CSE 404: Introduction to Machine Learning (Fall 2019)
Homework #9
Due 12/2/2019 11:59 NO EXTENSION WILL BE GRANTED

Note: (1) LFD refers to the textbook "Learning from Data". (2) Only hard-copy submission is required for this homework.

1. (25 points) Exercise 8.13 (e-Chap:8-31) in LFD.

   KKT complementary slackness gives that if $\alpha_n^* > 0$, then $(\mathbf{x}_n, y_n)$ is on the boundary of the optimal fat-hyperplane and $y_n(w_*^t \mathbf{x}_n + b_*) = 1$. Show that the reverse is not true. Namely, it is possible that $\alpha_n^* = 0$ and yet $(\mathbf{x}_n, y_n)$ is on the boundary satisfying $y_n(w_*^t \mathbf{x}_n + b_*) = 1$.

   [Hint: Consider a toy data set with two positive examples at (0,0) and (1, 0), and one negative example at (0, 1).]

   [Note: You don't have to read through the KKT complementary slackness to work out this question.]

2. (25 points) Exercise 8.15 (e-Chap:8-38) in LFD.

   Consider two finite-dimensional feature transforms Φ_1 and Φ_2 and their corresponding kernels K_1 and K_2.

   (a) Define $\Phi(x) = (\Phi_1(\mathbf{x}), \Phi_2(\mathbf{x}))$. Express the corresponding kernel of Φ in terms of K_1 and K_2.
   (b) Consider the matrix $\Phi_1(\mathbf{x})\Phi_2(\mathbf{x})^T$ and let $\Phi(\mathbf{x})$ be the vector representation of the matrix (say, by concatenating all the rows). Express the corresponding kernel of Φ in terms of K_1 and K_2.
   (c) Hence, show that if K_1 and K_2 are kernels, then so are $K_1 + K_2$ and $K_1 K_2$.

   The results above can be used to construct the general polynomial kernels and (when extended to the infinite-dimensional transforms) to construct the general Gaussian-RBF kernels.

3. (25 points) Experiment: A data set (data.mat) is given and you are asked to apply support vector machines on this data set. You can use the software package LIBSVM[1] for this purpose. Note that LIBSVM has a Python interface[2], so you can call the SVM functions in Python. You can use the first 150 samples for training and the rest for testing. In this task, you should try different values for the parameter c, different kernel functions, etc. and write a short report summarizing your observations. Also, please report how the number of support vectors changes as the value for c increases (while all other parameters remain the same).

   [Note: You can load the mat dataset into Python using the function `loadmat` in `Scipy.io`.]

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[2] https://github.com/cjlin1/libsvm/tree/master/python