COMP 5970/6970 Project 4: 100 points 20% Credit
Final Submission due before 11:59 PM Monday April 6

Instructions:

1. This is an individual project. You should do your own work. Any evidence of copying either from a public source or from the works of other without due credits will result in a zero grade and additional penalties/actions**.**
2. **Submissions by email or late submissions (even by minutes) will receive a zero grade.** No makeup will be offered unless prior permission has been granted, or there is a valid and verifiable excuse.
3. **No show for your project presentation will receive a zero grade. There is also a penalty for missing a presentation day in which you are not presenting.**

Submission:

For 5970, you are required to upload the following to canvas before **11:59 PM Monday April 6**:

1. **Source Code**: Python source files (upload .zip file in case of multiple files) containing your code only (no test data needed) and ReadMe.txt file (template provided) describing how to run your code. Note that we will NOT debug your code. If your code does not execute as described in ReadMe.txt, you will receive a zero grade.
2. **Presentation Slide**: One slide only in PPT/PPTX/PDF format to be used during the oral presentations (see below). If you submitted file spans more than a page, we will extract the first page for the oral presentation.

For 6970, you are required to upload the following to canvas before **11:59 PM Monday April 6**:

1. **Source Code**: Python source files (upload .zip file in case of multiple files) containing your code only (no test data needed) and ReadMe.txt file (template provided) describing how to run your code. Note that we will NOT debug your code. If your code does not execute as described in ReadMe.txt, you will receive a zero grade.
2. **Presentation Slide**: One slide only in PPT/PPTX/PDF format to be used during the oral presentations (see below). If you submitted file spans more than a page, we will extract the first page for the oral presentation.
3. **Project Report**: Completed report document in PDF format using template provided. Make sure to have all necessary sections of scientific writing: abstract, introduction, methods, results, discussion, references.

Presentations:

Presentations will be during the class on **Wednesday April 8** and **Friday April 10**.

Attendance is mandatory during all the presentation days. Missed presentation days without university-approved excuse will result in a penalty of 25 points for each missed class. Note that this penalty will be applied when you miss a presentation day in which you are not presenting. **No show for your project presentation will receive a zero grade.**

Everyone is required to deliver 3 minutes flash presentation accompanied by the submitted slide following the Three Minute Thesis (3MT) format, with additional 2 minutes for Q&A:

1. Your presentation should at least contain methods (i.e., implementation), results (e.g., output), and conclusion.
2. Having appropriate graphics and visuals (e.g., figures, plots) in the presentation slides to help illustrate key concepts or results will be positively graded.
3. Any additional scientific insights and/or challenges faced and/or limitations of your implementation and/or efficiency analyses and/or comparisons with alternative approaches will be positively graded.
4. Practice your talk not to exceed the time limit or finish too early.
5. No need to bring your slides. We will set things up and decide the presentation sequence.

Implementing Logistic Regression for Protein Contact Map prediction

Objective: Implement logistic regression for protein residue-residue contact map prediction.

Note: You must use standard Python programming language. You are NOT allowed to use non-standard packages or libraries (e.g. Biopython, scikit-learn, SciPy, NumPy, etc.).

A: Raw Data:

Two sets of 150 data files (<*pdb_id*>.*pssm* and the corresponding <*pdb_id*>.*rr*) are supplied. Each <*pdb_id*>.*pssm* file contains a evolutionary profile of a single protein sequence.

The corresponding <*pdb_id*>.*rr* file contains the all the residue-residue contacts in RR format.

RR format starts with the sequence of the protein target. The sequence is followed by the list of contacts in a five-column format (note that only contacts are present, not the non-contacts):

i j d1 d2 d :

- indices i and j of the two amino acid residues in contact such that i < j, i.e. only half of the contact map is supplied. Furthermore, |i – j| > 5, i.e. the sequence separation between the two residues in contact is at least six.
- the numbers d1 and d2 indicate the distance limits defining a contact. A pair of residues is defined to be in contact when the distance between them is less then 8 Angstroms (Å). Therefore, typically d1= 0Å and d2= 8Å.
- the real number d indicates the actual intra residue distance of the two residues being in contact in Angstroms. (During classification, this column can be used to store the probability of contact).

An example RR format is provided below:

```
AAYKVTLVTPTGNVEFQCPDDVYILDAAEEEGIDLPYSCRAGSCSSCAGKLKTGSLNQDDQSFLDDDQIDEGWVLTCAAYPVSDVTI
1 19 0 8 6.006
1 20 0 8 5.264
1 21 0 8 7.431
10 89 0 8 7.422
10 90 0 8 5.058
14 28 0 8 7.710
14 31 0 8 7.623
14 33 0 8 5.962
16 27 0 8 6.549
16 28 0 8 6.822
16 31 0 8 4.480
16 33 0 8 7.700
18 24 0 8 6.397
18 27 0 8 4.137
       .
       .
       .
```

N.B. The RR files are extracted from the tertiary structures of the proteins deposited in the Protein Data Bank (PDB).

B: Curating Training and Test Datasets:

Divide the raw data into non-overlapping sets of training (~75%) and test (~25%) datasets using simple random sampling without replacement.

C. Feature Generation:

Use the 20 PSSM values for each pair of residues (i, j), where j > i +5, in a protein from the .pssm file. Fist few lines for a test .pssm file are given below. For a protein sequence of N residues, there will be N x 20 PSSM values.

		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	E	-3	-2	-2	3	-6	4	5	-4	-3	-6	-5	3	-4	-6	-4	-3	-3	-5	-5	-5
2	A	1	-2	-2	2	-4	1	1	0	-4	-3	-1	-2	-2	-5	2	3	1	-2	-5	-1
3	E	-1	4	-2	1	-5	1	0	-3	0	-2	-3	3	-2	-4	3	0	0	-5	-4	-1
4	A	0	0	-2	0	-5	2	1	-1	-2	-4	-5	0	-3	-6	6	-1	-1	-6	-5	-4
5	S	1	-2	1	5	-5	-1	3	-1	-1	-5	-5	0	-3	-6	0	1	-1	-6	-5	-4
6	I	-1	-2	-5	-5	-4	-4	-3	-5	0	3	0	-3	-1	5	-2	-4	-1	-2	1	3
7	C	-2	-6	-5	-6	11	-6	-6	-5	-6	-4	-4	-6	-4	-5	-6	-3	-3	-5	-5	-3
8	S	-1	0	1	-1	-3	1	0	-3	3	-2	2	0	0	0	-5	1	0	-1	1	-2
9	E	-1	-3	-5	-3	-4	3	2	-5	-3	-2	5	-3	2	-1	-3	-3	-3	-2	-3	-1
10	P	-2	-3	-4	-3	-5	-1	0	-5	-3	-5	-4	-2	-4	-6	8	-1	-3	-6	-5	-4
11	K	2	0	-3	-3	-5	0	-1	-4	-1	-2	-2	4	1	-5	4	0	-3	-5	-4	0
12	K	-1	-2	-1	5	-5	-2	3	-4	-2	-2	-2	2	-2	-4	-4	-2	-3	-5	0	2
13	V	-1	0	-3	-3	-3	0	0	-3	-4	-1	-3	0	-2	-4	3	1	3	-5	-3	3
14	G	-2	-4	-3	-4	-5	-4	-4	7	-5	-7	-6	-4	-5	-6	-4	-3	-4	-5	-6	-6
15	R	-3	2	-2	-3	-5	-3	-2	-4	1	-1	-2	-1	-2	-3	7	-2	-1	-3	-2	-2

Additionally, use a sliding window of 5 around each residue pairs (i.e. i ± 2 and j ± 2) for feature generation. Therefore, there will be 20 x 5 x 2 = 200 PSSM values for each non-terminal residue pairs. For terminal residues that do not have one or more neighbors on either side, use -1 as dummy PSSM values. Use the corresponding RR files to generate binary class labels: 1 if (i, j) is in contact, 0 otherwise. Note that this needs to done for all pairs of residues in a protein.

D. Logistic Regression Learning on Training Set:

Implement the gradient ascent based optimization algorithm to learn the weight vector of Logistic Regression using the training set. You may choose to optimize MCLE via batch gradient ascent or stochastic gradient ascent (or a combination of both).

E. Logistic Regression Classification on Test Set:

Implement Logistic Regression classifier that uses the learned weight vector to calculate the probability all pairs of residue pairs (i, j), with $|i - j| > 5$, to be in contact give a single PSSM formatted file. Sort the contacts by non-increasing probabilities and save them in RR format.

N.B. Logistic Regression is an offline-learning algorithm. Therefore, training and classification should be implemented separately. The classification algorithm should take a test file in PSSM format as an input and predict RR file in a standalone mode. You may save the parameters learned during training in a file that can be fed into the classifier, in an offline mode.

F. Evaluate Accuracy:

Report % accuracy of '**top'** L/10, L/5, L/2 predicted contacts to evaluate the classification performance averaged over the proteins in the test dataset, where L is the length of the protein sequence and 'top' contacts are the contacts predicted with high probabilities (i.e. present at the top of the predicted RR file, which is sorted by non-increasing probabilities).