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

Carnegie Mellon University

Homework 2

Version: 1.1; updated 2/28/2021

Due: March 18 by 11:59pm

Hand-in: A single PDF to Gradescope that contains the following items:

1. A cover page that lists your name and Andrew id
    - If you worked on a team, the name(s) and Andrew id(s) of your teammate(s).
        - Teams cannot be larger than 3 people.
        - Note: Each team should only hand in one pdf. It does not matter who does the actual upload.
        - Indicate your partners in Gradescope
    - Points will be deducted if you do not have a cover page or fail to indicate your partners
2. A PDF export of the Jupyter notebook for questions 1 and 2

You can combine all the PDFs into one using Adobe Acrobat, or similar tool.

Overview

This homework consists of 2 questions, 50 points each. The purpose of the homework is to have you implement and test two Type II Active Learning Algorithms

Question 1 Implement the ZLG (50 points)

Provided Files: The dataset provided for this problem is the same as question 2, but you will only be using a subset of it; see the jupyter notebook template for more information.

You will have to complete the implementations of the functions **Laplacian_matrix(), minimum_energy_solution(), expected_estimated_risk,** and **zlg_query()** in the file *Q1.ipynb.*

The implementations should be very straightforward if you read the paper carefully and understand the equations. Note that this question does **not** utilize modAL.

Tasks:

A. (15 points) Read the paper and answer: (1) What is the idea behind the ZLG algorithm? (2) What are the assumptions behind the ZLG algorithm? (3) What are the pros and cons of the algorithm?

B. (5 points) Implement **Laplacian_matrix ()** and use it to compute the Laplacian matrix of X.

C. (5 points) Implement the function **minimum_energy_solution()**. It will produce the inverse matrix of the submatrix corresponding to unlabeled points ($\mathbf{\Delta}_{uu}^{-1}$) along with the minimum energy solution of all unlabeled points.

D. (15 points) Complete the implementation of **expected_estimated_risk()**. You will need to refer to the section of the paper starting from equation (4).

E. (5 points) Complete the function **zlg_query()**. Try query 100 points and print out the indices. Note that it may take a few minutes to run, depending on how fast your machine is.

F. (5 points) Compare with random queries and make a plot.

G. (For fun) For this dataset, how many labeled data points do you actually need, to train the model sufficiently well? And why?

Question 2 Implement DH algorithm (50 points)

Scenario: Subcellular localization of protein is important information, surface and secreted proteins are the potential target for vaccine and drug, aberrant of subcellular localization of protein is related to several diseases including cancer and Alzheimer's disease, however, experimentally determining the localization can be costly. Directly predicting the subcellular localization of a protein from its sequence is very attractive, we will try to address this problem.

The original dataset has 1484 samples (proteins), each sample has 8 attributes, each attribute is a feature of the protein sequence that is related to the subcellular localization of the protein. We will use only the mitochondrial "MIT" and nuclear "NUC" proteins, which contains a total of 673 samples (the corresponding samples has been extracted to data.csv file).

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:

H. (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.

I. (5 points) Run the function **call_DH('b')**. 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. 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.

J. (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('c')**. It will produce a plot similar as above but with your **select_case_2()** code. Compare the results with those from part B. Which selection strategy is more accurate?

K. Answer the questions related to DH algorithm (10 points):

   What is a "admissible pair" according to the paper (5 points)?

   Please explain the sampling bias that is dealt with in the DH algorithm and why it would be a problem if we just query the unlabeled point which is closest to the decision boundary (5 points)?

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