Clustering

Mathieu Lagrange

This document, the code and the data are available in the **tpClustering** directory:
https://github.com/mathieulagrange/master-learn

The aim of this practical is to familiarize yourself with some standard clustering techniques.

1 Data

We will consider two types of data, one synthetic and the second one taken from the dataset.

1. For the synthetic one, sample 1000 points in a 2 dimensional space split into 10 non overlapping clusters.
2. For the realistic one, please use the dataset available on the repository.

2 Metric

In this practical, we will consider the adjusted mutual information (AMI) score. Please use the **metrics.adjusted_mutual_info_score** available in scikit-learn.

$AMI(x, y)$ is zero when the two clusterings x and y completely differ, and 1 if they are equal up to a permutation of the labels.

3 Euclidean space clustering

1. Use the kmeans algorithm (**sklearn.cluster.KMeans**).
2. What is the AMI for the synthetic and real datasets ?

4 Model selection

1. Implement the gap statistic criterion. You can use the **inertia_** given by kmeans as the loss function.
2. Verify that the statistic performs well on the synthetic dataset.
3. Increase the overlapping between the clusters. Is the statistic robust ?
4. Apply the statistic to detect the number of clusters for the real data. Is the detected number of clusters correct ?

5 Non euclidean clustering

Some applications require that the input of the clustering algorithm is no longer the actual data points but their dissimilarities. Provided that the matrix is semi definite positive, the spectral clustering algorithm can be considered.

1. Consider a radial basis function to compute the dissimilarity matrix:

$$rbf(x,y) = e^{-\frac{|x-y|^2}{2\sigma^2}}$$

2. Implement the spectral clustering algorithm using the eigenvalue and the kmeans implementation. To do so, please implement each of those steps:
   (a) compute the Euclidean distance matrix of dimensions (nbItems x nbItems)
   (b) compute the weight matrix W using the radial basis function
   (c) compute the degree matrix D
   (d) compute the laplacian $L = D - W$
   (e) perform the eigenvalue decomposition of the Laplacian
   (f) select the k eigenvectors with highest absolute eigenvalues as clustering indicators, k being the number of clusters.
   (g) perform kmeans using those k eigenvectors as input
3. Verify that the eigenvector decomposition of the laplacian of the synthetic dataset is step wise.
4. Is it the same for the real dataset ?
5. Compare the results with the reference implementation available **sklearn.cluster.SpectralClustering**.
6. Optimize σ for best performance.

6 Conclusion

1. Present a table compiling the performance achieved by the different clustering algorithms on the two datasets.

A Report

Please send a commented version of your jupyter notebook. The report shall have for each question a brief description about the way things have been done and some discussion about the resulting behavior.

Send it no later than a week after the session to `mathieu.lagrange@ec-nantes.fr`, with the `[ECN]` flag within the title of the message.

B Help

The relevant libraries for this practical are:

- numpy: numeric arrays
- scipy: numeric computations
- matplotlib: data display
- scikit-learn: learning toolkit, in particular the clustering section (`https://scikit-learn.org/stable/modules/clustering.html`)
- from MATLAB to numpy: `http://mathesaurus.sourceforge.net/matlab-numpy.html`

References

[1] Ulrike von Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.

[2] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107, 2000.