Assignment 4

Anuraag Kansara 19111013
Ashwani Bhat 19111019
Chayan Dhaddha 19111025

October 6, 2019

Decision Tree

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous variables. In this technique, we split the sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables. The split should ensure maximum purity. Decision tree works on the principle of divide and conquer. Internal nodes perform the splits based upon certain classifier and the leaf nodes predict the outcome based on the applied classifier at leaf node.

Principal Component Analysis

Working directly with high-dimensional data, comes with some difficulties: It is hard to analyze, interpretation is difficult, visualization is nearly impossible, and (from a practical point of view) storage of the data vectors can be expensive. However, high-dimensional data often has properties that we can exploit. For example, many dimensions are redundant and can be explained by a combination of other dimensions. Furthermore, dimensions in high-dimensional data are often correlated so that the data possesses an intrinsic lower-dimensional structure. Principal component analysis (PCA) exploits structure and correlation and allows us to work with a more compact representation of the data, ideally without losing information. We can think of PCA as a compression technique, similar to jpeg or mp3, which are compression algorithms for images and music. In PCA, we take the largest singular values of the matrix (which are also known as leading singular values) and take the corresponding singular vectors.

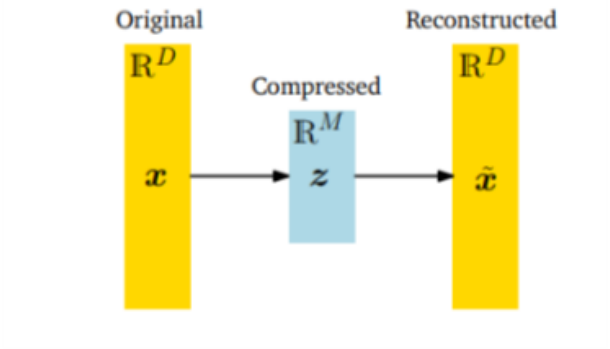

Figure 1: Graphical illustration of PCA.
In PCA, we find a compressed version z of original data x. The compressed data can be reconstructed into x˜, which lives in the original data space, but has an intrinsic lower-dimensional representation than x.

Support Vector Machine

Support Vector Machines are supervised learning models with associated learning algorithms that analyse data and recognize patterns used for classification and regression analysis.
SVM is a discriminative classifier defined by a separating hyperplane. Given labelled training data (supervised learning) the algorithm outputs an optimal hyperplane which categorizes new examples. In 2D space this hyperplane is a line dividing a plane in 2 parts where each class lie in either side.

In real world applications finding perfect class for millions of training data set takes lot of time. In order to overcome this we have regularization parameters (or, tuning parameters in SVM). Another parameter is the Kernel parameter through which we can choose whether the separator is linear or non-linear.

Hyperparameter Tuning

Decision Tree

- $criterion$: we have used 'gini' index as the function to measure the quality of the split.
- $max-depth$: We have used the maximum depth to be 12, so as to ensure that our tree is perfectly balanced. If not, we'll prune the rest of the nodes.
- $min-samples-leaf$: The minimum number of samples required to be at a leaf node are set to 2. We didn't find much difference when changing the default value (which is 1) to 2, but it will ensure that the tree doesn't grow one more level.

SVM (with PCA)

- $PCA(n-components)$: Tells us how many components to keep. We calculated the variance score of the leading components ([0.64, 0.28, 0.06...]). Only 2 leading components had some good variance score, so we choose only 2 components to keep. Thus we reduced the dimension of our dataset from 132 to 2.
- C : Penalty parameter in objective function. We found C=15 to give the best accuracy.
- $kernel$: 'rbf' kernel gave the best accuracy.

Best model selection

We have used K-fold validation and found that the mean accuracy of our K-fold cross validation was best for decision tree classifier. Validation set is 25% of the total data (i.e., 1/3rd of the training data, thus K=3). The mean validation scores of both the models is written below

Model	Mean validation accuracy
Decision Tree	99.29
SVM (with PCA)	98.32

Classification report

Classification Report		
	Decision Tree	SVM (with PCA)
Accuracy	99.42%	98.64%
True Negative	408	403
True Positive	624	621
False Negative	2	4
False Positive	4	10
Precision	0.9936	0.9841
Recall	0.9960	0.9936

- True Negative: Predicted Normal and Actual Normal
- True Positive: Predicted Stressed and Actual Stressed
- False Negative: Predicted Normal and Actual Stressed
- False Positive: Predicted Stressed and Actual Normal
- $Precision = \frac{True\ Positive}{(True\ Positive + False\ Positive)}$
- $Recall = \frac{True\ Positive}{(True\ Positive + False\ Negative)}$