Using Document Summarization Techniques for Speech Data Subset Selection

Kai Wei  Yuzong Liu  Katrin Kirchhoff  Jeff Bilmes
Electrical Engineering Dept.
University of Washington, Seattle

November 3, 2013
Outline

1. Introduction to speech data subset selection
2. Background on Submodularity
3. Experimental results
Outline

1. Introduction to speech data subset selection
2. Background on Submodularity
3. Experimental results
What is speech data subset selection

- Given a large set of unlabeled speech utterances.
- Procedure of speech data subset selection (batch active learning):
  - Select a subset of the training utterances
  - Transcribe only the selected utterances
  - Train a speech recognition model on the labeled utterances
  - Evaluate the performance of the trained model on a fixed test set of speech utterances
- Good data subset selection methods should:
  - Select and label a relatively small subset of training data
  - Trained model based on the subset of data works as well as the model trained on the entire set of utterances.
Motivations

- Unlabeled speech data is abundant but labels (e.g., transcriptions) are costly, time-consuming and error-prone.
- Training speech utterances can be fairly redundant:

  1. all_right **how are_you** doing
  2. **how are_you** with yours
  3. hi nadine my name is lorraine **how are_you**
  4. good **how are_you**
  5. hello hi **how are_you**
  6. good thanks **how are_you**
  7. uh **how are_you**
  8. i'm good **how are_you**
  9. fine **how are_you**

  Transcription of some utterances from Fisher Corpus

- Larger amount of training data, longer experimental turnaround time, and higher demands for compute resources.
“Diminishing return” of performance gain

- Performance gain curves of large-scale ASR systems with respect to the amount of training data often show “diminishing returns”.

![Graph showing performance gain curves](image-url)

[Wei et al., 2013]

[Riccardi, et. al. 2005]
Outline

1. Introduction to speech data subset selection
2. Background on Submodularity
3. Experimental results
Set functions $f : 2^V \rightarrow \mathbb{R}$

- We are given $V$, a finite set of objects.
- A set function $f : 2^V \rightarrow \mathbb{R}$ is a function that produces a value for any subset $A \subseteq V$. 
Set functions $f : 2^V \rightarrow \mathbb{R}$

For example, given $A \subseteq V$, we might produce value $f(A) = 3.3497$. 

- {$A = \{\text{banana}, \text{strawberry}, \text{apple}, \text{book}\}$
Set functions $f : 2^V \rightarrow \mathbb{R}$

For example, given $A \subseteq V$, we might produce value $f(A) = 3.3497$.

However, since there are $2^{|V|}$ subsets, a function $f : 2^V \rightarrow \mathbb{R}$ has exponential number ($2^{|V|}$) of values.
Submodular functions

- Submodular functions are a restricted class of set functions.
- Diminishing returns: Let $A \subseteq B \subseteq V \setminus \{v\}$ then $f$ is submodular iff
  \[
  f(v|A) \triangleq f(A + v) - f(A) \geq f(B + v) - f(B) \triangleq f(v|B) \tag{1}
  \]
  - i.e., conditioning reduces valuation (like entropy).
- $f$ is monotone if $f(v|A) \geq 0$.
- Example: $f$ gives number of colors of a set $A$ of balls in an urn.
Submodularity in machine learning

- Submodular optimization, widely and successfully applied in many of today’s machine learning applications:
  - Document summarization
  - Sensor placement
  - Feature subset selection
  - etc.

Document Summarization (Lin & Bilmes, 2011)

Sensor Placement (Krause et al, 2008)
Previous problems can be solved using discrete optimization:

$$\max \sum_{i \in S} c_i \leq B \quad f(S)$$

where $c_i$ is “cost” of element $i$, and $B$ total budget.
Constrained Submodular Maximization

- Previous problems can be solved using discrete optimization:

\[
\max_{\sum_{i \in S} c_i \leq B} f(S)
\]

where \( c_i \) is “cost” of element \( i \), and \( B \) total budget.

- In general, NP-hard, can be solved using ILP but not practical/scalable.
Constrained Submodular Maximization

- Previous problems can be solved using discrete optimization:

\[
\max_{\sum_{i \in S} c_i \leq B} f(S)
\]

where \(c_i\) is “cost” of element \(i\), and \(B\) total budget.

- In general, NP-hard, can be solved using ILP but not practical/scalable. Without submodularity, problem is inapproximable (provably exponential even for any kind of approximation guarantee).
Constrained Submodular Maximization

- Previous problems can be solved using discrete optimization:

\[
\max \sum_{i \in S} f(S) \quad \text{subject to } \sum_{i \in S} c_i \leq B
\]

where \( c_i \) is "cost" of element \( i \), and \( B \) total budget.

- In general, NP-hard, can be solved using ILP but not practical/scalable. Without submodularity, problem is inapproximable (provably exponential even for any kind of approximation guarantee).

- If \( f \) submodular, problem can be approximated by a simple greedy algorithm with worst case \((1 - 1/e)\) factor guarantee.
Constrained Submodular Maximization

- Previous problems can be solved using discrete optimization:

\[
\max_{\sum_{i \in S} c_i \leq B} f(S)
\]

(2)

where \( c_i \) is “cost” of element \( i \), and \( B \) total budget.

- In general, NP-hard, can be solved using ILP but not practical/scalable. Without submodularity, problem is inapproximable (provably exponential even for any kind of approximation guarantee).

- If \( f \) submodular, problem can be approximated by a simple greedy algorithm with worst case \((1 - 1/e)\) factor guarantee.

- Approximation guarantee is typically even much better than \((1 - 1/e)\) depending on how “curved” the submodular function is.
Constrained Submodular Maximization

- Previous problems can be solved using discrete optimization:

\[
\max_{\sum_{i \in S} c_i \leq B} f(S)
\]

where \( c_i \) is “cost” of element \( i \), and \( B \) total budget.

- In general, NP-hard, can be solved using ILP but not practical/scalable. Without submodularity, problem is inapproximable (provably exponential even for any kind of approximation guarantee).

- If \( f \) submodular, problem can be approximated by a simple greedy algorithm with worst case \((1 - 1/e)\) factor guarantee.

- Approximation guarantee is typically even much better than \((1 - 1/e)\) depending on how “curved” the submodular function is.

- Thanks again to submodularity, an accelerated greedy variant can do just as well, and runs in \( O(n \log n) \) complexity, where \( n = |V| \).
Submodular objective functions applied on synthetic data

**Facility location function**

\[ f_{\text{fac}}(S) = \sum_{i \in V} \max_{j \in S} w_{i,j} \]

**Saturate coverage function**

\[ f_{\text{sat}}(S) = \sum_{i \in V} \min\{C_i(S), \beta C_i(V)\} \]

where \( C_i(S) = \sum_{j \in S} w_{i,j} \)

---

Wei et al, 2013

Using Document Summarization Techniques for Speech Data Subset Selection
Facility location function

\[ f_{\text{fac}}(S) = \sum_{i \in V} \max_{j \in S} w_{i,j} \]

Saturate coverage function

\[ f_{\text{sat}}(S) = \sum_{i \in V} \min \{ C_i(S), \beta C_i(V) \} \]

\[ \text{where } C_i(S) = \sum_{j \in S} w_{i,j} \]
Outline

1. Introduction to speech data subset selection
2. Background on Submodularity
3. Experimental results
Data and systems

- Our approach is evaluated on subselecting data from TIMIT corpus for training a phone recognizer.
- A 3-state HMM monophone recognizer for all 48 TIMIT phone classes is trained given each selected subset.
- Performance of trained HMM monophone recognizer is evaluated on core test set of 192 utterances.
- Proof of concept task to test different combination of submodular functions and data subset selection methods.
Experimental results: comparison among different subset selection methods

Random selection: randomly select subsets and evaluate performance based on the randomly selected subsets.

Histogram-entropy baseline [Wu, Zhang, Rudnicky, 2007]: choose subset of utterances such that entropy of distribution of phones is maximized.
Experimental results: comparison between different submodular functions

![Graph showing comparison between saturate coverage function and facility location function.]

- **Saturate coverage function**
- **Facility location function**

Wei et al., 2013
Conclusions

1. Submodular framework greatly outperforms baseline methods (random baseline, histogram-entropy based method).
2. Submodular framework naturally fits the speech data subset selection problem.
3. Moreover, our submodular framework has theoretical guarantee in terms of optimization, and can have very fast implementation.