11752 Machine Learning
Master in Intelligent Systems
Universitat de les Illes Balears

Handout #5: **Unsupervised Learning** (optimization-based clustering)

NOTE 1:

- Scikit web pages on **clustering evaluation**[1] may be useful.
- In particular, the following objects/functions of `scikit-learn` will be necessary:

  `sklearn.metrics.v_measure_score`

  `sklearn.metrics.cluster.contingency_matrix`
- You can make use of other functions from scikit-learn or any other Python library which may be useful.

P7. (a) Following the pseudocode at slide 7 of the lecture notes, write a **crisp clustering function** for circumference-shaped clusters, 2D samples and assuming that the true circumferences are centered at point (0,0). This function must match the following definition:

```
def do_crisp_clust(X, M, n_iter, n_attempts, eps)
```

where: X is the dataset, M is the number of clusters, *n_iter* is the maximum number of iterations per attempt, *n_attempts* is the number of attempts to perform (to counteract the random initialization of the parameters of the clusters, $\Theta(0) = \{\theta_j(0)\}$) and *eps* is such that the clustering is stopped as soon as $J(t) - J(t-1) < eps$. The clustering leading to the best final value of the cost function J has to be returned together with the corresponding value of J and the resulting set of cluster parameters θ.

See the appendix for the description of the proximity function to use and the derivation of the calculation of the cluster parameters.

(b) **Given dataset** `dsxx7.txt`:

Using the crisp clustering function, set $M = 3$ and e.g. 5 attempts for clustering, adequate values for the maximum number of iterations, e.g. 20-30, and for the termination criterion value, e.g. 10^{-3}, and

- plot the value of J along the iterations performed for the best clustering (that one leading to the lowest value of the cost function J)
- plot separately the dataset using the true labelling and the labelling derived from the best clustering
- find the contingency table, calculate the number of samples which can be considered as incorrectly clustered and calculate the V-measure.

NOTE 2: The previous problem requires loading dataset `dsgg7.txt` where `gg` is the group number:

```
import numpy as np
group = '01' # assuming group 1
ds = 7       # dataset 7
data = np.loadtxt('ds'+group+str(ds)+'.txt')
X = data[:, 0:2]
y = data[:, 2]
```

Class labels are 0, 1, 2, etc. All datasets include the true labelling for all samples.

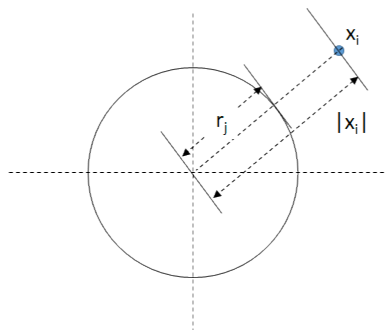
- A report of the work done has to be released by February 16, 2022 in electronic form as a notebook file (.ipynb).

[1] `https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation`

- Provide the requested data and plots/figures at each point above. For figures, use appropriate titles, axis labels and legends to clarify the results reported.
- Suitable <u>comments</u> are expected in the source code.
- This work has to be done individually (see the number of group in *Aula Digital*).

Appendix 1: distance between a point and a circumference of radius r

Let us consider a circumference of radius r centered at point (0,0), such as the following one:



Given a point $x_i = (x_{i1}, x_{i2})$, the squared (radial) distance d_r^2 between x_i and the circumference of radius r can be defined as:

$$d_r^2 = (\|x_i\| - r)^2$$

Consequently, for this problem, we can define as proximity function between a sample x_i and a cluster C_j described by its radius r_j:

$$\wp(x_i, C_j) = (\|x_i\| - r_j)^2$$

(We consider the squared distance in order to get rid of the sign of the subtraction.)

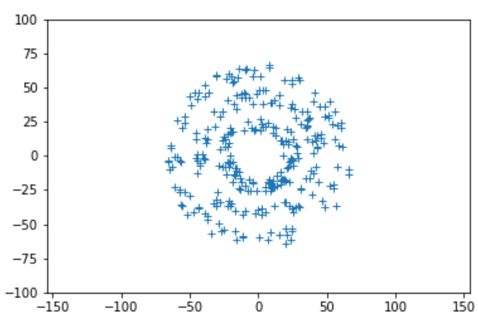
Step 3.3 of GHAS involves *solving for* $\theta_j(t+1) = r_j(t+1)$ *in* $\sum_{i=1}^{N} u_{ij}(t) \frac{\partial \wp(x_i,\theta_j)}{\partial \theta_j} = 0$. For this case:

$$\frac{\partial \wp(x_i, \theta_j)}{\partial \theta_j} = -2\left(\|x_i\| - r_j\right)$$

$$\sum_{i=1}^{N} u_{ij} \frac{\partial \wp(x_i, \theta_j)}{\partial \theta_j} = 0 \Rightarrow -2 \sum_{i=1}^{N} u_{ij}(\|x_i\| - r_j) = 0 \Rightarrow \sum_{i=1}^{N} u_{ij}\|x_i\| - \sum_{i=1}^{N} u_{ij} r_j = 0 \Rightarrow r_j = \frac{\sum_{i=1}^{N} u_{ij}\|x_i\|}{\sum_{i=1}^{N} u_{ij}}$$

Appendix 2: Example of dataset

`dsxx7` (unlabelled)



`dsxx7` (labelled)