Technische Universität München
Fakultät für Informatik
Prof. Nassir Navab

WS 2016
Exercise Sheet 4
May 2nd, 2016

Exercises in Machine Learning for Medical Imaging

Exercise 1 SVMs on Toy Dataset

In this exercise, you will apply SVM classification to the 2-D points in the [`twomoons dataset`]. Use panda [`import pandas as pd`] to read the files. The dataset is already split in training and testing data, where x stands for the point coordinates, and y for the labels.

a) Read and visualise the dataset (In Python, `from matplotlib import pyplot as plt`),

b) Create a function `svm_fit`, which takes a matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ of samples, a vector $\mathbf{y} \in \mathbb{N}^n$ of outcomes for each sample, and a *kernel*. `svm_fit` should return the estimated predicted value and a structure with the performance (including the accuracy, recall, FPR and precision) on the **training** samples. Use `from sklearn import svm`.

c) Create a function `svm_predict` receiving the model trained above, a matrix $\mathbf{X}_t$ of samples for testing and their expected outcomes $\mathbf{y}_t$. The function should return a vector of predicted outcomes and and a structure with the performance (including the accuracy, recall, FPR and precision) on the **testing** samples.

d) Use the functions above to train and test the following SVM settings:

- try both the `linear` and then `RBF` kernels.
- for the `RBF` kernel set C to values $C \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000\}$
- For each kernel and C parameter display a scatter plot of the training set indicating the support vectors (*c.f.* Fig. 2-left)
- For each kernel, draw the curves for the training and testing accuracies while varying the hyper-parameter C. Use a logarithmic horizontal scale for C.

e) For the RBF kernel create a series of 2D scatter plots (one per each value of C above) comparing the predicted vs. true values for every combination of parameters (Fig. 2-right)

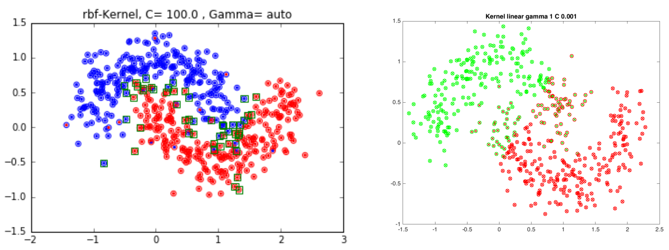

Figure 1: SVMs on the two moons dataset. **(left)** Results on training set, green squares indicate the support vectors. **(right)** Results on the test set. In both figures the colour of the external circles stands for the target class, inner color shows the predicted class. One such image should be created for each kernel and parameter value.

<u>Exercise 2</u> Digit Recognition

This exercise focuses on the application of the well known Support Vector Machine (SVMs) to the recognition of handwritten digits using the popular MNIST dataset. The problem consists in assigning to each image a label among the 10 digits. With this goal in mind:

a) **Read and visualise the dataset:** Download the `cvs file`, containing the images and labels. Note that each row encodes the information for a single image. Each image is of size 28×28 and has been flattened to a row vector. In the training file, use and remove the first column of the training set file which contains the label assigned to each image. Visualise some of the images and their labels to verify that you are correctly reading the data, e.g. using openCV `import cv2`. Note that for the test set there is no access to the labels.

b) **Dimensionality Reduction:** apply PCA to both the training and testing images to reduce their dimensionality while preserving a given percentage (e.g. 80 %) of the variance of the dataset. In Python you may use the sklearn functionality: `from sklearn.decomposition import PCA`.

c) **Binary classification:** redefine the problem as a binary classification task. Write a function that receives one digit as input and returns the corresponding training labels (1 if the sample corresponds to the desired digit, 0 otherwise).

d) **Create train and validation sets:** From the results of the previous step further split the train dataset in two subsets:

   - Use the first subset to fit an SVM model with probabilistic output.
   - Use the second subset to evaluate the prediction performance.

e) **Hyper-parameter analysis:** Use the train and validation sets to compute the performance of different SVM models. Consider both a linear and a RBF kernel and change the values of the hyper-parameter C as in the first exercise. For each case Compute the following performance measures: accuracy, recall, FPR and precision.

f) Draw the **ROC and Recall-Precision** curves for the different kernels and C values.

g) **Testing:** Choose the best model from the curves above, and try it on the test set. Visualise the predictions for random images from the test set.

h) **Bonus:** You can try to upload your results to the open challenge website (multi-class) (`https://www.kaggle.com/c/digit-recognizer`)

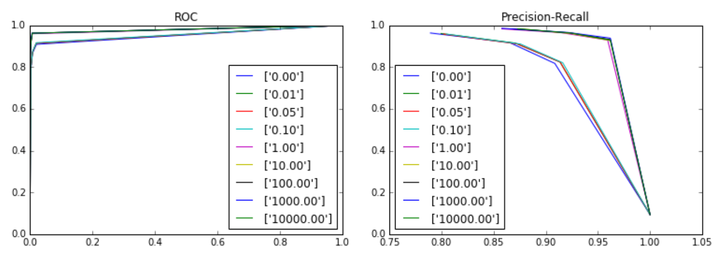

Figure 2: ROC and PR curves on the digit recognition dataset for the binary classification of digit 5