유전자 알고리즘을 이용한 벡터 양자화

Vector Quantization using Genetic Algorithm

임 현 택*, 송 근 배*, 김 정 우**, 이 행 세*

Hyun Taeg Lim*, Guen Bae Song*, Jung Woo Kim**, Haing Sei Lee*

*School of Electrical and Electronics Engineering, Ajou University

**Department of Electronics, Koje College

E-mail : limht@madang.ajou.ac.kr

요 약

본 논문에서는 유전자 알고리즘(genetic Algorithm)을 사용하여 벡터 양자화(vector quantization : VQ)를 수행하는 방법을 제안하고자 한다. 벡터 양자화를 수행하여 코드북(codebook)을 생성할 때 생성된 코드북과 학습벡터(training vector)사이에는 반드시 양자화 오차(quantization error)가 발생하는데 기존의 K-means 알고리즘을 사용하여 코드북을 생성했을 경우 양자화 오차를 줄이는데 한계가 있었다. 본 논문에서 제안하는 유전자 알고리즘을 이용한 벡터 양자화는 이 양자화 오차를 감소시키기 위해서 연구되었다. 제안한 방법의 성능을 평가하기 위해 음성데이터를 기존의 K-means 알고리즘에서 클러스터의 중심을 선택하는 방법중의 하나인 Minimax방법으로 코드북을 생성하여 제안한 방법과 양자화 오차를 비교한 결과 양자화 오차가 감소됨을 알 수 있었다.

I. 서 론

벡터 양자화는 데이터를 압축하는데 널리 쓰이는 방법 중의 하나이다. 벡터를 M개의 클러스터(cluster)로 분류하여 데이터량을 감소시킨다. VQ 코드북의 크기를 $M = 2^B$ 이라고 정의하면(일반적으로 이것을 B-bit 양자화 하였다고 한다), 학습벡터의 수 n은 M보다 충분히 커야한다. 일반적으로, VQ 코드북이 잘 동작하기 위해서는 n이 M보다 적어도 10배 이상은 커야한다고 알려져 있다[1]. 그림 1은 기본적인 VQ의 블록 다이어그램이다.

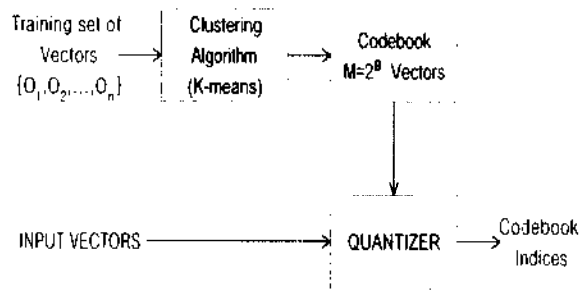

그림 1. VQ의 블록 다이어그램

K-means 알고리즘은 VQ를 수행하는데 가장 잘 알려진 알고리즘이다. K-means 알고리즘은 많은 실제적인 문제에 성공적으로 사용되어져 오기는 하였지만, 어떤 경우에는 최종적으로 찾은 해가 국부 최소값(local minimum)이 얻어질 수 있다는 단점을 가지고 있다[5]. 또 코드북의 크기를 몇 개로 하는 것이 가장 효율적인 VQ를 수행하는 것인지 알 수 없고 단지 시행착오를 통

해 코드북의 크기를 정해야만 한다는 단점도 가지고 있다. 코드북을 생성시키는 과정에서 반드시 양자화 오차(quantization error)가 발생한다. 여기서 양자화 오차란 코드북을 만드는 과정에서 학습벡터가 어떤 한 클러스터로 분류된 후, 그 분류된 클러스터의 중심(centroid)과 학습벡터간의 차이를 말한다. 양자화기의 좋고 나쁨은 바로 이 양자화 오차가 크고 작음에 의해 결정된다. K-means 알고리즘에서 클러스터의 중심을 선택하는 방법은 여러 가지가 있으나 그중 대표적인 것으로 Minimax 방법과 평균법등이 있다.

본 논문에서 제안하는 알고리즘은 유전자 알고리즘(genetic Algorithm)을 이용하여 K-means 알고리즘에서 클러스터의 중심을 선택하는 방법인 Minimax 방법보다 양자화 오차가 더 작은 코드북 벡터를 생성하는 것이다.

Ⅱ. 유전자 알고리즘

유전자 알고리즘은 자연의 법칙인 "적자생존의 원리"에 근거를 두고 있다. 즉 자연계에서 일어나는 환경에 맞추어 진화해 가는 과정을 인공적으로 만들어 알고리즘화한 것이다. 유전자 알고리즘에는 population이라는 개념이 도입되는데 이는 문제 영역에 있어서 후보해들의 집합을 의미한다. 이 후보해들 각각은 생물학에서 빌려온 용어인 염색체(chromosome)라고 불리운다. 염색체는 유전자(gene)로 구성되며 각각 적응 함수값(fitness value)을 갖는다.

유전자 알고리즘이 해를 찾는 문제에 있어서 우수한 잠재력을 보이는 것은 선택(selection), 교배(crossover), 그리고 돌연변이(mutation)라는 세 개의 유전자 조작자들 때문이다. 특히 교배는 유전자 알고리즘의 강력한 해 찾기 능력에 있어서 빠질 수 없는 역할을 한다[4]. 한 population은 이러한 유전자 조작과정을 거치면서 점점 준최적해(quasi- optimal solution)에 도달하게 된다. 유전자 알고리즘이 반드시 최적해를 제시하지는 않는다. 하지만 많은 연구가들이 해를 찾는 기법으로서 이 방법을 택하는 이유는 다른 전통적인 방법에 비해 몇 가지 장점들을 가지고 있기 때문이다. 한번에 하나의 해를 찾는 전통적인 방법은 국부 최소값(혹은 최대값)에 빠진다는 단점을 가지고 있지만 한번에 하나의 후보 해를 다루는 것이 아니라 여러 후보 해들의 집합이라고 할 수 있는 population을 만들어 동시에 해를 찾으므로 지역적 오류에 빠지는 것을 방지할 수 있다. 또 문제를 나타낼 수 있는 염색체와 이것을 평가할 적응함수만 있으면 되므로 단순성을 유지한다[2][3].

Ⅲ. 유전자 알고리즘을 이용한 벡터 양자화

이 절에서는 제안하는 유전자 알고리즘을 이용한 코드북 생성에 대해서 기술한다. n개의 학습벡터를 $O_1, O_2, \cdots, O_n$ 이라고 하고 각각의 벡터들은 P차원의 벡터라고 하면 $O_i=(O_{i1}, O_{i2}, \cdots, O_{ip})$이다. n개의 벡터들은 M개의 클러스터로 분류되는데 이 클러스터의 중심(centroid)을 각각 $C_1, C_2, \cdots, C_M$ 이라고 하자.

초기 중심은 n개의 학습벡터들 중에서 랜덤하게 M개의 벡터를 선택하여 구성된다. 이들 M개의 중심을 기준으로 n개의 벡터들을 분류하는데 (식 1)과 같은 유클리디언 거리(Euclidean distance)를 이용하여 M개의 클러스터를 형성한다.

$$d(O_i, C_j) = \sum_{q=1}^{p} (O_{iq} - C_{jq})^2 \qquad \text{(식 1)}$$

여기서 $O_i=(O_{i1}, O_{i2}, \cdots, O_{ip})$이고 $1 \le i \le n$ 이며, $C_j=(C_{j1}, C_{j2}, \cdots, C_{jp})$이고 $1 \le j \le M$ 이다. 만약 $d(O_i, C_j) \le d(O_i, C_k)$이 모든 k에 대하여 만족하면 O_i는 중심 C_j가 구성하는 클러스터에 분류된다. 여기서 $1 \le k \le M$ 그리고 $k \ne j$ 이다. 다시 말하면 n개의 벡터들이 M개의 중심중에서 가장 유사한 (유클리디언 거리값이 가장 작은) 중심에 분류되어 M개의 클러스터를 형성한다.

유전자 알고리즘은 세대(generation)을 반복하면서 수행되는데 각 세대마다 선택, 교배, 돌연변이라는 유전자 조작자들을 포함한다.

선택(selection)에서는 먼저 각 벡터들의 적응함수값이 계산된다. 적응함수는 유전자 알고리즘에서 가장 중요한 부분 중의 하나이다. 제안하는 알고리즘에서는 적응함수(fitness function)를 (식 2)와 같이 정의하였다.

$$\text{적응함수} = \frac{Interdistance}{Intradistance} \quad \text{(식 2)}$$

여기서 "Intradistance"는 각각의 학습벡터들과 이 벡터들이 분류되어진 클러스터의 중심 사이의 유클리디언 거리의 합이고, "Interdistance"는 각각의 벡터들과 자신이 분류되어진 클러스터의 중심을 제외한 나머지 중심들간의 유클리디언 거리 중에서 가장 작은 값의 합으로 정의한다. 이것을 수식으로 나타내면 각각 (식 3)과 (식 4)와 같다.

$$Intradistance = \sum d(O_i, C_j) \quad \text{(식 3)}$$

여기서 C_j는 벡터 O_i가 속한 클러스터의 중심이다

$$Interdistance = \sum_{i=1}^{n} \{\min\ d(O_i, C_k)\} \quad \text{(식 4)}$$
$$1 \le k \le M \text{ and } k \ne j$$

적응함수를 Interdistance 대 Intradistance 비율로 정한 이유는 클러스터링(clustering)의 목표가 Interdistance 대 Intradistance 비율을 최대화 시키는데 있기 때문이다 [1]. Interdistance는 크면 클수록 Intradistance는 작으면 작을수록 VQ의 성능이 좋아진다. 각 벡터의 적응함수값이 계산된 후 선택 연산은 룰렛-휠(roulette wheel)방법에 의해 행하여진다. 룰렛-휠 방법은 적응함수값이 높은 벡터에 대해서는 많이 선택될 기회를 주고 반대로 적응함수값이 낮은 벡터에 대해서는 적은 기회를 제공한다.

두 번째로 교배(crossover)연산자는 선택 연산자에 의해 선택되어진 벡터들을 교배한다. 랜덤하게 두 쌍의 벡터를 선택하고 교배할 임의의 위치를 선정하고난 후, 두 벡터의 일 부분을 서로 바꾼다. 교배가 일어날 위치는 난수에 의해 결정되는데, 교배가 일어날 두 벡터에서 이 위치는 같다. 또 항상 교배가 일어나는 것이 아니라 교배를 할것인가 안할것인가는 교배확률 P_c의 지배를 받는다.

세 번째로 돌연변이(mutation)연산자는 돌연변이 확률 P_m에 의해 이루어지고 돌연변이가 일어날 위치를 임의로 결정한 후, 그 위치에서의 유전자(gene)값을 변화시킨다. gene값이 이진수였을때는 0을 1로 또는 1을 0으로 바꾼다. 그러나 본 논문에서 다루는 데이터는 실수이므로 gene값에서 0.05를 빼주는 방식을 택하였다.

이러한 유전자 조작자들을 이용하여 한 세대(generation)를 만들고, 각 클러스터에 새로이 만들어진 벡터 중에서 적응함수값이 가장 큰 벡터를 새로운 중심으로 선택하여 다시 클러스터링한 뒤 미리 정해놓은 횟수만큼 위의 과정을 반복한다. 그림 2는 위의 과정을 흐름도(flowchart)로 나타내었다.

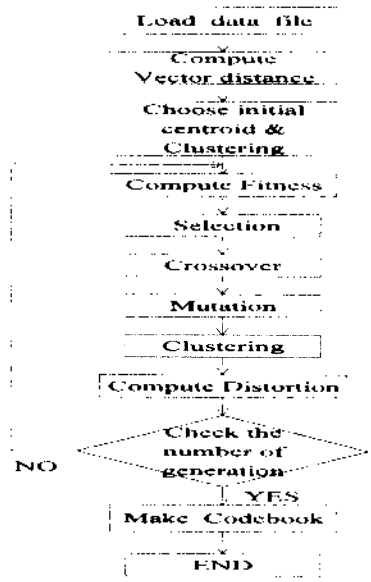

그림 2 유전자 알고리즘을 이용한 코드북 생성의 flowchart

Ⅳ. 실험 및 결과

제안한 알고리즘의 성능을 평가하기 위해 기존의 평균법과 Minimax 알고리즘을 이용한 코드북 생성방법과 비교하며 실험을 수행하였다. 남성화자가 발음한 영부터 구까지의 숫자음을 12차 선형 예측 계수(Linear Prediction Coefficient : LPC) 로 변환하여 12차원의 학습 벡터를 구성한 후 평균법, Minimax 방법 그리고 제안한 방법으로 각각 코드북을 생성하면서 Intradistance, Interdistance 그리고 Interdistance 와 Intradistance 비율을 계산하였다. <표 1> <표 2> <표 3>는 학습 벡터의 수가 각각 1200개와 5000개일 때 Intradistance, Interdistance 그리고 Interdistance 대 Intradistance 비율을 계산한 결과이다. 표에서 왼쪽의 숫자는 클러스터

의 수를 표시한 것이다.

	Minimax			Genetic			Mean		
	intra	inter	ratio	intra	inter	ratio	intra	inter	ratio
32	8.346	16.96	2.032	7.785	16.88	2.186	6.061	13.64	2.251
64	5.741	10.76	1.874	5.478	11.01	2.010	4.298	9.504	2.211

표 1. 학습벡터의 수가 1200개일 때

	Minimax			Genetic			Mean		
	intra	inter	ratio	intra	inter	ratio	intra	inter	ratio
32	15.99	29.81	1.863	13.81	30.88	2.231	11.46	24.78	2.162
64	10.14	19.06	1.879	9.72	20.22	2.081	7.809	16.91	2.145
128	7.388	13.63	1.845	6.99	13.85	1.976	5.794	12.29	2.121

표 2. 학습 벡터의 수가 2500개일 때

	Minimax			Genetic			Mean		
	intra	inter	ratio	intra	inter	ratio	intra	inter	ratio
64	22.76	40.18	1.765	21.16	41.92	1.981	17.33	35.83	2.067
128	17.03	29.15	1.711	15.84	29.36	1.853	13.09	26.39	2.016
256	12.28	21.18	1.724	11.64	21.73	1.867	9.027	18.98	2.103

표 3. 학습 벡터의 수가 5000개일 때

앞에서도 언급하였듯이 Interdistance 대 Intradistance의 비율을 최대화시키는 코드북을 생성시키는 것이 목표이다[1]. 위의 결과를 보면 Minimax 방법에 비해 제안한 알고리즘이 Interdistance 대 Intradistance 비율이 평균 9.08%향상되었다. 평균법으로 코드북을 생성시킬 때 Interdistance 대 Intradistance 비율이 제안한 알고리즘에 비해 크기는 하지만 평균법은 Intradistance의 관점에서만 코드북을 생성시켰고 Interdistance는 전혀 고려하지 않은 방법이므로 Interdistance의 관점에서 나쁜 결과를 예상할 수 있다. 실험결과에서도 평균법은 다른 알고리즘에 비해 Intradistance는 작지만 그에 반해 Interdistance가 가장 작음을 알 수 있다.

V. 결 론

본 논문에서는 유전자 알고리즘을 사용하여 벡터 양자화를 수행하는 방법을 제안하였다. 평균법은 제안한 방법에 비해 Intradistance의 관점에서는 좋지만 Interdistance의 관점에서는 다른 방법들에 비해 가장 좋지 않았고 Minimax 방법은 평균법에 비해 Interdistance는 향상이 되었으나 Intradistance 측면에서는 가장 좋지 못했다. 그러나 제안한 알고리즘에서는 Minimax 방법과 비교하여 Interdistance는 비슷한 수준을 유지하면서도 Intradistance를 평균 7.03% 감소시킬 수 있었다. 그래서 Interdistance 대 Intradistance 비율이 Minimax 알고리즘에 비해 평균 9.08% 향상되었다.

그러나 클러스터의 숫자를 미리 사용자가 정해야 하므로 몇 개의 클러스터가 가장 좋은 성능을 보이는지는 알 수 없는 K-mean 알고리즘의 단점을 그대로 가지고 있다.

향후 연구는 바로 이러한 단점을 보완하기 위해 가장 효율적인 양자화를 수행할 수 있는 클러스터의 숫자를 찾는 방법을 연구할 예정이다.

VI. 참 고 문 헌

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

[2] 김 성곤, 김 인택, 박 대희, 박 주영, 신 요안, "유전자 알고리즘", 그린, 1996.

[3] Melanie Mitchell, "An Introduction to Genetic Algorithms", Massachusetts Institute of Technology",1997.

[4] Park, k. and B. Carter, "On the effectiveness of Genetic Search in Combinatorial Optimization", Proc. 10th ACM Symposium on Applied Computing, Genetic Algorithms and Optimization Track, Feb., 1995.

[5] S. Z. Selim and M. A. Ismail, "K-means type algorithm : generalized convergence theorem and characterization of local optimality", *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, pp. 81-87, 1984.