AP Computer Science — Project 34b - English Frequency Analysis

Dr. Paul Bailey — Thursday, March 19, 2020

The frequency of each letter in standard English is given by the following table.

A	B	C	D	E	F	G	H	I	J	K	L	M
.082	.015	.028	.043	.127	.022	.020	.061	.070	.002	.008	.040	.024
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
.067	.075	.019	.001	.060	.063	.091	.028	.010	.023	.001	.020	.001

A given block of text may be stripped and capitalized, and the frequency of each letter in the block may be analyzed. Then, these frequencies can be compared to the frequencies in standard English. In this way, several possible decryptions of a block of ciphertext may be analyzed in an attempt to find the actual plaintext.

We view a frequency distribution as a point in $\mathbb{R}^n$. The distance between two points $\vec{x} = (x_1, x_2, \ldots, x_n)$ and $\vec{y} = (y_1, y_2, \ldots, y_n)$ may be computed as

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

For our application, $n = 26$. Note that $0 < d_1 < d_2$ if and only if $d_1^2 < d_2^2$; thus, in practice, taking the square root is unnecessary if we only want to compare distances.

Program 1. In your `Crypt` project, create a new class `Analyze` to analyze blocks of text.

(a) Create a `public static double[] eng` variable of length 26 which represents a vector of standard frequencies. Put the frequencies from the table above into this array.

(b) Create a method which takes a block of text, loops through all the characters, skips the nonletters, and counts the frequencies of the letters. It then divides by the total number of letters to obtain frequencies. This method should return an array of frequencies (a probability distribution), stored as an array of doubles of length 26.

(c) Create a method which accepts two frequency distributions and computes the sum of the squared differences between corresponding entries. The smaller this number, the closer the distributions should be.

Program 2. In the `Analyze` class, write a method `public static int caesarKey(String s)` which loops through all possible keys, and assuming that `s` is encrypted using the Caesar cipher, uses frequency analysis to discovers and returns the correct decryption key. In a testing program, write out the key and its inverse (its inverse will be the original encryption key).

Program 3. In the `Analyze` class, write a method `public static int[] affineKey(String s)` which loops through all possible keys, and assuming that `s` is encrypted using the affine cipher, uses frequency analysis to discovers and returns the correct decryption key. The return value will be an array of two integers, the first being the decryption a and the second being the decryption b. In a testing program, write out the key and its inverse (its inverse will be the original encryption key).