CMPE 462 Assignment 2

Boğaziçi University Department of Computer Engineering
Deadline: May 14th, 2024 by midnight

Spring 2024

In this assignment, you will implement the following models. You can use libraries such as Numpy, scipy, and Matplotlib in your experiments. If the task requires implementation from scratch, you are not allowed to use a library. If training and test splits are not provided in the datasets, please randomly split your data into training and test. Please submit a PDF report containing the link to our code, your answers, and references. Please cite all the resources used in the assignment. If you ever use an AI tool such as ChatGPT, please acknowledge. Each group member should be able to answer questions regarding any of the sections below. Please submit one report per group.

1 Decision Trees (30 pts)

In this task, please use the dataset you used in the Naive Bayes task of the first assignment, `https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic`.

1. Train a decision tree using the scikit-learn's function. Please tune the depth of the tree and visualize the learned tree using the library. You can use the default splitting criteria, which is gini.

2. Compare its test performance with the Naive Bayes classifier you trained in the first assignment.

3. Using the decision tree, obtain the most significant features. Select the most significant 5, 10, 15, and 20 features, train a linear classifier of your choice for each, and compare the performance. Comment on the effect of this feature selection approach on the performance.

4. Train a random forest using all the original features and compare its test performance with the decision tree in 1. Please plot the change in test and training performances with the varying number of trees in the forest.

2 Support Vector Machines (40 pts)

You will implement the SVM classifier for the MNIST [1] dataset in this task. MNIST has 50,000 training and 10,000 test images of 10 classes. Please consider the digits 2, 3, 8, and 9 in this section. Thus, the total number of samples will be 20,000 (5000 for each class) in the training set and 4,000 in the test set.

1. Please flatten the gray-scale images and feed these vectors directly to your soft-margin SVM model.

   (a) Please train a 4-class linear SVM using one-vs-all. Please train the primal formulation of SVM from scratch using a quadratic programming solver. Please clearly write the expressions you feed to the solver. Please tune the hyperparameters and report your training and test accuracy.

   (b) Please train a 4-class SVM using the scikit-learn's soft margin primal SVM function with linear kernel. Please tune the hyperparameters and report your training and test accuracy. Compare the results with part (a) regarding classification accuracy and training time.

   (c) Please train a 4-class non-linear SVM using one-vs-all. Please train the dual formulation of SVM from scratch using a quadratic programming solver. Please clearly write the expressions you feed to the solver. You may choose any kernel you like. Please tune the hyperparameters and report your training and test accuracy.

   (d) Please train a 4-class SVM using the scikit-learn's soft margin dual SVM function with a non-linear kernel. You may choose any kernel you like. Please tune the hyperparameters and report your training and test accuracy. Compare the results with part (c) regarding classification accuracy and training time.

2. Please extract features from the images. You may try any feature extraction technique you like. However, please explain the reason behind your choice. Repeat the experiments in 1. a-d with the extracted features and compare the performance in terms of accuracy and training time.

[1] `https://en.wikipedia.org/wiki/MNIST_database`

3. Please find the support vectors using one of the dual SVM models you trained and inspect the images. Please discuss whether there is any visual difference between the support vectors and other images.

3 Clustering (30 pts)

In this task, please use the 4-class MNIST data you use in the second task.

1. Is normalizing the data points before running k-means important? Please explain.

2. Please implement the k-means algorithm from scratch using Euclidean distance. Find 4 clusters using the flattened images. Repeat this experiment for the features you extracted. Please compare the clustering outputs using the external (clustering accuracy)and internal (SSE) metrics.

3. Please repeat step 2 with cosine similarity instead of Euclidean distance. Did you observe a significant difference in the clustering results?

If the variances of the original xi dimensions vary considerably, they affect the direction of the principal components more than the correlations, so a common procedure is to preprocess the data so that each dimension has mean 0 and unit variance, before using PCA. Or, one may use the eigenvectors of the correlation matrix, R, instead of the covariance matrix, S, for the correlations to be effective and not the individual variances.

Ethem Alpaydin p.148/640

We know from equation 5.15 that if x Nd(,), then after projection WT x Nk(WT , WT W). If the sample contains d-variate normals, then it projects to k-variate normals allowing us to do parametric discrimination in this lower-dimensional space. Because zj are uncorrelated, the new covariance matrices will be diagonal, and if they are normalized to have unit variance, Euclidean distance can be used in this new space, leading to a simple classifier.

Ethem Alpaydin p.150/640