Statistical data analysis
Spectral Clustering

Introduction

The aim of this tutorial is to get familiar with spectral clustering. You will use available building blocks and implement the algorithm of spectral clustering. You will apply this algorithm to the provided input data and compare the result with the known annotation as well as with the outcome of classic k-means algorithm. You will perform the comparison for different parameterizations of the input data and spectral clustering settings.

1 Input Data

In this tutorial, we will work with two synthetic datasets available online at `https://cs.joensuu.fi/sipu/datasets/`. First of all, we start with the Spiral dataset (`spiral.txt`) containing three well-separated spirals that is depicted in Figure 1. Second one, and more ”realistic”, is the Jain's toy problem (`jain.txt`) with two clusters. For loading of these datasets, you can use the existing function `LoadDataset` in `spectral_clustering_v1.2_rmd.Rmd` file.

2 k-means Algorithm

Here, we deal with a clustering task with non-compact clusters. K-means is likely to generate clusters different from the gold standard (the application of R function *kmeans* directly on the input data can be seen in Figure 2). Since k-means algorithm is also the last step of spectral clustering, one of our aims is to find out how the transformation performed in spectral clustering before the application of k-means algorithm influences its output.

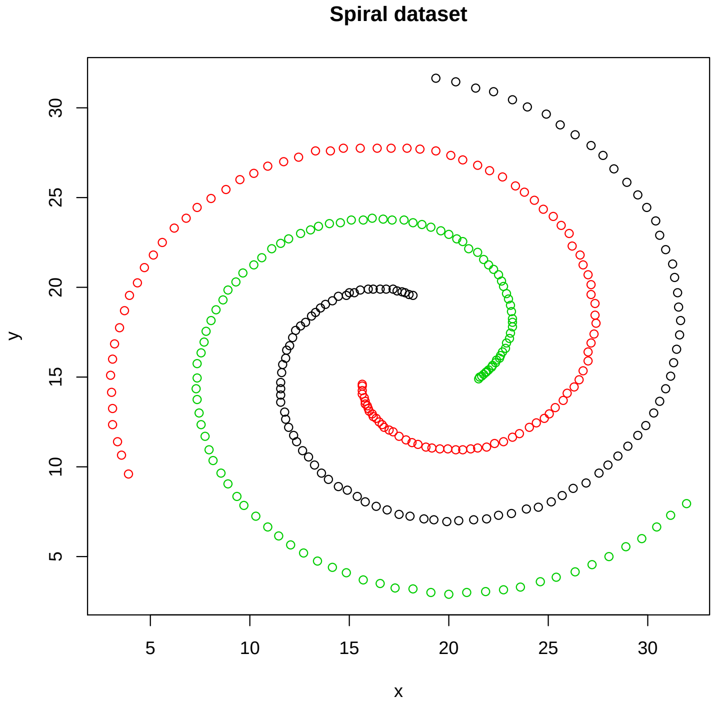

Figure 1: Spiral dataset with original labels.

3 Spectral Clustering

Spectral clustering can be divided into a few building blocks. The following list describes these functional blocks.

- **Computation of similarity matrix $\mathcal{S}$:** `CalcSimMatrix` function calculates Euclidean distances between pairs of points and afterwards applies Gaussian kernel, modification is not required, but it is good to verify the importance of parameter σ (a higher value means greater similarity between more distant points, i.e., similarity is less local).

- **Construction of similarity graph:** there are three popular different similarity graphs, i.e., fully connected graph, ϵ-neighborhood graph (`BuildEpsilonGraph` function), and k-nearest neighbor graphs. Fully connected graph, as a most trivial variant, keeps $\mathcal{S}$ without changes; ϵ-neighborhood graph connect all points whose pairwise distance are smaller then ϵ and k-nearest neighbor graph connect vertex v_i with vertex v_j if v_j is among the k-nearest neighbors of v_i. However, this leads

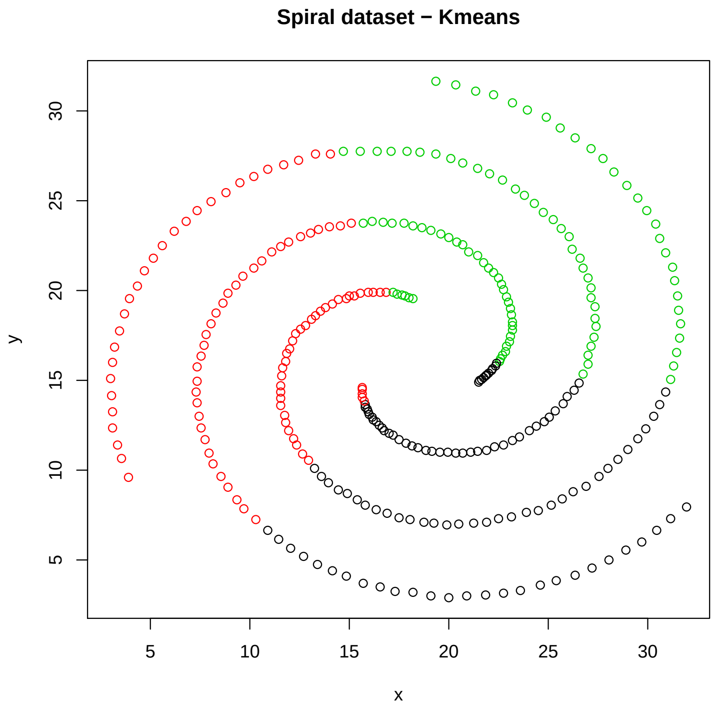

Figure 2: Kmeans clustering with $K = 3$.

to non-symetric relationships and the similarity graph is `directed` (`BuildDirectedKNNGraph` function). To making similarity graph undirected (`BuildUndirectedKNNGraph` function), we can apply two approaches. First variant connect v_i and v_j if v_i is among the k-nearest neighbors of v_j or if v_j is among the k-nearest neighbors of v_i. Second variant connect vartices v_i and v_j if both v_i is among the k-nearest neighbors of v_j and v_j is among the k-nearest neighbors of v_i. In the second case, the graph is called the `mutual k-nearest neighbor graph` (see [1]).

- **Derivation of Laplace matrices $\mathcal{L}$ with subsequent projection to the space of their k smallest eigenvectors:** `CalcLaplacian` function performs this step for non-normalized Laplacian (see [1]).

- **Application of k-means algorithm to the output of the previous function:** a straightforward application of R function `kmeans`.

4 Step by Step

You should go through the following steps:

1. read `spiral` and `jain` datasets,
2. perform clustering using k-means algorithm, evaluate its success rate visually using function `plot` and numerically using function `Purity`,
3. create the basic variant of the algorithm of spectral clustering from the existing functions,
4. perform clustering using spectral clustering, check the connectivity graph by the function `PlotConnectedGraph`, check results numerically using the function `Purity` and visually by R function `plot`,
5. repeat Step 4 with different variants of the spectral clustering algorithm (different variants of similarity graph),
6. summarize your experience from experiments (what option is important, what option does not have influence, etc.).

5 Evaluation

- BuildEpsilonGraph function [**0.5p**],
- BuildDirectedKNNGraph function [**1p**],
- mutual/unmutual version of BuildUndirectedKNNGraph function [**1p**],
- CalcLaplacian function [**0.5p**],
- presentation of experimental results of the two datasets [**1p**].

6 Submission Form

Submit your solution to the upload system. Submit only the extended rmarkdown file, change its name as follows: *spectral_clustering_$YOURFELUSERNAME.Rmd*. No changes that go beyond the required file extensions are permitted (changes in the input datasets and links to them, extra libraries, etc.).

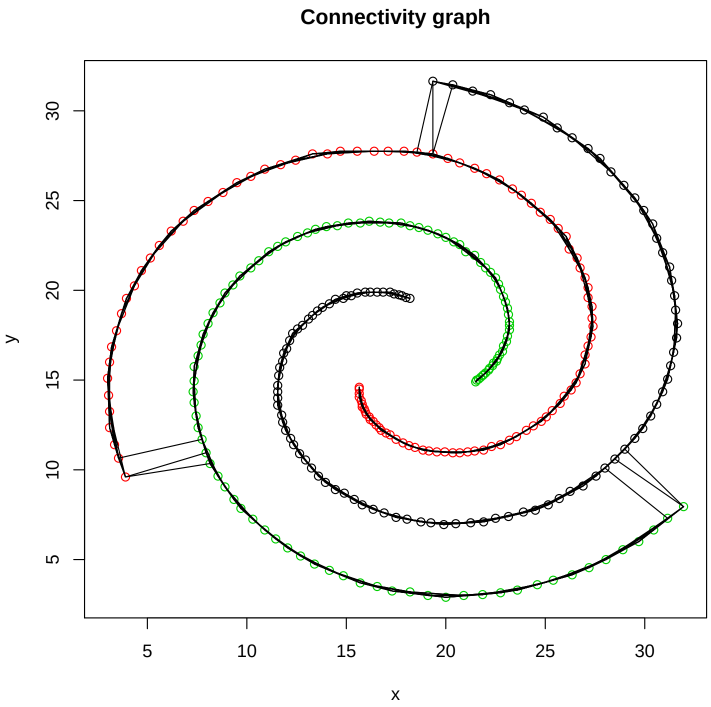

Figure 3: Almost ideal connectivity graph for Spiral dataset.

References

[1] Luxburg, Ulrike: *A tutorial on spectral clustering*, Statistics and Computing, 17/4, pp. 395–416, 2007.