Design of Index Assignment in Vector-Quantizers under Channel Errors

Gal Ben-David and David Malah
Technion - Israel Institute of Technology
Dept. of Electrical Eng. Haifa 32000, Israel

Correspondence: Gal Ben-David, Mitzpe Adi 238, D.N. Nazareth Illit 17940, Israel
Tel: + 972-4-9860 398, Fax: + 972-4-9860 401, Cell: +972-54-4698 289
e-mail: gal@galium.co.il , malah@ee.technion.ac.il

Abstract

Vector-Quantization (VQ) is an effective and widely implemented method for low-bit-rate coding. In VQ systems, signals are approximated by codewords whose indices are transmitted through a communication channel. A common assumption in the design of VQ-based systems is that the compressed digital information is transmitted through a perfect channel. Under this assumption, quantization distortion is the only factor in the output signal quality and the assignment of channel symbols to the VQ reconstruction vectors is of no importance. However, in practice, channel errors cause degradation in overall system performance and the effect of channel errors on the coding system performance depends on the VQ index assignment. The index assignment problem is known to be a special case of the Quadratic Assignment Problem which is NP-complete. For a VQ with N reconstruction vectors there are $N!$ possible assignments, meaning that an exhaustive search over all possible assignments is practically impossible. In this work we present a method to design an efficient index assignment for symmetric memoryless channels. The algorithm is based on our previous work concerning lower and upper bounds on the performance of VQ systems under channel errors, over all possible assignments. These bounds are obtained by linear programming arguments and provide hints how good and bad index assignments should be built. In simulations, the performance of the assignment obtained from the proposed algorithm is compared with the lower bound as well as with the well known index switching algorithm. The algorithm is shown to construct better assignments in much less computational effort.