



IIC 2433 Minería de Datos

<https://github.com/marcelomendoza/IIC2433>

- TSNE y UMAP -

Stochastic Neighbor Embedding (SNE)

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

Idea: Convertir distancias (Euclideanas) a probabilidades condicionales.

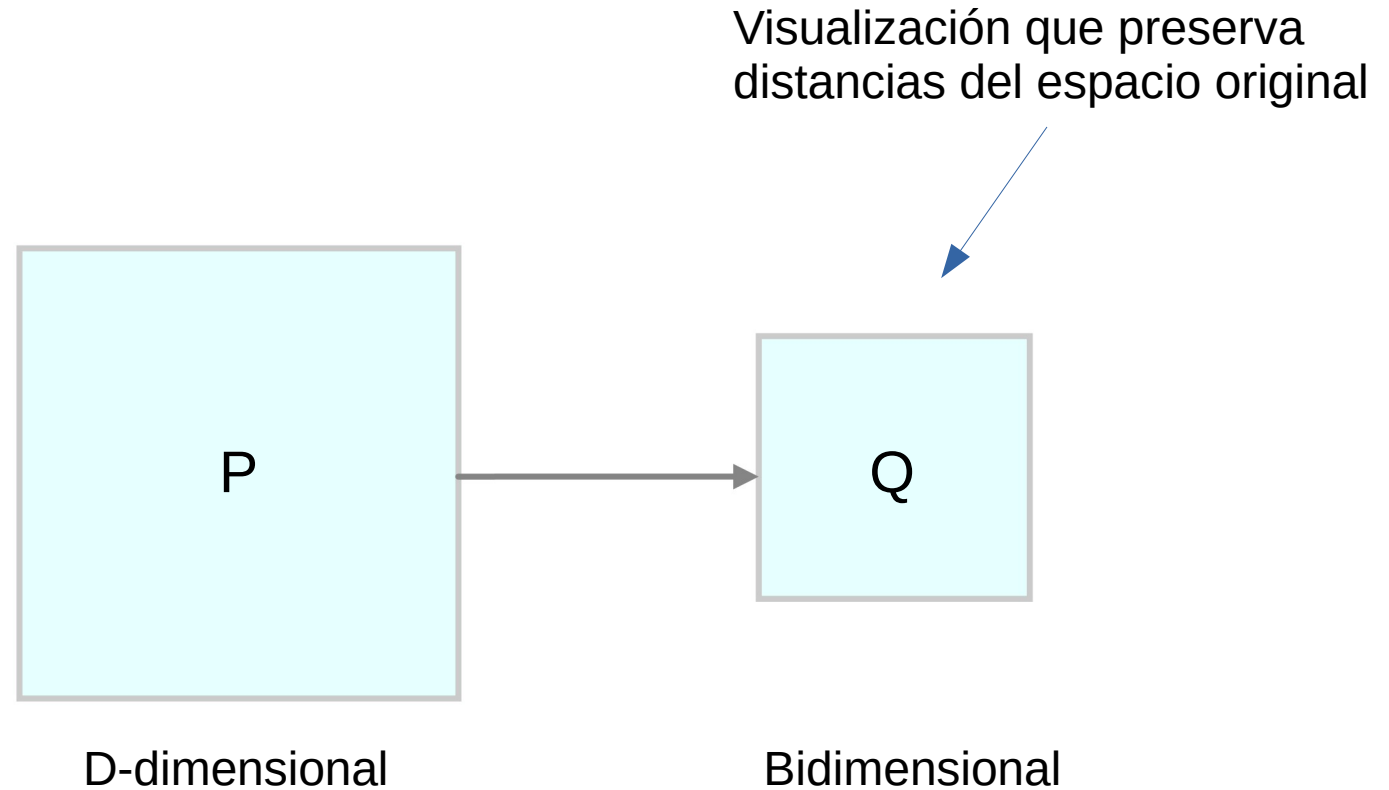
$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$$

Vecindario
(parametrizable)

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

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

Stochastic Neighbor Embedding (SNE)



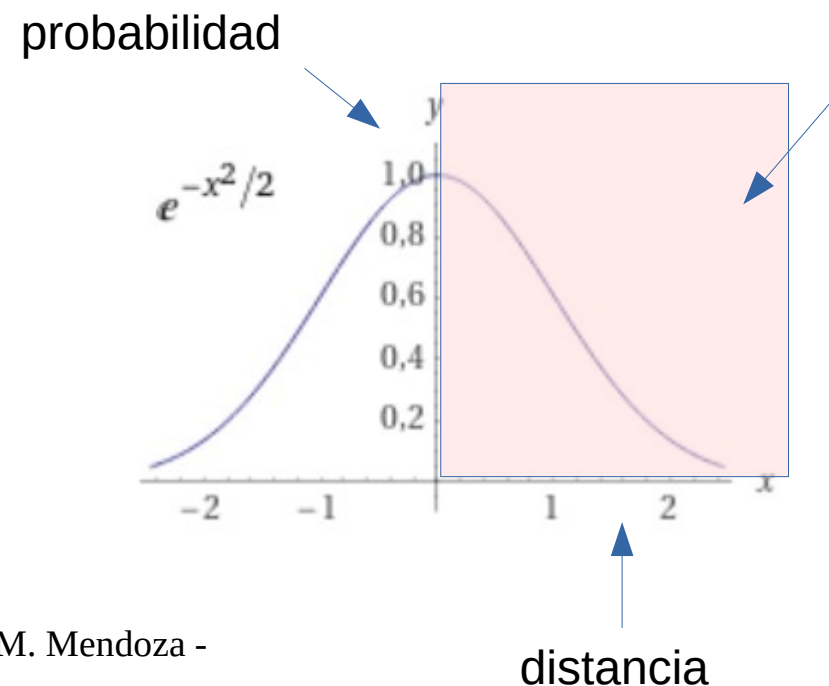
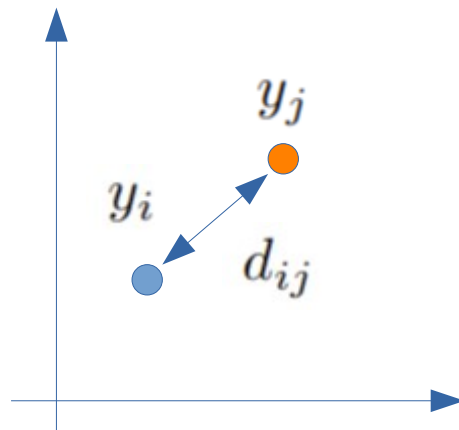
Stochastic Neighbor Embedding (SNE)

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

probabilidad

distancia

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



Stochastic Neighbor Embedding (SNE)

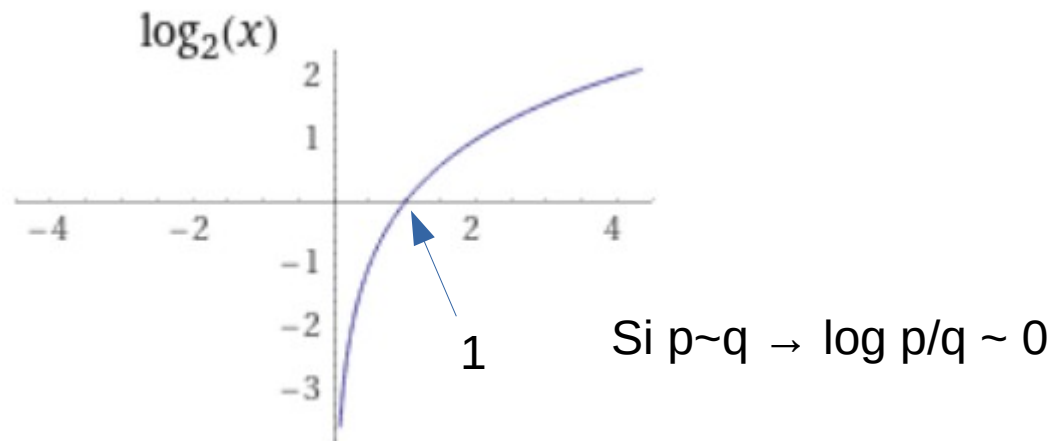
¿Cómo mido cuanto se parece el espacio original al proyectado?

Voy a comparar las distribuciones de probabilidad P y Q.

Divergencia de Kullback-Leibler:

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

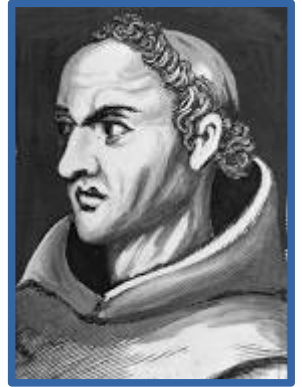
La divergencia es menor en la medida que ambas distribuciones son más parecidas.



Model complexity

Principio (navaja de Ockham o principio de parsimonia)

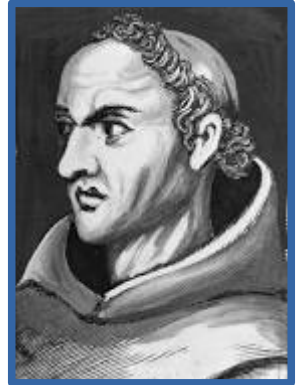
“El modelo más simple es también el modelo más plausible”



Model complexity

Principio (navaja de Ockham o principio de parsimonia)

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



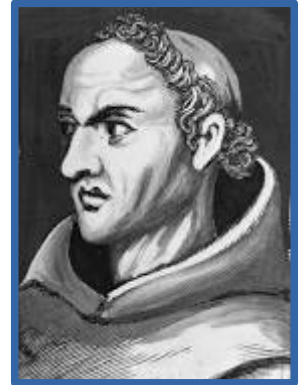
Una medida de complejidad: Entropía (basada en familias de objetos)

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

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



$$\log_2(16) = 4$$



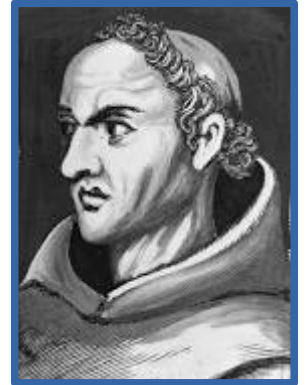
Bits para codificar los estados

ej. CSSC

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)

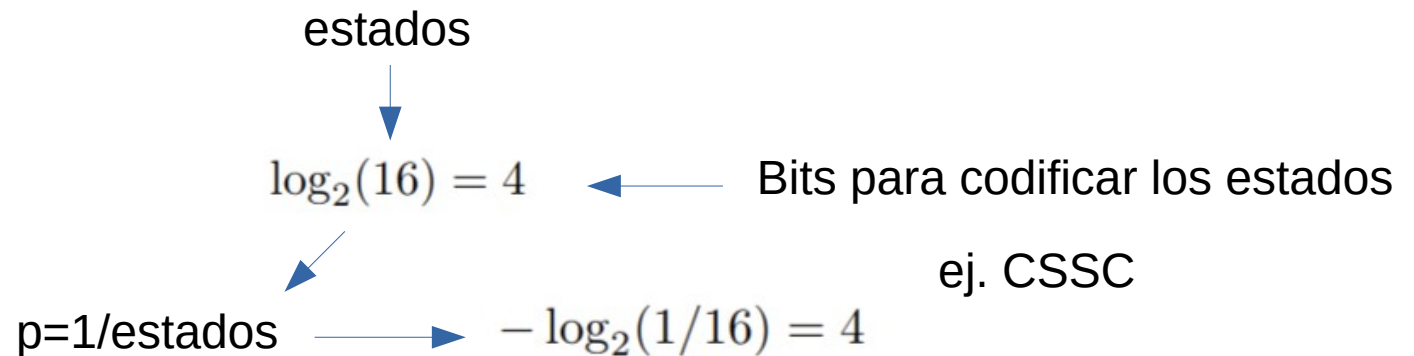
$$H(P_i) = - \sum_j p_{j|i} \log_2 p_{j|i}.$$

Explicación: entropía como medida de información.

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



Probabilidad
de un
resultado en
particular:



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

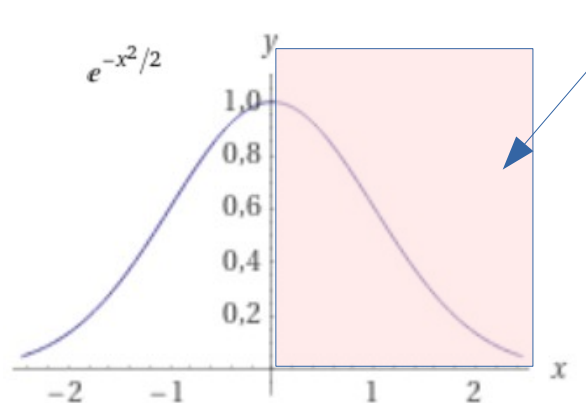
Volvamos a SNE:

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

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

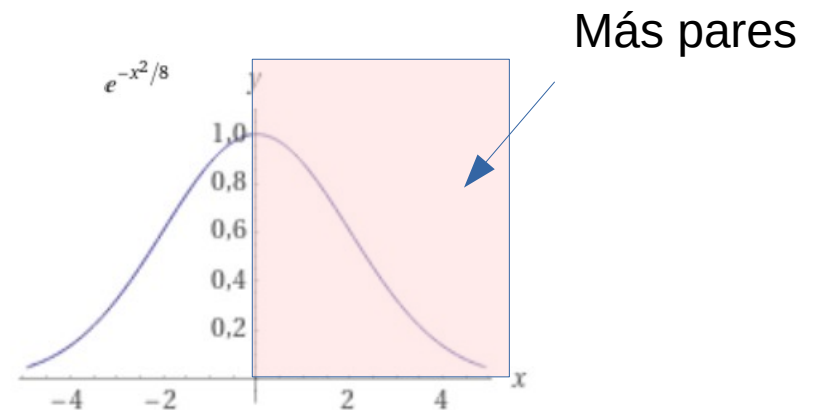
lo cual permite determinar σ_i (internamente).

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



Menos pares

$\sigma = 1$



Más pares

$\sigma = 2$

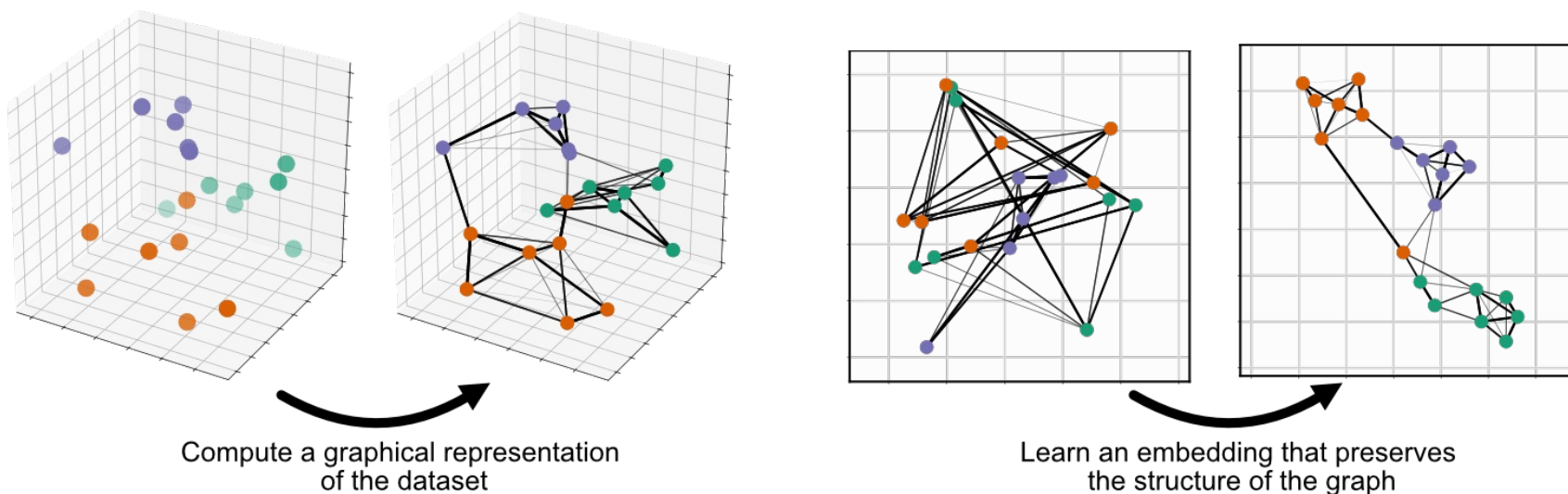
Stochastic Neighbor Embedding (SNE)

- Debemos calibrar el parámetro perplexity.
- El parámetro nos indica la complejidad de la proyección:
mayor perplejidad \rightarrow menor parsimonia
- Mayor perplejidad \rightarrow mayor p \rightarrow más vecinos \rightarrow mayor sigma

Nota: En rigor usaremos una versión simétrica de SNE denominada t-SNE (reemplaza KL por Jensen-Shannon).

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

