CSE 518A: Assignment 2

New Due: Midnight, October 9 (Wednesday), 2019

Notes:

- Please submit your assignments using Gradescope. There will be two separate submission links for reports and codes.
- 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 4 in total.
- You can use any programming language you like. **The grades will be based on your reports.** By default, we won't look at your codes. Your codes will only be checked when we have confusions/doubts about the reported results. Your codes will also be used to run plagiarism tests.
- 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.

- Instructions:
  You goal is to implement three label aggregation algorithms, majority voting, EM (the simplest version as we discussed in class), and SVD, on the provided dataset. In this dataset, there are 10 annotations/labels per task. You will be asked to sub-sample the datasets and create scenarios when each task has $k = 1, ..., 10$ labels. You will then examine how the aggregation algorithms perform when k increases.

  1. For each task (sentence pair), randomly draw $k = 1, 2, ..., 10$ annotations.
  2. Implement the following three aggregation algorithms: majority voting, EM, and SVD and run them on datasets with $k = 1, ..., 10$ annotations per task
  3. Calculate the aggregation error (the ratio of wrong predictions) across all 800 sentence pairs for $k = 1, ..., 10$. (If the votes are equal, you can break ties any way you like, but please be consistent.)
  4. Repeat the above process multiple times and calculate the average error.
  5. Draw a plot showing how increasing the number of workers per sentence pair decreases the aggregation error (for example, you can have a plot with x-axis being the value of k and y-axis being the average aggregation error). You should have one curve for the performance of each algorithm.
  6. Provide a brief discussion on what you have observed.
  7. (Optional) The performance of SVD is not ideal. You can also try to perform "extrapolation" and generate datasets with $k > 10$. (You know the true label, and you know the empirical worker skill. So you can *generate* additional labels assuming workers are behaving in a certain way). You can then see how many labels are needed to SVD to perform well.

- **Additional Note:** In this dataset, the labels are either 1 or 0. However, in the algorithms we introduced in class, we assume the labels are either $+1$ or -1. You might want to do the label conversions to make them consistent.