

## 음성인식을 위한 은닉마코프모형 연구<sup>1)</sup>

손건태<sup>2)</sup> · 정상화<sup>3)</sup> · 박민욱<sup>4)</sup>

### 요 약

음성자동인식을 위한 통계적 방법으로 은닉마코프모형이 널리 사용되고 있다. 이산형 은닉마코프모형보다 인식률이 우수한 연속형 은닉마코프모형을 고려하였으며, 인식을 위한 비터비(Viterbi) 알고리즘을 병렬화시켜 인식속도를 빠르게 하는 인식 알고리즘을 제안하였다. 제안된 방법으로 실험을 통하여 인식률과 인식속도 개선률(speed-up)을 살펴보았다.

### 1. 서 론

음성자동인식 기술에 대한 필요성은 통신분야, 멀티미디어분야, 자동통역시스템분야에서 더욱 가속화되고 있다. 음성자동인식을 위하여 훈련자료(training data)를 이용한 통계적 방법으로 은닉마코프모형(hidden Markov model, HMM)이 전세계적으로 이용되고 있다. 실험결과를 통하여 HMM은 음성지식을 기초한 방법보다 인식률에서 우수한 것으로 인정되고 있으며, 국내에서도 음향학회를 비롯하여 전자통신분야의 학과와 관련연구소에서 활발히 연구되고 있다. 그러나 특정 범위(domain)내에서 적은 수의 단어를 이용한 연속음 인식에서는 좋은 결과를 보여주고 있지만, 대용량의 단어를 포함하고 있는 음성인식 시스템을 실시간에 구현하기는 아직 미흡한 실정이다. 음성자동인식시스템은 훈련용 음성자료를 이용하여 대응되는 모형을 구축하는 훈련과정(training procedure)과 인식을 목적으로 하는 새로운 음성에 대한 인식과정(recognition procedure)으로 나뉘어지며, 훈련과정에서는 모형의 추정문제와 인식률이 주요 관심이 되고, 인식과정에서는 인식률과 인식속도가 주요 관심 대상이 된다.

HMM은 이산형, 연속형, 반연속형으로 나뉘어지고 있으며, 이산형은 인식속도면에서는 우수하나 인식률에서 떨어지는 단점이 있어, 본 연구에서는 연속형 HMM을 고려하였다. 그러나 연속형 HMM은 인식을 위한 계산량이 매우 많아 인식속도가 떨어지는 단점을 지니고 있으므로, 실시간 음성인식속도를 높이기 위하여 음성인식과정에서 필요한 각각의 작업들을 다중처리기(multiprocessor) 상에서 병렬처리(parallel processing)함으로써 음성인식 시스템에서 요구되는 대용량의 단어들의 인식시간문제를 해결하고자 한다.

현재까지 음성인식시스템 구현에 있어서 병렬화 알고리즘의 연구는 국내에서는 거의 없으며, 외국의 경우에 있어서도 서서히 연구되고 있다. Huijen(1996)은 이산형 HMM을 기본 모델로 하여

1) 이 연구는 1996년도 한국학술진흥재단 학술연구조성비 지원에 의하여 수행되었음  
2) (609-735) 부산 금정구 장전동 30 부산대학교 통계학과 부교수  
3) (609-735) 부산 금정구 장전동 30 부산대학교 컴퓨터공학과 조교수  
4) (609-735) 부산 금정구 장전동 30 부산대학교 컴퓨터공학과 박사과정

연속음성이 아닌 단일 음성에 대해 병렬화 연구를 하였으며, Phillip과 Rogers(1997)는 multi-threading 기법을 사용하여 연속음성인식을 시도하였다.

음성인식을 위한 HMM의 관련연구는 국내 통계전공자들에 의하여 이루어진 예가 거의 없어 2절에서는 HMM과 음성인식시스템에 대하여 소개를 하였으며, 3절에서는 연속형 HMM을 이용하여 실시간 음소인식의 병렬 알고리즘을 구현하고, 실험을 통하여 인식률과 인식속도의 증가율을 조사하여 성능을 평가하였다.

## 2. 은닉마코프모형과 음성인식시스템

HMM은 Baum(1966)에 의하여 제안되었으며 Baum 외(1970, 1972)에 기본적 이론이 연구되어졌다. 음성인식에의 응용은 Baker(1975), Jelinek 외(1975, 1976, 1983)에 의하여 이루어졌으며, Rabiner와 Juang를 중심으로 하여 매우 많은 연구가 진행되고 있다. 이절에서는 은닉마코프모형에서의 추정과 인식 알고리즘에 대하여 간략히 소개를 한다. 이산형 HMM에 대한 자세한 이론은 Juang과 Rabiner(1991, 1993)을 참고하면 된다.

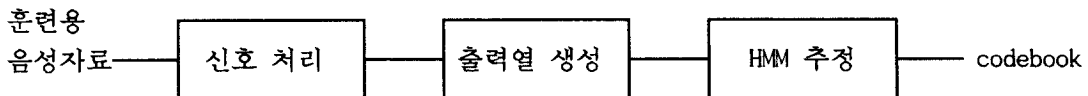
### 2.1 이산형 은닉마코프모형

상태공간  $S$ 의 원소수가  $N$ 이고 출력 종류의 수가  $L$ 일 때, 이산형 HMM에서의 출력치(observation) 생성에 대한 확률구조는 다음과 같이 나타낼 수 있다.

$$\begin{cases} \pi_{t+1} = \pi_t A \\ Q_t = \pi_t B \end{cases} \quad (2.1)$$

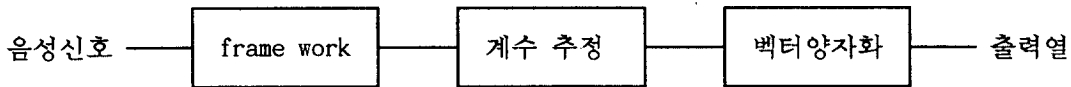
여기서  $\pi_t$ 는  $t$ 시점에서의 상태확률분포를 나타내는  $1 \times N$  행벡터이며,  $A = (a_{ij})$ 는 마코프 연쇄의  $N \times N$  전치확률행렬이고,  $Q_t$ 는  $t$ 시점에서의 출력치확률분포를 나타내는  $1 \times L$  행벡터,  $B = (b_j(q_k))_{N \times L}$ 는 출력확률행렬(output probability matrix)이며,  $b_j(q_k)$ 는 상태가  $j$ 일 때 출력치  $q_k$ 가 생성될 확률이다. 일반적으로 출력치만 관측되고 상태의 흐름은 관측되지 못한 경우에 사용하므로 은닉마코프모형이라 부른다. 출력치 생성을 위하여 초기값  $\pi_0$ 가 필요하므로, 이산형 HMM은  $\lambda = (A, B, \pi_0)$ 로 표현된다.

음성인식시스템의 훈련과정에서는 [그림 1]의 단계로 훈련용 음성자료를 이용하여 출력열(output sequence)  $O = (O_1, O_2, \dots, O_T)$ 를 만든 후 이에 대한 HMM을 최대가능도방법으로 추정한다. 수많은 단어(또는 음소, 연결음)에 대하여 동일한 훈련을 하여 대응된 모형(codebook)을 추정하고 새로운 음성에 대한 인식에 활용하도록 하는 것이 훈련과정의 목표이다.



[그림 1] 훈련과정

음성자료에 대한 출력열을 생성하기 위한 신호처리과정은 [그림 2]로 표현된다. 간략히 설명하면 첫째, 아날로그 형태의 음성신호를 디지털화하고, 둘째, 작은 시간대(10ms-20ms정도)로 세분화한다. 세분화된 시간대를 각각 frame이라 부른다. 셋째, 각 frame의 음성신호를 windowing으로 평활화시킨 후, 자기회귀모형(AR(12))으로 적합시켜 12개의 계수를 추정한다. 또한 스펙트럴 분석을 하여 음성 특성을 나타내는 켈스트럴 계수(cepstral coefficient)를 12개 구하고 에너지값을 2개 구하여 전체 26개의 값을 각 frame의 음성신호특성을 나타내는 벡터로 단순화한다. 다음은 벡터양자화(vector quantization)과정으로 집락분석과 판별분석을 이용하여 26차원 벡터에 대하여 하나의 군집번호를 부여한다. 즉, 각 frame에 대하여 하나의 값이 대응되는 것이다. 이값들이 출력열이 된다. frame의 수가 T일 때 출력열은  $O = (O_1, O_2, \dots, O_T)$ 로 나타난다.



[그림 2] 출력열 생성과정

HMM에서는 추정과 인식을 위하여 다음의 세 가지 문제를 해결해야 한다.

### 2.1.1 계산 문제

은닉마코프모형  $\lambda = (A, B, \pi_0)$ 이 주어진 경우  $P(O|\lambda)$ 의 계산을 빠르게 하기 위한 알고리즘으로 전진알고리즘과 후진알고리즘이 있다. 전진알고리즘은 전진변수  $\alpha_t(i) = P(O_1, \dots, O_t, s_t = i | \lambda)$ 를 사용하여 다음의 과정으로  $P(O|\lambda)$ 를 계산하며, 이때의 계산량은  $O(N^2T)$ 이 된다.

[단계 1]  $i \in S_T$ ,  $\alpha_1(i) = \pi_0^{(i)} b_i(O_1)$ 를 계산한다.

[단계 2]  $t=2, \dots, T$ 로 변화시켜가며, 각  $j \in S$ 에 대하여

$$\alpha_t(j) = [\sum_i \alpha_{t-1}(i) a_{ij}] b_j(O_t) \text{를 계산한다.}$$

[단계 3]  $P(O|\lambda) = \sum_i \alpha_T(i)$

여기서  $S_T$ 는 초기값으로 가능한 상태들의 집합이다.

후진알고리즘은 후진변수  $\beta_t(i) = P(O_{t+1}, \dots, O_T | s_t = i, \lambda)$ 를 사용하여 다음의 과정으로  $P(O|\lambda)$ 를 계산한다.

$$[\text{단계 1}] \beta_T(i) = \begin{cases} \frac{1}{N_F}, & i \in S_F \\ 0, & i \notin S_F \end{cases}$$

[단계 2]  $t = T-1, \dots, 1$  와 같이 거꾸로 변화시켜가며, 각  $j \in S$  에 대하여

$$\beta_t(j) = \sum_i a_{ij} b_i(O_{t+1}) \beta_{t+1}(i) \text{를 계산한다.}$$

[단계 3]  $P(O|\lambda) = \sum_i \pi_0^{(i)} b_i(O_1) \beta_1(i)$

여기서  $S_F$  는 최종값으로 가능한 상태들의 집합이다. 전진변수값들과 후진변수값들은 추정문제에서 사용되므로 반드시 계산되어야 한다.

### 2.1.2 추정 문제

관측되어진 출력열  $O = (O_1, O_2, \dots, O_T)$  를 이용하여  $\lambda = (A, B, \pi_0)$  을 이루고 있는 모수들을  $P(O|\lambda)$  를 최대로 하는 최대가능도방법으로 추정하기 위하여, Dempster, Laird와 Rudin(1977)에 의해 제안된 EM알고리즘을 사용한다. 음성인식에서 EM 알고리즘에 의한 반복계산으로 HMM 모수를 재추정해가는 알고리즘을 Baum-Welch 재추정 알고리즘(re-estimation algorithm)이라 한다. 먼저  $(O, \lambda)$  이 주어진 경우  $(t, i)$  에서  $(t+1, j)$  로 가는 확률과 시점 t에서 상태 i일 확률을 구하면 다음과 같다.

$$\begin{aligned} \gamma_t(i, j) &= P(s_t = i, s_{t+1} = j | O, \lambda) = P(O, s_t = i, s_{t+1} = j | \lambda) / P(O|\lambda) \\ &= a_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) / \sum_{k \in S_F} a_T(k) \end{aligned} \tag{2.2}$$

$$\gamma_t(i) = P(s_t = i | O, \lambda) = \sum_j \gamma_t(i, j) \tag{2.3}$$

위의 두 함수를 이용하여  $(O, \lambda)$  이 초기조건으로 주어진 경우 재추정된 모수값들은 다음으로 계산된다.

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{j \text{에서 } j \text{로 갈 확률의 평균}}{i \text{에서 출발할 확률의 평균}}, \tag{2.4}$$

$$\hat{b}_j(k) = \frac{\sum_{t \in \{t | O_t = q_k\}} \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} = \frac{j \text{에서 } O_t = q_k \text{가 될 갯수}}{j \text{에 머물러 있을 갯수}}, \tag{2.5}$$

$$\hat{\pi}_0^{(i)} = \gamma_1(i) = \hat{\pi}_0 \text{의 } i \text{번째 성분값.} \tag{2.6}$$

따라서 Baum-Welch 재추정 알고리즘은 다음 과정으로 이루어진다.

[단계 1] HMM의 초기모형  $\lambda^{(0)}$ 를 선택한다.

[단계 2] 반복단계:  $r = 0, 1, 2, \dots$ 로 변화시키며

주어진 모형  $\lambda^{(r)}$ 에서 출력열  $O$ 와 식(2.4)-(2.6)을 이용하여  $\hat{\lambda}$ 을 구한 후,  $\hat{\lambda}$ 를  $\lambda^{(r+1)}$ 로 놓고 다시 반복단계를 수행한다.

[단계 3] 모수 추정치들의 변화감소가 원하는 만큼 작아지면 반복을 중단하고, 이때

의 추정된 모형을 주어진 출력열  $O$ 에 대응되는 HMM으로 결정한다.

동일 음성을 독립적인 실험으로 반복한 훈련자료로 해당 음성의 HMM을 추정하는 경우에는 출력열이  $O^M = (O^1, \dots, O^m)$ 의 형태가 되며, 여러 형태의 경우를 종합하여 인식률을 높힐 수 있다. 이때의 로그가능도함수는

$$\log P(O^M|\lambda) = \sum_{n=1}^m \log P(O^n|\lambda)$$

가 되므로 EM알고리즘의 적용이 순쉬우며, 재추정치는 다음으로 계산된다.

$$\hat{a}_{ij} = \frac{\sum_n \sum_{i=1}^{T_n-1} \gamma_i^n(i, j)}{\sum_n \sum_{i=1}^{T_n-1} \gamma_i^n(i)}, \hat{b}_j(k) = \frac{\sum_n \sum_{t \in \{t | O_t^n = q_k\}} \gamma_t^n(j)}{\sum_n \sum_{i=1}^{T_n} \gamma_i^n(j)} \quad (2.7)$$

### 2.1.3 해독 문제

출력열과 HMM이  $(O, \lambda)$ 로 주어진 경우 최적상태열(optimal state sequence)을 찾는 문제로, 일반적으로 다음 과정으로 이루어진 Viterbi(1967) 알고리즘을 사용한다.

[단계 1] 모든  $i \in S$ 에 대하여,  $\delta_1(i) = \pi_0^{(i)} b_i(O_1)$ ,  $\psi_1(i) = 0$ 를 계산한다.

[단계 2] 반복단계:  $t = 2, \dots, T$ 로 변화시켜가며,

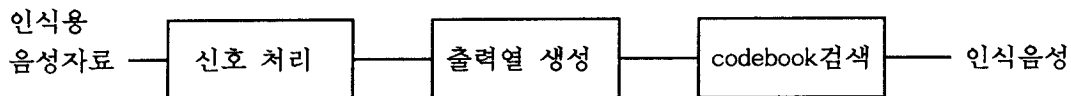
$$\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(O_t), \psi_t(j) = \arg \max_i [\delta_{t-1}(i) a_{ij}] \text{를 계산한다.}$$

[단계 3]  $P^* = \max_s \delta_T(s)$ ,  $s_T^* = \arg \max_s \delta_T(s)$ 를 계산한다.

[단계 4]  $t = T-1, \dots, 1$ 처럼 거꾸로 변화시키며, 다음 식을 통하여 경로 역추적방법(path backtracking)으로 최적 상태열을 찾는다.

$$s_t^* = \psi_{t+1}(s_{t+1}^*).$$

Viterbi 알고리즘은 새로운 음성에 대한 인식알고리즘으로 사용하기 때문에 해독 문제(decoding problem)라고 한다. 새로운 음성에 대한 인식과정은 [그림 3]으로 표현된다. 인식대상 음성자료에 대한 출력열  $O$ 가 신호처리과정을 통하여 구해지면 훈련과정으로 만들어진 codebook을 검색하여 각 codeword(즉, 훈련된 HMM)에 대한  $P(O|\lambda)$ 들을 계산하여 확률을 최대로 하는 HMM을 인식 음성으로 한다.



$P(O|\lambda)$  계산

[그림 3] 인식과정

$P(O|\lambda)$  계산에서 HMM  $\lambda$ 가 주어진 경우, 만일 상태열  $(s_1, s_2, \dots, s_T)$ 이 알려있다면  $P(O|\lambda) = b_{s_1}(O_1)b_{s_2}(O_2)\dots b_{s_T}(O_T)$ 로 쉽게 계산된다. 많은 실험을 통하여 볼 때 Viterbi알고리즘에 의하여 구해진 최적상태열  $(s_1^*, s_2^*, \dots, s_T^*)$ 을 실제상태열이라고 생각해 구한  $P^*(O|\lambda)$ 과 2.1.1의 전진알고리즘으로 구한  $P(O|\lambda)$ 는 거의 차이가 없음을 Rabiner와 Juang(1986)은 보이고 있다. 뿐만 아니라 Viterbi 알고리즘에서의 계산은 underflow 문제로 대부분 로그변환을 하여 계산을 한 후 역변환을 하기 때문에 덧셈연산만이 행해져 인식속도면에서 매우 우수함을 알 수 있다.

### 2.2 연속형 은닉마코프모형

음성자료에 대한 출력열을 생성하기 위한 신호처리과정에서 벡터양자화의 오류는 출력열의 변화를 주게되고 결과적으로 인식률을 떨어뜨리는 주요 원인이 된다. 따라서 벡터양자화과정을 생략하고 26차원 음성특징벡터를 직접 사용하여 인식률을 높이고자 Juang(1985), Euler와 Wolf(1987) 등에 의하여 연속형 은닉마코프모형이 연구되었다. 연속형 HMM은 일반적으로 혼합정규분포(mixture normal distribution)를 사용하여 연속형 HMM을 훈련하고 인식한다. 더욱 정확한 HMM을 추정하기 위하여 동일 의미를 지닌 많은 음성자료를 훈련할 때 발생하는 26차원 자료들은 다양한 화자의 개별적 특성에 의하여 다봉분포(multimodal distribution) 형태를 가지므로 화자독립 음성인식 시스템에 적합한 형태가 된다. Sohn과 Baik(1997)은 임의의 분포가 혼합정규분포로 근사될 수 있음을 연구하였다.

연속형 HMM에서는 이산형 HMM의 출력확률로  $b_j(k)$ 대신, 벡터  $\mathbf{x}$ 에 대한 혼합정규분포 확

률값  $b_j(\mathbf{x}) = \sum_{k=1}^M c_{jk} b_{jk}(\mathbf{x}) = \sum_{k=1}^M c_{jk} \mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})$ 를 사용한다. 여기서  $M$ 은 혼합분포를 구성

하는 분포개수이며,  $b_{jk}(\mathbf{x})$ 는 정규분포이고,  $\sum_{k=1}^M c_{jk} = 1$ 이다. 연속형 HMM의 Viterbi 알고리즘도

2.1.3의 이산형 HMM의 Viterbi 알고리즘 각 단계에서  $b_j(k)$ 대신  $b_j(\mathbf{x}) = \sum_{k=1}^M c_{jk} b_{jk}(\mathbf{x})$ 를 사용하

므로, 각 상태별로 혼합분포가 달라 인식과정에서 매우 많은 계산량을 필요로 하는 단점이 있다.

HMM  $\lambda$ 가 주어진 경우 벡터값  $\mathbf{X} = (x_1, x_2, \dots, x_T)$ 와 상태열  $\mathbf{s} = (s_1, s_2, \dots, s_T)$ 의 결합확률밀도함수와 주변확률밀도함수는 다음으로 구해진다.

$$f(\mathbf{X}, \mathbf{s}|\lambda) = \prod_{t=1}^T a_{s_{t-1}s_t} b_{s_t}(x_t) = \sum_{k_1=1}^M \dots \sum_{k_T=1}^M \left[ \prod_{t=1}^T a_{s_{t-1}s_t} b_{s_t k_t}(x_t) \right] c_{s_1 k_1} \dots c_{s_T k_T} \quad (2.8)$$

$$f(\mathbf{X}|\lambda) = \sum_{\mathbf{s}} \sum_{\mathbf{K} \in \Omega^T} f(\mathbf{X}, \mathbf{s}, \mathbf{K}|\lambda) \quad (2.9)$$

여기서  $f(\mathbf{X}, \mathbf{s}, \mathbf{K}|\lambda) = \prod_{t=1}^T a_{s_{t-1}s_t} b_{s_t k_t}(x_t) c_{s_t k_t}$ 이고,  $\Omega^T$ 는  $\Omega = \{1, 2, \dots, M\}$ 의 카티전곱이다. 위식

들과 이산형 HMM을 사용하여 연속형 HMM 모수의 재추정식을 다음과 같이 구할 수 있다.

$$\hat{\pi}^{(i)} = \frac{f(\mathbf{X}, s_1 = i | \lambda)}{f(\mathbf{X} | \lambda)}, \quad (2.10)$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \frac{f(\mathbf{X}, s_t = i, s_{t+1} = j | \lambda)}{f(\mathbf{X} | \lambda)}}{\sum_{t=1}^{T-1} \frac{f(\mathbf{X}, s_t = i | \lambda)}{f(\mathbf{X} | \lambda)}}, \quad (2.11)$$

$$\hat{c}_{jk} = \frac{\sum_{t=1}^T \frac{f(\mathbf{X}, s_t = j, k_t = k | \lambda)}{f(\mathbf{X} | \lambda)}}{\sum_{t=1}^T \frac{f(\mathbf{X}, s_t = j | \lambda)}{f(\mathbf{X} | \lambda)}} = \frac{\sum_{t=1}^T \zeta_t(j, k)}{\sum_{t=1}^T \gamma_t(j)}, \quad (2.12)$$

$$\gamma_t(i, j) = f(s_t = i, s_{t+1} = j | \mathbf{X}, \lambda) = \alpha_t(i) a_{ij} \left[ \sum_{k=1}^M c_{jk} b_{jk}(x_{t+1}) \right] \beta_{t+1}(j) \Bigg/ \sum_{k \in S_f} \alpha_T(k) \quad (2.13)$$

$$\gamma_t(i) = f(s_t = i | \mathbf{X}, \lambda) = \alpha_t(i) \beta_t(i) \Bigg/ \sum_{k \in S_f} \alpha_T(k), \quad (2.14)$$

$$\zeta_t(i, j) = f(s_t = j, k_t = k | \mathbf{X}, \lambda) = \sum_{i'} \alpha_{t-1}(i') a_{i'j} c_{jk} b_{jk}(x_t) \beta_t(j) \Bigg/ \sum_{k \in S_f} \alpha_T(k), \quad (2.15)$$

$$\hat{\mu}_{jk} = \frac{\sum_{t=1}^T \zeta_t(j, k) x_t}{\sum_{t=1}^T \zeta_t(j, k)}, \quad (2.16)$$

$$\hat{\Sigma}_{jk} = \frac{\sum_{t=1}^T \zeta_t(j, k) (x_t - \bar{\mu}_{jk})(x_t - \bar{\mu}_{jk})^t}{\sum_{t=1}^T \zeta_t(j, k)}. \quad (2.17)$$

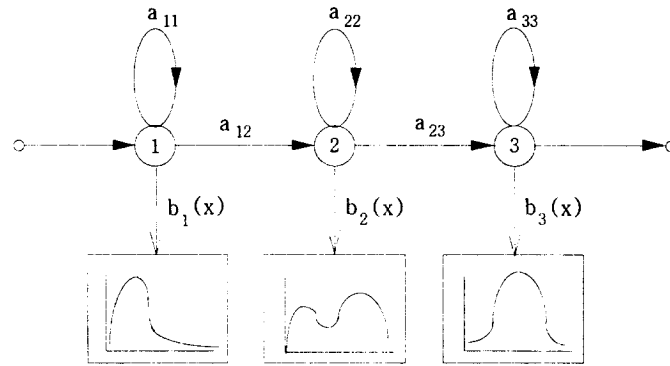
### 3. 은닉마코프모형 인식과정의 병렬화 실험

연속형 HMM은 이산형 HMM보다 인식률은 향상되나 인식과정에서 각 상태에 따르는 혼합정규분포를 사용하므로 상태의 수와 구성분포의 수가 조금만 증가해도 계산량이 매우 커지므로 실시간 음성인식에 문제가 많다. 이를 해결하기 위하여 연속형 HMM을 병렬 processor에 분산시키고 각 processor는 할당된 HMM에 대한 출력확률 계산과 Viterbi 알고리즘 계산을 담당하도록 하여 인식속도를 증가하는 병렬화 연구를 수행하였다.

본 연구에서는 한국어 음소-변이음 변환규칙에 의거한 변이음 기반의 HMM에 의한 음소인식기를 병렬 컴퓨터에서 실행하는 것을 목표로 하며, 인식 결과는 인식된 각각의 변이음들에 해당하는 음소부호로 출력하고 인식성능평가수단인 Correct와 Accuracy로 나타낸다. processor의 개수와 HMM 개수 증가에 따른 인식속도증가율을 측정하여 실시간 음성인식시스템에 대한 병렬화방안에 대한 실질적인 근거를 찾아낸다. 음성특징 분석을 위한 벡터의 생성은 mel-frequency cepstral 계수(MFCC)를 사용하였다. 사용한 음성자료는 KOREAN SPEECH DB(ETRI Wonkwang SPEECH DB)중 컴퓨터 비서 에이전트 영역 자료를 이용하였으며, 남성화자 56명이 1인당 77문장씩 발음한 음성자료를 훈련하고, 인식 평가를 위하여 남성화자 6명이 1인당 77문장씩 발음한 음성자료를 사용하였다. 또한 음소 모델링을 위하여 사용된 변이음은 46개로 하였다.

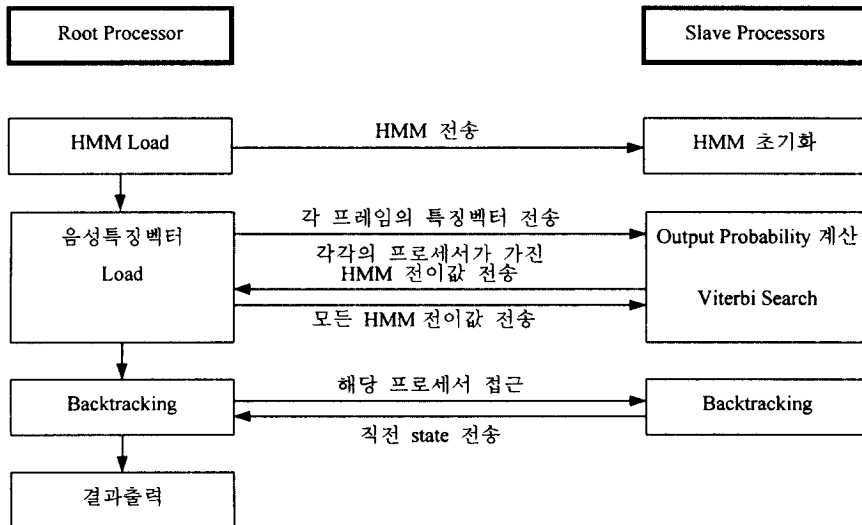
사용된 병렬처리시스템은 뛰어난 다중 Transputer 시스템으로, 전체 17개의 processor중 1개는 root processor로 병렬컴퓨터에서 인식이 진행되는 동안 interface로 사용되며, 16개는 인식과정에

사용되는 processor이다. 상태 전치구조는 [그림 4]에서 보듯이 음성인식에서 많이 사용되는 오른쪽 흐름(left-to-right)을 사용하고 3개의 상태와 각 상태마다 3개 정규분포로 구성된 혼합분포를 고려하였다.



[그림 4] left-to-right 구조

실험에서 제안한 병렬 음소인식 알고리즘은 [그림 5]와 같다. Slave processor들은 root processor에서 전송된 HMM을 초기화 한 후, 각 frame마다 생성되는 음성특징벡터를 이용하여 출력확률을 계산한다. 각각의 slave processor는 자신이 가진 HMM의 전이값을 root processor로 전송하고, root processor는 넘겨진 모든 HMM의 전이값을 취합하여 각 slave processor로 전송한다. Slave processor는 HMM의 전이확률과 자신이 가지고 있는 확률매칭값을 이용하여 Viterbi 과정을 실행한다.



[그림 5] 병렬 음소인식 알고리즘

실험에서 훈련 HMM의 수를 48, 192, 432개의 경우로 구분하였으며, 입력형태는 음성특징추출 벡터이며 출현시간별 발생음소를 결과로 출력하여 인식률을 평가하였다. 인식률과 정확률은 각각



Correct와 Acc로 나타내며 다음과 같은 수식으로 측정한다.

$$\%Correct = \frac{N - S - D}{N} \times 100 \quad (3.1)$$

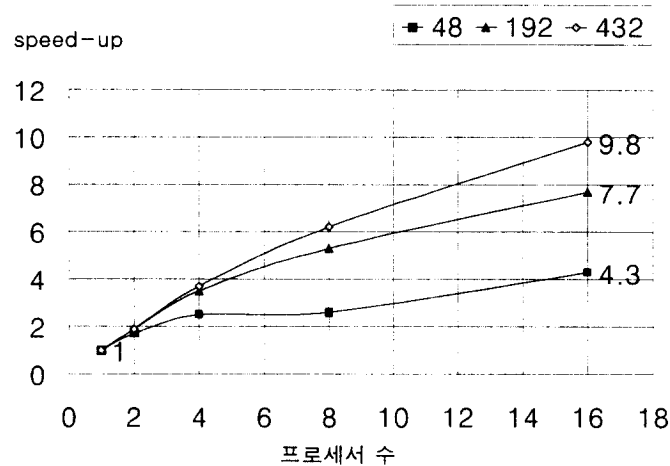
$$\%Acc = \frac{N - S - D - I}{N} \times 100 \quad (3.2)$$

여기서 N은 전체 변이음 개수, D는 삭제된 변이음 개수, S는 대치(substitution)된 변이음 개수, 그리고 I는 삽입된 변이음 개수를 의미한다. 본 실험의 결과에서 구한 인식률 및 정확률의 평균은 Correct=67.30%, Acc=49.37%이며 이러한 결과는 Entropic사의 HMM기반 음성인식 tool인 HTK version 2.0을 사용한 인식 결과와 동일하다.

병렬 계산을 통한 성능은 다음으로 정의 되는 speed-up으로 평가될 수 있다. p개의 processor를 사용할 경우에는 다음과 같다.

$$S_{up} = \frac{\text{1개의 프로세서를 이용했을 때의 수행시간}}{\text{p개의 프로세서를 이용했을 때의 수행시간}} \quad (3.3)$$

실험 결과 [그림 6]과 같이 음성인식과정의 병렬화에 참여하는 processor 수의 증가에 따른 speed-up을 조사하였다.



[그림 6] processor 수의 증가에 따른 speed-up

[그림 6]은 48, 192, 432개의 HMM을 대상으로 음소인식을 각각 실행하였을 경우 인식성능을 평가한 결과로써 processor 수의 증가에 따른 속도향상의 결과를 나타낸다. HMM의 수가 48개인 경우, 16개의 processor를 사용하더라도 속도향상이 4배 정도인 이유는 각각의 processor의 계산

량이 network간의 message passing에 비하여 월등하지 못하기 때문이다. HMM의 수를 증가시킬수록 message passing의 속도는 무시되어 10배에 가까운 속도향상을 가진다. 그러므로 HMM을 이용한 실시간 음성인식 시스템의 구현은 병렬화를 통하여 실시간에 더욱 접근할 수 있으며 실험 결과도 긍정적이라 볼 수 있다.

#### 4. 결 론

음성인식시스템에서 인식의 정확성 뿐만 아니라 인식속도의 실시간화는 매우 중요한 문제이다. 본 연구에서는 인식률을 위하여 연속형 HMM을 사용하였으며, 인식속도의 speed-up을 위하여 병렬화 기법을 제안하였다. 병렬처리시스템을 이용한 한국어 변이음 기반의 음성실험을 통하여 processor 수와 대상 HMM의 수에 따른 speed-up을 조사하였다. 실험 결과 연속 HMM의 병렬화는 대상 HMM의 수가 증가할수록 speed-up이 증가하고, 450 정도가 되도 10배 정도의 향상을 보임을 알 수 있어 인식속도 향상에 병렬화 알고리즘은 우수한 효과를 나타낸다고 생각된다.

음성인식을 포함한 각종 신호패턴인식을 위하여 HMM을 중심으로 한 통계적 모형에 대한 연구 과 응용분야는 매우 넓다고 생각된다. 앞으로의 연구 방향으로서는 인식률을 증가시키기 위하여 기존 HMM에서 사용되는 일단계 마코프연쇄모형을 다단계 마코프연쇄모형으로 확장한 HMM 알고리즘 연구가 필요하다고 생각되며, 이산형 HMM과 연속형 HMM의 단점을 보완한 반연속 HMM을 이용한 병렬화 기법 연구와 HMM 모수 추정에 사용되는 EM 알고리즘의 수렴속도를 향상시키기 위한 연구도 필요하다고 생각된다.

#### 참 고 문 헌

- [1] Bahl, L.R. and Jelinek, F. (1975), Decoding for channels with insertions, deletions and substitutions with applications to speech recognition, *IEEE Transactions on Information theory*, IT-21, 404-411.
- [2] Bahl, L.R., Jelinek, F. and Mercer, R.L. (1975), A maximum likelihood approach to continous speech recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5, 179-190.
- [3] Baker, J.K. (1975), The dragon system - an overview, *IEEE Transactions on Acoustics, Speech, Signal processing*, ASSP-23, 24-29.
- [4] Baum, L.E. and Petrie, T. (1966), Statistical inference for probabilistic functions of finite state Markov chains, *Annals of Mathematical Statistics*, 37, 1554-1563.
- [5] Baum, L.E., Petrie, T., Soules, G. and Weiss, N. (1970), A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Annals of Mathematical Statistics*, 41, 164-171.
- [6] Baum, L.E. (1972), An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov chains, *Inequalities*, 3, 1-8.
- [7] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977), Maximum likelihood from

- incomplete data via the EM algorithm, *Journal of Royal Statistician Society*, 39(1), 1-38.
- [8] Euler, S. and Wolf, D. (1987), Speaker independent isolated word recognition based on continous hidden Markov models using mutidimensional spherically invariant functions, *Digital signal Processing-87*, Florence, Italy, 539-542.
- [9] Huijsen, G. (1996), Parallel Implementation of Hidden Markov Models on the nCUBE2, *Thesis, Alparon report*, 96-03, Delft University of Technology, 1996.
- [10] Jelinek, F., Bahl, L.R. and Mercer, R.L. (1975), Design of a linguistc statistical decoder for the recognition of continous speech, *IEEE Transactions on Information Theory*, IT-21, 250-256.
- [11] Juang, B.H. (1985), Maximum likelihood estimation for mixture mutivariate stochastic observations of Markov chain, *AT&T Technical Journal*, 64, 1235-1249.
- [12] Juang, B.H. and Rabiner, L.R. (1991), Hidden Markov models for speech recognition, *Technometrics*, 33, 251-272.
- [13] Rabiner, L.R. and Juang, B.H. (1986), An introduction to hidden Markov models, *IEEE ASSP Magazine*, 4-16.
- [14] Rabiner, L.R. and Juang, B.H. (1993), *Fundamentals of Speech Recognition*, Prentice-Hall, New Jersey.
- [15] Sohn, K.T. and Baik, J.S. (1997), Estimation in a mixture normal distribution, *Korean journal of computational and applied mathematics*, 4, 223-233.
- [16] Viterbi, A.J. (1967), Error bounds for convolutional codes and an asymptotically optimal decoding algorithm, *IEEE transactions on information theory*, IT-13, 260-267.