Modified LBG Clustering Algorithms for Small Unit Inventory in Corpus-based TTS System

Jinyoung Kim[1], Joohun Lee[2]

[1] Deptartment of Electronics Engineering, Chonnam National University,
Yongbong-dong 300, Kwangjoo, South Korea
kimjin@dsp.chonnam.ac.kr
[2] Department of Internet Broadcasting, Dong-Ah Broadcasting College,
Ansung-si, Gyunggi-do, South Korea
vincelee64@freechal.com

Abstract Recently, Corpus-based Text-To-Speech (CB-TTS) has been actively studied through the world, for the improvement of synthesized speech heading to human-like naturalness. However, the application of TTS is very restricted due to its large database (DB) size. In this paper, to solve this problem, we propose two modified algorithms of LBG clustering algorithm (split k-means). We introduce a terminating threshold of total cost in the first modification. The number of selected inventories becomes less than target cluster number if total cost reduction is enough to end iteration process. Considering frequency information of unit instances, which is obtained during synthesizing large text corpus, makes the second modification. To consider frequency information we proposed modified cost function of MinMax commonly used in selecting centroids.

To evaluate the proposed method, we compared synthesized speech qualities of two modified LBG clustering algorithm with that of original DB. After reducing the DB to almost same size, we performed perceptual tests with some test sentences. From the perceptual test results, we can observe that our algorithm achieves the successful performance with reducing most the DB size and maintaining good speech quality.

1. Introduction

The growing popularity of speech synthesizer enabling comfortable man-machine-interfaces demands high quality of the synthesized speech. Corpus-based speech synthesis approach has become one of the most attractive synthesis methods since it guarantees high perceptual quality with good naturalness [1][2][3]. This high speech quality of the corpus-based synthesizer satisfies the needs of commercial product on the contrary of rule-based approach. However, to maintain high quality, that approach must use pre-recorded prompts which forces application developers to prepare fairly large size of speech DB, *i.e.*, large corpus [3][4]. This leads the application of the corpus-based synthesizer to limited area such as the server-oriented service requiring large DB. Considering the rapid growth of mobile communication services, this handicap has become one of the most serious problems in the real-world applications. For the embedded application in such as PDA, cellular-phone and e-book, the high quality speech synthesizer with small required DB size less than 10 Mbyte is strongly demanded.

In this paper, to develop the speech synthesizer with small unit inventory, we propose DB reduction methods with maintaining high speech quality. In these methods, for better clustering process, we consider total cost factor in iterations of clustering and the frequency of occurrence of each unit. We implement two modifications to general LBG (split k-means clustering) algorithm. These modifications are introduced based on the following two facts. Firstly, the total cost values for each unit widely spread. For examples, in case of unvoiced consonant units, the cost values are generally low compared with those of voiced units with the same number of instances. Secondly, the frequencies of each instance are not equal. Among many instances only small number of instances are used in real synthesizing. Considering these two facts, we propose two modified LBG clustering algorithms for database reduction.

To evaluate our approach, we compare our methods with the traditional split k-means clustering approach. The used CB-TTS is a Korean TTS developed for our own purpose. This TTS has 563Mbyte speech DB (16bit, 16kHz sampling rate). As shown in our experimental results, the proposed methods outperform the common k-means clustering approach. Moreover, the data size is reduced down to 1/4 with the synthesized speech quality being kept.

2. Overview of the used Corpus-based speech synthesizer

In our work, we used our own speech synthesizer called CNU TTS. To develop CB-TTS, we used generally adopted approach and implemented our own TTS very similar to Japanese speech synthesizer CHATRA [4] developed by ATR. The sentences for speech corpus were carefully selected to reflect the diverse Korean phonetic characteristics. For this, we selected 3200 sentences (32,671 phrases and 103,084 syllables) from various kinds of text including newspaper, editorial article, novel and essay. Those sentences were recorded with the help of professional announcer for 16-hour database. Every recording was performed by 2 channels (the one of which for Laryngograph signal and the other for speech signal). Speech signal was sampled at 16kHz with 16 bit A/D converting. Here, Laryngograph signal was used to extract pitch information.

The speech synthesizer engine used in this paper is a kind of corpus-based synthesizer. Especially, prosody generation was done by using CART (Classification And Regression Tree) and, for training, several parameters considering syntactic and phonetic context are used. As synthesis unit, we used tri-phone unit, total number of which was 12,021. The speech was synthesized by the selection-based synthesis so that the best N candidates were selected considering phone value and phonetic environment such as prosodic and phonetic context and, then, by using Viterbi algorithm, the optimum sequence guaranteeing the best speech quality was found. During the above process, various distance measures and functions were required to be defined such as phonetic distance function, target-distance function and concatenation-cost function. The main features of CNU TTS are summarized in table 1.

3. DB reduction algorithms

Table 1. The main features of our CB-TTS.

Speech corpus	- 32,000 sentences (32,671 phrases and 103,084 syllables) - 2 channel recording (speech, Larygograph signal) - 16 bit, 16 kHz sampling - Speaker: announcer of local broadcasting system
Synthesis unit	- Tri-phone unit - Number of tri-phone: 12,021 - Labeling: automatic labeling by HTK followed by manual correction
Prosody rules	- Trained using z-score based CART approach - Training corpus: 1,000 spoken sentences from speech corpus
Optimal unit search	- Distance measure: target and concatenation distance measures - Viterbi-search-based optimal unit selection

Speech synthesis database consists of tri-phone unit appeared in spoken sentences accompanied with their frequency of instance. Usually, the frequency of instance for each tri-phone varies from 1 to thousands and this means that the frequently occurred tri-phones contain more redundancy. Therefore, our proposed methods should focus on how to eliminate those redundancies by using the appropriate clustering algorithms with avoiding the quality degradation of synthesized speech caused by the reduced database.

To reduce the database size, the number of unit inventory for corpus-based TTS, we are considering the efficient clustering algorithm, especially LBG (split k-means) algorithm. For this, a distance function between two sample instances is necessarily defined. Here, the distance means that between two different instances of same tri-phone unit. In the following section, we define the distance functions used in this paper and, based on this function, propose two new DB reduction algorithms.

3.1 Distance function

The distance function between two different instances belonging to same tri-phone unit is defined as following equation.

$$D_t(a_i, a_j) = \lambda D_{phon}(a_i, a_j) + (1-\lambda) D_{pro-spec}(a_i, a_j), \quad (1)$$

where a_i and a_j are two different instances of same tri-phone unit. D_{phon} and $D_{pro-spec}$ represent the phonetic distance and the prosodic-spectral distance, respectively. Empirically, λ is set to 0.5 to give appropriate weight. This distance function is used in our CB-TTS system. From the equation, we can say that the distance is defined as sum of the phonetic distance and the prosodic-spectral distance. . According to our experience the synthesized speech quality become better if phonetic distance is used. Here, one noticeable fact is that some of the feature parameters are categorical variables. The phonetic distance and the prosodic-spectral distance are described as follows.

- Phonetic distance

Phonetic distance means how different the degree of contextual match considering phonetic environment of the selected tri-phone unit is. However, since tri-phone does not reflect well the whole articulation, we modified the distance by considering both the preceding phone and the following phone of the tri-phone unit. Therefore, finally, the phonetic distance can be defined as

$$D_{phon} = D_{ph}(p_{i-2}, p_{j-2}) + D_{ph}(p_{i+2}, p_{j+2}), \quad (2)$$

where D_{ph} is the distance between two phonemes as we call it phoneme distance. Thus the problem turns to depend on what the phoneme distance is defined as. In our work, we used the method base on phonological characteristics of phoneme, such as articulation, position and stress, for instance. We gave 0 or 1 according to whether two phonemes have the same environment for each phonological characteristic, respectively, and sums up those 0's or 1's finally to compute the Hamming distance between those phonemes.

- Prosodic-Spectral distance

Prosodic-spectral distance represents how much the prosodic and spectral difference between two instances of same tri-phone unit is. This is defined as the following equation as

$$D_{pro-spec}(a_i, a_j) = w_1 D_{dur}(a_i, a_j) + w_2 D_{pit}(a_i, a_j) + w_3 D_{\text{int}}(a_i, a_j) + w_4 D_{spec}(a_i, a_j), \quad (3)$$

where $w_1 + w_2 + w_3 + w_4 = 1$ [3][4][5]. In the equation, D_{dur}, D_{pit}, D_{int} and D_{spec} are the distances according to duration difference, pitch difference and intensity difference, spectral difference at boundary respectively. w_i's reflect the weighting factor and are determined empirically from the listening test as fixed values as 0.3, 0.3 and 0.3 and 0.1 for duration, pitch and intensity weighting. Each distance is defined by Mahalanobis distance, one of the modified Euclidean distances.

3.2 Modification 1 : LBG clustering with terminating condition

LBG algorithm, split k-means clustering is one of the methods to group the members according to their closeness from the view of neighboring distance or distortion. By considering their phonetic and prosodic distances with proper weights, each tri-phone unit is clustered and only the centroids of the clusters are registered as inventory units to reduce the database size [6][7]. Figure 1 shows the flow chart of modified LBG algorithm.

In the algorithm, the maximum number of cluster is determined by the variable of 'NumTarget'. However, even when the number of clusters is less than 'NumTarget', we decide to terminate further clustering process if the total average distance becomes less than the variable, 'Threshold_Value'. Also, when the number of total instances is less than 'NumTarget', this number replaces the value of 'NumTarget'. This explains our first idea for efficient data reduction, that is, how to achieve more data reduction with same TTS quality.

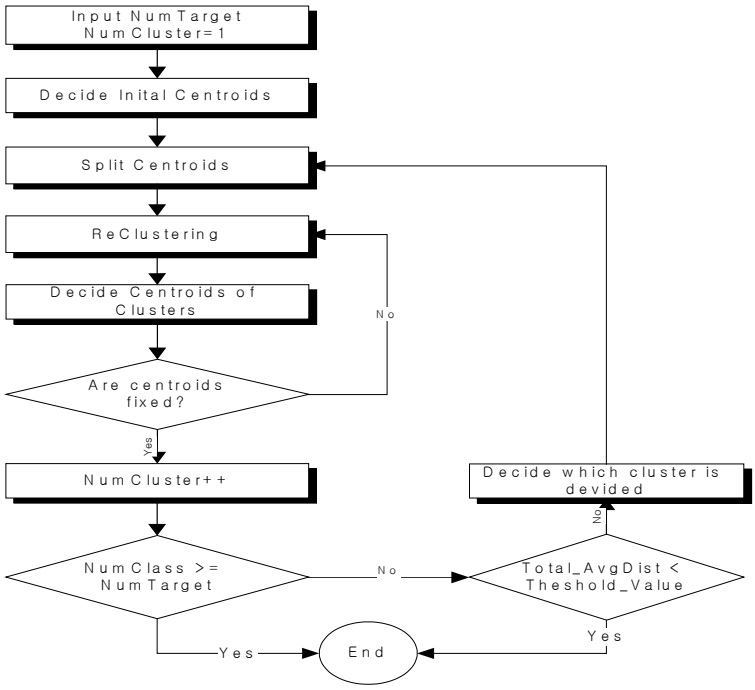

Figure 1. Modified LBG algorithm with terminating condition for database reduction.

By the way, during the clustering process, each instance member has non-numerical value so that each centroid of the cluster cannot be determined by numerical averaging. To solve this problem, we adopted 'MinMax' approach to find the centroid of each cluster. The proposed algorithm is explained as follows.

First, we choose the initial centeroid (C_1) representing whole members. This centroid is determined by MinMax center approach as like.

$$Centroid = \underset{i}{Min}(\underset{j}{Max}\, Dist(a_i, a_j)).$$

Secondly, we split this cluster into two separated clusters. The splitting is performed by selecting another member C_2 closest to the initial centroid C_1 and by regrouping the rest of whole members based on the neighboring distances to those C_1 and C_2. After regrouping the whole members into two clusters, a new centroid for each cluster is found repeatedly by using the above equation. Those regrouping and the finding of the new centroids for the clusters are repeated until the centroid of each cluster does not change any more.

By the way, if the distance 'Total_aveDist' becomes less than the predefined value of 'Threshold_Value', the splitting process is terminated before the total number of the obtained clusters gets to 'NumTarget'. 'Total_aveDist' is the average value of the distances between the rest members and the centroid of each cluster. If this 'Total_aveDist' is less than 'Threshold_Value', we can conclude the whole

Table 2. Comparison results of database of the original and the modified LBG.

	General LBG clustering	Modified LBG clustering (Modification 1)
Database size	563MB-> 201Mbyte	563MB->158MB

members are well clustered for each obtained centroid enough to represent all members.

Additionally, during the splitting process, we choose the cluster of the largest distortion value to be split. This maintains the balanced number of the members of each cluster and, therefore, those centroid can reflect the various phonological situations. Afterwards, with small inventory of tri-phone units, we can get good quality of synthesized speech. The experimental results are shown in table 2. From the result, with 'NumTarget' and 'Threshold_Value' being 30 and 0.4, respectively, the size of tri-phone database 563MB can be reduced into 158MB.

3.3 Modification 2 : LBG clustering considering frequency information

According to our experiences, some unit instances are rarely used in real speech synthesis. So, if we register only the unit instances occurred in the real synthesis, we can reduce speech DB size. However, the reduced DB based on the above approach might be still large when a large text corpus is applied to TTS system. Thus more data reduction is required. By the way, the frequencies of the occurred unit instances are observed not to be equal. The frequently occurred instances are less than a half of all the unit instances. Thus, the frequency information should be considered to select the optimum set of unit instances. This is our second idea for efficient DB reduction.

Therefore, we should modify the previous cost function by introducing new factor concerning with the frequencies of the units. This consideration is reasonable since the frequency can be important information to choose the optimum centroid for better clustering results. For this purpose, we proposed the modified distance function with reflecting the frequency of units as follows.

$$D(a_i) = \alpha \frac{1}{N} \sum_{j=1}^{M} N_j d(a_i, a_j) + (1-\alpha) Max_j d(a_i, a_j), \qquad (4)$$

where M is the number of members except i-th member in the cluster, N_j is the frequency of j-th member and N is the total sum of the frequencies of all members in the cluster. In (4), $D(a_i)$ represents the frequency-considered cost calculated for i-th member. The fore term of the given equation is the average distance from i-th member with considering N_j of j-th member. In the second term, $Max_j d(a_i, a_j)$ is the conventional cost function from i-th member to j-th member, which is based on MinMax approach. From the equation, we can say that the member with small frequency has less possibility to be selected as the centroid of the cluster since $D(a_i)$ increases proportionally to the decrease of the frequency.

Therefore, we select the member having the least $D(a_i)$ and the appropriate values of α to control the contribution weights between the traditional distance and the frequency-considered distance. Also, to choose which cluster to be split, we calculate the total frequency-considered distance from the centroid and find the cluster having large distance value. The following equation represents the total frequency-considered distance function.

$$D_{total} = \sum_{j=1}^{M} N_j D(a_{center}, a_j), \tag{5}$$

where $D(a_{center}, a_j)$ is the distance from the center member to j-th member in the cluster.

The above process leads to good clustering results since the clusters having large frequency-considered distance as well as those containing higher frequency member will be split. Moreover, the member of high frequency is more likely chosen as the centeroid of the cluster. Finally, for the not-occurred units, LBG clustering is performed with 'NumTarget' of 30 and 'Threshold-Value' of 0.4 as usual.

4. Listening test results and discussion

To compare the synthesized speech quality of the proposed two methods, we experimented with 10 randomly selected Korean sentences not belonging to the recording sentences. With those sentences, the following 8 tests were performed.

Test 1: TTS with original unreduced database of 563MB.
Test 2: TTS with the reduced database of 158MB by the modified LBG clustering with terminating condition (Threshold_Value=0.45, NumTarget=30).
Test 3: TTS with the reduced database using the modified LBG algorithm by considering the frequency information of the units in the 20000 sentences with various α = 0.3, 0.5, 0.7 and 1.0.

For the synthesized speech of the above tests, we compared their naturalness by MOS (Mean Opinion Score) subject evaluation. Total number of people participated into the tests is 6 and their ages are in 20~30. To avoid the expected affecting bias, the test listeners were not given and required any information about the sentences as well as speech signal processing. We averaged the rest 6 persons' except the highest and the lowest scores of 8 persons' to obtain the final score. The results are shown in figure 2.

In the figure, MOS score of full speech DB is 3.86 (NCLS). If α =0, the second modified method become the conventional clustering method (CLS1). From the figure, the best performance with average MOS score of 3.75 is achieved when frequency-weighting factor is 1.0 (CLS2(1.0)). This means that the modified LBG using only the frequency-considered cost is desirable. On the other hand, the DB sized of 563Mbyte is reduced to 161 Mbyte with the great reduction rate of 71.4%. The listening test shows that the proposed method can reduce the speech DB with maintaining the good synthesized speech quality.

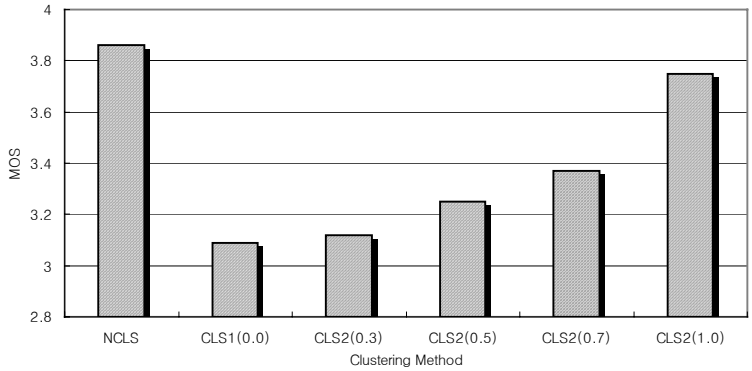

Figure 2. MOS scores with various α .

5. Conclusion

In this paper, we proposed two modifications of LBG clustering for reducing the size of unit inventory, the database of tri-phone units for speech synthesis. To compare the synthesized speech quality as well as the reduction amount, we performed several MOS tests with the reduced database obtained by the proposed algorithms. The proposed methods are 1) database reduction algorithm with terminating condition based on LBG clustering, 2) database reduction algorithm considering frequency information of unit instances. As shown in the experimental results, our methods can reduce database size effectively. We can get 71.4% of the database reduction rate with good synthesize speech quality. For future work, to reduce the database size, we are considering the speech compression technique without causing the degradation of speech quality. For example, if we transform speech DB to 8kHz and 8bit PCM, our CB-TTS might have smaller unit inventory, only 40Mbyte speech DB.

References

1. M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal, “The AT&T Next-Gen TTS System,” Joint Meeting of ASA, EAA, and DAGA, pp.18-24, 1998.
2. H. Hong, A. Acero, X. Huang, J. Liu and M. Plumpe, “Automatic generation synthesis units for trainable text-to-speech systems,” Proceedings of ICASSP’98, pp.293-296, 1998.
3. A. Hunt and A. Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” Proceedings of ICASSP’96, vol. 1, pp.373-376, 1996.
4. N. Campbell and A. Black, “Prosody and the selection of source units for concatenative synthesis,” Progress in speech synthesis edited by J. van Santen, pp. 279-282, Springer Verlag, 1996.
5. A. Black and N. Campbell, “Optimizing selection of units from speech databases for concatenative synthesis,” Proceedings of Eurospeech’95, vol. 1., pp.581-584, Madrid, Spain, 1995.
6. A. Black and P. Taylor, “Automatically clustering similar units for unit selection in speech synthesis,” Proceedings of Eurospeech’97, vol. 2, pp.601-604, Rhodes, Greece, 1997.
7. S. Nakajima and H. Hamada, “Automatic generation of synthesis units based on context oriented clustering,” Proceedings of ICASSP’88, pp.659-662, 1988.