Enhanced visualization: assignment 10

Dimensionality reduction

In this labwork, we are going to test different dimensionality reduction methods on the world population dataset used in the previous assignment (file `wpd.csv`). The objective is to reduce the dimensionality d of the numerical features of this dataset to $d = 2$ or $d = 3$, so that we can visualize them.

We are going to focus on 3 techniques: PCA, ISOMAP and t-SNE. The algorithms for applying these techniques are all available in the *Scikit-learn* library in *python*.

I. Dimensionality reduction applied to world population data

A. Data import and scaling: download the file `wpd.csv` from *Moodle* and generate a *pandas dataframe* named `wpd` from it using the command `read.csv()`. Use the command `dropna()` to remove observations with missing values.

From `wpd`, generate a matrix named `wpd_data` containing the values of the observations for the 5 following numerical variables: `imr`, `tfr`, `le`, `lem`, `lef`.

You can also generate two vectors named `wpd_area` and `wpd_country` containing, respectively, the area and country names for the observations.

Before proceeding to the application of the dimensionality reduction techniques, you should normalize the data so that the different scales of the variables do not bias the results. To do so, you can use `StandardScaler()` from the module `preprocessing()` of *Scikit-learn*.

B. PCA: Apply PCA to reduce the dimensionality of `wpd_data` to 2 and 3. In *Scikit-learn* the algorithm corresponding to PCA can be found in the module `decomposition`. For details see 1.

1) `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`

Visualize the dataset and indicate with different colors the observations from different areas.

- What are the areas which seem closer to each other in this dataset?
- What are the areas which have the smallest/largest spread?
- Are there areas which are divided in different clusters in the visualization?

C. ISOMAP: Apply ISOMAP to reduce the dimensionality of `wpd_data` to 2 and 3. ISOMAP can be found in *Scikit-learn* in the module named `manifold`. See 2 for details.

2) `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html`

The results given by ISOMAP may vary depending on the number of nearest neighbors used to calculate the length of the shortest-path between data points. Visualize the dataset for different numbers of nearest neighbors. Similarly as for PCA, indicate with different colors the observations for different areas.

- Are there any significant differences between the results of PCA and ISOMAP?

D. t-SNE: A modification of the Stochastic Neighbor Embedding (SNE) method named t-SNE is available is *Scikit-learn*. The differences between the 2 methods mainly rely on 2 points :

- t-SNE uses a symmetrized version of the SNE cost function.
- In t-SNE, the probability for a point to be in the neighborhood of another point is evaluated using the t-Student distribution instead of the Gaussian distribution.

For more details on this algorithm, see 3.

Apply t-SNE to reduce the dimensionality of `wpd_data`. The implementation of this technique can be found in *Scikit-learn*, also in the module `manifold`. See 4 for details.

Similarly to ISOMAP, t-SNE also has a tuning parameter P called perplexity. This parameter plays a similar role as the number of nearest neighbors in ISOMAP. If the perplexity is small, the algorithm preserves only very local neighborhood structure. If it is large, the algorithm tries to preserve the neighborhood structure in a more global manner.

1 - Visualize the dataset for different values of perplexity indicating with different colors observations from different areas.

- Are there any significant differences in the results with respect to the previous methods ?

2 - Using t-SNE, create a visualization with $d = 2$, $P = 20$ and impose a number of iterations of the underlying optimization algorithm to $n = 10000$. Visualize the results in a large figure (`figsize=(20,20)`). Indicate with different colors the observations for different areas. Add the names of the countries to the visualization with *pyplot* command `text()` and analyze the results.

3) `https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf`

4) `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html`