Aprendizaje Estadístico 2024

Lista 03

19.febrero.2024

En esta tarea se requiere realizar visualizaciones de *manifold learning* sobre varios conjunto de datos.

1. Usamos para tSNE la distancia de Kullback-Leibler. Para distribuciones discretas su definición es:

$$D_{KL}(\mathbb{P}_1||\mathbb{P}_2) = \sum_{\omega} \mathbb{P}_1(\omega) \log \frac{\mathbb{P}_1(\omega)}{\mathbb{P}_2(\omega)}.$$

   Calcular $D_{KL}(\mathbb{P}_1||\mathbb{P}_2)$ si $\mathbb{P}_1 \sim Ber(p_1)$ y $\mathbb{P}_2 \sim Ber(p_2)$. Para un parámetro p_1 fijo, graficar $D_{KL}(\mathbb{P}_1||\mathbb{P}_2)$ como función de p_2, y verificar que efectivamente mide de alguna manera la disimilitud entre $\mathbb{P}_1$ y $\mathbb{P}_2$.

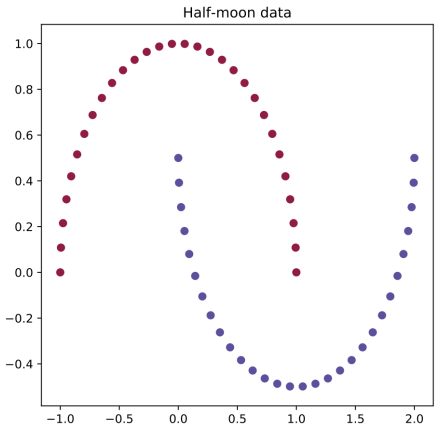
2. Escribir una función que implementa Kernel PCA con un kernel centrado de base radial con parámetro σ, y hacer $\mathbb{K}_c = \mathbb{J}\mathbb{K}\mathbb{J}$, con

$$K(i,j) = \exp(||x_i x_j||^2/\sigma),$$

   y $\mathbb{J}$ la matriz para centrar.

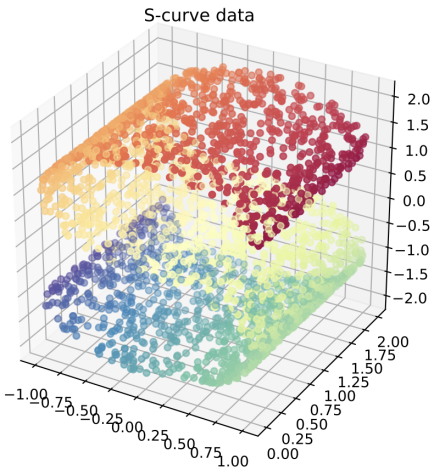
   Aplicar esto para datos en 2D con la siguiente estructura de la Figura 1 (tomar en cada clase 30 o más observaciones). Muestre cómo las proyecciones sobre los primeros dos componentes cambian en función de σ y cómo Kernel PCA se aproximan a PCA si $\sigma \to \infty$.

   (Cuidado! Algunas funciones ya implementadas en Python o R, en lugar de usar el parámetro σ utilizan el parámetro $\gamma = \sigma^{-1}$.)



   La distribución de puntos de la Figura anterior en Python puede llamarse como:
   `from sklearn.datasets import make_moon`
   `X, color = make_moon(n_samples=60).`

3. Hacer una comparación de distintos métodos de proyección local o manifold learning: Linear local embedding LLE, t-SNE, Spectral Embedding, ISOMAP y Escalamiento multidimensional MDS, para comparar diferentes visualizaciones de un conjunto de datos.



Aplicarlo para datos en 3D con la siguiente estructura de la Figura 2 (tomar 3000 observaciones). Muestre las proyecciones sobre los primeros dos componentes para cada método. Ilustre todas las proyecciones en un mismo plot, usando subplot, para facilitar la comparación visual.

La distribución de puntos de la Figura anterior en Python puede llamarse como:
`from sklearn.datasets import make_s_curve`
`X, color = make_s_curve(n samples=3000).`

4. El conjunto de datos **wines.csv**, contiene información sobre varios componentes químicos asociados a tres diferentes diferentes cepas de vino.

   Elija 3 métodos de *Manifold Learning* (t-SNE debe ser uno de estos), y realice visualizaciones en 2D ó 3D para este conjunto. A partir de las visualizaciones, tratar de:

   - Identificar alguna explicación para los ejes.
   - Identificar posibles grupos o clústers.
   - Verificar si dichos conglomerados separan bien a las tres diferentes cepas de vino.

   A partir de sus visualizaciones, explicar las principales diferencias entre las tres cepas.

   Más información sobre el origen de estos datos puede encontrarse en http://archive.ics.uci.edu/ml/machine-learning-databases/wine.

5. El índice de felicidad *Happy Planet Index* es un intento de medir el bienestar sostenible para todos, el cual puede encontrarse en el sitio http://happyplanetindex.org/.

   Puedes descargar los datos del índice de felicidad 2016 en http://happyplanetindex.org/resources o también están disponibles en el archivo `hpi-data-2016.xlsx`.

   Usar el método SOM para encontrar visualizaciones útiles para el conjunto de datos del índice de felicidad y de sus variables. Complementar estas visualizaciones con otros métodos, y hacer un análisis completo a partir de los datos.

   Construir un "mapa de calor" que permita visualizar el índice de felicidad de cada país mediante tonos de color. Muestre su visualización y explique sus hallazgos y conclusiones sobre la relación entre este índice y las regiones geográficas.