Statistical Learning and Fuzzy Logic Algorithms - CMSC 678
Project No. 2
Due Monday, Oct 15th, 2018, 2pm

Part 1) Hard margin linear SVM **5% of grade**

Use dataset P2_data.mat and design the Linear Support Vector Machine i.e., the hard margin classifier.

a) What are the alpha values of support vectors? What is the bias? What is the size of margin *M*? Calculate margin as *M* = 1/norm(**w**).
d) What are the values of the decision function for the test datapoints [3 4] and [6 6]?

Plot the data, separation boundary (solid blue) and both margins (dashed blue) in the input space. Clearly show which data are support vectors.

Part 2) Multiclass soft classification **20% of grade**

For a given dataset *glass* design the *1 vs All* classifier by using both polynomial kernel (you will write the code for getting kernel matrix for polynomial kernel) and Gaussian one (here, use my present to you, the code *grbf_fast.m*). What is the accuracy of each classifier?

Some hints:

- In part 1 neither shuffle nor scale the data. Use them as given to you. In part 2 do both shuffling and scaling.
- Define Hessian matrix and all the other matrices and vectors needed for matlab's routine *quadprog*. Note, Hessian matrix **H** may be badly conditioned. The remedy is as follows: add to the **H**'s diagonal elements small number by the line `H = H + eye(l)*1e-7;, where l is the number of training datapoints.`
- In identifying ALL support vectors find alphas bigger than some accuracy value, say ε = 1e-5. In finding the FREE support vectors use the line
  ind_Free = find(alpha >= ε & alpha <= C - ε);
- In calculating bias you have to differ between the free and bounded support vectors.
- In part 2 data is in sparse format. Read it in and change the format as follows:
  [Y X]=libsvmread('glass'); X=full(X);
- In part 2 **for each classifier** in *1 vs All* design do the 5-fold cross-validation (CV) following CV handouts. Use the following values for the cross-validation
  C0 = [1e-2 1e-1 1 1e1 1e2 1e3 1e4]
  parameters = [1 2 3 4 5] for polynomial kernel classifier i.e.
  parameters = [1e-2 1e-1 1 1e1 1e2 1e3] for Gaussian (i.e. RBF) kernel classifier
  After finding the best hyperparameters (C_{best} and $degree\ of\ polynomial_{best}$ i.e., C_{best} and σ_{best}) design each classifier by using ALL data.
- After designing all classifiers you will have 6 class predictions vectors Y_{pred}. Your final, single, classifier will be obtained by using MAX operator to decide about class. This is known as the Winner-takes-All approach.

- Write a single code for designing both classifiers. I mean don't write two codes, the first one for the polynomial kernel classifier and the other for the Gaussian one. Use the variable *kernel*, and say if *kernel* = 1 use polynomial kernel and if *kernel* = 2 use the Gaussian one. The difference between different kernels is only in calculation of kernel matrix. All the other lines are same. Sure, in designing CV loops work with these lines

```
for i = 1:length(C0)
    C = C0(i);
    for j = 1:length(parameters)
            param = parameters(j)
            ...
    end
end
```

  Where *param* for polynomial is its degree *d* and for the Gaussian is the σ of the kernel.

****

Submit both your written report (in an IEEE format) and program to me by Email.

ZIP your report and programs into a single zip file. Name it with your family name (say, lee.zip) and send it to me. A Subject field in your Email should be CMSC 678, Family name, Project 2. Don't hesitate to contact me in the case of need.

My Email is: vkecman@vcu.edu.

START EARLY, MEANING NOW PLEASE.