DATA-DRIVEN SUBVECTOR CLUSTERING USING THE CROSS-ENTROPY METHOD

Gue Jun Jung, Hoon Young Cho, Yung-Hwan Oh*

Department of Electrical Engineering and Computer Science, KAIST
*Digital Content Research Division, ETRI
373-1 Guseong-dong, Yuseong-gu, Deajeon 305-701, Republic of Korea
{sylph,yhoh}@speech.kaist.ac.kr, *hycho@etri.re.kr

ABSTRACT

Automatic Speech Recognition(ASR) systems are limited in the computational power and memory resources, especially in low-memory/low-power environments such as personal digital assistants. The parameter quantization is the one of the ways to achieve these conditions. In this work, we compare various subvector clustering procedures for the parameter quantization in the ASR system and propose a data-driven subvector clustering technique based on the entropy minimization. The Cross-Entropy(CE) method is a good choice for the combinatorial optimization problems. We compare the ASR performance on Resource Management(RM) speech recognition task and show that the proposed technique produces better performance than previous heuristic techniques.

Index Terms— subvector clustering, entropy minimization, Cross-Entropy method

1. INTRODUCTION

Currently the most widely used statistical model for Automatic Speech Recognition is continuous hidden Markov model (CHMM). CHMM provides the high recognition accuracy, but requires much training data and memory. In order to apply this model to the mobile devices, it is necessary to reduce the memory size for model representation, for example the number of parameters in acoustic models.

A simple solution without the effect on the performance is to use less bits per parameter than the typical model. Therefore more advanced numerical representation is necessary in the parameter quantization. Several techniques have been used to achieve such quantization. Scalar quantization jointly clusters the individual elements of parameter vectors (means and diagonal covariances) in order to achieve lower memory requirements [1]. The subvector clustering and quantization has been applied to this problem [2]. In most cases, the subvectors clustering in the quantization used a greedy algorithm that chooses pairs that are most strongly correlated [3].

This work was supported by the Korea Science and Engineering Foundation(KOSEF) through the National Research Lab. Program funded by the Ministry of Science and Technology (No. M10500000131-06J0000-13110)

In this paper, we evaluate and compare various methods for the subvector clustering of parameters in CHMM based ASR system. Specifically, we observe the systematic data-driven vector clustering techniques based on entropy minimization and compare their performance on Resource Management speech recognition task [4]. We propose new entropy minimizing parameter quantization method and show that this method is simple but find near optimal subvectors effectively.

In section 2, we describe previously published subvector clustering algorithms and Section 3 explains our subvector selection algorithm. Section 4 describes the RM speech corpus used and the experimental setup. In section 5 we show the memory-performance tradeoff results of our experiments. Finally, we draw our conclusions in Section 6.

2. SUB-VECTOR QUANTIZATION ALGORITHMS

In the general problem of sub-vector quantization, there are given N vectors $v^{(i)}$, $i = 1, ..., N$-each of dimension D-which are to be quantized in some way. These vectors consist of the N means or N diagonal covariances in a Gaussian-mixture HMM-based ASR system. In the sub-vector selection algorithm, one decides upon M subsets $\{C_j\}_{j=1}^{M}$ of the index set $\mathbf{S} \triangleq \{1, 2, ..., D\}$, where $C_j \subseteq \mathbf{S}$ and where $C_j \cap C_m = \emptyset$ for all $j \neq m$ and $\cup_j C_j = \mathbf{S}$. For each subvectors, there are K code words. This means that the goal is to find the functions

$$f_{C_j}(v_{C_j}^{(i)}) = \bar{v}_{C_j}^{k} \quad 0 \leq j \leq M, 1 \leq k \leq K, \forall i \qquad (1)$$

where $v_{C_j}^{(i)}$ is a partition of the vector $v^{(i)}$ corresponding to the elements within C_j, and $\bar{v}_{C_j}^{k}$ is the kth codeword for that partition. If $|C_j| = 1 \ \forall j$, then this corresponds to elementwise scalar quantization and if $|C_j| = D$, then this corresponds to full vector quantization. Anything in between, it is called as subvector quantization. In this general scheme, any vector element may be clustered with any set of other vector elements. The goal is to find the number of clusters M, the clusters themselves $\{C_j\}_{j=1}^{M}$ satisfying the above, the codebook size K (assumed to be the same for each cluster), and

the quantization function $\{f_{C_j}(\cdot)\}_{j=1}^{M}$. The above quantities need to be found such that both the total memory and computation required are minimized, and also such that the word error rate (WER) increment is at a minimum [3].

From the above, there are broadly two separate issues to solve. The first is how to select the set of subvectors $\{C_j\}_{j=1}^{M}$- that is called the subvector clustering problem. The second issue is how to perform the quantization with the selected set. In this work we focused on the first issue.

2.1. Subvector Clustering

Supposing that $v^{(i)}$ is a sample from a random variable V drawn from some distribution $p(v)$, the best quantization-in terms of number of bits per parameter-is given by $\frac{H(V)}{D} = \frac{H(V_1,V_2,...,V_D)}{D}$ where $H(\cdot)$ is the entropy function [5]. Assuming sufficient samples $v^{(i)}$, it can be shown by the law of large numbers that vector quantization is optimal. It will minimize the overall distortion between the original and the quantized data. There are two problems, however, with this scheme in practice. First, there is rarely enough data given the high dimensionality D of the parameter vectors. Second, the cost of storing the code book tables becomes prohibitive as the number of bits per quantized vector q_{vec} increases. Therefore subvector quantization is an attempt to achieve better results than scalar quantization while avoiding the problems mentioned above [3].

Fixing a particular clustering $\{C_j\}_{j=1}^{M}$, the smallest number of bits per parameter possible under the ideal sub-vector quantization scheme is given by $\frac{1}{M}\sum_{j=1}^{M} H(V_{C_j})$. By the entropic inequalities, it can be shown that:

$$\frac{H(V)}{D} \leq \frac{1}{D}\sum_{j=1}^{M} H(V_{C_j}) \leq \frac{1}{D}\sum_{j=1}^{D} H(V_j) \qquad (2)$$

An additional problem in designing the best subvector set $\{C_j\}_{j=1}^{M}$ is that it is an intractable problem. Even in the case where $|C_j| = 2$, finding the optimal clustering has exponential cost. One of the existing approaches is to manually divide the parameters into the subsets based on the prior knowledge about the vector elements [6]. A greedy algorithm finds clusters that have low entropy [7].

2.2. Greedy-n Pair

In the case where $|C_j| = 2\ \forall j$, minimizing entropy is equivalent to maximizing pair-wise mutual information [8]. The formulation is $H(V_m, V_n) = H(V_m) + H(V_n) - I(V_m; V_n)$ where $I(V_m; V_n)$ is the mutual information between V_m and V_n. This algorithm performs a tree search with branching factor n. The nodes of the tree represent the pair of vector elements with the following restriction: No two nodes on the path from the root to a leaf may contain the same element. The n children of a node are the top n ranked pairs in terms of the mutual information between the two corresponding vector elements. The goal is to find the path from the root to the leaf, this has the maximum sum of the mutual information values of all pairs along the path. This algorithm is summarized as follows [3]:

1) Sort the nodes in decreasing weight

2) Recursively, find the node that maximizes the sum of its weights and the weight of the best path below it.

3) Assign each node in the path with the maximum weight to a 2-d

2.3. Greedy-n m-let

Greedy-n m-let is the general case of Greedy-n Pair algorithm [9]. In the algorithm, n is the branching factor and m is the size of the clusters formed. The measure of dependency can be either average pairwise mutual information or the joint entropy over m variables.

2.4. Maximum Clique Quantization

The previous schemes require a uniform subvector size even though smaller or larger subvectors might exhibit a higher degree of correlation. The maximum-clique scheme adopts a structural approach which prunes the dependency graph. Therefore a part of edges above the threshold will remain [3]. A maximum clique finding algorithm is applied to the sparse graph. When there are overlapping cliques, the one with the maximum average mutual information is chosen and its elements are removed from the graph.

Applying above algorithms, they have some constraints. In Greedy-n m-let, each subvector has the same dimension even though smaller or larger subvectors might exhibit a higher degree of correlation. In maximum clique partition, the number of subset(M) didn't controlled. Next section, we explain general subvector clustering algorithm which don't need these constraints.

3. SUBVECTOR CLUSTERING USING THE CROSS-ENTROPY METHOD

3.1. Cross-Entropy method

The Cross-Entropy(CE) method is a general Monte Carlo approach to combinatorial and continuous multi-extremal optimization and importance sampling [10]. This method can be applied to static and noisy combinatorial optimization problems such as the traveling salesman problem, DNA sequence alignment, the max-cut problem as well as continuous global optimization problems with many local extrema. The CE method consists of two phases:

1) Generate arbitrary number of random data samples (trajectories, vectors, etc.) according to a specified point distributions.

2) Update the parameters of the random mechanism based on the data to produce a better sample in the next iteration. This step involves minimizing the Cross Entropy or Kullback-Leibler divergence.

3.2. Problem Formulation

The goal of sub-vector clustering algorithm is maximizing mutual information over $\{C_i\}_{i=1}^{M}$ subsets. To apply the CE method for solving the graph partition problem, the object function value must be monotonic decreased or increased as graph splitting is repeated. However the object function using mutual information doesn't satisfy this condition. However this condition can be satisfied with the reciprocal of mutual information instead of the mutual information itself. To compare the value of each subset, we use the sum of all pair-wise the reciprocal of mutual information in subset C_i.

Given the dependency graph **G** we partition the nodes of the graph into the arbitrary $\{C_i\}_{i=1}^{M}$ subsets such that the sum of the weights of the edges within each subset is minimized. Mathematically it can be written as:

$\mathbf{G} = (S, E) : S = \{V_1, V_2, ..., V_D\}$ and $E = \{\frac{1}{I(V_i;V_j)}$ $\forall i, \forall j, i \neq j\}$,

$$\min_{C}\{JE(C) = \sum_{i=1}^{M} JE(C_i)\} \tag{3}$$

where

$$JE(C_i) = \sum_{j \in C_i, k \in C_i, j \neq k} \frac{1}{I(V_j; V_k)} \tag{4}$$

3.2.1. Random subvector partition generation

The first step in the CE method is generating random subvector partitions based on D independent M point distributions. (Total number of point distributions is $D \times M$ for this algorithm.)

$$F(P), \quad P_j = (P_{1j}, ..., P_{Mj}) \tag{5}$$

where $\sum_{i=1}^{M} P_{ij} = 1, P_{ij} \geq 0, 1 \leq j \leq D, 1 \leq i \leq M$

1) Generate a D-dimensional random vector X from $F(P)$ with independent components X_j $(1 \leq j \leq D$, $1 \leq X_j \leq M)$

2) From X construct partitions C_i such that the vector C_i contains the set of indices $\{j : X_j = i\}$

3) Calculate the entropy of the sample function associated with the random partition C using equation (4)

3.2.2. Main algorithm

1) Choose an initial reference vector P, say the components $P_{11} = 1, P_{ij} = \frac{1}{M}(1 \leq i \leq M, 2 \leq j \leq D)$. Generate N random vectors X^n $(1 \leq n \leq N)$ from $F(P)$ and calculate their corresponding joint entropies.

2) Find the maximum γ_0 s.t. $E\{I(JE(X^n) \leq \gamma_0)\} \geq \rho$, where I is the indicator function and ρ is the rate of important samples. Set $t = 1$.

3) Calculate the following equation and update the point distributions.

$$P_{ij} = \frac{\sum_{1 \leq n \leq N : X_j^n = i} I(JE(X^n) \leq \gamma_{t-1})}{\sum_{1 \leq n \leq N} I(JE(X^n) \leq \gamma_{t-1})} \tag{6}$$

4) Generate new N random vectors X^n from $F(P)$ and find the solution γ_t such as step2.

5) For some $t \geq k$ and k(=5), if $\gamma_t = \gamma_{t-1} = \cdots = \gamma_{t-k}$, stop and deliver γ_t as an estimate of γ. Otherwise, set t=t+1 and go to Step 3).

4. EXPERIMENTS

4.1. Database and Initial System Configuration

To show the effectiveness of the proposed method, we performed several experiments on the speaker independent word recognition task using the Resource Managements (RM) database. For initial CHMMs, we trained word-internal State-Clustered Triphone models which are 3-state left-to-right HMMs with 6 mixture Gaussians per state (the number of total mixtures is 9492) [4] using 39-dimensional feature vectors (12 MFCCs and log energy, their first and second order time derivatives). The baseline system showed 2.81% word error rate(WER) with 3640 KB memory usage.

4.2. Experimental Results

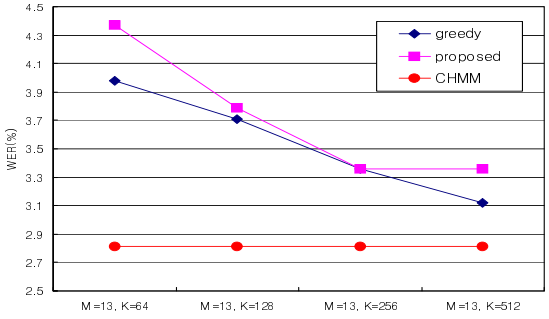

Fig. 1. Word Error Rate on 13 subvectors (K : No. of codewords)

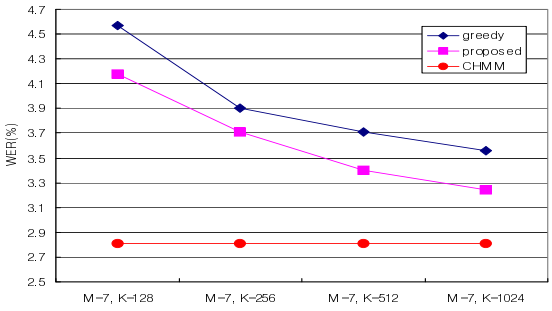

Fig. 2. Word Error Rate on 7 subvectors (K : No. of codewords)

We compared it with two kinds of conventional algorithms described in section 2. Figure 1 shows the WER in a case where the uniform subvector size is the most properable due to the orthogonal feature characteristic. Figure 2 shows the WER in a non-uniform subvector size. In our approach mutual information is indirectly maximized so first figure shows a little performance degradation. However in the second case, our method always outperformed. Next we compared it with the maximum clique quantization. As shown in the table 1, proposed method showed always better performance because each subvector always consists of clique in our method. Therefore we can assume that the maximum clique quantization is the subset of our algorithm.

The memory needed in each algorithm is compared in table 2. It can be seen that the memory efficiency of subvector clustering algorithms have superiority over other methods.

Finally we considered the minimum computation time. Previous algorithms need at most ${}_DC_{D/M}$ calculation to find the maximum parts in the graph. If D is large and D/M is a middle, for example $D = 39$ and $D/M = 13$, it needs a lot of time to evaluate the graph. However, our method only needs to generate $D \times M$ parameters so it can solve the problem quickly.

Table 1. Word Error Rate of the maximum clique quantization (threshold=30%)

No. of codwords	64	128	256	512
maximum clique	5.04	4.33	3.99	3.94
proposed method	4.37	3.79	3.36	3.3

Table 2. Memory usage for the whole model (KB)

No. of codewords	64	128	256	512
13 subvectors	221	270	339	447
7 subvectors	157	173	228	323

5. CONCLUSION

In this paper, we compared various subvector clustering algorithms and proposed a new data-driven subvector clustering technique using the Cross-Entropy method. This technique doesn't need any constraints used in previous algorithms such as uniform subvector size and the pruning rate of the edges.

In various experiments, the proposed algorithm produced better performance than previous heuristic techniques. In the future, we need to improve the optimizing and find a better measure which can describe the relationship between the subvector clustering and the vector quantization errors better than the entropy-based measure.

6. REFERENCES

[1] Vasilache M., "Speech recognition using hmms with quantized parameters," in *International Conference on Spoken Language Processing*, 2000, vol. 1, pp. 441–443.

[2] B.Mak E.Bocchieri, "Subspace distribution clustering hidden markov model," *IEEE Transaction on Speech and Audio Processing*, vol. 9(3), pp. 264–275, March 2001.

[3] X.Li K.Filali and J.Bilmes, "Data-driven vector clustering for low-memory footprint asr," in *International Conference on Spoken Language Processing*, 2002, pp. 1601–1604.

[4] J.Bernstein P.Price, W.M.Fisher and D.S. Pallett, "The darpa 1000-word resource management database for continuous speech recognition," *Computer Speech and Language*, vol. 6, 1992.

[5] J. L. Casti, "The shannon coding theorem.," in *Five More Golden Rules: Knots, Codes, Chaos, and Other Great Theories of 20th-Century Mathematics*. 2000, pp. 207–254, Wiley, New York.

[6] R.Bisiani M.Ravishankar and E.Thayer, "Sub-vector clustering to improve memory and speed performance of acoustic likelihood computation," in *Eurospeech*, 1997.

[7] E.Bocchieri and B.Mak, "Subspace distribution clustering for continuous observation density hidden markov models," in *Eurospeech*, 1997, vol. 1, pp. 107–110.

[8] T.M.Cover and J.A.Thomas, "Elements of information theory," 1991.

[9] X.Li K.Filali and J.Bilmes, "Algorithms for data-driven asr parameter quantization," vol. 20(4), pp. 625–643, October 2006.

[10] Reuven Y. Rubinstein, "Cross-entropy and rare events for maximal cut and partition problems," *ACM Transactions on Modeling and Computer Simulation*, vol. 12(1), pp. 27–53, January 2002.