

論文94-31B-8-2

은닉마르코프모델의 부합확률연산의 정수화 알고리즘 개발 (I)

(Development of an Integer Algorithm for Computation of the Matching Probability in the Hidden Markov Model (I))

金 鎮 憲*, 金 民 基** , 朴 貴 泰*

(Jin Heon Kim, Min Gi Kim and Gwi Tae Park)

要 約

일정 시간동안 관측된 신호열(O)과 은닉마르코프모델 (HMM : Hidden Markov Model, A)의 부합확률(P(O|A))은 관측된 신호열이 주어진 모델에서 나올 확률로 그 신호의 모델에 대한 부합의 정도를 나타낸다. 이를 이용하면 미지의 신호열이 관측되었을 때 기지의 다수의 신호모델들과 이 관측신호열과의 부합확률들을 구하여 대소를 비교하므로써 관측된 미지의 신호패턴을 인식할 수 있다. 그러나 이 알고리즘은 연산 과정에서 사용된 모든 변수가 부동소수 확률값으로 되어 있기 때문에 고속처리를 위하여 하드웨어로 이 연산을 구현하고자 할 때는 고가의 부동소수 연산자가 소요된다.

본 논문에서는 이 부합확률을 연산해 내는 하드웨어를 경제적으로 구현할 수 있도록 부동소수를 쓰지 않고 양의 정수연산으로 부합확률을 구하는 정수화알고리즘을 제안하였다. 이 알고리즘은 모델파라미터를 양의 정수를 곱해서 정수화하였고 매 단계마다 사용되는 변수를 정규화하는 절차를 통해 데이터의 발산을 막도록 하였다. 최종적으로 유도된 부합확률의 연산식은 상수항과 대수연산이 수반되는 변수항으로 구분되는 데 이 변수항의 연산에 수반되는 대수변환을 위한 테이블을 작게 구성하는 방안도 함께 검토되었다. 제안된 알고리즘의 오차에 대한 정성적인 특성을 알아보기 위해 찾수가 크게 다른 임의의 가상모델 그룹들에 대해 보정상수의 크기, 모델의 찾수, 관측시간의 길이에 따른 오차의 통계적인 특성을 살펴보았다.

Abstract

The matching probability, $P(O|A)$, of the signal sequence(O) observed for a finite time interval with a HMM (Hidden Markov Model, A) indicates the probability that signal comes from the given model. By utilizing the fact that the probability represents matching score of the observed signal with the model, we can recognize an unknown signal pattern by comparing the magnitudes of the matching probabilities with respect to the known models. Because the algorithm, however, uses floating point variables during the computing process, hardware implementation of the algorithm requires floating point units.

This paper proposes an integer algorithm which uses positive integer numbers rather than float point ones to compute the matching probability so that we can economically realize the algorithm into hardware. The algorithm makes the model parameters integer numbers by multiplying positive constants and prevents from divergence of data through the normalization of variables at each step. The final equation of matching probability is composed of constant terms and a variable term, which contains logarithm operations. A scheme to make the log conversion table smaller is also presented. To analyze the qualitative characteristics of the proposed algorithm we attach simulation result performed on two groups of 10 hypothetic models respectively and inspect the statistical properties with respect to the model order, the magnitude of scaling constants, and the effect of the observation length.

*正會員, 高麗大學校 電氣工學科
(Dept. of Electrical Eng., Korea Univ.)

**正會員, 高麗大學校 醫用電子工學科

(Dept. of Medical Electronics Eng., Korea Univ.)
接受日字 : 1993年 7月 22日

I. 서론

1960년 중반에 Baum과 Welch에 의해 정립된 은닉마르코프모델은 신호처리계통의 이론적 해석의 기초가 되는 신호모델중의 하나로 신호에 대한 경험적 지식이 주어진 데이터의 관찰만으로 그 신호군에 대한 확률적인 모델링이 가능하여 최근에는 음성인식을 비롯하여 텍스트, 형상, 문자인식등 신호처리의 여러분야에 걸쳐 점차 그 응용범위를 넓혀가고 있다. Rabiner가 그의 논문^[1]에서 제시한 은닉마르코프모델의 3가지 문제들중 문제1은 임의의 시간 τ 동안 관측된 신호열 O 와 알고 있는 신호모델(λ)의 부합확률($P(O|\lambda)$)을 구하는 문제로 이 확률은 관측된 신호와 알고 있는 여러 모델과의 일치여부를 판별하여 신호의 인식작업에 이용할 수 있다.^[2,3]

이러한 부합확률에 의한 인식방법은 분류모델이 여러개 있을 경우 각 모델에 대한 부합확률을 모두 구해야하기 때문에 모델의 갯수가 많아짐에 따라 계산량이 증대하여 최종 인식결과를 얻기까지 많은 시간이 소요된다. 고속으로 인식결과를 얻기 위해서는 다수의 프로세서를 사용하여 각 프로세서가 하나의 모델을 전담케 하거나 부합확률을 구하는 알고리즘을 하드웨어로 구현하여 이를 병렬로 운전시키는 방안이 바람직하다. 이중 후자의 방안은 각 부합확률을 동시에 구할 수 있을 뿐만 아니라 하나의 부합확률을 구하는 과정안에서도 병렬화가 가능하기 때문에 처리속도를 극대화할 수 있는 가장 근본적인 접근방식이라 할 수 있다. 그러나 이러한 하드웨어적인 접근 방식은 병렬성을 고도화함에 따라 고가의 부동소수연산자를 다수 필요하게 되므로 현실적으로 경제성이 떨어지는 단점이 있다.

본 논문에서는 은닉마르코프모델의 부합확률을 구하는 알고리즘을 정수화^[4]하여 부동소수연산자를 쓰지 않고 저렴한 정수 연산자를 이용하여 경제적으로 부합확률연산 하드웨어를 구현할 수 있는 정수화 알고리즘을 제안하였다. 이 알고리즘은 부동소수로 표현된 확률이 1보다 작은 값으로만 이루어져 있다는 점에 착안하여 모델파라미터를 일정크기의 보정상수를 곱해서 정수화하여 처리하였고 매 단계마다 사용되는 변수를 정규화하는 절차를 통해 데이터의 발산을 막도록 하였다.

이와 같이 연산과정의 정수화를 위해 기존에 제안된 방법으로는 각 모델파라미터를 대수변환을 취하여 정수화한 Lee^[5]의 알고리즘이 있다. 이 알고리즘은 부동소수의 곱셈연산을 덧셈으로 단순화한 반면 부동소수의 덧셈연산을 몇 단계의 동작이 수반되는 테이

블변환을 통하여 구현하였기 때문에 정밀도 혹은 병렬성을 높이기 위해서 필요한 메모리의 양은 급격히 증가하게 된다. 이들 두가지 기법은 각각의 장단점을 갖고 있다. 예를 들면 Lee의 알고리즘은 곱셈연산이 단순화되는 대신 이로 말미암아 덧셈연산이 복잡해지고 이를 위한 연산테이블이 요구된다. 반면에 제안된 기법은 덧셈연산이 단순한 반면 곱셈연산은 그 이상 단순화되지 않고 연산된 결과에 대해 별도의 대수변환이 필요하게 된다.

본 논문의 II장에서는 기존의 부동소수알고리즘을 간략히 소개하고, III장에서는 이러한 정수화 알고리즘의 유도과정을 보이기로 한다. 최종적으로 유도된 부합확률의 연산식은 상수항과 대수연산이 수반되는 변수항으로 구분되는 데 이 변수항의 연산에 수반되는 대수연산을 작은 규모의 테이블로 구현하기 위한 방안도 함께 검토하기로 한다. IV장에서는 제안된 알고리즘의 오차에 대한 정성적인 특성을 알아보기 위해 임의의 가상모델들에 대해 적용한 시뮬레이션을 통해 보정상수의 크기, 모델의 차수, 관측시간의 길이에 따른 오차의 통계적인 특성을 살펴보고 그 개선 방안을 검토하기로 한다.

II. 부합확률의 부동소수 알고리즘^[6]

은닉마르코프모델 $\lambda(\Pi, A, B)$ 는 다음과 같은 확률집합으로 구성된다.

$$\Pi = \{\pi_i\}, \quad i=1, \dots, N. \quad (1a)$$

$$A = \{a_{ij}\}, \quad i=1, \dots, N, \quad j=1, \dots, N. \quad (1b)$$

$$B = \{b_j(k)\}, \quad j=1, \dots, N, \quad k=1, \dots, M. \quad (1c)$$

여기서 N 은 상태의 수를, M 은 관측심볼의 수를 나타내며, π_i 는 시간 $t=1$ 일 때 상태(q_1)가 i 번째 상태(S_i)일 초기분포확률, a_{ij} 는 시간 t 에서 상태(q_t)가 S_i 이고 시간 $t+1$ 에서 상태(q_{t+1})가 S_j 일 상태전이확률, $b_j(k)$ 는 상태가 S_j 일 때 k 번째 심볼(v_k)이 관측될 심볼관측확률로 각각 아래와 같이 정의된다.

$$\pi_i = P[q_1 = S_i], \quad i=1, \dots, N. \quad (2a)$$

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad i=1, \dots, N, \quad j=1, \dots, N. \quad (2b)$$

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j], \quad j=1, \dots, N, \quad k=1, \dots, M. \quad (2c)$$

부합확률의 연산횟수를 줄이기 위해 다음과 같이 정의되는 순방향변수(forward variable), $\alpha_t(i)$ 를 도입하면.

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i | \lambda), \quad i = 1, \dots, N. \quad (3)$$

시간에 대해 일반화된 순방향변수는 아래와 같이 바로 전 단계의 값으로부터 계산된다.

$$\alpha_1(i) = \pi_i b_i(O_1), \quad i = 1, \dots, N. \quad (4a)$$

$$\alpha_t(j) = \left[\sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(O_t), \quad t = 2, \dots, \tau, \quad j = 1, \dots, M. \quad (4b)$$

시간 τ 동안 O의 신호열이 관측되었을 때 부합확률을 식(3)에서 정의한 순방향변수를 이용하여 구하면 다음과 같다.

$$P(O|\lambda) = \sum_{i=1}^N \alpha_\tau(i). \quad (5)$$

III. 정수화 알고리즘

부동소수로 구성된 모델 λ 에 일 보정상수 I_1 을 곱하여 정수화한 모델 $\tilde{\lambda}(\tilde{\pi}, \tilde{A}, \tilde{B})$ 를 다음과 같이 정의하기로 한다. 여기서 I_1 은 $\tilde{\lambda}$ 를 λ 로 복원할 때 오차가 없도록 충분히 큰 임의의 자연수로 가정한다.

$$\tilde{\pi} = I_1 \cdot \pi, \quad \tilde{A} = I_1 \cdot A, \quad \tilde{B} = I_1 \cdot B. \quad (6)$$

따라서 정수모델에 사용될 파라미터는 각각 다음과 같은 과정을 통해 정수화된다. 이 과정은 오프라인(off line)으로 이루어지기 때문에 연산자나 연산시간이 소요되지는 않는다.

$$\tilde{\pi}_i = I_1 \cdot \pi_i, \quad i = 1, \dots, N. \quad (7a)$$

$$\tilde{a}_{ij} = I_1 \cdot a_{ij}, \quad i = 1, \dots, N, \quad j = 1, \dots, N. \quad (7b)$$

$$\tilde{b}_j(k) = I_1 \cdot b_j(k), \quad j = 1, \dots, N, \quad k = 1, \dots, M. \quad (7c)$$

정수화된 모델 ($\tilde{\lambda}$)의 파라미터 $\tilde{\pi}, \tilde{A}$ 및 \tilde{B} 는 각각 아래 식(8)과 (9)를 만족하는 자연수로 구성된다.

$$0 \leq \tilde{\pi}, \tilde{A}, \tilde{B} \leq I_1. \quad (8)$$

$$\sum_{i=1}^N \tilde{\pi}_i = I_1. \quad (9a)$$

$$\sum_{i=1}^N \tilde{a}_{ij} = I_1, \quad j = 1, \dots, N. \quad (9b)$$

$$\sum_{k=1}^M \tilde{b}_j(k) = I_1, \quad j = 1, \dots, M. \quad (9c)$$

정수화된 순방향변수를 $\overline{\alpha}_t(i)$ 로 표기하기로 하고 이를 1차 순방향변수라 하면 그 초기치 $\overline{\alpha}_1(i)$ 는 식(4a)에 의하여 아래 식과 같이 정의할 수 있다.

$$\overline{\alpha}_1(i) = \tilde{\pi}_i \cdot \tilde{b}_i(O_1), \quad i = 1, \dots, N. \quad (10)$$

$\overline{\alpha}_1$ 를 구하기 위해 부동소수 알고리즘 식(4-b)에서 연산한 방식을 사용하여 $\overline{\alpha}_2$ 를 그대로 사용하면 시간 t 가 증가함에 따라 $\overline{\alpha}_t$ 는 점점 그 값이 커져서 오버플로우(overflow)가 발생하게 된다. 이를 방지하기 위해 $\overline{\alpha}_t(i)$ 가 시간이 증가함에 따라 발산하지 않게 그 크기를 줄여주는 시변보정계수 R_t 를 도입하여 식(4b)를 아래 식과 같이 정수화한다.

$$\overline{\alpha}_t(j) = R_{t-1} \cdot \sum_{i=1}^N \overline{\alpha}_{t-1}(i) \cdot \tilde{a}_{ij} \cdot \tilde{b}_j(O_t), \quad t = 2, \dots, \tau, \quad j = 1, \dots, N. \quad (11)$$

식(11)을 간략하게 표현하기 위해 $R_t \cdot \overline{\alpha}_t(i)$ 를 시간의 전범위에 걸쳐 아래 식(12)와 같이 $\tilde{\alpha}_t(i)$ 로 표기하고 이를 2차 순방향변수라 하자

$$\tilde{\alpha}_t(i) = R_t \cdot \overline{\alpha}_t(i), \quad t = 1, \dots, \tau, \quad i = 1, \dots, N. \quad (12)$$

따라서 1차 순방향변수 $\overline{\alpha}_t$ 는 아래 식과 같이 2차 순방향변수 $\tilde{\alpha}_t$ 와 정수모델의 연산식으로 나타낼 수 있다.

$$\overline{\alpha}_t(j) = \sum_{i=1}^N \tilde{\alpha}_{t-1}(i) \cdot \tilde{a}_{ij} \cdot \tilde{b}_j(O_t), \quad t = 2, \dots, \tau, \quad j = 1, \dots, N. \quad (13)$$

R_t 를 정하기 위해 그 선정기준으로 아래와 같이 모든 i 에 대한 $\tilde{\alpha}_t(i)$ 의 합이 양의 정수 I_2 가 되도록 하면,

$$\sum_{i=1}^N \tilde{\alpha}_t(i) = I_2, \quad t = 1, \dots, \tau. \quad (14)$$

식(12)에 의해 R_t 는 다음과 같이 정의된다.

$$R_t = \frac{I_2}{\sum_{i=1}^N \overline{\alpha}_t(i)}, \quad t = 1, \dots, \tau. \quad (15)$$

이때 $\tilde{\alpha}_t$ 는 식(14)에 의해 아래 식(16)의 범위를 갖

게 되며.

$$0 \leq \tilde{\alpha}_t(i) \leq I_2, \quad t=1, \dots, \tau, \quad i=1, \dots, N. \quad (16)$$

식(9)와 식(14)의 제한 조건에 의해 식(13)의 $\tilde{\alpha}_t$ 는 아래 식과 같이 일정 범위의 값을 갖는다.

$$0 \leq \overline{\alpha}_t(i) \leq I_1^2 \cdot I_2, \quad t=1, \dots, \tau, \quad i=1, \dots, N. \quad (17)$$

즉, R_t 를 식(15)와 같이 정의하면 $\overline{\alpha}_t$ 의 값이 윗식과 같이 일정 범위($I_1^2 \cdot I_2$)의 값을 갖게되어 $\overline{\alpha}_t$ 가 발산하는 것을 막을 수 있다.

식(15)를 식(12)에 대입하면 2차 순방향변수 $\tilde{\alpha}_t$ 와 1차 순방향변수 $\overline{\alpha}_t(i)$ 는 다음과 같은 관계가 성립한다.

$$\tilde{\alpha}_t(i) = \frac{\overline{\alpha}_t(i)}{\sum_{i=1}^N \overline{\alpha}_t(i)} \cdot I_2, \quad t=1, \dots, \tau, \quad i=1, \dots, N. \quad (18)$$

식(18)에서 I_2 는 0과 1사이의 실수 값을 정수화하는 보정상수의 기능을 갖고 있다. 따라서 정수화 과정에서 생기는 오차를 방지하기 위해 I_2 의 크기는 그 실수 값을 복원해 낼 수 있을 만큼 충분히 크다고 가정하기로 한다. I_2 의 크기를 2의 지수승의 형태로 정하면 I_2 를 곱하기 위한 곱셈연산자나 연산시간의 소요는 막을 수 있다.

식의 전개상의 편의를 위해 식(18)의 분모항을 아래와 같이 K_t 라 표기하기로 한다.

$$K_t = 1 / \sum_{i=1}^N \overline{\alpha}_t(i), \quad t=1, \dots, \tau. \quad (19)$$

K_t 의 정의를 식(18)에 적용해 $\tilde{\alpha}_t$ 와 $\overline{\alpha}_t$ 의 관계를 아래 식에 보였다.

$$\tilde{\alpha}_t(i) = K_t \cdot I_2 \cdot \overline{\alpha}_t(i), \quad t=1, \dots, \tau, \quad i=1, \dots, N. \quad (20)$$

정리 1 : 보정상수(I_1, I_2)가 충분히 클 때 식(10), (13), (18) 및 식(19)에서와 같이 $\overline{\alpha}_t(i)$, $\overline{\alpha}_t(i)$, $\tilde{\alpha}_t(i)$ 및 K_t 를 정의하면 $\tilde{\alpha}_t(i)$ 와 $\alpha_t(i)$ 는 다음 관계를 갖는다

$$\tilde{\alpha}_t(j) = (I_2 \cdot I_1^2)^t \cdot \prod_{i=1}^t K_i \cdot \alpha_t(j), \quad t=1, \dots, \tau, \quad j=1, \dots, N.$$

증명)

정리 1은 아래와 같이 수학적 귀납법을 사용하여 증명된다.

식(10)에서 $\overline{\alpha}_1(i)$ 와 $\alpha_1(i)$ 의 관계는 아래 식(21)과 같다.

$$\begin{aligned} \overline{\alpha}_1(i) &= (I_1 \cdot \pi_i) \cdot (I_1 \cdot b_i(O_1)) \\ &= I_1^2 \cdot \alpha_1(i), \quad i=1, \dots, N. \end{aligned} \quad (21)$$

$t=1$ 일 때 식(18)에 식(19)와 식(21)을 적용하면,

$$\begin{aligned} \tilde{\alpha}_1(i) &= \frac{1}{\sum_{i=1}^N \overline{\alpha}_1(i)} \cdot I_2 \cdot \overline{\alpha}_1(i) \\ &= K_1 \cdot I_2 \cdot I_1^2 \cdot \alpha_1(i), \quad i=1, \dots, N. \end{aligned} \quad (22)$$

$t=1$ 을 정리 1에 대입하면,

$$\tilde{\alpha}_1(j) = I_2 \cdot I_1^2 \cdot K_1 \cdot \alpha_1(j), \quad j=1, \dots, N. \quad (23)$$

따라서 정리 1은 $t=1$ 일 때 성립한다.

$t=m$ 일 때 정리 1이 성립한다고 가정하면,

$$\tilde{\alpha}_m(i) = (I_2 \cdot I_1^2)^m \cdot \prod_{i=1}^m K_i \cdot \alpha_m(i), \quad i=1, \dots, N. \quad (24)$$

식(24)의 양변에 $\tilde{a}_j \cdot \tilde{b}_j(O_{m+1})$ 를 곱하면,

$$\begin{aligned} \tilde{\alpha}_m(i) \cdot \tilde{a}_j \cdot \tilde{b}_j(O_{m+1}) &= (I_2 \cdot I_1^2)^m \cdot \prod_{i=1}^m K_i \cdot \alpha_m(i) \cdot \tilde{a}_j \cdot \tilde{b}_j(O_{m+1}), \\ i=1, \dots, N, \quad j=1, \dots, N. \end{aligned} \quad (25)$$

식(25)의 양변에 대하여 모든 i 에 대한 합산을 취하면,

$$\begin{aligned} \text{좌변} &= \sum_{i=1}^N \tilde{\alpha}_m(i) \cdot \tilde{a}_j \cdot \tilde{b}_j(O_{m+1}) \\ &= \tilde{\alpha}_{m+1}(j), \quad j=1, \dots, N. \end{aligned} \quad (26)$$

$$\begin{aligned} \text{우변} &= \sum_{i=1}^N \left\{ (I_2 \cdot I_1^2)^m \cdot \prod_{i=1}^m K_i \cdot \alpha_m(i) \cdot (I_1 \cdot a_j) \cdot (I_1 \cdot b_j(O_{m+1})) \right\} \\ &= (I_2 \cdot I_1^2)^m \cdot I_1^2 \cdot \prod_{i=1}^m K_i \cdot \sum_{i=1}^N \alpha_m(i) \cdot a_j \cdot b_j(O_{m+1}) \\ &= (I_2 \cdot I_1^2)^m \cdot I_1^2 \cdot \prod_{i=1}^m K_i \cdot \alpha_{m+1}(j), \quad j=1, \dots, N. \end{aligned} \quad (27)$$

따라서,

$$\tilde{\alpha}_{m+1}(j) = (I_2 \cdot I_1^2)^m \cdot I_1^2 \cdot \prod_{i=1}^m K_i \cdot \alpha_{m+1}(j), \quad j=1, \dots, N. \quad (28)$$

$t=m+1$ 일 때 식(20)의 우변에 식(28)을 대입하면,

$$\begin{aligned} \bar{\alpha}_{m+1}(j) &= K_t \cdot I_2 \cdot (I_2 \cdot I_1^2)^m \cdot I_1^t \cdot \prod_{i=1}^m K_i \cdot \alpha_{m+1}(j) \\ &= (I_2 \cdot I_1^2)^{m+1} \cdot \prod_{i=1}^{m+1} K_i \cdot \alpha_{m+1}(j), \quad j=1, \dots, N. \end{aligned} \quad (29)$$

식(29)는 정리 1에 $t=m+1$ 을 대입한 것과 같으므로 정리 1은 임의의 자연수 $t (\geq 1)$ 에 대하여 성립한다.

정리 2 : 보정상수(I_1, I_2)가 충분히 클 때 식(10), (13), (18) 및 식(19)에서와 같이 $\bar{\alpha}_1(j), \bar{\alpha}_t(j), \bar{\alpha}_\tau(j)$ 및 K_t 를 정의하면 $\bar{\alpha}_t$ 와 α_t 는 다음 관계를 갖는다

$$\bar{\alpha}_t(j) = \begin{cases} (I_1)^2 \cdot \alpha_t(j), & t=1, \quad j=1, \dots, N. \\ (I_2)^{t-1} \cdot (I_1^2)^t \cdot \prod_{i=1}^{t-1} K_i \cdot \alpha_t(j), & t=2, \dots, \tau, \quad j=1, \dots, N. \end{cases}$$

증명)

정리 2는 아래와 같이 수학적 귀납법을 사용하여 증명된다.

$t=1$ 일 때는 식(21)에 의해서 성립한다.

$t=2$ 일 때 식(13)에 식(22)의 $\bar{\alpha}_1(i)$ 를 대입하면,

$$\begin{aligned} \bar{\alpha}_2(j) &= \sum_{i=1}^N \bar{\alpha}_1(i) \cdot \bar{a}_{ij} \cdot \bar{b}_j(O_2) \\ &= \sum_{i=1}^N (K_1 \cdot I_2 \cdot I_1^2 \cdot \alpha_1(i)) \cdot (I_1 \cdot a_{ij}) \cdot (I_1 \cdot b_j(O_2)) \\ &= K_1 \cdot I_2 \cdot I_1^4 \cdot \sum_{i=1}^N \alpha_1(i) \cdot a_{ij} \cdot b_j(O_2) \\ &= K_1 \cdot I_2 \cdot I_1^4 \cdot \alpha_2(j), \quad j=1, \dots, N. \end{aligned} \quad (30)$$

$t=2$ 를 정리 2에 대입하면,

$$\bar{\alpha}_2(j) = K_1 \cdot I_2 \cdot I_1^4 \cdot \alpha_2(j), \quad j=1, \dots, N. \quad (31)$$

따라서 $t=2$ 일 때 정리 2는 성립한다.

$t=m(m \geq 2)$ 일 때 정리 2가 성립한다고 가정하면,

$$\bar{\alpha}_m(i) = (I_2)^{m-1} \cdot (I_1^2)^m \cdot \prod_{i=1}^{m-1} K_i \cdot \alpha_m(i), \quad i=1, \dots, N. \quad (32)$$

식(32)의 양변에 $K_m \cdot I_2$ 를 곱하면 식(20)에 의해,

$$\begin{aligned} \text{좌변} &= K_m \cdot I_2 \cdot \bar{\alpha}_m(i) \\ &= \bar{\alpha}_m(i), \quad i=1, \dots, N. \end{aligned} \quad (33)$$

$$\text{우변} = (I_2)^m \cdot (I_1^2)^m \cdot \prod_{i=1}^m K_i \cdot \alpha_m(i), \quad i=1, \dots, N. \quad (34)$$

따라서,

$$\bar{\alpha}_m(i) = (I_2)^m \cdot (I_1^2)^m \cdot \prod_{i=1}^m K_i \cdot \alpha_m(i), \quad i=1, \dots, N. \quad (35)$$

식(35)의 양변에 $\bar{a}_{ij} \cdot \bar{b}_j(O_{m+1})$ 을 곱하고 모든 j 에 대하여 합산을 취하면,

$$\begin{aligned} \text{좌변} &= \sum_{i=1}^N \bar{\alpha}_m(i) \cdot \bar{a}_{ij} \cdot \bar{b}_j(O_{m+1}) \\ &= \bar{\alpha}_{m+1}(j), \quad j=1, \dots, N. \end{aligned} \quad (36)$$

$$\begin{aligned} \text{우변} &= (I_2)^m \cdot (I_1^2)^m \cdot \prod_{i=1}^m K_i \cdot \sum_{i=1}^N \alpha_m(i) \cdot \bar{a}_{ij} \cdot \bar{b}_j(O_{m+1}) \\ &= (I_2)^m \cdot (I_1^2)^{m+1} \cdot \prod_{i=1}^m K_i \cdot \alpha_{m+1}(j), \quad j=1, \dots, N. \end{aligned} \quad (37)$$

따라서,

$$\bar{\alpha}_{m+1}(j) = (I_2)^m \cdot (I_1^2)^{m+1} \cdot \prod_{i=1}^m K_i \cdot \alpha_{m+1}(j), \quad j=1, \dots, N. \quad (38)$$

식(38)은 정리 2에 $t=m+1$ 을 대입한 결과와 같으므로 정리 2는 임의의 자연수 $t (\geq 2)$ 에 대해 성립한다.

정리 1을 이용하여 정수화된 파라미터와 1,2차 순방향변수를 통해 부합확률을 구할 수 있다. 정리 1의 양변을 모든 j 에 대하여 합산을 취하면,

$$\sum_{j=1}^N \bar{\alpha}_t(j) = (I_2 \cdot I_1^2)^t \cdot \prod_{i=1}^t K_i \cdot \sum_{j=1}^N \alpha_t(j), \quad j=1, \dots, N. \quad (39)$$

식(14)를 위의 식의 좌변에 적용하면,

$$I_2 = (I_2 \cdot I_1^2)^t \cdot \prod_{i=1}^t K_i \cdot \sum_{j=1}^N \alpha_t(j), \quad t=1, \dots, \tau. \quad (40)$$

위의 식에서 $t=\tau$ 라 두면 식(5)에 의해 부합확률은 아래와 같이 유도된다.

$$P(O|\lambda) = \frac{I_2}{(I_2 \cdot I_1^2)^t \cdot \prod_{i=1}^t K_i}. \quad (41)$$

위의 식은 양의 정수만으로 부합확률을 구하는 최 종식이라 할 수 있으나 이 값은 t 가 증가함에 따라 급격히 분모항이 증가하므로 언더플로우(under flow)를 유발하게 된다. 위의 식의 양변에 아래와 같이 밑수가 b 인 대수를 취하면 이를 방지할 수 있다.

$$\text{Log}_b P(O|\lambda) = (1-\tau) \text{Log}_b I_2 - 2\tau \text{Log}_b I_1 - \sum_{i=1}^{\tau} \text{Log}_b K_i. \quad (42)$$

여기서 K_i 의 값은 1보다 작으므로 하드웨어 구현이 어렵다. 따라서 다음과 같이 K_i 의 역수, H_i 를 도입하면,

$$\begin{aligned} H_i &= 1 / K_i \\ &= \sum_{i=1}^N \bar{\alpha}_i(i), \quad i=1, \dots, \tau. \end{aligned} \quad (43)$$

식(42)는 다음과 같이 하드웨어 구현이 보다 쉬운 식으로 바뀐다.

$$\text{Log}_b P(O/\lambda) = (1-\tau)\text{Log}_b I_2 - 2\tau \text{Log}_b I_1 + \sum_{i=1}^T \text{Log}_b H_i. \quad (44)$$

위의 식에서 첫번째와 두번째항은 상수항이기 때문에 오프라인으로 연산이 가능하며 이들 항은 부합확률의 크기를 이용하여 인식 작업을 하고자 할 때는 공통항이 되므로 연산할 필요가 없다. 변수항은 대수 연산을 수반하는 데 이는 정수 값을 출력하도록 원래의 대수 값에 상수 I_3 를 곱해준 값으로 보정된 테이블을 이용하면 쉽게 구현할 수 있다. N_{input} 비트를 입력으로 하는 N_{width} 넓이의 변환테이블의 경우 I_3 를 아래 식과 같이 정의하면 대수 출력을 정수화할 수 있을 뿐만 아니라 테이블 메모리의 전 공간(N_{input})을 활용할 수 있다.

$$I_3 = (2^{N_{input}} - 1) / \log_b(2^{N_{input}} - 1). \quad (45)$$

이때 필요한 메모리의 규모는 원래 H_i 의 비트 크기에 따라 배수로 증가하지만 대수변환 값은 인수가 증가함에 따라 그 크기가 점차 포화되어가는 대수 곡선의 출력 특성을 이용하면 적은 규모의 메모리를 가지고도 거의 같은 성능을 유지할 수 있다. 즉 H_i 의 값이 작을 때는 H_i 의 미세 변화가 $\text{Log}_b H_i$ 의 값에 큰 영향을 미치므로 H_i 데이터의 하위비트 정보들을 사용하고, H_i 가 클 때는 $\text{Log}_b H_i$ 의 값에 큰 영향을 주는 H_i 의 상위비트만을 유효비트로 선택하면 테이블의 규모를 줄이면서도 근사화된 결과를 얻을 수 있다. 테이블을 일정한 크기를 가진 다수의 페이지로 나누어, H_i 데이터의 MSB(most significant bit)부터 시작하여 1이 처음 나오기까지의 이동비트수를 테이블내의 페이지를 지정하는 어드레스로 사용하고, 그 비트 이후 일정수의 비트만큼을 오프셋(offset) 어드레스로 활용하면 이를 간단하게 구현할 수 있다.¹⁶⁾

예를 들어 총 24비트의 H_i 데이터 중에서 오프셋 어드레스 정보로 8비트만을 선택하기로 하면 상위 16개 비트중 어느 하나에 처음으로 1이 세트되는 경우에 대비한 16개의 페이지와 그중 어느 것에도 1이 없을 경우를 대비한 1개의 페이지를 합하여 총 17개의 페이지가 필요하다. 따라서 이 테이블이 16비트의 정수화된 대수값을 출력한다고 가정하면 8.704바이트의 용량이 필요하다. 원래 H_i 의 데이터 비트를 모두 테이블의 입력 어드레스로 사용하는 경우에는 32MB가 필요하다. 용량을 줄이기 위해 H_i 의 상위 16비트만 사용하여 근사적으로 처리할 경우 원래 128KB가 소

요되어야 하나 이 방법을 사용하면 적은 용량으로도 표준편차 1.8×10^{-7} , 평균 오차의 백분률 0.0095%의 정밀도를 유지할 수 있다.

IV. 시뮬레이션

제안된 알고리즘은 I_1 과 I_2 가 정수화과정에서 생기는 오차를 방지하기 위해 충분히 클것을 가정하였으나 이들의 크기는 곧 하드웨어의 규모를 결정하므로 실제로는 그 크기를 제한하지 않을 수 없다. 이에 따라 부동소수 알고리즘에서 구한 값과 오차가 발생하게 된다. 여기서는 찾수가 크게 다른 가상의 모델 2개 그룹을 선정하여 표 1과 같은 보정상수(I_1, I_2) 크기와 τ 의 변화에 따라 오차가 어떻게 변화하는가 검토하였다.

표 1. 보정상수의 크기 및 2차 순방향변수의 비트수

Table 1. The magnitude of scaling constants & bit size of the second forward variable.

case	I_1	I_2	bit size of $\tilde{\alpha}$	bit size of $\tilde{\lambda}$
case 1	2^8	2^8	8	8
case 2	2^8	2^{16}	8	16
case 3	2^{16}	2^{16}	16	8
case 4	2^{16}	2^{16}	16	16

시뮬레이션에 사용된 첫번째 그룹의 모델은 N=3, M=4, 두번째 그룹의 모델은 N=32, M=16의 찾수를 가지며 각 그룹은 10개의 모델로 이루어져 있어 100개의 관측신호에 대해 통계적인 값을 산출하도록 하였다. 즉 각 모델그룹마다 T(1000)번의 부합확률 연산되어 오차분석의 일반성을 유지하도록 하였다.

모델그룹 g에 속한 h번째의 모델을 s_h^g , g와 같은 차수의 임의의 모델을 사용하여 얻은 길이가 τ 인 T개의 관측신호열의 집합을 $O^{s, \tau}$, 그 중 s번째 관측신호열을 $O_t^{s, \tau}$, 그 중 시간이 t일 때 관측된 신호를 $O_t^{s, \tau}$ 로 표기하기로 하자.

본 실험에서는 2개 그룹에 대하여 τ 가 각각 10, 100, 1000인 6개의 관측신호열집합 $O^{1,100}$, $O^{1,1000}$, $O^{1,10000}$ 과 $O^{2,100}$, $O^{2,1000}$, $O^{2,10000}$ 을 만들어서 각 그룹에 속한 모델 λ_h^g (g=1,2, h=1,...,10)에 인가하였다. 이때 정수화 알고리즘에 의한 부합확률의 계산은 식

(44)에서 b를 10으로 한 식을 사용하였으며 부동소수연산은 식(4), (5)에 언더플로우(under flow)를 방지하기 위한 스케일링 기법^[1]을 적용하였다. 정수화 알고리즘에서 나눗셈 연산은 정수연산자로 수행함을 가정하며 소수점이하 버림 방식을 사용하였으며 대수연산은 III장에서 언급한 방식대로 테이블을 구성하면 정밀도에 따라 적당한 크기의 메모리를 통하여 구현할 수 있겠지만 여기서는 보정상수 및 τ 가 정밀도에 끼치는 영향을 고찰하기 위해 오차가 무시될 만큼 큰 규모의 정수화된 변환표를 이용하여 실현하였다고 가정하였다. 향후 하드웨어의 제한성을 고려하여 시뮬레이션 중에 정수화 알고리즘의 중간변수의 크기는 모두 32비트 이하로 제한하였으며 정확한 비교를 위해 부동소수 알고리즘은 64비트의 배정도(double precision)를 사용하였다.

분석을 위해 사용된 평가 지표는 정수연산에 의한 부합확률 최대오차백분률, 정수연산에 의한 부합확률의 표준편차백분률 및 정수연산에 의한 부합확률의 단위시간당 평균오차로 이에 대한 의미는 아래와 같다.

최대오차백분률 :

$$\Omega^{(k;\tau)} = \frac{\text{Max}\{P_{err}(s, O^{(k;\tau)}|\lambda_h^g), \dots, P_{err}(s, O^{(k;\tau)}|\lambda_h^g)\}}{|P_{avg}(s, O^{(k;\tau)}|\lambda_h^g)|_{float}} \cdot 100\% \quad (46)$$

$$\text{표준편차백분률} : \delta^{(k;\tau)} = \frac{\sqrt{\frac{1}{10} \sum_{h=1}^{10} \frac{1}{T} \sum_{s=1}^T P_{err}(s, O^{(k;\tau)}|\lambda_h^g)^2}}{|P_{avg}(s, O^{(k;\tau)}|\lambda_h^g)|_{float}} \cdot 100\% \quad (47)$$

$$\text{단위시간 평균오차} : \xi^{(k;\tau)} = \Xi^{(k;\tau)} / \tau \quad (48)$$

여기서 Max는 { }의 인수중에서 가장 큰 값을 취하는 연산자이며, 윗식에서 사용된 기호에 대한 정의는 아래와 같다.

순시오차 :

$$P_{err}(s, O^{(k;\tau)}|\lambda_h^g) = P(s, O^{(k;\tau)}|\lambda_h^g)_{float} - P(s, O^{(k;\tau)}|\lambda_h^g)_{integer} \quad (49)$$

부동소수 평균부합확률 :

$$P_{avg}(s, O^{(k;\tau)}|\lambda_h^g)_{float} = \frac{1}{10} \sum_{h=1}^{10} \frac{1}{T} \sum_{s=1}^T P(s, O^{(k;\tau)}|\lambda_h^g)_{float} \quad (50)$$

$$\text{평균오차} : \Xi^{(k;\tau)} = \frac{1}{10} \sum_{h=1}^{10} \frac{1}{T} \sum_{s=1}^T P_{err}(s, O^{(k;\tau)}|\lambda_h^g) \quad (51)$$

또한 첨자로 표시된 float 및 integer는 각각 부동소수 알고리즘과 정수화 알고리즘이 사용되었음을 뜻한다.

표 2. 부합확률의 최대오차분률

Table 2. Percentage of the maximum error of matching probabilities.

models	case	case 1	case 2	case 3	case 4
group 1	$\tau=10$	2.323546e+000	2.016998e+000	5.336253e-001	1.157311e-002
	$\tau=100$	2.084047e+000	1.643867e+000	4.827342e-001	7.719743e-003
	$\tau=1000$	1.770807e+000	1.520323e+000	4.473884e-001	9.417606e-003
group 2	$\tau=10$	7.427698e+000	5.487420e+000	2.256206e+000	2.747522e-002
	$\tau=100$	6.173337e+000	3.940263e+000	2.345089e+000	2.331462e-002
	$\tau=1000$	5.873356e+000	3.671392e+000	2.297655e+000	2.284322e-002

표 3. 부합확률의 표준편차백분률

Table 3. Percentage of standard deviation of the matching probability error.

models	case	case 1	case 2	case 3	case 4
group 1	$\tau=10$	1.352355e+000	1.027112e+000	3.700568e-001	6.006339e-003
	$\tau=100$	1.398993e+000	1.042874e+000	4.024762e-001	6.096468e-003
	$\tau=1000$	1.448366e+000	1.068851e+000	4.052796e-001	6.556886e-003
group 2	$\tau=10$	5.379047e+000	3.512064e+000	1.989021e+000	2.162713e-002
	$\tau=100$	5.634831e+000	3.504800e+000	2.228806e+000	2.221442e-002
	$\tau=1000$	5.648921e+000	3.490975e+000	2.252691e+000	2.218879e-002

표 4. 부합확률의 단위시간당 평균오차

Table 4. Average of the matching probability error per unit time.

models	case	case 1	case 2	case 3	case 4
group 1	$\tau=10$	8.384926e-003	6.337508e-003	2.296806e-003	3.717076e-005
	$\tau=100$	8.719698e-003	6.473129e-003	2.514029e-003	3.805375e-005
	$\tau=1000$	9.239630e-003	6.887280e-003	2.599518e-003	4.153905e-005
group 2	$\tau=10$	6.469468e-002	4.216246e-002	2.395389e-002	2.602251e-004
	$\tau=100$	6.806113e-002	4.232209e-002	2.692578e-002	2.683579e-004
	$\tau=1000$	6.842358e-002	4.227689e-002	2.729023e-002	2.687906e-004

위의 시뮬레이션 결과를 분석하여 보면 다음과 같은 사실을 발견할 수 있다.

- ① 보정상수 I_1 과 I_2 의 값이 클수록 오차의 지표는 점점 감소한다. 이것은 보정상수의 값이 클 수록 정수화 과정에서 생기는 소수점이하 버림 오차를 줄일 수 있다는 사실에서 당연시 된다. 특히 연산과정에는 $\tilde{\lambda}$ 보다는 $\tilde{\alpha}$ 관련항이 많이 사용되기 때문에 I_2 보다는 I_1 이 정밀도에 더욱 큰 영향을 미침을 알 수 있다. 모델의 차수가 커지면 오차없이 정수화하기 위해서는 더욱 큰 보정 상수가 요구되기 때문에 같은 크기의 보정상수에 대해서는 모델의 차수가 커지면 오차지표가 증가한다.
- ② 임의의 모델그룹에 대하여 τ 가 커질 수록 오차의 절대적인 크기는 증가하지만 오차지표의 상대적인 크기는 τ 에 크게 영향받지 않는다. 이것은 각 단계마다 오차가 누적되면 오차의 크기는 증가하지만 최종값에 대하여 그 비율이 크게 바뀌지 않는 범위를 유지하고 있음을 뜻하는 데 이

와 같은 현상은 일정비율의 오차범위를 정할 수 있게 하므로 실용적 측면에서 바람직한 특성이라고 할 수 있다.

위의 시뮬레이션에서는 가정된 바와 같이 충분히 큰 보정상수를 쓰지 않고 소수이하버림 방식을 사용하여 정수화하였기 때문에 1,2차 순방향변수들은 무한 크기의 보정상수를 사용할 때 보다 비례적으로도 작은 값을 갖게 된다. 이에 따라 식(43)에서 H_t 도 원래의 값보다 작은 값을 갖는다. 이렇게 유한크기의 보정상수를 사용하여 연산된 $\sum_{i=1}^n \bar{\alpha}_i(i)$ 라 H' 표기하면 아래 관계가 성립한다.

$$0 \leq H_t \leq H_t', \quad t=1, \dots, \tau. \quad (52)$$

보정상수의 크기를 유한 혹은 무한으로 설정할 때 이에 따른 부합확률을 각각 $P(O^{(k;\tau)}|\lambda_h^g)_{\text{limit}}$, $P(O^{(k;\tau)}|\lambda_h^g)_{\text{unlimit}}$ 라고 표기하면 식(49)의 순시오차 $P_{err}(O^{(k;\tau)}|\lambda_h^g)$ 는 부동소수로 표기하여 생기는 오차를 무시한다면 다음과 같이 표현할 수 있다.

$$P_{err}(O^{(k;\tau)}|\lambda_h^g) = P(O^{(k;\tau)}|\lambda_h^g)_{\text{unlimit}} - P(O^{(k;\tau)}|\lambda_h^g)_{\text{limit}} \quad (53)$$

임의의 s, g와 h 에 대해,

$$P(O^{(s;\tau)}|\lambda_h^g)_{\text{unlimit}} = (1-\tau) \text{Log}_b I_2 - 2\tau \text{Log}_b I_1 + \sum_{i=1}^{\tau} \text{Log}_b H_i \quad (54)$$

$$P(O^{(s;\tau)}|\lambda_h^g)_{\text{limit}} = (1-\tau) \text{Log}_b I_2 - 2\tau \text{Log}_b I_1 + \sum_{i=1}^{\tau} \text{Log}_b H_i' \quad (55)$$

따라서 순시오차 $P_{err}(O^{(s;\tau)}|\lambda_h^g)$ 는 아래 식을 만족한다.

$$P_{err}(O^{(s;\tau)}|\lambda_h^g) = \sum_{i=1}^{\tau} \text{Log}_b H_i - \sum_{i=1}^{\tau} \text{Log}_b H_i' \quad (56)$$

식(52)에서 $H_t \geq H_t'$ 이므로,

$$P_{err}(O^{(s;\tau)}|\lambda_h^g) \geq 0 \quad (57)$$

실제로 순시오차는 시뮬레이션 중에 항상 0보다 큰 값을 유지함을 확인하였다. 이같은 사실은 정수연산의 결과가 평균오차를 중심으로 산재하여 있기 때문에 실제로 사용할 때는 평균오차를 미리 예측하여 그 만큼을 보상해 주면 정수연산의 정밀도를 높일 수 있음을 암시한다. 표 5.6은 이러한 특성을 이용하여 평균오차 만큼을 각 연산 결과에서 보상하여 얻은 결과에 대한 최대오차의 백분률과 표준편차의 백분률을 보인 것이다. 이들 표는 정수연산결과가 실제 값과 바이어스된(biased) 오차를 갖고 있기 때문에 그 바

이어스 값을 보상해 주면 오차가 훨씬 작아짐을 보여주고 있다. 실제로는 이 바이어스 값을 예측해 낼 필요가 없는 데 이는 부합확률이 그 절대적 크기보다는 상대적 크기가 인식작업에 많이 활용되고 있기 때문에 같은 찾수로 모델링된 경우 에는 유사한 크기의 바이어스 값을 갖기 때문이다. 따라서 본 알고리즘의 실질적인 정밀도는 표 5.6과 같다고 할 수 있다.

표 5. 평균값 보상에 의한 부합확률의 최대오차 백분률

Table 5. Percentage of the maximum error of matching probabilities compensated by the average value.

models—case	case 1	case 2	case 3	case 4	
group 1	$\tau=10$	9.742651e-001	9.966567e-001	1.669840e-001	5.649063e-003
	$\tau=100$	6.854291e-001	6.064084e-001	1.030993e-001	1.847893e-003
	$\tau=1000$	1.770807e+000	4.430644e-001	5.013290e-002	2.948846e-003
group 2	$\tau=10$	1.956372e+000	1.922850e+000	2.835863e-001	5.888614e-003
	$\tau=100$	5.112572e-001	4.225503e-001	1.326213e-001	1.102946e-003
	$\tau=1000$	2.133534e-001	1.755642e-001	4.402048e-002	6.560458e-004

표 6. 평균값 보상에 의한 부합확률의 표준편차 백분률

Table 6. Percentage of standard deviation of the matching probability errors compensated by the average value.

models—case	case 1	case 2	case 3	case 4	
group 1	$\tau=10$	2.052680e-001	1.849686e-001	5.424428e-002	9.927942e-004
	$\tau=100$	1.719471e-001	1.539810e-001	3.968873e-002	6.457000e-004
	$\tau=1000$	1.448366e+000	1.842856e-001	1.955321e-002	1.072486e-003
group 2	$\tau=10$	3.506912e-001	3.105883e-001	8.909229e-002	1.343578e-003
	$\tau=100$	1.252088e-001	1.111314e-001	3.033487e-002	3.644695e-004
	$\tau=1000$	9.907252e-002	9.113173e-002	1.428062e-002	2.799716e-004

V. 결론

이상에서 보정상수를 도입하여 은닉마르코프모델의 부합확률을 정수연산으로 계산해 내기 위한 정수화 알고리즘을 도출하였다. 도출된 알고리즘은 상수항과 변수항으로 구분되어 있어 대소를 판단하는 일반적인 인식문제의 경우 하드웨어로 구현하는 과정은 변수항에 대해서만 구현해도 된다. 변수항의 연산을 위해 수반되는 대수변환은 페이지에 따라 유효비트의 자릿수가 다른 테이블을 이용하는 방안도 함께 거론되었다. 제안된 알고리즘은 부동소수를 사용하지 않기 때문에 연산자의 규모를 크게 줄일 수 있는 반면 유한한 크기의 보정상수 때문에 오차를 발생하게 되므로 보정상수의 크기는 상황에 따라 적정하게 설정해야 할 것이다. 정수화 연산이 부동소수 알고리즘으로 연

산한 부합확률과 어느 정도의 오차를 보이는지 알아보기 위해 가상모델그룹에 대해 인가한 신호에 대한 부합확률 오차의 통계를 통해 보정상수의 크기, 모델의 차수 및 τ 에 대한 정성적 특성을 살펴보았다. 또한 연산오차가 양의 부호만을 가지고 있다는 사실을 이용하여 정수연산의 정밀도를 주어진 조건(보정상수의 크기) 보다는 높일수 있다는 사실을 보였다. 부합확률을 신호패턴의 분류 목적으로 사용할 경우 여러 모델과의 부합확률들이 서로 근소할 경우 미세한 크기를 따져서 그중 한 모델의 것으로 판별하기 보다는 차라리 판단을 유보하는 것이 타당한 경우가 많은 것을 보면 시뮬레이션에서 보인 오차는 수용할만 하다고 할 수 있다.

제안된 정수화 처리기법은 부합확률을 이용한 신호의 인식과정을 하드웨어로 구현하는 데 활용할 수 있을 것으로 기대된다. 현재 제안된 접근방식에 의거하여 연산과정을 단순화하는 알고리즘의 개발과 각 파라미터의 제한조건들을 이용하여 하드웨어의 규모를 더욱 줄일 수 있는 방안을 연구중에 있다.

參 考 文 獻

[1] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition." *Proc.*

IEEE, vol. 77, No.2, pp. 257-286, Feb. 1989.

[2] 박희선, 이성환, "은닉Markov모델을 이용한 필기체 한글의 오프라인 인식," 제2회 인공지능신경망 및 퍼지시스템 종합학술대회 논문집, pp. 298-309, 1992.

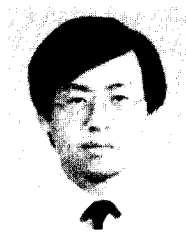
[3] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probalistic functions of a Markov process to automatic speech recognition," *The Bell System Technical Journal*, April 1983.

[4] 김진현, 김민기, 박귀태, "은닉마르코프모델의 부합확률을 구하기 위한 정수화 알고리즘," 제1회 문자인식 워크샵 발표 논문집, pp. 277-283, 1993.

[5] K. F. Lee, *Automatic Speech Recognition : the Development of the SPHINX system*, Kluwer Academic Publishers, pp. 28-29, 1989.

[6] 김진현, 유재규, 김민기, 박귀태, "은닉마르코프모델의 정수화를 이용한 부합확률연산의 H/W 구현," 제6회 신호처리합동학술대회 논문집, pp. 457-460, 1993.

著 者 紹 介



金 鎮 憲(正會員)

1959年 10月 10日生. 1982年 2月 고려대학교 전기공학과 졸업. 1984年 2月 고려대학교 전기공학과 석사 졸업. 1983年 12月 ~ 1986年 10月 동양정밀공업(OPC) 중앙연구소 연구원. 1986年 10月 ~ 1989年 10月 삼성종합기술원 정보 시스템 연구소 선임연구원. 1990年 3月 고려대학교 전기공학과 박사 입학. 1994年 8月 고려대학교 전기공학과 박사. 주관심 분야는 Computer Vision, Vision System HW 등임.



金 民 基(正會員)

1954年 2月 12日生. 1977年 2月 고려대학교 전기공학과 졸업. 1985年 1月 Columbia Univ in NewYork 석사 졸업. 1991年 6月 Polytechnique Univ 박사 졸업. 1992年 2月 ~ 현재 고려대학교 의용전자공학과 조교수. 주관심 분야는 Stereo Image, Multirate Signal Processing Medical imaging 등임.

朴 貴 泰(正會員) 第 31 卷 B編 第 6 號 參 照

현재 고려대학교 전기공학과 교수