Advanced Learning for Text and Graph Data (ALTEGRAD)
Cellular Component Ontology Prediction

École Polytechnique

December 2022

1 Description of the Challenge

The goal of this project is to study and apply machine learning/artificial intelligence techniques to a classification problem from the field of bio-informatics. Machine learning for protein engineering has attracted a lot of attention recently. Proteins are large biomolecules and macromolecules composed of one or more long chains of amino acids, and are an essential part of all living organisms. Among other, they enable chemical reactions to occur in cells by acting as enzymes and promoting specific reactions, and also provide structural support and play a key role in the immune system's ability to distinguish self from invaders. They consist of small molecules called amino acids, with long proteins containing up to $4,500$ of these amino acids. There are 20 different amino acids commonly found in the proteins of living organisms. When amino acids bind together, they form a long chain called a polypeptide that can be represented by the protein sequence. The sequence of amino acids then begins to fold, creating the 3D shape of the protein. This structure determines its specific chemical functionality, but the exact details of this process are not yet fully understood. In this challenge, you are given the sequence of $6,111$ proteins along with the graph representation of their structure. The nodes of these graphs represent the amino acids and two nodes are connected by an edge based on the Euclidean distance between these residues, their order in the sequence, and the different chemical interactions between them. The goal of this challenge is to use the sequence and structure of those proteins and classify them into 18 different classes, each representing a characteristic of the location where the protein performs its function obtained from the Cellular Component ontology.

The challenge is hosted on Kaggle and is available at the following link: `https://www.kaggle.com/c/altegrad-2022`. To participate in the challenge, use the following link: `https://www.kaggle.com/t/5b4594555b1d4d7fb25d33286df1208f`.

2 Dataset Description

As mentioned above, you will evaluate your methods on a dataset consisting of proteins. The dataset contains $6,111$ proteins in total. You are given the following files (which are available at: `https://tinyurl.com/2p9cp9m3`).

1. **sequences.txt**: it contains the sequence of each one of the $6,111$ proteins. Each line consists of a string of letters that represent the sequence of one protein.

2. **edgelist.txt**: it lists the edges of all proteins. Each line corresponds to an edge which is represented by the ids of its endpoints. Note that proteins are modeled as undirected graphs. There are $15, 213, 222$ edges in total (for all $6, 111$ proteins).

3. **edge_attributes.txt**: it provides attributes for the edges of the $6, 111$ proteins. Each line stores the attributes of an edge (the i-th line stores the attribute of the corresponding edge in the **edgelist.txt** file). There are 5 attributes in total (described below) separated with the comma character.

Attributes	Description
1	distance between the two connected nodes
2-5	binary indicator of whether the edge is of each one of the following four types: (i) distance-based edge, (ii) peptide bond edge, (iii) k-NN edge, and (iv) hydrogen bond edge

4. **node_attributes.txt**: it contains the attributes of the nodes (i.e., amino acids) of the $6, 111$ graphs. Each line stores the attributes of a node (the i-th line stores the attribute vector of the node with id i where i starts from 0). There are 86 attributes in total (described below) separated with the comma character, and $1, 572, 264$ nodes in the $6, 111$ proteins.

Attributes	Description
1-3	3D coordinates of the node represented by x, y, z
4-23	one hot encoding of the amino acid type (there are 20 amino acids in total)
24	hydrogen bond acceptor status
25	hydrogen bond donor status
26-86	amino acid features derived from the EXPASY protein scale (`https://web.expasy.org/protscale`)

5. **graph_indicator.txt**: this file contains graph identifiers for all nodes of all $6, 111$ graphs. The value in the i-th line denotes the graph to which node with id i belongs. There are $1, 572, 264$ values in total.

6. **graph_labels.txt**: this file contains the names of all proteins along with the class labels of those proteins that belong to the training set. Thus, the i-th line contains the name of the protein with id i and potentially its class label (in case it belongs to the training set). The comma character (,) is used to separate the name from the class label. The proteins for which the class label is not available belong to the test set. The final evaluation of your methods will be done on these proteins and the goal will be to predict the class label of each one of those proteins.

3 Evaluation

The performance of your models will be assessed using the logarithmic loss measure. This metric is defined as the negative log-likelihood of the true class labels given a probabilistic classifier's predictions. Specifically, the multi-class log loss is defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij})$$

where N is the number of samples (i.e., proteins), C is the number of classes (i.e., the 18 categories), y_{ij} is 1 if sample i belongs to class j and 0 otherwise, and p_{ij} is the predicted probability that sample i belongs to class j.

4 Provided Source Code

You are given two scripts written in Python that will help you get started with the challenge. The first script (`sequence_baseline.py`) uses solely features extracted from the sequence of a protein with a logistic regression classifier for making predictions. The second script (`structure_baseline.py`) feeds the graph representation of the structure of each protein to a graph neural network to predict the class to which the protein belongs. As part of this challenge, you are asked to write your own code and build your own models to predict the class to which each protein belongs. You are advised to use both graph-theoretical features and features extracted from the sequence.

5 Useful Python Libraries

In this section, we briefly discuss some tools that can be useful in the challenge and you are encouraged to use.

- A very powerful machine learning library in Python is `scikit-learn`[1]. It can be used in the preprocessing step (e.g., for feature selection) and in the classification task (several regression algorithms have been implemented in `scikit-learn`).
- A very popular deep learning library in Python is `PyTorch`[2]. The library provides a simple and user-friendly interface to build and train deep learning models.
- Since you will deal with data represented as a graph, the use of a library for managing and analyzing graphs may be proven important. An example of such a library is the `NetworkX`[3] library of Python that will allow you to create, manipulate and study the structure and several other features of a graph.
- Since you will also deal with textual data, the Natural Language Toolkit (`NLTK`)[4] of Python can also be found useful.
- `Gensim`[5] is a Python library for unsupervised topic modeling and natural language processing, using modern statistical machine learning. The library provides all the necessary tools for learning word and document embeddings.

6 Rules and Details about the Submission of the Project

Rules. The following rules apply to this challenge: (i) one account is allowed per participant (ii) there is a limit in the size of each team (at most 3 members), (iii) privately sharing code outside of teams is not permitted, (iv) there is a limit in the number of submissions per day (at most 5 entries per day), (v) use of external data is not allowed (except from pre-trained models). For instance, you are not allowed to use external data to determine the class label of a protein.

[1] `http://scikit-learn.org/`

[2] `https://pytorch.org/`

[3] `http://networkx.github.io/`

[4] `http://www.nltk.org/`

[5] `https://radimrehurek.com/gensim/`

Evaluation and Submission. Your final evaluation for the project will be based on (1) the presentation you will give (**50**%), (2) on your position on the private leaderboard and the log loss that will be achieved (**20**%), and (3) on your total approach to the problem and the quality of the report and the code (**30**%). As part of the project, you have to submit the following:

- A 4-5 pages report, in which you should describe the approach and the methods that you used in the project. Since this is a real classification task, we are interested to know how you dealt with each part of the pipeline, e.g., how you created your representation, which features did you use, which classification algorithms did you use and why, the performance of your methods (log loss and training time), approaches that finally didn't work but is interesting to present them, and in general, whatever you think that is interesting to report.
- A directory with the code of your implementation (not the data, just the code).
- Create a `.zip` file containing the code and the report and submit it here. Make sure that the name of the file is as follows: `team_name.zip` where `team_name` is the name of the team on Kaggle. Also, the names of all members of each team must be listed in the report.
- **Deadline (both competition and for submitting code and report): 22/1/2023 23:59**

Presentation: As mentioned above, you will be asked to present the approach you followed. Therefore, you will need to prepare some slides (using ppt or any other tool you like).
Date of presentation: TBA