1999 年度 大韓電子工學會 夏季綜合學術大會 論文集 第 22 卷 第 1 號 99/6

코드북과 VQ 최적화에 의한 음소/고립단어 인식률 분석

안홍진, 주상현, 진원, 김기두
국민대학교 전자공학부
전화 : (02) 910-4707 / 팩스 : (02) 910-4449

Analysis of Phoneme/Isolated Word Recognition Rate Using Codebook and VQ Optimization

Hong-Jin Ahn, Sang-Hyun Joo, Won Chin, and Ki-Doo Kim
Department of Electronics Engineering, Kookmin University
E-mail : hong@dsp.kookmin.ac.kr

요약

본 논문에서는 음소별 코드북 개수의 선택과 벡터 양자화에 따른 음소 인식률과 고립단어 인식률에 대하여 다룬다. 음성모델은 이산 확률 밀도를 갖는 DHMM(Discrete Hidden Markov Model)을 사용하였으며, 코드북 생성과 벡터 양자화 알고리즘으로는 K-means 알고리즘과 LBG (Linde, Buzo, Gray) 알고리즘을 사용하였다. 음소별 코드북 개수와 벡터 양자화를 최적화함으로써 음소 인식률을 향상시킬 수 있으며, 그 결과 안정된 고립단어 인식률을 얻을 수 있다.

I. 서론

음성 인식을 위해 음성 데이터의 특징 벡터를 추출하게되며, 이 특징 벡터는 코드북을 생성하는 과정을 거치게 된다. 코드북은 음성 데이터의 중심 벡터를 인덱스로 가지고 있으며, 높은 인식률을 기대하기 위해 많은 량의 훈련 데이터를 필요로 하게 된다. 또한 코드북 인덱스의 개수가 많을수록 인식 성능이 좋아지기는 하지만 계산량이 상대적으로 커진다. 본 논문에서는 음소에 대한 코드북 개수의 최적화를 통해 음소 인식률을 향상시켰다.

벡터 양자화의 결과는 코드북 인덱스로 표현되는 데이터열이며, 벡터 양자화의 분해능이 높을수록 그 결과는 정교하다. 벡터 양자화는 벡터간 거리계산을 통해 가장 가까운 코드북 인덱스가 부여된다. 이 때 지정된 코드북 인덱스와 멀리 떨어져 있음에도 불구하고 가장 가깝기 때문에 억지로 양자화되어 인식률을 떨어뜨리는 경우가 있다. 본 논문에서는 이러한 현상을 피하기 위해 VQ 최적화를 한다.

단어 인식 알고리즘으로는 DHMM[1]을 사용하였다. 실험은 코드북을 개선한 경우와 벡터 양자화 알고리즘을 조합하여 여러 가지 상황에 대해 결과를 비교하였다.

본 논문에서는 코드북과 벡터 양자화 알고리즘에 의한 음소 인식률과 고립단어 인식률의 변화를 알고자 한다. 또한 음소 모델에 대한 정교한 음성 데이터 베이스가 필요하며, 본 논문에서는 음소의 천이과정을 포함하였다. 즉, 음소를 어떻게 레이블링 하는가에 따라 인식률에 큰 영향을 주게 된다. 실험을 통해 천이과정을 포함시키지 않은 음소인식 결과와 포함시킨 음소인식 결과는 약 30%의 차이를 보였다.

2장에서는 벡터 양자화에 대한 설명을 하고, 3장에서는 음성 인식 알고리즘인 DHMM에 대해 설명한다. 4장은 인식 시스템에 대하여 기술하며, 5장에서 실험 결과에 대하여 분석하고, 마지막으로 6장에서 결론을 맺는다.

II. 코드북 생성과 벡터 양자화

입력 웨이브 데이터를 인식하기 위해 우선적으로 요구되는 것이 코드북이다. 코드북은 유사한 값을 갖는 군집에 대한 중심값을 가지고 있으며, 벡터 양자화 과정에서 양자화 기준으로 사용된다. 본 논문에서는 K-means 알고리즘[2]과 LBG 알고리즘[3]을 사용하였다.

입력으로 들어온 데이터는 코드북의 인덱스 중 하나로 표현이 되는데, 이를 벡터 양자화라고 하며, 이러한 기술은 일종의 음성 압축이라고 볼 수 있다. 양자화 알고리즘은 코드북 생성과정에서 사용한 K-means 알고리즘과 LBG 알고리즘을 사용하였다.

2.1 Optimized K-means 알고리즘

K-means 알고리즘은 가장 일반적이고, 최소 거리 (왜곡)에 의한 군집화를 시도하는 군집화 알고리즘 중의 하나이며, 본 논문에서는 다음과 같이 최적화하여 구현한다.

단계 1 : 초기화

군집의 개수 K를 정하고 군집의 중심 벡터 $z_j(l)$을 초기화한다. 본 논문에서는 K는 1~16의 경우를 실험했다. 이렇게해서 구해진 코드북 수는 $K \times 89$개이다.

본 논문에서는 전체 훈련 데이터의 중심값으로부터 일정거리 만큼 이동시킨 값을 초기값으로 사용하였다.

단계 2 : 군집화

각각의 표본벡터 $\boldsymbol{x}^{(n)}$를 K개의 군집 중 유클리디안 거리가 가장 가까운 군집으로 포함시킨다. 이 때, 거리가 가장 가깝지만 문턱값 ($\tau=5$)을 넘는 표본벡터에 대해서는 군집에 포함시키지 않았다. 이는 벡터공간으로 표현된 훈련 데이터 중 값이 알맞지 않은 경우를 제외시키는 효과 (특히 잡음 성분)가 있다.

단계 3 : 새로운 군집 중심 계산

단계 2에서 얻은 군집 멤버를 사용하여 각 군집의 중심을 재계산한다.

단계 4 : 수렴 체크

다음식을 만족하면 작업을 마치고 그렇지 않으면 단계 2로 되돌아 간다.

$$z_j(l+1)=z_j(l), \quad j=1, 2, \ldots, 128 \qquad (1)$$

2.2 LBG (Linde, Buzo & Gray)알고리즘

LBG알고리즘은 K-means 알고리즘의 개념을 확장시킨 것으로 Linde, Buzo, Gray에 의해 발표되었으며, 벡터 양자화에 널리 사용되고 있다. LBG 알고리즘은 훈련 데이터를 2, 4, ..., 2^m 개로 단계적으로 나누어, 각 부분마다 중심값을 구하게 되며, 본 논문에서는 다음과 같이 최적화하여 구현한다.

단계 1 : 초기화

$L=1$ (클러스터 수)로 설정한다. 훈련 프레임 전체의 중심을 찾는다. 문턱값 $\zeta=0.005$를 설정한다.

단계 2 : 클러스터 증식

$L=2L$로 설정한다. 즉, 클러스터의 수를 두 배로 한다. 본 논문에서는 음소별 1~16개의 경우로 나누어 실험하였다.

단계 3 : 군집화

훈련 데이터 $\{\boldsymbol{x}_k\}$를 최소 거리 분류기 (classifier)에 의해 클러스터 C_i에 분류한다.

단계 4 : 코드북 갱신

각 클러스터에 중심값을 계산하여 모든 클러스터의 코드워드를 갱신한다.

단계 5 : 결정1

상대 왜곡 D가 문턱값보다 작다면 단계 6으로, 그렇지 않다면 단계 3으로 돌아간다.

$$D = \frac{D_l - D_t}{D_t} \qquad (2)$$

여기서, D_l은 이전 단계에서의 전체 왜곡이며, D_t는 현재 단계에서의 왜곡이다.

단계 6 : 결정2

L이 원하는 코드북의 크기와 같다면 작업을 마치고, 그렇지 않다면 단계 2로 돌아간다.

LBG 알고리즘의 단계 3과 4는 K-means 알고리즘과 같음을 알 수 있다.

벡터 양자화를 할 때, 코드북 벡터와의 거리가 비교적 큰 것에도 불구하고, 인덱스가 부여된 데이터에 대해서는 인식에서 제외시켰다. 즉, 테스트 벡터와 코드북 벡터간의 거리의 문턱값을 정하여 이 값을 넘는 테스트 벡터를 생략하는 방법이다. 이 값은 실험적으로 얻어진다. 실험에서는 1.5로 설정하였다. 이렇게해서 나온 음소인식 결과는 코드북 개수가 89개일 경우, 문턱값을 주지 않았던 방법보다 약 0.7%의 향상을 보였다.

III. 단어 인식 알고리즘

본 논문에서는 고립단어 인식 알고리즘으로 DHMM을 사용하였다. 이는 논문의 초점이 코드북 생성과 벡터 양자화에 의한 인식률의 변화를 실험하는 것이므로 시스템 설계가 간단한 DHMM을 이용하여 고립단어를 인식하는 실험을 하였다. DHMM의 출력 확률분포는 이산 확률분포이며, 고립단어 인식의 성능은 우수하다. 각 음소마다 하나의 state를 할당하였으며, 최소 2에서 최대 12개의 state로 구성한다. HMM 모델은 사용하는 고립단어의 수가 20개이므로 20개로 하였으며, 벡터 양자화를 통해 얻은 음소열을 가지고 유사도가 가장 높은 모델에 해당하는 고립단어를 인식하게 된다. 4장의 표 1은 인식에 사용한 고립단어이며, 그림 1은 설명을 예로 든 이상적인 DHMM state diagram이다. 실제로는 음소의 지속시간에 의해 state의 점유비율이 달라진다.

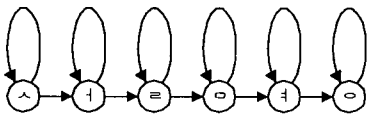

그림 1 '설명'의 state diagram

IV. 인식 시스템

본 논문에서 인식 시스템은 크게 두 가지로 나뉜다. 입력 음성이 벡터 양자화되어 음소로 인식되는 음소 인식 단계와, 벡터 양자화를 통해 얻어진 음소열이 DHMM에 의해 단어로 인식되는 단계이다. 사실 음소 인식 단계는 인식이라기 보다는 벡터 양자화 과정으로 생각해야 하는데, 인식 단계로 구분한 것은 테스트 패턴이 벡터 양자화를 거쳐 음소열로 압축되는 과정이 일종의 인식으로 생각할 수 있기 때문이다. 즉, 음소 인식 단계에서 인식률은 테스트 패턴이 원하는 음소로 양자화 되는가를 관찰하는 것이다.

PC의 마이크를 통해 들어온 음성을 16비트의 분해능을 가진 16kHz로 표본화하고, 다음과 같은 전달함수를 갖는 pre-emphasis를 거친다.

$$H(z) = 1-0.98z^{-1} \qquad (3)$$

이 음성은 10msec씩 프레임으로 분할된다. 이 프레임은 5msec씩 중첩 (50%)되며, 12차 LPC (Linear Predictive Coding) 켑스트럴 계수를 구한다. 이 계수는 아래의 윈도우 W_m로 weighting 되며[1], 정규화된 LPC 켑스트럼 계수를 사용한다.

$$W_m = 1+\frac{Q}{2}\sin\left(\frac{\pi m}{Q}\right),\ 1\le m\le Q \qquad (4)$$

음소를 인식하기 위해 크기 $K\times 89$ ($K=1,2,4,8$)개의 코드북을 사용하였으며, 사용된 음소는 모두 89개이다.

실험에 사용한 단어는 모두 20개이며, 각각 2~12개의 state를 갖는다. 아래의 표 1에 실험에 사용한 단어 모델을 나열하였다.

표 1. 단어 모델

01	열기	11	소트
02	닫기	12	히스토그램
03	다음	13	팔레트
04	설명	14	가우시안
05	확인	15	메디안
06	취소	16	세선화
07	컬러모델	17	침식
08	휴	18	팽창
09	세츄레이션	19	소벨
10	인텐시티	20	라플라시안

V. 실험 방법

5.1 음성 DB

음성 DB는 PC의 마이크를 통해 얻은 단어로부터 수작업에 의해 음소 단위로 구축하였으며, 음소간의 오른쪽 천이과정을 포함하였다. 음소는 모두 89개이다. 음성 데이터는 모두 10명의 남성 화

자를 사용하였으며, 20개의 단어를 5번씩 발성하여 8명분의 3번 발성한 데이터를 훈련용으로 나머지 2번 발성한 데이터와 2명분의 5번 발성한 데이터를 테스트용으로 사용하였다.

5.2 음소인식 실험방법

이 단계의 실험은 입력 음성 데이터를 벡터 양자화하는 실험이다. 코드북 개수와 벡터 양자화 단계에서 벡터 양자화 알고리즘의 종류로 실험하였다. 실험은 다음의 방식으로 하였다.

1) 음소별 코드북 개수
   1, 2, 4, 8개
2) VQ 알고리즘
   K-means 알고리즘
   LBG 알고리즘

5.3 고립단어 인식 실험방법

고립단어의 인식은 코드북 생성시 사용했던 훈련 데이터로 벡터 양자화를 통해 얻은 음소열을 가지고 DHMM 모델을 훈련시켰다. 이 단계에서의 실험은 코드북의 성능에 의해 고립단어의 인식률이 어떻게 영향을 받는지 알아보는 실험이다.

VI. 실험 결과

음소인식 실험결과는 표 2에 나타나 있다.

표 2. 음소인식 실험결과

K-means 알고리즘				
코드북 개수	89	178	356	712
인식률 (%)	48.5	51.5	56.4	59.2
LBG 알고리즘				
인식률 (%)	48.5	51.8	56.2	58.9

표 2에서 보는 바와 같이 코드북 개수의 최적화에 따른 음소인식 성능의 향상을 기대할 수 있다. 이는 고립단어 인식 단계에서 음소열의 출력 확률을 높이는 효과가 있으며, 인식률이 향상되었음을 실험결과를 통해 알 수 있다.

고립단어 인식실험 결과를 보면 코드북 크기가 178을 사용하여, K-means 알고리즘으로 벡터 양자화했을 경우가 가장 우수한 결과를 나타내었다.

표 3. 고립단어 인식 실험결과

K-means 알고리즘				
코드북 개수	89	178	356	712
인식률 (%)	92.5	95.8	92.5	91.9
LBG 알고리즘				
인식률 (%)	92.5	91.7	94.2	93.1

VII. 결론

본 논문에서는 코드북과 벡터 양자화 알고리즘에 따른 음소 인식률과 고립단어 인식률 변화에 대해서 다루어 보았다. 코드북을 최적화하고 K-means 알고리즘과 LBG 알고리즘을 사용하여 음소 인식성능을 비교하였으며, 그에 따른 고립단어 인식에서의 영향을 실험을 통해 검증하였다. 인식 성능 실험결과 음소별 2개의 코드워드를 갖는 코드북 (코드북 크기가 178 일때)을 사용하여 K-means 알고리즘으로 벡터 양자화를 했을 경우, 고립단어 인식에서 가장 높은 인식률을 얻었다.

본 논문에서는 고립단어 인식의 성능을 높이기 위해 음소 단위의 인식률을 높인 것이 DHMM에서 음소열의 출력 확률을 높이는 결과를 보임을 확인하였다.

참고문헌

[1] L. Rabiner, B. H. Juang, "Fundamentals of Speech Recognition", Prentice-Hall, 1993.

[2] A. S. Pandya, R. B. Macy, "Pattern Recognition with Neural Networks in C++", IEEE Press, 1996.

[3] X. D. Huang, Y. Ariki, M. A. jack, "Hidden Markov Models For Speech Recognition", Edinburgh Univ. Press, 1990.

[4] Y. Zhang, R. Togneri, C. deSilva, M. Alder, "Optimization of Phoneme-Based VQ Codebook in a DHMM System", The Univ. of Western Australia.