말소리와 음성과학 제1권 제2호 (2009)
pp. 3~8

The Effect of the Number of Clusters on Speech Recognition with Clustering by ART2/LBG

Lee, Chang-Young[1]

ABSTRACT

In an effort to improve speech recognition, we investigated the effect of the number of clusters. In usual LBG clustering, the number of codebook clusters is doubled on each bifurcation and hence cannot be chosen arbitrarily in a natural way. To have the number of clusters at our control, we combined adaptive resonance theory (ART2) with LBG and perform the clustering in two stages. The codebook thus formed was used in subsequent processing of fuzzy vector quantization (FVQ) and HMM for speech recognition tests. Compared to conventional LBG, our method was shown to reduce the best recognition error rate by 0~0.9% depending on the vocabulary size. The result also showed that between 400 and 800 would be the optimal number of clusters in the limit of small and large vocabulary speech recognitions of isolated words, respectively.

Keywords: Speech recognition, ART2, codebook size, number of clusters

1. Introduction

As a method of communication between man and machine, speech recognition affords a very effective interface. It is known in practical applications that the absolute level of performance is relatively unimportant so long as the accuracy exceeds some level [1]. When the accuracy of the recognizer is above a certain threshold (e.g. 92%), the user tends to attribute the occasional error to an improper and/or uncooperative speaking mode of his (or her) own part, rather than to an inadequacy in the speech recognition system. If the performance falls below a certain level, on the other hand, the perception of the user is that the system makes too many errors and is hence unreliable. There are so many factors affecting the performance of the speech recognition system and lots of endeavors for enhancement have been made for several decades.

One of the main elements governing the system accuracy might be phrased in terms of the clustering procedure. As a method to expedite the processing and save the memory, vector quantization (VQ) of the feature vectors extracted from the speech signal is frequently used. In this procedure, we consider some number of representative vectors (centroids or clusters) and use their indices in the pattern classifier such as HMM or neural networks. Here, the following question naturally arises: how many exemplary feature vectors are optimal for the best performance of a specific speech recognizer?

The number of clusters should somehow reflect the number of the basic elements of speech, i.e., phonemes in a language. It is usual to consider about 50 phonemes for speech processing [2], even though there are minor differences from language to language. Therefore, if we choose to use, for example, 256 clusters for vector quantization, it means that five variations for each phoneme on average are being considered. By 'variations' we mean not only the person-to-person differences but the context-dependence in speech production.

It is not known a priori how many variations for each phoneme would yield the best performance in speech recognition. If the number of clusters is too small, then the mesh of discrimination in the feature vector space becomes so coarse that the resolving power becomes weak and distinct enough patterns

[1] Division of Information System Engineering, Dongseo University.

Received: April 30, 2009
Revision received: June 17, 2009
Accepted: June 17, 2009

might be grouped together. If the number of clusters is too large, on the other hand, then the mesh is so refined that similar enough patterns might be classified as different. The best number of clusters should be determined in such a way that discrimination between and identification of similar patterns be optimally reconciled.

For the clustering of the feature vectors, the Linde-Buzo-Gray (LBG) algorithm has long and extensively been used. In this method, the number of clusters are successively doubled on each bifurcation (or binary split) starting from a single cluster. Therefore, the number of clusters can not be chosen arbitrarily in this scheme. Codebooks of orders 8~10 corresponding to 256~1024 clusters are commonly used on empirical grounds.

To examine the effect of the number of clusters on speech recognition in more detail than permitted by the LBG algorithm, we need to employ another tool that permits us to choose the number of clusters. A good candidate for this purpose might be found from the field of neural networks, one of whose main functions is the pattern classification. Among the numerous neural networks, we adopt the network of adaptive resonance theory [3] in this paper.

There are two kinds of this net and the one applicable to continuous-valued inputs is abbreviated as ART2. This network has long been used for unsupervised classification of patterns [4] but applications to speech recognition are found only a little [5]. This network does not fix the number of classes (output nodes) beforehand but has the freedom of creating new classes according to a criterion through the vigilance parameter. Among many revisions of ART2 derived from the original one introduced by Grossberg [3], we will adopt a simple version as described by Kung [6].

The role of ART2 in our study is to choose the number of clusters and the remaining job of clustering is passed to LBG. The combined function is equivalent to K-means clustering with the number of clusters determined by ART2.

2. Clustering by ART2/LBG

We briefly describe the procedure of clustering proposed in this paper, i.e., the serial combination of ART2 with LBG. Let $\boldsymbol{x}$ and $\boldsymbol{w}_j$ denote the input vector and the weight of neuron j, respectively. The criterion of selecting the winner is based on a minimum distance measure (e.g. Euclidean distance). The steps are as follows:

(1) Selection of the winner: Given a new input, a MINNET is used to select the winner $J = \arg\min_j \| \boldsymbol{x}-\boldsymbol{w}_j \|$, that yields the minimum distance.

(2) Vigilance test: The selected neuron passes the vigilance test if

$$\| \boldsymbol{x}-\boldsymbol{w}_j \| < \rho$$

where the vigilance parameter ρ determines the radius of a cluster.

(3) Creation of a new output unit: If the winner fails the vigilance test, a new neuron unit k is created with weight $\boldsymbol{w}_k = \boldsymbol{x}$.

(4) Weight update: If the winner passes the vigilance test, adjust the weight of the winner J by a learning rule. These four steps complete the first stage of our clustering.

(5) Computation of the centroids for the created clusters

(6) Classification of the feature vectors: For each feature vector $\boldsymbol{x}$ in the training set, find the nearest centroid and reassign that cluster to $\boldsymbol{x}$.

(7) Centroid update: The cluster centroids are recomputed according to the newly assigned vectors.

(8) Convergence test: If the changes in centroids fall below a specified level, then stop the clustering. Otherwise, go to the step (6).

The architecture of the above-described procedures is shown in Figure 1.

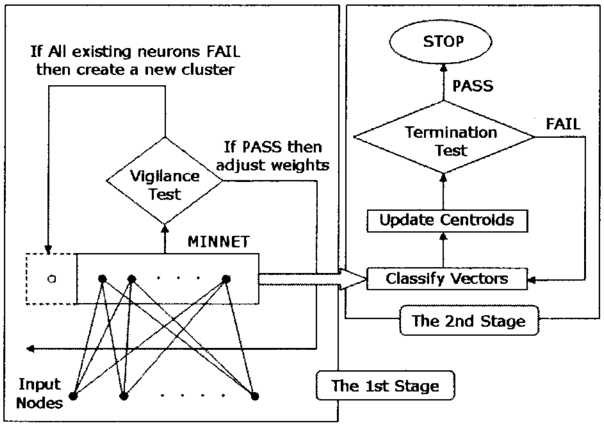

Figure 1. The architecture of the clustering procedures proposed in this paper. In the first stage, the number of clusters is chosen by ART2. The cluster centroids are refined in the second stage by the LBG algorithm.

In the first stage, some number of clusters are formed and fixed hereafter. In this stage, however, cluster assignments to the input vectors are somewhat unsatisfactory. The reason is that the competition is not fair all through the processing since the recently created output node did not have the chance of competition before. Another undesirable feature of the result in this stage is that the vectors are too crowded in some clusters while many clusters have only a single vector residing on them.

As a prescription to remedy these problems and refine the cluster centroids, we employ the LBG algorithm as the second stage of clustering. In this procedure, iterations are continued until no changes occur in centroids. The overall framework of the proposed method is equivalent to the K-means clustering algorithm with the number of clusters K determined by ART2. The combined clustering method of ART2/LBG constitutes the kernel of this paper.

Once the codebook is thus generated, the next procedures are to apply fuzzy vector quantization (FVQ) and fed the resultant vectors into the pattern recognizer. For that purpose, we employ hidden markov model (HMM), one of the popular recognizers, the details of which are expounded in the following.

3. Experiments

Our experiments were performed on a set of phone-balanced 350 Korean words. In order to include the effect of the vocabulary size, the words were divided into four sets as follows. The sets A, B, and C are disjoint each other and D is the union of those three.

Table 1. Four sets of speech data divided for studying the effect of the vocabulary size

Set ID	Number of Words
A	50
B	100
C	200
D	350

Forty people including 20 males and 20 females participated in speech production. In spite of insufficient training data, speech utterances of 40 people were divided into three disjoint groups as follows.

Table 2. Division of the 40 people of speech production into three groups

Group ID	Number of People
I	32
II	4
III	4

The group I consisting of 32 people's speech was used for the codebook generation and training of HMM parameters $\lambda=(\pi,A,B)$. These parameters are continually updated during iterations. In order to choose which values of λ to use in actual test of speech recognition, some test speeches are necessary. The parameters that yield the best performance on the group II were stored and used for the group III to get the final performance of the speaker-independent speech recognition system. This prescription prevents the system from falling too deep into the local minimum driven by the training samples of the group I and hence becoming less robust against the speaker-independence when applied to the group III.

Each speech utterance was sampled at 16 kHz and quantized by 16 bits. 512 data points corresponding to 32 ms of time duration were taken to be a frame. The next frame was obtained by shifting 170 data points, thereby overlapping the adjacent frames by ≒2/3 in order not to lose any information contents of coarticulation. To each frame, the Hanning window was applied after pre-emphasis for spectral flattening. MFCC feature vectors of order 13 were then obtained.

Codebooks of variable sizes were generated by the procedures described above on the MFCC feature vectors of the group I. The distances between the feature vectors and the codebook centroids were calculated and sorted. Appropriately normalized fuzzy membership values [7] were assigned to the nearest two clusters and fed to HMM for speech recognition test.

For the HMM, a non-ergodic left-right (or Bakis) model was adopted. The number of states that is set separately for each class (word) was made proportional to the average number of frames of the training samples in that class [8]. Initial estimation of HMM parameters was obtained by K-means segmental clustering after the first training. By this procedure, convergence of the parameters became so fast that enough convergence was reached after several epochs of training iterations.

Backward state transitions were prohibited by suppressing the state transition probabilities a_{ij} with $i>j$ to a very small value but skipping of states was allowed. The last frame was restricted to end up with the final state associated with the word being scored within a tolerance of 3. Parameter reestimation was performed by Baum-Welch reestimation formula with scaled multiple observation sequences to avoid machine-errors caused by repetitive multiplication of small numbers. After each iteration, the event observation probabilities $b_i(j)$ were boosted above a small value [9].

Three features were monitored while training the HMM parameters: (1) the recognition error rate for the group II of Table 2, (2) the total probability likelihood of events summed over all the words of the training set according to the trained model, and (3) the event observation probabilities for the first state of the first word in the vocabulary list. Training was terminated when the convergences for these three features were thought to be enough. The parameter values of $\lambda = (\pi, A, B)$ that give the best result for the group II were stored and used in speech recognition test on the group III.

4. Results and Discussion

Figure 2 shows the recognition error rate E vs. the number of clusters N as determined by successive bifurcations of LBG algorithm. The abscissa is in logarithmic scale. From this result, we see that the recognition error rate is minimum in the vicinity of N=512 clusters corresponding to the bifurcation order of 9. This result implies that consideration of around 10 variations for each phoneme is appropriate for speech recognition task.

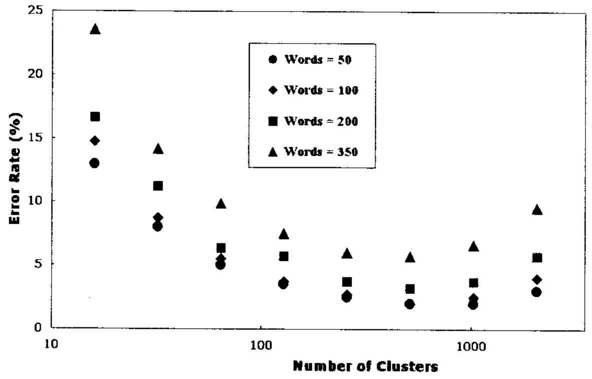

Figure 2. The recognition error rate vs. the number of clusters. The abscissa is in logarithmic scale. It is seen that the recognition error rate shows its minimum in the vicinity of 512 clusters.

If the number of clusters is too small, then the resolution of discrimination between distinct patterns becomes correspondingly small. If it is too large, on the other hand, then the system might discern similar enough patterns as distinct. The system performance therefore shows its maximum somewhere in between the two extremes of being too coarse and too fine clusters.

Though the difference might not be referred to be very notable near the minimum of the error rate, it is desirable, if possible, to find the optimal number of clusters that would yield the best result. In order to investigate more details of the error rate vs. the number of clusters, however, we have to employ another tool that allows adjustment of the number of clusters.

Figure 3 shows the number of clusters formed by ART2 as the vigilance parameter is changed for the 4 sets of different vocabulary sizes. For a fixed vigilance parameter, more clusters are formed as the vocabulary size becomes larger. This is as expected since the radius of the hypersphere encompassing the whole feature vectors should increase as more words and thus more feature vectors are involved.

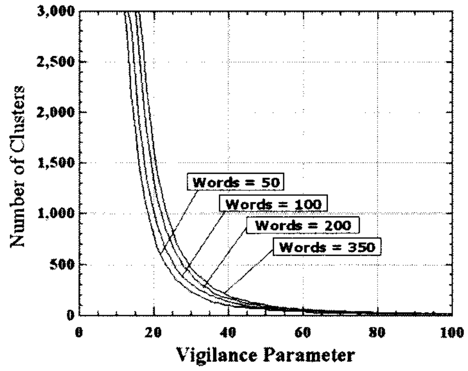

Figure 3. The number of clusters formed by ART2 vs. the vigilance parameter for 4 sets of different vocabulary sizes. For a given vigilance parameter, more clusters are formed as the vocabulary size becomes larger.

In order to see more detailed feature than the one of Figure 2, we use the codebook generated by ART2/LBG clustering procedure proposed in this paper. The subsequent procedures of FVQ/HMM are the same as before. Figure 4 shows the result for the vocabulary size of $W = 350$ words. The solid line represents the curve-fitting result assuming that the data is parabolic. The overall behavior for other vocabulary sizes was found to be similar to Figure 4.

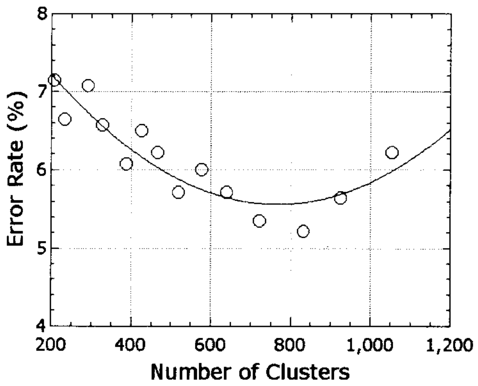

Figure 4. The recognition error rate vs. the number of clusters formed by ART2/LBG for the vocabulary size of $W = 350$ words. The solid line represents the curve-fitting result assuming that the data is parabolic.

Table 3 shows the minimum error rates for the two methods of clustering and four vocabulary sizes. By the conventional LBG clustering, the minimum occurred at $N=512$ for all the vocabulary sizes. By ART2/LBG, meanwhile, the best number of clusters was found to increase as the vocabulary size. The overall performance was enhanced by 0~0.9% by using ART2/LBG.

Table 3. The minimum recognition error rate and the best number of clusters obtained from the two clustering methods.

Number of words		50	100	200	350
LBG	Minimum error rate (%)	2.0	2.0	3.3	5.7
	The best number of clusters	512			
ART2/LBG	Minimum error rate (%)	1.8	2.0	2.4	5.2
	The best number of clusters	545	620	727	831

The best (or optimal) number of clusters in Table 3 was obtained from curve-fitting of the data in Figure 4 by assuming a parabola near the minimum of the error rate. Figure 5 shows the number of clusters that yields the best performance for various vocabulary sizes.

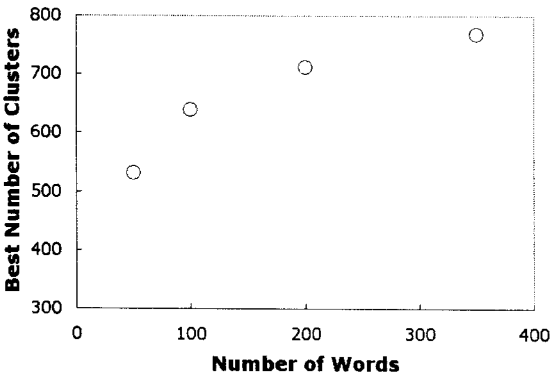

Figure 5. The number of clusters that yields the best performance for various vocabulary sizes obtained from curve-fitting of the experimental data.

For numerical analysis of the data in Figure 5, we employ the following model

$$N(W) = N_\infty - (N_\infty - N_0)\exp(-\alpha W) \qquad (1)$$

where N_0 and N_∞ denote the values of N in the limits of small and large vocabulary words W. The parameter α is also to be determined from the data.

Once N_∞ is known, it is not difficult to fit the data by converting the above expression to

$$N_\infty - N(W) = (N_\infty - N_0)\exp(-\alpha W) \qquad (2)$$

By taking the logarithm of both sides and applying the routine of the least square method, N_0 and α can be calculated. Since N_∞ is not known beforehand, however, we have to vary it, obtain N_0 and α from the least square method, calculate the sum of squares of the difference between Eq. (1) and the actual data, and select the values that minimizes the associated errors. The result is given in Figure 6.

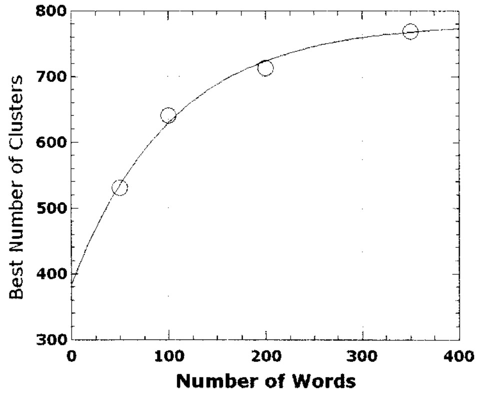

Figure 6. The curve-fitting result for the data in Figure 5 according to Eq. (1).

From the fit, we found that

$$N_0 = 383 \quad , \quad N_\infty = 781$$

This result implies that it would be desirable to consider about 8 and 16 variations for each phoneme in the speech recognition task in the limits of small and large vocabularies, respectively.

5. Conclusion

In order to find the optimal number of clusters for speaker-independent speech recognition, we introduced the neural network of adaptive resonance theory for continuous-valued parameters (ART2), which enables us to control the number of clusters via the vigilance parameter. By combining this network with the conventional LBG algorithm without bifurcation, the clustering was performed in two stages. From the speech recognition test, the best recognition error rate was shown to be improved by 0~0.9% depending on the vocabulary sizes. It was also revealed that the best number of clusters was found to be around 400 and 800 in the limits of small and large vocabularies, respectively. This result suggests that about 8 and 16 variations per phoneme might be desirable in the two liming cases of speech recognition.

References

[1] Rabiner, L. & Juang, B. (1993). *Fundamentals of Speech Recognition*, Prentice Hall, pp. 485-486.

[2] Deller, J. R., Proakis, J. G., & Hansen, J. H. L. (1993). *Discrete-Time Processing of Speech Signals, Macmillan*, pp. 115-119.

[3] Carpenter, G. A. & Grossberg, S. (1987). "ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns", *Applied Optics*, Vol. 26, pp. 4919-4930.

[4] Fausett, L. (1994). *Fundamentals of Neural Networks*, Prentice Hall, pp. 246-282.

[5] Haiyan, H. & Chengyi, W. (1992). "ART2-Based MLPs Neural Network for Speaker-Independent Recognition of Isolated Words", *11th IAPR International Conference on Pattern Recognition*, pp. 590-593.

[6] Kung, S. Y. (1993). *Digital Neural Networks*, Prentice Hall, pp. 80-85.

[7] Lee, C. Y., Nam, H. S., Jung, H. S., & Lee, C. B. (2005). "The Effect of Membership Concentration in FVQ/HMM for Speaker-independent Speech Recognition", *Speech Sciences*, Vol. 12, No. 4, pp. 7-15.

[8] Dehghan, M., Faez, K., Ahmadi, M., & Shridhar, M. (2001). "Unconstrained Farsi Handwritten Word Recognition Using Fuzzy Vector Quantization and Hidden Markov Models", *Pattern Recognition Letters*, Vol. 22, pp. 209-214.

[9] Levinson, S. E., Rabiner, L. R. & Sondhi, M. M. (1983). "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition", *Bell Systems Tech. J.*, Vol. 62, No. 4, pp. 1035-1074.

• **Lee, Chang-Young**
Division of Information System Engineering
Jurye San 69-1, Sasang, Pusan 617-716, Korea
Tel: +82-51-320-1719 Fax: +82-51-320-2389
E-mail: seewhy@dongseo.ac.kr