

HMM Segmentation과 LVQ를 이용한 한국어 음절인식에 관한 연구

°안종영* 허강인*

* 동아대학교 전자공학과

A study on the recognition of Koreans syllable using HMM segmentation and LVQ

° Jong-Young Ahn* Kang-In Hur*

*Dept. of Electronic Eng., Dong-A Univ.

요 약

본 논문에서는 HMM 세그멘테이션(segmentation)을 이용하여 LVQ알고리즘에 적용시킨 하이브리드(hybrid) 음성인식에 관한 연구이다. LVQ 학습 알고리즘은 정적 패턴 분류를 위한 참조벡터 축, 고정차원인 벡터들을 생성하는데 유리하다. 하이브리드 알고리즘은 정적 패턴 인식에 사용되어지는 LVQ알고리즘에 HMM 세그멘테이션을 접목시켜 입력패턴을 정규화된 의미있는 값으로서 바꾸어 사용하는데 있다. 한국어 음절 중 8개 모음 /아/, /이/, /우/, /에/, /오/, /애/, /어/, /으/를 추출하여 인식실험을 하였다. 인식률은 화자종속일 경우 코드북 수 256개를 기준으로 LVQ1, LVQ2, LVQ3, OLVQ1 알고리즘 순으로 91.7%, 91.8%, 91.3%, 91.1%의 인식률을 구했고 화자 독립의 경우는 83.4%, 83.9%, 86.8%, 85.3%의 인식률을 구했다.

I. 서 론

과학기술의 점진적 발달에 따른 정보화구조, 나아가서는 인간과 기계와의 의사 소통에 대한 필요성이 증대되고 있다. 기계와의 대화는 음성 인식으로서 가능해진다. 음성 신호는 화자고유의 특징과 음성 신호자체를 나타낸다. 주어진 음성 신호의 특징파라미터를 분석하여 특정 화자를 인식한다. 현재 연구 진행중인 음성 인식법은 DTW, VQ, Neural network(신경 회로망), FUZZY, HMM 등이 있으나 DTW의 경우 기준패턴을 구성하는데 어려움이 있고 또한 인식과정에 있어서도 긴 시간이 소요되는 단점이 있다. 코호넨이 제안한 LVQ는 다양한 패턴인식에 있어서 계산량이 적고 인식률도 높은 장점이 있으나 음성인식에 있어서는 입력 패턴이 시간에 따라 데이터가 증가하고 정규화 과정이 복잡한 단점이 있다.

본 연구에서는 HMM 세그멘테이션에 LVQ를 결합한 하이브리드(hybrid) 알고리즘을 구성 하여 학습·인

식 실험하였다. 하이브리드 알고리즘의 목적은 음성 신호로부터 얻은 정규화 벡터를 입력패턴으로 사용하여 데이터 량을 줄이는데 있다. 정규화 벡터는 정적 학습 HMM의 세그멘테이션과 각 상태에서 계산되어진 평균벡터로부터 얻어지므로 음성 정보량이 줄어들어 효율적이다.

II. Hybrid 알고리즘

1. 개요

HMM/LVQ 하이브리드 알고리즘은 다음과 같다.

- 1) HMM-Based 세그멘테이션을 이용하여 Dynamic Speech Sample을 Static Pattern Sequence로 바꾼다.
- 2) LVQ 학습분류 (LVQ-Trained Distance Classifier)를 이용하여 입력패턴을 순차적으로 인식 한다. HMM에서 각 상태의 평균벡터를 취함으로써 LVQ 학습벡터로 사용이 가능하다.

그림 1은 하이브리드 알고리즘을 나타낸다.

정규화 되어진 입력벡터에 대한 초기 참조벡터로부터 코드북을 얻어 입력벡터와 평균적인 거리가 최소인 참조벡터를 코드북에 찾아 인식되어진다.

2. 학습단계

각 입력 패턴은 LVQ 거리 분류(distance classifier)를 학습하기 위해서 학습 가능한 정규화벡터로 생성되어 만들어진다.



- 기본 블록도 -

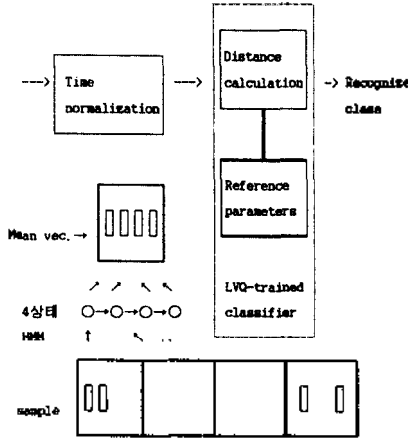
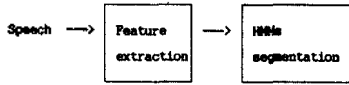


그림1. HMM/LVQ hybrid 알고리즘 블록선도.

HMM 세그멘테이션에서 학습에 가능한 정규화벡터를 가지며 이 벡터를 LVQ 알고리즘으로 참조벡터를 학습한다.

그림2에 학습과정을 보였다.

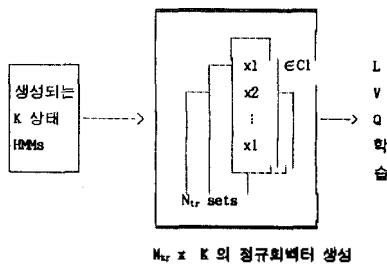


그림2. HMM/LVQ hybrid 알고리즘에 대한 학습단계

구성되는 K-HMMs이 각 상태 내에서 N_{tr} 음성샘플의 학습을 위해 활용되어지며 각 상태의 샘플들은 c_i 's 군들 중의 한 Label로 할당된다. 위 그림2에서 ($N_{tr} \times K$)개의 학습 정규화벡터를 가지며 이 벡터들을 LVQ 거리분류 알고리즘을 통하여 학습한다.

3. 인식단계

인식단계에서 모든 K-HMMs은 세그멘테이션을 위해

활용된다. 비교될 음성 입력샘플은 K-HMMs 으로 세그멘트하여 K-TN벡터 n_i ($i = 1, \dots, k$) 를 만든다. 그림3에 인식과정을 나타내었다.

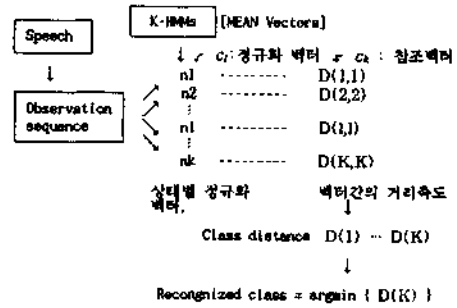


그림3. HMM/LVQ Hybrid 알고리즘 인식단계

여기서 n_i 은 c_i 의 HMM으로 만들어진 정규화벡터를 나타낸다. 정규화벡터를 취급하기위해 다음의 그 거리 척도를 정의한다.

첫째, segmentation-oriented(SO) distance $D(i,k)$.

둘째, c_k 의 Class distance $D(k)$.

셋째, $D(i,k) = q_k(n_i, \Delta)$.

$$D(k) = \frac{1}{k} \sum_{i=1}^k V_i D(i,k)$$

여기서 V_i 은 가중계수이다. $D(i,k)$ 는 n_i 이 c_k 에 어떻게 속해지는가를 나타내는 거리척도이다. 가중계수 V_i 은 유사 균일때는 $V_i=1$ 로 같은 균으로 카운터되고 그렇지 않을 경우는 $V_i=0$ 로 되어 카운터되지 않는다. 즉, 미지의 입력벡터에 대해서 코드북과 비교해서 같은 부류일때는 같다고 판단하여 갯수를 카운터하고 부류가 서로 틀릴 경우는 갯수에 포함시키지 않는다. 따라서, 후자의 경우는 인식률의 저하를 초래한다. $D(k)$ 는 벡터 거리의 모든 가중합이다.

III. Learning Vector Quantization(LVQ) 알고리즘

LVQ는 생성되어지는 참조벡터의 각 군에 대해서 그 거리 척도로서 학습되어지는 알고리즘의 하나이다. 이 LVQ방법은 VQ를 개선한 것으로 참조벡터를 뉴런들 간의 시냅스 결합에 대응시켜서 교사 학습으로 적절한 참조벡터를 구하는 방법이다.

1. LVQ1

입력벡터 x 를 여러영역으로 분류하기 위해 양자화된 값을 참조벡터 m_i 라고 한다. 여러 참조벡터 m_i 의 명들은 입력벡터 x 의 클래스 명에 따라 부여된다. 그때 입력벡터 x 는 가장 근접한 m_i 에 포함시키기 위해서 동일 클래스에 속하는가를 결정하게 된다. 그림4은 기본적인 LVQ 네트워크의 구조를 나타낸다.

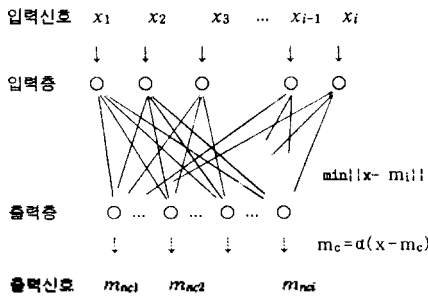


그림4. LVQ 네트워크 구조

$$c = \arg \min \{ \text{length} (x - m_i) \}$$

참조벡터 m_i 에 가장 근접한 입력벡터 x 를 m_c 로 나타낸다.

가장 근접한 분류를 하기 위해서 오차를 최소로 하는 m_i 의 값은 식(1),(2)의 학습과정으로 생성된다. $x(t)$ 는 입력벡터의 t 번째 요소이고, $m_i(t)$ 는 참조벡터 m_i 의 t 번째 요소를 나타낸다.

x 와 m_c 가 동일 Class 일때 :

$$m_c(t+1) = m_c(t) + \alpha(t)[x(t) - m_c(t)] \quad (1)$$

x 와 m_c 가 다른 Class 일때:

$$m_c(t+1) = m_c(t) - \alpha(t)[x(t) - m_c(t)] \quad (2)$$

i 와 c 가 같지 않을때

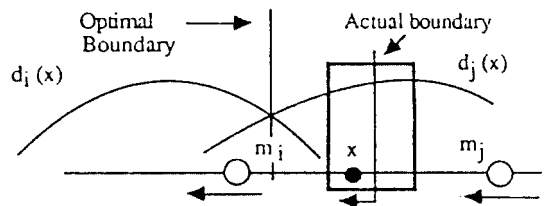
$$m_i(t+1) = m_i(t)$$

여기서, $0 < \alpha(t) < 1$, 그리고 $\alpha(t)$ 는 상수이며 시간에 따라 단조 감소한다.

2. LVQ2

LVQ1은 학습에 이용되어지는 입력벡터의 수가 많으면 패턴분류능력이 우수하지만 그 패턴분류능력을 더욱 향상시키기 위해 두 개의 참조벡터를 동시에 갱신하는 방법이 LVQ2이다.

LVQ2에서는 근접한 두 벡터군에 대한 중첩이 발생한다면 정확히 이 군들을 분류할 수 없으므로 분류 방법으로 두 벡터군중의 적당한 교차점에서 최적경계선(optimal line)을 결정하는 것이다.



x : j 분류에 속하는 학습벡터

m_i, m_j : i, j 군의 참조벡터

d_i, d_j : i, j 군의 분포

그림5. LVQ2 알고리즘

카테고리의 결정은 LVQ1과 동일하지만 학습에서 참조벡터 m_i 와 m_j 는 입력벡터 x 에 가장 근접한 벡터라면 두 벡터는 동시에 갱신된다. 두 카테고리중 한 카테고리는 정 분류이고, 다른 카테고리는 오 분류된 것이다. 그래서 x 는 m_i 와 m_j 의 경계치로 정의된 "창(window)"이라는 영역에 포함되는 벡터만 갱신된다. d_i, d_j 는 m_i, m_j 에서 유클리드 거리이다.

식(4)는 참조벡터와 입력벡터를 분류하는 것이고, 식(5)는 참조벡터와 입력벡터를 더욱 근접화 시키는 것이다.

$$\min (d_i / d_j ; d_j / d_i) > s ,$$

$$\text{where } s = \frac{(1-W)}{(1+W)} \quad (3)$$

식(3)에서는 근접벡터의 "창" 의 크기를 정의하고 있다. w 는 임의로 주어지는 상수값이며 s 는 창 의 크기를 결정하는 임계치가 된다.

알고리즘:

$$m_i(t+1) = m_i(t) - \alpha(t)[x(t) - m_i(t)] \quad (4)$$

$$m_j(t+1) = m_j(t) + \alpha(t)[x(t) - m_j(t)] \quad (5)$$

3. LVQ3

LVQ2 알고리즘은 두 개의 근접 카테고리중에서 한 카테고리는 정 분류이고 다른 카테고리는 오 분류일 때만을 고려하였다. 그러나 학습의 횟수가 많을 경우 각 분류중 계속 점근적으로 갱신 될 경우 문제가 발생하게 된다. 그래서 이렇게 학습이 지속될 경우는 LVQ2에서는 각각 분류가 안정성을 잃을 우려가 있다. LVQ3에서는 각분류에 이런 경우도 고려하여 식(6)에서와 같이 α 값에 또다른 학습가중치 ε ($0 < \varepsilon < 1$)를 곱해서 보정한다. 이것이 LVQ3 알고리즘이다.

$$m_k(t+1) = m_k(t) + \varepsilon(t) \cdot \alpha(t)[x(t) - m_k(t)] \quad (6)$$

여기서 k 가 j 번째 혹은 i 번째의 입력벡터 x 라면 m_j 와 m_i 는 같은 분류에 속한다.

여기서 ε 의 최적값은 존재하는 '창'의 크기에 의해서 정해진다. 이 알고리즘은 자기안정화, 즉 m_i 의 최적화는 계속적인 학습(learning)에서 변화하지 않음을 알 수 있다.

4. The Optimized - learning - rate LVQ1 (OLVQ1)

LVQ1의 알고리즘에 각 참조벡터 m_i 에 적용되는 $\alpha(t)$ 값을 수정함으로써 LVQ1의 식(1),(2)에 의해서 빠른 접근을 가능하게 한다. 최적화된 α 값을 가짐으로써 정 분류일때와 오 분류일때의 수렴 속도를 더욱 빠르게 한다.

정 분류일때 α 의 수정치;

$$\alpha^{new} = \frac{\alpha^{old}}{1 + \alpha^{old}} \quad (7)$$

오 분류일때 α 의 수정치;

$$\alpha^{new} = \frac{\alpha^{old}}{1 - \alpha^{old}} \quad (8)$$

위의 두 식으로 주어지는 α 값을 수정함으로써 LVQ1 보다 빠르게 수렴한다.

IV. 실험결과 및 고찰

1. 음성데이터 및 분석방법

음성데이터는 남자화자 7명이 5회 발성한 한국어 음절 59자 중에서 모음부분을 추출하여 HMM 세그먼트이션하였다. 음성의 분석조건은 표1과 같으며 특징 파라미터는 10차 mel cepstrum 계수를 이용하였다.

표1. 음성의 분석조건

샘플링 주파수	10 kHz
창길이	20 ms
프레임 주기	5 ms
분석	14 차 LPC 분석
특징 파라미터	10 차 LPC mel Cepstrum 계수

2. 결과 및 고찰

본 실험에서는 화자 7명의 5회 발성중 3회분을 학습 데이터(화자중속)로 하였고 나머지 2회분을 평가데이터(화자독립)로 하였다. LVQ에서는 입력패턴을 별 캡스 트럼계수로 인식 실험하였고, HMM/LVQ는 4상태 HMM으로 세그먼트하여 1회분 발성 데이터가 40개의 정규화된 평균벡터를 구하고 이를 LVQ방법들로 인식 실험하였다.

2.1 실험1

총 1,245개(7명×3회×52음절)의 학습데이터로 LVQ와 HMM/LVQ의 학습을 수행하였다. 표2는 인식 실험 결과이다.

표2. 화자중속 인식률[%]

방법	codebook	LVQ1	LVQ2	LVQ3	OLVQ1
HMM/LVQ	6 4	75.1	81.5	80.6	79.3
	1 2 8	84.6	88.0	87.5	87.4
	2 5 6	91.7	91.8	91.3	91.1
LVQ	6 4	74.5	78.8	80.3	77.5
	1 2 8	83.4	83.3	82.6	82.6
	2 5 6	87.4	89.6	89.5	88.3

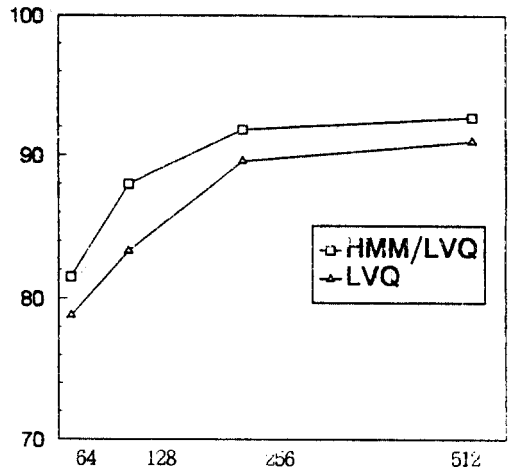


그림6. 중속화자 인식률 그래프

2.2 실험2

화자독립의 경우 7명 발성분중 학습에는 참여하지 않은 820개 데이터로 실시하였다. 화자독립 인식에서는 비교적 좋은 인식률을 얻었다.

표3. 화자독립 인식률[%]

방 법	codebook	LVQ1	LVQ2	LVQ3	OLVQ1
HMM/LVQ	6 4	70.2	80.9	81.1	76.2
	1 2 8	80.2	82.7	87.8	81.1
	2 5 6	83.4	83.9	86.8	85.3
L V Q	6 4	74.0	75.8	79.3	77.0
	1 2 8	80.7	79.8	79.6	81.2
	2 5 6	82.2	80.3	82.5	81.2

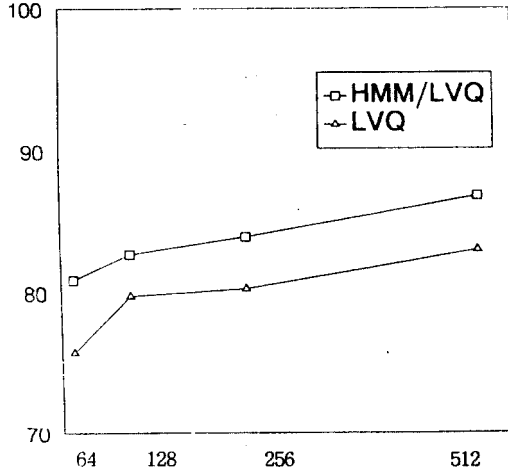


그림7. 독립화자 인식률 그래프

V. 결 론

본 논문에서는 LVQ와 HMM/LVQ를 이용한 하이브리드 알고리즘을 이용하여 인식실험을 행하였다.

실험 결과 HMM/LVQ가 기존 LVQ보다 평균적으로 향상된 인식률을 나타내었다. 화자중속의 경우 LVQ2에서 91.8%로 가장 높은 인식률을 얻었고, 화자 독립인 경우는 LVQ3이 86.8%로 가장 높은 인식률을 보였다. 하이브리드 알고리즘이 기존의 LVQ 보다는 5% 정도의 향상된 인식결과를 나타내었다. 또한 코드북 수에 따른 학습횟수의 적절한 변화를 준 결과 인식률의 향상을 얻을 수 있었다. 코드북 수의 증가에 따라 인식률도 높아지지만 데이터 갯수에 근접 할수록 양자화의 의미가 없어지므로 적당한 코드북 수도 중요한 의미를 가진다. 실험에 사용되었던 1,245개의 학습데이터의 경우 코드북수 256개가 실험결과 적절함을 알 수 있었다.

본 논문에서는 HMM 세그멘테이션만을 이용하여 하이브리드 알고리즘을 구성하였으나, 향후로는 HMM 학습과 인식과정에 적용 가능성을 검토하고 PNN등의 신경망 모델과의 접목을 연구할 필요성이 있다.

參 考 文 獻

- 1) Stanley C. Ahalt, Ashok K. Krishnamurthy, Prakoon Chen, and Douglas E. Melton "Competitive Learning Algorithms for Vector Quantization" Neural Networks, Vol. 3, pp. 277~290, 1990.
- 2) Pietro Burrascano "Learning Vector Quantization for the Probabilistic Neural Networks" IEEE TRANSACTIONS ON NEURAL NETWORKS. Vol. 2, NO. 4, pp. 456~461, 1991.
- 3) Erik Mc Dermott and Shigeru Katagiri "Shift-Invariant, Multi-Category Phoneme Recognition using Kohonen's LVQ2", IEEE, 1989.
- 4) 안태욱, 이상훈, 김순협 "VQ와 Multi-layer perceptron을 이용한 단모음 인식에 관한 연구", 韓國音響學會誌, VOL. 12, NO. 1, 1993.
- 5) 임창현, 이창수, 은종관 "모음 인식과 백터 양자화를 이용한 화자 인식", 韓國音響學會誌, VOL. 8, NO. 4, 1989.
- 6) 中川聖一, "確率モデルによる音節認識", 電子情報通信學會誌, 1988.
- 7) 中川聖一, "連続出力分布型HMMによる日本語認識", 音響學會論文誌 Vol. 46, PP.486~496, 1990.
- 8) 은종관 외, "연속분포 HMM을 이용한 한국어 연속 음성인식 시스템 개발", 한국음향학회 VOL. 13 NO. 1, PP.24~31, 1994
- 9) 허강인 외 2인, "사후화합·확률밀도함수의 추정과 Probabilistic Neural Network을 이용한 모음인식에 의한 평가", 韓國音響學會誌, VOL. 12, NO. 16, pp.21~27, 1993.