CSE 572: Data Mining

Fall 2016

Assignment 4 / Mini Project 2

Instructions

- **Submission Deadline**: Friday, November 4, 2016 (11:59pm). You will submit this assignment through Blackboard
- There are two problems. Total points possible: 20
- You have to use Matlab for this assignment
- You will work in groups of 2 or 3 for this assignment
- Only one submission is required from each group. Mention the names of all group members in the report (please see details below)

In this assignment, you will study the application of the Support Vector Machine (SVM) classifier on two real-world classification problems. The datasets to be used for this assignment are uploaded under the "Datasets" folder. X_train, y_train, X_test and y_test denote the training features, training labels, testing features and testing labels respectively. In X_train and X_test, each row denotes a data sample and each column denotes a feature.

You are allowed to use Matlab's in-built functions for training and testing an SVM for this project. Take a close look at the functions *svmclass* and *svmval* in Matlab. Use c = 10000 and lambda = 0.0001 for training.

Problem 1 (10 points)

The VidTIMIT dataset consists of video and audio recordings of 43 subjects reciting short sentences. In this assignment, we will use a subset of the dataset with 25 subjects. We will also use only the video modality. The videos were sliced into images and the discrete cosine transform function was used to extract feature vectors of dimension 100 from each image. The training set contains 3,500 samples and the test set contains 1,000 samples. Our objective is to recognize a subject from a given image.

Train an SVM classifier with a polynomial kernel with parameter 2 on the training set and test on the test set. You need to train one SVM for each class; for predicting a test sample, use the maximum of the values returned by all the SVMs to decide the final class. Report the percentage accuracy on the test set.

Problem 2 (10 points)

The previous problem is an example of a **multi-class classification** problem, where there are multiple classes in the dataset, but each data sample can belong to only one class. **Multi-label classification** is a generalization of multi-class classification, where each data sample can belong to multiple classes simlutaneously. For instance, consider the problem of classifying an outdoor image of a scene. Suppose the possible classes are beach, mountain, field and sunset. It is possible for a particular image to contain both beach and mountain or beach, mountain and sunset all together. The objective of multi-label learning is to predict all the classes present in a data sample.

The Scene dataset consist of 2407 images of an outdoor scene, where each image is represented by a feature vector of dimesnion 294. Also, there are 6 classes in the problem and an image can belong to one or more of the 6 classes. The dataset has been divided into a training set (with 1500 samples) and a test set (with 907 samples). Each row of X_train and X_test denotes a sample and each column denotes a feature. Each row of y_train (y_test) denotes the labels of the corresponding training (testing) sample, where 1 means the class is present and 0 means the class is absent. For instance, in the training set, sample 462 belongs to classes 4 and 5.

One strategy to solve a multi-label learning problem is to train an SVM separately for each class. To predict a test sample, each SVM is applied separately on it. A positive output indicates that the corresponding class is present and a negative output indicates that it is absent. The accuracy is computed using the following expression:

$$A = \frac{|T \cap P|}{|T \cup P|}$$

where T is the true class label vector of a test sample and P is the predicted class label vector. Train an SVM classification model on the training set and test on the test set. Report the percentage accuracy on the test set using the following classification models: (i) SVM with polynomial kernel with parameter 2 and (ii) SVM with Gaussian kernel with parameter 2.

You need to submit (only one submission is required from each group)

- The Matlab code files
- A ReadMe file with clear instructions on how to run your code
- A brief report (1-2 pages) summarizing your findings and the accuracies obtained. **Mention the names of all the group members in the report.**

Submit all the documents as a single zip file through Blackboard.