A SUBMODULAR OPTIMIZATION APPROACH TO SENTENCE SET SELECTION

Yusuke Shinohara

Corporate Research and Development Center, Toshiba Corporation
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan
yusuke.shinohara@toshiba.co.jp

ABSTRACT

A new method for selecting a sentence set with a desired phoneme distribution is presented. Selection of a sentence set for speech corpus recording is a fundamental step in speech processing research. The problem of designing phonetically-balanced sentence sets has been studied extensively in the past. One of the popular approaches is to select a sentence set so that its phoneme distribution gets close to a given (desired) distribution. Several methods have been proposed in the literature to realize this approach. However, these methods were designed by heuristics, which means they are not optimal. In this paper, we propose a near-optimal method for selecting sentence sets along this approach. We first define our objective function, and show it to be a submodular function. Then, we show that a greedy algorithm is near-optimal for this problem, according to the submodular optimization theory. We also show that a significant speedup is possible by exploiting the submodularity of the objective function. Our experimental result on Japanese phonetically-balanced sentence set selection shows the effectiveness of the proposed method.

Index Terms— Corpus design, phoneme distribution, Kullback-Leibler divergence, submodular optimization, speech recognition and synthesis

1. INTRODUCTION

Designing a phonetically-balanced sentence set for recording a speech corpus is a fundamental step in developing speech processing systems. For instance, in order to record a speech corpus for building a speech recognition (or synthesis) system, a sentence set to be read by the narrators should be designed firstly. Phonetically-balanced sentence sets are preferred for building statistical models of speech, while at the same time smaller sets are preferred for reducing the recording cost.

Various techniques for making a phonetically-balanced sentence set have been proposed in the literature. The problem is usually formulated as selecting the *best* subset from a given sentence set under a given budget (e.g. a given number of sentences). One of the oldest examples is the entropy maximization method [1]; the phonetic-balancedness of a sentence set is measured by the entropy (i.e. uniformness) of its phoneme distribution; after randomly selecting an initial set of sentences from a given pool of sentences, the entropy is maximized by iteratively swapping randomly selected sentence pairs (one from the sentence set and the other from the pool). More recent example is the Kullback-Leibler (KL) divergence minimization approach [2]; given a desired phoneme distribution and a sentence pool, sentences are selected from the pool so that the phoneme distribution of the selected sentences gets closest, in a KL divergence sense, to the given distribution; a simple heuristic algorithm was used to minimize the divergence. A similar approach was also adopted in [3], where mathematically the same problem was considered for selecting sentences with a desired i-vector distribution; that is, sentences were selected so that the the i-vector distribution gets closest to the given distribution; another simple heuristic algorithm, originally developed for language model adaptation by minimizing the KL divergence between n-gram distributions [4], was utilized to minimize the divergence.

These algorithms are not necessarily optimal for minimizing (or maximizing) the respective objective functions. For instance, the swapping method [1] is optimal only if a sufficient number of swappings are carried out; however, to build a sentence set of a practical size, the algorithm should be stopped way before convergence, which makes the algorithm non-optimal. The algorithm used in [3] and [4] takes each sentence in a given sentence pool in a random order, and adopts it (adds the sentence to the set) if the KL divergence gets smaller, or discards it otherwise. The algorithm is simple and fast, but clearly not optimal.

The current paper proposes a provably near-optimal algorithm for building a phonetically-balanced sentence set. We first define the "utility" of a sentence set as the weighted sum of the log-frequencies[1] of the phonemes, where the weights are defined according to the desired phoneme distribution. The utility can be interpreted as saying *there is no data like more and balanced data*. Then, we show that our objective function (the utility) has a special property called *submodularity*. We show that a greedy algorithm is provably near-optimal for solving this problem, according to the submodular optimization theory. We further show that a significant speedup of the greedy algorithm is possible by exploiting the submodularity of the objective function. It is also shown, under some mild conditions, that the sentence set build in this way is equivalent with the one that minimizes the KL divergence to the desired distribution. Finally, a simple experimental result is given to show the validity of the theory.

2. RELATED WORK

For recording speech corpora intended to be used for building speech synthesis systems, algorithms to find the smallest sentence set that covers all of the given phoneme set (e.g. diphones) have been used. This problem is known as the set-cover problem, which can be solved efficiently by a simple greedy algorithm. It is known that the

[1]Throughout this paper, the term "frequency" means the *absolute* frequency (count of an event), not the *relative* frequency (probability of an event).

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

greedy algorithm is near-optimal for solving the set-cover problem. Notable algorithms in this line include [5, 6]. For larger corpora, mainly intended to be used for training acoustic models of speech recognition systems, the set-cover objective is not suitable. Instead, algorithms to find phonetically-balanced sentence set are preferred. Examples in this line include [1, 7, 2, 8, 9]. We focus on the latter case in this paper.

Submodular optimization [10, 11] is attracting much attention these days. For instance, many tutorials are held at major conferences, including ICML2008 and ICML2013. The technique can be applied to wide variety of problems, for instance outbreak detection in water networks and the web [12]. Submodularity in discrete optimization is similar to *convexity* in continuous optimization, and is the key to use powerful optimization tools. Although the technique is getting popular among machine learning researchers, it is not widely known in the spoken language processing community yet, except for a series of papers by Lin and Bilmes [13, 14]. To the best of our knowledge, the submodular optimization techniques have never been applied to the phonetically-balanced sentence set selection problem in the literature.

3. METHODOLOGY

3.1. Problem statement

Given a sentence set U and a budget B, we want to select a subset S of U so that the sentence set S has maximum utility, while the sentence set S should not be too big to exceed the given budget. Let $J(S)$ be a set function to evaluate the utility of a sentence set S, we formulate the sentence set selection problem as follows,

$$S^* = \arg\max_{S \subseteq U} J(S) \;\; \text{subject to} \;\; \sum_{s \in S} c(s) \leq B, \tag{1}$$

where $c(s)$ is the cost of sentence s. For instance if we set unit cost for all the sentences, $c(s) = 1$, then the constraint means that at most B sentences can be selected. Namely, in the unit-cost case, the problem reduces to

$$S^* = \arg\max_{S \subseteq U} J(S) \;\; \text{subject to} \;\; |S| \leq B. \tag{2}$$

On the other hand, distinct costs can be set to different sentences. For instance, setting higher costs to longer sentences is reasonable.

3.2. Objective function

Let P be the set of phonemes. For instance, if context-independent phonemes (monophones) are used, P consists of some 50 phonemes. Alternatively, if triphones are used, P consists of some 5,000 (context-dependent) phonemes. Let $\pi = (\pi_1, \ldots, \pi_{|P|})$ be the desired distribution of phonemes, where π_i represents the probability of the i-th phoneme, and $f_i(S)$ be the frequency of the i-th phoneme in S.

Under these notations, we define the utility of a sentence set S as follows,

$$J(S) \stackrel{\text{def}}{=} \sum_{i=1}^{|\mathcal{P}|} \pi_i \log f_i(S). \tag{3}$$

The design concept for this utility is as follows. First, we want the utility to be a linear combination of the log-frequencies of phonemes. This idea is based on an empirical knowledge that the utility of data is proportional to logarithm of its amount. The weights for the linear combination is set as the desired probability of each phoneme. This is intended to represent the expected utility under the desired phoneme distribution π.

3.3. Interpretation of the objective function

Utility function $J(S)$ can be interpreted as saying that *there is no data like more and balanced data*. In the following, we describe this interpretation in detail. First, let us define the balancedness of the sentence set as the KL divergence from the desired distribution, that is,

$$D_{\mathrm{KL}}(\pi \parallel p(S)) \stackrel{\text{def}}{=} \sum_{i=1}^{|\mathcal{P}|} \pi_i \log \frac{\pi_i}{p_i(S)} \tag{4}$$

$$= -\sum_{i=1}^{|\mathcal{P}|} \pi_i \log p_i(S) + \text{Const.} \tag{5}$$

where we defined $p_i(S)$ as the probability of the i-th phoneme in S,

$$p_i(S) = \frac{f_i(S)}{f_T(S)}. \tag{6}$$

Here, $f_T(S)$ is the total frequency, defined by $\sum_i f_i(S)$. Then, we have

$$D_{\mathrm{KL}}(\pi \parallel p(S)) = -\sum_i \pi_i \log \frac{f_i(S)}{f_T(S)} + \text{Const} \tag{7}$$

$$= -\sum_i \pi_i \log f_i(S) + \log f_{\mathrm{T}}(S) + \text{Const.} \tag{8}$$

By re-arranging the terms, we get

$$J(S) = \log f_{\mathrm{T}}(S) - D_{\mathrm{KL}}(\pi \parallel p(S)) + \text{Const.} \tag{9}$$

The meanings of the respective terms are as follows:

- $J(S)$: Utility of the data
- $\log f_{\mathrm{T}}(S)$: Amount of the data (in log-scale)
- $D_{\mathrm{KL}}(\pi \parallel p(S))$: Imbalancedness of the data

So the equation is saying that the utility of a data set is the sum of its amount and balancedness. In other words, *there is no data like more and balanced data*.

3.4. Submodularity

A set function $J(\cdot)$ is said to be *submodular* if it satisfies the following inequality for all $S \subseteq T \subseteq U$ and $s \in U \backslash T$,

$$J(S \cup \{s\}) - J(S) \geq J(T \cup \{s\}) - J(T). \tag{10}$$

In the context of sentence set selection, this inequality means that adding a sentence s to a smaller set S brings a larger gain (i.e. additional utility) than adding it to a larger set T. In other words, it is representing the law of *diminishing returns*.

Next, we show that the objective function $J(S)$ defined in (3) is submodular. We use the following inequality that holds for $0 < x \leq y$ and $0 \leq d$,

$$\log(x + d) - \log(x) \geq \log(y + d) - \log(y). \tag{11}$$

This implies that the following inequality holds for all i,

$$\log f_i(S \cup \{s\}) - \log f_i(S) \geq \log f_i(T \cup \{s\}) - \log f_i(T). \tag{12}$$

Since π_i is non-negative for all i, we get

$$J(S \cup \{s\}) - J(S) \geq J(T \cup \{s\}) - J(T), \tag{13}$$

which means that our objective function $J(S)$ is submodular.

3.5. Greedy algorithm

3.5.1. Unit-cost case

The problem of searching the best sentence set to maximize the utility is a combinatorial problem, and is known to be NP hard. So we have to resort to some approximate algorithm to find a solution that is not too far from the best possible one. It is known that the greedy algorithm is a near-optimal algorithm for solving the submodular maximization problem under the unit-cost constraint ($c(s) = 1$). In other words, no other polynomial-time algorithm can have a better bound than the greedy algorithm. Formally speaking, the following theorem is known.

Theorem [15, 16] *If $J(\cdot)$ is a nonnegative, monotone, submodular function, the utility of the sentence set S^* found by the greedy algorithm has the following lower bound, $f(S^*) \geq (1 - 1/e) \max_{|S| \leq B} f(S)$.*

Algorithm 1 shows the greedy algorithm for selecting the optimal sentence set. Starting from the empty set, for each iteration, the algorithm selects the best sentence that maximizes the utility, and stops when the given number of sentences are selected.

Algorithm 1 Sentence set selection with a greedy algorithm (unit cost case)

Input: U, B, π
$S \leftarrow \phi$
repeat
  $s^* \leftarrow \arg\max_{s \in U \setminus S} J(S \cup \{s\}) - J(S)$
  $S \leftarrow S \cup \{s^*\}$
until $|S| = B$
Output: S

3.5.2. Non-unit-cost case

The greedy algorithm for the unit-cost case tends to select longer sentences. However, reading longer sentences usually costs more than to read shorter sentences. When the given sentence set U contains sentences of various length, the sentence set found by the algorithm may lead to a high recording cost. So setting different cost to each sentence is preferable in such a situation. A variant of the greedy algorithm proposed in [16] can be used for the non-unit-cost case. A pseudo code is given in Algorithm 2 [2]. The algorithm gives a bound of

$$f(S^*) \geq \frac{1}{2}\left(1 - \frac{1}{e}\right) \max_{S \subseteq U : \sum_{s \in S} c(s) \leq B} f(S). \tag{14}$$

[2]To avoid clutter, we show an algorithm that returns a solution that slightly exceeds the budget. It is straightforward to modify the algorithm to get a solution strictly meeting the budget.

Note that there is an algorithm with a better bound [17], i.e. $1 - 1/e$, but the algorithm is much more complex, and does not scale to large problems.

Algorithm 2 Sentence set selection with a greedy algorithm (non-unit-cost case)

Input: U, B, π

`//Unit-cost`
$S_{uc} \leftarrow \phi$
while $\sum_{s \in S_{uc}} c(s) \leq B$ **do**
  $s^* \leftarrow \arg\max_{s \in U \setminus S_{uc}} J(S_{uc} \cup \{s\}) - J(S_{uc})$
  $S_{uc} \leftarrow S_{uc} \cup \{s^*\}$
end while

`//Cost-benefit`
$S_{cb} \leftarrow \phi$
while $\sum_{s \in S_{cb}} c(s) \leq B$ **do**
  $s^* \leftarrow \arg\max_{s \in U \setminus S_{cb}} \frac{J(S_{cb} \cup \{s\}) - J(S_{cb})}{c(s)}$
  $S_{cb} \leftarrow S_{cb} \cup \{s^*\}$
end while

Output: $\arg\max\{J(S_{uc}), J(S_{cb})\}$

3.6. Speeding-up

The greedy algorithms for selecting sentence set presented in Algorithms 1 and 2 are not very fast. A much faster alternative, called the *lazy greedy algorithm*, proposed in [16], can be used instead. The main idea is that the number of function evaluations can be reduced dramatically by exploiting submodularity. This speeding-up technique can be used to both of the unit-cost and non-unit-cost cases. In [16], a problem of sensor placement was considered; specifically, a strategy for selecting an optimal subset from a given set of possible sensor placement locations was studied. In their experiment, a speed-up by a factor of 700 was realized by using the lazy greedy algorithm, instead of the standard greedy algorithm. The sentence set selection problem considered in the present paper can be solved very quickly in the same manner as well. In our experiments, this speeding-up technique is used with a non-unit-cost formulation.

3.7. Equivalence with KL minimization

We show, under some mild conditions, that the sentence set found by the greedy algorithm presented in the previous section is equivalent with the sentence set that has the minimum KL divergence from the given distribution π.

First, let us study the unit-cost case. The sentence set selected by our algorithm is the one that (approximately) maximizes $J(S)$ among all possible sets of sentences $S \subseteq U$ with the given size $|S| = B$. Suppose that the length of each sentence (i.e. the length of the phoneme sequence representing the sentence) in U is (roughly) constant. This assumption is realistic in many situations; for instance, sentences for corpus recordings are usually arranged to be roughly the same length for the sake of readability. Let the constant length be denoted by L. Then, the constraint $|S| = B$ implies $f_{\mathrm{T}}(S) = BL$. That is, $\log f_{\mathrm{T}}(S)$ is constant for all S under this setting. Therefore, the sentence set S^* that maximizes $J(S)$ is the minimizer of the KL divergence $D_{\mathrm{KL}}(\pi \parallel p(S))$ among all $S \subseteq U$ with the given size B. For the case of non-unit-cost, if we use a constraint $\log f_T(S) = B$, the same argument as above can be made.

4. EXPERIMENTAL RESULT

We have compared our proposed algorithm of sentence selection with a random selection algorithm. As the sentence set U from which sentences are to be selected, we have used a set of 248,530 Japanese sentences internally collected from various sources, such as news papers and novels. The average number of phonemes per sentence was about 21. A total of 4,911 distinct triphones appeared in U were used as the phoneme set P. We have selected a sentence set S with our proposed greedy algorithm at a budget of 400,000 phonemes, that is, we used the length (number of phonemes) of a sentence as the cost $c(s)$. In this experiment, the uniform distribution was used as the target distribution π, because the result can be understood easily (the flatter distribution is better), although arbitrary distributions can be used as the target. On the other hand, for comparison, we have randomly selected sentences at the same budget, and obtained a sentence set consisting of 18,939 sentences. Figure 1 shows the frequencies (in log scale) of the phonemes (triphones) for the two cases. It can be seen from the figure that the sentence set given by the proposed algorithm is *flatter*, that is, closer to the uniform distribution. The sentence set collected with the proposed algorithm has higher frequencies, especially at the rare triphones. This is a preferable property of a sentence set because speech corpora based on such a sentence set are suitable for building acoustic models for speech recognition (or synthesis), which are especially good at recognizing (or synthesizing) those rare triphones. Our C++ implementation of the greedy algorithm selected 18,989 sentences in just 237 seconds.

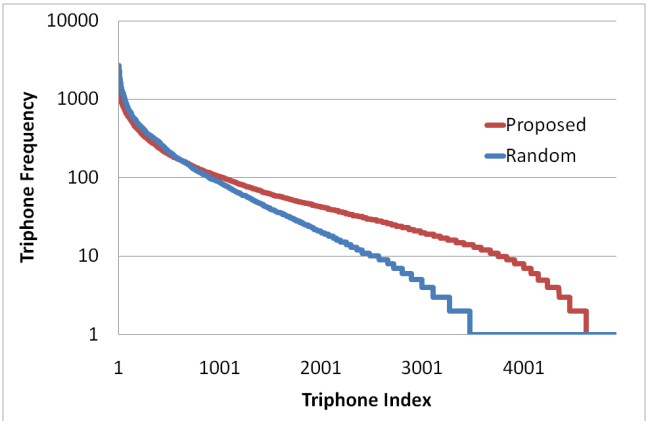

Fig. 1. Comparison of the proposed algorithm and the random selection algorithm.

5. CONCLUSIONS

A new method for selecting phonetically-balanced sentence sets was proposed. The problem of sentence set selection was formulated as a set function maximization problem with a constraint (budget). We first defined our objective function that measures the utility of a sentence set. The objective function can be interpreted as saying that the utility of a sentence set is the sum of its amount and balancedness; in other words, there is no data like more and balanced data. We have then shown that our objective function is submodular, and that the objective function can be maximized efficiently by a greedy algorithm. An experimental result in selecting some 19,000 sentences out of 248,530 have shown the effectiveness of the proposed algorithm.

6. REFERENCES

[1] Ken-ichi Iso and Takao Watanabe, "Design of a Japanese sentence list for a speech database," in *Proceedings of the Acoustical Society of Japan Spring Meeting, 2-2-19*, March 1988.

[2] Xiaodong Cui and Abeer Alwan, "Efficient adaptation text design based on the Kullback-Leibler measure," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2002.

[3] Olivier Siohan and Michiel Bacchiani, "iVector-based acoustic data selection," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2013.

[4] A. Sethy, P. G. Georgiou, B. Ramabhadran, and S. S. Narayanan, "An iterative relative entropy minimization-based data selection approach for n-gram model adaptation," *IEEE Transactions on Audio, Speech and Language Processing*, pp. 13–23, 2009.

[5] Jan P.H. van Santen and Adam L. Buchsbaum, "Methods for optimal text selection," in *Proceedings of the Eurospeech*, 1997.

[6] Helene Francois and Olivier Boeffard, "Design of an optimal continuous speech database for text-to-speech synthesis considered as a set covering problem," in *Proceedings of the Eurospeech*, 2001.

[7] Jia-lin Shen, Hsin-min Wang, Ren-yuan Lyu, and Lin-shan Lee, "Automatic selection of phonetically distributed sentence sets for speaker adaptation with application to large vocabulary mandarin speech recognition," *Computer Speech and Language*, 1999.

[8] Yi Wu, Rong Zhang, and Alexander Rudnicky, "Data selection for speech recognition," in *Proceedings of ASRU*, 2007.

[9] Evandro Gouvea and Marelie H. Davel, "Kullback-Leibler divergence-based ASR training data selection," in *Proceedings of INTERSPEECH*, 2011.

[10] Satoru Fujishige, *Submodular Functions and Optimization*, 2005.

[11] Andreas Krause and Daniel Golovin, *Submodular Function Maximization*, chapter in Tractability: Practical Approaches to Hard Problems, Cambridge University Press, 2012.

[12] Andreas Krause and Carlos Guestrin, "Optimizing sensing: From water to the web," *IEEE Computer*, 2009.

[13] H. Lin and J. Bilmes, "How to select a good training-data subset for transcription: Submodular active selection for sequences," in *Proceedings of the Interspeech*, 2009.

[14] H. Lin and J. Bilmes, "Optimal selection of limited vocabulary speech corpora," in *Proceedings of the Interspeech*, 2011.

[15] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of the approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, 1978.

[16] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the ACM International Conference on KDD*, 2007.

[17] M. Sviridenko, "A note on maximizing a submodular set function subject to knapsack constraint," *Operations Research Letters*, vol. 32, pp. 41–43, 2004.