Assignment 2

Machine Learning FCT NOVA
Claudia Soares, Ludwig Krippahl, Ricardo Ferreira

December 1, 2022

1 Dates and Rules

This assignment is for groups of 2 students.

The assignment can be submitted between November 30 and December 18. The deadline for submitting the assignment is December 18, 17:00 Lisbon time. There is an additional tolerance up to December 18 at 23:59 for solving any problems with your submission. This period should be used exclusively for this purpose. No submissions will be accepted after 23:59 on December 18.

You must submit by email to your Tutorial class instructor a zip file containing at least these five files, named exactly as specified (names are case-sensitive):

- TP2.txt This is the questions and answers file;
- TP2.py This is a Python 3.x script that can be used to run your code for this assignment;

In addition, some questions require other files. You can include images and reports. Optionally, you can include other python modules if you wish to separate your code into several files.

Do not include the original (bacteria) image files. These are not necessary and can make the archive significantly larger. Just assume that these will be in a folder named ./images/ when your reports and code are evaluated.

2 Weakly supervised learning: helping biologists labeling cells.

The goal of this assignment is to examine a set of bacterial cell images using machine learning techniques, including feature extraction, feature selection, and clustering, to help biologists organize similar images. In this zip file, tp2.zip, you have a set of 563 PNG images (in the ./images/ folder) taken from a super-resolution fluorescence microscopy photograph of Staphylococcus aureus, a common cause of hospital infections and often resistant to multiple antibiotics.

The images provided for this assignment were obtained by automatic segmentation and included cells in different stages of their life cycle and segmentation errors, i.e., not corresponding to real cells. Figure 1 shows a sample of the images provided.

In this assignment, you will load all images, extract features, examine them and select a subset for clustering to reach some conclusion about the best way of grouping these images.

3 Implementation

In the tp2.zip file provided, there is a Python module, tp2_aux.py, with a function, images_as_matrix(), that returns a 2D NumPy array with one image per row (563 rows) and one pixel per column (50x50=2500 columns) from the images in the images folder.

From this matrix, you will extract features using three different methods:

Principal Component Analysis (PCA) Use the PCA class from sklearn.decomposition.

Kernel PCA (kPCA) Use the kPCA class from sklearn.decomposition, with an "rbf" kernel. When creating an object of this class, for otherwise, the kPCA constructor will use a faster, approximate, computation.

Isometric mapping with Isomap Use the Isomap class from the sklearn.manifold module.

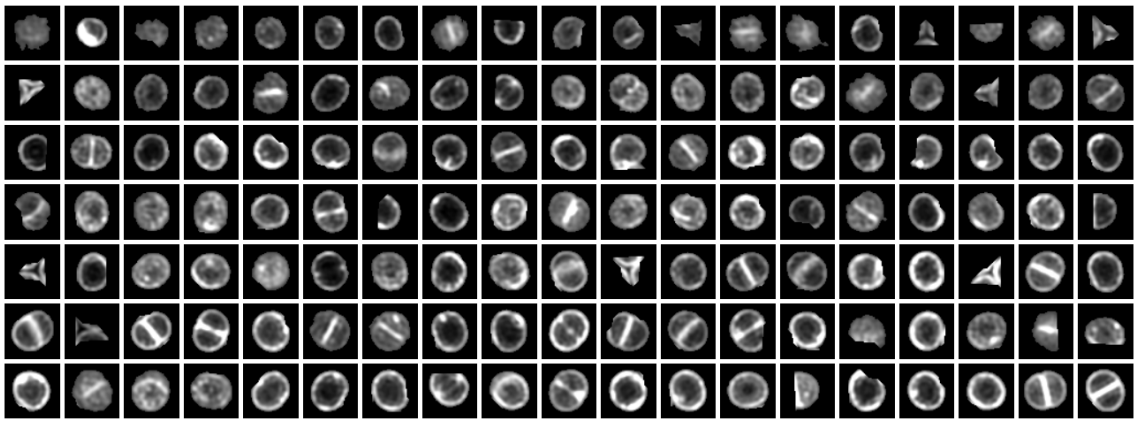

Figure 1: All images have the same dimensions, 50 by 50 pixels, with a black background and the segmented region centered in the image.

With each method, extract six features from the data set, for a total of 18 features. In addition to the images, the labels.txt has information on identifying the cell cycle phase for each cell. The first column has the cell identifier and the second column has a label for the cell cycle phase. These cells were manually labeled by biologists. The figure below illustrates examples from the 3 phases, labeled with integers 1, 2, and 3. The first phase is before the cell starts to divide, the second covers the first part of the division, with the formation of a membrane ring where the cells will divide, and the third phase corresponds to the final stage of cell division. However, note that only some images are labeled. Images that were not labeled have a label value of 0 in this file.

For this assignment, you will parameterize and compare three clustering algorithms: Hierarchical (agglomerative) clustering, Spectral Clustering, and K-Means. When instantiating the Spectral Clustering object from Scikit-Learn use the option assign_labels="cluster_qr".

Examine the performance of each algorithm by varying the main parameter of each one (neighborhood distance and the number of clusters k; you can leave the other parameters with their default values) and using an internal index, the k-means loss, and external indexes computed from the labels available: purity, Precision, Recall, the F1 measure, and the Rand index.

Finally, select a subset of parameter values for closer examination by visually inspecting the clusters generated. For this, you can use the report_clusters(ids, labels, report_file) function in the tp2_aux.py module. Considering all the information gathered at this stage, recommend a procedure for the biologists to help them process the segmented images, both for cell classification and to help discard segmentation errors.

4 Optional: Divisive clustering

Implement the bisecting K-Means hierarchical clustering algorithm, as described in the lectures. This can be done using the KMeans classifier available in the Scikit-Learn library to split each cluster into two sub-clusters with k = 2. Repeat this process by splitting the cluster with the most examples in each iteration for a predetermined number of iterations. The output should be a list of lists, each corresponding to one example and listing all cluster labels to which the example was assigned in order. Here is an example of bisecting K-Means for three iterations on five examples. The first example was placed on a cluster of index 1 in the first iteration, with the remainder on the cluster of index 0. Then the third example was placed in the sub-cluster of index 1, and the other three in the sub-cluster of index 0. Of these, the second and fourth examples were placed in sub-sub-cluster 0 ([0, 0, 0]) and the fifth example in sub-sub-cluster of index 1 ([0, 0, 1]):

```
[ [1],
  [0, 0, 0],
  [0, 1],
  [0, 0, 0],
  [0, 0, 1] ]
```

If you want to examine the clusters generated after implementing the bisecting K-Means algorithm, you can use the function report_clusters_hierarchical(ixs,label_lists,report_file). This function works similarly to the report_clusters function but expects the labels to be a list of lists as described above.

5 Implementation details

When testing different parameters for your clustering algorithms, note that the used scores can only be computed if you have at least 2 clusters. If all data is placed in the same cluster, the program will raise an exception.

NumPy arrays have a sort method for sorting in place. This sorts the array from smallest to largest values, but you can invert any array by simply slicing it this way: a = a[::-1].

The feature extraction steps in this assignment can take a few minutes. If you need to run your code many times for experiments it may be useful to save the extracted features to a file. A simple way of doing this is is to use the savez and load functions in NumPy.

To view your clusters, you can use the function report_clusters(ids, labels, report_file) available in the tp2_aux.py module. This function saves an HTML file with the clusters indicated by the labels argument. This HTML file assumes there is a folder images/ with the images provided. The first argument is a list or 1D array with the identifiers for the images to show, and the second argument is a list or 1D array with the cluster labels for the images in the same order. The last argument is the name of the html file where the report will be saved. The file example_labels.html shows an example of a report (in this case, created with the manual labels as clusters).

Read the questions carefully and give clear and concise answers. The main focus of this assignment should be the selection of the best features and a discussion of the advantages and disadvantages of each algorithm for this dataset, informed by an analysis of their behavior with different parameters and the different scores used.

Plagiarism

Plagiarism is any attempt to make you seem the author of text, code or any work which is not really yours. Plagiarism breaks the necessary trust for fair evaluation and any student submitting plagiarized work will fail the course immediately. If you rely on material other than what was provided in this course you must identify it and credit your sources. Also note that you will be graded on the work you actually did, so even if you avoid plagiarism by crediting your sources you still need to do your own implementation.