Automation of Scientific Research
Comp Bio 02-450/02-750
Spring 2024

Homework Assignment #3

Assigned: Mar 12, 2024

Due: Thursday, Mar 21, 2023 by 11:59pm

Three exercises: 100 points in total (Exercise 1: 30 points; Exercise 2: 30 points; Exercise 3: 40 points)

Instructions:

Please submit this assignment in two parts: well-commented code and document. For the code, please submit a single package containing Jupyter notebooks and corresponding datasets. Your submission package should be compressed and named firstname_lastname_hw3.zip (e.g., jose_lugo-martinez_hw3.zip). **In your package there should be everything necessary to successfully execute your code.** For this homework, you should submit three Jupyter notebooks with prefix exercise-1, exercise-2, and exercise-3 corresponding to Exercise 1, Exercise 2, and Exercise 3, respectively. Each program should solve the assigned exercise. Make sure to add comments to each program, including your name. In the case of the report, you should submit a single PDF file named the firstname_lastname_hw1.pdf reporting **all answers, all figures along with description, and all relevant results and discussion.** This report must be **typed** and make sure that you type your name and CMU username on top of the first page of PDF file. **Finally, you may use whatever ML packages you find helpful; however, the implementation of the query selection algorithms should be your own.**

Academic Integrity: All assignments are individual, except when collaboration is explicitly allowed. All the sources used for problem solving must be acknowledged (e.g., web sites, books, research papers, personal communication with people). Academic integrity is taken seriously! For detailed information refer to the syllabus section on Academic Integrity.

Exercise 1 (30 points): Active Learning vs Design of Experiments The file "ex1_data.csv" contains the dataset you should use for a regression-based simulation.

a. (20 points) Implement a Design of Experiments (DOE) strategy of your choice. Describe in a few sentences **how you approximate the solution of the DOE strategy and provide corresponding details for that approach.** Generate a plot comparing the performance (and standard deviation) on unobserved instances between your DOE strategy against passive learning and uncertainty sampling (Exercise 3 of Homework 1) after 30 observations across 10 simulations. Since your DOE implementation will most likely be non-deterministic, each simulation may lead to different DOE results. Discuss whether the results matched your expectations and **explain your reasoning**.

b. (10 points) Run active learning simulations using uncertainty sampling and your favorite base learner on the regression dataset. For one set of simulations, initialize with 10 randomly selected instances. For another set of simulations, initialize with 10 DOE selected instances. Continue both types of simulations until 50 instances are observed. **In a single plot**, show the performance (and standard deviation) of predictions on the unobserved set as a function of the number of instances observed. Discuss whether the results matched your expectations and **explain your reasoning**.

Exercise 2 (30 points): Implementing a Type II algorithm (ZLG, DH, or PLAL) The file "ex2_data.csv" contains the dataset you should use for implementing a clustering-based algorithm.

a. (25 points) Implement ONE of the following three Type II algorithms discussed in class: Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions (ZLG algorithm) as described in Zhu et al. (2003), Hierarchical Sampling for Active Learning (DH algorithm) as described in Dasgupta & Zhu (2008), or PLAL: Cluster-based Active Learning (PLAL algorithm) as described in Urner et al. (2013).

b. (5 points) Comment on the results in part 2a compare against any of the Type I algorithms implemented in previous homework assignments. Discuss whether the results matched your expectations and **explain your reasoning**.

Exercise 3 (40 points): Sequential Bayesian Optimization In this exercise you will be working with a dataset of MHC class I binding affinity prediction dataset ("ex3_data.csv"), which is important for peptide-based vaccine development. You will simulate a peptide design experiment, trying to find peptides with high binding affinity to MHC class I using a Bayesian optimization approach. Notice the goal here is not trying to find a peptide sequence that maximize the binding affinity to MHC, Since a sizable proportion of the sequence data we are using contains maximum binding affinity out of the data (9.0). In this exercise, you will examine several techniques to maximize the percentage of sequence with affinity of 9.0 for stringent querying budget.

Since we are dealing with machine learning models, you will need to convert peptide sequences into feature vectors. The simplest way to do this is to use a one-hot encoding. Each character in the amino acid alphabet will correspond to a binary vector with a single 1 and all 0s. The size of the vector is equal to the size of the amino acid alphabet. The position of 1 encodes a specific amino acid. The resulting feature vector for a peptide is a concatenation of the feature vectors of its amino acids. Since we are dealing with 9-mers here, the size of the feature vector for a peptide should be equal to 9*(size of the amino acid alphabet). **Finally, randomly split the data into train and test data sets.**

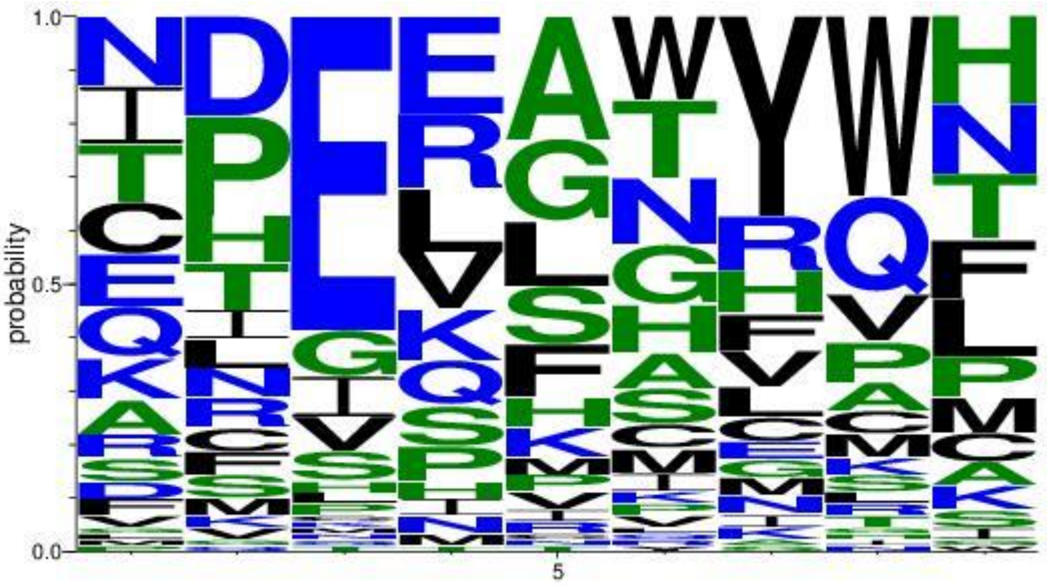
a. (20 points) Implement a Bayesian optimizer with Gaussian Process as regressor and use any selection/acquisition function that combines exploitation and exploration. **Make sure to concisely describe (i.e., no more than a few**

sentences), how you defined your selection function. If the data selected is a new sequence with binding affinity of 9.0, append it to a list. After each query selection, measure the percentage of sequences with binding affinity 9.0 found by the strategy. Do this for 200 sampling steps.

b. (5 points) Implement a random query strategy for randomly selecting a sample to query from the data. If the data selected is a new sequence with binding affinity of 9.0, append it to a list. After each query selection, measure the percentage of sequences with binding affinity 9.0 found by the strategy. Do this for 200 sampling steps. This will serve as the baseline to compare with performance in part 3a.

c. (5 points) Plot the cumulative percentage of sequences with maximum binding affinity with respect to number of sequences queried for 3a and 3b.

d. (5 points) Create sequence logo based on sequence found with each querying strategy. A *sequence logo* is a graphical representation of the sequence conservation of amino acids in protein sequences, as amino acids that are important for functions are likely to be conserved. Hence, a sequence logo is a way to visualize such an importance. Convert each set of sequences obtained by both of your strategies to a sequence logo. Below is an example using all the sequences of affinity 9.0 from the original data. You can use seqlogo to create sequence logo from our set of sequences.



e. (5 points) Compare the approaches in parts 3a and 3b. Discuss whether the results matched your expectations and **explain your reasoning**.

Recommended approach for 5-fold cross validation accuracy calculations for one round of one simulation: Split your training data at that round into 5 different sets of equal size (or as equal as they can be). Train a model using 4 of the 5 sets. Assess that model on the remaining set. Continue this process until each set has been used for assessment once. Add up the errors from all 5 folds and divide by the total number of observed instances. This will yield an average error that will not be severely biased by imbalances in the fold sizes. This will give you a good estimate of the model performance given the data you have available to you at that round. This will not necessarily give you an estimate of generalization performance.

Recommended approach for determining accuracy on the unobserved set for one round of one simulation: Using all your training data for that round, train a model. Assess that model on all the unobserved instances in your simulation at that round. This will give you an estimate of the performance on unobserved experiments.

Note: Your simulations should have the same random initialization sets. You can accomplish this by picking 10 random seed values and using those same seeds to initialize each of your simulations for different selection methods. This eliminates any potential bias in your results caused by different initialization sets.