VQ 코드북 디자인을 위한 개선된 Modified K-Means 알고리듬

Improved MKM Algorithm for Vector Quantizer Design

백 성 준*, 김 종 득**, 배 명 진**, 성 굉 모*
(Seong Joon Baek*, Jong Deuk Kim**, Myung Jin Bae**, Koeng Mo Sung*)

요 약

본 논문에서는 MKM(Modified K-Means) 알고리듬의 성능을 개선하기 위해 새로운 학습알고리듬을 제안한다. MKM 알고리듬에서 새로운 코드워드는 이전 코드워드와 새로 얻은 중심점을 이은 직선 상의 임의점으로 선택된다. 따라서 MKM 알고리듬은 통계적 이완 방법의 코드북 교란 알고리듬으로 이해될 수 있다. MKM 알고리듬을 통계적 이완 알고리듬과 비교해보면 도입되는 교란의 양이 상대적으로 적고 그 교란 자체도 임의적이지 않다는 걸 알 수 있다. 따라서 MKM 알고리듬에 도입되는 교란의 양을 보다 크고 임의적이게 하면 MKM 알고리듬이 국소 최적화에 빠질 가능성이 줄어들 것이다. 따라서 본 논문에서는 MKM 알고리듬의 코드북 갱신과정을 변화시킨 새로운 알고리듬을 제안하였으며, 화상 데이터와 음성 데이터를 이용하여 실험한 결과 제안된 알고리듬이 MKM 알고리듬보다 우수한 성능을 보인다는 걸 확인할 수 있다.

ABSTRACT

In this paper, we propose a new VQ design algorithm which improves the performance of the Modified K-Means(MKM) algorithm. In MKM algorithm new codeword is found somewhere on the line from the old codeword through the new centroid for the cluster. So, it can be thought as a kind of decoder perturbation of stochastic relaxation(SR-D). When compared with perturbation in SR-D algorithm, the perturbation introduced in MKM algorithm is relatively small and is not random. Then, if we perturb the codeword more randomly with larger magnitude, we can expect that the algorithm have more chance not to be trapped in a local minimum. So, we modifies the codeword updating step of MKM algorithm to improve the performance. Experimental results using image data and speech data show that the performance of the proposed algorithm is superior to that of MKM algorithm.

I. Introduction

Vector quantization(VQ) is a very efficient approach to low-bit-rate image compression[1]. One major advantage of VQ is that the hardware structure of the encoder, and especially the decoder, is very simple. The data to be encoded are first processed to yield a set of vectors. Then a codebook is generated using, for example, the iterative clustering algorithm proposed by Linde, Buzo, and Gray-LBG algorithm[2]. The input vectors are then individually quantized to the closest codewords in the codebook. Compression is achieved by using the indices of codewords for transmission or storage. Reconstruction of the data can be implemented by the table lookup technique: the indices are simply used as addresses to the corresponding codewords in the codebook.

The key step in VQ is to generate a good codebook from the training images. The K-means and the closely related generalized Lloyd clustering algorithm proposed by Linde, Buzo, and Gray are typically used to generate a codebook. These algorithms are basically iterative processes to minimize the distortions between the training vectors and their corresponding codewords.

It is known that K-means algorithm converges to a locally optimal codebook in certain conditions but it converges to a different codebook when a different initial condition is applied. It has also been observed that both convergence speed and performance of the codebook depend on the initial codebook. Thus, many algorithms were proposed to obtain a good initial codebook, including the well known splitting, pruning, pairwise nearest neighbor design, and maximum distance initialization[3][4].

Zeger *et al.*[5] proposed other types of VQ algorithm

* Seoul National University
** Soongsil University

Manuscript Received: July 7, 1998.

using stochastic relaxation such as simulated annealing to get a better codebook than K-means algorithm. They introduced the concepts of encoder perturbation and decoder perturbation. Each perturbation is implemented by corrupting the training data or codeword by additive noise. By reducing the amount of perturbations at each iteration, this method showed performance improvement over K-means algorithm for image and speech data. However, the algorithm requires significantly more computation time than K-means algorithm.

Recently, a new VQ design algorithm called modified K-means algorithm(MKM)[6] is proposed. It also iteratively updates the initial codebook like the conventional K-means algorithm but modifies the codebook updating step only. Experimental results show that the algorithm converges to a better locally optimal codebook with the same initial codebook and needs less computation time than K-means algorithm.

In this paper, we propose a new VQ design algorithm which exploits the concepts of stochastic relaxation to improve the performance of the MKM algorithm. In MKM algorithm, a new codeword is found somewhere on the line from the old codeword through the new centroid for the cluster. So, it can be thought as a kind of decoder perturbation of stochastic relaxation(SR-D). When compared with perturbation in SR-D algorithm, the perturbation introduced in MKM algorithm is relatively small and is not random. Then, if we perturb the codeword more randomly with larger magnitude, we can hope that the algorithm have more chance not to be trapped in a local minimum. So, the proposed algorithm modifies the codeword updating step of MKM algorithm. Experimental results using image data and speech data show that the performance of the proposed algorithm is superior to that of MKM.

II. The Proposed Algorithm

The proposed algorithm is almost the same as MKM algorithm. So, before showing the proposed algorithm, we describe the detailed procedure of the MKM algorithm.

Let $S(y)$ be the set of training vectors that have the same codeword y in the codebook. The centroid of the set $S(y)$ is defined as

$$centroid(S(y)) = \frac{1}{|S(y)|} \sum_{x \in S(y)} x$$

where $|S(y)|$ denotes the number of training vectors in $S(y)$. The average distortion between training vectors and their closest codewords is defined as

$$D = \frac{1}{|T|} \sum_{x \in T} d(x, \hat{x})$$

where T is the training set and $\hat{x}$ is the closest codeword of the training vector x. Let C_i be the codebook in the i-th iteration and ε be the distortion threshold to decide when to terminate an iterative process. Then, the MKM algorithm is given as follows:

Step 0: An initial codebook C_0 is given. Set $i=0$ and $D_{-1} = \infty$ \$.

Step 1: For each training vector x, find its closest codeword in the codebook C_i.

Step 2: Update C_{i+1} with $y^{i+1} = centroid(S(y^i)) + \text{scale} \cdot (centroid(S(y^i)) - y^i)$

Step 3: Compute the average distortion D_i. If $(D_{m+1} - D_m)/D_m \leq \varepsilon$, then halt. Otherwise, set $m = m+1$ and go to Step 1.

The scale factor from the above algorithm is between −1 to 1. It is reported that the performance is relatively better when the value of scale is between 0.7 and 0.9. Considering both the convergence speed and performance, the value of scale is set to 0.8[6]. In SC-D algorithm, codebook updating step is modified in following way.

Step 2: Update C^{i+1} using new codewords $y^{i+1} = centroid(S(y^i)) + \triangle y^i(T_i)$

$\triangle y^i(T_i)$ is controlled by the cooling schedule. An effective cooling schedule is found to be

$$T_i = \sigma_y^2 (1 - i/I)^3$$

where I is the total number of iterations to be run and σ_y^2 is the average variance of the codeword components.

Comparing two algorithms, we can find $\text{scale} \cdot (centroid(S(y^i)) - y^i)$ in MKM algorithm replaces $\triangle y^i(T_i)$ in SR-D algorithm. These perturbation factors can be viewed as playing the same role, which controls the cooling process. The perturbation factor in MKM algorithm is relatively small compared to that in SR-D algorithm because the variance of the codeword components is larger than the distance between current centroid and current codeword.

But in the context of simulated annealing, the high levels of perturbation noise essentially randomize the state. As the noise is reduced, the amount of energy that can be added also decreases, making it more difficult for the algorithm to leave deep minima in a single step. On the other hand, shallow local minima will not confine the state, and since the added noise goes to zero, it will be much more probable that the state will be in a deep minimum of the energy. So, if we increase the perturbation factor in MKM algorithm, we can hope that the state is more randomized and the performance of the MKM algorithm is improved. Thus we modifies the MKM algorithm in the following way.

Step 0: Using MKM algorithm, obtain C_0 and C_1 and set $i=2$.

Step 1: For each training vector x, find its closest codeword in the codebook C_i.

Step 2: Update C_{i+1} with $y^{i+1} = centroid(S(y^i)) + \text{scale} \cdot (centroid(S(y^i)) - y^{i-1})$

Step 3: Compute the average distortion D_i. If $|D_{i-1} - D_i| / D_i \leq \epsilon$, then halt. Otherwise, set $i=i+1$ and go to Step 1.

Step 3 can be viewed as codeword jiggling step by alternating codeword updating step of MKM algorithm. This codeword jiggling introduces perturbation at more randomized fashion than MKM algorithm because new codeword is calculated using both of the information in current and previous iteration. In MKM algorithm, the perturbation is based on the distance between current centroid and current codeword. Since the distance between current centroid and previous codeword is usually larger than that between current centroid and current codeword, perturbation in the proposed algorithm is larger than that in MKM algorithm. So this modification essentially introduces the larger and more random perturbation to MKM algorithm.

As iteration goes on, codewords jiggle less and gradually settle down. Thereby, the perturbation introduced in the algorithm gradually decreases with time and convertgence is achieved. Next section we will show the effectiveness of this codeword jiggling algorithm through experiments.

III. Experimental Results

We performed an experiment using four images and speech data to evaluate the algorithm. The images are 512×512 monochrome still images with 256 gray levels and speech data is 143500 samples sampled at 16 khz. $16(4 \times 4)$ dimension vector is used for the image data and 4 and 8 dimension vector is used for the speech data. In the case of the image data, the quality of the encoded data is evaluated by the peak signal to noise ratio(*PSNR*), the most popular measure of quality in image coding, while the *SNR* is used for the speech data. The *PSNR* and *SNR* are defined as

$$PSNR = 10\log(255^2/MSE), \quad SNR = 10\log(P_s/MSE)$$

where P_s is the signal power and *MSE* is a mean-squared-error.

Initial codebook is obtained using well known splitting algorithm[2]. The proposed algorithm is termed IMKM (Improved Modified K-Means) algorithm for simplicity. Before implementing the algorithm, it should be noted that the training algorithm often creates null clusters due to limited number of training data. In the case, we split the cluster with largest distance as usual. The scale factor of MKM and IMKM algorithm is set to 0.8.

Table 1 shows the *PSNR* values of each algorithm for image data. It can be observed that IMKM algorithm outperforms the K-Means and MKM algorithm by 0.26dB and 0.15dB on the average. When the size of the codebook varies from 256 to 1024, average *PSNR* values of the IMKM algorithm outperforms the K-Means algorithm by 0.17dB and 0.23dB and 0.37dB. This indicates that the proposed algorithm is effective for the larger codebook. The same tendency is observed in the case of speech data. However, the improvement in performance is accompanied by an increase in the computational complexity. Comparing the number of iteration of each algorithm, we can find that computational requirement of the IMKM algorithm is approximately doubled.

Table 1. Performance comparison using image data. Parenthesis denotes the number of iteration.

Codebook Size	Training Method	Image			
		Lena	Man	Boats	Baboon
256	K-Means	31.83(31)	29.61(25)	24.48(26)	24.15(25)
	MKM	31.90(22)	29.67(39)	28.53(19)	24.20(23)
	IMKM	32.04(45)	29.78(39)	28.66(47)	24.27(43)
512	K-Means	32.93(22)	30.56(22)	29.36(18)	24.87(26)
	MKM	33.03(20)	30.66(22)	29.47(22)	24.93(24)
	IMKM	33.24(40)	30.82(43)	29.61(44)	25.05(44)
1024	K-Means	34.25(18)	31.72(19)	30.38(17)	25.67(17)
	MKM	34.45(20)	31.89(19)	30.54(18)	25.82(18)
	IMKM	34.66(39)	32.10(58)	30.74(39)	25.99(50)

Table 2 shows the *SNR* values of each algorithm for speech data. In this table, IMKM algorithm outperforms the K-Means and MKM algorithm by 0.52dB and 0.17dB on the average. As mentioned above, when the size of the codebook increases from 256 to 1024, average SNR values of the IMKM algorithm outperforms the K-Means algorithm by 0.22dB and 0.36dB and 0.99dB.

Table 2. Performance comparison using speech data. Parenthesis denotes the number of iteration.

크기	방법	Speech	
		4 Dimension	8 Dimension
256	K-Means	17.06(26)	13.28(29)
	MKM	17.10(17)	13.40(26)
	IMKM	17.27(39)	13.51(42)
512	K-Means	19.08(22)	15.04(20)
	MKM	19.19(18)	15.18(16)
	IMKM	19.42(41)	15.42(33)
1024	K-Means	19.83(16)	17.36(15)
	MKM	20.07(18)	17.62(17)
	IMKM	20.87(52)	17.84(42)

IV. Conclusion

In this paper, we propose a new VQ design algorithm which exploits the concepts of stochastic relaxation to improve the performance of the MKM algorithm The proposed algorithm modifies the codeword updating step of MKM algorithm. The modification is made by alternating codeword updating step of MKM algorithm. This modification essentially introduces the larger and more random perturbation to MKM algorithm. Experimental results using image data and speech data show that the performance of the proposed algorithm is superior to that of MKM algorithm. However, the improvement in performance is accompanied by an increase in the computational complexity.

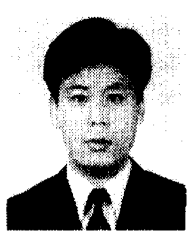
Reference

1. R. M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, vol. 1, pp. 4-9, Apr., 1984.
2. Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan., 1993.
3. I. Katsavounidis, C.-C. J. Kuo, and Z. Zhang, "A new initialization technique for generalized lloyd iteration," *IEEE Signal Processing Letters*, vol. 1, No. 10, pp. 144-146, Oct., 1994.
4. A. Gersho, R. M. Gray, *Vector Quantization and Signal Compression*, KAP, 1992.
5. K. Zeger, J. Vaisey, and A. Gersho, "Globally Optimal Vector Quantizer Design by Stochastic Relaxation," *IEEE Trans. Signal Processing*, vol. 40, pp. 310-322, Feb., 1992.
6. D. Lee, S. Baek, and K.-M. Sung, "Modified K-means Algorithm for Vector Quantizer Design," *IEEE Signal Processing Letters*, vol. 4, No. 1, pp. 2-4, Jan., 1997.

▲SeongJoon Baek

SeongJoon Baek was born on Jan. 16 in 1967. He received the B.S. degree and the M.S. degree in electronics engineering from Seoul National University in 1989 and 1992, respectively. Currently, he is working toward Ph.D. degree in the same university. His research interests include speech coding, speech/speaker recognition and signal processing. He is a member of Acoustical Society of Korea.

▲JongDeuk Kim



JongDeuk Kim was born in Seoul on Feb. 16, 1966. He received the B.S. degree in Information and Telecommunication Engineering from Soongsil University in 1996, and enrolled the M.S. degree in Information and Telecommunication Engineering in Soongsil University. His research interests include speech synthesis, speech/speaker recognition and speech coding. He is a member of Acoustical Society of Korea.

▲MyungJin Bae

Refer to 1996. Vol. 14. No. 5.

▲Koeng-Mo Sung

Refer to 1995. Vol. 14, No. 2.