Lab 3.1 - FMRI
Stat 214, Spring 2025

These instructions are specific to **Lab 3.1** and cover the background and deliverables needed to complete the assignment. Note that Lab 3 consists of three parts. Please also see the general lab instructions in **discussion/week1/lab-instructions.pdf** for information on how to set up your environment, write the lab report, and submit the final product.

Contents

1 Submission **1**
Special coding instructions 1
2 Academic honesty and teamwork **2**
Academic honesty statement 2
Collaboration policy 2
LLM usage policy 2
3 Lab 3 Overview **2**
4 Lab 3.1 Instructions **3**
Data 3
Part 2: Modeling & Evaluation (50% of grade) 3
5 Peer Grading **4**

1 Submission

Push a folder called `lab3` to your `stat-214 GitHub` repository by **23:59 on Monday, April 7th**. Only submit **one** report per group. We will run a script that will pull from each of your GitHub repositories promptly at midnight so take care not to be late as late labs will not be accepted.

The 12-page limit is not in place for this lab. The page limit for this lab is 20 pages. The bibliography and academic honesty sections don't count towards this limit.

Follow the general lab instructions in stat-214-gsi/discussion/week1/lab-instructions.pdf for more details. Please do not try to make your lab fit the requirements at the last minute! In your `lab3` folder, please provide everything (except the data) that is needed for someone else to be able to compile your report and receive the exact same pdf. Please follow the provided template for your report.

Special coding instructions

For any ridge regression model you train, please save the trained model in your `results` directory. It should be a `.pkl` file.

2 Academic honesty and teamwork

Academic honesty statement

We ask you to draft a personal academic integrity pledge, addressed to Bin, that you will include with all of your assignments throughout the semester. This should be a short statement, in your own words, that the work in this report is your own and that all sources you used are properly cited, including your classmates. Please answer the following question: Why is academic research honesty necessary? If you feel it is not, make a clear argument why not.

Collaboration policy

Within-group: In a file named `collaboration.txt` in your `report` directory, you must include a statement detailing the contributions of each student, which indicates who worked on what. After the labs are submitted, we will also ask you to privately review your group members' contributions.

With other people: You are welcome to discuss **ideas** with the course staff or other students, but your report must be written up and completed by your group alone. Do not share code or copy/paste any part of the writeup. If you discuss with other students, you must acknowledge these students in your lab report.

LLM usage policy

You are allowed to use LLMs (ChatGPT, GitHub Copilot, etc.) to *assist* in this lab, but are not allowed to use it for more than creating visualizations or helping correct grammar in the report. If we have reason to believe an LLM wrote a significant portion of your code (more than 5%) without your editing or iteration, or any section of your report word-for-word, this will constitute an honor code violation.

3 Lab 3 Overview

Goal Lab 3 focuses on predicting blood-oxygen level across brain voxels [1] via an FMRI as subjects listen to various podcasts. Specifically, the goal is to create *embeddings* from the text of the podcasts to predict voxels. Lab 3 consists of three parts as follows:

1. Using NLP techniques to extract text embeddings, and training a linear model to predict voxels.
2. Pre-training your own languge model to generate embeddings for prediction.
3. Utilizing and interpreting pre-trained LLMs for prediction.

This document contains instructions for part 1.

Scientific Motivation A key aspect of intelligence is the ability of our brains to process rich, complicated language. Accurately predicting how the brain responds to textual stimuli implies that we have a model that could be used to understand how the brain processes language. Further, embeddings from this model can be used to understand representations in the brain. For example, these embeddings can be used to derive insights about how different parts of the brain function to process language. As a result scientists, have dedicated significant effort to measuring brain function via FMRIs. This lab focuses on experiments conduced by Alexander Huth and his lab at UT Austin [1].

[1] A "voxel" in the brain refers to a three-dimensional unit of volume.

4 Lab 3.1 Instructions

Data

We have data from two subjects listening to stories measured by the Huth Lab at UT Austin.The data for each subject consists of whole-brain blood-oxygen dependent (BOLD) signals measured at various points across the podcast. That is, for each subject-story pair, we have measurements $Y \in R^{T' \times V}$, where T' represents the number of FMRI measurements, and V is the number of voxels.

For each subject-story pair, their data can be found on dropbox: `https://www.dropbox.com/scl/fo/0ygk6tzohn4h16k0mprt2/AAxyZSQkZAZE9l96-9Uo6aw?rlkey=m17ot0ut8a1oq2ip8e8lfuept&st=3np5ip0i&dl=0` . Note that these files are really big, ≈ 20GB each. We also have the raw text for each story on the link above which we will use to generate our embeddings to predict the matrix Y.

The following are the instructions for the two main parts of the lab. Please carefully document your analysis pipeline, justify the choices you make, and place your work within the domain context. You should have an introduction and conclusion section in your report.

Coding

For this lab, we provide code to help you process some of the FMRI data. You will find a `ridge_utils` folder that is required to process the data. You can think of this as a black-box that you do not have to worry about.

Additionally, under `code/preprocessing.py`, you will find multiple functions that you will use below to process the data.

Part 1: Generating Embeddings (50% of grade)

An embedding refers to a numerical vector representation of text. This step is essential to convert our raw text into something a predictive model can use. Before we try embedding, for each subject, split the stories into a training and test set. That is, you will need to fit and evaluate a model *per* subject.

1. Provided the list of stories, generate embedding vectors via bag-of-words. Notice that the dimensions of the dimensions of the resulting matrix do not match up with the response matrix. We will have to down-sample it (see [1] for details) to match dimensions. Explain what is being done here.
2. Call `downsample_word_vectors` from the file `code/preprocessing.py` to get the dimensions to match. Further, trim the first 5 seconds and last 10 seconds of the output.
3. Create lagged versions of the features using `make_delayed` from `code/preprocessing.py` with delays ranging form $[1, 4]$ inclusive. Explain what this does.
4. Repeat the process above for **2 other pre-trained embedding methods**: Word2Vec and GloVe. You will have to find these pre-trained embedding methods online, and use them. The processed embeddings resulting from these steps will serve as the features (X matrix) in our regression.
5. Describe potential benefits of using pre-trained embeddings.

Part 2: Modeling & Evaluation (50% of grade)

We will now fit a linear model to each of the embeddings you generated in the previous step.

- Fit a ridge regression model, and report the mean correlation coefficient (CC) for different embeddings
- Devise a scheme to cross validate the different models and select the best performing linear regressor.

Detail your procedure and report the mean test CC, median test CC, top 1 percentile CC and top 5 percentile CC.

- For the best embedding, perform a more detailed evaluation by examining the distribution of the CC across voxels. Plot the distribution. What do you notice?
- Does this model perform well across all voxels? What does this imply scientifically? What is a reasonable interpretation criterion for interpreting voxels according to PCS?
- Perform a stability analysis, e.g., examine performance across various test stories, or across subjects.

5 Peer Grading

So that you each have the opportunity to see alternative approaches to the labs, we will be doing peer-grading for this class.

You will each receive 2 reports from your peers to grade. A detailed rubric will be provided and you will be expected to provide both written feedback as well as a numeric grade on a variety of topics including communication, quality of data cleaning, relevance of visualizations, and reproducibility (can you easily re-compile their report).

After you have all submitted your own assignments (and shortly after the deadline), we will run a script that will automatically push two randomly selected reports into a folder called `lab3/peer_review/`. To retrieve your allocated reports, you will need to `git pull`. You will have one week to review these two reports and return your feedback in the form of a Google questionnaire that we will send by email to you. We will use these two grades for your report as a guide for grading, rather than as a final decision on your grade.

References

[1] Shailee Jain and Alexander Huth. "Incorporating Context into Language Encoding Models for fMRI". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: `https://proceedings.neurips.cc/paper_files/paper/2018/file/f471223d1a1614b58a7dc45c9d01df19-Paper.pdf`.