University of Tehran

School of Electrical and Computer Engineering

Pattern Recognition

Assignment 5

Due Date: 1st Dey

Corresponding TAs:

Mohammad Farazi(mohammad.farazi1991@gmail.com)
Rasool Taban

TAs Office Hours:

Wednesday, 22nd Azar, 12 – 3
(Room 527, Biomedical Eng. Lab)

Saturday, 25th Azar, 12 – 3
(Room 527, Biomedical Eng. Lab)

Azar 96

IMPORTANT NOTES

1. In problems 3 and 4, you have to use the small version of uploaded dataset, named "**TinyMNIST**". The dataset is available on your course page in "Dataset" section. Load the dataset with the attached file, named "TinyMNIST_loader(withoutPrep).py".
2. You are allowed to use pre-defined algorithms in this assignment. Using **scikit-learn** package is recommended.

PROBLEM 1

SVM - Error bound

The inequality constraints for the non-separable case in SVM are given by:

$$x_i \cdot w + b \geq +1 - \xi_i \quad \text{for} \quad y_i = +1$$
$$x_i \cdot w + b \leq -1 + \xi_i \quad \text{for} \quad y_i = -1$$
$$\xi_i \geq 0 \quad \forall i$$

Where x_i, $i = 1, \ldots, M$ are the M data points with associated labels y_i is either +1 or -1 and associated margins ξ_i. Show that $\sum_{i=1}^{M} \xi_i$ is an upper bound on the error of the classifier

(The error is 1 if a point is misclassified, 0 otherwise).

PROBLEM 2

Kernels

(a) Create an SVM that calculates the XOR function. Use +1 and -1 (not zero and one) for both inputs and outputs. Map the input [x1, x2] into a space consisting of x1 and x1x2. Draw all possible points, and the maximal margin separator. Specify the margin. Now draw the separating line back in the original Euclidian input space.

(b) Consider the equation of a circle with its center on (a, b). Show that every circular area in the feature space (x_1, x_2, x_1^2, x_2^2) is linearly separable.

PROBLEM 3

As you know, in a multi-class problem we can do the classification by repeatedly solving a two-class problem and using approaches like one-against-one and one-against-all. Design a Soft Margin SVM classifier for the given dataset, concerning different approaches such as one class against one class, and one class against all classes. Use **polynomial kernel** with **appropriate order** to obtain your result. Your order of polynomial is important since it might affect the generalizability of your model. You should use a goodness of fit analysis with different values of orders (1,2,3,4…) to obtain a good result. Generally, for choosing the optimum parameter of your classifiers in any case, you can use **GridSearchCV** (please peruse the Scikit-learn instruction and find out about its attributes and parameters).

1- Compare the results of the two approaches with each other concerning the CCR (Correct Classifier Rate) on the test data, and also the time required for training and testing the dataset.
2- Plot the goodness of fit curve for finding the optimum polynomial order.
3- Explain completely advantages and disadvantages of each case.

4- Now use **linear SVM** with appropriate value of parameter "C". Compare the performance of the linear SVM with your results in section (1) using the polynomial kernel.

PROBLEM 4

Change the kernel type of the one against one approach, and compare the results of these kernels. In this part it is obligatory to use the **polynomial kernel** and **RBF kernel.**

Each of these kernels has some parameters that must be set. For each kernel, find the best parameters which would result in a high CCR. As an instance, you can use **GridSearchCV** for gaining the appropriate parameters.

1- In each case, draw diagrams showing the CCR versus parameter(s).
2- Now, apply PCA to your data before classification. For choosing the number of principal components: The total number of principal components should preserve 95% of the variance of the information. Afterwards, using the parameters of your kernel in previous section, classify your data and report the final performance by using the test dataset.

Some useful links are provided below regarding the SVM tutorial both its theory and implementation:

Scikit-learn tutorial for SVM: http://scikit-learn.org/stable/modules/svm.html

SVM A-Z in 45 Minutes | MIT Open Courseware with Patrick Winston

SVM lectures by Andrew Ng. : part1, part2, part3, part4, part5, part6