Batch Acquisition for Deep Bayesian Active Learning with Imperfect Oracles

Pushkar Kolhe
Introduction

- Active Learning is a promising, but rarely used due to practical challenges
- Labeling is imperfect/noisy
  - Some instances are difficult to label
  - Quality can change over time

- Related Work
  - End to end frameworks: Modify model or loss function [1, 2]
  - Standard frameworks: Learn, then label

Repeated Labeling works [1]

Number of human responses to be considered correct

- 67.87% of the words required two transcriptions
- 17.86% required three
- 7.10% required four
- 3.11% required five
- 4.06% required six [2]

Bayesian Neural Network

Algorithm: Input \((D_{\text{train}}, D_{\text{pool}}, D_{\text{test}})\)

1. Learn a model on \(D_{\text{train}}\)
2. Run a MC dropout pass on \(D_{\text{pool}}\)
3. Find batch using BatchBALD acquisition function

\[
\{x_1^*, \ldots, x_b^*\} = \arg \max_{\{x_1^*, \ldots, x_b^*\} \in D_{\text{pool}}} \sum_{i=1}^{b} I(y_i; \omega | x_i, D_{\text{train}})
\]

4. Generate a candidate query \(D_{\text{batch}}\) with
   a. Control Queries
   b. High Uncertainty Queries \(\{x_1^*, \ldots, x_b^*\}\)
5. Update label uncertainty while gathering labels from multiple labelers (see next slide)
6. Transfer \(D_{\text{batch}}\) from \(D_{\text{pool}}\) to \(D_{\text{train}}\)
7. Repeat

---

Repeated labeling on a candidate batch

**Batch 1**

Control Queries: Use to model proficiency of labeler

\[
p = H(y_i, \hat{y}_c) = \sum_{i=1}^{c} y_i \log \frac{1}{\hat{y}_c}
\]

High Uncertainty Queries: Improve uncertainty based on labels

\[
P(C|D) = \frac{P(D|C)P(C)}{\sum_{C' \in C} P(D|C')P(C')}
\]

- **P(C):** Current Uncertainty or prior on labels
- **P(D|C):** Proficiency of labeler
- **P(C|D):** Updated Uncertainty after the labeler labels this batch
Results

- X axis shows the number of labelers used for a batch
- Y axis shows uncertainty in the labels
- As we increase the batch size, fewer labelers are needed to gather labels which are considered correct
- Loss also decreases as we gather labels, this shows that the labels are accurate
Conclusion

- BatchBALD can be extended easily to use labels from imperfect oracles
- A candidate batch with control and high uncertainty query points can be used to model proficiency of the labelers and gather labels with confidence

- Future Work
  - Find the best control queries for a given batch
  - Experiment with different models of proficiency of labeler

- Applications
  - Peer Review in Online Classrooms