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

Handout #2: **Instance-based Learning**

NOTE 1: Problem P2 requires loading dataset `dsxx2.txt` where `xx` is the group number:

```
import numpy as np
group = '01' # assuming group 1
ds = 2       # assuming problem 2
data = np.loadtxt('ds'+group+str(ds)+'.txt')
X = data[:, 0:2]
y = data[:, 2:3]
```

Class labels are 1 for ω_1 and 0 for ω_2.

NOTE 2: Problem P2 requires the use of a Quadratic Programming solver, which can be obtained from library `qpsolvers` (`https://pypi.org/project/qpsolvers/`). This library can be installed by means of:

```
pip install cvxopt --user
pip install qpsolvers
```

When calling function `solve_qp`, choose solver `'cvxopt'`.

NOTE 3: Problem P2 also requires the use of `scikit-learn` (`https://scikit-learn.org`) and `matplotlib` (`https://matplotlib.org/`).

P2. **Given dataset** `dsxx2.txt`:

a) Mapping the *training samples* onto an alternative 2-dimensional space using $\Phi(x_1, x_2) = (x_1 x_2, x_1^2 + x_2^2)$, solve for the SVM analytically using a quadratic programming solver and

1. find and report the *support vectors* in the original space (NOTE: due to round-off errors, it is likely none of the λ_i are exactly 0, but close, e.g. 10^{-6}); and
2. calculate and report the resulting *decision function* both in the transformed space $g_1(x') = w^T x' + w_0$ $[x' = \Phi(x)]$ and in the original space $g_2(x) = w^T \Phi(x) + w_0$[1].

b) Generate the following plots:

1. a first plot with the *training samples* in the transformed space, highlighting the *support vectors* and plotting the 2D *decision curve*;
2. a second plot with the *training samples* in the original space, highlighting the *support vectors* and plotting the 2D *decision curve*; and
3. a third plot with the *classification map* in the original space, i.e. evaluate the *decision function* for a 'regular' subset (grid) of points.

Use different markers and/or colours for each class. See the appendix for examples of the requested plots.

c) Compare the results obtained with the ones resulting from the `scikit-learn` `SVC` object: i.e. report the *support vectors* returned by `SVC` and the corresponding *decision function*, and provide the same kind of plots requested before.

NOTE: the SVC object solves the soft-margin kernel-based problem, hence you will have to supply the kernel specified in a) –use either `kernel = 'precomputed'` and compute the *gram matrix*, or supply a *callable* kernel when invoking the SVC object constructor– and set constant C with a high value, e.g. 10^{16}, to force a perfect classification of the training set.

[1] See https://jupyterbook.org/content/math.html for typesetting mathematical expressions in notebooks

d) Also by means of the `scikit-learn` `SVC` object, repeat point c) for the `'rbf'` kernel ($\gamma = 1$). Additionally, draw the corresponding RBF network (slide 42 of the SVM lecture notes), replacing $K(x_i, x)$, λ_i and y_i by your values.

- A report of the work done has to be released by December 12, 2021 in electronic form as a notebook file (.ipynb).
- 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 comments are expected in the source code.
- This work has to be done individually (see the number of group in *Aula Digital*).