A FREQUENCY ANALYSIS APPROACH TO CAESAR DECIPHERING

POGĂCEAN PAUL-ANDREI

Abstract. The purpose of this paper is to document the results of the project for the first bonus of this semester. This is to say, the paper will discuss a trivial approach to breaking the Caesar encrypted text by using frequency analysis and common distance functions in $\mathbb{R}^n$. It is common practice to view each frequency of a character over a text as a coordinate of a multi-dimensional space point. This implies that such a point describes a histogram. Therefore, the distance between 2 such points describes how similar their histograms are. This project evaluates the effectiveness of three distance metrics: Chi-Squared, Euclidean, and Cosine distances, in decrypting Caesar ciphers.

1. Introduction

We will define the following terms:

- frequency list = a list consisting of the frequencies of each character in a language or input.
- histogram = the frequency list with the property that on the first position we retain the frequency of 'a', on the second position the frequency of 'b'.
- C_i-the frequency of the i^{th} character in the input text.
- E_i-the frequency of the i^{th} character in the English alphabet according to the given frequency list.
- shift = a permutation of characters set on a fixed distance. For instance, for a shift of 7 characters, 'a' becomes 'h'.
- best shift = a shift for which the distance between the english histogram and input histogram is minimal.
- chi-squared distance = the distance defined by the formula:

$$\chi^2(C, E) = \sum_{i=a}^{z} \frac{(C_i - E_i)^2}{E_i}$$

- euclidean distance = the distance defined by the formula:

$$L_2 = \sqrt{\sum_{i=a}^{z} (C_i - E_i)^2}$$

Date: 11th March 2025.

- cosine distance = the distance given by the formula:

$$D_{\cos}(C, E) = 1 - \frac{\sum_{i=a}^{z} C_i E_i}{\sqrt{\sum_{i=a}^{z} (C_i)^2} \cdot \sqrt{\sum_{i=a}^{z} (E_i)^2}}$$

The algorithm tries to determine the most likely shift of characters in order to decipher the code. This is done by considering each character frequency as a coordinate of a point in a 26-dimensional space. Hence, a histogram becomes a point in this multidimensional space. By considering 2 histograms, namely the english and the one extracted from the text, one can determine the similarity between them by applying distance formulas in the vector space $\mathbb{R}^n$.

This project determines the most likely shift with the help of 3 distance functions: the chi-squared distance, the euclidean distance and the cosine distance and analyzes the results. The rest of this paper is structured as follows:

- Section 2-this section showcases a variety of tests, part of which are errors and texts that were not deciphered correctly.
- Section 3-this section discusses the tests in the previous section and describes the accuracy of the resulted algorithm. This is done by interpreting the standard cases, where the algorithm succeeded in deciphering the code and the special ones, in which the algorithm failed.

2. Tests

In this section we will discuss some input generated with the help of the encryption methods provided in the algorithm. For each input, the characters in it will be shifted by randomly fixed numbers from 0 to 26. Then, the result will be given to the algorithm to automatically decipher using each of the 3 distance functions.

For an easier representation, a table will be provided as follows:

- column 1 - contains the number of the text given as input from the list below. This is done either by using the file reading function or the console reading method.
- column 2 - the randomly chosen shift $\in [0, 26]$
- column 3 - a '*' symbol if the test passed (correct shift found) or a '-' symbol if it did not for the chi-squared distance.
- column 4 - a '*' symbol if the test passed or a '-' symbol if it did not for the euclidean distance.
- column 5 - a '*' symbol if the test passed or a '-' symbol if it did not for the cosine distance.

The list below recalls the texts given as inputs to the algorithm:

(1) hello world
(2) HELLO WORLD
(3) 'This is an encrypted message'

(4) Hellen loves calculus. Every day she wakes up and practices some basic topology on vector fields and limits. She has no limit to how many exercises she can take before crying. That's because she loves doing those exercises.
(5) pink fluffy unicorns dancing on rainbows
(6) A pony is a type of small horse, usually measured under a specified height at maturity. Ponies often have thicker coats, manes and tails, compared to larger horses, and proportionally shorter legs, wider barrels, heavier bone, thicker necks and shorter heads. In modern use, breed registries and horse shows may define a pony as measuring at the withers below a certain height; height limits varying from about 142 cm (14.0 h) to 150 cm (14.3 h). Some distinguish between horse or pony based on its breed or phenotype, regardless of its height. The word pony derives from the old French poulenet, a diminutive of poulain meaning foal, a young, immature horse. A full-sized horse may sometimes be called a pony as a term of endearment.
(7) Up until the 1980s, most natural language processing systems were based on complex sets of hand-written rules. Starting in the late 1980s, however, there was a revolution in natural language processing with the introduction of machine learning algorithms for language processing. This was due to both the steady increase in computational power (see Moore's law) and the gradual lessening of the dominance of Chomskyan theories of linguistics (e.g. transformational grammar), whose theoretical underpinnings discouraged the sort of corpus linguistics that underlies the machine-learning approach to language processing.
(8) this kshgdsb is a isjhfiwv noisy oiihadflbal text

Table 1: Decryption Results for Various Shifts Using Different Distance Metrics

Nr. crt.	**Shift**	**CSD**	**ED**	**CD**
1	0	-	-	-
1	2	-	-	-
1	4	-	-	-
1	19	-	-	-
1	16	-	-	-
2	1	-	-	-
2	3	-	-	-
2	9	-	-	-
2	18	-	-	-
2	23	-	-	-

Nr. crt.	Shift	CSD	ED	CD
3	3	*	*	*
3	22	*	*	*
3	23	*	*	*
3	24	*	*	*
3	25	*	*	*
4	26	*	*	*
4	1	*	*	*
4	2	*	*	*
4	3	*	*	*
4	4	*	*	*
5	5	*	-	-
5	10	*	-	-
5	15	*	-	-
5	20	*	-	-
5	25	*	-	-
6	7	*	*	*
6	14	*	*	*
6	21	*	*	*
6	2	*	*	*
6	9	*	*	*
7	1	*	*	*
7	2	*	*	*
7	3	*	*	*
7	4	*	*	*
7	5	*	*	*
8	11	*	*	*
8	13	*	*	*
8	17	*	*	*
8	19	*	*	*
8	23	*	*	*

3. Results and conclusions

To conclude, the number of characters in an input plays an important role in correctly deciphering the code. However, in some particular cases, where characters present on the first few positions in the english histogram match those in the histogram of the input, the deciphering can succeed even for small strings of 10-20 characters and noisy text. As the number of characters

grow, the characterization of the input becomes more rigorous, hence the distance functions indicate the right shift.

In some cases, such as 5, the correct shift is given by one single distance function, namely the Chi-Squared distance. This suggests that the choice of distance function is crucial when trying to decipher a Caesar code. Even in cases where the attribution has failed, the distances output different best shifts, indicating that each metric has its strengths depending on the text characteristics. Overall, the Chi-Squared distance demonstrated superior performance in the majority of cases, particularly with longer texts.