Problem set 12: Dimension reduction

BIO-369

Prof. Anne-Florence Bitbol
EPFL

1 Breast cancer cell features

In Ref. [1], cell samples were obtained by fine needle aspiration from 569 patients with suspected breast tumors. Other tests determined that 212 of these breast masses were malignant (cancer) and 357 were benign (fibrocystic breast masses). In Ref. [1], computer-based analytical techniques were used to measure size, shape, and texture features of the nuclei of the cells. We are going to work on reducing the dimension of this dataset and visualizing it, and we will see how these features can help to distinguish between benign and malignant breast cytology.

a) The data is available in `sklearn.datasets`. In Python, load it by executing:

```
from sklearn.datasets import load_breast_cancer
raw_data = load_breast_cancer()
```

Next, extract the measured features, their names and the labels indicating whether the tumor is malignant (0) or benign (1) by executing:

```
data = raw_data.data
features = raw_data.feature_names
labels = raw_data.target
```

Explore the extracted data, by looking at it and checking that the sizes are the expected ones. What does each row and column in `data` represent? By looking at the names of the features, do you think they have the same units and can be all compared together on the same scale or not? Thus, would you advise to standardize each column of the data?

b) We will work on the correlation matrix of the data, which contains the correlation coefficients between each feature (estimated across all patients). Compute this matrix and visualize it in color. What can you observe in this matrix? Focus on the first feature (feature 0), by looking at the first row: what features are highly correlated with the first one, with a correlation coefficient larger than 0.9? Comment on this. Do the same for the second feature and comment again. Do you think the data has 30 independent dimensions?

c) Find the eigenvalues and eigenvectors of the correlation matrix, and plot the eigenvalues normalized by their sum versus their rank, and next the cumulative sum of these normalized eigenvalues versus their rank. What do these plots represent? What do they tell us about the effective dimension of the data?

d) We will now reduce the dimension of the dataset by retaining only the components of the data on the top two eigenvectors (those associated to the two largest eigenvalues). Because here it is more convenient to work on scaled data, first scale the data by executing:

```
from sklearn.preprocessing import StandardScaler
datascaled = StandardScaler().fit_transform(data)
```

What is the mean and variance of the first column in this scaled data? Next, change the basis of the scaled data to the basis of eigenvectors of the correlation matrix by computing:

```
newdata=np.matmul(datascaled,eigvecs)
```

Finally, plot the first two columns of this transformed data versus one another, coloring the markers using the labels (malignant versus benign). In this plot, where the dimension of the data has been reduced to 2 by PCA, are the two types of cell samples well separated?

e) So far, we performed PCA to reduce the dimension of our dataset to 2 step-by-step. In fact PCA is directly implemented in `sklearn`. Perform it by executing:
```
from sklearn.decomposition import PCA as sklearnPCA
sklearn_pca = sklearnPCA(n_components=2)
newdata2 = sklearn_pca.fit_transform(datascaled)
```
Check that the result is the same as before, e.g. by plotting this new transformed data.

f) At this point, we may wonder whether focusing on the first two PCA directions is really much better than focusing on two real features of the data. In light of your answers to question b), if you were to focus on just two real features of the data, which pair of features would you choose among (0,1) and (0,2)? Plot the data when retaining just this pair of features. Does the separation between the two types of cells look better or worse compared to the PCA case?

g) Now perform a dimension reduction to 3 dimensions instead of two, first by PCA and then by retaining features (0,1,2). Plot the results and discuss their comparison.

h) Here we will explore a nonlinear dimension reduction method, UMAP (with default parameters). You may need to first install UMAP, e.g. via `pip install umap-learn`. To apply UMAP to the data, execute:
```
import umap
reducer = umap.UMAP()
embedding = reducer.fit_transform(datascaled)
```
Look at the shape of `embedding`. This is our data reduced to two dimensions by UMAP. Plot the data points, again distinguishing benign and malignant. Comment.

i) Here we will explore another nonlinear dimension reduction method, t-SNE (with default parameters). To apply it to the data, execute:
```
from sklearn.manifold import TSNE
embedding2 = TSNE(n_components=2).fit_transform(datascaled)
```
Look at the shape of `embedding2`. This is our data reduced to two dimensions by t-SNE. Plot the data points, again distinguishing benign and malignant. Comment.

2 Single-cell RNA sequencing and cell types

In Ref. [2], large-scale single-cell RNA sequencing (RNA-seq) was performed to classify cells in the mouse somatosensory cortex and hippocampal CA1 region. The idea is that cells of different types have different expression profiles of genes, so counting mRNA molecules corresponding to each gene in single cells may reveal their cell types.

a) In Python, load the data in the text file `Data12.txt`. It contains a list of numbers of mRNA molecules observed in 3005 different cells for 19,972 different genes. As the data has multiple rows of headers, you can use `pandas` to load it, and suppress some header rows:
```
import pandas as pd
mrna=pd.read_csv(‘Data12.txt’,delimiter=‘\t’,skiprows=[0,1,2,3,4,5,6,9,10])
```
Next, extract the raw data (counts) and the labels (cell types) by running:
```
mrna_data=(mrna.iloc[1:,2:]).to_numpy(dtype=‘float’)
mrna_labels=list(mrna.iloc[0,2:])
```
Look at the data and check that the sizes are what you expect. What do rows and columns correspond to? Also extract unique labels, they will be useful to label points in our representations of the data.

b) Now we will make a transformation of the data that is usual in the single-cell RNA-seq field. First, add a pseudocount by replacing each count of 0 mRNA molecules by 1. Next, normalize the data for each cell by dividing the number of mRNA molecules for each gene in that cell by the total number of mRNA molecules in that cell. The idea of this normalization is to eliminate cell-specific biases. Finally, take the logarithm of this normalized count of mRNA molecules. Note that the pseudocount is needed to avoid taking logarithms of zero.

c) Apply PCA to the transformed data (be careful about what rows and columns are, we want to compare cells in the end), and plot a two-dimensional representation of the data using the labels (cell types) to color the markers. Comment on the plot obtained.

d) Now apply UMAP to the transformed data from question b) (with default parameters), and plot a two-dimensional representation of the data using the labels (cell types) to color the markers. Comment on the plot obtained and on how it compares to the previous one that used PCA.

e) Perform the same analysis with t-SNE and comment.

References

[1] W. H. Wolberg, W. N. Street, D. M. Heisey, and O. L. Mangasarian. Computer-derived nuclear features distinguish malignant from benign breast cytology. *Hum Pathol*, 26(7):792–796, Jul 1995.

[2] A. Zeisel, A. B. Muñoz-Manchado, S. Codeluppi, P. Lönnerberg, G. La Manno, A. Juréus, S. Marques, H. Munguba, L. He, C. Betsholtz, C. Rolny, G. Castelo-Branco, J. Hjerling-Leffler, and S. Linnarsson. Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347(6226):1138–1142, Mar 2015.