**Pattern Recognition**
**Department of Electrical and Information Engineering**
**University of Cassino and Southern Latium, Second Semester 2018**
**Homework Assignment 3**
**Assigned 30 may 2018; due 11:59pm, 9 June 2018**

**Problem 3.1** [60%] Consider the dataset contained in the file `hw3data.csv` available on the course site. It contains 8,000 samples coming from a two–class problem, each made of 10 numerical features and a binary label ($\pm 1$). Consider the following tasks:

(a) Split the data into a training set (40%), validation set (40%) and test set (20%).

(b) Use the training set for building a $\nu$-SVM and the validation set for choosing the optimal value (i.e. the one maximizing the AUC) of $\nu$.

(c) Evaluate the AUC on the test set.

Execute the steps above with 4 different SVM kernels (linear, polynomial of degree 2, RBF, sigmoid). For each test, consider at least 5 different splits and evaluate average AUC and standard deviation. Discuss the results obtained.

**Hint:** when you search the optimal value of $\nu$, remember that it is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors relative to the total number of training examples. For example, if you set it to 0.05 you are guaranteed to find at most 5% of your training examples being misclassified (at the cost of a small margin, though) and at least 5% of your training examples being support vectors.

**Problem 3.2** [40%] Use the dataset of Problem 3.1 and perform several splits into a training set and a test set (also with different sizes) to determine which combination of options among the ones you considered in Problem 3.1 ensures the highest AUC. Once you picked out the best model, save (see note below) and submit it together with your report. Your model will be run on a separate matrix containing new test data. Your grade will be based on the performance of your classifier on the new test data, which will contain a very large number of examples generated from the same distribution.

**Note:** to save your model you can use the method `joblib.dump(clf, 'mymodel.pkl')` from the module `joblib`. Example:

```
# Import the joblib module
from sklearn.externals import joblib
...
# Train the SVM
clf = svm.SVC( ...
clf.fit(X, y)
...
# Save the model in the file 'mymodel.pkl'
joblib.dump(clf, 'mymodel.pkl')}
```