ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1999
Seria: MATEMATYKA-FIZYKA z. 84 Nr kol. 1411

Andrzej CHYDZIŃSKI, Michał MOMOT

ON APPLICATION OF FINITE MARKOV CHAINS TO ENCODING OF AUDIO DATA

Summary. This paper presents convenient method for encoding digital audio data using basic properties of discrete Markov chains. The main idea is to find the "easy calculable" prognosis function and encoding the set of data as the differences from prognosis.

O ZASTOSOWANIU SKOŃCZONYCH ŁAŃCUCHÓW MARKOWA DO KOMPRESJI DANYCH AUDIO

Streszczenie. Artykuł prezentuje metodę kompresji cyfrowych danych audio opartą na algorytmie wykorzystującym własności łańcuchów Markowa.

1. Introduction

Let X_k for $k = 1, 2, \ldots$ be the finite homogeneous Markov chain with the state space $\{1, \ldots, N\}$. We make no assumption on the initial distribution (it may be stationary). The transition probabilities are:

$$p_{ij} = \Pr\{X_k = j | X_{k-1} = i\} \quad \text{for} \quad k = 2, 3, \ldots$$

and create the transition matrix:

$$\mathbf{T} = \begin{pmatrix} p_{11} & p_{21} & p_{31} & \cdots & p_{N1} \\ p_{12} & p_{22} & p_{32} & \cdots & p_{N2} \\ p_{13} & p_{23} & p_{33} & \cdots & p_{N3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{1N} & p_{2N} & p_{3N} & \cdots & p_{NN} \end{pmatrix}.$$

After [1] we use the *conditional entropy* of X_k:

$$H_i = H(G_k | X_{k-1} = i) = -\sum_{j=1,\ldots,N} p_{ij} \log p_{ij},$$

where G_k is probability event which shows the state where X_k is in. In the case when X_k is stationary, the entropy of the whole chain is given by:

$$H = \mathrm{E}H(G_k | X_{k-1} = i) = -\sum_{i=1,\ldots,N} p_i \sum_{j=1,\ldots,N} p_{ij} \log p_{ij},$$

where $p_i = \Pr\{X_1 = i\}$. Our case of interest is to encode data using binary numbers, so the base of logarithm in all formulas will be equal 2.

Suppose that $N = 2^m$. If the method of writing the information of states of X_n uses "plain" binary code, for encoding the data at n points we need nm binary digits. As shown in [1], for $\epsilon > 0$ fixed there exist 2^{nH} binary sequences, for which the probability of appearance will be equal to $1 - \epsilon$. Such 2^{nH} sequences may be encoded using nH binary digits. Since $H \leq$

$\log N = m$, this gives us the m/H profit in request for number of digits with rather great probability. The general theorem on encoding of Markov chains gives the upper bound for efficiency of this method in term of entropy.

The method for constructing of economical code and calculating its efficiency explicitly is to make the binary trees. The most popular and convenient method for constructing the binary tree was given by Huffman in [2]. This algorithm is also described in standard handbooks for data compression. The detailed description of it with implementation may be found in [4]. In later part of our paper we assume that all binary trees are constructed using Huffman's method.

For the sequence of realisations of independent, identically distributed random variables this gives the most efficient and easy "decodable" code.

2. Base algorithm

Let us have the realisation x_k for $k = 1, \ldots, n$ of Markov chain X_k. The base algorithm is constructed as follows: the first observation is written unchanged, the all following are considered in terms of translation of the previous ones. Thus we have N conditional probability distributions p_{ij} for $i = 1, \ldots, N$. Even if these are unknown, we may use the sample probabilities:

$$\hat{p}_{ij} = \frac{\#\{x_k = j, x_{k-1} = i; k = 2, \ldots, n\}}{\#\{x_{k-1} = i; k = 2, \ldots, n\}}$$

for $i = 1, \ldots, N$, where the denominator is positive. Otherwise the state i may be omitted. For all substantial i we construct the binary tree, which encodes the transitions from the state i to each of N. The process of decoding the data consists of browsing through the appropriate tree for current state $x_k = i$, finding the given binary code and writing the value of $x_{k+1} = j$, where j is the state corresponding to such code.

The procedure described above is time efficient, but requires some space for storing all N (or a few less then N) binary trees. Each of them may be of length up to N. The way to simplify this method is to consider the properties of stationary Markow chain in some special case. Let us write the state $X_k = j$ as $X_{k-1} + r = i + r$, so r will be the translation term of j in relation of i. Suppose that the transition probabilities p_{ij} depend only on $r = j - i$. Thus we may write $q_r = p_{i,i+r}$. Note that such X_k does not satisfy our assumption, since its state space is infinite. The conditional entropy of X_k simplifies to:

$$H_i = -\sum_{r \in D} q_r \log q_r,$$

where D is the set of all possible translations and H_i does not depend on i. The best method for encoding information on such Markov chain is to construct the common binary tree for probabilities q_r. It requires that D should be finite. Obviously the procedure described above does not work in our case, when we have N states. However it gives the idea to simplificate the algorithm. From sequence of realisation of X_k we compute the sample probability distribution of translations:

$$\hat{q}_r = \frac{\#\{x_k - x_{k-1} = r; k = 2, \ldots, n\}}{N-1}, \quad r = -N+1, \ldots, N-1.$$

The common binary tree for translations may then be created using probabilities defined above. When it is done the total length of data encoded using this method may be easily evaluated. If l_r is the number of binary digits needed for describing the translation by r, the $n-1$ observations occupy:

$$(N-1) \sum_{r=-N+1}^{N-1} \hat{q}_r l_r.$$

Adding a few digits for writing the state of x_1 and some space for storing the binary tree with its description we get the total length of encoded data. For given set of data this lets us to compute the estimated compression ratio.

3. Generalization

Now we assume that X_k is generalized Markov chain, where X_{k+1} depends not only on the previous value, but also on a fixed number d of previous ones. Based on this model the theoreticaly best method for encoding information on this chain is to find N^d conditonal probability distributions and to construct the binary tree for each of them. In practical applications this method fails for small d and moderately large N because the calculating the N^{d+1} probabilities and storing N^d binary trees requires too much resources. Considered the difficulties described above we use another, simpler method. We build the function $\tilde{X}_{k+1} = f(X_k, X_{k-1}, \ldots, X_{k-d+1})$ which have to satisfy following property: the entropy of probability distribution of difference $X_{k+1} - \tilde{X}_{k+1}$ is minimal possible. This function of d variables is called prognosis.

For $d = 2$ the simple method of prognosing in the case, when we are concerned with audio data is to use linear prognosis $\tilde{X}_{k+1} = 2X_k - X_{k-1}$, which corresponds to drawing the straight line on the plot of x_k vs. k through the points $(k-1, x_{k-1})$, (k, x_k) and taking its vertical coordinate for $k+1$. This prognosis seems good, since the original audio signal is continuous.

In general the entropy of distribution of $X_{k+1} - \tilde{X}_{k+1}$ is quite difficult to analyse and will be replaced by its variance, the standard measure of dispersion for random variables, since it may be analysed using algebraic methods. Theorem 1. lets us create linear unbiased prognosis with minimal variance (UMVP) for the sequence of random variables consisting of deterministic trend and stochastic noise. In some special cases it corresponds to generalized Markov chain model and may be useful in application to encoding of digital audio data.

Consider the sequence $X_k = \sum_{i=1}^{s} \alpha_i f_i(k) + Y_k$. Let $[f_1, \ldots, f_s]$ denotes the linear space generated by $f_1, \ldots, f_s$ and $f^{t(p)}$ denotes the function with translation in argument, so $f^{t(q)}(k) = f(k+q)$.

Theorem 1. *Let us have:*

$$X_k = \sum_{i=1}^{s} \alpha_i f_i(k) + Y_k$$

where Y_k is non-degenerated, stationary sequence of random variables with $EY_k = 0$, $Cov(Y_k, Y_{k+j}) = \gamma_j$, $f_1, \ldots, f_s$ are linear independent, and let $f^{t(q)} \in [f_1, \ldots, f_s]$ for every integer q. Then for given positive integer p we have:

$$\underline{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{d-1} \end{pmatrix} = \Gamma^{-1} F^T (F \Gamma^{-1} F^T)^{-1} \underline{f},$$

where:

$$\underline{f} = \begin{pmatrix} f_1(p) \\ f_2(p) \\ \vdots \\ f_s(p) \end{pmatrix}, \qquad F = \begin{pmatrix} f_1(0) & \ldots & f_1(-d+1) \\ f_2(0) & \ldots & f_2(-d+1) \\ \vdots & & \vdots \\ f_s(0) & \ldots & f_s(-d+1) \end{pmatrix},$$

$$\Gamma = \begin{pmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{d-1} \\ \gamma_1 & \gamma_0 & \cdots & \gamma_{d-2} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{d-1} & \gamma_{d-2} & \cdots & \gamma_0 \end{pmatrix},$$

gives UMVP $\hat{X}_{k+p} = \sum_{j=0}^{d-1} \beta_j X_{k-j}$, in the sense that

$$\begin{cases} E\hat{X}_{k+p} = EX_{k+p} \\ Var\hat{X}_{k+p} \to min. \end{cases} \tag{1}$$

Proof. Since $EY_k = 0$ and $f_1, \ldots, f_s$ are linear independent, $E\hat{X}_{k+p} = EX_{k+p}$ is equivalent to $\sum_{j=0}^{d-1} \beta_j f_i(k-j) = f_i(k+p)$ for $k \geq d$, $i = 1, \ldots, s$.

Assuming that $f^{t(q)} \in [f_1, \ldots, f_s]$ for every integer q, the values of f_i in equalities above may be taken at any d consecutive integers, say $-d+1, \ldots, 0$. For $\hat{X}_{k+p}$ unbiased, $Var\hat{X}_{k+p} = \sum_{i,j=0}^{d-1} \beta_i \beta_j \gamma_{|i-j|} = \underline{\beta}^T \Gamma \underline{\beta}$. Thus (1) is equivalent to:

$$\begin{cases} F\underline{\beta} = \underline{f} \\ \underline{\beta}^T \Gamma \underline{\beta} \to min. \end{cases} \tag{2}$$

Since Γ is positive definite as covariance matrix, our goal is to find local minimum of quadratic form subject to linear equality constraints. The problem may be solved by using Lagrange multiplayers. Let us put $\underline{\lambda} = (\lambda_1, \ldots, \lambda_s)^T$. Then differentiating by $\underline{\beta}$, we have:

$$\begin{cases} F\underline{\beta} = \underline{f} \\ 2\Gamma\underline{\beta} - F^T\underline{\lambda} = 0. \end{cases} \tag{3}$$

Writing:

$$A = \begin{pmatrix} 2\Gamma & | & -F^T \\ - & & - \\ F & | & \mathbf{0} \end{pmatrix}, \quad \underline{\tilde{\beta}} = \begin{pmatrix} \underline{\beta} \\ - \\ \underline{\lambda} \end{pmatrix}, \quad \underline{\tilde{f}} = \begin{pmatrix} \underline{0} \\ - \\ \underline{f} \end{pmatrix},$$

we may write (3) in short form $A\underline{\tilde{\beta}} = \underline{\tilde{f}}$. Let us search for A^{-1} in form:

$$A^{-1} = \begin{pmatrix} B & | & -C^T \\ - & & - \\ C & | & D \end{pmatrix}.$$

From:

$$\begin{pmatrix} 2\Gamma & | & -F^T \\ - & & - \\ F & | & \mathbf{0} \end{pmatrix} \begin{pmatrix} B & | & -C^T \\ - & & - \\ C & | & D \end{pmatrix} = \begin{pmatrix} \mathbf{I} & | & \mathbf{0} \\ - & & - \\ \mathbf{0} & | & \mathbf{I} \end{pmatrix}$$

follows:

$$\begin{cases} 2\Gamma B - F^T C = \mathbf{I} \\ -2\Gamma C^T - F^T D = \mathbf{0} \\ FB = \mathbf{0} \\ -FC^T = -\mathbf{I} \end{cases}$$

This system of matrix equations may be solved for C as follows:

$$\begin{aligned} 2B &= \Gamma^{-1}(\mathbf{I} + F^T C), \\ \mathbf{0} &= F\Gamma^{-1} + (F\Gamma^{-1}F^T)C, \\ C &= -(F\Gamma^{-1}F^T)^{-1}F\Gamma^{-1}, \\ -C^T &= \Gamma^{-1}F^T(F\Gamma^{-1}F^T)^{-1}. \end{aligned}$$

Calculating $\underline{\tilde{\beta}} = A^{-1}\underline{\tilde{f}}$ leads us to the result.■

The simple heuristic linear prognosis described at the beginning of this section may now be obtained from Theorem 1., for $d = 1$ by assuming $f_1(k) \equiv 1, f_2(k) = k$, and $\gamma_0 = 1, \gamma_1 = 0$. The linear prognosis for polynomial trends of higher orders in some special cases can be evaluated in explicit form using the properties of differences of X_k. At first note, that for the linear prognosis $\hat{X}_{k+1} = 2X_k - X_{k-1}$ we have:

$$X_{k+1} - \hat{X}_{k+1} = (X_{k+1} - X_k) - (X_k - X_{k-1}) \tag{4}$$

and this correction term may be expressed as second difference of sequence X_k, in the sense of following definition.

Definition 1. $X_k^{(s)}$ *is called the s-th order difference of sequence* X_k *if*

$$\begin{aligned} X_k^{(0)} &= X_k, \\ X_k^{(s)} &= X_k^{(s-1)} - X_{k-1}^{(s-1)} \qquad \textit{for} \quad i \geq 1. \end{aligned}$$

Remark 2. $X_k^{(s)} = \sum_{i=0}^{s} (-1)^i \binom{s}{i} X_{k-i}$.■

These properties of differences of X_k give us the simple method for constructing linear prognosis by generalizing (4).

Definition 2. *$\hat{X}_{k+1}^{(s)}$ is called the s-th order linear prognosis if*

$$X_{k+1} - \hat{X}_{k+1}^{(s)} = X_{k+1}^{(s)}. \tag{5}$$

Remark 3. *For $f_i(k) = k^{i-1}$, $i = 1, \ldots, s$, $\hat{X}_{k+1}^{(s)}$ is UMVP.*

Proof. The s-th order difference of the polynomial of order less or equal to $s-1$ is constant zero, so the difference $X_{k+1} - \hat{X}_{k+1}^{(s)}$ has zero mean. Since $\{1, k, \ldots, k^{s-1}\}$ are linear independent, such unbiased prognosis is unique and must be optimal.∎

In the later case we may write $\beta_j = (-1)^j \binom{s}{j+1}$ for $j = 0, \ldots, s-1$.

Note, that prognosis discribed above for integer valued data always give integer valued prognosis. This is very useful in using them for encoding digital data sequences.

4. Numerical examples

In later part of our paper we consider the results of application of this method in encoding digital audio data. Such data are created by sampling the analog signal at regular times and storing the values scaled to appropriate digital measurement. In practical applications the sampling rate is equal to 44.1 kHz or 48 kHz for high quality data. The encoding of digital acoustic data in telecommunication (i.e. speech) requires much lower rate, typically 8 kHz and the data may be compressed using some lossy techniques. Since our goal is to store the data without losing the quality, we assume the sampling rate to be rather high. The number of states in the set of digital audio data is the power of 2, usualy 2^8 or 2^{16}.

Figure 1 presents the typical audio waveform (16 bit, 44.1 kHz), which includes 139 observations. It is the part of longer set of data (27450 observations), the next calculations are based on.

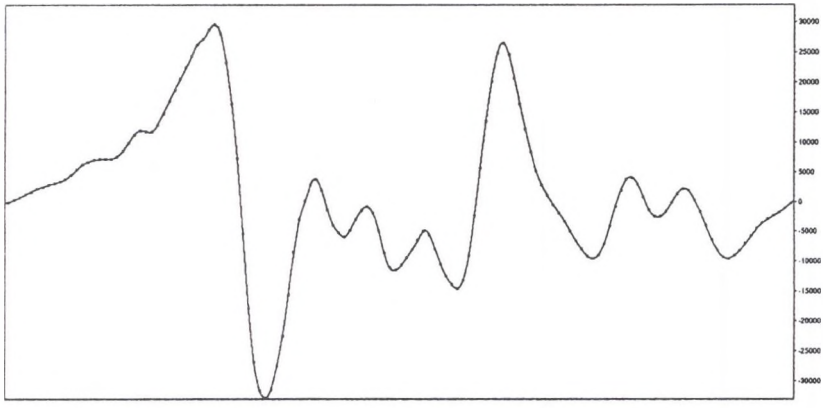

Fig. 1

Below several results of numerical computation of entropy are presented. They concern sample probability distributions of x_k and differences for various prognosis based on the data presented above. The sample entropy of this sequence is equal to 5.85189. Figure 2 presents the sample probabilities for the states of x_k.

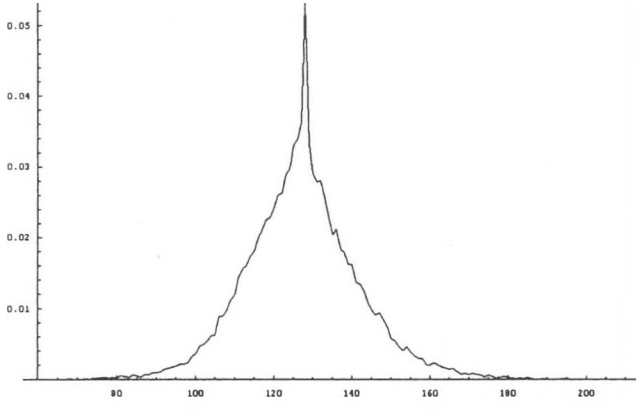

Fig. 2

Figure 3 presents the sample probabilities for the differences $x_{k+1} - \hat{x}^{(2)}_{k+1}$. The sample entropy for this distribution is equal to 4.75102.

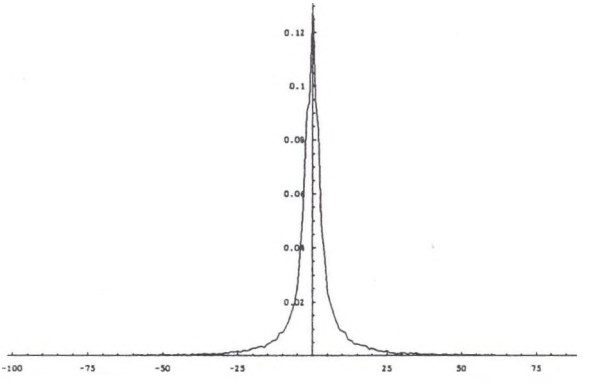

Fig. 3

Table 1 presents the similar results for various degrees of polynomials and numbers of observations the prognosis were based on.

Table 1

Sample entropies for various prognosis coefficients

Base functions	Number of obs.	Prognosis coefficients	Sample entropy of differences
$\{1, k\}$	2	(2,-1)	4.75102
$\{1, k\}$	3	$(\frac{4}{3}, \frac{2}{3}, -\frac{1}{3})$	5.07537
$\{1, k\}$	4	$(1, \frac{1}{2}, 0, -\frac{1}{2})$	5.2712
$\{1, k, k^2\}$	3	$(1, 3, -3)$	5.20793
$\{1, k, k^2\}$	4	$(\frac{9}{4}, -\frac{3}{4}, -\frac{5}{4}, \frac{3}{4})$	5.31193

These results of numerical computation show that in general the prognosis based on the model with polynomials with added independent random variables is a good way to reduce the entropy of sample probability distribution for typical audio data. However, the increasing of number of base functions, as well as the number of observations concerned do not improve quality of this method. The best results were obtained using the simple linear prognosis based on two observations.

Now we present the results of computations for three sets of audio data using only simple linear prognosis. Obviously at some points the prognosed value was out of possible range $(1, \ldots, N)$, thus it was truncated. In each case the binary tree for sample distribution of differences was created. Then the size of set of encoded data was calculated and compared with those generated by popular compression utilities. The compression ratio is given as the percentage of original data size.

Table 2

Human voice, 44.1 kHz, 16 bit, mono, 7 526 268 bytes

Tool	Compr. data size (bytes)	Compr. ratio
arj 2.20	6 646 733	88.314%
pkzip 2.04g	6 644 330	88.282%
rar 2.0	6 659 710	88.486%
rar 2.0 (multimedia compr.)	4 733 234	62.889%
our method	4 960 539	65.909%

As we can see, assuming the specific structure of digital audio data we obtain the radically better ratios, when using standard tools. The extension is required for stereo data, which may be treated as the realisation of two-dimensional stochastic chain $(X_{1,k}, X_{2,k})$. Fortunately both its components are corralated, so we may concern $(X_{1,k} - X_{2,k}, X_{1,k})$ ($(X_{1,k} - X_{2,k}$ is called "separacy channel") instead of original channels, which lets us get additional profit, as shown in Tables 3, 4.

Table 3

Instrumental music, 48 kHz, 16 bit, stereo, 15 963 676 bytes

Tool	Compr. data size (bytes)	Compr. ratio
arj 2.20	14 345 202	89.862%
pkzip 2.04g	14 360 284	89.956%
rar 2.0	14 341 546	89.839%
rar 2.0 (multimedia compr.)	9 267 589	58.054%
our method	8 521 336	53.379%

5. Final remarks

The possible way to improve the algorithm based on Theorem 1. is to consider the other sets of base functions, i.e. trigonometric ones. The quasi-periodical nature of audio data suggests this method as promising. In this case the Fourier transformation may be useful to detect the base frequencies. Another way is to concentrate on the covariance function γ_i and search for the convenient method of estimating it for given set of audio data. Such problems will be considered in future.

Table 4

Instrumental music, 44.1 kHz, 16 bit, stereo, 16 575 712 bytes

Tool	Compr. data size (bytes)	Compr. ratio
arj 2.20	14 965 025	90.282%
pkzip 2.04g	14 979 212	90.368%
rar 2.0	14 963 643	90.274%
rar 2.0 (multimedia compr.)	9 962 447	60.103%
our method	9 443 341	56.970%

References

1. A. A. Borovkov, *Kurs teorii veroyatnostei (A course in probability theory),* Nauka, Moscow 1972.

2. D. A. Huffman, *A method for the construction of minimum-redundancy codes,* Proceedings of the IRE **40** (1952), 1098-1101.

3. J. H. Karl, M. Nelson, *An Introduction to Digital Signal Processing,* Academic Press, London 1989.

4. M. Nelson, *The Data Compression Book,* M&T Publishing, San Mateo 1992.

Andrzej Chydziński
Michał Momot
Institute of Mathematics
Silesian Technical University
Kaszubska 23
44-100 Gliwice

Streszczenie

Artykuł prezentuje metodę kompresji cyfrowych danych audio opartą na algorytmie wykorzystującym własności łańcuchów Markowa. Główna idea polega na wyznaczeniu łatwej do obliczenia funkcji prognozującej, a następnie kodowaniu zbioru danych jako odchyleń od prognozy. Funkcja prognozująca jest dobierana tak, aby rozkład odchyleń od prognozy miał możliwie małą entropię. Zamieszczone wykresy przedstawiają próbkowe rozkłady odchyleń dla przykładowego zbioru danych audio. Uzyskane wyniki porównano z osiągnięciami popularnych programów kompresujących.