IIC 2433 Minería de Datos

https://github.com/marcelomendoza/IIC2433

Cierre de la clase 1 – Preprocesamiento de datos y PCA

Preprocesamiento de datos:

¿Cuál técnica usaré para codificar variables nominales?

¿Cuál técnica usaré para codificar variables ordinales?

¿Cuál función de distancia o similitud debo usar si tengo datos en alta dimensionalidad?

PCA:

¿Qué ocurre con la varianza acumulada en la medida que aumento el número de Componentes principales?

¿Cómo determino el número de componentes que necesito para capturar el x% de la varianza de un dataset?

- TSNE y UMAP -

Clase 2 – t-SNE y UMAP

Objetivos de la clase

- Reconocer las técnicas de visualización de datos t-SNE y UMAP.
- Comprender cómo funcionan estas técnicas.
- Distinguir entre una técnica de visualización y una de reducción de dimensionalidad.

Resultado de aprendizaje

Aplicar una tećnica de reducción de dimensionalidad y una de visualización a un dataset identificando diferencias y similitudes entre ellas.

Plan

- Sesión 1: clase convencional.
- Sesión 2: clase activa, resolverán un desafío en clases (actividad formativa en equipo).

Stochastic Neighbor Embedding (SNE)

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

Idea: Convertir distancias (Euclideanas) a probabilidades condicionales.

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

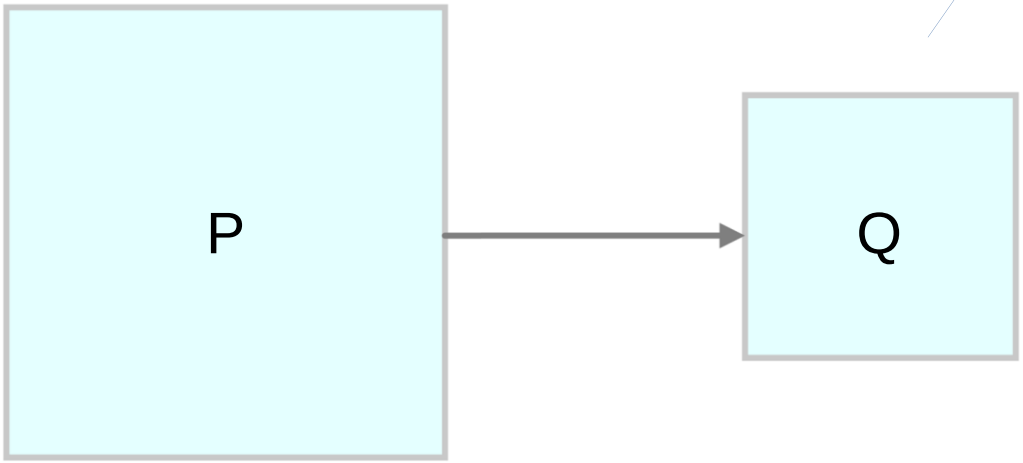
Vecindario (parametrizable)

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

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

Stochastic Neighbor Embedding (SNE)

Visualización que **preserva distancias** del espacio original



D-dimensional

Bidimensional

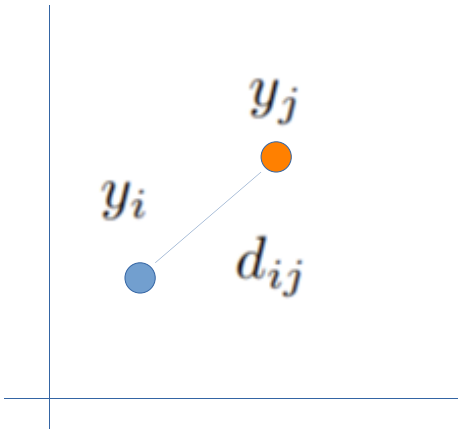
Stochastic Neighbor Embedding (SNE)

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

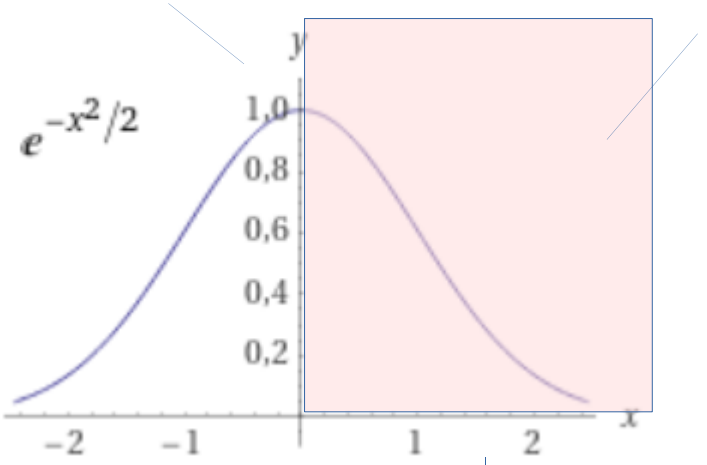
probabilidad

distancia

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



probabilidad



distancia

Stochastic Neighbor Embedding (SNE)

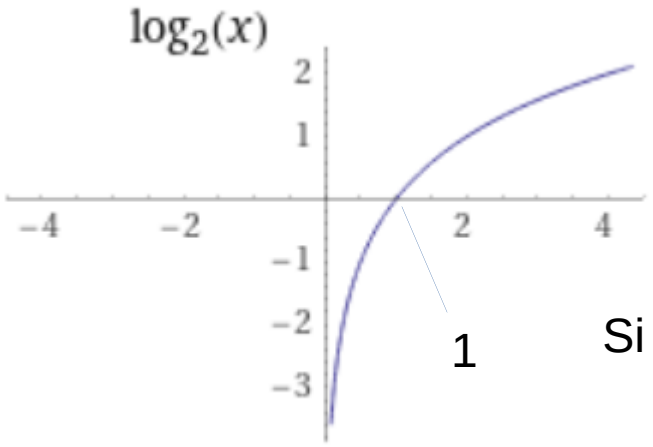
¿Cómo mido cuanto se parece el espacio original al proyectado?

Voy a comparar las distribuciones de probabilidad P y Q.

Divergencia de Kullback-Leibler:

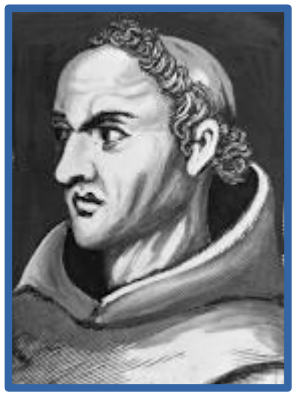
$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

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



Si p~q → log p/q ~ 0

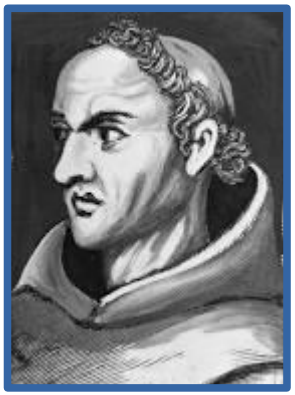
Model complexity



Principio (navaja de Ockham o principio de parsimonia)

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

Model complexity



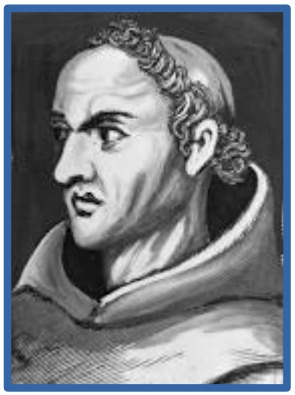
Principio (navaja de Ockham o principio de parsimonia)

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

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

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

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.

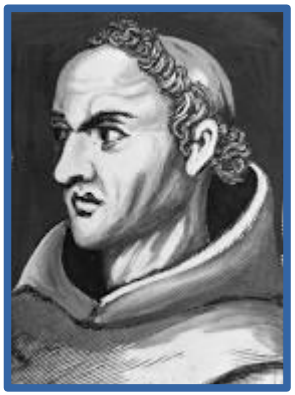
Lanzamos una moneda 4 veces. Posibles estados del ejercicio: $2 \cdot 2 \cdot 2 \cdot 2$

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: $2 \cdot 2 \cdot 2 \cdot 2$

estados

Probabilidad de un resultado en particular: $\log_2(16) = 4$ — 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:

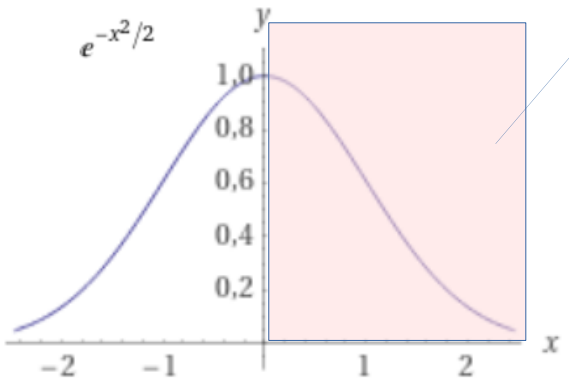
Me da el # de estados promedio (vecinos de cada punto)

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

lo cual permite determinar σ_i (internamente).

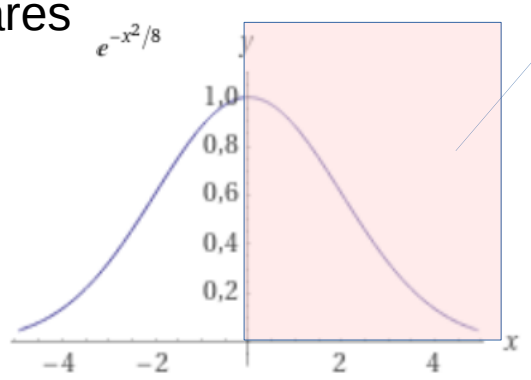
Es decir, el usuario define la complejidad de la proyección, la cual es modelada en sigma!!!

Menos pares



$\sigma = 1$

Más pares



$\sigma = 2$

Stochastic Neighbor Embedding (SNE)

- Debemos calibrar el parámetro perplejidad.

- El parámetro nos indica la complejidad de la proyección:

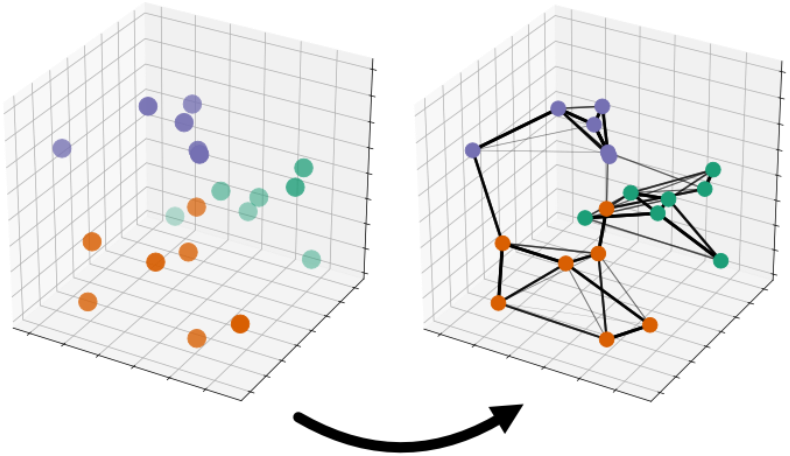
mayor perplejidad → menor parsimonia

- Mayor perplejidad → mayor p → más vecinos → mayor sigma

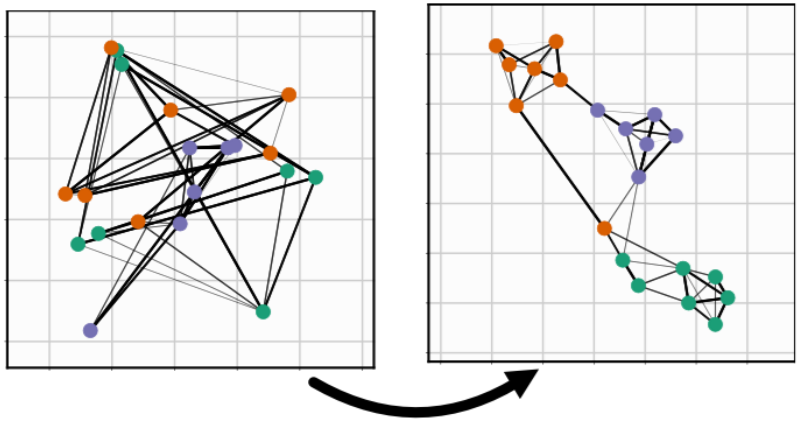
Nota: En rigor usaremos una versión simétrica de SNE denominada t-SNE (reemplaza KL por Jensen-Shannon).

Uniform Manifold Approximation and Projection (UMAP)

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



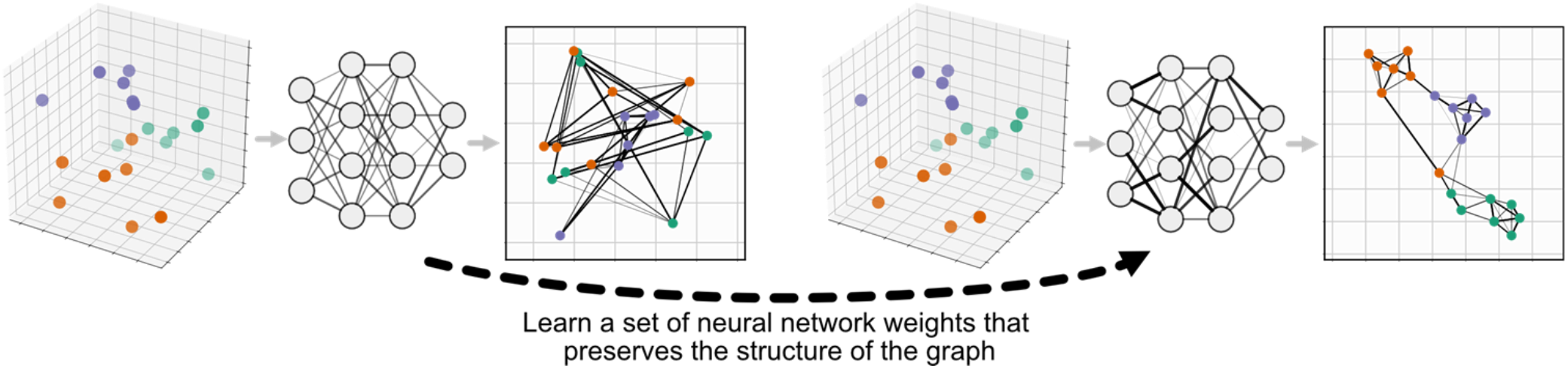
Compute a graphical representation of the dataset



Learn an embedding that preserves the structure of the graph

Uniform Manifold Approximation and Projection (UMAP)

UMAP paramétrico



De esta forma, UMAP disminuye la dependencia de la técnica en relación con el parámetro de perplejidad.