Automation of Biological Research: 02-450/02-750

Carnegie Mellon University

Homework 3

Version: 1.0; updated 2/24/2019

Due: March 18 by 11:59pm

Hand-in via Canvas:

- Your completed code
- A *single* pdf file that contains:
  - your name
  - your Andrew ID
  - your responses to the questions.

Overview

This homework has 2 questions that will require you to implement the DH and DHM algorithms. Both questions will require programming, so **START EARLY AND USE YOUR TIME WISELY**.

Question 1: "DH" (50 points)

Scenario: Cancer is a complex disease involving the uncontrolled growth of genetically heterogeneous masses of cells called tumors. People informally refer to a given cancer by its location (e.g., 'lung cancer', 'brain cancer', 'pancreatic cancer', etc), but the reality is that there may be several cancer subtypes associated with each location which, in turn, may require different treatments. Therefore, it is important to determine the cancer's subtype prior to selecting a treatment. The most accurate way to do this is to perform a biopsy which is an expensive and invasive procedure. Over the past 10 to 15 years, significant research has been devoted to identifying cancer subtype 'biomarkers' from less expensive procedures, including measuring the quantities of various proteins in serum or urine.

In this question you will use the DH algorithm to label the data from 1,000 patients. Each sample is a 1 by 25 vector containing the concentrations of 25 proteins from one of two sources (serum or urine). There are two cancer subtypes (type 1 and type 2). The goal is to learn a model capable of distinguishing the two types from a protein panel. You will also determine

whether it is better to use serum data or urine data when making a prediction. An oracle is available to get the true label for any instance by performing a biopsy.

Provided Files: The files *get_leaves.py, assign_labels.py , best_pruning_and_labeling.py, update_empirical.py* and *load_data.py* are provided to you as subroutines. You do not need to change these files, but you should look at them so that you understand what they take as input and what they return as output. The function **call_DH()** is provided within the file *HW3_call_DH.py* to run your code for parts B-E. It will run the necessary experiments and plot the results.

You will have to complete the implementations of the functions **select_case_1()** and **select_case_2()** in the file *dh.py.* You should find that the implementations largely depend on figuring out how to appropriately call the functions provided to you listed above. Note that this question does **not** utilize modAL.

Tasks:

A. (15 points) Refer to Algorithm 1 in *Hierarchical Sampling for Active Learning (Dasgupta & Hsu)* and complete the implementation of function **select_case_1()** in the file *dh.py.* Here you only need to select nodes from the pruning **proportional to the size of subtree rooted at each node** (see Case 1 in section named "*The select procedure"* in the DH paper). You should read the docstrings so that you understand the inputs and outputs.

B. (5 points) Run the function **call_DH('b')** within the files *HW3_call_DH.py*. It will produce a plot charting the fraction of mis-inferred labels as a function of the number of iterations averaged over 5 separate runs of your DH code using the serum data. Include the plot as part of your report. Provide a qualitative description of your plot- how fast does it converge, how does error change as we apply more iterations? Note that it may take a few minutes for this routine to run, depending on how fast your machine is.

C. (5 points) Run the function **call_DH('c')**. It will produce a plot similar to that above, but using the urine data. Include the plot as part of your report. Describe any differences between the curves in parts B and C. What might account for these differences (hint, you should look at and/or plot the data returned by the functions **load_data("serum")** and **load_data("urine")**

D. (20 points) Complete the implementation of **select_case_2()** in *dh.py.* The only difference between this and the version in part A is that Selection Case 2 uses a **confidence-adjusted selection probability** (see Case 2 in section named "*The select procedure"* in the DH paper). Run the function **call_DH('d')**. It will produce a plot similar as above but with your **select_case_2()** code using the serum data. Compare the results with those from part B. Which selection strategy is more accurate?

E. (5 points) Run the function **call_DH('e')**. It will produce another plot of your **select_case_2()** code using the urine data. Compare the results with those from part D. Which selection strategy is more accurate?

What to hand in: a zip file containing your code and a pdf with your plots and their accompanying explanations.

Question 2 DHM Implementation (50 points)

In this question, you will be implementing DHM algorithm on your own. The pseudocode for DHM can also be found in the paper "A general agnostic active learning algorithm" https://papers.nips.cc/paper/3325-a-general-agnostic-active-learning-algorithm.pdf (see the top of page 4). A code template DHM.py is also provided with some starter code to help you with the implementation and the use of modAL. You will be also comparing DHM query strategy with random query strategy on the same data set. Additionally instructions can be found in DHM.py. Please read the comments in the code template! Please do not change the function signatures, and make sure they return the values asked for when called. The estimator to be used for this part is the support vector machine (SVM) classifier, and you can simply take advantage of scikit-learn package for convenience.

The data used in this question are synthetic two-dimensional data that are not linearly separable by Support Vector Machine (SVM) classifier. The source code for generating the data is included in the code template just for your reference. You can find the data for this part in the /data folder, named as data_DHM.npy and labels_DHM.npy, which can be easily loaded into your workspace by np.load() method.

Part 2.1 (30 points): Implement DHM algorithm as the query strategy. Complete the function DHM() in DHM.py., and you can find more instructions in detail in the comment under this function.

In DHM, there is a step to learn an estimator that is consistent with the pool of data with inferred labels and minimizes the error on the pool of data with queried labels from the oracle, and you can handle this by setting different sample weights. Specifically, for this question, set the sample weights of the data from the pool with inferred labels to be some large number, say 1e9, and set the sample weights of the data from the pool with queried labels as 1.

The expression of the error term involves the VC dimension theory. In your implementation, you are going to use a simplified way of calculating this error term. Use $\beta_t = 0.11 \times \sqrt{dlog\,(t)/t}$, where d = 2 since the dimension of the data is 2.

Part 2.2 (15 points): Complete the function DHMLearner() in DHM.py, which creates an ActiveLearner object and performs active learning with query strategy defined by DHM algorithm. Please read the comment in this function! For this part, you would need to properly maintained the labeled pool (pool of data with inferred labels and pool of data with queried labels from the oracle) and unlabeled pool of data. Make sure fit the estimator on the current labeled pools of data before calculating the accuracy. You may find the source code in the link https://github.com/modAL-python/modAL/blob/master/modAL/models/base.py useful because there you can find a few methods, such as .query(), ._add_training_data(), ._fit_on_new(), etc. that can help you finish the task and can have a better understanding of the methods in the ActiveLearner object in modAL.

Part 2.3 (5 points): Using the function RandomQuery() and RandomLearner() in DHM.py, which perform active learning with a random selection query strategy, make the accuracy plots for DHM and random active learner. Please plots the two curves in one figure and include it in the report. Also include at most three sentences of comments on the results of DHM and random active learner in the plots.