

Universal Coding of Function Spaces as a Model for Signal Compression. *

Boris Ryabko[†] and Jaakko Astola[‡]

[†] Siberian State University of Telecommunication and Computer Science, Russia

[‡] Technical University of Tampere, Finland

Abstract

We address the problem of signal compression, basing on the mathematical model, in which a set of all possible signals is considered as a function space with a metric ρ . The main attention is focused on the minimization of the size of compressed representation, when function characteristics are not known precisely.

Keywords. *data compression, universal coding, epsilon-entropy, Kolmogorov entropy.*

1 Introduction.

Nowadays there exist several mathematical approaches to signal processing and compression, see [3, 5]. One of them, which is based on Kolmogorov's notation of ε -entropy, is quite popular and has some real applications to data compression, see for a review, for ex., [1, 3]. When that approach is applied, any signal is considered as a function f from some function space F with a metric ρ . For example, the set F can contain all differentiable functions f , defined on $[0, 1]$ and such that $|f(x)| < 1$, $|f'(x)| < \theta$ while $\rho(f, g) = \sup_x |f(x) - g(x)|$ and θ is a parameter characterizing "smoothness" of functions. It is known that if someone wants to encode such functions with accuracy ε by binary words, the shortest codeword length is equal to $\theta/\varepsilon + O(1)$, when ε goes to 0 [6]. It is important to note, that, on the one hand, the codeword length essentially depends on the parameter θ , which, informally speaking, characterizes the smoothness of the functions (or signals). On the other hand, there are many applications where the smoothness parameter can be upper bounded only and such a bound can be very top-heavy.

*The first author was supported by INTAS grant no. 00-738 and Russian Foundation for Basic Research under Grant no. 03-01-00495.

Moreover, it is typical for many functional spaces, which are used as models for images, speech and other real signals. It poses a problem of universal coding, i.e. an efficient encoding of elements of function spaces where certain parameters of the spaces are not known precisely. In this paper we suggest the universal code, which can be applied to encoding of functions with unknown smoothness asymptotically as efficiently as in case of known parameters.

Definitions and Preliminaries. One of the efficient tools of the signal representation theory is the *epsilon-entropy*, which is defined as follows. Let U be a subset of a metric space, ρ be the metric. A set U_ε called an ε -net for the set U , if for any $x \in U$ there exists $\hat{x} \in U_\varepsilon$ such that $\rho(x, \hat{x}) \leq \varepsilon$. A set, which for any $\varepsilon > 0$ has got a finite ε -net, is called totally bounded, see, for ex., [1, 3, 6, 9].

Let U be a totally bounded set and U_ε be some ε -net with minimal number of elements. Then, the ε -entropy of the set U is defined as follows: $H_\varepsilon(U) = \log |U_\varepsilon|$. (Here and below $\log x \equiv \log_2 x$.) Obviously, the ε -entropy of the set U is of the great interest for the signal representation theory, because it is the minimal length of a codeword, if someone wants to encode functions (or signals) from U by binary words in such a way that the distance between the original element and decoded one is less than or equal to ε .

There are several examples, where ε -entropy is estimated quite precisely. The first example is the set A_θ which contains 2π -periodic functions, which are defined on $(-\infty, \infty)$, analytic in the band $|Im z| \leq 1/\theta$, $z = x + iy$, and the value of functions is bounded by a constant C (i.e. $|f| \leq C$.) Let, as before, $\rho(f, g) = \sup_x |f(x) - g(x)|$. It is known [6] that

$$H_\varepsilon(A_\theta) = 2\theta(\log(1/\varepsilon))^2 / \log e + O(\log(1/\varepsilon) \log \log(1/\varepsilon)). \quad (1)$$

Another example is as follows. Let θ and t be positive integers. We denote by $L_t(\theta)$ the set of real functions f defined on $[0, t]$, $t > 0$, which satisfy $|f(x)| \leq C$ and $|f(x) - f(y)| \leq \theta|x - y|$ for $f \in L_t(\theta)$, $x, y \in [0, t]$ and let $\rho(f, g) = \sup_{x \in [0, t]} |f(x) - g(x)|$. Then for any nonnegative θ and C

$$H_\varepsilon(L_t(\theta)) = t\theta/\varepsilon + O(\log(1/\varepsilon)).$$

The statement of the problem and summary of the contribution. We can see from the examples that the minimal codeword length is proportional to the parameter θ , where, informally speaking, θ characterizes "smoothness" of functions. (Naturally, the "smoother" is a function, the shorter is the codeword length.) This situation is typical for signal representation. As a rule, there exists a parameter, which simultaneously determines the smoothness and the codeword length, see, for example, [1, 3, 6, 9]. On the other hand, in practical applications such a parameter cannot be estimated quite precisely beforehand. Moreover, as a rule, only rough upper bounds of the parameter are known and the code length is proportional to this rough bound, and, hence, even optimal codes are not efficient, because such a representation becomes very redundant for smooth functions.

One approach to the problem was suggested in [8], see also its description in [7]. We extend this approach and obtain a universal code $\Upsilon(f)$, which, loosely speaking,

gives the asymptotically minimal codeword length in a case, where the "smoothness" of an encoded function is not known beforehand. More exactly, we consider a model in which an encoded function f belongs to a function space $\{F_\theta\}$, where θ is not known precisely, but it is known that θ belongs to the interval Q and for each pair of parameters θ_1, θ_2 , $F_{\theta_1} \subset F_{\theta_2}$ if $\theta_1 \leq \theta_2$. Loosely speaking, the code word length of the suggested code is given by formula

$$\text{length}(\Upsilon(f)) = H_\varepsilon(F_\theta) + \log(H_\varepsilon(F_\theta))(1 + o(1)),$$

for each $\theta \in Q$ and $f \in F_\theta$, when ε goes to 0. The value $\log(H_\varepsilon(F_\theta))(1 + o(1))$ can be considered as a payment for a unknownness of the exact value of θ (or smoothness of the function f). So, we can see that this payment is asymptotically negligible, if we compare it with the ε - entropy $H_\varepsilon(F_\theta)$. Informally speaking, the universal code quite precisely estimates the "smoothness" of the function and encodes it according to this estimation.

This code is applied to encoding of several function spaces, which are of some interest for signal processing.

2 The main results

We now give some formal definitions. Let F be a nonempty subset of a metric space M and B be the set of all finite words from the alphabet $\{0, 1\}$. A mapping φ from the set F into B is called a coding of F and a mapping ψ from B into F is called a decoding. If for a coding φ there exists a decoding ψ such that for all f from F the inequality $\rho(f, \psi(\varphi(f))) \leq \varepsilon$ holds, we say that the precision of the coding φ is not less than ε . (Indeed, the inequality means that the distance between the original function f and its presentation after decoding $\psi(\varphi(f))$ is not larger than ε .)

Let $\varepsilon > 0$ and let $\{F_\theta, \theta \in \Theta\}$ be a family of subsets of a metric space such that Θ is an interval $[0, \alpha]$, $\alpha \leq \infty$, and $F_{\theta_1} \subset F_{\theta_2}$, if $\theta_1 < \theta_2$. Our immediate goal is to build a universal code for this family, i.e. such a code $\Upsilon_{\Theta, \varepsilon}$, whose codeword length of any $\theta \in \Theta$ and any function $f \in F_\theta$ is close to the ε - entropy of F_θ .

Theorem . Let $\varepsilon > 0$ and $\{F_\theta, \theta \in \Theta\}$ be such a family of sets that the following properties hold:

- i) the index set Θ is an interval $[\alpha, \beta]$, $0 \leq \alpha < \beta \leq \infty$,
- ii) if $\theta_1 < \theta_2$, $\theta_1, \theta_2 \in \Theta$, then $F_{\theta_1} \subset F_{\theta_2}$,
- iii) $H_\varepsilon(F_\theta) < \infty$ for each $\theta \in \Theta$,
- iv) For any subset of indexes $\bar{\Theta} \subset \Theta$ the following inequality holds $H_\varepsilon(\bigcup_{\theta \in \bar{\Theta}} F_\theta) - \sup_{\theta \in \bar{\Theta}} H_\varepsilon(F_\theta) < C^*(\varepsilon)$, where $C^*(\varepsilon)$ does not depend on $\bar{\Theta}$ and ε .

Then, there exists a code $\Upsilon_{\Theta, \varepsilon}$ such that for any subset $F_\theta, \theta \in \Theta$ and a function $f \in F_\theta$

$$|\Upsilon_{\Theta, \varepsilon}(f)| < H_\varepsilon(F_\theta) + \log(H_\varepsilon(F_\theta)) + C^*(\varepsilon) + O(1 + \log \log(H_\varepsilon(F_\theta))),$$

where ε goes to 0. (Here and below $|x|$ is the length of the word x .)

Comment. The requirements i)-iii) is a formal model of the case when there exists a parameter characterized the function smoothness. The requirement iv) is caused

by two reasons. First, it means, that a small change of the index θ cannot drastically change the value of the ε - entropy. (It is supposed that $C^*(\varepsilon)$ is quite small.) Second, there are many cases, where the ε - entropy is estimated with $C^*(\varepsilon)$ precision. (For example, in the case of the function space A_θ , $C^*(\varepsilon) = \text{const} \log(1/\varepsilon) \log \log(1/\varepsilon)$, see (1)). Naturally, this function $C^*(\varepsilon)$ is to be a part of any upper bound of the universal code.

It is easy to see that all requirements i)-iv) are true for all above mentioned examples.

The proof is based on a direct description of the code $\Upsilon_{\Theta, \varepsilon}$. Let us first define

$$\bar{F}_i = \bigcup_{\theta \in \{\theta: H_\varepsilon(F_\theta) \leq i\}} F_\theta \quad (2)$$

for any integer $i \in [0, H_\varepsilon(\bigcup_{\theta \in \Theta} F_\theta)]$. From iv) we obtain

$$H_\varepsilon(\bar{F}_i) < i + C^*(\varepsilon) \quad (3)$$

Now we can describe how a function f from $\bigcup_{\theta \in \Theta} F_\theta$ is encoded. First, we define a set of indexes of subsets, which contains f as follows: $\hat{\Theta}_f = \{\theta : f \in F_\theta\}$. Let, by definition, j_f be such an integer that f belongs to \bar{F}_{j_f} and f does not belong to \bar{F}_{j_f-1} . (Taking into account the definition (2), we can see that such j_f exists, because f belongs to at least one set F_θ and, according to iii), $H_\varepsilon(F_\theta)$ is finite.)

The codeword of any function f is defined by a concatenation of two following subwords:

$$\Upsilon_{\Theta, \varepsilon}(f) = \text{code}(j_f) \text{code}(f \in \bar{F}_{j_f}), \quad (4)$$

where $\text{code}(j_f)$ is a binary code of the integer j_f . This code for integers will be described below, whereas here we describe the second subword. The part $\text{code}(f \in \bar{F}_{j_f})$ is a binary word of the length $\lceil H_\varepsilon(F_{\theta_i}) \rceil$, which encodes such a function g from the minimal ε - net of \bar{F}_{j_f} that $\rho(f, g) \leq \varepsilon$. (It is possible, because, by definition of the ε - entropy, the number of elements in the minimal ε - net is $2^{H_\varepsilon(F_{\theta_i})}$). Then, from (2) and (3) we obtain

$$|\text{code}(f \in F_{\theta_i})| < H_\varepsilon(F_{\theta_i}) + C^*(\varepsilon) + 1. \quad (5)$$

Let us now describe the first subword $\text{code}(j_f)$ in (4). There are codes for integers $0, 1, \dots$, which codeword length is, loosely speaking, minimal. One of such codes, (λ) was described by V. Levenshtein and later by P. Elias, see, for ex., [7]. The length of the code λ is given by the following formula: $|\lambda(i)| = \log(i+1) + \log \log(i+2)(1 + o(1))$, where i goes to infinity. If we use such a code for encoding of j_f in (4) and take into account the last inequality and (5), we obtain the statement of the theorem. The description of the code λ is rather complicated (see [4, 7]), that is why we describe in Appendix the other code k , whose description is simpler, but asymptotical behavior is the same.

So, if we use the code k for encoding integers, from (4) we have got the following description of the suggested code.

$$\Upsilon_{\Theta, \varepsilon}(f) = k(j_f) \text{code}(f \in F_{j_f}).$$

3 Examples of Universal Coding

If we apply the theorem to the above considered example of the function spaces $L_t(\theta)$ we obtain that the code length of the universal code $\Upsilon_{\Theta, \varepsilon}(f)$ for each f is less than $t\theta/\varepsilon + O(\log(1/\varepsilon))$, if $f \in L_t(\theta)$. If we compare this equality and the ε -entropy of $L_t(\theta) = t\theta/\varepsilon + O(1)$, we can see that the "payment" for the code universality ($O(\log(1/\varepsilon))$) is asymptotically negligible, when ε goes to 0. So, the suggested universal code $\Upsilon_{\Theta, \varepsilon}$ has asymptotically the same codeword length as a code built for the case of exactly known characteristics of the encoded function. The same situation is true for the above described function space A_θ and many function spaces from [1, 3, 6, 9].

Universal code for periodical functions. Here we consider the realization of the universal code for the space of function A_θ in more details, because this example is close to many practically used methods of signal representation and compression. Let it be known that a function f belongs to A_θ and let the exact value of the parameter θ be unknown, but it is known that $\theta \in [0, \Theta_{max}]$.

Let us first describe basic properties of function space A_θ , given in [6]. Each function f from A_θ can be presented as

$$f(x) = \sum_{k=-\infty}^{+\infty} c_k e^{-ikx}, \quad (6)$$

where c_k are the Fourier coefficients of the f . It is known that $c_k \leq C e^{-|k|/\theta}$, where, as before, C is such a constant that $|f| \leq C$ for any $f \in A_\theta$. The asymptotically optimal code, whose codeword length is asymptotically equal to ε -entropy (1), can be described as follows. First, we find a minimal $n(\theta)$ such that $|f(x) - \sum_{|k| \leq n(\theta)} c_k e^{-ikx}| < \varepsilon/2$. Then, we approximate any coefficient $c_k = \alpha_k + i\beta_k$ with the precision $\varepsilon/2(2n+1)$ by $c_k^* = \alpha_k^* + i\beta_k^*$, which are defined by the following equations:

$$\alpha_k^* = m_k^1(\varepsilon/(2\sqrt{2}(2n+1))), \beta_k^* = m_k^2(\varepsilon/(2\sqrt{2}(2n+1))),$$

where, by definition, $m_k^1 = \lfloor 2\sqrt{2}(2n+1)\alpha_k/\varepsilon \rfloor$, $m_k^2 = \lfloor 2\sqrt{2}(2n+1)\beta_k/\varepsilon \rfloor$. It is proven in [6] that $|f(x) - \sum_{|k| \leq n(\theta)} c_k^* e^{-ikx}| < \varepsilon$ and $|m_k^i| \leq N_k$, $i = 1, 2$, where $N_k = 2\sqrt{2}C(2n+1)e^{-|k|/\theta}/\varepsilon$. So, any function f from A_θ can be presented by the sequence of coefficients (m_k^1, m_k^2) , $k = -n(\theta), -(n(\theta) - 1), \dots, n(\theta) - 1, n(\theta)$. Having taken into account the inequality $|m_k^i| \leq N_k$, we can see, that the function f can be encoded by the word of the length $\sum_{|k| \leq n(\theta)} \lceil \log N_k \rceil$ bits and the direct calculation shows that this sum equals the ε -entropy (1), see [6] for details.

Now we can consider the case of universal coding. So, according to the description of the universal code Υ and estimation of the ε -entropy (1), we define the values $i_{max} = \lceil 2\Theta_{max}(\log(1/\varepsilon))^2/\log e \rceil$. Let θ_i be the solution of equation $2\theta_i(\log(1/\varepsilon))^2/\log e = i$, $i = 1, \dots, i_{max}$. Then, the universal code for any f is built as follows. First we calculate $n_{\Theta_{max}}$ of the Fourier coefficients in (6) and find the smallest i such that $|\alpha_k| < e^{|k|/\theta_i}$, $|\beta_k| < e^{|k|/\theta_i}$ for all k , $|k| < n_{\Theta_{max}}$. Now we define this i as j_f , $A_{\theta_{j_f}}$ as F_{j_f} and build the universal code for f , using general description (4). From the theorem and (1) we can see that the codeword length of this code is asymptotically

equal to $2\theta(\log(1/\varepsilon))^2/\log e + 2\log(1/\varepsilon) + O(\log(1/\varepsilon)\log\log(1/\varepsilon) + 1)$. So, we can see that, in fact, the universal code and the "common" code have asymptotically the same codeword lengths.

Universal code and JPEG scheme. It is interesting to compare the above described universal code with real data compression methods based on transform coding, say, with JPEG scheme, which briefly can be described as the combination of discrete cosine transform, scalar quantization and entropy coding [1]. On the one hand, the both codes are very similar. Indeed, in both settings an encoded signal is presented as a sum of the Fourier coefficients. Besides, the coefficients, whose values are less than a certain crucial value, are equated to zero and, hence, their values are not encoded. On the other hand, there is an important distinction. Namely, in the JPEG scheme the length of a codeword for each coefficient depends on its value and is encoded by variable-length code, whereas in the above described code one general characteristic, which corresponds to smoothness of the encoded function, is estimated and, then, all coefficients are encoded correspondingly to this estimation by a code with the fixed codeword length. It is worth noting, that the quantization of the smoothness estimation is determined by the ε -entropy, whereas in the JPEG scheme the quantization of parameters is based on an intuition and experience.

So, this brief analysis shows that the suggested universal coding of function spaces can be useful as a model for signal compression, especially in a case, when function characteristics are not known precisely.

4 Appendix.

The description of the code k .

Let us first explain why "common" binary representation of integers cannot be used. The point is that such a representation is not decipherable. For instance, the word 11001 can be deciphered either as 1 1001 = 1 9 or as 1100 1 = 12 1. The following simple code φ , which is defined by the formula $\varphi(m) = 1^n 0$, is decipherable. (Here 1^n means a sequence from n 1's.) For ex., $\varphi(4) = 11110$. It is obvious how to decode. If someone wants to decode a word $\varphi(n)$, he/she can read and calculate the 1's till the first 0. The number of 1's is equal to the encoded integer n . So, the code φ can be deciphered, but its codeword length is too large. Indeed, the length of the common binary representation of the integer n is around $\log n$ bits, whereas the length of $\varphi(n)$ is much larger: $|\varphi(n)| = n + 1$. The suggested code k is decipherable and has a short codeword length. In order to describe it we define two following functions: $Bin(n)$ is the common representation of the integer n and $bin(n)$ is $Bin(n)$, if we drop its first digit. For example, $Bin(5) = 101$, $bin(5) = 01$. The code $k(n)$ is defined as follows:

$$k(n) = 0^{bin(|bin(n+1)|)} 1 bin(|bin(n+1)|) bin(n+1)$$

(or $k(n) = \varphi(bin(|bin(n+1)|) bin(|bin(n+1)|) bin(n+1))$). For example, let us find $k(4)$. First we calculate $bin(5) = 01$, $|bin(5)| = 2$, $bin(|bin(5)|) = bin(2) = 0$, $|bin(|bin(5)|)| = |0| = 1$ and obtain from the definition

$$k(4) = 0^{bin(|bin(5)|)} 1 bin(|bin(5)|) bin(5) = 0^1 1 0 0 1 = 0 1 0 0 1.$$

Analogously, $k(0) = 1, k(20) = 001000101$. It is easy to decipher n from $k(n)$. We explain it through an example. Suppose, we have a word 001000101. We can see that the number of 0's till the first 1 is 2. So, the $|bin(|bin(n+1)|)| = 2$, where n is a number, which should be found. The next two digits are 00. So, it means that $bin(|bin(n+1)|) = 00$ and $Bin(|bin(n+1)|) = 100$. Hence, the length of the word $bin(n+1)$ equals 4 and the last four digits (0101) are encoded $bin(n+1)$. It means that the binary notation of $n+1$ ($Bin(n+1)$) is 10101, $n+1 = 21$ and, finally, $n = 20$.

References

- [1] A. Cohen, W. Dahmen, I. Daubechies, and R. DeVore, Tree approximation and optimal encoding, Institut fr Geometrie und Praktische Mathematik, Bericht Nr. 174, September 1999.
- [2] Cover T.M., Thomas J.A. Elements of Information Theory. Wiley, 1991.
- [3] D.L. Donoho, M. Vetterli, R.A. DeVore, I. Daubechies, Data compression and harmonic analysis, IEEE Transactions on Information Theory, v. 44, no. 6, 1998, pp. 2435-2476.
- [4] Elias P. Universal codeword sets and representation of integers. IEEE Transactions on Information Theory, v. 21, n.2, 1975, pp.194-203.
- [5] Gray, R.M.; Neuhoff, D.L., Quantization. IEEE Transactions on Information Theory, v.44, n.6 , 1998, pp. 2325 -2383.
- [6] Kolmogorov A. N., Tihomirov V.M. ε - entropy and ε - capacity of Sets in Metric Spases. American Mathematical Society Translations, vol. 17, n.2, 1961.
- [7] Krichevsky R. Universal compression and retrival. Kluwer, 1994.
- [8] Ryabko B.Ya. Universal coding of compacta. Soviet Math. Dokl., Vol.21, 1980, n.3, pp.948-951.
- [9] Vitushkin A.G. Theory of transmission and processing of information. Pergamon Press, 1961.