

VQ 코드북의 빠른 검색을 위한 알고리즘

이강성, 김순협

광운대학교 전기·계산기 공학과

An Algorithm for Fast Searching of VQ Codebook

LEE, GANG SUNG

KIM, SOON^{HYOB}

Dept. of Computer Eng., Kwangroon Univ

요약

벡터 양자화(VQ)는 신호 처리분야에서 정보의 압축을 위해 사용하는 아주 잘 알려진 방법이다. 벡터 양자화는 정보를 대량으로 줄이면서 그 효율을 떨어 뜨리지 않는 방향으로 발전해 왔다. VQ코드북의 크기가 커지면 하나의 코드워드를 찾기위한 시간이 증가하게 된다. 코드북의 빠른 검색을 위하여 다른 방법이 제안 되기도 했으나 최적 검색 방법이라고는 볼 수 없다. 본 고에서는 음성인식에 적용할 목적으로 기존의 방법으로 구성된 코드북의 구성을 변형 하지 않고 검색 속도를 증가시킬 수 있는 방법을 기술하고 그 효율에 대해서 설명한다.

1. 서론

양자화는 연속 신호를 디지털 신호로 변환하는 과정으로 데이터 압축 또는 코딩의 중요한 요소로, 데이터 전송시간을 줄이거나 신뢰성을 높이기 위한 방법으로 사용되고 있다. VQ는 디지털 전송에 있어서 유연성, 실외성 그리고 저렴한 비용을 제공하며 음성인식, 화상인식등의 패턴 인식분야에서도 계산량이나 기억 용량을 줄이기 위하여 자주 사용되는 기법이다[4].

VQ는 인코더와 디코더 두부분으로 나누어 볼 수가 있다. 인코더는 입력된 벡터를 코드북내에 저장된 코드워드와 비교하여 가장 적은 거리를 내는 코드워드의 인덱스를 생성하는 것이며, 디코더는 전송되어온 코드워드 인덱스를 이용하여 코드북에서 코드워드를 찾아 그 벡터를 출력하는 것이다 [1].

디코더에서 인덱스가 입력 되었을 때 해당되는 코드워드를 출력하는 것은 단순하게 구성이 가능하다. 중요한 것은 첫째, 코드북을 어떻게 잘 설계할 것인가 하는 것과 둘째, 입력 벡터 x 가 들어 왔을 때 코드북에서 최적 코드워드를 어떻게 찾을 것인가 하는것이다. 검색 방법으로는 크게 2가지로 분류해 볼 수 있다. 첫번째로는 전체를 비교하는

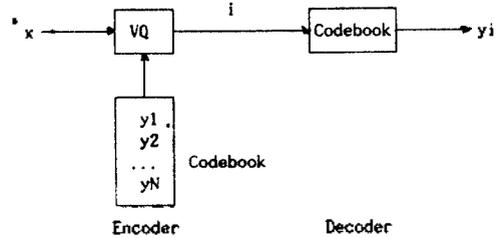


그림 1. Vector Quantizer

최적화 VQ 검색이다. 이것은 가장 이상적인 방법이나 전체를 비교하는 시간이 많이 소모된다. 예를 들어 이산 변환을 이용하는 음성 인식에서 전체 코드북의 크기가 512라고 한다면 입력이 하나 들어 올 때 마다 512개의 코드워드와 비교 되어야 한다. 물론 대상 인식 단어가 많을 경우 \sqrt{N} 모델과 비교되는 시간이 가장 많이 걸리겠으나 VQ 인코딩 하는 시간이 줄어든다면 성능은 향상될 수 있을 것이다. 실시간 음성인식을 하기 위해서 DSP전용 프로세서를 사용하는 경우, 음성 샘플을 A/D변환을 하는 동시에 특징 벡터를 계산 하게 되는데, 만일 VQ를 사용하는 인식기법을 적용하고 있다면, VQ 인코딩 까지 탑재할 수 있다면 인식기 성능은 크게 향상될 것이다.

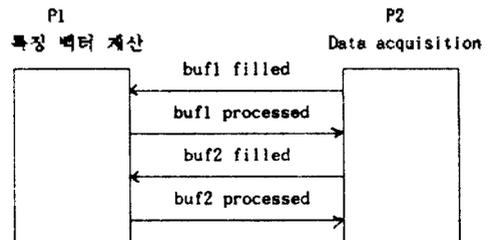


그림 2. 샘플값 입력과 특징 벡터를 구하는 두개의 프로세스

코드북내의 모든 코드워드와 비교 한다는 것은 우리가 아닐 수 없다. 따라서 트리 검색등의 준 최적화 검색 기법을 사용하기도 한다[4-5]. 그러나 준 최적화 VQ는 최적화 VQ 보다 평균 거리가 증가하게 되므로 정밀한 정보를 요구하는 곳에는 최적화 VQ방법이 주로 사용된다. 본 고에서는 최적화 VQ로 설계된 코드북을 좀 더 빠른 방법으로 검색하는 알고리즘을 소개한다. 기존의 코드북의 구조를 변경하지 않고 슈퍼 코드북 만을 하나더 생성하는 것으로 30%이상의 계산 시간을 줄일 수 있다.

11. 알고리즘

본 알고리즘은 전체 검색 대상 범위를 축소하는 것으로 2개의 코드북이 요구된다. 하나는 기존에 사용되던 VQ 코드북이며 또 하나는 기존의 코드북을 한번 더 집단화 하여 생성한 슈퍼 코드북이다. 기존에 사용되는 코드북을 생성하기 위해서 제한되는 내용은 없다. 여기서는 전체 코드북 검색을 수행해야 입력 벡터에 가장 가까운 해당 코드워드를 찾을 수 있는 집단화 기법을 사용한다고 가정한다. 다수의 입력 벡터들이 들어 오면 그 벡터들은 기존의 집단화 방법을 이용하여 중심값을 결정하고 그 중심값들을 모아 코드북을 생성한다. 그런 다음에 다시 한번 집단화를 이용하여 코드북을 생성하는데 이때 사용되는 입력 벡터는 일단에서 생성된 코드북이 되며 생성되는 새로운 코드북을 슈퍼 코드북(SVQ)이라고 한다(그림 3).

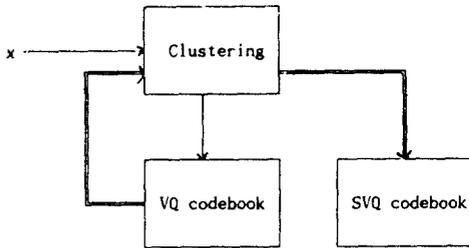


그림 3. 코드북 생성

SVQ 코드북은 새로 생성된 중심값과 각각의 중심값에 속한 VQ 코드북의 점자 값을 갖고 있다.

y1	s1	2, 3, 9
y2	s2	0, 1, 7, 13
y3	s3	4, 9, 3
yN	sM	-----

그림 4. 코드북의 구조

인코딩은 다음 단계별로 이루어 지게 된다. 입력 벡터 x 가 들어오게 되면 SVQ 코드북의 코드워드와 선행 비교를 하게 된다. 여기서 입력벡터 x 와 가장 가까운 k 개의 코드워드를 선택한다.

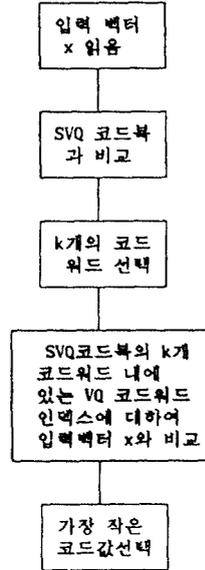


그림 5. 인코딩 절차

SVQ코드북에는 각각의 코드워드 내에 포함되는 VQ 코드워드들의 인덱스 정보가 관리되고 있는데(그림 4) 선택된 k 개의 코드워드에 대해서 이들 VQ코드워드의 인덱스의 벡터와 입력 벡터 x 와의 거리를 계산한다. 그래서 가장 작은 거리값을 낸 코드워드를 최종 코드워드로 해석하고 출력한다.

입력 벡터가 들어 왔을 때 VQ 코드북의 코드워드들이 확보하는 공간은 일반적으로 그림 6 과 같은 Voronoi 분할이 된다. 이 코드워드들을 입력 벡터로 한 SVQ 코드북이 이 공간위에 생성하는 SVQ 코드워드의 공간은 또한 다양한 모양을 구성한다. 따라서 어떤 입력벡터 x 가 들어 왔을 때 어떤 SVQ 코드워드의 공간에 속한다고 해서 그 코드워드가 포함하는 VQ 코드워드들 중 하나와 가장 가깝다고 말할 수는 없다.

따라서 SVQ의 코드워드들 중 k 개의 가장 가까운 벡터를 대상으로 그들에 속해 있는 VQ코드워드를 조사한다면, 적당한 k 가 주어진 경우 거의 완벽한 VQ를 할 수 있을 것이다. 문제는 어느정도의 k 값이 타당하며 그 때의 계산량 감속 비효율은 어느 정도나 되는지를 찾는 것일 것이다.

III. 실험 및 고찰

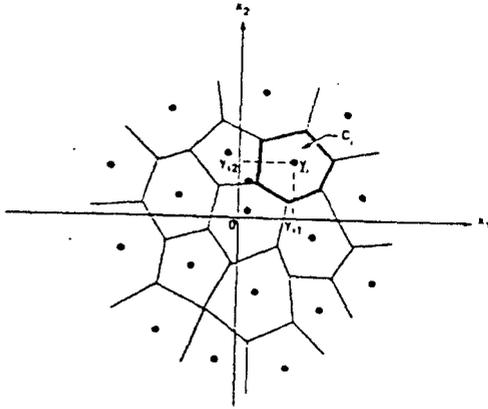


그림 6. 공간 분할

실험은 다음 세가지 기준에서 행해졌다. 1) VQ 코드북 크기에 따라 몇개의 코드워드를 SVQ에 할당해야 할 것인가 2) SVQ의 크기가 정해져 있을 때 k 값을 얼마로 결정할 것인가 3) 시간 감속은 어느정도나 이루어 지는가를 고려할 필요가 있다.

1. 음성 데이터

음성 데이터는 마이크를 통하여 남성 1인이 도명 및 전화 번호를 발음한 약 4000개의 벡터로 사용하였다. 음성은 3.4kHz 지역 통과 필터를 통과하여 8kHz로 샘플링 되었고 16비트 A/D변환 되었다. 16ms를 한프레임으로 $1-0.95z^{-1}$ 의 전처리와 해방창을 통과 시켜 10차의 LPC 분석을 하였다. 그후 10차의 LPC cepstrum으로 변환하여 특징 벡터로 사용하였다.

2. VQ 코드북의 작성

VQ 코드북은 크기 64, 128, 256의 3개를 구성하였다. 집단화 방법은 k-means 방법을 사용하였으며 추출된 음성 전체에 대해서 수행 되었다. 임계벡터들 간의 거리를 계산하기 위하여 다음식을 사용하였다.

$$d(x, y) = w(c_{x0} - c_{y0})^2 + \sum_i (c_{xi} - c_{yi})^2$$

여기서, x, y는 두개의 켈스트럼 벡터이고 w는 무게 상수로 0.04로 설정하였다.

3. 인코딩 실험

입력된 벡터를 VQ 코드북 전체와 비교하여 최적 코드워드를 찾는 기준의 선형 검색 방법과 본 논문에서 제안한 SVQ 코드북을 통한 2차 검색한 방법의 결과를 비교하는 실험을 통해 그성능을 확인한다. 실험 대상이 되는 VQ 코드북의 크기는 64, 128, 256각각에 대하여 다양한 크기의 SVQ를 작성하여 실험하였다. VQ 코드북 크기 64인 경우는 SVQ코드북

의 크기를 5 - 10개, VQ 코드북 크기 128인 경우는 SVQ의 크기를 6 - 13개로 1씩 증가하면서 실험하였으며, VQ 코드북 크기 256인 경우는 SVQ 코드북 크기를 6 - 26개로 2씩 증가하면서 실험하였다. SVQ 코드북의 크기는 대략 VQ 코드북수를 10으로 나눈값 까지로 설정하였다. 또한 입력 벡터와 SVQ 코드북과의 비교에서 생성되는 k개의 후보 코드워드의 수도 SVQ코드북의 크기에 따라서 1 부터 14까지 다양하게 적용하였다. 그림 7에 실험 결과를 보인다.

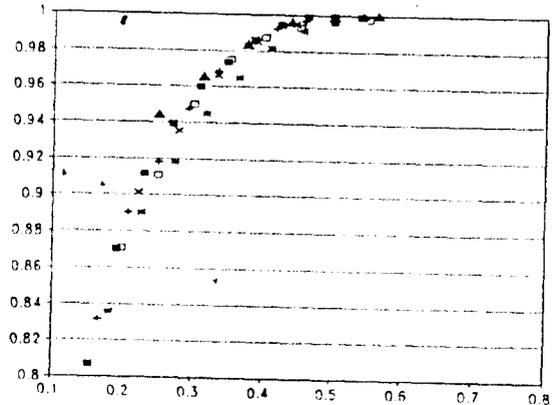


그림 7. SVQ를 통한 분류율

평균은 SVQ의 후보 갯수를 SVQ크기로 나눈 값이고, 세로 축은 VQ코드북을 선택 전제 탐색한 결과와 SVQ를 이용한 탐색 결과 코드의 일치도 (분류율)을 나타낸 그림이다. 1이변 두경우가 완전히 일치하는 것을 나타낸다. 그림 8을 통해 알수 있는 사실은 SVQ의 크기에 따라서 데이터의 분포가 결정 되는 것이 아니고 k개의 후보 코드워드와 SVQ 크기의 비가 중요한 요소임을 알 수 있다. k/SVQ크기 값이 0.5 부근에서 분류율이 거의 수렴하는 것을 알 수 있다. 0.5일 때의 분류율은 0.997 로서 충분한 비율임을 알 수 있다. 또한 분류가 정확히 이루어 나지 않는다고 해도 분류된 코드워드 값이 정확히 분류되면 값을 갖을 코드워드의 아주 인접한 근처에 있을 것임이 명백하다. 따라서 음성인식 등이 어느 정도의 스트리엄의 다양성에 적용을 하도록 구성되어 있으므로 영향을 거의 미치지 않을 것으로 보인다.

4. 계산 감속량

전체 검색인 경우 크기 N인 코드북 일때 비교 회수가 N번이지만, 본 알고리즘에서는 SVQ코드북의 크기 비교회수 M과 k개의영역에 속하는 후보 코드워드 갯수를 합한 횟수가 계산량이 된다.

산술적으로 평균값을 쉽게 예측 해 볼 수 있지만 실제 입력 되는 벡터의 종류에 따라서 또한 SVQ코드워 각각의 영역에 속하는 VQ 코드워드의 갯수에 따라서 계산량이 달라질 수 있기 때문에 실험을 통해서 계산량을 측정하였다. 그림 8에 k와 SVQ크기와의 비율을 평균적으로 하고 계산량 감속비

율을 증속으로 하여 그래프를 그렸다. 1차 직선으로 나타나게 되는데 SVQ의 크기에 따라서 계산량이 결정 된다는 것을 쉽게 알 수 있다. 그림 7에서 분류율이 거의 최적 검색과 같은 수준이 유지되는 k/SVQ 의 비율이 0.5 일 때 그에 해당하는 계산량 감소는 약 30% 이상이 된다. 따라서 기존의 VQ계계를 무너뜨리지 않고 순수히 계산량만을 감축할 수 있다는 것이다.

4. John Makhoul, Salim Roucos and Herbert Gish, "Vector Quantization in Speech Coding," Proceedings of the IEEE, Vol. 73, No. 11, November 1985.

5. L. C. Stewart, R. M. Gray, and Y. Linde, "The design of trellis waveform coders," IEEE Trans. on Communications, Vol. COM-30, No. 4, April, 1982.

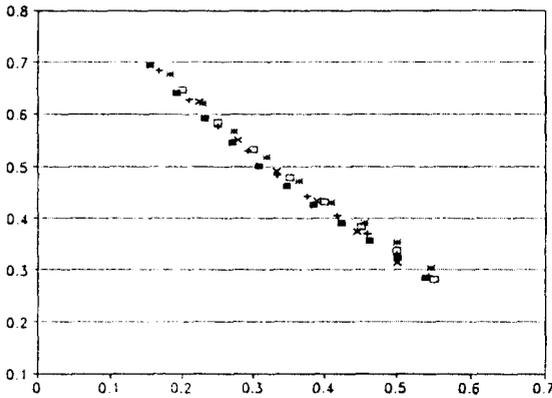


그림 8. 계산 감축량

IV. 결론

기존에 형성된 VQ 코드북의 코드워드를 검색하는 시간을 단축 시키는 알고리즘을 소개하였다. 간단히 구성할 수 있는 SVQ의 추가와 인코딩 부분의 약간의 수정으로 그 속도를 향상시킬 수 있는 방법이다. 이 방법은 기존의 코드북을 그대로 이용할 수 있고, 아주 약간의 기억용량만을 필요로 하며, 간단한 알고리즘의 추가로 쉽게 구성할 수 있다. 분류율이나 계산 감축량은 SVQ의 크기에 따라서 좌우되지 않고 k/SVQ 크기 비에 따라서 결정되며, 이 비가 0.5 일 때 인코딩 결과 분류율은 0.997 이상에 달하며 계산량 감축은 30%이상으로 나타났다.

참고 문헌

1. Robert M. Gray, "Vector Quantization," IEEE ASSP Magazine, April 1984.

2. Jay G. Wilpon and Lawrence R. Rabiner, "A Modified K-Means Clustering Algorithm for Use in Isolated Word Recognition," IEEE Trans. on Acoustic, Speech and Signal Processing, Vol. ASSP-33, No.3, June 1985.

3. Yoseph Linde, Andrés Buzo and Robert M. Gray, "An Algorithm for Vector Quantization Design," IEEE Trans. on Communications, Vol. COM-28, No. 1, January 1980.