Motivations

- Problems with ever-increasing data set:
  - More data comes at the price of more compute.
  - Increased effort of labeling the data.
  - Performance of new data diminishes when data set grows.

Goal:
Select the most informative and representative subset of a large data set.

Data Subset Selection Scenarios:
- Supervised data subset selection (with access to the transcriptions)
- Unsupervised data subset selection (without access to the transcriptions).

Applications:
- Tune parameters on a small and representative subset.
- Identify a subset of data to transcribe for bootstrapping purposes.

Submodular Framework for Speech Data Subset Selection

Submodular Functions:
- A special class of set functions that have diminishing returns property.
- A set function \( f: 2^V \rightarrow \mathbb{R} \) is submodular, if
  \[ f(k \cup S) - f(S) \geq f(k \cup R) - f(R), \]
  \[ \forall R \subseteq S \subseteq V \text{ and } \forall k \in V \setminus S. \]

Examples of Utility (Submodular) Functions:

- Facility location function:
\[
 f_{\text{fac}}(S) = \sum_{i,j} \max_{k \in B} w_{ij},
\]
  where \( w_{ij} \) is the similarity measure between speech utterances \( i \) and \( j \).

- Diversity reward function:
\[
 f_{\text{div}}(S) = \sum_{k=1}^{K} \sum_{i,j \in P_k} \frac{w_{ij}}{V},
\]
  where \( P_1, \ldots, P_K \) is a partition of \( V \) into \( K \) disjoint blocks.

- Feature-based Submodular Function:
\[
 f_{\text{fea}}(S) = \sum_{u \in U} g(m_u(S)).
\]
  - \( g() \) is a concave function.
  - \( U \): a set of features (e.g., phones, words, triphones, triphone states).
  - \( m_u(S) = \sum_{v \in S} m_u(v) \): relevance score of the feature \( u \in U \) within the set \( S \).

- Two-layer Feature-based Submodular Functions:
\[
 f_{\text{two}}(S) = \sum_{u_1 \in U} g_1(\sum_{u_2 \in U} W(u_2, u_1)g_2(m_u(S))).
\]
- \( g_1() \) and \( g_2() \) are concave functions.
- \( W(u_2, u_1) \) measures the interaction between the feature \( u_1 \) and the feature \( u_2 \).

Greedy Algorithm for Problem 1

Algorithm 1 Greedy algorithm for knapsack constrained submodular max [1]

1. Input: a monotone submodular function \( f \), budget constraint \( B \), and a list of costs \( \{c(v_1), \ldots, c(v_n)\} \).
2. Initialization \( S \leftarrow \emptyset \).
3. repeat
   4. Pick an element \( v^* \in \arg \max_{v \in V \setminus S} \frac{f(v \cup S) - f(S)}{c(v)} \)
   5. Update \( S \leftarrow S \cup v^* \)
   6. until Reaching the budget, i.e., \( c(S) > B \)

- Approximately solves Problem 1 with the worst-case guarantee \( \frac{1}{2}(1 - 1/e) \).
- Empirically is often close to optimum.
- Can be sped up to almost linear-time complexity, thanks again to submodularity.

Data and Systems

Task: Speech data subset selection for phone recognition.

- TIMIT Corpus
- Subset Selection
- Train HMM
- Evaluation

Set-up:
- Data set: TIMIT training, development, and core test set.
- Acoustic model: a 3-state GMM-HMM monophone recognizer for each phone class.

Tokenization:
- Unsupervised: produced by an HMM trained in an unsupervised way.
- Supervised: produced by simple monophone recognizer trained on the TIMIT transcriptions.

Submodular selection:
- Choose a utility (submodular) function.
- Instantiate it with supervised/unsupervised tokenization.
- Solve Problem 1 with the Greedy algorithm.

Baseline selection methods:
- Random baseline: Randomly draw specified amount of the training data.
- Histogram-entropy baseline [2]: Choose the subset with a maximum-entropy distribution over phones.

Empirical Results

This work is based on research sponsored by Intelligence Advanced Research Projects Activity (IARPA) under agreement number FA8650-12-2-7263

References