Speech Recognition Using Dynamical Model of Speech Production

Ken-ichi Iso*

September 1992
CMU-CS-92-187

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We propose a speech recognition method based on the dynamical model of speech production. The model consists of an articulator and its control command sequences. The latter has linguistic information of speech and the former has the articulatory information which determines transformation from linguistic intentions to speech signals. This separation makes our speech recognition model more controllable. It provides new approaches to speaker adaptation and to coarticulation modeling. The effectiveness of the proposed model was examined by speaker-dependent letter recognition experiments.

*Visiting Scientist from C & C Information Technology Research Laboratories, NEC Corporation, 4-1-1 Miyazaki, Miyamae-ku, Kawasaki 216, JAPAN

Keywords: speech recognition, neural networks, nonlinear prediction, hidden markov models, speaker adaptation

Contents

List of Figures

List of Tables

1 Introduction

In Hidden Markov Models (HMM) for speech recognition, the speech signal is modeled by a sequence of independent states. Each state represents a piece-wise stationary segment of speech. However the actual speech signal is highly dynamical and varies continuously. Coarticulation phenomena arise from the dynamical and continuous nature of speech signals. In modeling speech signal using HMMs, many states are used to approximate the speech signal piece-wise constantly. Furthermore dynamic features like delta-cepstrum are included in the feature vector representation [1]. To deal with coarticulation phenomena, context dependent phone model like generalized triphones is used [2]. These improvements increase enormously the number of model parameters to be estimated from training data.

It appears that the HMM approach (piece-wise static models with dynamic feature vectors) is limited and a true dynamical model of speech signals must be explored. We propose a novel speech recognition method based on the dynamical model of speech production.

2 Model

The proposed model consists of a speech articulator model and control commands for the articulator. Each speech unit (word or subword) has a control command sequence. The control command sequence for larger speech segments (sentence or word) is obtained as a concatenation of the ones for the basic speech units.

Figure 1 shows the model architecture. Input speech is represented by a feature vector sequence (length T).

$$\mathbf{a}_1, \cdots, \mathbf{a}_t, \cdots, \mathbf{a}_T. \tag{1}$$

where each feature vector has P components (P dimensional vector). We use 16 FFT mel spectral coefficients for feature vector ($P = 16$). The control command sequence for sentence “ABA” is a concatenation of the ones for the basic speech unit “A” and “B”,

$$\mathbf{c}_1, \cdots, \mathbf{c}_n, \cdots, \mathbf{c}_N. \tag{2}$$

where each control command is a Q dimensional real-valued vector and N is a length of the control command sequence. It drives an articulator to produce speech. The articulator is represented by a nonlinear vector function $\mathbf{f}$,

$$\hat{\mathbf{a}}_{t,n} = \mathbf{f}(\mathbf{a}_{t-1}, \mathbf{c}_n). \tag{3}$$

This is a nonlinear predictor with control input $\mathbf{c}_n$ which modulates the mapping generated by the function $\mathbf{f}$. The function $\mathbf{f}$ can be realized by a multilayer perceptron (MLP) with $P + Q$ input units and P output units [3, 4, 5, 6](Figure 2). The distance between input speech feature vector $\mathbf{a}_t$ and predicted feature vector $\hat{\mathbf{a}}_{t,n}$ is defined by prediction error,

$$d(t, n) = \|\mathbf{a}_t - \hat{\mathbf{a}}_{t,n}\|^2. \tag{4}$$

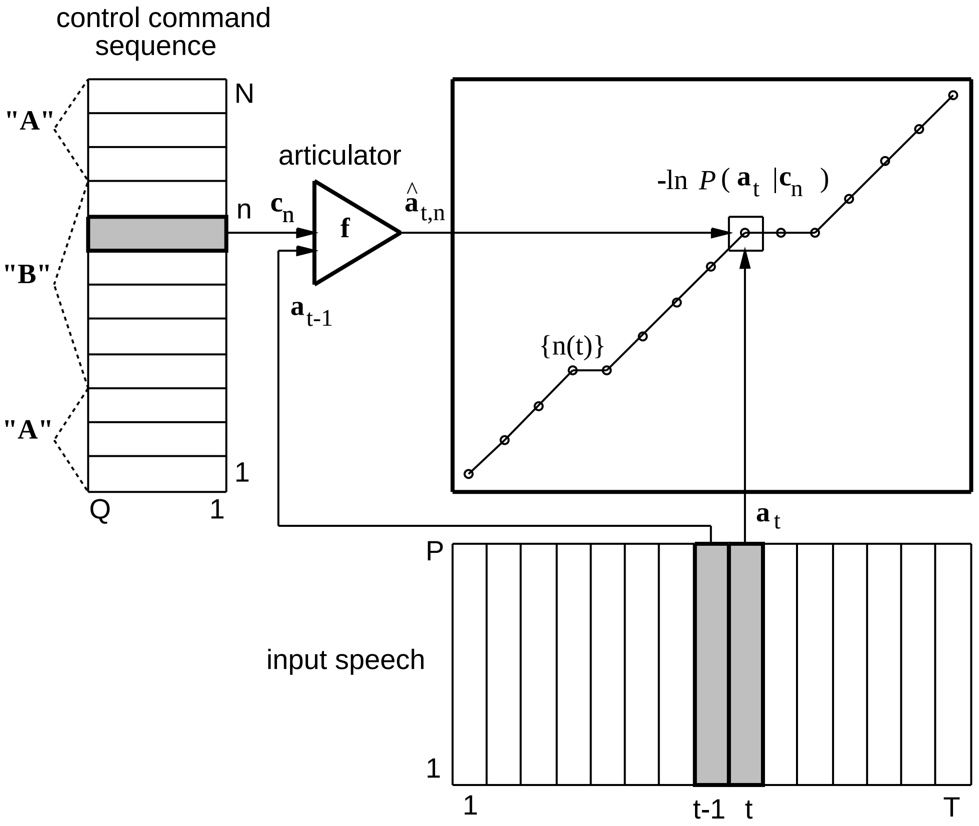

Figure 1: Model Architecture

This can be generalized to the gaussian probability of producing the speech feature vector $\mathbf{a}_t$ given the control command $\mathbf{c}_n$, $P(\mathbf{a}_t|\mathbf{c}_n)$,

$$-\ln P(\mathbf{a}_t|\mathbf{c}_n) = \frac{1}{2}(\mathbf{a}_t - \hat{\mathbf{a}}_{t,n})^T \Sigma_n^{-1} (\mathbf{a}_t - \hat{\mathbf{a}}_{t,n}) + \frac{1}{2}\ln(2\pi)^P |\Sigma_n|, \tag{5}$$

where Σ_n is $P \times P$ covariance matrix for the control command $\mathbf{c}_n$. In the following discussion, we use diagonal covariance matrix for computational simplicity,

$$(\Sigma_n)_{p,q} = \delta_{p,q}\sigma_{n,p}^2, \tag{6}$$

where $(\Sigma_n)_{p,q}$ is a (p,q) component of matrix Σ_n, $\delta_{p,q}$ is Kronecker's delta and $\sigma_{n,p}^2$ is the p-th diagonal component. Then equation 5 can be written as

$$-\ln P(\mathbf{a}_t|\mathbf{c}_n) = \frac{1}{2}\sum_{p=1}^{P}\{\frac{(a_{t,p} - \hat{a}_{t,n,p})^2}{\sigma_{n,p}^2} + \ln(2\pi)\sigma_{n,p}^2\}, \tag{7}$$

where $a_{t,p}$ is the p-th component of $\mathbf{a}_t$ and $\hat{a}_{t,n,p}$ is the p-th component of $\hat{\mathbf{a}}_{t,n}$. With this definition, we can compute the probability of acoustic observation given the control command sequence,

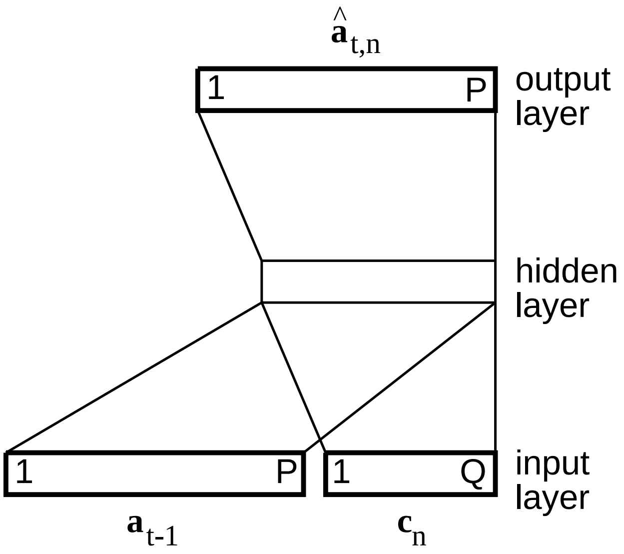

Figure 2: Articulator Model by Multilayer Perceptron

$$P(\mathbf{a}_1, \cdots, \mathbf{a}_T | \mathbf{c}_1, \cdots, \mathbf{c}_N) = \max_{\{n(t)\}} \prod_{t=1}^{T} P(\mathbf{a}_t | \mathbf{c}_{n(t)}), \tag{8}$$

where function $n(t)$ determines time-alignment between input speech feature vector sequence and control command sequence. The optimal time-alignment $\{n^*(t)\}$ which gives maximum probability is determined by dynamic programming (DP). Using this probability as a score, we can perform speech recognition.

This formulation of time-alignment is formally similar to Viterbi algorithm for HMM [7]. By the analogy with HMM, the max operation in equation 8 can be replaced by summation over all possible time-alignment, which corresponds to Forward algorithm for HMM. In the following discussion, however, we use Viterbi time-alignment for simplicity.

3 Training Algorithms

The proposed model has a set of unknown parameters to be estimated from training data. One is given by the control command sequences for speech units (words or subwords) and the other by the articulatory parameters which are implemented as MLP weight coefficients. The training algorithm is formulated as an optimization problem based on the maximum likelihood (ML) criterion or maximum a posteriori probability (MAP) criterion. The general discussion about the ML and MAP training criteria is given by [8].

The objective function for maximum likelihood (ML) training is an average of the logarithmic probabilities defined by equation 8 for all the training utterances,

$$E_{ML} = \frac{1}{M} \sum_{m=1}^{M} \ln P(\mathbf{a}_1^{(m)}, \cdots, \mathbf{a}_{T_m}^{(m)} | \mathbf{c}_1^{(m)}, \cdots, \mathbf{c}_{N_m}^{(m)})$$

$$= \frac{1}{M}\sum_{m=1}^{M} \ln P(A_m|k_m). \tag{9}$$

where the subscript m represents the m-th training utterance and M is the total number of training utterances. To simplify notations, A_m for an observation sequence and k_m for a control command sequence are used. The control command sequence for the m-th training utterance, $k_m = \{\mathbf{c}_1^{(m)}, \cdots, \mathbf{c}_{N_m}^{(m)}\}$, is uniquely determined by its linguistic transcription (concatenation of the basic units' control command sequences). The ML training can be performed by gradient ascent iterative maximization of this objective function. The training rule for parameter λ (control command, covariance matrix or weight coefficient) is given as

$$\begin{aligned} \delta\lambda &= \epsilon\frac{\partial E_{ML}}{\partial\lambda} \\ &= \epsilon\frac{1}{M}\sum_{m=1}^{M}\frac{\partial \ln P(A_m|k_m)}{\partial\lambda}, \end{aligned} \tag{10}$$

where $\delta\lambda$ is a correction for parameter λ at each iteration and ϵ is small positive constant. The differentiation in the above equation requires the differentiation of DP operation (defined by equation 8) which is inherently not differentiable. There are at least two methods to deal with this problem. One method is to fix time-alignment $\{n(t)\}$ by the optimal one $\{n^*(t)\}$ at each iteration. In this method, the optimal time-alignment by DP must be re-calculated after every update of parameters by the above equation. Convergence proof of this combined optimization of gradient ascent and DP was given in [4]. Another method is to replace non-differentiable DP operation in equation 8 by differentiable one, which is a summation over all possible time-alignments. In the experiments, we use the former method for computational simplicity.

To improve the performance of our proposed model, we have also introduced another discriminative training criterion called maximum a posteriori (MAP) training. The objective function for MAP training is an average of the logarithmic posterior probabilities for all training utterances,

$$\begin{aligned} E_{MAP} &= \frac{1}{M}\sum_{m=1}^{M}\ln P(\mathbf{c}_1^{(m)}, \cdots, \mathbf{c}_{N_m}^{(m)}|\mathbf{a}_1^{(m)}, \cdots, \mathbf{a}_{T_m}^{(m)}) \\ &= \frac{1}{M}\sum_{m=1}^{M}\ln P(k_m|A_m). \end{aligned} \tag{11}$$

The posterior probability is computed from the probability defined by equation 8 using Bayes' rule,

$$\begin{aligned} P(k_m|A_m) &= \frac{P(A_m|k_m)P(k_m)}{P(A_m)} \\ &= \frac{P(A_m|k_m)P(k_m)}{\sum_{k=1}^{K} P(A_m|k)P(k)}, \end{aligned} \tag{12}$$

where $P(A_m|k)$ is the probability of producing the m-th training utterance given the k-th linguistic transcription. For example, if training utterance is spoken as an isolated word, k represents the control command sequence for the k-th word in the recognition dictionary and K is the total number of words. For isolated word recognition, we can assume equal prior probability $P(k)$ for every word in the vocabulary. Then the objective function for MAP training is

$$\begin{aligned} E_{MAP} &= \frac{1}{M}\sum_{m=1}^{M} \ln \frac{P(A_m|k_m)}{\sum_{k=1}^{K} P(A_m|k)} \\ &= \frac{1}{M}\sum_{m=1}^{M} (\ln P(A_m|k_m) - \ln \sum_{k=1}^{K} P(A_m|k)). \end{aligned} \tag{13}$$

This objective function can be also maximized by gradient ascent method. Then MAP training rule for parameter λ (control command, covariance matrix or weight coefficient) is given as

$$\begin{aligned} \delta\lambda &= \epsilon\frac{\partial E_{MAP}}{\partial \lambda} \\ &= \epsilon\frac{1}{M}\sum_{m=1}^{M}(\frac{\partial \ln P(A_m|k_m)}{\partial\lambda} - \sum_{l=1}^{K}\alpha_l^m \frac{\partial \ln P(A_m|l)}{\partial\lambda}), \end{aligned} \tag{14}$$

$$\alpha_l^m = \frac{P(A_m|l)}{\sum_{k=1}^{K} P(A_m|k)}. \tag{15}$$

where α_l^m is defined as a weighting factor. The first term in equation 14 is the same as the ML training rule. We call this as the *positive* training term because it increase the probability of producing the m-th training utterance A_m by the correct class model (control command sequence) k_m. On the other hand, the second term (*negative* training term) has the opposite sign, which decreases the probability of producing A_m by the incorrect class model l. The α_l^m determines the amount of *negative* training for incorrect class model l based on its similarity to the correct model.

4 Experimental Evaluation

In order to examine the validity of the proposed model, speaker-dependent English spoken letter recognition experiments have been carried out. The vocabulary consists of the 26 letters of the English alphabet. The database consists of 1000 connected strings of letters, some of which correspond to grammatical words and proper names, others of which are simply random. There is an average of five letters per string. The strings are labeled with an automatic procedure using the discrete-HMM based SPHINX system in a forced-alignment mode [2]. The SPHINX phone models used in the forced-alignment were trained on a

	training data	test data
ML training without covariance matrices	88.3%	87.1%
+ diagonal covariance matrices	90.6%	89.8%
+ MAP discriminative training	92.8%	91.2%

Table 1: Experimental Results

speaker-independent vocabulary-independent task. The speaker is a native male American. As a feature vector, 16 FFT mel spectral coefficients are calculated at a 10 msec frame rate.

We use the phoneme as the basic recognition unit. The total number of phonemes is 26 and each letter is represented by a concatenation of phonemes. For example, letter "H" is a concatenation of phonemes /EY/ and /CH/. Each phoneme has a control command sequence which consists of three control command vectors $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$. A control command vector is an 8 dimensional vector $(Q = 8)$ and it has a diagonal covariance matrix which has 16 independent diagonal components. The articulator model is a three-layer perceptron with one hidden layer. It has 24 input units, 8 for the control command and 16 for the speech feature vector (at time $t-1$). The number of hidden units is 24 and the number of output units is 16. Total number of model parameters are 2872, where 1000 is for articulator, 624 is for control commands for 26 phonemes, and 1248 for covariance matrices.

During training and testing, the boundaries between letters in connected letter utterances are known by label information. 500 utterances are used for training and another 500 are used for testing. Each data set of 500 utterances has about 2500 letters. Table 1 shows the recognition results.

As the baseline experiment, all covariance matrices for control commands are fixed as an identity matrix. By introducing the diagonal covariance matrices, the error rate for the test data set was reduced by 17%. The discriminative training by the MAP criteria further improves the performances. Experimental comparison with other methods and applications to other tasks (speaker-independent large vocabulary recognition) will be examined in future.

5 Discussion and Conclusion

The proposed model automatically extracts the control command sequences for speech signals by training. The linguistic information of speech signals should be included in the control command sequences. Furthermore the articulator model does not have any linguistic dependency because it is common to all speech units. It only has the articulatory information which determines transformation from linguistic intentions to speech signals. This separation of linguistic and articulatory information makes our speech recognition model more controllable.

Several dynamical models for speech recognition based on the predictive neural networks have been proposed so far [4, 5, 6, 9]. The Hidden Control Neural Network (HCNN) [5, 6] has a control input to the predictor, which is similar to our model. Their control inputs, however, are fixed to manually determined binary values. Then HCNN cannot separate linguistic and articulatory informations.

One useful application of the separation in our model is speaker adaptation. Using the utterances of many speakers, we can train speaker-independent control command sequences and speaker-dependent articulator models. For new speakers, only the articulator model should be trained. Because of the linguistic independence of the articulator model, we can greatly reduce the amount of adaptation data. This is a new approach to speaker adaptation.

Another application is the coarticulation modeling. From the physiological consideration of speech production process [10], it was suggested that the control command sequence should be smooth over time. We also can introduce this smoothness constraint into our objective function of ML training (equation 9) as additional term, E_{smooth}.

$$E_{smooth} = \frac{1}{2M} \sum_{m=1}^{M} \sum_{n=2}^{N_m} \|\mathbf{c}_n^{(m)} - \mathbf{c}_{n-1}^{(m)}\|^2. \tag{16}$$

This constraint works as a regularization term in the objective function. We expect the smooth concatenation of speech units at control command level may model the coarticulation phenomena and improve generalization performance.

Acknowledgements

The author would like to thank Dr. Alex Waibel for his continuous encouragement and suggestions and members of the neural network speech group at Carnegie Mellon University for their help. The author also acknowledges the support of NEC Corporation.

References

[1] S. Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-34:52–59, February 1986.

[2] Kai-Fu Lee, Hsiao-Wuen Hon, and Raj Reddy. An overview of the sphinx speech recognition system. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(1):35–45, January 1990.

[3] Ken ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.

[4] Ken ichi Iso and Takao Watanabe. Speaker-independent word recognition using a neural prediction model. In *IEEE Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 441–444. IEEE, April 1990.

[5] Naftali Tishby. A dynamical systems approach to speech processing. In *IEEE Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 365–368. IEEE, April 1990.

[6] Esther Levin. Word recognition using hidden control neural architecture. In *IEEE Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 433–436. IEEE, April 1990.

[7] L. R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, pages 4–16, January 1986.

[8] Herve Bourlard and Christian J. Wellekens. Links between markov models and multi-layer perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1167–1178, December 1990.

[9] Joe Tebelskis and Alex Waibel. Large vocabulary recognition using linked predictive neural networks. In *IEEE Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 437–440. IEEE, April 1990.

[10] Makoto Hirayama, Eric Vatikiotis-Bateson, Mitsuo Kawato, and Michael I. Jordan. Forward dynamics modeling of speech motor control using physiological data. In John E. Moody, Steven J. Hanson, and Richard P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4, pages 191–198. Morgan Kaufmann, 1991.