EECS 469 / ME 495: Machine Learning and Artificial Intelligence for Robotics

Assignment #2: Machine Learning

Approval due: Nov. 4th **Code A due:** Nov. 11th **Code B due:** Nov. 16th **Writeup due:** Nov. 16th

This assignment will focus on the implementation and assessment of a learning algorithm on a robot dataset. You have a choice of both *what* to learn and *how* to learn it; suggestions and guidelines will be detailed below. Your work can use either of the robot datasets from Assignment #0. The goal is to build a training dataset, implement a learning algorithm, and asses its performance on the dataset.

Make sure that both your dataset *and* learning algorithm are **different** from that of your Assignment #3 partner.

Approval Deliverable: Learning aim, dataset and algorithm approved (over email) by instructor.

Implementation

PART A

1. The first part of this implementation is to build a training dataset. What exactly gets learned is up to you, and there usually is more than one dataset formulation that might accomplish a given learning aim. In formulating a learning aim, try to think of deficiencies in the robot's behavior that might be improved by learning (e.g. dead reckoning error), or hidden structure in the datasets that might be uncovered by learning (e.g. a partitioning of the state space based on landmark visibility in ds1). You are free to use these suggestions or to come up with your own learning aim. Once you have settled on a learning aim, it will be necessary to build a suitable training dataset from either ds0 or ds1. Remember to submit this dataset with your code.

In your write up, describe the reasoning behind your dataset formulation. Include any diagrams or plots to support your reasoning. For example, if your learning aim relates to improving the performance of your motion controller from Assignment #0, you might provide a plot of the positional error, or compute the variance on this error.

2. The second part of this implementation is to code a learning algorithm. You may choose from the following list of algorithms: Neural Networks, Genetic Algorithms, Locally Weighted Regression (but not the no-model formulation), Support Vector Machines, Gaussian Processes or Reinforcement Learning. If you wish to implement a modified version of any of these algorithms (e.g. Sparse Online Gaussian Processes), this is allowed (but not expected).

In your write up, briefly describe the operation of your learning algorithm (including equations as appropriate). Justify (with data) why you expect this algorithm will be able to accomplish your learning aim. For example, if you observe a locally linear relation between the inputs and outputs of your dataset from Step 1, this would support a choice to use Locally Weighted Linear Regression. Make sure to clearly define any parameters, and how you tuned their values. Demonstrate the algorithm's functioning on a simple dataset (e.g. noisy data generated by a sine function).

PART B

3. Now apply your learning algorithm to your dataset developed in Step 1. Evaluate its performance using at least two measures (your choice).

In your write up, include data plots or illustrations as appropriate when presenting the results of the performance measures. Identify any good and poor behavior, and ground your explanations in the maths of the algorithm (or formulation of the dataset).

If you weren't able to get your machine learning algorithm working in Step 2, or see very poor performance from your learning algorithm on your dataset, you might want to verify that your dataset is indeed even learnable, by testing it with someone else's (e.g. Matlab's) machine learning algorithm implementation. You can receive partial credit for Step 3 if you run all of the above analyses with this code (i.e. that you did not write). *This must be clearly acknowledged in your write up.*

Code

To be submitted with the report. (Remember to also submit your dataset.) Development in Matlab, Octave or Python is a requirement. *You must write your own code, from scratch* (i.e. you cannot use existing machine learning libraries). A single executable script file named "run.m" or "run.py" should output, when executed, all of the data (plots, table values) reported in the write-up (as instructed above), with clear presentation and labels. *Code that does not run will receive zero points.* The run file should be easy to read – a sequence of calls to standalone functions, that hide away the meat of code (rather than placing that code in the main function). Clear commenting for all functions and function calls is encouraged. (No need for excessive verbosity, but the quicker and easier we are able to read the code, the better for your grade...)

Write-up

For each Step above, explain thoroughly, and with clean consistent maths, all design decisions (motion model, measurement model, how parameters were set...) and reported assessment results. Remember to label all plot axes, title all plots, provide captions for any figures, include (as appropriate) units for all reported numbers, etc. Language counts – specifically with respect to clarity. If you use other sources, be sure to cite them.