

코우드북의 절충탐색에 의한 벡터양자화

Vector Quantization by N-ary Search of a Codebook

이 창 영*

Chang-Young Lee

ABSTRACT

We propose a new scheme for VQ codebook search. The procedure is in between the binary-tree-search and full-search and thus might be called N -ary search of a codebook. Through the experiment performed on 7200 frames spoken by 25 speakers, we confirmed that the best codewords as good as by the full-search were obtained at moderate time consumption comparable to the binary-tree-search. In application to speech recognition by HMM/VQ with Bakis model, where appearance of a specific codeword is essential in the parameter training phase, the method proposed here is expected to provide an efficient training procedure.

Keywords: Speech Recognition, Vector Quantization, Codebook, HMM/VQ

1. 서 론

주어진 아날로그 신호의 각 샘플링 값 하나하나에 대해 수행하는 scalar quantization에 비해, 보다 큰 단위 즉 frame에 대해 적용하는 벡터양자화(Vector Quantization, VQ)가, 전송률 · 신뢰도 등에 있어서 큰 이점을 갖고 있다는 것은 rate-distortion 이론에 의해 쉽게 밝혀지는 기본적인 결과이다. 특히 샘플 사이의 의존도 즉 correlation을 고려하는 경우에는, 스칼라 양자화에 비해 대단히 효율적이어서, 벡터양자화는 디지털 신호처리에서 매우 폭넓은 활용을 보고 있는 부호화 기법의 하나로 자리잡고 있다.

벡터양자화를 수행하기 위해, 우선 N 개의 표준패턴을 저장하는 codebook의 생성이 필요하다. N 에 비해 충분히 큰 개수의 다양한 패턴에 대해 clustering을 적용함으로써, 그들을 N 개의 group 즉 cluster로 분류하고, 각 cluster의 centroid vector가 codebook에 기록된다. 입력패턴을 그 N 개의 centroid vector와 비교하여, 가장 유사한 것을 선택하고, 그에 해당하는 codebook index를 전송하는 것이 VQ의 개요라 할 수 있다.

Codebook에서 입력패턴에 가장 가까운 index 즉 codeword를 찾는 방법으로는 완전탐색법(full search)과 2진트리-탐색법(binary-tree-search)이 사용되고 있다. 전자는 입력신호에

* 동서대학교 정보시스템공학부

가장 가까운, 즉 입력신호와의 distortion이 가장 작은 codeword를 찾을 수 있다는 장점이 있는 반면 시간이 많이 걸린다는 단점이 있다. 특히 codebook의 크기가 커짐에 따라, 계산량은 지수적으로 증가하여[1] 실시간 처리가 요구되는 곳에서는 사용이 곤란하다. 예를 들어 음성 인식에의 적용으로서, 40 개의 음소와 그 각각에 대해 25 가지의 변화를 고려한 $N=1024$ 의 codebook을 사용하는 경우에는 완전탐색법이 거의 사용되지 않는다.

이리하여 등장한 것이 2진트리-탐색법이지만[2], 이는 입력신호에 가장 가까운 codeword를 찾지 못하는 경우가 많다는 결함을 갖고 있다. 이는 2진트리 구조에 내재된 구조적인 문제를 극복하지 못한 것이 그 원인으로, 이를 보완하고자 codebook의 설계를 nonuniform tree로 하는 방법도 시도되었지만, 보통 “가장 가깝지는 않더라도”를 감수하며 2진트리-탐색법이 보편적으로 사용된다. 이 이 논문에서는 앞의 두 방식을 보완한 방법으로서 아주 탁월하게 작동하는 codebook 탐색법을 소개한다.

2. Codebook의 절충탐색법(N-ary Search)

2진트리 탐색에서는 트리의 어느 level에서건 둘 중의 하나를 선택한다. 따라서 입력패턴과 가장 가까운 벡터 V_B 로부터 멀어질 가능성이 level 증가에 따라 점점 높아진다. 2진트리는 root에서 시작하여 이를 둘로 나누고, 그에 의해 생성된 둘을 다시 넷으로 나누는 일련의 bifurcation으로 이어지기 때문에, 일단 root에서 V_B 로 이르는 path P_B 에서 이탈하면 다시는 되돌아올 수 없게 된다. 이런 의미에서, 이 방법은 구조적으로 “suboptimal search”를 벗어날 수가 없는 것이다.

본 연구에서 제안하는 것은 “둘 중의 하나”라는 개념을 “ $2N$ 개 중에서 N 개”로 확장하는 것이다. 즉, 어느 level에서건 $2N$ 개의 후보자 가운데 N 개를 선정하고, 최후의 level에서 선택되는 N 개의 후보자(leaves)에 대해서는 완전탐색법을 적용하여, 하나의 codeword를 선별하는 것이다. 즉, leaf에 도달하는 과정으로서 종전에 단일 edge만을 고려하던 것을 이번에는 여러 edge를 동시에 고려하는 것인데, 이것이 P_B 로부터의 이탈을 억제하는 효과는 매우 뛰어난 것으로 본 연구 결과 밝혀졌다.

본 논문에서의 탐색 방법은 “ N -ary Search” 또는 “Hetero-Search”라고 부를 수 있을 것이다. $N=1$ 일 때는 2진트리-탐색과 동일하고, $N=2^{L-1}$ 인 경우에는 완전탐색으로 환원된다. 여기서 L 은 2진트리의 height 즉 최종 level을 의미한다. 그림 1은 이를 실현하는 algorithm을 Block Diagram으로 구성한 것이다. 본 연구에서 제안하는 탐색법과 유사한 방법들이 DTW(Dynamic Time Warping)에 의한 음성인식에서 사용된 바가 있는데, “tree-trellis search”를 그 한 예로 들 수 있다[3].

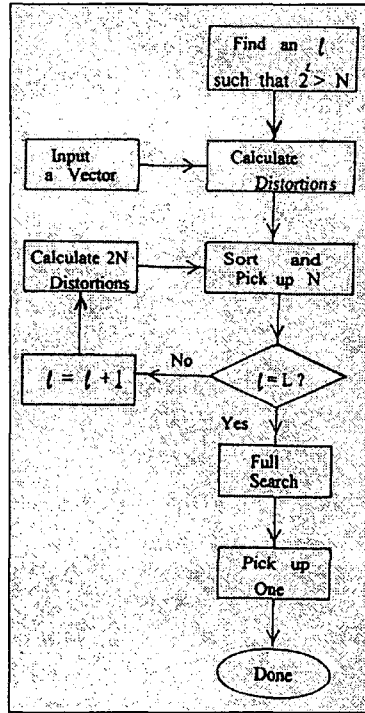


그림 1. N-ary Search를 위한 Block Diagram.

3. 실험 방법

마이크에 입력된 음성신호를 8 kHz의 차단주파수를 갖는 low-pass-filter를 거쳐, 12 kHz의 sampling 주파수로 digitization시켰다. DC bias의 제거 · endpoint detection · spectral flattening을 수행한 다음, 21.25 msec에 해당하는 256 개의 data point를 한 frame으로 하고, 100 개의 data point를 이동시켜 다음 frame으로 잡았다. 특징 추출은 15 차 LPC로부터 유도된 22 차 cepstral coefficient를 사용하였다.

실험에 참가한 사람은 남자 13 명 · 여자 12 명의 25 명으로, 이들 각자가 열 개의 숫자를 발성하여 얻어진 250 개의 token · 7,200 개의 frame이 codebook 생성을 위한 clustering에 사용되었다. LBG clustering algorithm[4]에 의해 cluster 수가 64, 즉 $L=6$ 인 codebook을 만들었다. 각 level에서는 더 이상의 centroid 변화가 없을 때까지 iteration을 한 후 bifurcation이 이루어지도록 하였다.

이렇게 생성된 codebook에 의해 앞서 추출된 특징들을 벡터양자화하였다. 각 token마다 8가지의 N 에 대해 VQ를 행하고, 그로 얻어진 codebook index를 완전탐색법에 의해 얻어진 결과와 비교하여 match되는 비율을 조사하였다. 그림 1에 등장하는 sorting에는 잘 알려진 “Quick-Sort” algorithm을 적용하였다.

4. 결 과

우선 N -ary search의 성능을 알아보기 위하여 N 값의 변화에 따른 VQ 결과를 알아보았다. 표 1은 남성음에 의해 발음된 한국어 숫자 "1"의 음성신호를 $L=6$ 의 codebook에 의해 양자화시킨 예를 보인 것이다. 표에서 F 는 frame index를 나타낸다. 비교를 쉽게 하기 위하여 완전탐색에 해당하는 $N=2^{L-1}=32$ 인 경우를 맨 처음 적었고, 따라서 그 결과는 해당 frame의 V_B 를 나타낸다. 이와 틀린 결과는 밑줄을 그어 표시하였으며, 맨 마지막 열의 %는 matching 비율을 나타낸다.

표 1이 보여주는 특징 중 가장 눈여겨볼 것은 $N=2$ 의 경우만 하더라도 2진트리-탐색법에 비해 대단히 높은 matching ratio를 낸다는 사실이다. 또 하나의 특징은 $N=4$ 일 때 벌써 완전탐색과 동일한 결과가 나타났다는 점이다. 표 1은 하나의 전형적인 예로서, 대부분의 경우에 $N=4\sim 5$ 에 의해 완전탐색과 동일한 결과에 도달하였다.

표 1. 여러 탐색법에 의한 벡터양자화의 결과 예

$N \backslash F$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	$F\%$
32 (완전)	32	32	32	32	32	43	34	41	41	41	42	43	22	22	22	42	22	22	44	22	
1 (2진트리)	32	<u>45</u>	32	<u>45</u>	<u>45</u>	<u>45</u>	<u>45</u>	<u>45</u>	<u>20</u>	<u>20</u>	<u>19</u>	<u>29</u>	<u>29</u>	<u>29</u>	<u>30</u>	<u>25</u>	<u>30</u>	<u>29</u>	<u>31</u>	<u>31</u>	10
2	32	<u>43</u>	32	32	32	43	34	<u>43</u>	41	41	<u>16</u>	43	22	22	<u>43</u>	42	22	22	44	<u>25</u>	75
3	32	32	32	32	32	43	34	<u>43</u>	41	41	42	43	22	22	22	42	22	22	44	<u>44</u>	90
4	32	32	32	32	32	43	34	41	41	41	42	43	22	22	22	42	22	22	44	22	100

보다 객관적인 평가를 위하여 clustering에 사용된 7200 frame 모두에 대해 표 1과 마찬가지로의 평가를 수행하여, 누적된 matching 비율을 계산하였다. 본 연구에서는 codebook의 크기 L 에 따른 조사를 병행하고 있으므로, $L=5$ 인 경우와 $L=6$ 인 경우를 같이 다루었다. 그림 2는 N 의 값에 따른 완전탐색과의 matching 비율을 그래프로 나타낸 것이다.

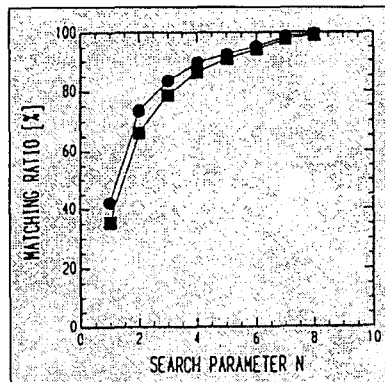


그림 2. 2진트리-탐색 및 N -ary search가 완전탐색과 같은 결과를 내는 비율. 원은 $L=5$ 일 때를, 사각형은 $L=6$ 일 때를 나타낸다.

완전탐색이 실제 사용되지 못하는 이유는 계산의 load와 그에 따른 시간소요에 있다. 즉 계산상의 짐이 가볍지 않으면 실용화가 곤란한 것이다. 이 점에서는 2진트리-탐색이 단연 뛰어나다. 그러면 본 연구에서 제안하는 N -ary search는 어느 정도의 시간이 소요될까? 서론에서 언급되었듯이, codebook height L 의 값에 따라 필요한 계산량(floationg point operation, flop)은, 완전탐색의 경우에는 지수적으로, 그리고 2진트리-탐색의 경우에는 선형적으로 증가한다. 본 연구에서 제안하는 N -ary search의 경우에는 “sorting”의 문제가 있어서 해석적으로 이를 조사하기는 다소 어려움이 따른다.

경험적인 결과를 얻기 위하여, 앞의 7200 frame의 VQ에 필요한 시간을 측정하였다. 그림 3은 N 의 값에 따른 탐색소요시간을 2진트리-탐색에 소요되는 시간과의 비율로 나타낸 것이다. 이 결과가 보여주는 것은 N 의 값의 증가에 따른 계산상의 부담이 다소 완만하다는 점이다. 따라서 $N=2\sim3$ 의 N -ary search는 기존에 2진트리-탐색으로 실시간 처리가 가능한 응용에 별 차이 없이 사용될 수 있다고 사료된다.

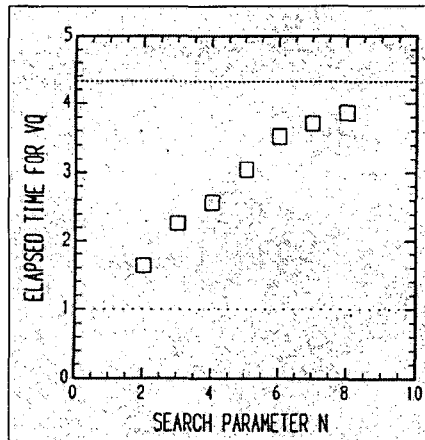


그림 3. $L=6$ 일 때, N 의 값 변화에 따른 계산 소요시간.

세로축 값은 2진트리-탐색법에 소요되는 시간과의 비율을 나타낸다.

아래 점선은 2진트리-탐색에, 위 점선은 완전탐색에 대한 소요시간이다.

5. 결론

본 연구에서는 새로운 codebook 탐색법을 제안하였다. 2진트리-탐색에 버금가는 소요시간에 의해, 완전탐색에 근접하는 최선의 벡터양자화가 수행됨을 실험적으로 확인하였다. 이는 Bakis model을 사용하는 HMM/VQ에 의한 음성인식에서, 파라미터 훈련문제 및 인식률을 매우 효과적으로 개선시키는 방법을 제공할 것으로 기대된다.

참 고 문 헌

- [1] Dell, J. R. Jr. et al. 1993. *Discrete-Time Processing of Speech Signals*. Macmillan, New York, 425-434.
- [2] Cheng, D. Y. et al. 1984. "Fast Search Algorithms for Vector Quantization and Pattern Matching." *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Diego, California, paper 9.11.
- [3] Rabiner, L. & B. H. Juang. 1993. *Fundamentals of Speech Recognition*. Prentice-Hall, New Jersey, 232-238.
- [4] Linde, Y., A. Buzo. & R. M. Gray. 1980. "An algorithm for vector quantizer design." *IEEE Trans. on Communications*, Vol. 28, 84-95.

접수일자: 2001. 7. 10.

게재결정: 2001. 8. 26.

▲ 이창영

부산광역시 사상구 주례동 산 69-1 (우: 617-716)

동서대학교 정보시스템공학부

E-mail: seewhy@kowon.dongseo.ac.kr