Problem Set 2
ABE 598 Autonomous Decision Making in the Real World

150 points

Due on 18 April 2018

The problems in this problem set are computer implementation problems. You are free to use any programming language of your choice, including MATLAB. You will submit the code with your assignment. Your code will be graded on its style, commenting, and readability. You are suggested to use good programming techniques, such as writing functions for repeating tasks, indenting your code properly, and choosing most efficient operations (e.g. if possible use matrix manipulations in MATLAB instead of for loops).

If you use online resources or collaborate with others to write your software, please make sure you are citing them correctly. Provide enough comments in your code to illustrate your understanding of the algorithmic process.

Problem 1: (40 points)
Regression problem

The dynamics of the Wing rock Problem are given by:

$$\dot{\phi} = p$$
$$\dot{p} = 0.8 + 0.2314\phi + 0.6918p - 0.6245|\phi|p + 0.0095|p|p + 0.0214\phi^3$$

The file wingrock_main in the folder wingrock simulates the dynamics of an airplane with wingrock and a PD controller. You will first learn a linearly parameterized model of the dynamics. Then, your goal is to learn a GP representation of that dynamics. For this purpose, you can use the GP regression class onlineGP. The folder simple_gp_regression_example contains an example implementation of the onlineGP class which implements Csato and Opper's sparse GP algorithm (with a few modifications). Csato's paper is also in that folder. The file main_simple_gp_regression_example.m shows how the class can be used to learn GP representations of the dataset in gp_regression_example_data.

You should proceed as follows:

1. Collect a dataset: simulate the wingrock dynamics system using wingrock_main with different initial conditions and for 5 seconds runtime in each condition.
2. Write code to learn the weights of the wingrock model for $\dot{p}$ assuming they are unkown. Use the basis from the parameterized model provided. Compare

learned weights with the original weights. Plot time histories of the predicted and true states.

3. GP regression: Now we will learn a data driven GP model with RBF bases using Csato's online GP algorithm. Run the regression: use the onlineGP class to learn a GP representation of wingrock dynamics using the data collected in 1. The output (dependent variables) of the GP should be the second state derivative. The input to the GP (the independent variables) should be the control input and the entire state vector.
4. Validate: create a different dataset on which learning was not performed, and compare the predictions of your learned model with that dataset. Compare in terms of the following metrics:
    a. Mean squared error between your predictions and the true values
    b. What is the mean squared error for the initial conditions you had in the training data set, what is it for those that were far from those in the training data set (be careful, the wing rock system can be unstable for some initial conditions)?
5. Plot time histories of the predicted and true states.

Problem 2: (20) Clustering with K-means

The file gauss_mix_data.mat contains data from 5 Gaussian distributions. Your goal is to implement the K-means algorithm to identify the 5 clusters. The file EM_Gaussian_mixture_model_setup.m shows how you can extract data from the gauss_mix_data.mat file and plots it in different clusters.

How long does your algorithm take to run?
Does it work every time? Do at least 20 runs of your algorithm and report a histogram of showing the number of times your algorithm found the right clusters.

Problem 3: (20) Dirichlet Means: Implement the Dirichlet Process Means clustering algorithm (given below) for the given data-set. Given algorithm takes an input of the cluster penalty parameter lambda, you are required to learn lambda using Farthest first heuristic discussed in the class. *Reference: Revisiting k-means: New Algorithms via Bayesian Nonparametrics, by Kulis and Jordan ICML 2012*.
Given data-set comprises of data generated by 2-D Gaussian distributions as well as ground truth clusters. Your implementation should output

i) The unknown number of generating distributions,
ii) Parameters of the generating distributions,
iii) Value of the hyperparameter, the cluster penalty term lambda,
iv) Computed normalized mutual information (NMI) between ground truth clusters and your algorithm output clusters.

Algorithm 1 DP-means

```
Input: $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$: input data, $\lambda$ : cluster penalty parameter
Output: Clustering $\ell_1, ..., \ell_k$ and number of clusters $k$

1. Init. $k = 1, \ell_1 = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ and $\boldsymbol{\mu}_1$ the global mean.
2. Init. cluster indicators $z_i = 1$ for all $i = 1, ..., n$.
3. Repeat until convergence
    • For each point $\boldsymbol{x}_i$
        – Compute $d_{ic} = \|\boldsymbol{x}_i - \boldsymbol{\mu}_c\|^2$ for $c = 1, ..., k$
        – If $\min_c d_{ic} > \lambda$, set $k = k + 1$, $z_i = k$, and $\boldsymbol{\mu}_k = \boldsymbol{x}_i$.
        – Otherwise, set $z_i = \text{argmin}_c d_{ic}$.
    • Generate clusters $\ell_1, ..., \ell_k$ based on $z_1, ..., z_k$: $\ell_j = \{\boldsymbol{x}_i \mid z_i = j\}$.
    • For each cluster $\ell_j$, compute $\boldsymbol{\mu}_j = \frac{1}{|\ell_j|} \sum_{\boldsymbol{x} \in \ell_j} \boldsymbol{x}$.
```

Problem 3 (20) Neural Networks

Using the dataset from Problem 1 we will create a single hidden layer neural network for wingrock dynamics. Choose the number of hidden neurons to be 10, use the sigmoidal activation function. Perform the following:

1. Write a script to program the neural network. Initialize the weight randomly
2. Implement backpropoagation algorithm
3. Train on the same data set you are using for training in problem 1 and validate the output of the model using the same dataset you are using for validation on problem 1
4. Compare the performance of GP and SHL NN. How does changing the number of hidden neurons affect the performance (reducing and increasing with increments of 2 from 4 to 16)?
5. Does the convergence depend on initial weights for the default case of 10 hidden layers?

Problem 4 (50) Deep Neural Network (ResNets)

In this problem, we will walk you through what a ResNet is and you will program parts of this state of the art classification system and use it to train a classifier to recognize signs. Refer to the jupyter notebook ResNets.ipynb for more details.