CMSC 818B: Decision-Making for Robotics (Fall 2019)
Homework 2
Due: October 11th, 11:59PM

October 2, 2019

You must work on this homework individually. Your submission must be your original work. Please follow the submission instructions posted on canvas exactly.

Problem 1 10+10 points

Let X denote a ground set of elements, $X = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$. Let $w_i \geq 0$ denote the weight of element of x_i. Let A be any subset of X.

- Let $f_{\max}(A) = \max_{x_i \in A} w_i$. That is, $f_{\max}(A)$ is a function that returns the maximum weight of all the elements contained in A. Is $f_{\max}(A)$ a submodular function? If yes, then prove. If no, then give a counterexample.
- Let $f_{\min}(A) = \min_{x_i \in A} w_i$. That is, $f_{\min}(A)$ is a function that returns the minimum weight of all the elements contained in A. Is $f_{\min}(A)$ a submodular function? If yes, then prove. If no, then give a counterexample.

Problem 2 10 points

Let $G(V, E)$ be an undirected graph. Let S denote some subset of vertices V. Let $\overline{S}$ denote the vertices in V that are not already included in S. We will define a function $c(S)$ to be the number of edges that have one endpoint in S and the other in $\overline{S}$. Is $c(S)$ a submodular function? If yes, then prove. If no, then give a counterexample.

Problem 3 20 points

Consider the problem of assigning papers for peer review. We have a set of n papers that must be assigned to a pool of k reviewers. We will make a modeling assumption that assigning more reviewers to one paper yields diminishing returns in terms of the quality. We want to maximize the overall quality of the reviews which is the sum of the quality of reviews obtained for each paper. But we also have an additional requirement — we would like the reviews for each paper to be from a diverse set of researchers. For the purposes of this exercise, let's say that the k reviewers from $m < k$ research labs. That is, there are labs with more than one reviewer in our pool. Our constraint is that we do not want to assign more than one reviewer from the same lab to the same paper (presumably, since they have the same research background and we are likely to get back similar reviews).

Formalize this problem and discuss potential solutions. Specifically,

- (15 points) Use mathematical notation to describe how you will formalize the problem (5 points). This is a combinatorial optimization problem similar to the ones we have studied in class. At a minimum, show whether the cost function is submodular or not (5 points)? How can we model the additional diversity constraint (5 points)?

- (5 points) Discuss potential solutions. You can start by discussing whether a greedy strategy (clearly explain the greedy strategy) will be a good one or not. If yes, why? If not, why not and what could be a better strategy?

Problem 4 50 points

Perform Gaussian Process regression. You do not need to implement GP regression from scratch. Instead, use either the GPML MATLAB library[1] or the scikit-learn Python library[2].

Problem 4.A 20 points

Here, we want to learn the following 1D function: $f(x) = sin(3x)$ using noisy observations. Our training data is corrupted by additive Gaussian noise. The training inputs are provided in the `problem4a_train.csv` file. Each line in this file has two entries: the first one is the training input location x and the second one is the noisy observation corresponding to that location y. The test input locations are provided in the `problem4a_test.csv` file. This file only contains one entry per line, the test input locations.

You will need to choose an appropriate kernel function. You will also need to set or learn the hyperparameters for that function. Refer to the documentation for GPML MATLAB toolbox and the scikit-learn libraries.

Submission Instructions. You must submit a pdf report that gives a short description of your implementation. In particular, explain which covariance function you used (3 points) along with the hyperparameters that you set or learned (2 points), as well as a plot of the test outputs (5 points). The plot must have on the X-axis, the test input locations $x*$ and on the Y-axis the mean of the GP for that test output, $\mu(x*$. You must also plot two curves: $\mu(x*) + 2 * \sigma(x*)$ and $\mu(x*) - 2 * \sigma(x*)$. Here, $\sigma(x*)$ is the standard deviation given as output by the GP regression. A sample figure is given on ELMS. In the pdf, also compute the mean square error (since we know the ground truth function value, sin($3x$), for the test inputs) (5 points). You must also submit your code with either a Python file named `problem4a_sol.py` or a MATLAB file `problem4a_sol.m` which is the main file that reads in the input and produces the output (5 points). Other supporting files can be submitted but the main ones should be named exactly as described here.

Problem 4.B 30 points

In this problem, instead of using synthetic data, we will use an actual dataset. Specifically, we will use the Combined Cycle Power Plant Data Set[3]. The dataset contains 9568 data points collected from a power plant over 6 years. The input is four-dimensional: temperate (T), ambient pressure (AP), relative humidity (RH), and exhaust vacuum (V). The goal is to predict the net hourly electrical energy output (EP) of the plant given these four quantities.

To make it easy to evaluate, I have separated the dataset into training inputs provided in the `problem4b_train.csv` and test inputs provided in the `problem4b_test.csv`. There are 5 entries (T, AP, RH, V, PE) in the training file and only four (T, AP, RH, V, PE) in the test. You need to report the mean and the standard deviation of the predicted function for the test inputs.

You will need to choose an appropriate kernel function. You will also need to set or learn the hyperparameters for that function. Refer to the documentation for GPML MATLAB toolbox and the scikit-learn libraries.

Submission Instructions. You must submit a pdf report that gives a short description of your implementation. In particular, explain which covariance function you used (10 points) along with the hyperparameters that you set or learned (5 points), as well as a file containing the test outputs (5 points) called as `problem4b_output.csv` (two entries per line giving the predicted mean and standard deviation separated by

[1] `http://www.gaussianprocess.org/gpml/code/matlab/doc/`

[2] `http://scikit-learn.org/stable/modules/gaussian_process.html`

[3] `https://archive.ics.uci.edu/ml/datasets/combined+cycle+power+plant`

a comma for the corresponding inputs in the test file). The ground truth values for the test inputs are also provided on ELMS. Compute and report the mean square error in the predictions (5 points). Student(s) with the lowest error get a bonus of 5 points. You must also submit your code with either a Python file named `problem4b_sol.py` or a MATLAB file `problem4b_sol.m` which is the main file that reads in the input and produces the output (5 points). Other supporting files can be submitted but the main ones should be named exactly as described here.