Unsupervised learning: Clustering

Pattern Recognition and Machine Learning - 2017

Davide Abati

September 28th, 2017

University of Modena and Reggio Emilia

Agenda

K-means

Spectral clustering

K-means

K-means

- Is a partitional clustering model
  - splits data $\{x_i\}_1^n$ into k disjoint sets
  - the number of sets k has to be provided as input
- solves the following optimization problem:

$$\underset{\{c_1,\ldots,c_k\}}{\arg\min} = \sum_{j=1}^{k}\sum_{i=1}^{n} \mathbf{I}(i,j)||x_i - c_j||^2$$

$$\mathbf{I}(i,j) = \begin{cases} 1, & x_i \text{ belongs to cluster } j \\ 0, & \text{otherwise} \end{cases}$$

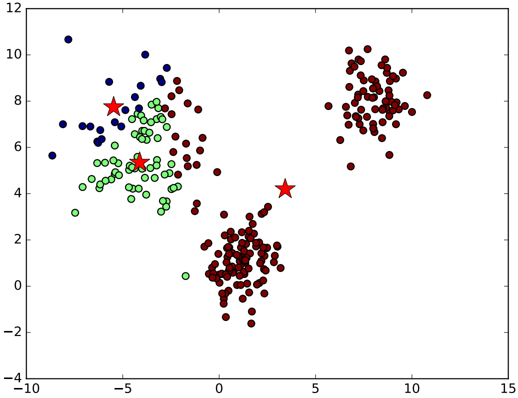
K-means: algorithm

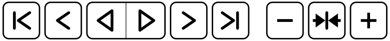
The problem is NP-hard. A simple heuristic algorithm can be employed to converge to a *local* minimum:

- Initialize k centers randomly
- Repeat until convergence:
  - assign each example to the closest center (i.e. lower euclidean distance)
  - re-estimate centers as the mean of their clusters

Try to implement it from scratch!

K-means: iterations





Application: color segmentation

K-means can be employed for image segmentation, simply by grouping pixels in the color space. You can also add coordinates to each pixel to obtain a smooth output.

Image

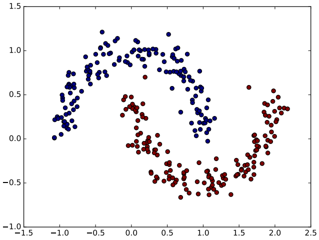
Segmentation

Try it!

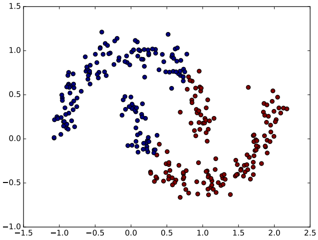
Kmeans: limitations

- it can get stuck into bad local minima
  - OPTIONAL: run the algorithm many times and choose the most recurrent solution
- can only be employed in spaces where the mean operation is defined
- due to its cost function, it can only cope with compact ball-shaped clusters

GT



Result



Spectral clustering

Spectral clustering: algorithm (1)

Clustering model based on the spectral graph theory.

- build a graph over examples, representing it with the adjacency matrix A

$$A_{i,j} = e^{-\frac{\sum_{k=1}^{d} ||x_i^k - x_j^k||^2}{\sigma^2}}$$

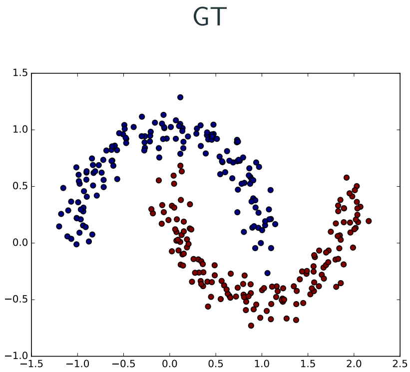
- build the degree matrix D of the graph. It is a diagonal matrix holding for each element the sum of the incoming adges.
- compute the normalized laplacian L

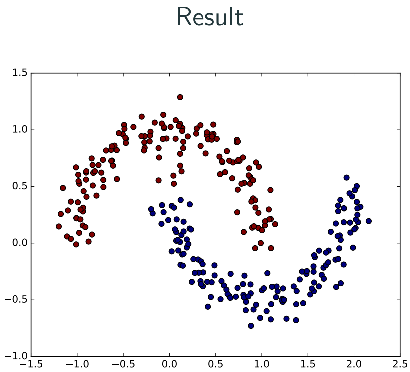
$$L = I - D^{-0.5} A D^{-0.5} \tag{1}$$

Spectral clustering: algorithm (2)

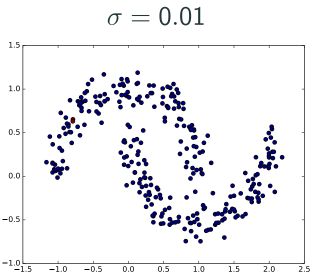
- Compute the eigenvectors and sort them for increasing eigenvalues
- Choose the eigenvectors <u>from the second to the desired number of clusters</u>
- Those eigenvectors provide a representation of data in a fancy embedding space: run K-means over such eigenvectors.

Spectral clustering: result





Beware: σ counts (in large amounts!)



Useful functions

- datasets.gaussian_dataset

```
data, cl = gaussians_dataset(3, [100,100,70], [[1, 1],[-4, 6],[8, 8]], [[1, 1],[3, 3],[1, 1]])
```

- datasets.two_moon_dataset

```
data, cl = two_moon_dataset(n_samples=300, noise=0.1)
```

- matplotlib.pyplot.plot
- matplotlib.pyplot.scatter
- scipy.linalg.fractional_matrix_power
- numpy.linalg.eig
- numpy.argsort