CSE 6363 - *Machine Learning*

Homework 2- Spring 2019

Due Date: Mar. 16, 2019

Support Vector Machines

1. Consider the following linearly separable training data set:

$$\begin{aligned} D = \{ \quad & ((1,2), \quad -1), \\ & ((2,3), \quad 1), \\ & ((2,1), \quad -1), \\ & ((3,4), \quad 1), \\ & ((1,3), \quad -1), \\ & ((4,4), \quad 1) \qquad \} \end{aligned}$$

   a) Formulate the optimization function as well as the constraints for the corresponding linear maximum margin optimization problem without a regularization term. Also show the corresponding Lagrangian as well as the Lagrangian Dual for this problem.

   b) Manually perform 4 iterations of the SMO algorithm on this data. You do not have to use any specific heuristic to pick the two α parameters in each iteration.

   c) Use a SVM solver (e.g. MatLab's *fitcsvm* function) to learn the linear SVM parameters for this problem. Show the resulting decision boundary and identify the support vectors in this problem.

Decision Trees

2. Consider the problem where we want to predict whether a mushroom is edible or poisonous from a set of discrete attributes, namely cap-shape (6 possible values), cap-surface (4 possible values), cap-color (10 possible values), bruises (2 possible values), and odor (9 possible values). Data is given in the files as a comma separated list $\{e, x, s, y, t, a\}$ where the first entry is the class (e or p), the second is the cap-shape (b, c, x, f, k, or s), the third is the cap-surface (f, g, y, or s), the fourth entry is the cap-color (n, b, c, g, r, p, u, e, w, y, t, or f), the fifth entry is whether it bruises (t, or f), and the last entry is the odor (a, l, c, y, f, m, n, p, or s). There is a training and a test data set for this problem (datasets are derived from the more expansive UCI machine learning mushroom data set).

   a) Show the construction of a 2 level decision tree using minimum Entropy as the construction criterion on the training data set. You should include the entropy calculations and the construction decisions for each node you include in the 2-level tree.

b) Implement a decision tree learner for this particular problem and derive the complete tree for the training data set.

c) Apply the tree from part $b)$ to the test data set and compare the classification accuracy on this test set with the one on the training set. Does the result indicate overfitting ?