Random Data Generator

Information Theory Lab 3

Objective

Understand the concepts of entropy and discrete memoryless source. Generate a data file from a memoryless source and attempt to compress it.

Theoretical notions

The entropy of a discrete memoryless source is defined as:

$$H(S) = \sum_i p(s_i) \cdot \log_2(p(s_i))$$

See the lecture notes for more details.

Practical issues

Exercises

1. Write a C program to generate random data, according to a specified distribution.
   - The program shall receive the name of the file, the data size and the distribution as command-line arguments:
     `entropy.exe data.txt 10000 0.5 0.1 0.1`
     The arguments are:
     - the name of the output file (`data.txt`);
     - the number of bytes to generate (10000);
     - the distribution (0.5 0.1 0.1, 0.3, four different messages). Note: the last probability is inferred automatically, equal to (1 - sum of all the others).
   - The program should follow the following steps:
     - Convert numerical data from command-line to actual number variables, with `sscanf()`, and display the probabilities. The distribution must be stored as a vector;

  - Allocate an array of `unsigned char` of necessary size;
  - Generate numbers randomly using `rand()`, then bring them range 0 - N and make according to distribution.
  - Convert messages to characters, i.e. with three different messages, generate the letters `a`, `b`, `c`
  - Write the final array to file (in binary format).
2. Generate a 10000-bytes long file with only two messages, with equal probability.
   a. Compute its entropy using the program from the previous lab;
   b. Compress the file using zip or 7zip. What is the compression ratio achieved? How is it related to the entropy?
3. Repeat the previous exercise with a distribution of four messages and eight, with equal probability.

Implementation hints

- The following C functions may be used for file-based operations. Look up their documentation on the Internet (e.g. *cplusplus.com*, or Google search).
  - `fopen(...)`, to open a file for reading;
  - `fread(...)`, to read byte data from the file;
  - `fwrite(...)`, to write byte data to the file;
  - `fclose()`, to close the file when finished.
- Use `sscanf()` to read numerical data from a string variable. The syntax is just like the usual `scanf()`, but with an extra parameter in front to indicate the string where the data is read from. For example, to read a float number from a string `str`, use:

  ```
  sscanf(str, "%f", &destination);
  ```
- The random number generator must be initialized with `srand(time(NULL))`, and then it is called like `x = rand()`.
- The `rand()` function returns a random integer in range 0 to RAND_MAX, with uniform distribution. The number can be made according to distribution by splitting the range 0 - RAND_MAX in subintervals proportionally with the probabilities, and comparing.
- Possible implementation: do `x = x - p(i) * RAND_MAX; i++;` until the result is negative. Then `i-1` is the final number. See explanations at blackboard.
-

Final questions

1. Can you make a file which cannot be compressed at all? How?