CSE 518A: Assignment 3

Due: Midnight, March 21 (Thursday), 2019

Notes:

- Please submit your assignments using Gradescope.
- The assignment is due **by 11:59 PM on the due date.** Remember that you may not use more than 2 late days on any one homework, and you only have a budget of 3 in total.
- You can use any programming language you like. You only need to submit a report of your results and do not need to submit your codes. However, you should keep a copy of your codes until the end of the semester. If needed, I might request to examine the codes.
- Please keep in mind the collaboration policy as specified in the course syllabus. You can (and are encouraged to) discuss with other students, however, you **must write down the solutions on your own**. You must also write, in the beginning of the submission, the names of students you discuss the questions with and any external sources you used in a significant manner in solving the problem.

Assignment Description:

This is a programming assignment. The goal is to implement label aggregation algorithms on the provided dataset.

- Download the dataset at the following website:
  `https://sites.google.com/site/nlpannotations`.
  You will be implementing label aggregation algorithms on the Recognizing Textual Entailment (RTE) dataset (rte.standardized.tsv).
- In the RTE dataset, for each crowdsourced question, the worker is presented with two sentences and is asked to check if the second hypothesis sentence can be inferred from the first. This dataset contains 800 sentence pairs and 164 workers. Each sentence pair has 10 annotations.
- Description on the fields of the file:
  - *!amt_worker_ids*: The IDs of workers.
  - *orig_id*: The IDs of sentence pairs.
  - *response*: The worker's answer (annotation) to the sentence pair (0 or 1).
  - *gold*: Ground truth of the sentence pair.
- Homework requirements:

1. Implement majority voting:
   For each sentence pair, randomly draw k annotations. Use majority voting for aggregation. Calculate the average aggregation error (the ratio of wrong predictions) across all 800 sentence pairs. Repeat this for $k = 1, ..., 10$. Draw a plot showing how increasing the number of workers per sentence pair increases the aggregation accuracy (for example, you can have a plot with x-axis being the number of workers per sentence pair, and y-axis being the average aggregation error).
2. Implement another aggregation method:
   Pick any algorithm from the papers in our label aggregation lectures (in the required or optional readings). Repeat the above process by replacing majority voting with the algorithm you pick. Generate the same plot.
3. Write a report:
   Showing the plot(s). It might make sense to put the above results in the same plot for easier comparison. (optional: include any additional plots you find interesting). Briefly describe the aggregation algorithm you pick. Give a concise explanation on why you think the algorithm outperforms (or underperforms) majority voting.