CSI 5180. Topics in Artificial Intelligence: Machine Learning for Bioinformatics

Assignment 1

Marcel Turcotte

Version: Feb 3, 2025 17:40

🎯 Learning Objectives

- **Write** and **execute** a Jupyter Notebook.
- **Download** and **analyze** data in a cloud environment.
- **Prepare** biological data for a machine learning project.
- **Perform** an exploratory data analysis.
- **Train** a machine learning model.

Assignments 1 and 2 will utilize the dataset associated with the publication by Umarov and Solovyev (2017), which investigates the recognition of prokaryotic and eukaryotic promoters through convolutional deep learning neural networks[1].

- Umarov, R. K. and Solovyev, V. V. (2017). Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks. *PLoS ONE*, 12(2):e0171410.

In Assignment 1, the task involves developing a Jupyter Notebook to load and encode the dataset, visualize the data, and train one machine learning model. In Assignment 2, the analysis is extended by exploring alternative encoding techniques and evaluating the performance of different machine learning algorithms.

Promoter sequences are specific regions of DNA that signal the initiation site for gene transcription. In eukaryotic organisms, these sequences are typically situated 25 to 35 base pairs upstream of the transcription start site (TSS), as detailed by Scitable by Nature Education.

The publication by Umarov and Solovyev (2017) investigates two categories of promoters: TATA box promoters and non-TATA box promoters, which serve as positive examples. To construct a set of negative examples, the authors randomly selected DNA sequences from the reverse complementary strand of protein-coding regions.

- CNNPromoterData on GitHub

In our experiments, we achieved an accuracy rate of approximately 90% by employing a straightforward k-mer encoding strategy in conjunction with logistic regression, as implemented in the scikit-learn library.

[1] I have some reservations regarding the publication, particularly concerning the methodology for generating negative examples and the protocol employed in the machine learning experiments. Consequently, I would advise against using this paper as a definitive model. Despite these issues, the underlying problem is straightforward, allowing for the derivation of robust results with relative ease. Furthermore, the paper conveniently provides access to valuable data!

Submission

- **Deadline:**
  - The deadline for submitting your notebook is February 24 at 8 PM.
- **Individual Assignment:**
  - This task requires individual effort; collaboration is not permitted.
- **Submission Platform:**
  - Please upload your completed notebook to the Assignment section (Assignment 1) on Brightspace.
- **Submission Format:**
  - Please ensure that your submission comprises a Jupyter Notebook containing all results and an accompanying PDF version of the notebook. The PDF will facilitate discussions regarding the evaluation of your work with your teaching assistant.
  - The code should be executable on Google Colab. If your work involves additional libraries, include appropriate installation commands for those libraries.

Note for Submission: To avoid receiving a zero, ensure that your code executes correctly. You are responsible for verifying that your submission functions properly on a computer other than your own. Make sure all cells in your notebook are executable.

Deliverable

1. Jupyter Notebook

Develop a Jupyter Notebook that explicitly includes the course title, assignment details, and your personal information, such as your name and student identification number.

2. Loading the Dataset

Your notebook must read the data directly from the GitHub repository:

- CNNPromoterData on GitHub

To prevent errors when accessing data from GitHub, it is advisable to utilize URLs that direct to the raw data files. For example, the raw data for the file Arabidopsis_non_prom.fa can be accessed via the following URL. Here are the three links for your work.

- Arabidopsis_tata.fa (positive)
- Arabidopsis_non_tata.fa (positive)
- Arabidopsis_non_prom_big.fa (negative)

You will be conducting two experiments:

- **positive** = 'Arabidopsis_tata.fa', **negative** = 'Arabidopsis_non_prom_big.fa'.
- **positive** = 'Arabidopsis_non_tata.fa', **negative** = 'Arabidopsis_non_prom_big.fa'.

The data is encoded using the FASTA format format, a widely recognized and straightforward file format in bioinformatics. A FASTA file comprises one or more sequences, each initiated by a single-line description that begins with a '>' character. This descriptor line, which can typically be disregarded during data processing, is followed by the sequence data itself. The sequence is usually distributed across multiple lines, each typically not exceeding 80 characters. This line-length convention enhances readability and was historically required for compatibility with older computer systems. In the dataset utilized for this assignment, the line-by-line convention is not implemented, as each sequence is presented on a single line.

3. Data Encoding

To effectively utilize machine learning algorithms in bioinformatics, it is essential to transform biological sequence data into a format amenable to computational analysis. One prevalent method for achieving this is through the use of k-mers. In this context, each biological sequence is converted into a frequency vector that represents the occurrence of all possible nucleotide tuples of length k, where k is a user-defined parameter, typically set to $k = 4$ in our study.

For a given sequence S and a specified k, the frequency of each k-mer, denoted as x_j, is calculated based on the tuple $[ACGT]^k(j)$ present in S. It is important to note that any tuple containing characters outside the nucleotide alphabet (A, C, G, T) are excluded from this analysis.

This k-mer based frequency vectorization offers a robust framework for comparing sequences. Identical or highly similar sequences yield similar frequency vectors, whereas sequences that have diverged due to mutations exhibit increasingly dissimilar vectors. This approach is advantageous as it facilitates the comparison of sequences of varying lengths and is resilient to certain evolutionary changes, such as segment rearrangements or duplications. Nonetheless, it is important to acknowledge that this method does lead to the loss of some sequence information.

To illustrate, consider the sequence S = GAAGAC, composed of the nucleotides A, C, G, and T. When employing k-mers of size 2, the frequency distribution vector is constructed as follows:

```
AA = 1/5
AC = 1/5
AG = 1/5
AT = 0
CA = 0
CC = 0
CG = 0
CT = 0
GA = 2/5
GC = 0
GG = 0
GT = 0
TA = 0
TC = 0
TG = 0
TT = 0
```

4. Understanding your Data

The success of a machine learning experiment is fundamentally contingent upon a comprehensive understanding of the data.

4.1 Class Distribution

For the upcoming experiments, assess the distribution of instances within each class, distinguishing between positive and negative examples. Determine whether the datasets for these experiments are balanced or exhibit class imbalance.

4.2 Visualization

Our dataset is characterized by a high dimensionality that poses challenges for straightforward visualization. Specifically, for the case where $k = 4$, each instance is represented as a 256-dimensional vector. To address the visualization challenge, t-Distributed Stochastic Neighbor Embedding (t-SNE) is frequently employed to project high-dimensional data into a lower-dimensional space, typically two dimensions, that can be readily visualized. For each experimental condition, you are asked to:

1. Generate a graph representing the positive examples.
2. Generate a graph representing the negative examples.
3. Generate a composite graph that includes both positive and negative examples.

It is advisable to investigate the influence of various parameters. For instance, consider altering the value of k between 4 and 6. Additionally, parameters such as 'perplexity' and 'early_exaggeration' are critical and warrant exploration due to their potential impact on the results.

Regarding expected outcomes, an optimal scenario would be the emergence of two distinct clusters from your analysis (positive and negative examples). Such a clear separation would imply that the classification problem is likely to be straightforward. Write down your observations.

5. Data Partitioning

Divide your dataset into distinct training and testing subsets, allocating 20% of the data specifically for testing purposes.

6. Training and Testing

For each experimental condition:

- Train a logistic regression.
- Measure the performance of your model on the test set. Use the method `classification_report` to show the precision, recall, and f1-score.
- Show the resulting confusion matrix.

Write down your observations.

Evaluation Criteria

The report, which is your Jupyter Notebook, should comprehensively document the entire process followed during this assignment. The Jupyter Notebook must include the following:

- Your name, student number, and a report title.
- A section for each step, containing the relevant Python code and explanations or results.
  - For sections requiring Python code, include the code in a cell.
  - For sections requiring explanations or results, include these in a separate cell or in combination with code cells.
- Ensure logical separation of code into different cells. For example, the definition of a function should be in one cell and its execution in another. Avoid placing too much code in a single cell to maintain clarity and readability.
- The notebook you submit must include the results of the execution, complete with graphics, ensuring that the teaching assistant can grade the notebook without needing to execute the code.

Resources

If you do use AI assistance, thoroughly document your interactions. Include the tools and their versions in your report, along with a transcript of all interactions. Most AI assistants keep a record of your conversations. The recommended practice is to create a new conversation specifically for the presentation and consistently reuse this conversation throughout your work on the presentation. Ensure that this conversation is solely dedicated to the presentation. Submit this conversation transcript in the reference section of your summary.

Questions

- You may ask your questions in the Assignment topic of the discussion forum on Microsoft Teams.
- Alternatively, you can email the teaching assistant. However, using the forum is strongly preferred, as it allows your fellow students to benefit from the questions and the corresponding answers provided by the teaching assistants.

References

Umarov, Ramzan Kh, and Victor V Solovyev. 2017. "Recognition of Prokaryotic and Eukaryotic Promoters Using Convolutional Deep Learning Neural Networks." Edited by Igor B Rogozin. *PLoS ONE* 12 (2): e0171410. https://doi.org/10.1371/journal.pone.0171410.