# MIXTURE GAUSSIAN HMM-TRAJCTORY METHOD USING LIKELIHOOD COMPENSATION

*Yasuhiro Minami*
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto-fu, Japan
minami@cslab.kecl.ntt.co.jp

## ABSTRACT

We propose a new speech recognition method (HMM-trajectory method) that generates a speech trajectory from HMMs by maximizing their likelihood while accounting for the relationship between the MFCCs and dynamic MFCCs. One major advantage of this method is that this relationship, ignored in conventional speech recognition, is directly used in the speech recognition phase. This paper improves the recognition performance of the HMM-trajectory method for dealing with mixture Gaussian distributions. While the HMM-trajectory method chooses the Gaussian distribution sequence of the HMM states by selecting the best Gaussian distribution in the state during Viterbi decoding and calculating HMM trajectory likelihood along with the sequence, the proposed method compensates for HMM trajectory likelihood using ordinary HMM likelihood. In speaker-independent speech recognition experiments, the proposed method reduced the error rate about 10% for the task compared with HMMs, proving its effectiveness for Gaussian mixture components.

***Index Terms—*** HMM, Trajectory

## 1. INTRODUCTION

Since HMMs model the acoustic feature vector sequence as a piecewise stationary process, the probability of a given acoustic feature is independent of the sequence of acoustic features preceding and following the current feature. This means that statistics in a HMM state are stationary, and thus HMMs cannot treat the time-dependent characteristics of speech within that state. This is a widely recognized drawback of speech recognition using HMMs, even though several methods have introduced time dependency to overcome this drawback [1][2][3][4][5]. To introduce the speech dynamics ignored by the HMM assumption in speech recognition, we propose a new method (hereafter the HMM-trajectory method) that employs smoothed speech feature trajectory generated from HMM statistics [6][7][8][9]. The HMM-trajectory method uses the relationship between the static and dynamic features (delta features and delta-delta features), which is ignored in the conventional speech recognition phase despite their important speech dynamics information. The HMM-trajectory method generates a smooth feature vector trajectory by a Kalman smoother that maximizes HMM likelihood while simultaneously considering the relationships [9].

We also extended the method for dealing with mixture Gaussian distributions [8]. The method chooses the sequence of Gaussian distributions by selecting the best Gaussian distribution in the state during ordinary HMM Viterbi decoding. Although in that paper, speaker-independent speech recognition experiments reduced error rates, when we performed an additional experiment with a large amount of data, the error rate improvement decreased. Perhaps our previous method only considered the best Gaussian distribution sequences and ignored the other Gaussian distributions in a state. In this paper, we propose a new method that improves this drawback, explain an overview of the HMM-trajectory method, and formulate Gaussian mixture distributions in it. Then compensated likelihood, which combines HMM trajectory and ordinary HMM likelihoods, is introduced into the HMM-trajectory method.

## 2. OVERVIEW OF HMM-TRAJECTORY METHOD

This section describes an overview and the theoretical aspects of the HMM-trajectory method based on [6][7][9] as well as HMM and HMM trajectory likelihoods. To simplify the equations, this section treats one-dimensional speech features and a single mixture HMM.

### 2.1. Definition and assumptions

Since the HMM-trajectory method uses almost identical parameters as in a HMM, the basic variables used in HMM are defined as:

$y_t$ : Static feature vector (one-dimension) at frame t

$S_t$ : HMM state in frame t

$Y_t = [y_t, \Delta y_t, \Delta\Delta y_t]'$ : Static and dynamic features at frame t

$Y^t = \left[ y_{t+2} \;\; y_{t+1} \;\; y_t \;\; y_{t-1} \;\; y_{t-2} \right]'$ : Five frame static feature vector

$Y_t = CY^t$ : Matrix equation to generate speech recognition features from a five frame static feature vector

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1/5 & 1/10 & 0 & -1/10 & -1/5 \\ 1/14 & -1/28 & -1/14 & -1/28 & 1/14 \end{bmatrix}$$ : An operator that generates a set of static and dynamic features: $Y_t$ from $Y^t$ (This matrix is one realization for this purpose),

$Y_{1:T}$ : $Y_1, Y_2, ..., Y_T$ ,

where T is the final frame of the data and ' means the transpose of the matrix.

After a HMM is trained, the following HMM probabilities are obtained:

$P(Y_t \mid S_t)$ : Emission probability at state $S_t$

$P(S_t \mid S_{t-1})$ : Transition probability from $S_{t-1}$ to $S_t$

### 2.2. HMM likelihood

Viterbi likelihood calculation is widely used in speech recognition as an approximation of trellis likelihood. The likelihood calculation for a HMM is formulated by

$$\begin{aligned} P(Y_{1:T}) &\approx \max_{S_{1:T}} P(Y_{1:T}, S_{1:T}) = \max_{S_{1:T}} \left[ P(Y_{1:T} \mid S_{1:T}) P(S_{1:T}) \right] \\ &= \max_{S_{1:T}} \left[ P(S_1) P(Y_1 \mid S_1) \prod_{2}^{T} P(Y_t \mid S_t) P(S_t \mid S_{t-1}) \right], \end{aligned} \quad (1)$$

where $P(Y_t \mid S_t)$ is emission probability that can be calculated by

$$P(Y_t \mid S_t) = N(y_t; \mu_{S_t}, \rho_{S_t}{}^2) N(\Delta y_t; \Delta\mu_{S_t}, \Delta\rho_{S_t}{}^2) N(\Delta\Delta y_t; \Delta\Delta\mu_{S_t}, \Delta\Delta\rho_{S_t}{}^2) . \quad (2)$$

$N(y_t;\mu_{S_t},\rho_{S_t}{}^2)$ is the Gaussian distribution of $y_t$ whose mean and variance are $\mu_{S_t}$ $\rho_{S_t}{}^2$. $\mu_{S_t},\Delta\mu_{S_t},\Delta\Delta\mu_{S_t}$ are the HMM mean values at $S_t$. $\sigma_{S_t}{}^2,\Delta\sigma_{S_t}{}^2,\Delta\Delta\sigma_{S_t}{}^2$ are the HMM variance values at $S_t$. To discuss the relationship between HMM and HMM trajectory likelihoods, we modify $P(Y_t \mid S_t)$ into

$$\int_{X_t} P(Y_t \mid X_t, S_t) P_\delta(X_t \mid S_t), \tag{3}$$

where $X_t$ is a hidden variable and $P_\delta(X_t \mid S_t)$ is the multiplication of the delta functions defined as:

$$P_\delta(X_t \mid S_t) = \delta(x_t - \mu_{S_t})\delta(\Delta x_t - \Delta\mu_{S_t})\delta(\Delta\Delta x_t - \Delta\Delta\mu_{S_t}). \tag{4}$$

$P(Y_t \mid X_t, S_t)$ is defined as

$$P(Y_t \mid X_t, S_t) = N(y_t; x_t, \rho_{S_t}{}^2)N(\Delta y_t; \Delta x_t, \Delta\rho_{S_t}{}^2)N(\Delta\Delta y_t; \Delta\Delta x_t, \Delta\Delta\rho_{S_t}{}^2). \tag{5}$$

Using this formulation, $P(Y_{1:T} \mid S_{1:T})$ can be reformulated by

$$\begin{aligned} P(Y_{1:T} \mid S_{1:T}) &= \prod_t \int_{X_t} P(Y_t \mid X_t, S_t) P_\delta(X_t \mid S_t) \\ &= \int_{X_{1:T}} \prod_t P(Y_t \mid X_t, S_t) \prod_t P_\delta(X_t \mid S_t) \\ &= \int_{X_{1:T}} P(Y_{1:T} \mid X_{1:T}, S_{1:T}) P_\delta(X_{1:T} \mid S_{1:T}). \end{aligned} \tag{6}$$

This equation denotes that $P_\delta(X_{1:T} \mid S_{1:T})$ selects the HMM mean sequence from $X_{1:T}$ using the HMM assumption that each state generates the time independent mean value, and then $P(Y_{1:T} \mid X_{1:T}, S_{1:T})$ calculates the likelihood using the selected mean sequence.

### 2.3. HMM trajectory likelihood

As shown in 2.2., in HMM likelihood calculation, $P_\delta(X_{1:T} \mid S_{1:T})$ selects the mean values from $X_{1:T}$ by assuming that they don't change in a state. If we use this assumption, for example, the mean sequences have a contradiction: the mean values stay at the same value while delta mean values are not zero. We believe that this contradiction reduces recognition accuracy. To avoid this contradiction, we introduce the following constraints for the hidden states:

$$X^t = \left[x_{t+2}\ x_{t+1}\ x_t\ x_{t-1}\ x_{t-2}\right]', \tag{7}$$

$$X_t = [x_t, \Delta x_t, \Delta\Delta x_t]', \tag{8}$$

$$X_t = CX^t. \tag{9}$$

These constraints denote that the means of delta features and delta-delta features are generated from the sequence of the static feature means; this assumption is quite natural considering the definition of delta and delta-delta features. Introducing the above constraints dynamically changes the mean values. To obtain the hidden states,

$$\hat{x}_{-1:T+2}(S_{1:T}) = \arg\max_{x_{-1:T+2}} P(X_{1:T} \mid S_{1:T}) \tag{10}$$

is performed under constraints (7), (8), and (9), where $P(X_{1:T} \mid S_{1:T})$ is the HMM probability of $X_{1:T}$. This calculation was originally used for speech synthesis [10]. We introduced another formulation that uses a Kalman filter [11], which is shown in the appendix (see also [9]). We call $\hat{x}_{-1:T+2}(S_{1:T})$ the trajectory and define a likelihood function along the state sequence of $S_{1:T}$ as follows:

$$\hat{P}(Y_{1:T} \mid S_{1:T}) = \int_{X_{1:T}} \hat{P}(Y_{1:T} \mid X_{1:T}, S_{1:T}) \hat{P}_\delta(X_{1:T} \mid S_{1:T}), \tag{11}$$

where we define $\hat{P}_\delta(X_{1:T} \mid S_{1:T})$ as

$$\hat{P}_\delta(X_{1:T} \mid S_{1:T}) = \prod_t \delta(x_t - \hat{x}_t(S_{1:T}))\delta(\Delta x_t - \Delta\hat{x}_t(S_{1:T}))\delta(\Delta\Delta x_t - \Delta\Delta\hat{x}_t(S_{1:T})). \tag{12}$$

$\hat{P}(Y_{1:T} \mid X_{1:T}, S_{1:T})$ is obtained by

$$\hat{P}(Y_{1:T} \mid X_{1:T}, S_{1:T}) = \prod_t \hat{P}(Y_t \mid X_t, S_t), \text{ and} \tag{13}$$

$$\hat{P}(Y_t \mid X_t, S_t) = N(y_t; x_t, \hat{\rho}_{S_t}{}^2)N(\Delta y_t; \Delta x_t, \Delta\hat{\rho}_{S_t}{}^2)N(\Delta\Delta y_t; \Delta\Delta x_t, \Delta\Delta\hat{\rho}_{S_t}{}^2), \tag{14}$$

whose means and variances are $x_t$ $\Delta x_t$ $\Delta\Delta x_t$ and $\hat{\rho}_{S_t}{}^2$ $\Delta\hat{\rho}_{S_t}{}^2$ $\Delta\Delta\hat{\rho}_{S_t}{}^2$, respectively. Note that we introduce new variances, $\hat{\rho}_{S_t}{}^2$ $\Delta\hat{\rho}_{S_t}{}^2$ $\Delta\Delta\hat{\rho}_{S_t}{}^2$, for each HMM state. We assume that from here $x_t$, $\Delta x_t$, and $\Delta\Delta x_t$ are independent of each other. Although if the state sequence is given, likelihood can be calculated, and the actual state sequence is unknown. Assuming that the state transition probability is identical to HMM, the formulation of the obtained state sequence can be calculated by

$$\begin{aligned} \max_{S_{1:T}} \hat{P}(Y_{1:T}, S_{1:T}) &= \max_{S_{1:T}} \int_{X_{1:T}} \hat{P}(S_{1:T}, Y_{1:T}, X_{1:T}) \\ &= \max_{S_{1:T}} \int_{X_{1:T}} \hat{P}(S_{1:T}, Y_{1:T}, X_{1:T}) \\ &= \max_{S_{1:T}} \int_{X_{1:T}} \hat{P}(Y_{1:T} \mid X_{1:T}, S_{1:T}) \hat{P}_\delta(X_{1:T} \mid S_{1:T})\, P(S_{1:T}) \\ &= \max_{S_{1:T}} \hat{P}(Y_{1:T} \mid \hat{X}_{1:T}(S_{1:T}), S_{1:T}) P(S_{1:T}). \end{aligned} \tag{15}$$

$\hat{X}_t(S_{1:T})$ is obtained by

$$\hat{X}_t(S_{1:T}) = C\hat{X}^t(S_{1:T}), \text{ and} \tag{16}$$

$$\hat{X}^t(S_{1:T}) = \left[\hat{x}_{t+2}(S_{1:T})\ \hat{x}_{t+1}(S_{1:T})\ \hat{x}_t(S_{1:T})\ \hat{x}_{t-1}(S_{1:T})\ \hat{x}_{t-2}(S_{1:T})\right]' \tag{17}$$

from the sequence of $\hat{x}_t(S_{1:T})$ calculated by Equation (10).

However, no efficient algorithm, such as the Viterbi algorithm, effectively searches for the best state sequence. To approximate the state sequence, we use the state sequence obtained by the Viterbi algorithm using ordinary HMMs as

$$\overline{S}_{1:T} = \arg\max_{S_{1:T}} P(Y_{1:T} \mid S_{1:T}) P(S_{1:T}). \tag{18}$$

Therefore the approximate trajectory and the corresponding likelihood are calculated as

$$\hat{x}_{-1:T+2}(\overline{S}_{1:T}) = \arg\max_{x_{-1:T-2}} P(X_{1:T} \mid \overline{S}_{1:T}) \text{ and} \tag{19}$$

$$\begin{aligned} \hat{P}(Y_{1:T}) &\approx \max_{S_{1:T}} \hat{P}(Y_{1:T} \mid \hat{X}_{1:T}(S_{1:T}), S_{1:T}) P(S_{1:T}) \\ &\approx \hat{P}(Y_{1:T} \mid \hat{X}_{1:T}(\overline{S}_{1:T}), \overline{S}_{1:T}) P(\overline{S}_{1:T}). \end{aligned} \tag{20}$$

### 2.4 Training in trajectory likelihood

The method described in Section 2 requires training variables $\widehat{M}_{s_t} = [\hat{\sigma}^2_{S_t}, \Delta\hat{\sigma}^2_{S_t}, \Delta\Delta\hat{\sigma}^2_{S_t}]'$. A simple training method called Viterbi training is introduced to calculate variances $\widehat{M}_{s_t}$ for each state along with the trajectory. The following is the basic procedure:

(1) Calculate Viterbi paths with HMM for all training data using

$$\overline{S}_{n,1:T_n} = \arg\max_{S_{1:T_n}} P(Y_{n,1:T_n} \mid S_{1:T_n}) P(S_{1:T_n}), \tag{21}$$

where n is the amount of data.

(2) Generate trajectories for the training data using the Kalman smoother whose objective function is

$$\hat{x}_{n,-1:T_n+2}(\overline{S}_{n,1:T_n}) = \arg\max_{x_{-1:T+2}} P(X_{1:T_n} \mid \overline{S}_{n,1:T_n}). \tag{22}$$

(3) Calculate equation

$$\hat{\rho}_i{}^2, \Delta\hat{\rho}_i{}^2, \Delta\Delta\hat{\rho}_i{}^2 = \arg\max_{\hat{\rho}_i{}^2, \Delta\hat{\rho}_i{}^2, \Delta\Delta\hat{\rho}_i{}^2} \sum_n^N \log[\hat{P}(Y_{n,1:T_n} \mid \hat{X}_{n,1:T_n}(\overline{S}_{n,1:T_n}), \overline{S}_{n,1:T_n}) P(\overline{S}_{n,1:T_n})] \tag{23}$$

to obtain $\hat{\rho}_i{}^2, \Delta\hat{\rho}_i{}^2, \Delta\Delta\hat{\rho}_i{}^2$, where $N$ denotes the amount of training data.

## 3. EXTENTION TO MIXTURE GAUSSIAN COMPONENTS

In this section, we discuss an extension of the HMM-trajectory method for treating a mixture Gaussian component framework. Before addressing the new likelihood, an ordinary HMM likelihood for mixture Gaussian components is discussed in 3.1, and our previous method described in [8] is explained in 3.2. Then we discuss the new method that compensates for HMM trajectory likelihood in 3.3.

### 3.1. HMM likelihood for mixture Gaussian components

HMM likelihood for Gaussian mixture components is obtained by

$$P(Y_{1:T}) \approx \max_{S_{1:T}} P(Y_{1:T}, S_{1:T}) = \max_{S_{1:T}} \sum_{K_{1:T}} P(Y_{1:T}, S_{1:T}, K_{1:T})$$
$$= \max_{S_{1:T}} \left[ \sum_{K_{1:T}} P(S_1, K_1) P(Y_1 \mid S_1) \prod_{2}^{T} P(Y_t \mid S_t, K_t) P(S_t, K_t \mid S_{t-1}) \right], \quad (24)$$

where $K_{1:T}$ is a sequence of mixture component numbers. $P(Y_t \mid S_t, K_t)$ is the emission probability for each Gaussian component defined as

$$P(Y_t \mid S_t, K_t) = N(y_t; \mu_{S_t,K_t}, \rho_{S_t,K_t}) N(\Delta y_t; \Delta\mu_{S_t,K_t}, \Delta\rho_{S_t,K_t}) N(\Delta\Delta y_t; \Delta\Delta\mu_{S_t,K_t}, \Delta\Delta\rho_{S_t,K_t}), \quad (25)$$

$P(S_t, K_t \mid S_{t-1})$ is the multiplication of transition probability from $S_{t-1}$ to $S_t$ and the weight for the $K_t$-th mixture component in $S_t$. $\mu_{S_t,K_t}$, $\Delta\mu_{S_t,K_t}$, and $\Delta\Delta\mu_{S_t,K_t}$ are the mean values for the $K_t$-th component in state $S_t$. $\rho_{S_t,K_t}{}^2$, $\Delta\rho_{S_t,K_t}{}^2$, and $\Delta\Delta\rho_{S_t,K_t}{}^2$ are the variance values for the $K_t$-th component in state $S_t$.

### 3.2. HMM trajectory likelihood for mixture Gaussian components

We define the likelihood for the mixture Gaussian components as

$$\hat{P}(Y_{1:T}) = \sum_{S_{1:T},K_{1:T}} \hat{P}(Y_{1:T}, S_{1:T}, K_{1:T}) = \sum_{S_{1:T},K_{1:T}} \int_{X_{1:T}} \hat{P}(Y_{1:T}, X_{1:T}, S_{1:T}, K_{1:T})$$
$$= \sum_{S_{1:T},K_{1:T}} \int_{X_{1:T}} \hat{P}(Y_{1:T} \mid X_{1:T}, S_{1:T}, K_{1:T}) \hat{P}_\delta(X_{1:T} \mid S_{1:T}, K_{1:T}) P(S_{1:T}, K_{1:T}) \quad (26)$$
$$= \sum_{S_{1:T},K_{1:T}} \hat{P}(Y_{1:T} \mid \hat{X}_{1:T}(S_{1:T}, K_{1:T}), S_{1:T}, K_{1:T}) P(S_{1:T}, K_{1:T}),$$

where $\hat{X}_{1:T}(S_{1:T}, K_{1:T})$ can be obtained by

$$\hat{x}_{-1:T+2}(S_{1:T}, K_{1:T}) = \arg\max_{x_{-1:T+2}} P(X_{1:T} \mid S_{1:T}, K_{1:T}) \quad (27)$$

with conditions (7), (8), and (9). Probability density for input feature $Y_{1:T}$, given $S_{1:T}, K_{1:T}$, is defined as

$$\hat{P}(Y_{1:T} \mid \hat{X}_{1:T}(S_{1:T}, K_{1:T}), S_{1:T}, K_{1:T}) = \prod_t \hat{P}(Y_t \mid \hat{X}_t(S_{1:T}, K_{1:T}), S_t, K_t), \quad (28)$$

where frame-wise probability density $\hat{P}(Y_t \mid \hat{X}_t(S_{1:T}, K_{1:T}), S_t, K_t)$ is defined as

$$\hat{P}(Y_t \mid \hat{X}_t(S_{1:T}, K_{1:T}), S_t, K_t) = N(y_t; \hat{x}_t(S_{1:T}, K_{1:T}), \rho'_{S_t,K_t}) N(\Delta y_t; \Delta\hat{x}_t(S_{1:T}, K_{1:T}), \Delta\rho'_{S_t,K_t}) \cdot N(\Delta\Delta y_t; \Delta\Delta\hat{x}_t(S_{1:T}, K_{1:T}), \Delta\Delta\rho'_{S_t,K_t}). \quad (29)$$

$P(S_{1:T}, K_{1:T})$ is the multiplication of state transition probabilities and Gaussian component weights, whose values are identical to the mixture Gaussian HMM. We introduce new variances, $\hat{\rho}_{S_t,K_t}{}^2$ $\Delta\hat{\rho}_{S_t,K_t}{}^2$ $\Delta\Delta\hat{\rho}_{S_t}{}^2$, for each HMM Gaussian component.

Obtaining Equation (26) for all combinations of $S_{1:T}, K_{1:T}$ requires much calculation. Instead of calculating Equation (26) for all possible $S_{1:T}, K_{1:T}$, we selected $\bar{S}_{1:T}, \bar{K}_{1:T}$ to maximize HMM probability, given $S_{1:T}, K_{1:T}$, as

$$\bar{S}_{1:T}, \bar{K}_{1:T} = \arg\max_{S_{1:T}K_{1:T}} P(Y_{1:T} \mid S_{1:T}, K_{1:T}) P(S_{1:T}, K_{1:T}). \quad (30)$$

Using $\bar{S}_{1:T}, \bar{K}_{1:T}$, in [8] we calculated the trajectory and likelihood as $\bar{x}_{-1:T+2}(\bar{S}_{1:T}, \bar{K}_{1:T}) = \arg\max_{x_{-1:T+2}} P(X_{1:T} \mid \bar{S}_{1:T}, \bar{K}_{1:T})$ and (31)

$$\hat{P}(Y_{1:T}) \approx \hat{P}(Y_{1:T} \mid \hat{X}_{1:T}(\bar{S}_{1:T}, \bar{K}_{1:T}), \bar{S}_{1:T}, \bar{K}_{1:T}) P(\bar{S}_{1:T}, \bar{K}_{1:T}). \quad (32)$$

### 3.3. Compensated likelihood for HMM-trajectory method

After we evaluated the method described in 3.2. with a large amount of data, the error rate improvement decreased, presumably for the following reason. If the trajectory is near the sequence of input speech features, Equation (32) might work well. However, if the obtained trajectory is far from the sequence of input speech features, the likelihood value decreases, and consequently recognition accuracy decreases because the likelihood described in 3.3. ignores term $\sum_{K_{1:T} \neq \bar{K}_{1:T} \text{and} S_{1:T} \neq \bar{S}_{1:T}} \hat{P}(Y_{1:T} \mid \hat{X}_{1:T}(S_{1:T}, K_{1:T}), S_{1:T}, K_{1:T}) P(S_{1:T}, K_{1:T})$. The trajectories in $\sum_{K_{1:T} \neq \bar{K}_{1:T} \text{and} S_{1:T} \neq \bar{S}_{1:T}} \hat{P}(Y_{1:T} \mid \hat{X}_{1:T}(S_{1:T}, K_{1:T}), S_{1:T}, K_{1:T}) P(S_{1:T}, K_{1:T})$ are smoothed sequences of HMM mean sequences along $S_{1:T}, K_{1:T}$. This means the ignored likelihood values are similar to the HMM likelihood (if we neglect the probability density for the HMM best path). We believe that the ignored likelihood can be approximated by the HMM likelihood. Therefore, we compensate for HMM-trajectory likelihood with HMM likelihood in the log domain and propose a new likelihood equation:

$$\log \hat{P}(Y_{1:T}) \approx \frac{1}{2} \log(\hat{P}(Y_{1:T} \mid \hat{X}_{1:T}(\bar{S}_{1:T}, \bar{K}_{1:T}), \bar{S}_{1:T}, \bar{K}_{1:T}) P(\bar{S}_{1:T}, \bar{K}_{1:T})) + \frac{1}{2} \log(\max_{S_{1:T}} P(Y_{1:T}, S_{1:T})) \quad (33)$$

Although a weight for the compensated likelihood might generally be effective, we do not use one here.

## 4. RECOGNITION EXPERIMENTS

We performed city name recognition experiments to evaluate our method. Table 1 shows the experimental conditions. The evaluation task was the recognition of 100 city names. About 20,000 training utterances were used to train the tri-phone HMMs using only male speakers. Each HMM had three states. The test data were recognized using HMMs by a full search. State-based segmentation was performed by the Viterbi algorithm for each candidate to obtain state alignments for the input utterances. The trajectory for each candidate was then generated using a Kalman smoother. Frame-wise likelihood between the generated trajectory and the input speech MFCC features was calculated. We used three types of likelihood to reorder the candidates: HMM, HMM-trajectory calculated by Equation (32), and HMM-trajectory calculated by Equation (33).

Table 1 Conditions in experiments

| | Condition |
|---|---|
| Number of training sentences | 20,093 |
| Test data | 7198 city names uttered by 75 speakers |
| Feature parameters | MFCC 1-12<br>16 kHz sampling rate<br>10 msec frame shift |
| Gender | Male |

We performed experiments under two conditions; the numbers of mixture components in the state were either two or three.

Table 2 shows the word error rates of the experiment using two Gaussian mixture components. To increase the experiment's reliability, we used three different amounts of states in the tri-phone HMMs: 2859, 1992, and 1611. While the baseline system obtained 2.33%, 2.27%, and 2.36% error rates at 2859, 1992, and 1611 states, respectively, HMM trajectory likelihood obtained

2.22%, 2.13%, and 2.27%, respectively. For this task, HMM trajectory likelihood outperforms HMM likelihood. Compensated HMM trajectory likelihood obtained 2.06%, 2.00%, and 2.14% error rates for the same task. Error reduction rates from HMMs were 11.8%, 11.9%, and 9.3%, respectively. Compared with the previous method, compensated HMM trajectory likelihood shows better results. These results indicate that compensated likelihood significantly improves recognition accuracy.

Table 3 shows the word error rates for the experiment for three mixture Gaussian HMMs. We evaluated using tri-phone HMMs with only 1992 states.

The baseline system obtained 1.89% error, and HMM trajectory likelihood obtained 1.89%. While these two methods show identical error rates, the proposed HMM trajectory likelihood obtained an 1.72% error rate for the same task. The error reduction rate was 9.0%, showing that the proposed likelihood improved recognition accuracy more than our previous HMM trajectory likelihood.

We compared the proposed method's memory and calculation with ordinary HMMs. The proposed method requires an extra variance matrix for each HMM distribution. This means that in 1992 state HMMs with three mixture components, 5976 extra diagonal variance matrices are required. In this case, a full search of the proposed method was about 10 times slower than the HMM Viterbi full search (note that this comparison was performed using different computer languages for two methods and full search is an exhausted method in speech recognition).

Table 2 Word error rates (for two mixture components)

| Number of states | HMM trajectory likelihood | Compensated HMM trajectory likelihood | HMM likelihood (Baseline) |
|---|---|---|---|
| 2859 | 2.22% | 2.06% | 2.33% |
| 1992 | 2.13% | 2.00% | 2.27% |
| 1611 | 2.27% | 2.14% | 2.36% |

Table 3 Word error rates (for three mixture components)

| Number of states | HMM trajectory likelihood | Compensated HMM trajectory likelihood | HMM likelihood (Baseline) |
|---|---|---|---|
| 1992 | 1.89% | 1.72% | 1.89% |

## 5. CONCLUSION

This paper proposed a new speech recognition method that improves the HMM-trajectory method and enhances the performance of Gaussian mixture distributions. The previous HMM-trajectory method obtained state and mixture component sequences using the Viterbi algorithm with ordinary HMMs, generated the trajectory using the information, and calculated likelihood along the trajectory. However since this method ignores the likelihood generated by the other sequences of the states and mixture components, the method degraded recognition accuracy. To improve performance, we compensated for HMM trajectory likelihood using ordinary HMM likelihood. Our proposed method was evaluated with speaker independent speech recognition experiments that yielded about a 10% reduction in error rate for evaluations with two mixture Gaussian components and three mixture components, proving that our method improved the recognition performance for Gaussian mixture components.

## 6. REFERENCES


[1] J. S. Bridle, L. Deng, J. Picone, H. B. Richards, J. Ma, T. Kamm, M. Schuster, S. Pike, and R. Regan, "An investigation of segmental hidden dynamic models of speech coarticulation for automatic speech recognition," http://www.clsp.jhu.edu/ws98/projects/dynamic/presentations/final/ WS98 final report, 1998.

[2] H. B. Richards and J. S. Bridle, "The HDM: A segmental hidden dynamic model of coarticulation," Proc. ICASSP, 357-360, 1999.

[3] J. Picone, S. Pike, R. Reagan, T. Kamm, J. Bridle, L. Deng, Z. Ma, H. Richards, and M. Schuster, "Initial evaluation of hidden dynamic models on conversational speech," Proc. ICASSP, pp. 109-112, 1999.

[4] L. Deng, "A dynamic, feature-based approach to the interface between phonology and phonetics for speech modeling and recognition," Speech Communication, 24 (4), pp. 299-323, 1998.

[5] V. Digalakis, J. R. Rohlicek, and M. Ostendorf, "A dynamical system approach to continuous speech recognition," IEEE Trans. Speech Audio Processing, Vol. 1, No. 4, pp. 431-442, 1993.

[6] Y. Minami, E. McDermott, A. Nakamura, and S. Katagiri, "A recognition method using synthesis-based scoring that incorporates direct relations between static and dynamic feature vector time series," Workshop for Consistent & Reliable Acoustic Cues for Sound Analysis, 2001.

[7] Y. Minami, E. McDermott, A. Nakamura, and S. Katagiri, "A recognition method with parametric trajectory synthesized using direct relations between static and dynamic feature vector time series," Proc. ICASSP, pp. 957-960, 2002.

[8] Y. Minami, E. McDermott, A. Nakamura, and S. Katagiri, "A recognition method with parametric trajectory generated from mixture distribution HMMs," Proc. ICASSP, pp. 124-127 2003.

[9] Y. Minami, E. McDermott, A. Nakamura, and S. Katagiri, "A Theoretical Analysis of Speech Recognition based on Feature Trajectory Models," Proc. ICSLP, vol. I, 2004.

[10] K. Tokuda, T. Kobayashi, and S. Imai, "Speech parameter generation from HMM using dynamic features," Proc. ICASSP, pp. 660-663, 1995.

[11] T. Kailath, A. H. Sayed, and B. Hassibi, "Linear estimation," Prentice Hall, 2000.


## APPENDICES

To calculate Equation (10), we used a Kalman smoother that models state space as

$$M_{S_t} = C\,X^t + W_{S_t} \quad \text{and} \tag{34}$$

$$X^{t+1} = AX^t + N_t, \tag{35}$$

where matrix A is a shift operator:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{36}$$

$W_{S_t}$ is a random variable vector whose mean vector and covariance matrix are defined as: $[0,0,0]'$,

$$\Sigma_{S_t} = diag\left[\sigma_{S_t}{}^2 \;\; \Delta\sigma_{S_t}{}^2 \;\; \Delta\Delta\sigma_{S_t}{}^2\right]. \tag{37}$$

$M_{S_t}$ is defined as

$$M_{S_t} = [\mu_{S_t}, \Delta\mu_{S_t}, \Delta\Delta\mu_{S_t}]'. \tag{38}$$

$N_t = \left[n_{t+2}, n_{t+1}, n_t, n_{t-1}, n_{t-2}\right]'$ are random variables whose mean vectors and covariance matrixes are defined as:

$[0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]'$, and

$$\Theta = \begin{bmatrix} \theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{39}$$

where $\theta$ is a large positive number.