DISTRIBUTED SUBMODULAR MAXIMIZATION FOR LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

Jun Qi[1*], *Xu Liu*[2*], *Shunshuke Kamijo*[2], *Javier Tejedor*[3]

1. Electrical Engineering, University of Washington, Seattle, USA
2. Institute of Industrial Science, The University of Tokyo, Japan
3. Escuela Politecnica Superior, Universidad San Pablo-CEU, CEU Universities, Madrid, Spain

ABSTRACT

Huge training datasets for automatic speech recognition (ASR) typically contain redundant information so that a subset of data is generally enough to obtain similar ASR performance to that obtained when the entire dataset is employed for training. Although the centralized submodular-based data selection methods have been successfully applied to obtain a representable subset involving the most significant information of the whole dataset, the submodular data selection conveys problems in adapting to an extremely massive dataset.

This paper proposes to use distributed submodular maximization (DSM) for efficiently selecting a data subset that maintains the ASR performance, while reducing tremendously the computational overhead. There are two approaches for the distributed submodular maximization problem: one is based on an homogeneous submodular function, and the other relies on decomposable submodular functions in which heterogeneous submodular functions are applied. Our experiments show that the data subset output by the DSM algorithms can maintain the ASR performance, while significantly reducing the computational overhead. [1]

Index Terms— distributed submodular maximization, automatic speech recognition, decomposable submodular function, greedy algorithm

1. INTRODUCTION

A previous work on submodular data selection for automatic speech recognition (ASR) [1] showed that speech training data are always redundant and a submodular subset of data is sufficient to obtain an ASR result close to that obtained when all the training data are fed in the system.

The submodular data selection bases on the diminishing return property of submodularity, which suggests that the gain obtained from the additional data tends to become marginal as long as an informative data subset is selected [2]. The submodular diminishing return property can be inductively obtained via the definition of a submodular function. Specifically, a function f is a submodular function if and only if for any two subsets A, B and a ground set V, there are $A, B \subseteq V$ and an element $k \notin V$, so that equation (1) is satisfied.

$$f(A \cup \{k\}) - f(A) \geq f(B \cup \{k\}) - f(B). \quad (1)$$

Although the selection of a data subset is an 'NP-hard' discrete optimization problem, an approximated solution with a constant guarantee always exists if the problem can be cast as a monotone submodular maximization problem, as shown in equation (2).

$$\max_{S \subseteq 2^V} f(S),\ s.t., |S| \leq l, \quad (2)$$

where the symbols 2^V, S, and l denote a set of all possible subsets of a ground set V, a subset from elements of 2^V, and a constant for the budget constraint, respectively. Besides, $f(S)$ refers to a normalized monotone submodular function over a subset S, which means that for an empty set Φ, there is $f(\Phi) = 0$, and $f(A) \leq f(B), \forall A \subseteq B$. For simplicity, the feature-based submodular function shown in equation (3) is employed.

$$f(S) = \sum_{v \in S} g(\sum_{u} m_u(v)), \quad (3)$$

where g is defined as a square root function and $m_u(v)$ refers to the number of frames of the feature u in the utterance v. When applied to ASR, clustered tri-phone states are employed as the features [1].

A simple and efficient greedy algorithm can return an approximated solution to equation (2) with a constant performance guarantee. However, the naive greedy algorithm has problems with the increasing amount of training data because the computation of the marginal submodular function in the greedy algorithm is needed through the remaining training data in each iteration. That is the reason why the distributed submodular maximization (DSM) methods [3, 4] are applied in this work.

In our previous works on submodular data partitioning for distributed speech recognition [5, 6], the partitioned disjoint subsets of data are used for training a distributed speech recognition system composed of 8 deep neural networks (DNNs) from the linguistic knowledge that a tri-phone can be transformed to 8 bi-phones. However, the methods presented in this work focus on designing a distributed system for data

[1] *The first and second authors are equivalently contributed to the paper.

Algorithm 1 The greedy algorithm for submodular maximization with a cardinality constraint

1. Set $S = \Phi$, a ground set V, and a constraint l.
2. While $|S| \leq l$ do:
3. $\quad \hat{e} \leftarrow \arg\max_{e \in V} f_S(e)$.
4. $\quad S \leftarrow S \cup \{\hat{e}\}, V \leftarrow V \backslash \{\hat{e}\}$.
5. End while.
6. Return $\hat{S}$.

selection in which the subset is used for training a centralized ASR system. In this work, a decomposable submodular function composed of 8 heterogeneous submodular functions is also created for the distributed submodular data selection.

The rest of the paper is organized as follows: Section 2 presents the greedy algorithm to solve equation (1). Section 3 introduces the algorithms for the distributed submodular maximization. Experiments are reported in Section 4 and the paper is concluded in Section 5.

2. THE GREEDY ALGORITHM

The problem in equation (1) is in fact a submodular maximization problem with a cardinality constraint. The greedy algorithm ensures an approximated solution with a constant performance guarantee [7]. The greedy algorithm to solve the problem related to equation (1) is shown in Algorithm 1, where $f_S(\cdot)$ is a marginal submodular function as shown in equation (4), and $\hat{S}$ refers to an approximated solution which ensures a constant lower bound to the optimal solution S^*, as shown in equation (5).

$$f_S(e) = f(S \cup \{e\}) - f(S) \tag{4}$$

$$f(\hat{S}) \geq (1 - \frac{1}{e}) f(S^*). \tag{5}$$

However, the greedy algorithm has to compute the marginal submodular function $f_S(e)$ with $(|V| - |S|)$ elements each iteration. When the size of V becomes large, the computational overhead tends to be large consequently. Although an accelerated greedy algorithm can speed up the naive greedy algorithm by using the data structure of a priority queue smartly, it is not able to overcome the data scalability issue.

3. DISTRIBUTED SUBMODULAR DATA SELECTION

Since the naive greedy algorithm for data selection has problems in dealing with a massive dataset, distributed submodular maximization methods are necessary. The framework of DSM is shown in Figure 1 and Figure 2, where the entire dataset is randomly split into M batches $\{V_1, ..., V_M\}$ assigned to M clusters, and the clusters output M subsets of data $\{S_1^{gc}, ..., S_M^{gc}\}$ by applying the naive greedy algorithm. The M subsets are combined together into one cluster B and then a final subset S^{gd} is selected from the union set.

Algorithm 2 The greedy DSM with an homogeneous submodular function f

1. Input: a ground set V, # of partitions M, constraint l.
2. Output: Set S^{gd}.
3. Partition V into M sets $V_1, V_2, ..., V_M$ randomly.
4. Run the naive greedy algorithm on each set V_i to find a solution S_i^{gc}.
5. Find $S_{\max}^{gc} = \arg\max_S \{f(S) : S \in \{S_1^{gc}, ..., S_M^{gc}\}\}$.
6. Merge the resulting sets: $B = \cup_{i=1}^{M} S_i^{gc}$.
7. Run the naive greedy algorithm on B to find a solution $S_B^{gc} \leftarrow \arg\max_{S \subseteq B} f(B)$.
8. Return $S^{gd} = \arg\max_S \{f(S) : S \in \{S_{\max}^{gc}, S_B^{gc}\}\}$.

Next, we formulate two frameworks for the greedy DSM: the first one relies on an homogeneous submodular function used for data selection in all machines; the second one depends on a decomposable submodular function with heterogeneous submodular functions, each of which is responsible for data selection in each machine.

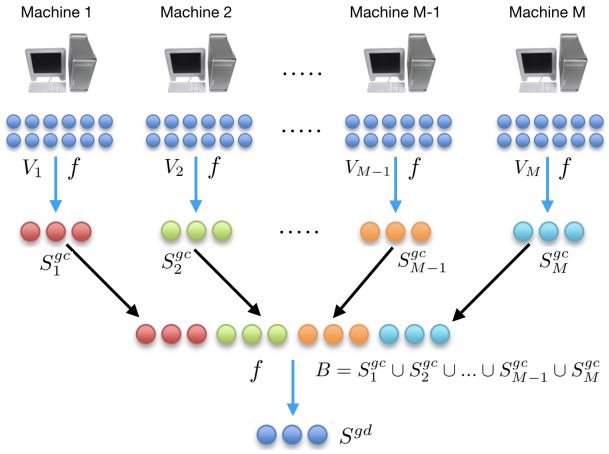

Fig. 1. The greedy DSM with an homogeneous submodular function.

3.1. The greedy DSM with an homogeneous submodular function

The first formulation is based on the greedy DSM with an homogeneous submodular function [3]. Specifically, a feature-based submodular function is used for data selection in all machines. The feature relies on clustered tri-phone states. The related algorithm is shown in Algorithm 2, where the submodular function $f(S)$ is shown as equation (3). Note that in Algorithm 2, $S_{\max}^{gc}$ is the local optimal subset associated with a maximum submodular function value from M machines. The final output S^{gd} should be obtained by comparing the two submodular function values of the two subsets $S_{\max}^{gc}$ and S_B^{gc}. The distributed DSM algorithm ensures that the solution has a constant lower bound to the optimal solu-

Algorithm 3 The greedy DSM with heterogeneous submodular functions $\{f_1, ..., f_8\}$ and $f_U(S) = \sum_{i=1}^{8} f_i(S)$

1. Input: a ground set V, constraint l.
2. Output: Set S^{gd}.
3. Partition V into 8 sets $V_1, V_2, ..., V_8$ randomly.
4. Run the naive greedy algorithm on each set V_i to find a solution S_i^{gc} by $f_i(V_i)$.
5. Find $S_{\max}^{gc} = \arg\max_S\{f(S) : S \in \{S_1^{gc}, ..., S_8^{gc}\}\}$.
6. Merge the resulting sets: $B = \cup_{i=1}^{8} S_i^{gc}$.
7. Run the naive greedy algorithm on B to find a solution $S_B^{gc} \leftarrow \arg\max_{S \subseteq B} f_U(S)$.
8. Return $S^{gd} = \arg\max_S\{f_U(S) : S \in \{S_{\max}^{gc}, S_B^{gc}\}\}$.

tion S^*, as shown in equation (6).

$$f(S^{gd}) \geq \frac{(1 - e^{-1})}{\min(M, l)} f(S^*). \quad (6)$$

3.2. The greedy DSM with heterogeneous submodular functions

The second DSM approach is the greedy distributed submodular maximization with heterogeneous submodular functions [3]. Our previous work [5] showed that a tri-phone corresponds to 8 bi-phones, and hence a submodular function built on clustered tri-phone states corresponds to 8 submodular functions based on clustered bi-phone states. Here, the submodular function is decomposable, as shown in equation (7).

$$f_U(S) = \sum_{i=1}^{8} f_i(S). \quad (7)$$

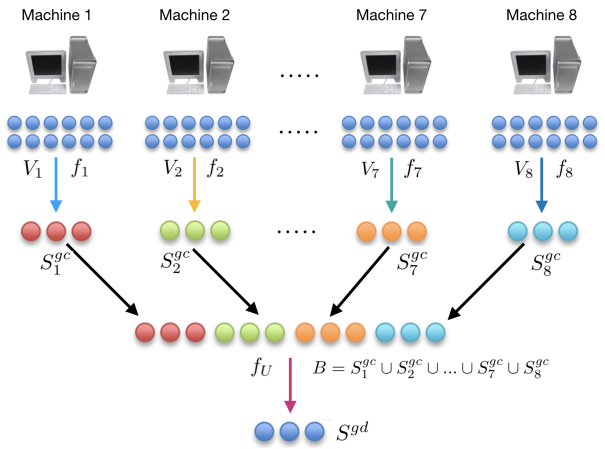

Fig. 2. The greedy DSM with heterogeneous submodular functions.

The composition of the heterogeneous submodular functions $\{f_1, ..., f_8\}$ relies on the linguistic knowledge shown in Table 1, which suggests that a tri-phone state can be converted into 8 broad classes of bi-phone states [5, 8]. For example, a tri-phone state sh-iy+n[2] corresponds to 8 broad classes of bi-phone states (palatal-iy[2], fricative-iy[2], iy[2]+nasal, unvoiced-iy[2], continuent-iy[2], iy[2]+coronal, iy[2]+voiced, and iy[2]+alveolar). Thus, there are 8 heterogeneous submodular functions based on the bi-phone state features in total for the greedy DSM algorithm.

The greedy DSM algorithm with heterogeneous submodular functions ensures a constant lower bound to the optimal value as that obtained with equation (6).

Place of articulation
1. *Front Vowel*: iy ih eh ae aw ey y
2. *Central Vowel*: ah er hh
3. *Back Vowel*: aa ao uh uw ay ow oy
4. *Coronal*: d l n s t z r th dh
5. *Palatal*: sh zh jh ch
6. *Labial*: b f m p v w
7. *Velar*: g k ng
8. *Silence*: sil
Production manner
1. *High Vowel*: ih iy uh uw
2. *Mid Vowel*: ah eh ey ow er
3. *Low Vowel*: aa ae aw ay oy ao
4. *Fricative*: jh ch s sh z f zh th v dh hh
5. *Nasal*: m n ng
6. *Stop Consonant*: b p t d k g
7. *Approximant*: w y l r
8. *Silence*: sil
Voicedness
1. *Voiced*: iy ih eh ey ae aa aw ay ah ao oy ow uh uw er b d dh g jh l m n ng r v w y z zh
2. *Unvoiced*: p f th t s sh ch k hh
3. *Silence*: sil
Miscellaneous
1. *Short Vowel*: eh ih uh ae ah y oy
2. *Long Vowel*: iy uw aa
3. *Diphthong*: ey aw ow ao
4. *ay*: ay
5. *Retroflex*: er r
6. *Affricate*: ch jh
7. *Alveolar*: s z t d n l
8. *Continuent*: sh th dh hh m f ng v w zh
9. *Non Continuent*: p b g k
10: *Silence*: sil

Table 1. Phonetic knowledge from tri-phones to bi-phones.

4. EXPERIMENTS

4.1. Experimental setups

Our experiments were conducted using the 1300 hours of conversational English telephone speech data from the Switchboard, Switchboard Cellular, and Fisher databases as the acoustic training material. The development and test datasets were the 2001 and 2002 NIST Rich Transcription development sets, with 2.2 hours and 6.3 hours of acoustic data,

respectively. Data preprocessing includes extracting 39-dimensional Mel Frequency Cepstrum Coefficient (MFCC) features that correspond to 25.6ms of speech signals. In addition, mean and variance speaker normalization were also applied [9].

The acoustic models are initialized as clustered tri-phones modeled by 3-state left-to-right hidden Markov models (HMMs). The state emission probability in the HMMs was modeled by the Gaussian mixture model (GMM). The DNN targets consisted of approximately 7800 clustered tri-phone states. All sequential labels corresponding to the training data were generated by forced-alignment based on HMM-GMM. A 3-gram language model, built from the training material, was used for the decoding.

The units at the input layer of each DNN correspond to a long-context feature vector that was generated by concatenating 11 consecutive frames of the primary MFCC feature followed by a discrete cosine transformation (DCT). Thus, the dimension of the initial long-context feature was 429 and the number was reduced to 361 after DCT. There were 4 hidden layers in total with a setup of 1024-1024-1024-1024 for the DNN construction. The parameters of the hidden layers were initialized via Restricted Boltzmann Machine pre-training [11], and then fine-tuned by the Multi-layer Perceptron Back-propagation algorithm. Besides, the feature-based maximum likelihood linear regression was applied to the DNN speaker adaptation [12].

As for the subsets of training data selection for acoustic modeling, the sizes of the data subsets were 20%, 10%, 5%, and 1% of the total training data. Since a tri-phone state corresponds to 8 bi-phone states, 8 machines were used in the greedy DSM with heterogeneous submodular functions. In addition, this value represents an acceptable trade-off between the data selection efficiency and ASR results in the greedy DSM with the homogeneous submodular function, as will be presented next. In both greedy DSM algorithms, the ground set V and the constraint l correspond to the index of the training data and the cardinality of the final subset, respectively.

4.2. Experimental results

First, the number of machines M used in the greedy DSM algorithm with the homogeneous submodular function is tested and the corresponding ASR results are given in Table 2.

Number of machines	1%	5%	10%	20%
M=4	41.1	32.0	29.6	28.3
M=8	41.6	32.4	29.9	28.7
M=12	42.4	33.5	30.5	29.3

Table 2. WERs for the greedy DSM with the homogeneous submodular function (%).

The results in Table 2 suggest that ASR performance decreases to varying degrees when M increases. Particularly, for $M = 12$, the ASR results are decreased significantly. Thus, it is necessary to balance the data selection efficiency and the ASR results. In the next experiments, $M = 8$ was chosen in the greedy DSM algorithm with the homogeneous submodular function since this represents an acceptable trade-off between these factors.

Table 3 shows the ASR results from the different subsets of training data output by the two greedy DSM algorithms, the naive greedy algorithm, and a random data selection approach. Note that the word error rate (WER) obtained by the KALDI toolkit [13] when all the training data are fed in the system for acoustic model training is 25.8%.

Methods	1%	5%	10%	20%
Random	43.5	33.9	31.2	29.6
Naive greedy	40.9	31.5	29.2	28.1
Greedy DSM homo.	41.6	32.4	29.9	28.7
Greedy DSM heter.	41.2	31.9	29.5	28.3

Table 3. WERs for submodular data selection algorithms (%). 'Greedy DSM homo.' refers to the greedy DSM algorithm with the homogeneous submodular function, and 'Greedy DSM heter.' represents the greedy DSM algorithm with heterogeneous submodular functions.

As shown in Table 3, both the naive greedy and the two greedy DSM algorithms obtain much better ASR results than the random data selection, while the greedy DSM algorithms obtain ASR results that are marginally below the naive greedy algorithm. However, our experiments show that the data selection process from the two greedy DSM algorithms can speed up the system training more than 6 times in average with respect to the naive greedy algorithm.

5. CONCLUSION

This paper has presented two greedy distributed submodular maximization algorithms to efficiently extract the most meaningful speech data from huge training datasets so that these speech data can be then employed for acoustic model training. An homogeneous submodular function and heterogeneous submodular functions are involved in those algorithms. The experiments show that the two DSM algorithms can significantly speed up the data selection process with respect to the naive greedy algorithm, with a slight reduction in the ASR performance when 8 machines are employed in the DSM algorithms.

6. ACKNOWLEDGEMENT

This research is funded by the Graduate Program for Social ICT Global Creative Leaders of the University of Tokyo, courtesy of the donation of Titan-X GPU workstation mentioned by the program above. Besides, we would like to thank Prof. Shayan Oveis Gharan from University of Washington for his excellent course 'Design and Analysis of Algorithms' in which the authors completed the formulations of the submodular algorithms presented in this paper.

7. REFERENCES

[1] Kai Wei, Yuzong Liu, K Kirchhoff, and C Bartels, "Submodular Subset Selection for Large-scale Speech Training Data," in *ICASSP*, 2014, pp. 3311–3315.

[2] Satoru Fujishige, "Submodular Functions and Optimization," *Elsevier*, vol. 58, 2005.

[3] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause, "Distributed Submodular Maximization: Identifying Representative Elements in Massive Data," in *Neural Information Processing Systems*, 2013, pp. 382–384.

[4] Rafael Da Ponte Barbosa, Alina Ene, Huy L. Nguyen, and Justin Ward, "The Power of Randomization: Distributed Submodular Maximization on Massive Datasets," *CoRR*, vol. abs/1502.02606, pp. 1236–1244, 2015.

[5] Jun Qi and Javier Tejedor, "Robust Submodular Data Partitioning for Distributed Speech Recognition," in *ICASSP*, 2016, pp. 2254–2258.

[6] Jun Qi and Javier Tejedor, "Unsupervised Submodular Rank Aggregation on Score-based Permutations," *arXiv preprint arXiv:1707.01166*, 2017.

[7] Richard Guy, Haim Hanani, Norbert Sauer, and Johanan Schnheim, "Combinatorial Structures and their Applications," *Gordon and Breach, Science Publishers, New York-London-Paris*, 1970.

[8] Guangsen Wang and Khe Chai Sim, "Regression-Based Context-Dependent Modeling of Deep Neural Networks for Speech Recognition," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 22, no. 11, pp. 1660–1669, 2014.

[9] Jun Qi, Dong Wang, Ji Xu, and Javier Tejedor, "Bottleneck Features based on Gammatone Frequency Cepstral Coefficients," in *Interspeech*, 2013, pp. 1751–1755.

[10] Jun Qi, Dong Wang, Yi Jiang, and Runsheng Liu, "Auditory features based on gammatone filters for robust speech recognition," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 305–308.

[11] Geoffrey Hinton and Ruslan Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[12] Jun Qi, Dong Wang, and Javier Tejedor, "Subspace Models for Bottleneck Features," in *Interspeech*, 2013, pp. 1746–1750.

[13] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlek, Yanmin Qian, and Petr Schwarz, "The Kaldi Speech Recognition Toolkit," in *ASRU Workshop*, 2011.