Integration of Acoustic Information in a Large Vocabulary Word Recognizer

V.N. Gupta, M. Lennig, and P. Mermelstein

BNR and INRS-Télécommunications

3 Place du Commerce

Montreal, Quebec, Canada H3E 1H6

Abstract

This paper proposes a new way of using vector quantization for improving recognition performance for a 60,000 word vocabulary speaker-trained isolated word recognizer using a phonemic Markov model approach to speech recognition. We show that we can effectively increase the codebook size by dividing the feature vector into two vectors of lower dimensionality, and then quantizing and training each vector separately. For a small codebook size, integration of the results of the two parameter vectors provides significant improvement in recognition performance as compared to the quantizing and training of the entire feature set together. Even for a codebook size as small as 64, the results obtained when using the new quantization procedure are quite close to those obtained when using Gaussian distribution of the parameter vectors.

I. Introduction

This paper proposes a new way of using vector quantization in a phoneme-based Markov model approach to speech recognition to improve the recognition performance when using a small codebook size. Several previous studies have shown that use of vector quantization with a small codebook size results in a significant degradation in recognition performance as compared to Gaussian distribution of parameters. Rabiner et. al (1983) obtain 2.9% digit error rate for isolated word recognition using hidden Markov models (HMM) and a codebook size of 64. The 2.9% digit error rate drops down to 0.7% (Rabiner et. al 1985) when a Gaussian distribution is used for the output probabilities. Bahl et. al (1979) report significant improvement in recognition performance for sentence recognition using continuous parameters. Poritz and Richter (1986) show reduction in isolated-word recognition error rate from 3.7% to 0.55% when vector quantized parameters are replaced by continuous parameters. In our experiments, vector quantization with a codebook size of 64 resulted in a drop in recognition performance[1] from 76% to 54% for the speaker-dependent large vocabulary recognizer. Detailed results for different codebook sizes are outlined in Table 1.

Parameters	# of Codes	% corr.	Av. rk.
Static&Dynamic	64	54%	4.3
Static&Dynamic	128	58%	4.2
Static&Dynamic	256	66%	3.3
Static&Dynamic	512	66%	3.5
Static&Dynamic	1024	71%	3.4
Static&Dynamic	(continuous)	76%	2.6
Static only	64	44%	6.8
Static only	256	52%	6.9
Static only	512	53%	6.3
Static only	(continuous)	69%	5.2

Table 1. *Percent correct* and *average rank of correct word* for isolated word recognition of a 60,000 word vocabulary using different codebook sizes. All words have equal prior probabilities. Test set size = 178 words

In this paper, we show that we can effectively provide performance comparable to a large codebook size by splitting the feature vector into two sets of vectors and coding the two sets independently using a small codebook size. The paper is organized in the following manner. Section II outlines the 60,000 word vocabulary recognizer. Section III discusses the splitting of a feature vector into two vectors for separate vector quantization, and discusses the different ways of integrating the two vectors to provide enhanced word recognition performance. Section IV compares the recognition results using vector quantization with those using continuous distribution of parameters.

[1] All recognition rates are the percent of words correctly recognized as the top choice (homophone confusions are not counted as errors). Test and training sets are disjoint. All words are equiprobable.

CH2396-0/87/0000-0697 $1.00

II. Overview of the 60,000 word vocabulary isolated word recognizer

Our 60,000 word vocabulary *speaker-trained* isolated word recognizer uses a phonemic Markov model approach to speech recognition. The 60,000 words correspond to the contents of Merriam Webster's Seventh Collegiate Dictionary (1965), together with the 1000 most frequent words in the Brown Corpus (Francis and Kucera 1979), and all of the words in the training and test sets used for recognition. The recognizer uses one Markov model per phoneme for the 44 phonemes used in recognition. The output of the recognition process is a sequence of lexically valid, most likely phoneme strings, together with their likelihoods. No language model is used. All the 60,000 words are considered a priori equiprobable.

The recognition is performed in four stages. The first stage segments the sentence into words using a loudness parameter. The second stage generates a number of hypotheses for the length of the word in number of syllables. The third stage is a fast matcher which computes the sequence of most likely phonetic strings for each hypothesized number of syllables of the word. The fast matcher uses a syllable network (Gupta et. al 1986) for scoring partial phoneme strings and a lexical tree for decoding only lexically valid phoneme strings. The branches of the lexical tree correspond to phonemes and the leaves of the tree correspond to words in the vocabulary. The fast matcher extends the most likely paths using the *stack algorithm* (Jelinek 1976), also known as the A^* *algorithm* in the artificial intelligence literature (Nilsson 1980). The matcher stops when the 100 most likely complete paths have been found. The fourth stage computes the maximum likelihood scores for the phoneme strings which were generated by the third stage, and re-orders the phoneme strings, based on these scores.

We have been evaluating the accuracy of the recognizer via two quantitative measures: One criterion is the percent of correct words as the top word choice (homophone confusions are not counted as errors). The other criterion is the average rank of the correct word in the list of ordered word hypotheses.

The recognition environment is as follows. Speech is recorded in a sound booth from one speaker. The training and test sets consist of sentences read from text which was selected randomly from magazines, books, newspaper articles, and office correspondence. Some additional training words are chosen to represent phonemes in consonant clusters and in CVC context where C stands for one of the consonants [p t k b d g r l]. All of the training and test words are read in isolation. The training set consists of approximately 1300 word tokens.

Markov models using vector quantization are trained in two steps. The first step trains the Markov models on approximately 800 word tokens which have been manually segmented into phones. Each phoneme model is trained by using the forward-backward algorithm (Jelinek 1976, Levinson et. al 1983) and the phone segments comprised by these word tokens. In the second step, we use 1300 word tokens without any phone segmentation. The entire word is processed by the forward-backward algorithm with the parameters obtained in the first step as initial parameters. The output probabilities are smoothed after each step. The structure of the phoneme models is kept fixed in all experiments. The topology corresponds to the *left-to-right* or *Bakis* model (Jelinek 1976). Each state has a self-loop, next state, and skip transition probability. The number of states for the consonants is between 3 and 5, and for the vowels it is between 6 and 10.

Markov models using Gaussian probability distribution use only the 800 word tokens which have been segmented into phones for training. The second step of training is not performed. Each transition in a phoneme model has an associated mean vector and covariance matrix. While the mean vector is unique to the particular transition, the covariance matrix is pooled over all the transitions within a particular phoneme model to generate a common covariance matrix for that phoneme. Generation of one covariance matrix for each phoneme model results in a significant reduction in computations and in elimination of singular covariance matrices for some transitions during training.

III. Performance improvement by the integration of split and separately quantized feature vectors

We first discuss our acoustic representation of speech and then describe the splitting of feature vectors before vector quantization. Each frame of speech has seven *static* mel-frequency cepstral coefficients $(C_1, \ldots, C_7)$, and a perceptually weighted energy coefficient, C_0 (Davis and Mermelstein 1980) associated with it. Eight *dynamic* parameters $(\Delta C_0, \ldots, \Delta C_7)$ measure the direction and the magnitude of spectral change over 50 milliseconds.

First we outline why we split the feature vector into two sets of vectors. In training the HMM models using vector quantization, we estimate the code output probability for each code in the codebook for each transition. With increase in codebook size, we require more training data to obtain reliable estimates of the code output probabilities. Because only a limited amount of training data is generally available, there will be an optimal value for the codebook size beyond which the recognition performance will drop due to insufficient training data. We can overcome this problem by dividing the feature vector into two sets, quantizing each set separately, and then computing the joint probabilities of the two codes – one from each codebook. If the two codes are independent of each other, then the joint probability will be the product of the probability of each code. Implicitly, we have increased the codebook size to the product of the two codebook sizes while still having to estimate probabilities of only a small

number of codes corresponding to the size of each codebook. For example, the size for each codebook could be 64, while the combined codebook size would be 4096. To get the maximum benefit, the features used for the two codebooks should be independent and they should provide similar recognition performance when used independently. This method of integrating the codes just described would be called *frame-level* integration.

How can we split the 15-dimensional feature vector consisting of the seven static and eight dynamic cepstral parameters? The static and dynamic parameters were found experimentally to be largely uncorrelated. By quantizing static and dynamic parameters independently to 64 codes and by computing the joint code output probability of the two codes as the product of the output probability of each code, we improved the recognition performance from 58% to 65% correct words as the top word choice. The performance obtained using a total of 128 codes is comparable to the performance obtained using a codebook size of 256 and quantizing all the 15 parameters together.

We can delay the integration of the split vectors to the phone or the word level. The rationale for the delayed integration can be explained in the context of Fujimura's iceberg model (1984). Fujimura showed that articulator movements for the place-specified consonants is relatively invariant, while the relative timings of these articulators may vary. Similar effects may be observed in different parameters for the acoutic manifestation of these movements. Delayed integration of the parameters allows for flexibility in the temporal alignment of the acoustic parameters.

In phone-level integration, we train separate Markov models for each phoneme for each of the split vectors. During recognition, we compute the joint probability that the two models for the same phoneme generate the observed sequence of codes assuming each model to be independent of the other model. We will explain the computation of the joint probability in more detail for the splitting of the feature vector into static and dynamic parameters.

We delay the integration of the static and dynamic feature vectors to the phone level by training separate HMM models for static and dynamic parameters and then combining the likelihoods obtained using each model separately at phone boundaries. The likelihoods are combined by computing the joint probability of the static and dynamic HMM models of the phoneme generating the observed partial sequence of codes. Given a phoneme string p(1:n), and the observed static and dynamic sequence of codes S(1:T) and D(1:T), we compute the likelihood as

$$\sum_{t} \prod_{i=1}^{n} (P(S(t_i : t_{i+1}) | M_s(p_i)) \cdot P(D(t_i : t_{i+1}) | M_d(p_i)))^{1/2},$$

where the sum is over all possible segmentations of the input sequence into n phones. $M_s(p_i)$ represents the Markov model for phoneme p_i using static parameters. $M_d(p_i)$ represents Markov model for phoneme p_i using dynamic parameters. Phone-level integration of the feature vectors improved the recognition to 67% from 65% for frame-level integration. The average rank of the correct word improved from 3.9 for frame-level integration to 3.6 for phone-level integration.

Another possibility of integrating the separately quantized feature vectors is at the word level. As in phone-level integration, we compute separate static and dynamic HMM models for each phoneme. The likelihood of the test word is computed as the joint probability of observing the input sequence using a sequence of static and dynamic HMM phoneme models corresponding to the word. In other words, the likelihood of the test word is

$$P(S(1:T)|M_s(p(1:n))) \cdot P(D(1:T)|M_s(p(1:n))).$$

Word-level integration gives a recognition rate of 66% (average rank = 3.7). The results for feature parameter integration using different codebook sizes are summarized in Table 2. Integration of feature parameters provides similar performance for all codebook sizes, and the performance is comparable to that obtained for the largest codebook size using vector quantization of all feature parameters together as shown in Table 1.

Integration	# of Codes	% correct	Av. rank
Frame-level	64	65%	3.9
Phone-level	64	67%	3.6
Word-level	64	66%	3.7
Word-level	256	62%	3.7
Frame-level	256	67%	8.0
Word-level	512	66%	3.1

Table 2. *Percent correct* and *average rank of correct word* for isolated word recognition of a 60,000 word vocabulary using different integration techniques and codebook sizes. Test set size = 178 words

Another question of interest is whether we can get similar performance improvements when we integrate another split feature vector, for example the static parameters. We split the static parameters into the parameter vectors (C_1, C_2) and $(C_3, \ldots, C_7)$ and obtained significant improvement by using separate codebooks of size 64 for the two sets of parameters. The word-level integration gives 62% recognition and an average rank of 4.2 for the correct word. Quantization of static parameters using a codebook size of 512 gives only 53% recognition with an average rank of 6.3.

Integration experiments at small codebook size using the entire feature set or only the static parameters show that we can get performance comparable to the largest codebook size when the entire feature set is quantized together.

IV. Markov models using Gaussian distribution of feature vectors

To compare the results using vector quantization and continuous distribution of parameters, we computed the recognition performance using Gaussian distributions for two different feature sets. These sets are static parameters only, and the static and dynamic parameters together. The results are summarized in Table 3. Use of 64 codes for vector quantization with split cepstral parameters gives 62% recognition while the Gaussian distribution of cepstral parameters gives 69% recognition. The average rank of 4.2 for the correct words using vector quantization for the cepstral parameters is better than the average rank of 5.2 using Gaussian distribution for the cepstral parameters.

Integration of split features provides significant improvement in recognition performance for small codebook sizes. The results are quite close to the results obtained using Gaussian distribution of output probabilities. Similar results are obtained for another speaker as shown in Table 4. The training and test conditions for the new speaker are same as those of the first speaker.

Condition	% Corr.	Av. rank
Static parameters (Gaussian)	69%	5.2
$(C_1, C_2) * (C_3, \ldots, C_7)$ {VQ}	62%	4.2
Static & dynamic (Gaussian)	76%	2.6
$(C_1, \ldots, C_7) * (\Delta C_0, \ldots, \Delta C_7)$ {VQ}	66%	3.7

Table 3. Comparison of recognition results for different HMM models. Word level integration is denoted by $*$. Test set size = 178 words.

Condition	% Corr.	Av. rank
Static & dynamic(Gaussian)	79%	1.6
Frame-level integration(VQ)	70%	2.3
Word-level integration(VQ)	62%	3.2

Table 4. Comparison of recognition results for different HMM models for a new speaker. Test set size = 180 words.

V. Conclusions

We have shown that integration at the frame or word level of split feature vectors improves the recognition performance for a Markov model based 60,000 word vocabulary recognizer. The recognition performance for the integration of feature parameters employing a codebook size of 64 is quite close to the results obtained when we use Gaussian distributions of feature parameters. Thus significant computational savings are obtainable without degrading recognition.

References

Bahl, L.R., Bakis, R., Cohen, P.S., Cole, A.G., Jelinek, F., Lewis, B.L., and Mercer, R.L. (1979). "Recognition results with several experimental acoustic processors" *Proc. 1979 IEEE Int'l. Conf. Acoustics, Speech, and Signal Processing,* pp. 249-251.

Davis, S.B. and Mermelstein, P. (1980). "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-28, no. 4, pp. 357–365.

Francis, W.N. and Kucera, H. (1979). *Manual of information to accompany a standard sample of present-day edited American English, for use with digital computers,* Department of Linguistics, Brown University.

Fujimura, O. (1984). "Relative Invariance of Articulatory Movements – An Iceberg Model," in *Invariance and Variability of Speech Processes,* J. S. Perkell et. al, editors, Lawrence Erlbaum Associates, Inc.

Gupta, V., Lennig, M., Marcus J. and Mermelstein, P. (1986). "Syllable network for phonemic decoding of speech," *Proc. Montreal Symp. Speech Recog.,* pp. 45–46.

Jelinek, F. (1976). "Continuous speech recognition by statistical methods," *Proceedings of the IEEE,* vol. 64, no. 4, pp. 532–556.

Levinson, S.E., Rabiner, L.R. and Sondhi, M.M. (1983). "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *B.S.T.J.,* vol. 64, part 1, pp. 1211–1234.

G. & C. Merriam Co. (1965). *Webster's Seventh New Collegiate Dictionary.*

Nilsson, N.J. (1980). *Principles of Artificial Intelligence,* Tioga Publishing Co., CA.

Poritz, A.B. and Richter, A.G. (1986). "On hidden Markov models in isolated word recognition," *Proc. 1986 IEEE Int'l. Conf. Acoustics, Speech, and Signal Processing,* pp. 705–708.

Rabiner, L.R., Levinson, S.E. and Sondhi, M.M. (1983). "On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition," *B.S.T.J.,* vol. 62, no. 4, pp. 1075–1105.

Rabiner, L.R., Juang, B.-H., Levinson, S.E. and Sondhi, M.M. (1985). "Recognition of isolated digits using hidden Markov models with continuous mixture densities," *B.S.T.J.,* vol. 64, no. 6, pp. 1211–1234.