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

Handout #7: **Instance-based and unsupervised learning**
(graded assignment)

T1. a) Using a quadratic programming solver and the **Wolfe dual representation**, find a linear SVM for dataset `ds`*gg*`-t1a.txt`. Next

i. report the *support vectors* (NOTE: due to round-off errors, it is likely that none of the λ_i are exactly 0, but close, e.g. 10^{-6})

ii. report the resulting *decision function* $g(x) = w^T x + w_0$ [1]

iii. plot the *classification map*, i.e. evaluate the *decision function* for a 'regular' subset (grid) of points of the feature space, and plot superimposed the *training samples*, highlighting the *support vectors*. Use different markers and/or colours for each class.

b) Using the transformation $\Phi(x) = (x_1 x_2, x_1^2 + x_2^2)$ and a quadratic programming library, solve dataset `ds`*gg*`-t1b.txt` by means of a non-linear SVM formulation. Next

i. report the *support vectors* (NOTE: due to round-off errors, it is likely that none of the λ_i are exactly 0, but close, e.g. 10^{-6})

ii. report the resulting *decision function* $g(x) = w^T \Phi(x) + w_0$ [1]

iii. plot **in the original feature space** the *classification map*, i.e. evaluate the *decision function* for a 'regular' subset (grid) of points of the feature space, and plot superimposed the *training samples*, highlighting the *support vectors*. Use different markers and/or colours for each class.

T2. Consider the dataset `ds`*gg*`-t2.txt` and cluster it using the *Ward*, *K-means* and the *Fuzzy K-means* algorithms for $m = 2$, 3, 4 and 5 clusters. Report on the performance attained in each case using the *v-measure*.

Next, for the best algorithm:

(a) show the *contingency matrix*,

(b) determine the assignment of classes to clusters, i.e. which cluster corresponds to which class,

(c) identify the number of incorrectly clustered samples and calculate also the percentage of errors as *number of incorrectly clustered samples / total number of samples*,

(d) report also on the *homogeneity* and the *completeness* measures.

NOTE 1: To load datasets `ds`*gg*`-t1a.txt`, `ds`*gg*`-t1b.txt` and `ds`*gg*`-t2.txt`, where `gg` is the group number:

```
import numpy as np
gg = 1   # assuming group 1
ds = 'a' # assuming task T1.a
data = np.loadtxt('ds%02d%c.txt' % (gg,ds))
X = data[:, 0:2]
y = data[:, 2]
```

Class labels are 1 for ω_1 and 0 for ω_2.

NOTE 2: Regarding the quadratic programming solver referred above, you are advised to use the Python library `cvxpy` (`https://www.cvxpy.org/`). This library can be installed by means of:

[1] In https://jupyterbook.org/content/math.html you can find tools for typesetting mathematical expressions in notebooks

```
pip install cvxpy                  # if you use pip
conda install -c conda-forge cvxpy # if you use conda (inside an anaconda installation)
```

NOTE 3: For the *Ward* algorithm use the implementation of the *hierarchical agglomerative clustering* method available in *scikit-learn*.[1]

For the *K-means* and the *Fuzzy K-means*, use the implementations available in the adaptation of the *fuzzy_kmeans* library that can be downloaded from the course web page. Have a look at the implementation to understand how to make use of it.

NOTE 4: Scikit-learn web pages on **clustering methods**[2] and **clustering evaluation**[3] will be useful for this assignment. In particular, the following objects/functions of `scikit-learn` will be necessary:

```
sklearn.metrics.cluster.contingency_matrix
sklearn.metrics.v_measure_score
sklearn.metrics.homogeneity_score
sklearn.metrics.completeness_score
```

DELIVERY INSTRUCTIONS:

- Delivery date: February 15, 2026. You have to deliver the notebook (.ipynb file) and the corresponding PDF file inside a ZIP file.
- Provide the results requested and suitable comments in the source code.
- This work can be done in groups of two people (same groups as for the first assignment) or individually. The list of group numbers are available in the course web page.
- IMPORTANT NOTICE: An excessive similarity between the reports/source code released can be considered a kind of plagiarism.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html
[2] https://scikit-learn.org/stable/modules/clustering.html#clustering
[3] https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation