Stochastic Cryptoanalysis

Nicolò Alessandro Girardini, Mat. number 203265
nicolo.girardini@studenti.unitn.it

I. Introduction

This work aims at understanding how statistical tools can be used to make sense out of a sample of data and thus gather knowledge about a dataset.

In this task the problem is to retrieve an encrypted message. The encryption works such that each letter is represented by the two parameters of a gamma distribution α and λ. A fixed and constant number of samples *N* from that specific distribution is used to represent each letter, after being multiplied by either *1* or *-1*, so the decoder can learn the parameters from the sample, confront them with its mapping and retrieve the letter.

It is given a dataset with the known letters (which is not the complete alphabet). The tasks to address are four:

- Learn the constant *N*
- Learn the mappings of known letters
- Decrypt the secret message given
- By exploiting the decryption infer the missing letters and learn all possible mappings

The solution has been developed by fitting the given samples and associating the learned parameters to letters. To decrypt the message those mappings and the learned parameters inferred from the secret message has been compared.

II. Dataset and Tools

To assess all the given tasks it is needed to know how the dataset is built and the tools that will be used to solve the problem.

As for the first point, the dataset is composed by the secret message file and by some files needed for the basic mapping of known letters. Each of those files just contains a series of samples taken from different gamma distributions, defined as in Equation (1), in a column named **sample**.

$$f_X(x) = \frac{\lambda^\alpha x^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)}; \quad \alpha, \lambda, x > 0 \tag{1}$$

$$\Gamma(a) = \int_0^\infty y^{a-1} e^{-s} ds \tag{1}$$

The secret file *'secret.csv'* has samples to be divided in size N batches, obtaining the letters representations. It contains the secret message to decode: it is important to underline that it is a sentence that makes sense in english. Instead, the files containing the known letter to parameters mapping are organized like this: the file's name is a series of letters, like *'abcdef'*, representing the letters contained in the file, in the same order in which the associated samples appear.

Regarding the tools used, both to process data and to plot graphs, the language *R* was used, along with the library *MASS*, used to fit data to distributions [1].

Before processing any data, the absolute value function is applied to all samples, since they were randomly multiplied by either 1 or -1.

III. Letters' Representation

As described in Section I letters are encoded to the two parameters of the gamma distribution. To learn those two parameters it is needed the constant N, so the given files can be split into batches, each one corresponding to a letter.

Each file has in it 8 letters and the number of the rows, namely samples, contained is $80\,000$. Being N a constant the trivial result is that:

$$N = 10\,000$$

It is easy to see this even graphically, by looking at a simple plot of data present in one of the files after applying the absolute value function, as in Figure 1.

Following this it is learned that, being composed by $490\,000$ rows, the secret message contains 49 characters.

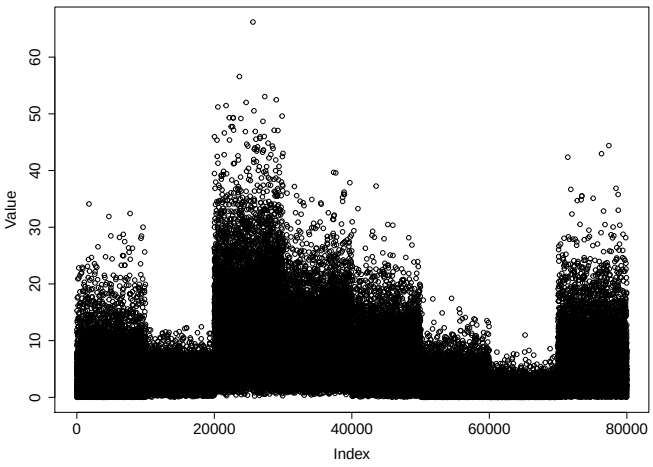

Figure 1. Absolute value of samples of one file

IV. Letters' Mappings with Parameters

As for what concerns the letter mappings the procedure applied is a standard distribution fitting.

Thanks to the R library *MASS* it is possible to apply to a sample of data the function *fitdistr*, which, by setting the right parameters, retrieves the *shape* and the *rate* of the gamma

distribution. These correspond respectively to α and $1/\lambda$ in the given task, enabling us to find the mapping parameters-letters. Its procedure involves performing a maximum log-likelihood estimation to search for the optimal parameters.

A bit of refinement is needed for this procedure, due to the fact that a letter could appear multiple times in different files of the given dataset. This allows to have more samples for some letters and thus more precision and confidence for the learned parameters.

The algorithm developed iterates over each file, fitting batches of samples of size N (so one character) and saves the mapping in a matrix. If a letter is found for the first time the parameters learned by the function are directly saved into the matrix, whereas if some parameters are already stored inside it, the mean between the old and the new ones is computed and then stored.

All the mappings found with this procedure are displayed in Table I.

Table I
DECODED CHACACTERS

Letter	α	λ	Letter	α	λ
a	0.50	2.96	n	3.02	2.98
b	0.51	5.04	o	1.01	4.96
c	1.50	4.96	p	2.50	5.03
e	1.97	4.03	q	2.52	3.98
f	1.52	1.98	r	0.50	4.04
h	2.93	2.01	s	1.00	0.99
i	1.49	4.03	t	2.02	2.96
j	2.99	1.01	u	0.51	1.98
k	1.01	3.00	v	1.48	3.02
l	0.50	1.03	x	1.00	2.00
m	1.00	3.98	y	3.04	4.93

It can be seen that there are values of the same parameters really close one another, but with the other parameter's value one could easily discriminate between the different letters. This is even more evident when the distributions are plotted, as it is better explained in Appendix, Section A.

V. DECODING OF THE MESSAGE

The first observation needed for the description of the decoding algorithm is that each letter is still represented with N samples in the file, giving then the means to split it.

For each of these batches the parameters of the gamma distribution represented are learned in the same fashion as described in Section IV.

The next step needed is to compare the obtained results with the already known mappings, to learn which letter is represented by each batch of the secret file. To perform such a task, each time parameters are computed, the absolute difference between them and the ones for each known letter are compared with two experimental thresholds, respectively 0.3 for α and 0.04 for $1/\lambda$. When both differences respect these thresholds, that character is decoded and saved in a message placeholder. Instead, when no matching letter can be found an asterisk is stored in the placeholder.

Following the known mapping and the asterisk rule this is the message decode obtained:

$$there * *as * no * ice * cream * in * the* \\ free * er * so * he * crie* \tag{A}$$

VI. COMPLETE MAPPING

As said in Section II, it is important that the secret message is an english sentence. This allows to infer from the decoded Message A some missing letters.

The first observation is that not only the letters, but also spaces are present in the sentence. The other observations instead identify unknown letters:

- The second word is clearly a *was* and so its first letter is a *w*
- The first word on the second row is *freezer*, so the 5th letter is z
- The last word is *cried*, so the last letter is d

After these observations it can be stated that the full message would look like this:

$$there * was * no * ice * cream * in * the* \\ freezer * so * he * cried \tag{B}$$

Other three letter-parameters mappings can so be decoded, following what was just stated. Since there are no repetitions of the letters just inferred it is sufficient to apply the *fitdistr* function to the correct set of samples in the secret message and learn the parameters.

After this last operation the table of all known letters (from the dataset) and the inferred ones (*w*, *z*, *d*) is Table II: all the english alphabet is covered but the letter *g* (represented with *unk* as parameters). This would make easy to decode future messages using the same mappings, while also learning the only missing character.

Table II
ALPHABET MAPPING TO PARAMETERS

Letter	α	λ	Letter	α	λ
a	0.50	2.96	n	3.02	2.98
b	0.51	5.04	o	1.01	4.96
c	1.50	4.96	p	2.50	5.03
d	3.00	4.03	q	2.52	3.98
e	1.97	4.03	r	0.50	4.04
f	1.52	1.98	s	1.00	0.99
g	*unk*	*unk*	t	2.02	2.96
h	2.93	2.01	u	0.51	1.98
i	1.49	4.03	v	1.48	3.02
j	2.99	1.01	w	2.49	3.01
k	1.01	3.00	x	1.00	2.00
l	0.50	1.03	y	3.04	4.93
m	1.00	3.98	z	2.50	0.99

VII. Alternative Method

The approach of using maximum likelihood estimation (MLE) to fit parameters is the most standard procedure used when a task like this one is presented, but in the case of a gamma distribution there exists a close-form analytical solution to estimate parameters from a sample, which is therefore faster. This procedure relies on the Method of Moments (MOM) [2].

It is known that mean and variance can be computed using the parameters α and λ in the following way:

$$E[X] = \alpha\lambda \tag{2}$$
$$Var[X] = \alpha\lambda^2 \tag{3}$$

Solving those equations w.r.t. the shape (α) and scale(λ) we can find that:

$$\alpha = E[X]^2/Var[X] \tag{4}$$
$$\lambda = Var[X]/E[X] \tag{5}$$

It is evident that the computation is much lighter than computing parameters with MLE, because just mean and variance are needed. Also, the parameters given should be more precise, but close to the previous estimation, the confidence depending just on the mean and variance estimators and not also on MLE: this can be seen comparing the tables of the known characters mappings, Table I and Table III.

The algorithm of the decryption still remains intact, there is just the need to substitute the parameters estimation using MLE with the one exploiting MOM. The same experimental thresholds worked also with this method: the same sentence, Message A, is obtained.

Table III
Characters decoded with MOM

Letter	α	λ	Letter	α	λ
a	0.50	2.94	n	2.98	3.03
b	0.51	4.99	o	1.00	5.02
c	1.49	4.97	p	2.50	5.02
e	1.98	4.01	q	2.49	4.02
f	1.51	1.99	r	0.50	3.98
h	2.96	1.99	s	1.00	0.99
i	1.49	4.05	t	2.04	2.94
j	2.98	1.01	u	0.51	1.96
k	1.00	3.01	v	1.47	3.05
l	0.50	1.04	x	1.01	2.00
m	1.00	3.98	y	3.02	4.96

VIII. Conclusions

The evaluation part in this task can not be algebraic, as there is only one validation needed and it is performed by manually checking the secret message. To perform such an evaluation it is just to read the obtained decrypted message and check if it makes sense in common English.

The resulting message with inferred letters Message B makes perfect sense in English and so the first three tasks of the work are validated. Following this, also the inferred letters are correct, even if maybe with less precision than others, having at disposal just one sample for each of them.

Eventually it can be stated that encrypting some message with this technique can be efficient and secure if possible attackers can not enter in possession of the mapping or the encrypting procedure: even with few clues about it, the algorithm becomes weak.

Appendix A
Similar distributions

In some of the mappings there are parameters with very similar values: this is the case for example for letters *a*, *b*, *l*, *r*, that have a value α which is pretty much the same (0.50). It can easily be seen, both in Table I and in Figure 2, that the resulting distributions are indeed very different, thanks to the different value of λ: letter *l* has a more peaked distribution (so a lower value of variance) w.r.t. the others.

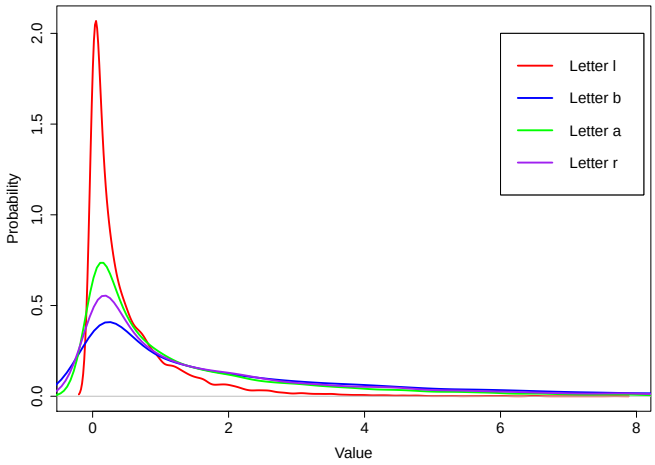

Figure 2. Distributions of letters "a", "b", "l" and "r"

References

[1] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, 4th ed. New York: Springer, 2002, iSBN 0-387-95457-0. [Online]. Available: http://www.stats.ox.ac.uk/pub/MASS4

[2] Wikipedia. Method of moments. Last access: 2019-05-21. [Online]. Available: https://en.wikipedia.org/wiki/Method_of_moments_(statistics)