MATLAB Exercise – Vector Quantizer (VQ) for Cepstral Coefficients

Program Directory: `matlab_gui\vector_quantization`
Program Name: `VQ_GUI25.m`
GUI data file: `VQ.mat`
Callbacks file: `Callbacks_VQ_GUI25.m`
TADSP: Section 11.6.2, pp. 722-732, Problem 11.31

This MATLAB exercise designs a set of 11 "digit-dependent" VQ codebooks (one for each of the 11 digits, zero-nine plus oh) with codebook sizes of from 1 (the grand centroid of cepstral vectors for each digit) to 64 codebook vectors, using a standard VQ design method (the Linde, Buzo, Gray algorithm).

Vector Quantizer (VQ) for Cepstral Coefficients – Theory of Operation

This exercise designs a set of Vector Quantization (VQ) codebooks for a set of 11 isolated digit sequences (the so-called "isolated tidigits" set). The program assumes that the tidigit sequences have all been carefully endpointed (to remove external background signal) and signal processed into a set of frames of size either $L = 12$ or $L = 2$ cepstral coefficients. The cepstral coefficients of size $L = 12$ are stored in the set of files: `cc_tidig_endpt_{1-9,0,Z}.mat` and can be loaded using the Matlab load command; i.e., `load cc_tidig_endpt_1.mat` giving the Matlab array (for the digit 1) `c(nframes,L)` where `nframes` is the total number of frames of cepstral coefficients, and with each frame consisting of $L = 12$ cepstral coefficients. (For cases in which we use $L = 2$ cepstral coefficients, the array `c(nframes,L)` is reduced from size $L = 12$ to $L = 2$ by merely using the first pair of cepstral coefficients for each cepstral vector.)

Vector Quantizer (VQ) for Cepstral Coefficients – GUI Design

The GUI for this exercise consists of two panels, 1 graphics panel, 1 title box and 10 buttons. The functionality of the two panels is:

1. one panel for the graphics display,
2. one panel for parameters related to the VQ analysis, and for running the program.

The graphics panel is used to display the following:

1. the curves of average VQ distance versus the number of VQ cells (on a logarithmic scale), for each of the 11 digits for which VQs were designed.

The title box displays generic information about the files used in the VQ design process. The functionality of the 10 buttons is:

1. an editable button that specifies the maximum size, `vqsize`, VQ codebook to be designed for each of the digits; (the default value for `vqsize` is 4),
2. an editable button that specifies the size, L_m, of the cepstral coefficient vector used for VQ codebook design; (the default value for L_m is 12),
3. an editable button that specifies the splitting factor on vq size, `epsilon`; (the default value for `epsilon` is 0.001),
4. an editable button that specifies the threshold on change for terminating VQ design iterations, `thresh`; (the default value for `thresh` is 0.001),
5. an editable button that specifies the prefix for cepstral analysis files; (the default for the prefix is 'cc'),
6. an editable button that specifies the header for the database files; (the default for the header is 'tidig_'),

7. an editable button that specifies the number of digits, `nrun`, for which VQs are to be designed; (the default for `nrun` is 11, i.e., the number of digits in the TI digits database),
8. a popupmenu button that lets the user choose 'no verbose output', or 'verbose output'; (the default is 'no verbose output' for normal runs; the option 'verbose output' is used for fine detail or when debugging the code),
9. a pushbutton to run the code and display the average distance of each VQ as a function of the number of VQ cells, for all `ndigit` runs,
10. a pushbutton to close the GUI.

Vector Quantizer (VQ) for Cepstral Coefficients – Scripted Run

A scripted run of the program 'VQ_GUI25.m' is as follows:

1. run the program 'VQ_GUI25.m' from the directory 'matlab_gui\vector_quantization',
2. using the editable buttons, set the initial values for the parameters as `vqsize=4`, $L_m = 12$, `epsilon=0.001` `, thresh=0.001` and `nrun=11`,
3. using the popuplists, set the values for prefix as cc, for dataset info as tidig_, and for verbosity as 'no verbose output',
4. hit the 'Run VQ Design' button to compute the set of VQs for the range of `vqsize` (1/2/4/.../`vqsize`) and for each of the `ndigit` data files; plot the average VQ distance (on a logarithmic scale) versus the number of vq cells (again on a logarithmic scale) for the set of VQs and for each of the ndigit training sets,
5. experiment with different values for `vqsize`, L_m, `epsilon, thresh`, and `nrun`,
6. hit the 'Close GUI' button to terminate the run.

An example of the graphical output obtained from this exercise using the TI digits cepstral vectors training set is shown in Figure 1. The graphics panel shows plots of average VQ distance (on a logarithmic scale) versus number of VQ cells (again on a logarithmic scale) for each of the `ndigit` training sets (usually 11 digits), and for VQ size from 1 to 64.

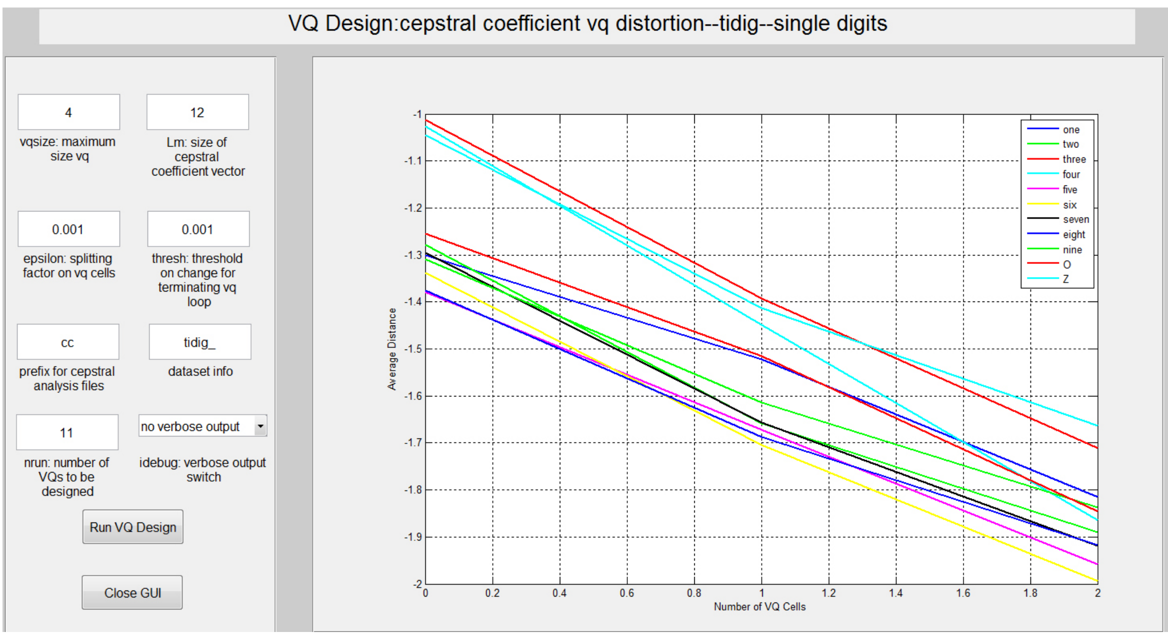

Figure 1: Graphical output from VQ design exercise. The graphics panel shows plots of average VQ distance (on a logarithmic scale) versus number of VQ cells (again on a logarithmic scale) for each of the 11 digits in the TI digits database.