 70% de los documentos**.



Solo se observa un **cambio abrupto** en la dimensión del espacio **cuando se descartan los términos que aparecen** en prácticamente **todos los documentos**.

Term frequency - Inverse document frequency **(Tf-idf)**

Parámetros que se pueden regular y valores escogidos:

- **Min_df: descartar términos por raros:** Podemos fijar un límite inferior, es decir, solo tomamos términos que aparezcan en más de un cierto dado de documentos. Tomo los **términos que aparezcan en dos o más documentos**.
- **N-gram_range:** podemos tomar, además de palabras sueltas, bigramas, trigramas, etc. **Tomo de 1 a 3 gramas.**
- **Los vectores pueden estar o no normalizados:** los vectores documentos pueden estar normalizados a norma 1 → vectores dentro de la esfera unitaria multidimensional, o bien no estarlo. En principio la mayor parte de los cálculos lo hago con vectores normalizados.

Descripción de corpus de prueba

Documentos = 31 notas del Diario La Nación donde claramente podemos identificar **4 tópicos**:

- 10 notas sobre ***Tarifas***.
- 7 notas sobre ***Tratado de paz en Colombia***.
- 7 notas sobre ***Campaña de Trump***.
- 7 notas sobre ***Acuerdo Malvinas***.

Las notas están descriptas en un espacio de **5164 dimensiones**.

Recordar: esto surge de tomar todos los **1-gramas, 2-gramas y 3-gramas**, y **descartar** aquellos que aparezcan en más del 70% de las notas y **en menos de dos notas** (es decir aquellos que aparezcan en solo una nota).

Detección de tópicos

Objetivos: detectar de forma semi-automática los tópicos en un conjunto de notas surgidas del diario.

Presentación: análisis de métodos sobre un conjunto de notas conocido:

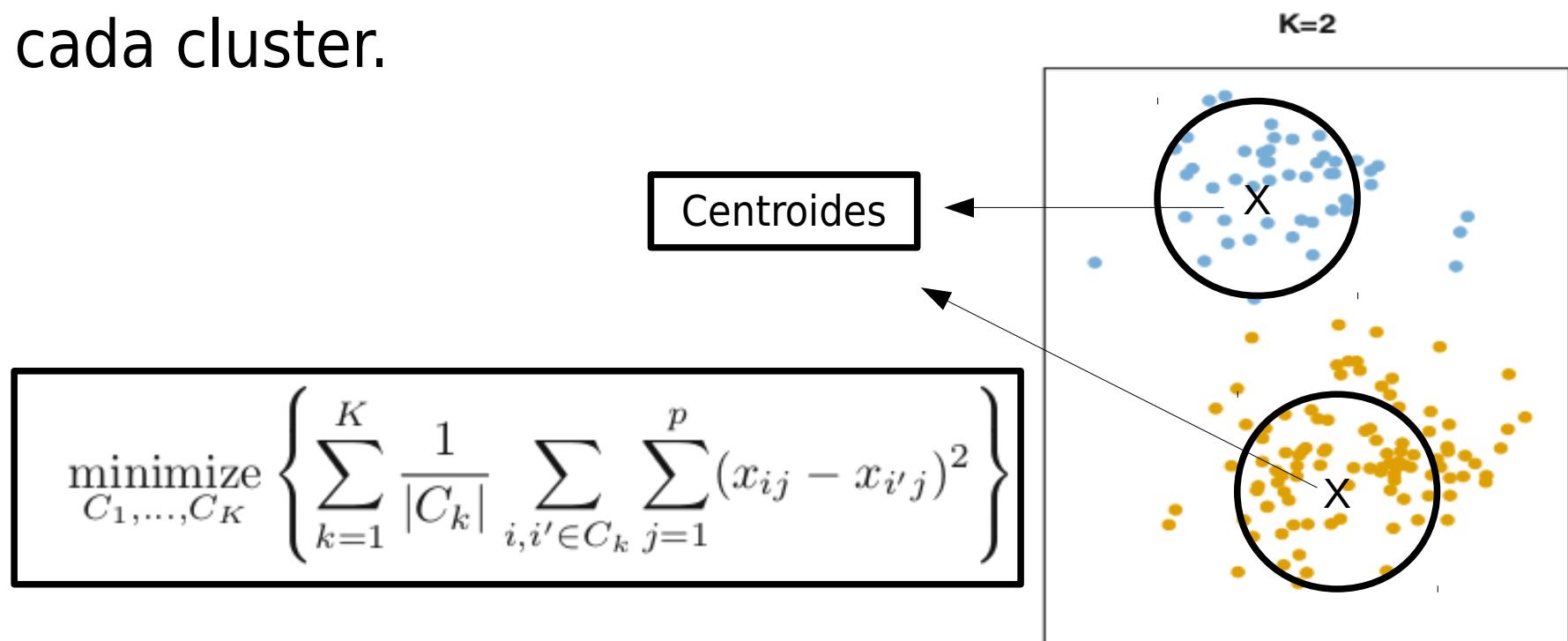
- Descripción vectorial de las notas (tf-idf).

Métodos de detección:

- **K-means + PCA.**
- Detección de comunidades en redes complejas.

K-means

- Busca la mejor partición del espacio multidimensional basado en **minimizar la suma de las varianzas** de cada cluster.



- La **cantidad de particiones K es un parámetro del algoritmo** → se debe buscar un criterio para elegir la mejor.

Evaluación de las particiones

- Coeficiente de silhouette de un punto i :

Distancia media entre el punto i y los puntos del cluster más cercano (mínima distancia media).

Distancia media entre el punto i y los puntos dentro del mismo cluster.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$-1 < s(i) < 1$

Un $s(i)$ cercano a 1 → punto i más cercano a los puntos de su mismo cluster, y/o más alejado de los puntos de los clusters vecinos.

Para evaluar las particiones se toma el valor medio de s .
El número de particiones van de 2 a (N° puntos - 1).

K-means sobre la base de datos de prueba

Particiones propuestas ordenados según el silhouette:

K (#clusters)	Silhouette	#clusters esperados
4	0.13	→
5	0.12	
10	0.12	
9	0.11	
8	0.11	

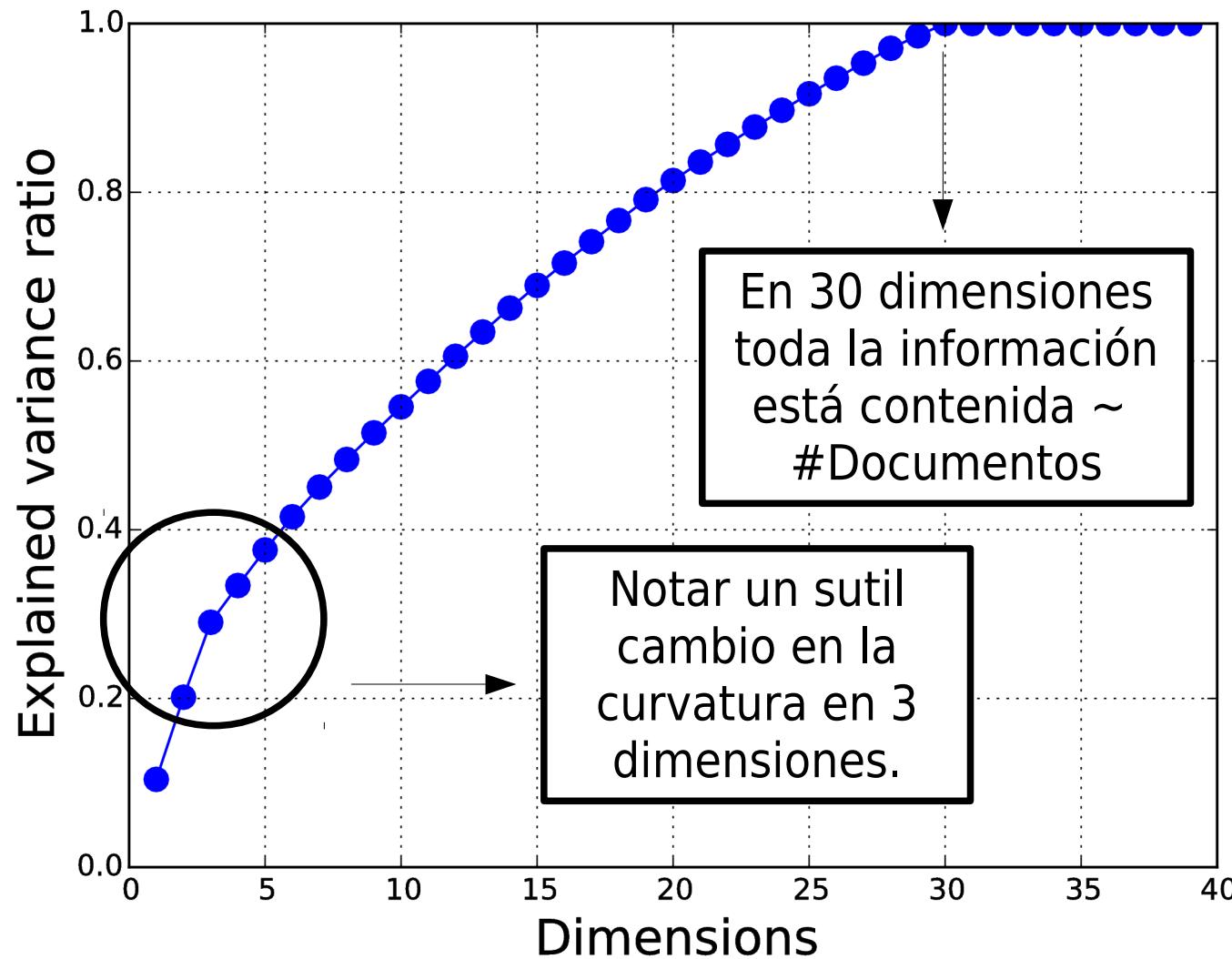
El algoritmo encuentra que la mejor partición coincide con los tópicos esperados.

Sin embargo el coeficiente de silhouette parece ser muy bajo.

K-means + reducción de la dimensionalidad (PCA)

- **PCA** (*principal components analysis*) busca **direcciones** en el espacio multidimensional que mejor expliquen la variabilidad de los datos.
- Las nuevas direcciones son una **combinación lineal de los términos** que describen el espacio original.
- Los **vectores documentos son proyectados** a un espacio de $D < F$ dimensiones.

Información contenida



Fracción de varianza explicada en función de la cantidad de dimensiones utilizadas para explicar los datos. Con 3 dimensiones se explica ~ 33% de la información.

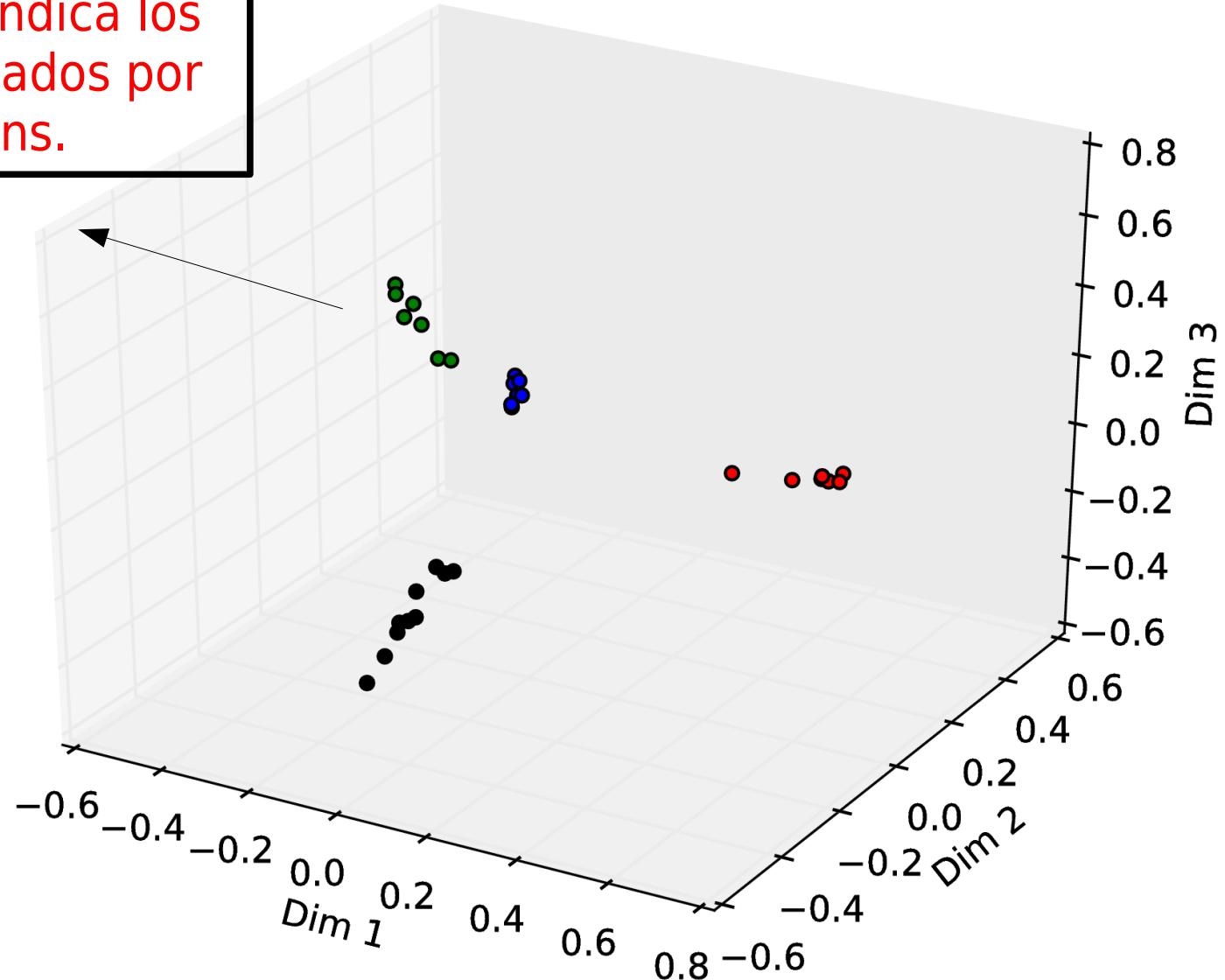
K-means + reducción de la dimensionalidad (PCA)

- Barrido en cantidad de dimensiones y número de clusters propuestos:

#Dimensiones	K (#clusters)	Silhouette
3	4	0.86
2	3	0.84
1	2	0.83
1	3	0.78
1	4	0.77

Aumenta considerablemente el coeficiente de silhouette.

Cada color indica los clusters hallados por K-means.



Proyección en el espacio tridimensional

Alternativa: Reducción de dimensionalidad **NMF**

Non-negative matrix factorization: otra técnica de reducción de dimensionalidad, descompone en forma aproximada una matriz con componentes no-negativos como la **multiplicación de dos matrices con componentes no-negativos.**

$$M \approx H \cdot W$$

Matriz de documentos
por términos
(tf-idf)
Dim(M) = N x F

Matriz de documentos
por “factores”
Dim(M) = N x D

Matriz de “factores”
por términos
Dim(M) = D x F

D es un parámetro a elegir

Reducción de dimensionalidad **NMF**

Ventajas:

- La interpretación puede ser más natural que en **PCA**, ya que todos los documentos están representados con vectores de componentes positivas.

Desventajas:

- Es un método aproximado.
- Hay que elegir D con un criterio.

$$M \approx H \cdot W$$



Matriz de documentos
reducida

K-means + reducción de la dimensionalidad (**NMF**)

- Barrido en cantidad de dimensiones y número de clusters propuestos:

#Dimensiones	K (#clusters)	Silhouette
2	2	0.97
3	3	0.96
4	4	0.96
4	5	0.90
5	5	0.88

NMF
requiere de
una
dimensión
más que
PCA

Los **4 clusters** esperados **aparecen bien rankeados**, aunque no está tan claro cuál es la mejor partición.

Interpretación NMF

Las dimensiones surgidas de **NMF** son vectores con componentes no negativos, representados en el espacio surgido de **tfidf**.

Componentes con mayor peso, ordenados de mayor a menor:

- **Dimensión 1:** *farc, las farc, paz, acuerdo, colombia, santos, la paz, guerrilla, de las farc, colombiano.*
- **Dimensión 2:** *trump, clinton, hillary, campaña, republicano, de trump, ryan, magnate, demócrata, hillary clinton.*

Interpretación NMF

Las dimensiones surgidas de **NMF** son vectores con componentes no negativos, representados en el espacio surgido de **tfidf**.

Componentes con mayor peso, ordenados de mayor a menor:

- **Dimensión 3:** *tarifas, corte, la corte, gas, aumentos, los aumentos, aumento, luz, fallo, las tarifas.*
- **Dimensión 4:** *malvinas, islas, las islas, argentina, malcorra, la argentina, vuelos, macri, soberanía, acuerdo.*

Conclusiones K-means

- *El enfoque correcto parece ser K-means + PCA:* la **reducción de la dimensionalidad** aporta cierto grado de **abstracción** necesaria para detectar tópicos.
- *El enfoque con NMF también resulta en un buen método y la interpretación de los resultados es más natural.*

Pérdida de información →
Abstracción

Detección de tópicos

Objetivos: detectar de forma semi-automática los tópicos en un conjunto de notas surgidas del diario.

Presentación: análisis de métodos sobre un conjunto de notas conocido:

- Descripción vectorial de las notas (tf-idf).

Métodos de detección:

- K-means + PCA.
- **Detección de comunidades en redes complejas.**

Enfoque redes complejas

- Se interpreta cada documento como un nodo en una red compleja.
- Podemos construir una red pesada, definiendo los pesos entre los nodos i y j como:

$$w_{ij} = \vec{v}_i \cdot \vec{v}_j = \cos(\theta)$$

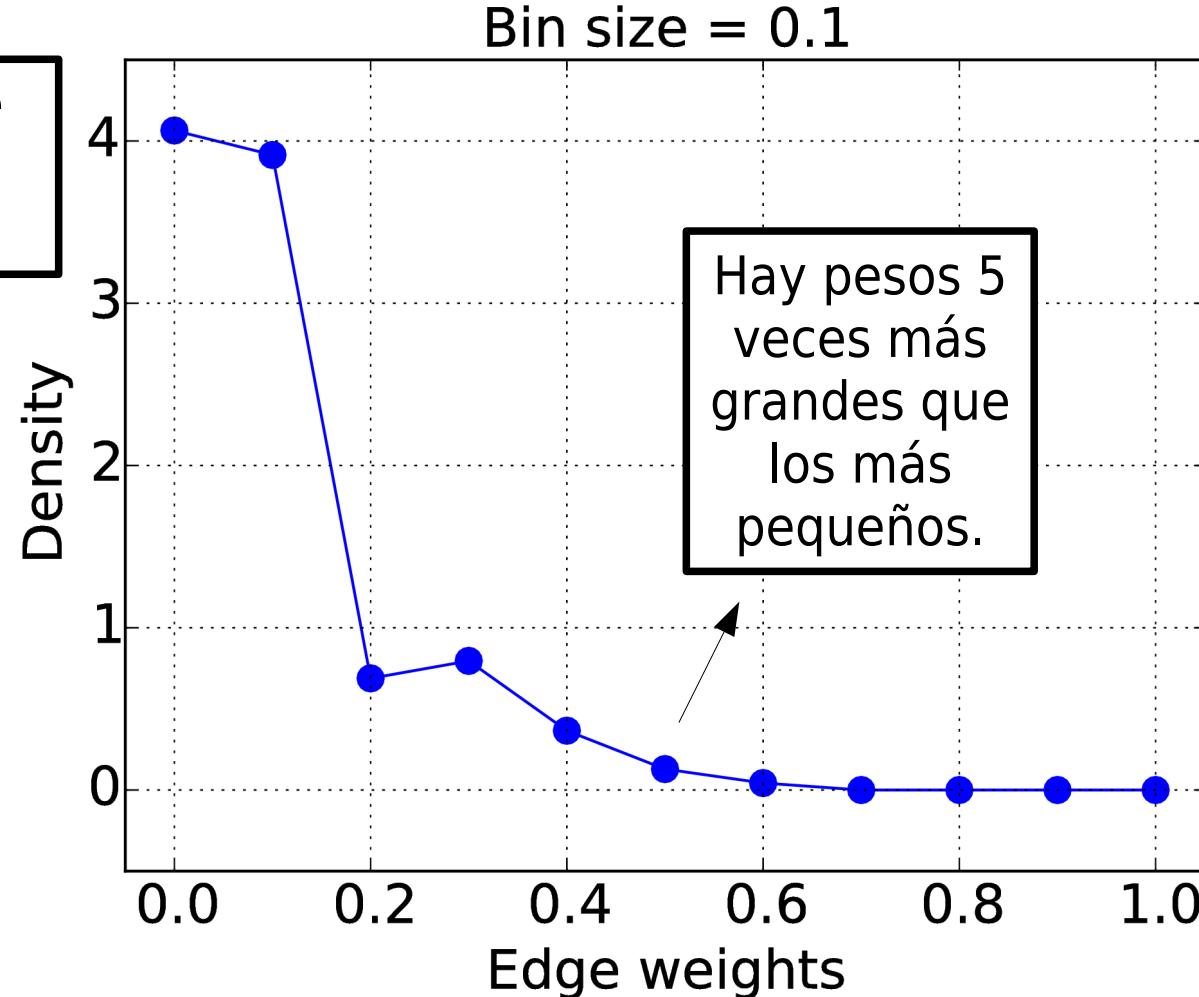
Vectores
documentos con
norma 1

Medida de
similaridad

$0 < w_{ij} < 1$

Redes sobre la base de datos de prueba

La mayoría de los pesos son cercanos a 0.



Histograma de los pesos de los enlaces en la red

Evaluación de las particiones

- Modularidad (hasta que encuentre la implementación de otro coeficiente...):

Matriz de adyacencia pesada.

Suma de los pesos sobre el nodo i

N.^º de enlaces en la red.

Etiquetas de los comunas.

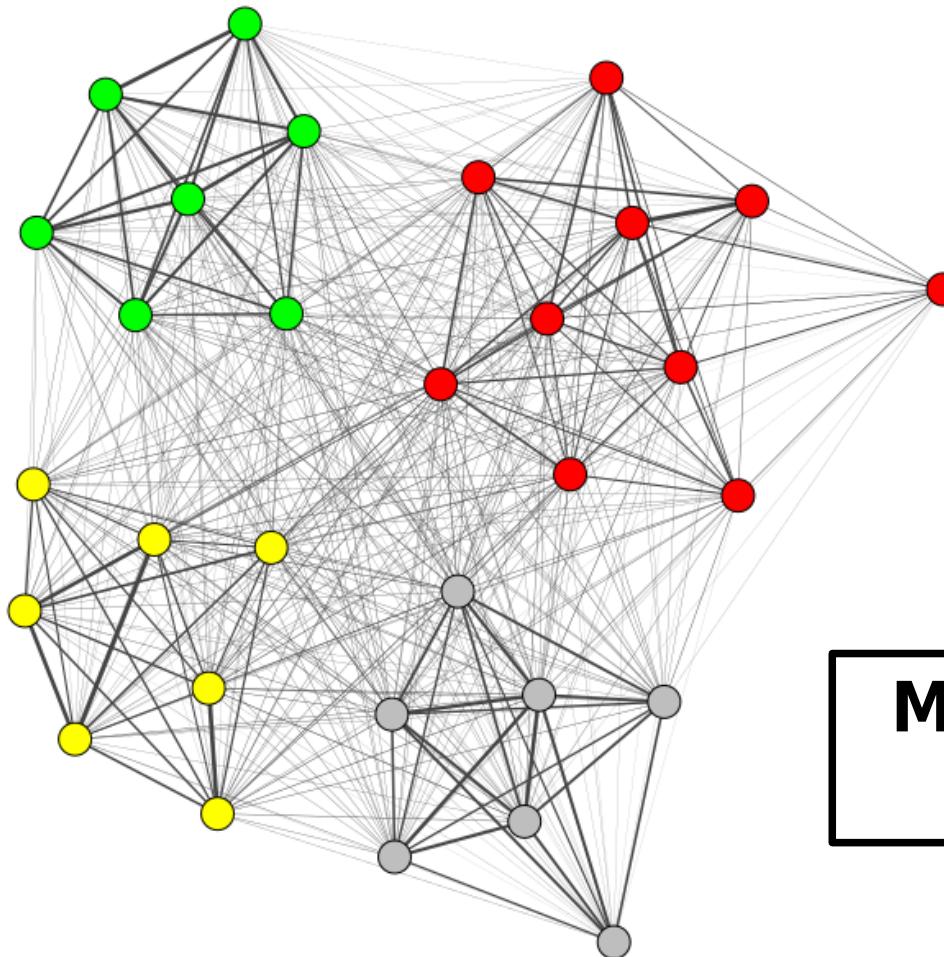
$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

$$-1 < Q < 1$$

Layout pesado

Espesor de los enlaces proporcional al peso.

Colores → comunas detectadas

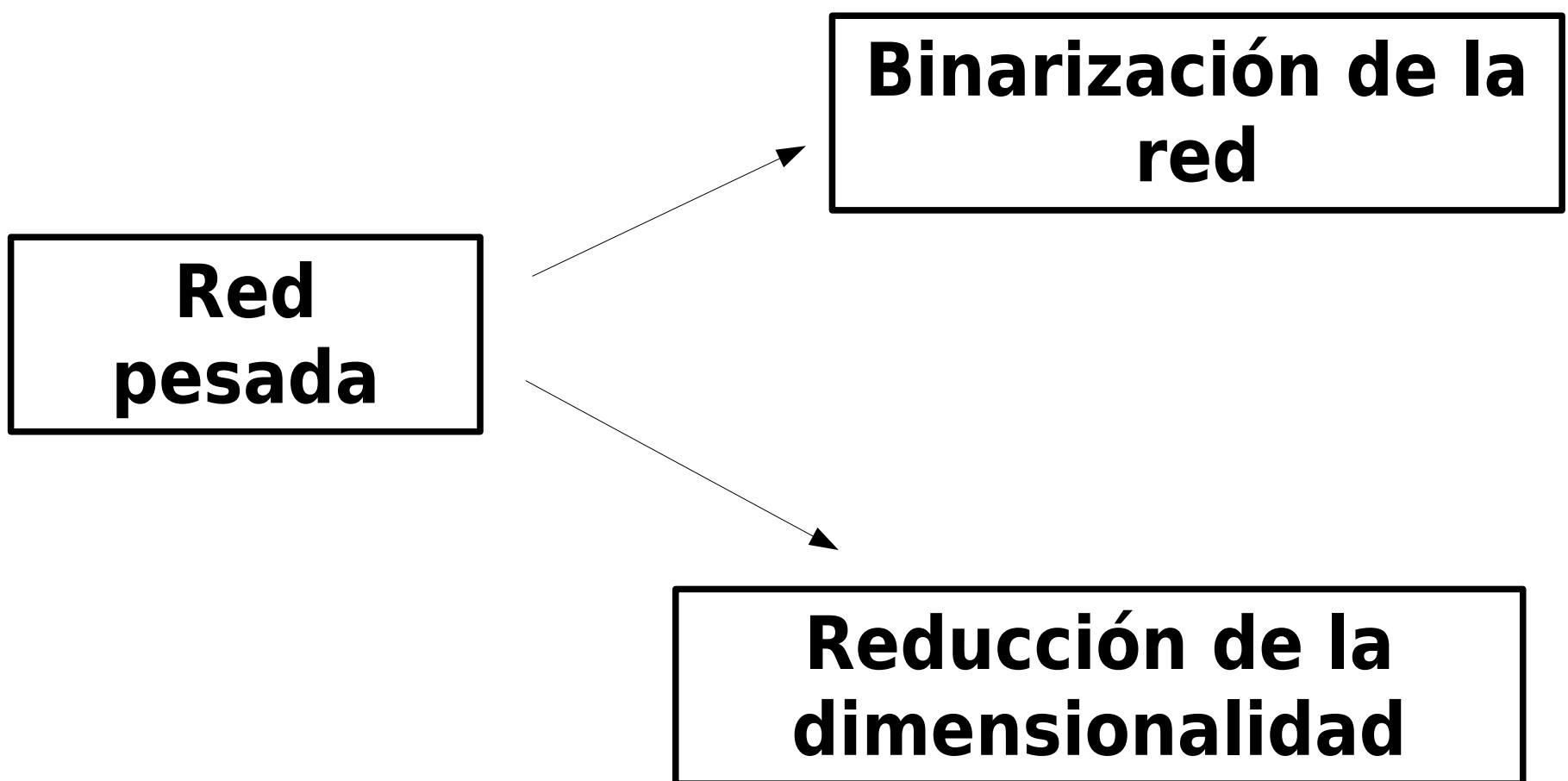


Modularidad = 0.374

- Es robusto ante el cambio de algoritmo: *infomap*, *fastgreedy*, y *label propagation*.
- Falla con *edgebetweenness*: detecta un único cluster.

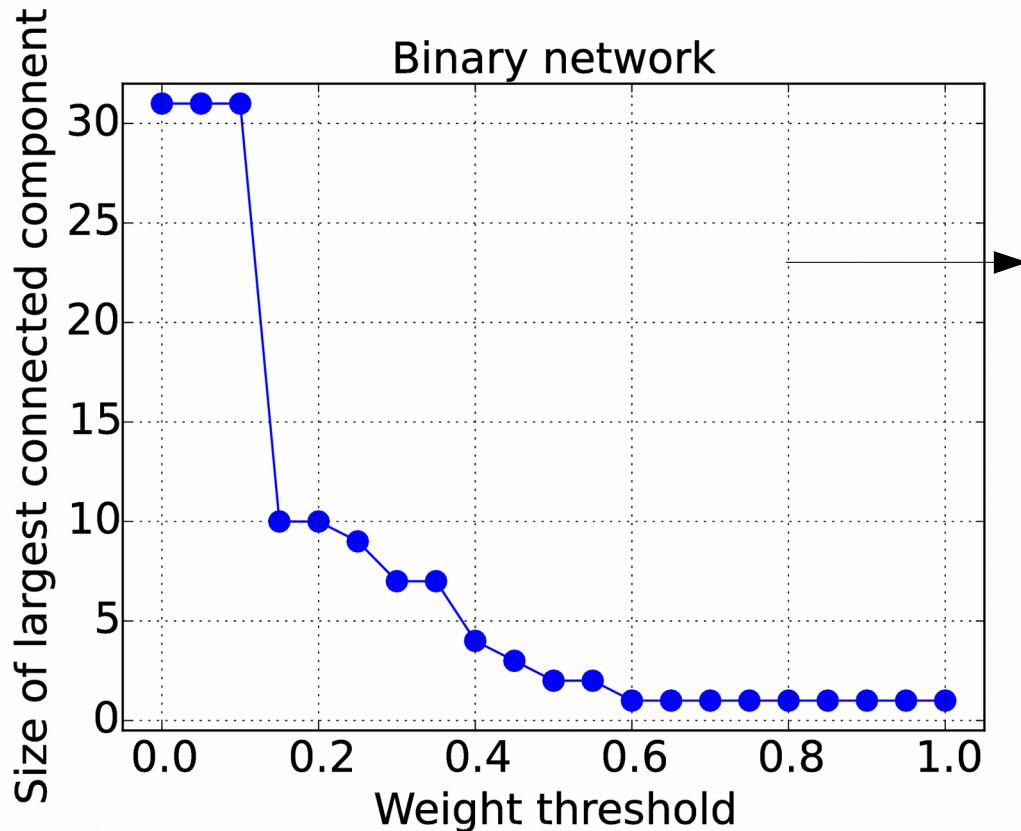
Podemos mejorar la formación de comunidades?

Dos posibles caminos:



Binarización de la red

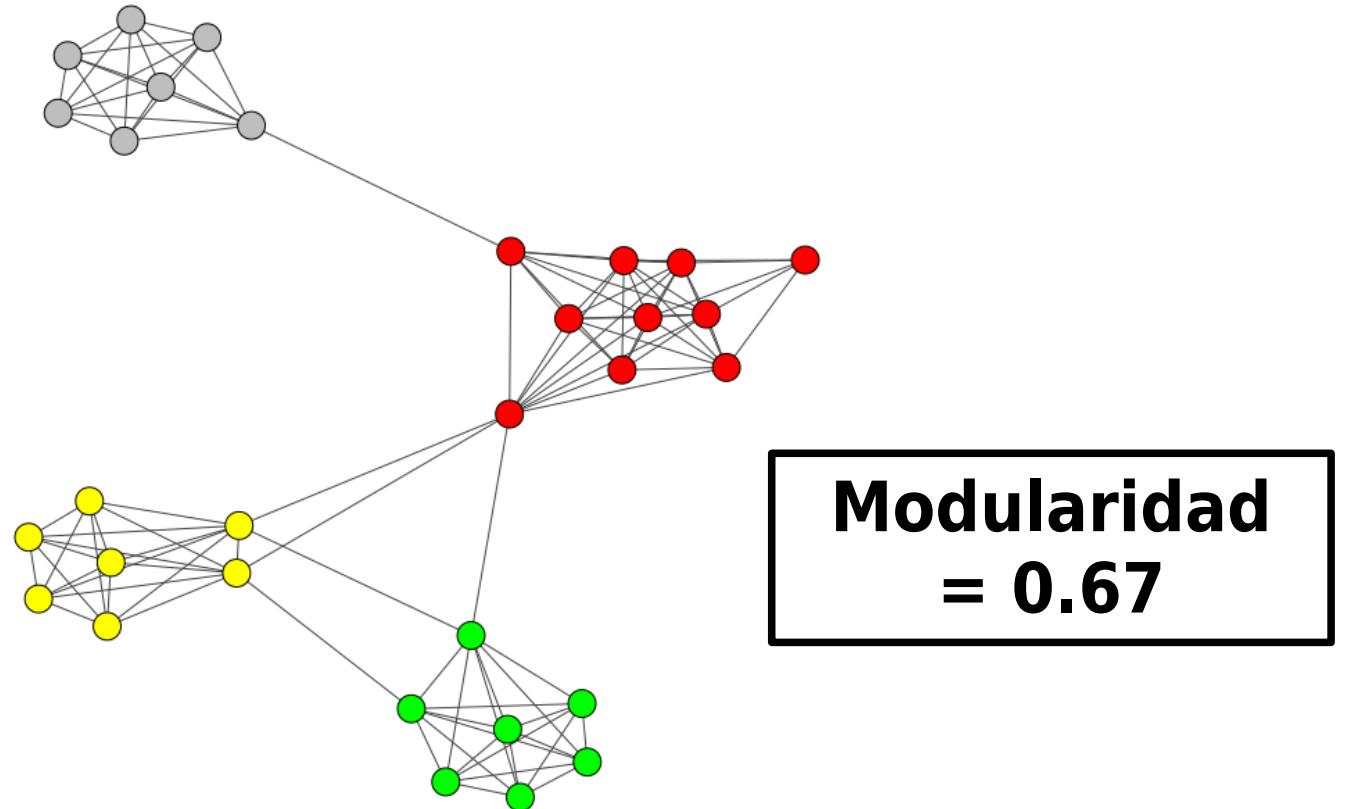
Este enfoque consiste en binarizar la red: si $w_{ij} >$ umbral $\rightarrow a_{ij} = 1$ (matriz de adyacencia).
Convertimos la red pesada en una red no pesada.



Componente conectado más grande en función del umbral.

Para los documentos de prueba, no hay nodos aislados para un umbral < 0.10 .

Binarización de la red



Binarización + detección de comunas

Las comunas coinciden con la red pesada.

Al binarizar se favorece los enlaces con más peso.

Sin embargo los pesos de los enlaces dentro de las comunidades deberían ser siempre mayores a los enlaces entre comunas.

Reducción de dimensionalidad **NMF**

NMF permite **mantener** el criterio de **construcción de los pesos de la red** como producto escalar entre los vectores documentos (previamente normalizados). Con PCA, los componentes negativos de los vectores obligarían a modificar el criterio.

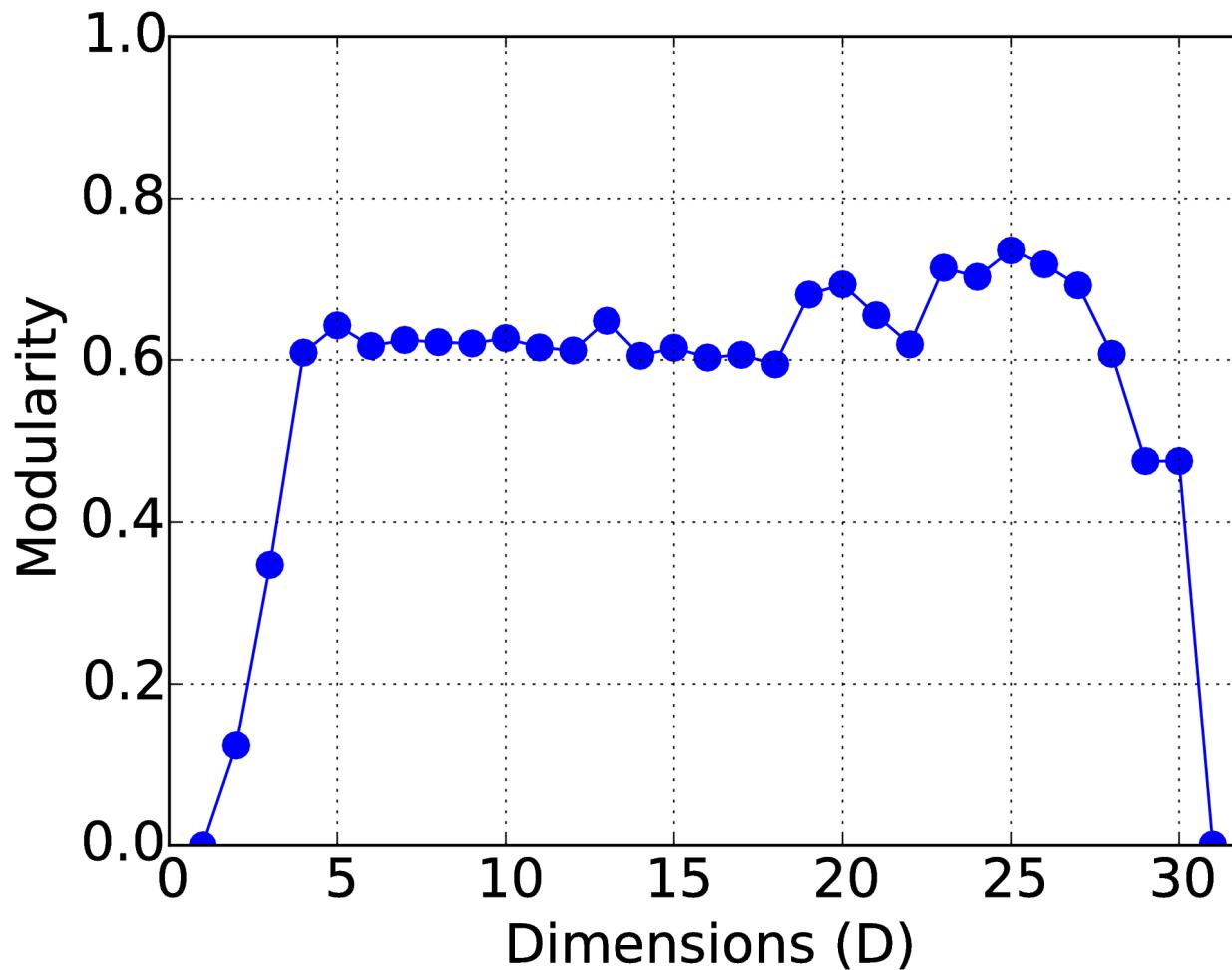
$$w_{ij} = \vec{v}_i \cdot \vec{v}_j = \cos(\theta)$$



Matriz de
documentos
reducida

Matriz de
adyacencia
pesada

Reducción de dimensionalidad **NMF**



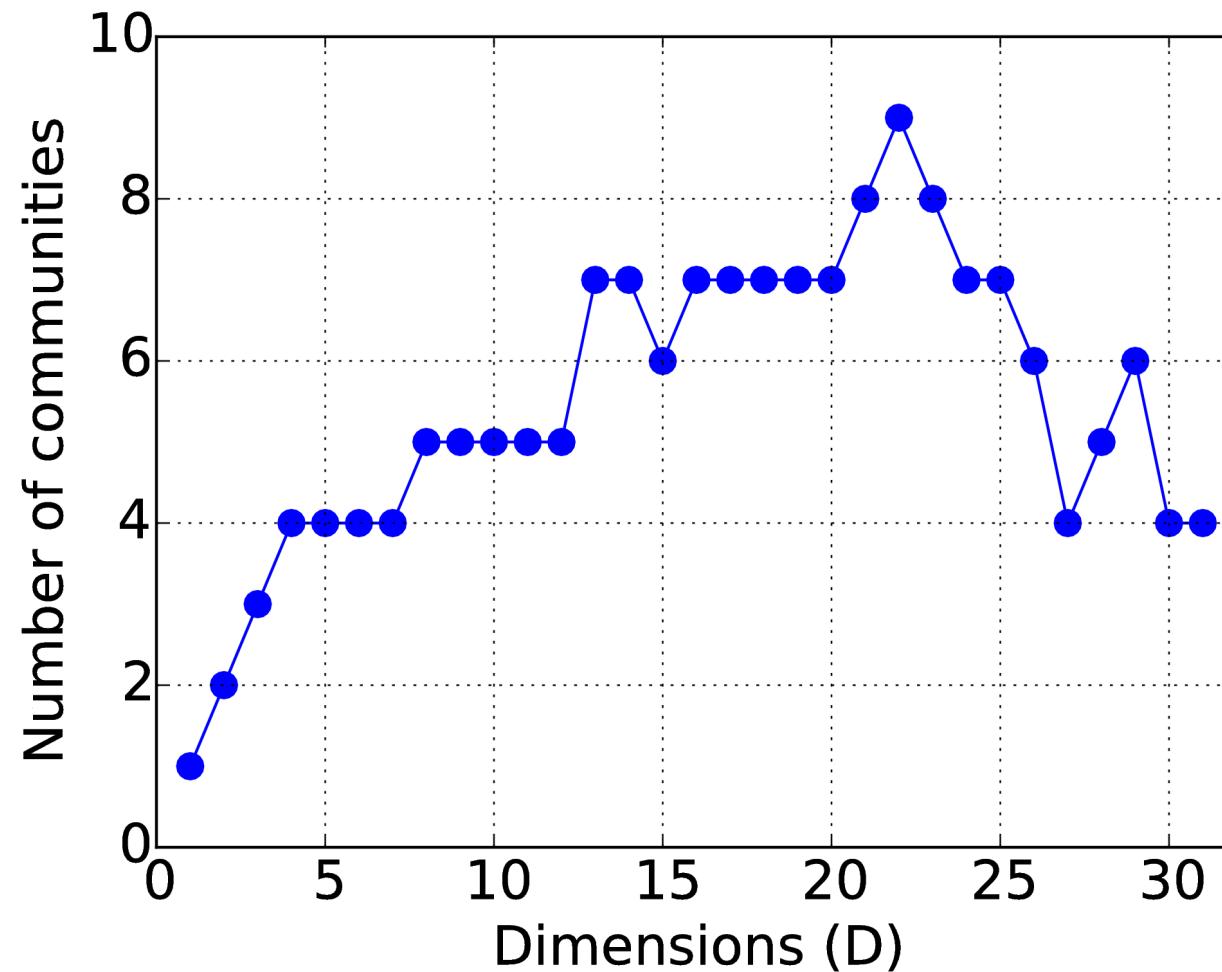
Algoritmo
Infomap

Modularidad en función del número de dimensiones empleadas:

Con 4 dimensiones la modularidad trepa a más de 0.60
(en la red original la modularidad era ~ 0.37).

Si bien no es la modularidad más alta, **más dimensiones no implica una alta ganancia en modularidad.**

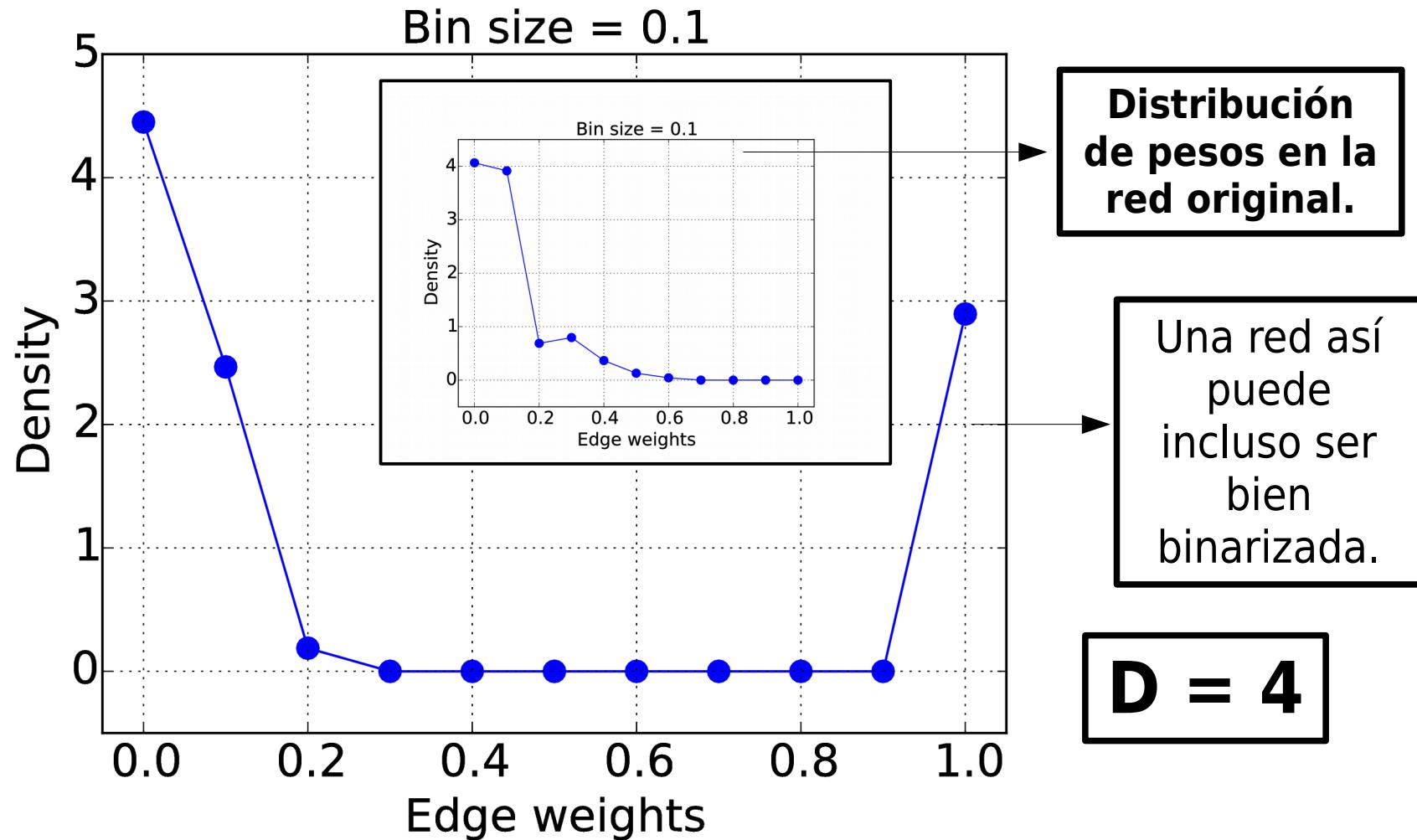
Reducción de dimensionalidad **NMF**



Número de comunidades detectadas:

Con 4 dimensiones se detectan los cuatro tópicos esperados.

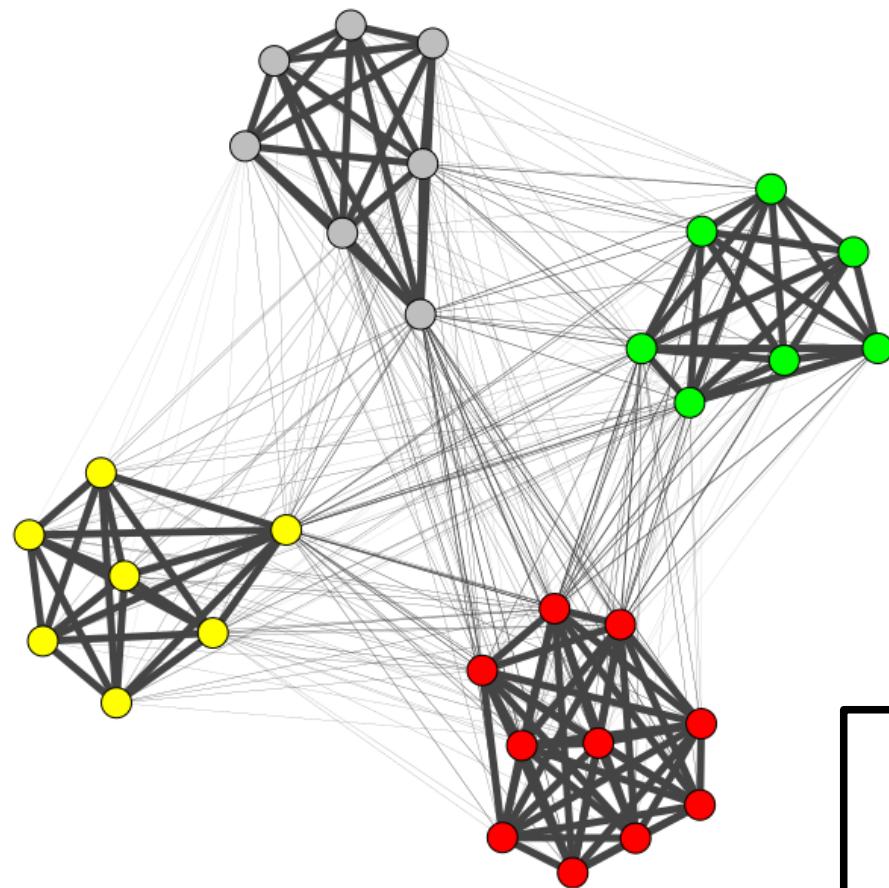
Reducción de dimensionalidad **NMF**



Histograma de los pesos de los enlaces para la red construida a partir de una reducción a 4 dimensiones con NMF.

La distribución es más similar a una U-shape que la red original.

Reducción de dimensionalidad **NMF**



Layout de la red pesada, luego de procesar los datos con **NMF** y $D = 4$.

Conclusiones Redes complejas

- *Redes pesadas parecen reproducir la estructura en tópicos de la red:* más aún si se aplica una **reducción de la dimensionalidad.**
- *La binarización provee una alternativa:* pero **requiere cierta homogeneización** en los pesos de los enlaces dentro de las comunidades.

Trabajo inmediato

- *Estudiar estas metodologías sobre conjunto de notas más grandes, con temáticas más similares.*
- *Sacar los perfiles temporales de cada tópico.*