Vector Quantization for Lossless Textual Data Compression

WEE K. NG  CHINYA V. RAVISHANKAR

Department of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor, MI 48109-2122
E-mail: {wkn,ravi}@eecs.umich.edu

Vector Quantization (VQ) is a common data compression technique with successful applicability in speech and image coding. However, it has not been applied to textual data compression primarily because it is lossy, and also because textual data cannot be represented as vectors. Textual data constitutes a large portion of real world data that are conventionally compressed with techniques along the line of textual substitution.

VQ may be adapted for lossless textual data compression if the data exhibit vector structures. In particular, a textual relational database is a natural candidate for the application of VQ. A relation database consists of a collection of relations or tables. Each table contains a collection of n-tuples, each of which is a vector, or a point in n-dimensional space. A direct application of VQ to encode a table would be to find a set of representative tuples, and replace each tuple in the table with a codeword or index that indicates the representative tuple that is closest to it. Unfortunately, this method of coding is lossy; the original tuples are no longer completely recoverable. Instead of replacing each tuple in a relation by its codeword as VQ does, we also include the difference between the tuple and its representative tuple. This adaptation is called Loss-Less Vector Quantization (LLVQ).

LLVQ has two advantages over VQ. (1) In conventional VQ, the design of codebooks follows an iterative refinements (e.g., the LBG algorithm) which is locally optimal. LLVQ computes the codebook in constant time. (2) The performance of VQ is defined by the structure of the codebook. The codebook should be structured as to permit efficient searching of codewords during decoding. For large codebooks, this search process can become computationally intensive. With LLVQ, no searching is required.

LLVQ is not only lossless, it can be practically implemented for textual database compression. In particular, we have demonstrated that a relation of tuples may be encoded and allocated to physical disk blocks such that standard database operations such as access, insertion, deletion and update may be easily supported.

A full exposition is available in postscript by sending an email to the first author. Parts of this work also appeared in the following:


This work was supported in part by the Consortium for International Earth Science Information Networking.