CSC 242: Homework 2

Assigned on Feb. 10
Hand in before 2pm Feb. 24 (Tuesday)

Requirement

Please hand in the hard copy of your homework in our class (that is, before 2pm) on Feb. 24 (Tuesday). If you need to use your delay coupon, you can hand in your homework to my office. Include the **class ID number** assigned to you for this class, **your name**, and **how many days** late it is (if handed in late) in the headline. The ID number is used to simply the grading procedure. If you do not know your ID number assigned to you, you can check the link.[1] If you are not on the list, please contact our TA Sean Esterkin (sesterki@u.rochester.edu). (*), (**), or (***) indicates the difficulty of each question.

Policy

We apply the late policy explained in syllabus to all homework. Any grading questions must be raised with the TA in two weeks after the homework is returned.

The homework must be completed individually. However, you are encouraged to discuss the general algorithms and ideas with classmates in order to help you answer the questions. You are also allowed to share examples that are not on the homework in order to demonstrate how to solve problems. If you work with one or more other people on the general discussion of the assignment questions, please record their names over every question they participated.

However, the following behaviors will receive heavy penalties (lose all points and apply the honest policy explained in syllabus)

- explicitly tell somebody else the answers;
- explicitly copy answers or code fragments from anyone or anywhere;
- allow your answers to be copied;
- get code from Web.

[1] `https://docs.google.com/spreadsheets/d/1hgxcxqFxBxZYb55_HHi2pWQtOrEyEppcStSZBKni8MU/edit?usp=sharing`

1 (**) Probability Game (Monty Hall problem): 3 points

The **Monty Hall problem** is a brain teaser, in the form of a probability puzzle, loosely based on the American television game show. Let us Make a Deal and named after its original host, Monty Hall. The problem was originally posed in a letter by Steve Selvin to the American Statistician in 1975 (Selvin 1975a), (Selvin 1975b). It became famous as a question from a reader's letter quoted in Marilyn Savant's "Ask Marilyn" column in Parade magazine in 1990 (Savant 1990a). More background can be found in `http://en.wikipedia.org/wiki/Monty_Hall_problem`. You will have more fun if you do not check this link first.

The following question is a variant to the classic version.

Suppose you are on a game show, and you are given the choice of 100 doors: Behind one door is a car; behind the others, goats. (Our assumption is that car is much more valuable than the goat. The car is what you desire.) You pick a door, say No. 1, and the host, *who knows what is behind the doors*, opens a door which has a goat, say No. 100. Now the host offers you three options

(A) Insist your option door No. 1 and open it;

(B) Randomly select a door from No. 1 to No. 99 and open it;

(C) Randomly select a door from No. 2 to No. 99 and open it.

If the car is behind the door you decide to open, you get the car. The question is which option you will choose? Please provide the probabilities that you can get the car by three options (A), (B), and (C) respectively.

2 (*) Ask Google for help: 1 points

List as least four applications of the classification algorithm.

3 (***) K-NN: 4 points

Given four training samples $A = (1, 1), B = (-1, -1), C = (1, -1), D = (-1, 1)$. A and B are in the positive class, while C and D are in the negative class. Assume to use the Euclidean distance to define the distance between any two points.

- Let $K = 1$. Draw the decision boundaries based these four training samples and indicate the class label for each subarea.
- Let $K = 3$. Draw the decision boundaries based these four training samples and indicate the class label for each subarea.

4 (**) SVM: 2 points

Given 9 training samples. The positive class has 4 points: $(-1, 0)$, $(1, 0)$, $(0, 1)$, and $(0, -1)$. The negative class has 4 points: $(0, 0)$, $(1, 1)$, $(-1, 1)$, $(1, -1)$, and $(-1, -1)$. They are not linear separable. You are asked to design a differentiable mapping function $f(\cdot) : \mathbb{R}^2 \to \mathbb{R}$ such that this problem is linear separable after this mapping.

5 (**) K-NN: 8 points (programming homework)

This is a programming homework. You are asked to implement K-NN. In this question, we are playing with an "old" data set but an interesting problem.

The Disputed Federalist Papers: The Federalist Papers were written in 1787-1788 by Alexander Hamilton, John Jay, and James Madison to persuade the citizens of the State of New York to ratify the U.S. Constitution. As was common in those days, these 77 essays, about 900 to 3500 words in length, appeared in newspapers signed with a pseudonym, in this instance, "Publius". In 1778, these papers together with eight additional articles on the same subject – a total of 85 articles – were published in book form. Since then, the consensus has been that John Jay was the sole author of 5 of a total 85 papers, that Hamilton was the sole author of 51, that Madison was the sole author of 14, and that Madison and Hamilton collaborated on another three. The authorship of the remaining 12 papers, known as the "disputed papers," has been a matter of long-standing controversy. It has been generally agreed that the disputed papers were written by either Madison or Hamilton, but there was no consensus about which were written by Hamilton and which by Madison. In this project, we look at the frequencies with which Madison and Hamilton used certain words, and use this information to try to determine which of them wrote each of the 12 disputed papers. More background story can be found from `http://en.wikipedia.org/wiki/The_Federalist_Papers`.

In this homework, we look at the frequencies with which Madison and Hamilton used certain words, and use this information to try to determine which of them wrote each of the 12 disputed papers.

The data is available in the files "train_86_by_71.txt", "tune_20_by_71.txt", and "test_12_by_70.txt". The total number of samples is 118 (one line per paper). (A number of other papers with known authorship of either Hamilton or Madison were added to the dataset, to provide extra data on how the two authors made use of vocabulary.) The "train_86_by_71.txt" file is used to train your model (K-NN). The "tune_20_by_71.txt" is used to tune the parameter K. The first entry in each line contains the code number of the author: 1 for Hamilton and 2 for Madison. The remaining entries contain 70 floating point numbers that correspond to the relative frequencies (number of occurrences per 1000 words of the text) of the 70 function words, which are also available in the data file as an array of strings. The "train_86_by_71.txt" has 86 samples and the tune_20_by_71.txt" file has 20 samples.

The "test_12_by_70.txt" file is used to test your K-NN model. It has 12 samples, that is, 12 disputed papers. Each line in this file corresponds to a disputed paper. Off course you have no labels on them.

You are required to implement the K-NN algorithm to classify the samples in The "test_12_by_70.txt" file. Your model can only be constructed from the training data set, that is, "train_86_by_71.txt". The tuning data set, that is, "tune_20_by_71.txt", can only be used for deciding the parameters in K-NN. The best value for K is decided by the best prediction accuracy on the tuning data set. Use your optimal value for K and the model learned from the training data set to predict the authorship for 12 disputed papers in file "test_12_by_70.txt".

Please report the following things:

- your best value K;
- the prediction accuracy on the tuning data set "tune_20_by_71.txt";
- the prediction result on the testing data set "test_12_by_70.txt".

(Important Hint: To obtain a reasonable accuracy, you might consider to normalize the data set. Data normalization is a very common trick to pre-process data. For example, the first attribute is the salary, whose range is $[0, 100000]$, while the second attribute is number of kids the object has, whose range is $[0, 5]$. Apparently, the scales of these two attributes are not comparable. To make two attributes have roughly equal significance, one way is to scale all attribute into the same range. This pre-process is call "data normalization".)

You can use any programming language you feel comfortable. The code is not required to submit, but if TA had any question about your implementation or result, it is your obligation to provide runnable code which generates the result you submit. The fail to do that will cause losing points in this question. You should implement your own algorithm. Directly calling existing functions for this algorithm from any programming languages is not permitted.

6 Bonus question (***) SVM: 1 points

Recall that the SVM formulation (with slack variable) is

$$\begin{array}{lll} \min\limits_{w,b,\xi} & \frac{1}{2}w^\top w + C\sum\limits_{i=1}^{n}\xi_i & \\ s.t. & y_i(w^\top X_i + b) \geq 1 - \xi_i & \text{for all} \quad i = 1, \cdots, n \\ & \xi_i \geq 0 & \text{for all} \quad i = 1, \cdots, n. \end{array} \tag{1}$$

This is a constrained optimization problem. In most situations, people hate solving this constrained problem directly, because the constraint is too complicated. One way is to reformulate it into a dual optimization problem (we mentioned it in our class) and use the solution to the dual problem to recover the solution to the primal problem (1).

Now I want to introduce another way to get rid of the constraint in (1). An equivalent problem to (1) is to solve the following formulation

$$\min_{w,b} \quad \frac{1}{2}w^\top w + C\sum_{i=1}^{n} \underbrace{\max(\alpha(w, b, X_i, y_i), 0)}_{\text{this term is called hinge loss}}, \tag{2}$$

where $\max(p, q)$ returns the bigger value of p and q, and $\alpha(w, b, X_i, y_i)$ is an function in terms of $\{w, b, X_i, y_i\}$. Now I ask you to define the $\alpha(\cdot, \cdot, \cdot, \cdot)$ function such that (1) is equivalent to (2). "Equivalent" means that the solution to (1) and the solution to (2) are the same. (You can find the solution to this question from INTERNET, but please do not do that.)

7 (**) SVM: 8 points (programming homework)

Use the SVM classifier to predict the authorship in Problem 5.

Please report the following things:

- your best value C, where C is defined in (1);
- the prediction accuracy on the tuning data set "tune_20_by_71.txt";
- the prediction result on the testing data set "test_12_by_70.txt".

You can use any programming language you feel comfortable. The code is not required to submit, but if TA had any question about your implementation or result, it is your obligation to provide runnable code which generates the result you submit. The fail to do that will cause losing points in this question. In this question, you do not have to implement SVM, you can call the matured SVM solvers from anywhere, for example, liblinear `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`.