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

Carnegie Mellon University

Homework 3

Version 1.0; updated 3/8/2021

Due: April 6, 2021 by 11:59pm

Overview

This homework consists of 2 questions, 50 points each. You will be working with a dataset of MHC class I binding affinity prediction dataset, which is important for peptide-based vaccine development. In the first question, you will build a regression model for predicting pIC50 (negative logarithm of half maximal inhibitory concentration) of 9-mer peptides to a specific allele of MHC-1 (see the dataset file included with this homework). You will use active learning to train an efficient model with little labeled data. In the second question, you will simulate a peptide design experiment, trying to find peptides with high binding affinity to MHC class I using a bayesian optimization approach. Both questions require using scikit-learn and modAL packages.

Your submission should include a pdf file with your code, plots, and comments.

Question 1 (50 points)

In this question, you need to build a regression model with active learning for predicting binding affinity between MHC class I and small peptides. The dataset is provided in file hw3_data.csv. Below are the step-by-step instructions:

1. (5 points) Read the data into the jupyter notebook. Columns 2 and 3 in the dataset file have peptide sequences and pIC50 values correspondingly.
2. (10 points) Since we are dealing with machine learning models, you 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 zeros. 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).

3. (20 points) Then we split data into train and test datasets. Now you can start building the model with active learning! We do not give you any specific instructions as to which algorithm to implement. You can use something similar to this example in modAL documentation, or query-by-committee when a label is requested for samples which the committee is least confident about or anything else covered in the lectures.
4. (5 points) The quality of the model should be measured on the test set as R^2 score. The minimum acceptable R^2 score is 0.6.
5. (5 points) Compare your results with no active learning scheme by training a random forest classifier on the same amount of data points, but selected randomly.
6. (5 points) Write a paragraph about your method, describe your observations and difficulties.

Question 2 (50 points)

In this question, 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). Using the same feature encoding as question 1, we will examine several techniques to maximize the percentage of sequence with affinity of 9.0 for stringent querying budget.

Feel free to define any helper functions as you see fit.

1. Random Sampling (5 pts. total)

   Create 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 section 2.2 and 2.3.

2. Bayesian Optimization with Gaussian Process (15 pts. total)

   Create a Bayesian optimizer with Gaussian process as regressor and Max Expected improvement as the queuing strategy. 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.

   Check the modAL documentation for how to set up a Bayesian optimizer.

3. Bayesian Optimizer with Random Forest (15 pts. total)

Although Bayesian optimization often uses the Gaussian process, Bayesian optimizer in ModAL can take any other regressor that has a predict function with a return_std input parameter. If return_std is set to True, the function returns the predicted values and standard deviation in the prediction. Create a Bayesian optimizer with random forest regressor and Max Expected improvement as the queuing strategy. 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.

4. Plot Percentage of sequences with maximum binding affinity with respect to number of sequence queried (10 pts. total)

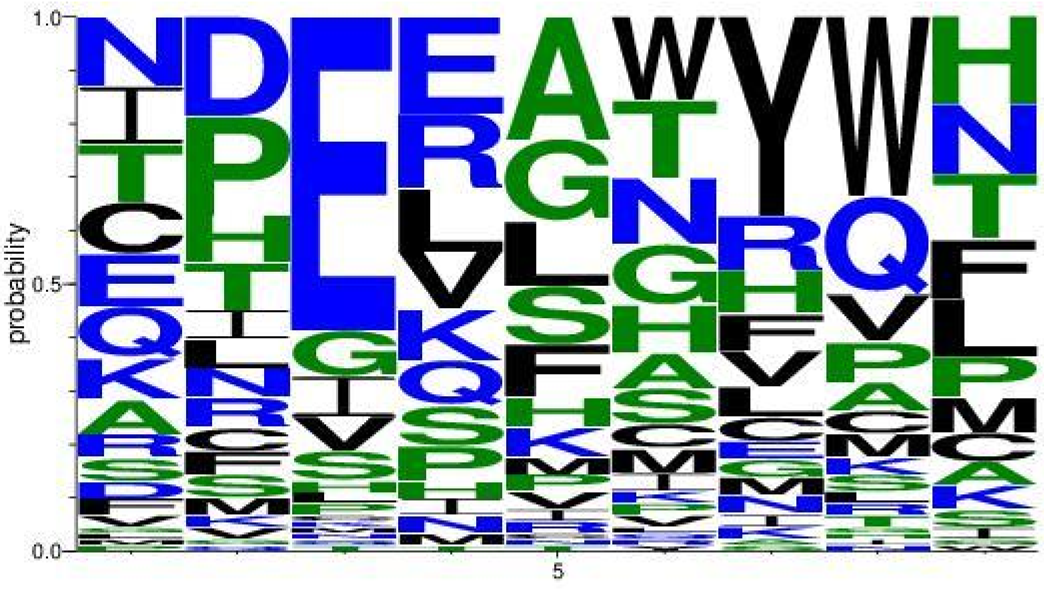
5. Create sequence logo based on sequence found with each querying strategies (5 pts. total)

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 the each sets of sequences obtained by one of your optimization strategies to a sequence logo. Below is an example using all of the sequences of affinity 9.0 from the data.

We are using seqlogo to create sequence logo from our set of sequences. You can install seqlogo by entering the command

conda install -c bioconda seqlogo

in your conda terminal


probability