Article title: Optimization of entropy encoding using random variations

Authors: Andrea Berdondini[1]

Affiliations: University of Bologna[1]

Orcid ids: 0000-0001-6277-3242[1]

Contact e-mail: andrea.berdondini@libero.it

License information: This work has been published open access under Creative Commons Attribution License <http://creativecommons.org/licenses/by/4.0/>, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Conditions, terms of use and publishing policy can be found at <https://www.scienceopen.com/>.

Preprint statement: This article is a preprint and has not been peer-reviewed, under consideration and submitted to ScienceOpen Preprints for open peer review.

DOI: 10.14293/S2199-1006.1.SOR-.PPUKDIG.v1

Preprint first posted online: 18 April 2021

Optimization of entropy encoding using random variations

Andrea Berdondini

ABSTRACT: This article describes an optimization method concerning entropy encoding applicable to a source of independent and identically-distributed random variables. The algorithm can be explained with the following example: let us take a source of i.i.d. random variables X with uniform probability density and cardinality 10. With this source, we generate messages of length 1000 X^{1000} which will be encoded in base 10. We call XG the set, of dimension 10^{1000} , containing all messages that can be generated from the source. According to Shannon's first theorem, if the average entropy of X , calculated on the set XG , is $H(X) \approx 0.9980$, the average length of the encoded messages will be $1000 \cdot H(X) = 998$. Now, we increase the length of the message by one and calculate the average entropy concerning the 10% of the sequences of length 1001 $X10^{1001}$ having less entropy. We call this set, of dimension 10^{1000} , $XG10$. The average entropy of $X10$, calculated on the $XG10$ set, is $H(X10) \approx 0.9964$, consequently, the average length of the encoded messages will be $1001 \cdot H(X10) = 997.4$. Now, we make the difference between the average length of the encoded sequences belonging to the two sets (XG and $XG10$) $998 - 997.4 = 0.6$. Therefore, if we use the $XG10$ set, we reduce the average length of the encoded message by 0.6 values in base ten. Consequently, the average information per symbol becomes $(1001 \cdot 0.9964)/1000 = 0.9974$, which turns out to be less than the average entropy of X $H(X) \approx 0.998$. We can use the $XG10$ set instead of the XG set, because we can create a biunivocal correspondence between all the possible sequences generated by our source, which we know to be 10^{1000} , and ten percent of the sequences with less entropy of the messages having length 1001, in fact $10^{1001} \cdot 0.1 = 10^{1000}$. In this article, we will show that this transformation can be performed by applying random variations on the sequences generated by the source.

Introduction

This article describes an optimization method concerning entropy encoding applicable to a source of independent and identically distributed random variables. The optimization method can be described with the following example: let us take a source of i.i.d. random variables X with uniform probability density and cardinality 10. We use this source to generate messages of length 1000 $X^{1000} = (X_1, X_2, \dots, X_{1000})$, which must be compressed into base 10. We call XG the set, of size 10^{1000} , containing all messages that can be generated by the source. At this point, we formulate the following problem: what is the minimum average length of compressed messages? We try to answer this question with two methods. The first method transmits the 1000 symbols, emitted by the source, to the decoder without any coding. In this way, the messages are transmitted using 1000 symbols in base 10 (the same ones emitted by the source). This method reaches the theoretical limit of compression of the entire message, in fact, having no information on the generated sequence on average, we cannot decode it in less than 1000 symbols in base 10. Instead, in the second method we use entropy encoding. In this case, the average length of the encoded messages is obtained from the average entropy in base ten of X , calculated on the set XG , $H(X) \approx 0.9980$.

In this way, the average length of the encoded message will be $1000 \cdot H(X) = 998$, plus the codewords of the 10 symbols. Since the entropy $H(X) \approx 0.9980$, the codewords will have a length of approximately 1 value in base ten, having 10 symbols we must send 10 values in base ten to the decoder. Therefore, the total message will have an average size of about $998 + 10 = 1008$, higher than the limit of 1000.

In this article, we show experimentally through Monte Carlo simulations, that it is possible to

optimize the entropy coding in order to approach the limit of 1000 base ten values.

In order to obtain this result, the optimization method randomly changes 10 times, through a function $R(X)$, the sequence generated by the source $XR^{1000} = R(X^{1000})$, and then applies the entropy encoding on the modified sequence $XR^{1000} = (XR_1, XR_2, \dots, XR_{1000})$ with less entropy. We call the set made up of the modified sequences with less entropy $XR10$. The average entropy in base ten of XR , calculated on the $XR10$ set, is $H(XR) \approx 0.9964$. Therefore, the encoded message will have an average size equal to $1000 \cdot 0.9964 + 10 + 1 = 1007.4$. We add a base-ten value, because the decoder to obtain the original sequence must know which of the 10 variations has been chosen. By making the difference between the average length of the coded message belonging to the XG set and the average length of the coded message belonging to the $XR10$ set, we obtain a gain equal to $1008 - 1007.4 = 0.6$ and we approach the limit of 1000.

In the chapter in which the experimental results are presented, it is possible to note that with increasing the number of random variations the two limits tend to get closer.

Regarding the theory of this method, two points of view will be presented. The first exploits a combinatorial approach and was developed to mathematically demonstrate the experimental results obtained.

The second point of view is based on the study of random processes and represents the approach that was used to develop this optimization algorithm.

The theory from the combinatorial point of view

A source of i.i.d. random variables X with uniform probability density and cardinality C which generates messages of length N , can create C^N sequences each with its own entropy. We call XG the set consisting of all the C^N sequences that can be generated from the source. Consequently, according to Shannon's first theorem [1], if the average entropy X , calculated on the set XG , is $H(X)$, the average length of the encoded messages can never be greater than $NH(X)$.

This result implies that there is no other set, containing C^N sequences of cardinality C , whose average length of the encoded sequences is less than $NH(X)$. If such a set existed, the limit defined by Shannon's first theorem could be overcome by defining a transform that creates a biunivocal correspondence between the sequences of the two sets.

Now we demonstrate experimentally, with a simple and easily reproducible example, that it is possible to find a set containing C^N sequences of cardinality C that can be encoded with an average length less than $NH(X)$.

Let us take a source of i.i.d. random variables X with uniform probability density and cardinality 10, which generates messages of length 1000 $X^{1000} = (X_1, X_2, \dots, X_{1000})$. We call XG the set, of size 10^{1000} , containing all messages that can be generated by the source. We obtain the average value of entropy in base ten of X , calculated on the XG set, through Monte Carlo simulations. Generating a large number of sequences, we obtain a result very close to the theoretical value. We use base 10 only to make this analysis simpler and easier to understand.

In this way, we obtain the following entropy value $H(X) \approx 0.9980$.

Now we increase the message length by one and calculate the mean entropy concerning the 10% of the sequences of length 1001 $X10^{1001} = (X10_1, X10_2, \dots, X10_{1001})$ having less entropy. We call this set, of dimension 10^{1000} , $XG10$.

The average entropy in base ten of $X10$, calculated on the $XG10$ set, is $H(X10) \approx 0.9964$.

Now we make the difference ΔM between the average length of the encoded messages belonging to the two sets:

$$\Delta M = 1000 \cdot H(X) - 1001 \cdot H(X10) = 1000 \cdot 0.998 - 1001 \cdot 0.9964 = 0.6$$

Therefore, if we use the $XG10$ set to encode the sequence, we reduce the length of the message by 0.6 values in base ten. Consequently, the average information per symbol becomes:

$$\frac{(1001 \cdot 0.9964)}{1000} = 0.9974$$

This value is less than the average entropy of X $H(X) \approx 0.998$.

Furthermore, the average length of the codewords that must be sent to the decoder turns out to be shorter when using the $XG10$ set ($H(X10) < H(X)$ with $X10 \in XG10$ and $X \in XG$). In this way, we obtain an additional gain in terms of message compression.

The $XG10$ set can be used, because we can create a biunivocal correspondence between all the possible sequences generated by our source, which we know to be 10^{1000} , and ten percent of the sequences with less entropy of the messages having length 1001, in fact $10^{1001} \cdot 0.1 = 10^{1000}$. In conclusion, we define a transform that uniquely associates each of the 10^{1000} sequences, generated by the source, to one of the 10^{1000} sequences of length 1001 which represent 10% of the sequences with lower entropy among the 10^{1001} possible combinations.

The method presented in this article applies this transform by performing random variations on the sequence to be encoded. If, for example, we randomly modify 10 times, through a function $R(X)$, the sequence generated by the source $XR^{1000} = R(X^{1000})$. Then, the entropy coding will be performed on the modified sequence $XR^{1000} = (XR_1, XR_2, \dots, XR_{1000})$ with lower entropy. We call the set made up of the modified sequences with less entropy $XR10$. The average entropy in base ten of XR , calculated on the set $XR10$, is $H(XR) \approx H(X10) \approx 0.9964$. Finally, the decoder to go back to the original sequence must know which of the ten random variations has been chosen. By adding this parameter to the sequence, the length of the message increases by one unit passing from 1000 to 1001. So the new sequence of length 1001 is the equivalent of one of 10^{1000} sequences of length 1001 belonging to the $XG10$ set.

The reason for choosing random variations to perform this transform will be discussed in the next section.

The theory from the random processes point of view

Regarding the development of this optimization method, it could simply be said that it represents the natural practical application of the concepts expressed in my article "the information paradox" [2].

This article is based on the following consideration, which represents the fundamental problem of statistics. *"A statistical data does not represent useful information, but becomes useful information only when it is shown that it was not obtained randomly."*

The concept just exposed helps us to understand the logical principle of Occam's razor, in which it is advisable to choose the simplest solution among the available hypotheses. The simplest solution is the one that has fewer parameters, reducing the parameters of a model we reduce the probability of obtaining a high correlation randomly. Consequently, the simplest model is also the model that minimizes the probability of having a random correlation, so it is the model to be preferred.

The idea behind the algorithm presented in this article stems from the intuition to exploit to our advantage the ability of random processes to create information.

Experimental results concerning the entropy encoding of a source of i.i.d. random variables with uniform probability density simulated by the Monte Carlo method

In this section, we simulate a source of i.i.d. random variables with uniform probability density, which generates messages of variable length and alphabet. In this way, we study the optimization method with respect to these two parameters. We also study the optimization with respect to the number of random variations applied.

Regarding the concept of information, we use the definition given by Claude E. Shannon, which defines the entropy $H(X)$ of a discrete random variables X with cardinality C and probability mass function $p(x)$ as:

$$H(X) = - \sum_{i=1}^C p(x_i) \log_b p(x_i) \quad (1)$$

In order to make the results easier to interpret, the entropy will be calculated using the base 10. Given a sequence X^N containing C symbols, the random variation $R(X)$ used in the simulations is:

$$R(X_n) \equiv (X_n + f(n)) \bmod C \quad (2)$$

The function $f(n) \in \{1, 2, 3 \dots \dots, C\}$ is a discrete function with a period greater than or equal to the length of the message to be encoded. The function $f(n)$ was generated using a pseudo-random number generator. So, for example, if we perform 10 random variations the pseudo-random number generator will be initialized with 10 different values (seed). The chosen value must be sent to the decoder.

If we have a source of cardinality 10 with $X_1 = 5$ and $f(1) = 7$ we get:

$$R(X_1) = 2 \equiv (5 + 7) \bmod 10$$

The sequence generated by the source can be obtained by applying the inverse of $R(X)$:

$$X_n \equiv (R(X_n) - f(n)) \bmod C \quad (3)$$

In this case, we have:

$$5 \equiv (2 - 7) \bmod 10$$

the value of the original sequence. In decoding, knowing the seed with which the pseudo-random number generator was initialized, we are able to obtain all the values of the sequence generated by the source.

Now we analyze the experimental results. All the mean entropy values, reported in the tables, were obtained by simulating 100000 sequences. This value was chosen because it represents a good compromise between the accuracy of the results and acceptable simulation times.

Before starting the data analysis it is important to make some considerations regarding the source coding, which will allow us to understand the results obtained. The purpose of source coding is to match the symbols of a source with the codewords of a source code, using the probability of emission of the source symbols. So optimizing the source encoding means finding a minimum length codeword to associate with each symbol. The message in which the symbols are replaced by the codewords cannot be decoded if the decoder does not know the codewords. Therefore, the message generated by the source is uniquely determined by the encoded message plus the codewords of each symbol. As shown, in the example of the introduction, the length of the encoded message plus the codewords is greater than the theoretical compression limit of the entire message.

This limit is intuitive to calculate when the source has a uniform distribution. In the simulations performed, the source generates messages of cardinality 10 that must be encoded in base 10. In this case, the minimum average length of a compressed message $\bar{M}$ is:

$$\bar{M} = N \quad (4)$$

The aim of the optimization method, presented in this article, is to obtain an encoded message plus the codewords whose length converges to the limit (4).

We start by analyzing the optimization method according to the alphabet of the source. To do this, we simulate a source of i.i.d. random variables X with uniform probability density and cardinality C variable between 2 and 10, which generates sequences of constant length $N = 1000$. On the sequences generated by the source, we will apply 10 random variations (2).

The results are presented in table 1. In the first column, we have the number of elements that make up the alphabet of the source. In the second column, we have the average entropy of X in base 10. In the third column, we have the average length of the coded message $1000 \cdot H(X)$. In the fourth column, we have the average entropy $H(XR)$ in base 10, calculated by choosing the lowest

entropy value obtained by applying ten random variations (2) on the sequence generated by the source $XR^{1000} = R(X^{1000})$. In the fifth column, we have the average length of the message encoded using the average entropy $H(XR)$. One is added to the coded sequence because the decoder, to generate the original sequence, must know which variation has been chosen. Having chosen to perform 10 variations, a base-ten value is enough to define this parameter. Finally, in the sixth column, we have the difference ΔM between messages encoded using $H(X)$ and messages encoded with $H(XR)$ (the value of the third column minus the value of the fifth column).

Cardinality of the source C	$H(X)$	$1000 \cdot H(X)$	$H(XR)$	$1000 \cdot H(XR) + 1$	ΔM
2	0.301	300.813	0.300	301.206	-0.393
3	0.477	476.688	0.476	476.851	-0.164
4	0.601	601.405	0.600	601.403	0.002
5	0.698	698.098	0.697	697.963	0.136
6	0.777	777.064	0.776	776.812	0.252
7	0.844	843.792	0.842	843.422	0.371
8	0.902	901.567	0.900	901.109	0.457
9	0.952	952.498	0.951	951.935	0.563
10	0.998	998.045	0.996	997.420	0.625

Table 1: Results related to a source of i.i.d. random variables with uniform probability density and cardinality varying from 2 to 10, which generates messages of length $N = 1000$.

From the analysis of the data shown in the table, we note that this optimization method works when the alphabet of the source is greater than 3. In fact, when the alphabet is less than 4, this method increases the length of the coded sequence. On this result, we can be made the following comments:

1) if the alphabet of the source is composed of only 2 or 3 elements, the length of the encoded message plus the codewords is very close to the theoretical limit (4), so there is not much room for improvement.

2) This is a non-definitive result, because there are infinite sets containing C^{1000} elements. Only a few of these sets have been considered in this article, so this finding needs further study.

Now we analyze the optimization method by varying the message length. To do this, we simulate a source of i.i.d. random variables X with uniform probability density and cardinality 10, which generates sequences of length N varying from 100 to 1000 elements. On the sequences generated by the source, we will apply 10 random variations (2).

Table 2 shows the data of these simulations. In the first column, we have the length N of the simulated sequences, in the other columns we have the same parameters presented in table 1.

Sequence length N	$H(X)$	$NH(X)$	$H(XR)$	$NH(XR) + 1$	ΔM
100	0.980	98.004	0.963	97.333	0.671
200	0.990	198.024	0.982	197.377	0.647
300	0.993	298.034	0.988	297.390	0.643
400	0.995	398.040	0.991	397.398	0.642
500	0.996	498.043	0.993	497.419	0.625
600	0.997	598.039	0.994	597.401	0.638
700	0.997	698.046	0.995	697.414	0.633
800	0.998	798.044	0.996	797.415	0.629
900	0.998	898.040	0.996	897.400	0.640
1000	0.998	998.040	0.996	997.398	0.642

Table 2: Results related to a source of i.i.d. random variables with uniform probability density and cardinality 10, which generates messages of length N varying from 100 to 1000 elements.

Analyzing the data, a correlation is found between ΔM and the difference between the theoretical limit of compression (4) and the length of the sequence encoded using $H(X)$. Since limit (4) is equal to N , this difference is $N - NH(X)$ (the value of the first column minus the value of the third column). Since this correlation is low, this analysis requires further studies.

Now, we analyze the optimization method according to the number of random variations (2). To do this, we simulate a source of i.i.d. random variables X with uniform probability density and cardinality 10, which generates sequences of constant length $N = 1000$. On the sequences generated by the source, we will apply 10, 100, 1000 and 10000 random variations (2).

Table 3 shows the data of these simulations. In the first column, we have the number of random variations applied $nR(X)$, in the other columns we have the same parameters presented in table 1. In the fifth column, R represents the length of the parameter that identifies the random variation chosen. So we will have: $R = 1$ for $nR(x) = 10$, $R = 2$ for $nR(x) = 100$, $R = 3$ for $nR(x) = 1000$ and $R = 4$ for $nR(x) = 10000$.

Number of random variations $nR(X)$	$H(X)$	$1000 \cdot H(X)$	$H(XR)$	$1000 \cdot H(XR) + R$	ΔM
10	0.998	998.040	0.996	997.398	0.642
100	0.998	998.043	0.995	996.952	1.091
1000	0.998	998.042	0.994	996.623	1.419
10000	0.998	998.043	0.992	996.377	1.666

Table 3: Results related to a source of i.i.d. random variables with uniform probability density and cardinality 10, which generates messages of length $N = 1000$.

Analyzing the data in table 3, the following two considerations can be deduced.

1) By increasing the number of random variations, the average length of the coded sequence decreases and, consequently, we have a reduction of the average information per symbol.

2) The gain obtained by increasing the number of random variations by 10 times decreases. Indeed, we have the following increments: $1.091 - 0.642 = 0.449$, $1.419 - 1.091 = 0.329$ and $1.666 - 1.419 = 0.247$. This behaviour is correct because we know that the compression limit (4) cannot be exceeded. As previously mentioned, the value reported in columns 3 and 5 represents the source coding, therefore it represents the length of the message where the symbols have been replaced by the codewords. The decoder to decode the message must know the codewords of each symbol, this information, in this case, occupies about 10 values in base ten ($H(X) \cdot 10$). So by adding 10, to the data in columns 3 and 5, we always get a value above the theoretical limit (4) which, in this case, is 1000 values in base ten.

Unfortunately, by further increasing the number of random variations, the simulation times become too long. Therefore, in order to understand the maximum optimization limit of the entropy coding (also considering the information regarding the codewords) an analytical approach must be used. However, it can be assumed that this limit approaches the theoretical limit of compression, which, in the simulated case, is defined by formula (4).

Conclusion

In this article, it has been experimentally demonstrated that the set of dimension C^N generated by a source of i.i.d. random variables X with uniform probability density and cardinality $C > 3$ which generates messages of length N , is not the set of dimension C^N which minimizes the average length of the encoded message. Indeed, in column 5, of table 1, we find the average length of the messages encoded using a set different from that generated by the source. If these values are divided by 1000 (the length of the message), we obtain the average information per symbol which turns out to be, when the alphabet of the source is greater than 3, less than the average entropy of X $H(X)$.

The new set is constituted by the sequences having the same cardinality of the source and length $N + I$ with $I \in \mathbb{N}$ (in the simulated cases $I = \{1, 2, 3, 4\}$). The greater length generates more combinations C^{N+I} , so we can define a subset that includes sequences with less entropy. The size of the subset is defined so as to have the same number of elements C^N as the set generated by the

source. Therefore, it is possible, through a transform, to create a biunivocal relationship between the elements of the two sets.

From the experimental point of view, the transform was performed by applying random variations on the sequence generated by the source. This choice was not made only to make the algorithm computationally efficient; its use has a much more important meaning, it represents the first application in which the ability of random processes to create information is exploited. In this way, we can optimize the entropy coding, and this allows us to approach the theoretical limit of compression defined, for the simulated cases, by formula (4).

In conclusion, a source of i.i.d. random variables X defines a set, whose dimension C^N is determined by the alphabet C and the length of the message N . The sets of dimension C^N are infinite. Consequently, in order to define an absolute limit about the source coding, we have to find the set that minimizes the average length of the encoded message among all the sets of dimension C^N containing sequences having the same cardinality as the source.

References

- [1] Claude E. Shannon, "A Mathematical Theory of Communication", Bell system Technical Journal, vol 27, lug e ott 1948.
- [2] Berdondini Andrea, "The Information Paradox", (July 8, 2019). Available at SSRN: <https://ssrn.com/abstract=3416559>.

E-mail address: andrea.berdondini@libero.it