Exercise 4: Kernels, decision trees, model selection, and statistical validation.

I. Goal of the exercise

The goals of this exercise are:

1) Understand the basis of kernels and implement a kernelized version of SVM.
2) Show how to build and use cross-validation.
3) Understand the importance of statistical comparison and use some methods for hypothesis testing.

II. Deliverables

As you progress in this exercise, you will find several questions. You are expected to answer them properly with adequate figures when required and deliver a document with all these evidences in due time. A file or files with the working code used for generating and discussing the results must be also delivered.

III. Your first kernel

Implement the Radial Basis Function kernel and modify the SVM code from last exercise to support it. The RBF kernel is parameterized by σ and it is defined as follows:

$$k(x_i, x_j) = e^{\frac{-\|x_i - x_j\|_2^2}{2\sigma^2}}.$$

Remember that in order to kernelize the method we need to create the Gramm matrix, $\mathbf{K}$, using all the N examples of the training set,

$$\mathbf{K} = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots \\ k(x_2, x_1) & k(x_2, x_2) & \dots \\ \vdots & \vdots & \ddots \end{pmatrix},$$

and using the Representer's theorem, the evaluation of the final model in the primal on a sample point x is,

$$h(x) = sign(\sum_{i=1}^{N} \alpha_i k(x_i, x)),$$

and in the dual

$$h(x) = sign(\sum_{i=1}^{N} y_i \nu_i k(x_i, x)),$$

Question block 1:

1) Load the dataset `'example_dataset_1.mat'`.
2) Create the Gram matrix for $\sigma = 1$ and plot it (you may use `imagesc` to display the matrix and `L2_distance` for computing distances between points).
3) Describe the Gram matrix displayed. Which are the maximum values of the matrix? And the minimum? Is it positive definite (check if all the eigenvalues are positive)?
4) Modify the learning code of the dual SVM to handle the kernel. Do not use the offset b.
5) Run your training algorithm on that dataset with $\lambda = 1$ and $\sigma = 1$.
6) Plot the dataset and the separating hyperplane. What do you observe?

IV. Working with decision trees.

We will use a MATLAB built-in implementation of the decision tree `classregtree`.

Question block 2:

1) Consider that you train a full decision tree. What is the expected training error to obtain? Why?
2) Load the dataset `'example_dataset_1.mat'`.
3) Train a tree using the default parameters and plot the result on the training set. Compute and report the training error? Is the result what you expected?
4) Find the parameterization of the tree that allows you to build the full tree and plot the result on the training set. Which is the value of the training error?

V. CROSS-VALIDATION.

In this section you will code a K-fold cross validation process. When comparing classifiers it is important to reduce the variability due to the random sampling process. For this reason, it is important to store each of the folds created. The simplest way of doing that without replicating the dataset is to store the indexes in each of the partitions.

Question block 3:

1) Load the dataset `'example_dataset_1.mat'`.
2) Create a function that given a dataset creates the K folds by storing the indexes corresponding to training and test for each of the folds.
3) Compute and report each class frequency value for the original problem and for each of the training and test partitions executing your code using K=10.

VI. MODEL SELECTION.

Regularization parameters such as λ (C) and σ in SVM and the number of nodes in the decision tree are found by cross-validation. In this section we want to select those parameters. In the case of the decision tree we will set the parameter `minparent` as a node control mechanism.

Question block 4:

1) Load the dataset `'example_dataset_1.mat'`.
2) Use the cross-validation function to create a set of 5-fold indexes.
3) Use the cross-validation set created for finding the best parameters of an RBF SVM. Plot the average error surface for each set of parameters. You may use a grid search, i.e. define a set of parameters to be tested and select the best one.
4) Use the cross-validation set created for finding the best value for `minparent`. Plot the average error surface for each parameter value
5) Show the best validation errors obtained.

VII. MODEL SELECTION AND OUT-OF-SAMPLE ERROR ESTIMATION.

In this section we want to compare the prediction performance of the model, i.e. we want to find E_{out}. For this task, we will use two different strategies. First, we will set 1/5 of training data for test and then run a cross-validation for model selection. Second, we will use nested cross-validation. Remember that nested cross-validation runs an external K-fold cross validation

in order to estimate E_{out} and for each fold of the external cross-validation runs another cross-validation (internal cross-validation) that is used for model selection.

Question block 5:

1) Load the dataset `'example_dataset_1.mat'`.
2) Split the data set in train and test, using 1/5th of the data for testing purposes.
3) Use the cross-validation function to create a set of 5-fold indexes on the training set.
4) Find the best parameters for RBF-SVM (σ, C) and the best parameter for `classregtree` (`minparent`) using the cross-validation set created.
5) Report the best parameters and the validation error for those parameters. Train a model with those parameters on the complete training set and report the out-of-sample error using the test set.

Question block 6:

1) Load the dataset `'example_dataset_1.mat'`.
2) Use the cross-validation function to create a set of 5-fold indexes on the complete training set.
3) Use a nested strategy for finding and reporting the best parameters for RBF-SVM (σ, C) and the best parameter for `classregtree` and the out-of-sample errors for each of the folds of the external cross-validation.
4) Is the best model selected in each of the folds the same? Why?
5) Report the average out-of-sample error and compare it with the one obtained in *block 5*. Are they similar?

VIII. Statistical significance.

The goal of reporting out-of-sample error is to show that the performance of one (usually the proposed) method is superior to another. But, how can we assure that two different error rates do not come from distributions with the same mean error value, i.e. how can we be sure that the difference we observe is statistically significant or it is produced by randomness in the sampling process? Hypothesis testing allows to answer this question. The null-hypothesis being tested is that all classifiers perform the same. If the null-hypothesis is rejected then we may say that the

differences we observe are statistically significant with $p < \delta$ (i.e. not due to randomness) or with $1 - \delta$ confidence. A good method for testing two methods is Wilcoxon's signed rank test.

The next exercise will guide you throughout the process of experiment design and reporting. We want to compare SVM with RBF kernel with a decision tree.

Question block 7:

1) The first step in your experiment design is to define the datasets on which you are going to compare the methods. Create a table detailing the amount of data, dimensionality and the class balance ratio for each of the datasets in the folder `datasets`.
2) Enumerate the methods you are going to compare.
3) Define and write the performance metric you will use to compare the results and the methodology you will use for obtaining each of the performance results, e.g. stratified 10-fold cross-validation, bootstrapping, etc.
4) Define and write the parameter setting methodology. Which parameters are you going to tune for each method? Which method will you use for searching for the optimal model? e.g. nested cross-validation, how many folds, etc. If you use a grid search detail the grid values.
5) Define and write the hypothesis test you will use for assessing the statistical significance of your results.

And now, go for the experiments.

Question block 8:

1) Run the experiments you designed on all datasets in folder `datasets`. When running your K-fold splitting code save the resulting folds. Use the same folds for both methods. WARNINGS: This may take a while so be patient. Storing the out-of-sample error after each fold for each data set is recommended. We will use the same data sets and folds to improve the current table so store them and have them at hand for next week.
2) Create a table with all the average performances.
3) Run Wilcoxon's signed rank test and report the result. Check statistical significance for $p < 0.05$ and $p < 0.10$. Are the methods statistically different? Write a short paragraph describing the results and the statistical results obtained.