Boston University
Department of Electrical and Computer Engineering

ENG EC 414 Introduction to Machine Learning

HW 5

Issued: Fri 9 Oct 2020 **Due:** 10:00am Fri 16 Oct 2020 in Gradescope (non-code) + Blackboard

Important: Before you proceed, please read the documents pertaining to *Homework formatting and submission guidelines* in the Homeworks section of Blackboard. **In particular, for computer assignments you are prohibited from using any online code or built-in MATLAB functions except as indicated in the problem or skeleton code (when provided).**

Note: Problem difficulty = number of coffee cups ☕

Problem 5.1 [13pts] *(SVM by hand)*
Consider again the two-class classification with the following training feature vectors: $\mathbf{x}_P = (2, 0)^\top, \mathbf{x}_Q = (0, 4)^\top, \mathbf{x}_R = (3, 3)^\top, \mathbf{x}_S = (7, 5)^\top$ with labels $-1, -1, +1, +1$ respectively.

(a) [6pts] Hand-compute the parameters of the maximum margin linearly separating hyperplane (SVM hyperplane) in *canonical* form.

(b) [3pts] Determine which training points lie on the margin.

(c) [4pts] We said that the SVM hyperplane can always be written as a linear combination of the support vectors: $\boldsymbol{w}_{\text{SVM}} = \sum_{i=1}^{N_{\text{SV}}} \alpha_i z_i$, where z_i are the N_{SV} support vectors. Find the coefficient α_i of this combination for the hyperplane in the point (a).

Problem 5.2 [20pts] *(k-folds cross-validation for Regularized Least Square Regression with polynomials)*
In this question, you have to implement linear regression from $\mathbb{R}^d$ to $\mathbb{R}$ using polynomials and choose the best degree of the polynomials using k-folds cross-validation. In other words, **for each coordinate i of the input** x_i, we generate new features $(x_i^2, x_i^3, \dots, x_i^p)$ and we append them to the original features. Then, we learn a linear predictor in this space. This corresponds to learn a predictor of the form

$$\hat{y} = w_1 x_1 + \cdots + w_d x_d + w_{d+1} x_1^2 + \dots w_{2d} x_d^2 + \cdots + w_{d(k-1)+1} x_1^p + \cdots + w_{dp} x_d^p + b\ .$$

We also assume that all the coordinates of the input are positive.

(a) [5pts] First, complete the skeleton code of `generate_poly_features.m` to implement a function that generates a matrix of input samples that contains the polynomials of each feature from the linear term to the polynomial of degree `p`, with prototype

```
[X_poly] = generate_poly_features(X,p)
```

Using our notation, `X` is $m \times d$, where m is the number of training samples and d is the dimension. `X_poly` will have the same number of rows and columns equal to $d \times p$. Do not include the term of order 0, that is, the column of 1s.

(b) [6pts] Complete the skeleton code of the function `cross_validation_rls.m` to implement k-folds cross-validation for regularized least square. The prototype is

```
[validation_loss] = cross_validation_rls(X,y,lambda,k)
```

As we have seen in class, in `k`-folds cross-validation, we divide the training data contained in `X` and `y` in k disjoint sets. We assume the training data to be shuffled, so it does not matter how you create the folds. Then, we use one of the k folds as the validation fold and we use the remaining $k-1$ to train our RLS predictor with λ =`lambda`. Evaluate the loss of the trained predictor on the validation fold, repeat the above k times, and return the averages of the mean losses on the validations folds in `validation_loss`. Note that `validation_loss` is a scalar. The function to run RLS is also provided in `train_rls.m`.

(c) [4pts] In the zip file there is also the "cadata" training/test data in the file "cadata_train_test.mat". It is a random train/test split of the Housing dataset from the UCI repository. The task is to predict the median house value from features describing a town. I normalized the features for you, to be in $[0, 1]$. Complete the skeleton code in `problem_5_2c.m` to use `cross_validation_rls` and `generate_poly_features` to try polynomials up to degree 10 with 8-folds cross-validation and $\lambda = 0.001$. The code should record the cross-validation loss for each choice of the degree of the polynomial from 1 to 10 in a vector of dimension 10.

(d) [2pts] Plot the 8-folds validation loss for each degree of the polynomial using the code in the previous point and report the degree of the polynomial that gives the best 8-folds cross-validation loss. Discuss the results: is this what you expected? Does it make sense? (No code to submit here.)

(e) [3pts] Re-train a RLS predictor with the best degree found in the previous point and report its mean square loss error on the test set. (No code to submit here.)

Problem 5.3 [9pts] *(Questions)*
Clearly explain your answers.

(a) [2pts] Consider least square regression with polynomials. Does the bias decrease with the degree of the polynomials?

(b) [2pts] Can I alleviate underfitting increasing the number of training samples?

(c) [2pts] Overfitting is due to too much bias or too much variance?

(d) [3pts] ☕ We want to use a hard-margin SVM with polynomials features. For a fixed training set, does the margin of the trained SVM increase or decrease with the degree of the polynomial?

Code-submission via Blackboard: You must prepare 3 files: `generate_poly_features.m` for Problem 5.2(a), `cross_validation_rls.m` for Problem 5.2(b), and `problem_5_2c.m` for Problem 5.2(c). Place them in a **single** directory which should be zipped and uploaded into Blackboard. Your directory must be named as follows: <yourBUemailID>_hwX where X is the homework number. For example, if your BU email address is `charles500@bu.edu` then for homework number 5 you would submit a single directory named: `charles500_hw5.zip` which contains all the MATLAB code (and only the code).

Three corresponding skeleton code files are provided for your reference. Reach-out to the TAs via Piazza and their office/discussion hours for questions related to coding.