IIC 2433 Minería de Datos

https://github.com/marcelomendoza/IIC2433

- TSNE, MDS Y UMAP -

Stochastic Neighbor Embedding (SNE)

Objetivo: Proyectar los datos a 2D o 3D para visualización.

Idea: Convertir distancias (Euclideanas) a probabilidades condicionales.

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$$

Vecindario (parametrizable)

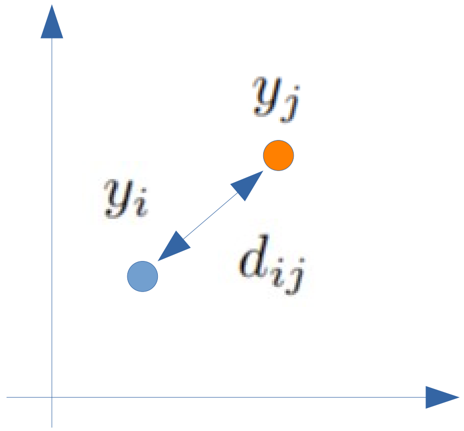
Definimos la proyección tal que: $$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)}$$

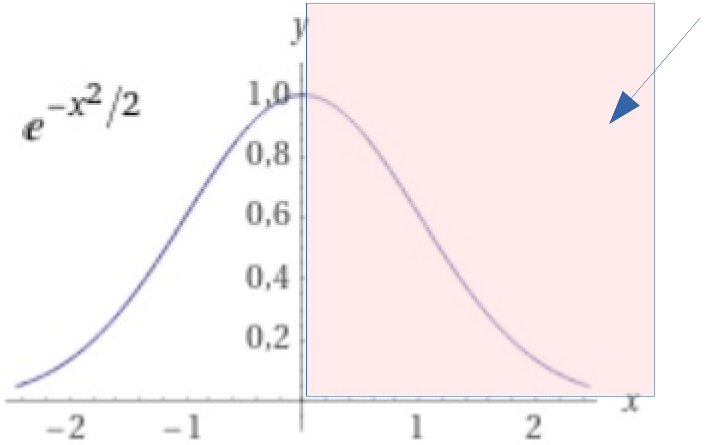
Notar que: $p_{i|i} = q_{i|i} = 0.$

Stochastic Neighbor Embedding (SNE)

Hacemos lo mismo en un espacio de menor dimensionalidad (proyección):

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)}$$





Stochastic Neighbor Embedding (SNE)

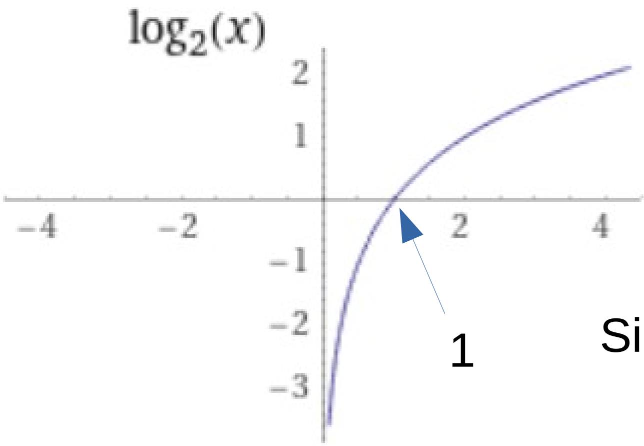
¿Cómo mido cuanto se parece el espacio original al proyectado?

Voy a comparar distribuciones de probabilidad.

Divergencia de Kullback-Leibler:

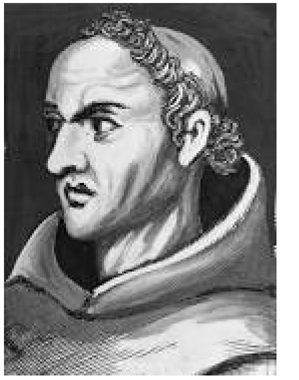
$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

La divergencia es menor en la medida que ambas distribuciones son más parecidas.



Si p~q → log p/q ~ 0

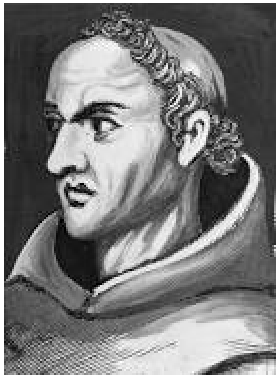
Model complexity



Principio (navaja de Ockham o principio de parsimonia)

“El modelo más simple es también el modelo más plausible”

Model complexity



Principio (navaja de Ockham o principio de parsimonia)

“El modelo más simple es también el modelo más plausible”

Una medida de complejidad: Entropía (basada en familias de objetos)

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}.$$

Explicación: entropía como medida de información.

Lanzamos una moneda 4 veces. Posibles estados del ejercicio: $2 \cdot 2 \cdot 2 \cdot 2$

estados

$\log_2(16) = 4$ ← Bits para codificar los estados

ej. CSSC

p=1/estados → $-\log_2(1/16) = 4$

Model complexity

Si los eventos no son equiprobables, debemos promediar:

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}.$$

Información codificada en el espacio original

Volvamos a SNE:

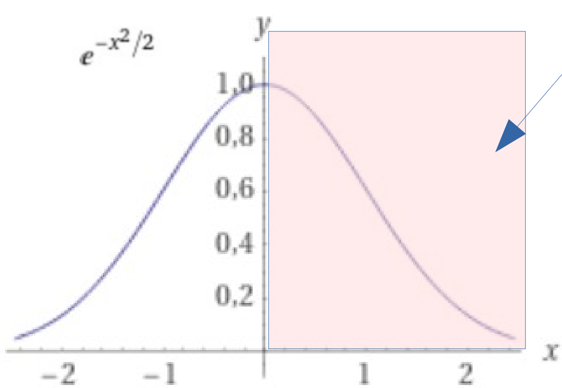
Me da el # de estados promedio (vecinos de cada punto)

El usuario define: $Perp(P_i) = \boxed{2^{H(P_i)}}$

lo cual permite determinar σ_i (internamente).

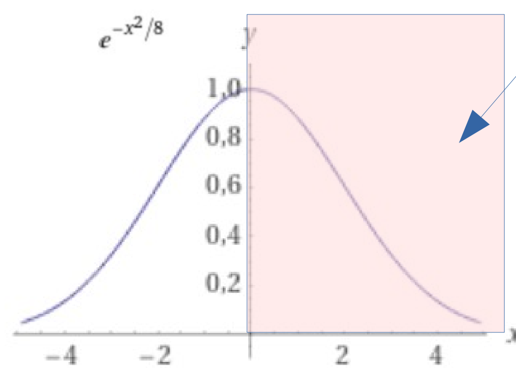
Es decir, el usuario define la complejidad de la proyección, la cual es modelada en sigma!!!

Menos pares



$\sigma = 1$

Más pares



$\sigma = 2$

Multi-Dimensional Scaling (MDS)

MDS = Principal Coordinate Analysis (PCoA)

Dos variantes: métrica (datos continuos) y no métrica (datos ordinales)

MDS: se calcula una matriz de proximidades o distancias en el espacio original. La proyección preserva las distancias (valores) originales.

Non metric MDS: se calcula una matriz de proximidades o distancias en el espacio original. La proyección preserva el orden entre los objetos.

Matriz de distancias entre objetos

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Multi-Dimensional Scaling (MDS)

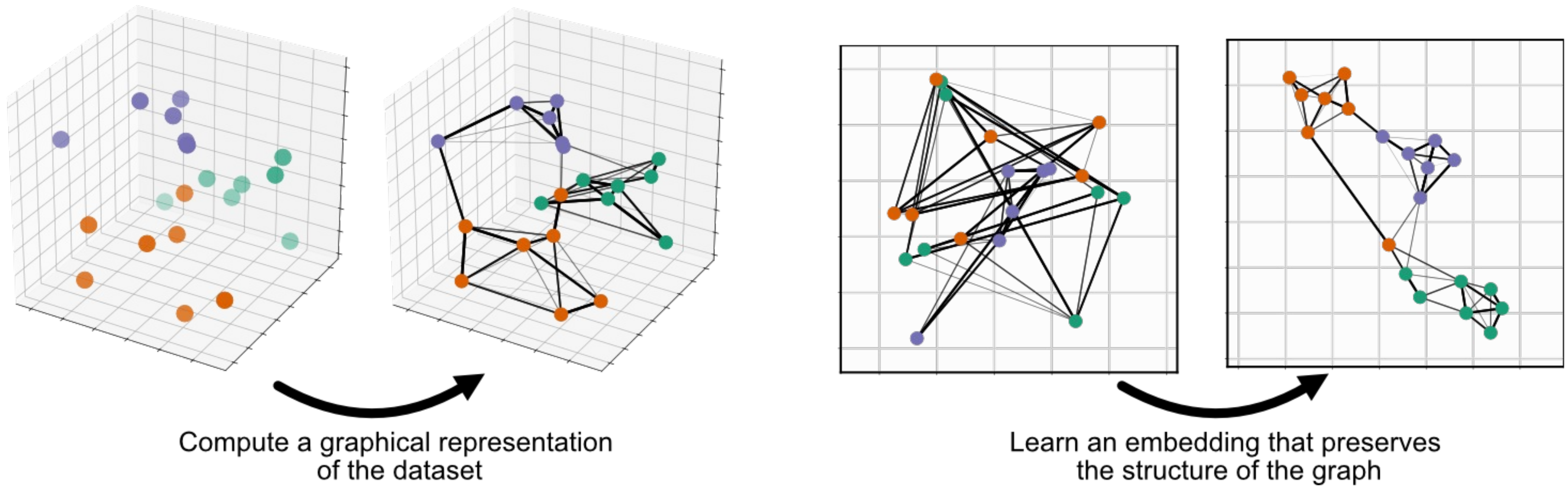
MDS: para random projections de X, se calcula:

$$stress = \sqrt{\frac{\sum_i \sum_j (d_{ij} - \hat{d}_{ij})}{\sum_i \sum_j \left(d_{ij}^2\right)}}$$

Luego se usa un algoritmo iterativo que optimiza la función objetivo.

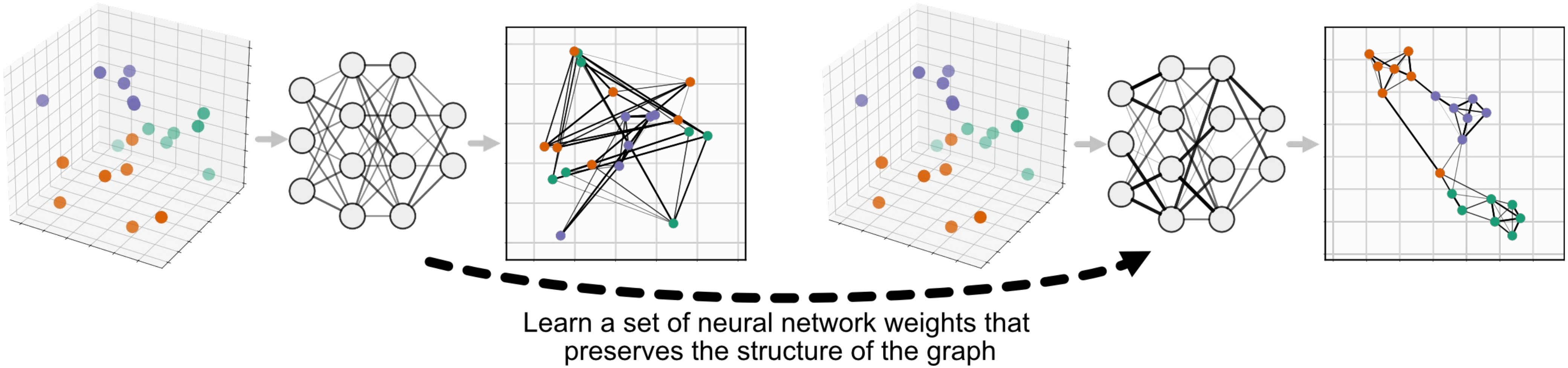
Uniform Manifold Approximation and Projection (UMAP)

Idea básica: UMAP calcula un grafo que representa los datos, luego aprende un embedding a partir del grafo.



Uniform Manifold Approximation and Projection (UMAP)

UMAP paramétrico



- KMEANS -

Clustering permite entender como se agrupan los datos

Clustering con k-means

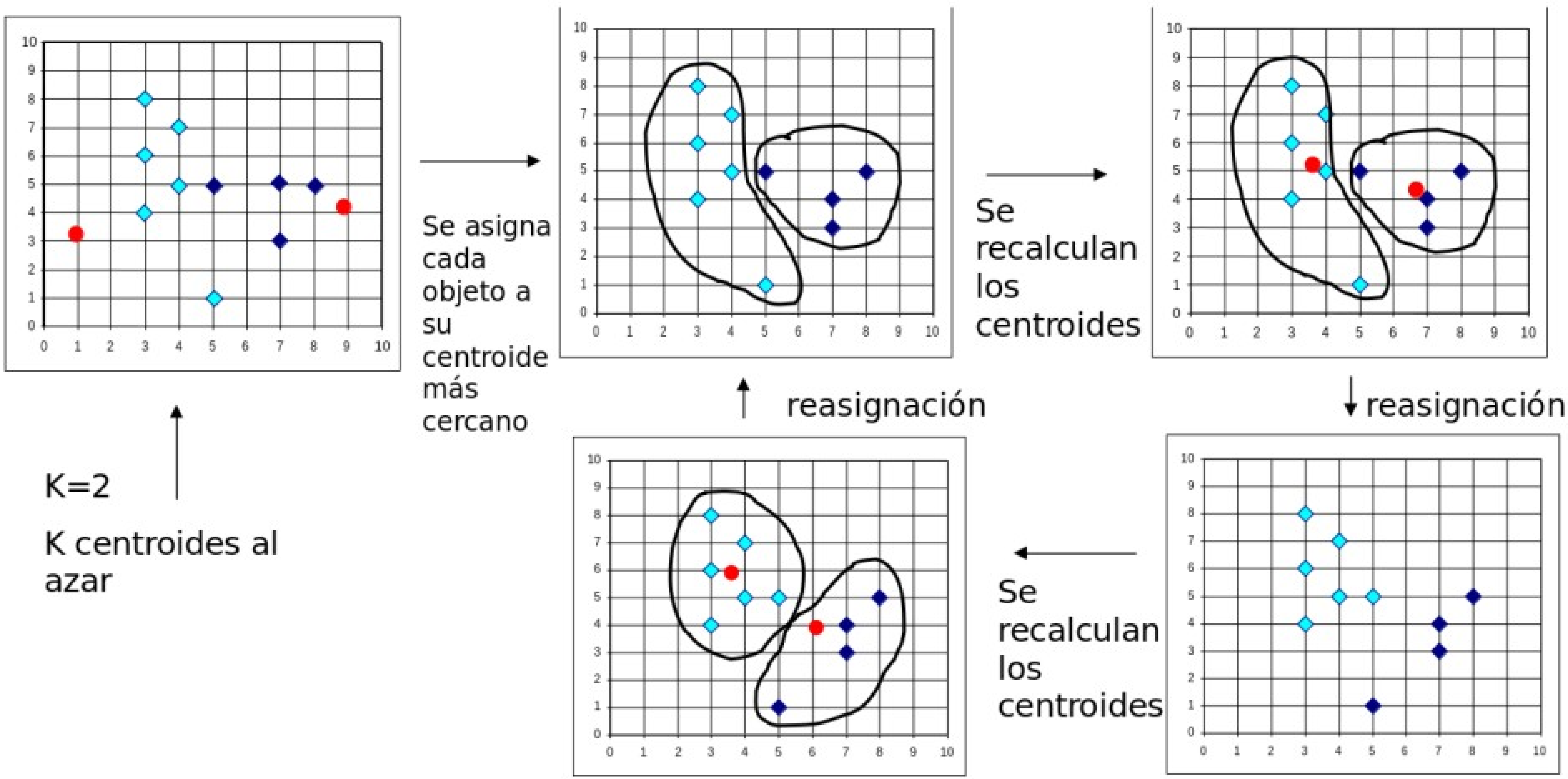
- Cada cluster en K-means es definido por un centroide.
- Objetivo: optimizar alguna noción de distancia:
  1. Intra-cluster: (Minimizar) distancia entre objetos de un cluster a su centroide.
  2. Inter-cluster: (Maximizar) distancia entre objetos de clusters distintos.
- Centroide:

$$c_i = \frac{1}{m_i} \sum_{x \in C_i} x$$

  donde C_i denota un cluster.
- Idea del algoritmo:
  - Asignación inicial: `k` centroides al azar.
  - Reasignación: asignar cada objeto a su centroide más cercano (algoritmo avaro).
  - Recomputación: recalcular los centroides.

Clustering con k-means

Ejemplo



Clustering con k-means

Hechos importantes:

- K-means converge. (McQueen, 67)
- Criterios de parada
  1. Iteraciones: (Máximo) número de iteraciones.
  2. Error tolerado: (Optimizar) alguna noción de distancia entre objetos.
- Complejidad:
  1. K-means es $\mathtt{NP-hard}$ en cualquier espacio d-dimensional con distancia Euclideana o coseno.
  2. K-means es $\mathtt{NP-hard}$ para cualquier valor de k.

SSE: Suma de errores al ²

Clustering con k-means

k-means minimiza el SSE: implícitamente

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} (c_i - x)^2$$

SSE: Suma de errores al ²

Clustering con k-means

k-means minimiza el SSE: implícitamente

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} (c_i - x)^2$$

$$\begin{aligned}
\frac{\partial}{\partial c_k}\mathrm{SSE} &= \frac{\partial}{\partial c_k} \sum_{i=1}^{K} \sum_{x \in C_i} (c_i - x)^2 \\
&= \sum_{i=1}^{K} \sum_{x \in C_i} \frac{\partial}{\partial c_k} (c_i - x)^2 \\
&= \sum_{x \in C_k} 2 * (c_k - x_k) = 0
\end{aligned}$$

SSE: Suma de errores al ²

Clustering con k-means

k-means minimiza el SSE: implícitamente

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} (c_i - x)^2$$

$$\begin{aligned}
\frac{\partial}{\partial c_k}\mathrm{SSE} &= \frac{\partial}{\partial c_k} \sum_{i=1}^{K} \sum_{x \in C_i} (c_i - x)^2 \\
&= \sum_{i=1}^{K} \sum_{x \in C_i} \frac{\partial}{\partial c_k} (c_i - x)^2 \\
&= \sum_{x \in C_k} 2 * (c_k - x_k) = 0
\end{aligned}$$

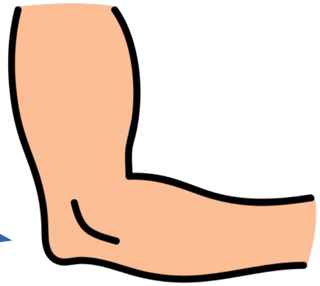
$$\sum_{x \in C_k} \cancel{2} * (c_k - x_k) = 0 \Rightarrow m_k c_k = \sum_{x \in C_k} x_k \Rightarrow c_k = \frac{1}{m_k} \sum_{x \in C_k} x_k$$

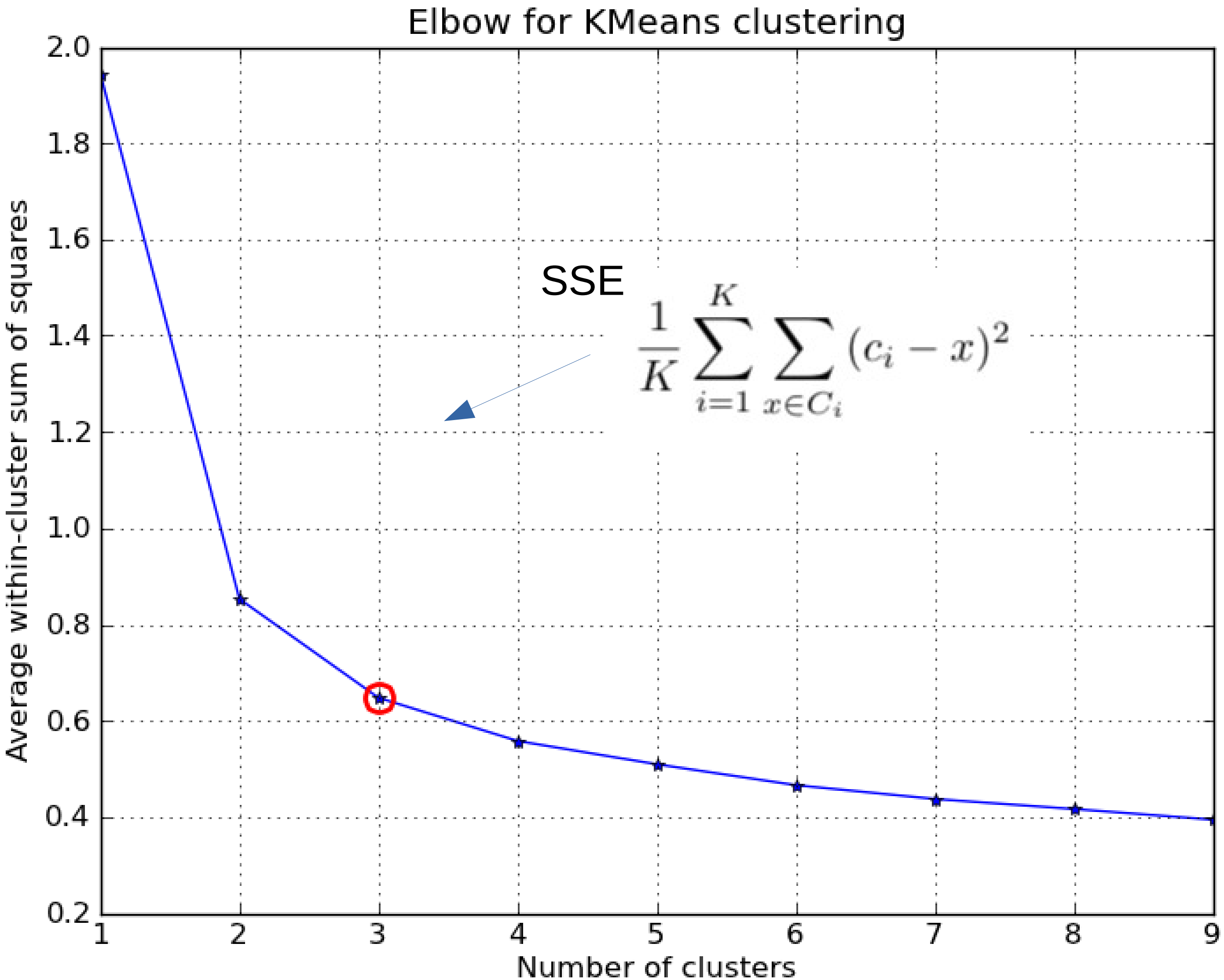
elementos en el clúster

¿Cuántos prototipos usamos?

ELBOW (codo):

Variar k buscando el codo





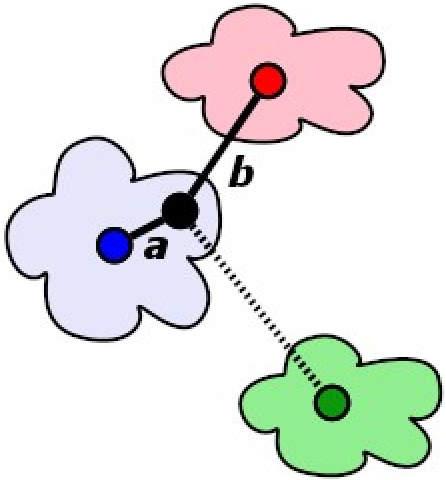
$$\frac{1}{K}\sum_{i=1}^{K}\sum_{x\in C_i}(c_i - x)^2$$

¿Cuántos prototipos usamos?

Silhouette:

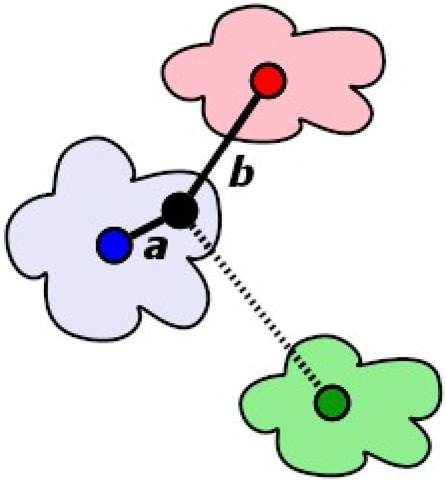
Congruencia de x_i a C_i: $$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

Congruencia de x_i a otros clusters: $$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$



¿Cuántos prototipos usamos?

Silhouette:



Congruencia de x_i a C_i: $$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

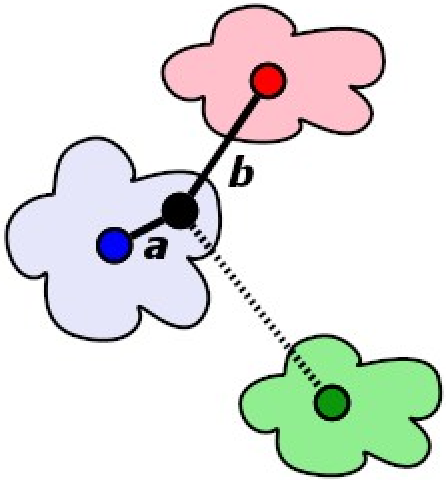
Congruencia de x_i a otros clusters: $$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

Silhouette Coef.: $$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad \text{si} \quad |C_i| > 1,$$

$$s(i) = 0, \quad \text{si} \quad |C_i| = 1.$$

¿Cuántos prototipos usamos?

Silhouette:



Congruencia de x_i a C_i: $$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j)$$

Congruencia de x_i a otros clusters: $$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

Silhouette Coef.: $$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad \text{si} \quad |C_i| > 1,$$

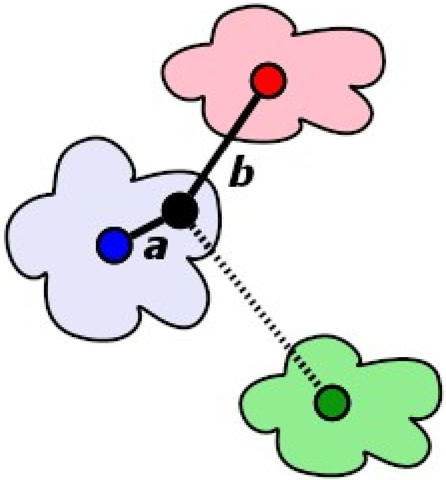
$$s(i) = 0, \quad \text{si} \quad |C_i| = 1.$$

¿Intervalo?

¿Cuántos prototipos usamos?

Silhouette:

Congruencia de x_i a C_i: $$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j)$$



Congruencia de x_i a otros clusters: $$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

Silhouette Coef.: $$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad \text{si} \quad |C_i| > 1,$$

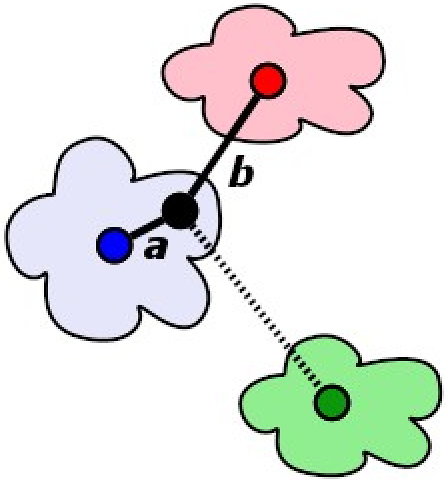
$$s(i) = 0, \quad \text{si} \quad |C_i| = 1.$$

[-1, 1]

¿Cuántos prototipos usamos?

Un valor alto indica poca congruencia

Silhouette:



Congruencia de x_i a C_i: $$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j)$$

Congruencia de x_i a otros clusters: $$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j)$$

Silhouette Coef.: $$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad \text{si} \quad |C_i| > 1,$$

$$s(i) = 0, \quad \text{si} \quad |C_i| = 1.$$

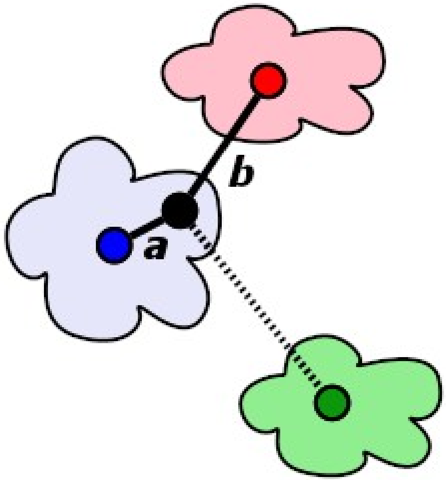
[-1, 1]

¿Cuántos prototipos usamos?

Silhouette:

Un valor alto indica poca congruencia

Congruencia de x_i a C_i: $$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$



Congruencia de x_i a otros clusters: $$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

Silhouette Coef.: $$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad \text{si} \quad |C_i| > 1,$$

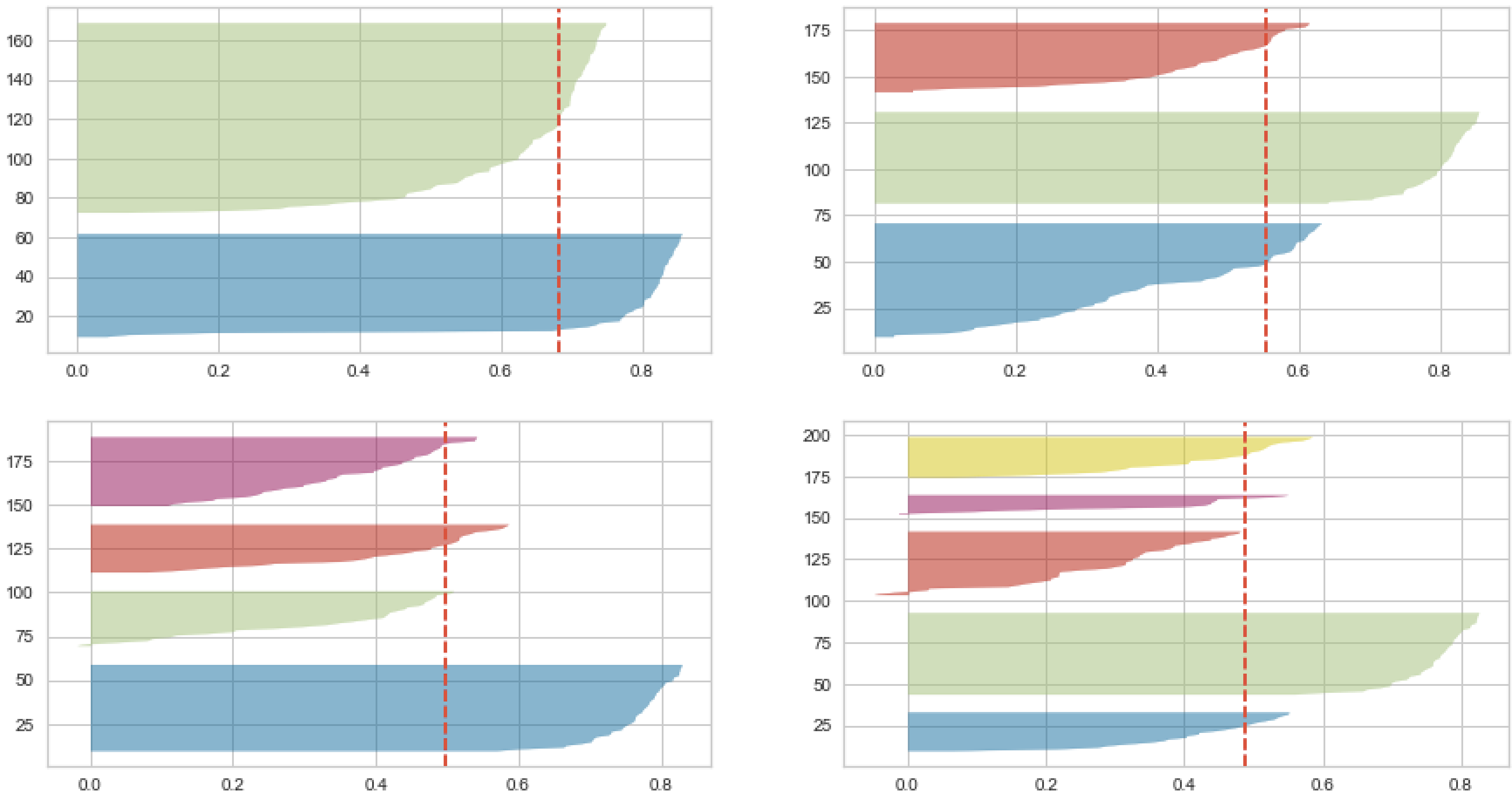
$$s(i) = 0, \quad \text{si} \quad |C_i| = 1.$$

Un valor alto indica alta congruencia

[-1, 1]

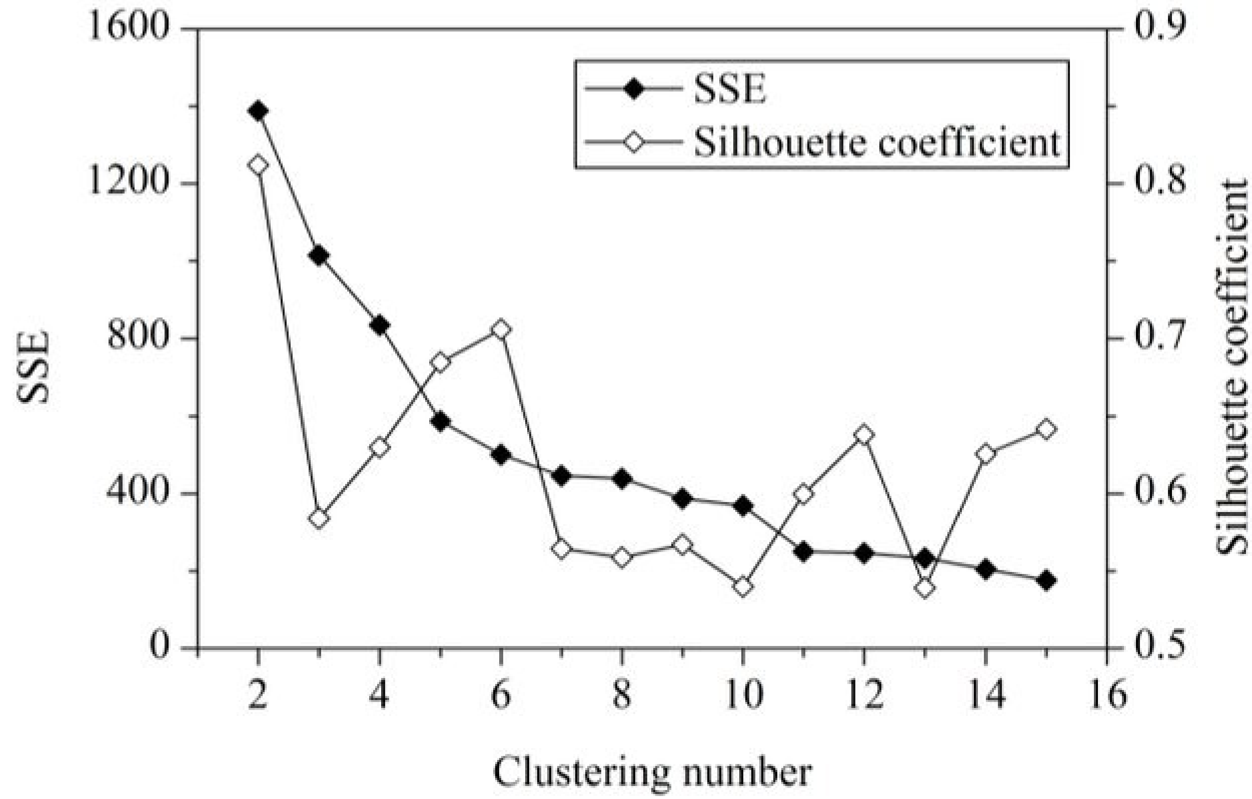
¿Cuántos prototipos usamos?

Silhouette promedio:



¿Cuántos prototipos usamos?

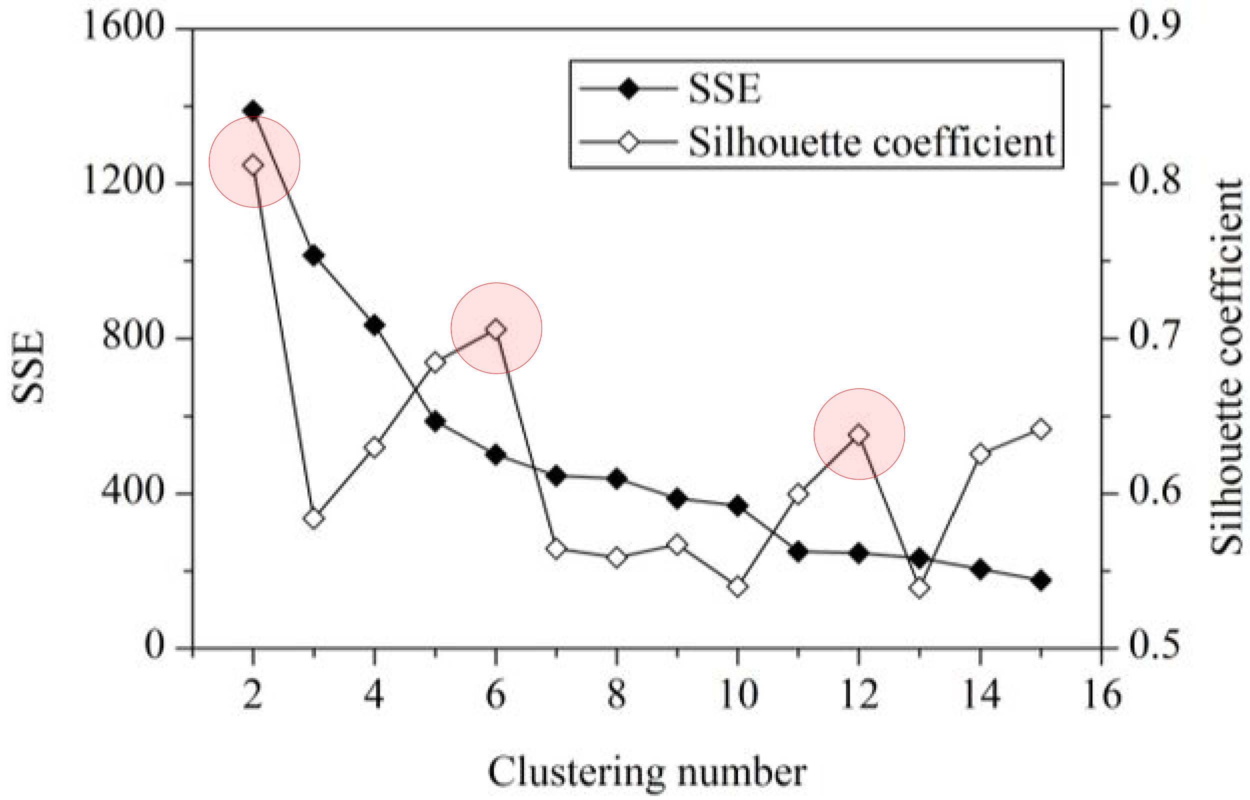
Silhouette v/s ELBOW:



¿Con cuál se quedan?

¿Cuántos prototipos usamos?

Silhouette v/s ELBOW:



¿Con cuál se quedan?