

ロスレス音声符号化のためのSVRによるFIRシステム同定

峰尾 太陽^{1,a)} 庄野 逸^{1,b)}

概要: ロスレス音声符号化において、再帰的 (recursive) Golomb-Rice 符号はコーデックとして採用実績のある符号である。予測符号化の観点では、本符号の符号長が短くなるように FIR (Finite impulse response) フィルタの係数を定めるのが望ましい。本研究では、符号長は Hinge 損失のもとで最小化できることを指摘し、損失最小化のために SVR (Support vector regression) による FIR フィルタの係数設定手法を提案する。学習は勾配法により行うことができ、得られた係数による残差は古典的な Levinson-Durbin アルゴリズムと比べ、Hinge 損失が小さくなることを確かめた。提案手法を組み込んだコーデックを実装し、実音楽データに対する性能比較を行った。評価の結果、FLAC よりも優れた圧縮率を示し、デコード速度は WavPack よりも高速であることが示された。

FIR System Identification using SVR for Lossless Audio Compression

Abstract: In lossless audio compression, the recursive Golomb-Rice code is practical since it is used in existing codecs. Furthermore, in predictive coding, it is desirable to set the coefficients of the FIR (finite impulse response) filter so that the code length is small. In this study, we highlight that the code length can be minimized under the Hinge loss and propose a method for setting FIR filter coefficients by SVR (support vector regression). Furthermore, the learning can be done by the gradient method, and the residuals with the obtained coefficients have a minor Hinge loss than those with the classical Levinson-Durbin algorithm. Finally, we implemented a codec incorporating the proposed method and compared its performance on music data. We show that the proposed codec compresses music data better than FLAC and decodes faster than WavPack.

1. はじめに

近年のデジタルメディアコンテンツの高級化に伴い、音声のデータ容量は増加の一途を辿っている [1]。音声データを劣化なく圧縮するロスレス音声コーデックは、音声の半永久的保存、高品質配信、編集用途において重要な地位を占める [2]。ロスレス音声コーデックの性能向上は、今後の容量拡大に応えるためにも重要な課題である。

ロスレス音声コーデックは一般に図 1 に示す構成からなる [3]。まず PCM 符号化された音声に対して予測モデルを適用し、予測とのずれを残差信号として出力する。次いで残差信号に対してエントロピー符号を適用し、ビット列に出力する。ここで、予測が正確であれば残差信号は 0 に集中し、より小さく圧縮できる。

Shorten [4] の開発を契機として、現在に至るまで図 1 の

構成に従って多くのコーデックが開発された。とくに、FLAC [5]、MPEG4-ALS [6] は線形予測符号化 (Linear Predictive Coding, LPC) を主な予測モデルとして使用している。符号は Golomb-Rice 符号 [7] が一般的に使われている。WavPack [8]、TTA [9] は本符号を改良した再帰的 (recursive) Golomb-Rice (RGR) 符号 [10] を採用している。

LPC の係数決定は FIR (Finite impulse response) フィルタのシステム同定に相当し、一般的に Levinson-Durbin (LD) アルゴリズム [11]、[12] が使用される。また、Golomb-Rice 符号は残差が Laplace 分布にしたがう際に最も短い符号量を達成できるため、その仮定による補助関数法を用いた係

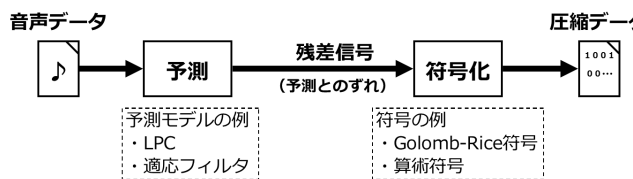


図 1 一般的なロスレス音声コーデックの構成

¹ 電気通信大学 大学院情報理工学専攻 情報学専攻
〒182-8585 東京都調布市調布ケ丘 1-5-1

a) m2040009@edu.cc.uec.ac.jp

b) shouno@uec.ac.jp

数決定手法 [13] が提案され、圧縮率の改善が示された。しかし、本手法は処理負荷が大きい [13]。このため、符号長の仮定を前提にしつつ、高速に FIR フィルタの係数を決定する手法提案が重要である。

本研究では、RGR 符号の符号長を最小化する基準による FIR フィルタの同定手法を提案する。適切な仮定のもとで RGR 符号の平均符号長は Hinge 損失と一致することを示し、その最小化のために SVR(Support vector regression)[14] を用いる。提案手法は主変数（フィルタ係数）を勾配法によって学習できるため、LD アルゴリズムで得られた係数を改善する用途に適している。実音楽データに対する実験の結果、LD アルゴリズムよりも Hinge 損失が小さい係数が得られることを確かめた。

提案手法の有効性を示すため、本手法を組み込んだコーデック SRLA(Svr-fIR Lossless Audio) を実装した。実音楽データを使用した性能比較の結果、FLAC よりも優れた圧縮率を、WavPack よりも高速なデコード速度を示した。

2. 背景

2.1 Levinson–Durbin アルゴリズム

離散時間系列 x_t が以下の AR(p) 過程

$$x_t = \sum_{i=1}^p h_i x_{t-i} + \nu_t, \quad (1)$$

に従う場合を考える。ここで、 $h_i \in \mathbb{R}$ ($i = 1, \dots, p$) は AR 係数、 $\nu_t \in \mathbb{R}$ は残差項である。表記の簡略化のため、係数ベクトル $\mathbf{h}_p := [h_1, \dots, h_p]^T$ (T : 行列転置)、入力信号ベクトル $\mathbf{x}_t := [x_t, x_{t-1}, \dots, x_{t-p+1}]^T$ を用いて、

$$\nu_t = x_t - \mathbf{x}_{t-1}^T \mathbf{h}_p, \quad (2)$$

と表す。残差が平均 0、分散 σ^2 の Gauss 分布に従って発生すると仮定する。独立同一分布から N 個の標本を得たときの対数尤度 $\log L(\mathbf{h}_p)$ は、

$$\log L(\mathbf{h}_p) = -\frac{1}{2\sigma^2} \sum_{t=1}^N (x_t - \mathbf{x}_{t-1}^T \mathbf{h}_p)^2 + \text{const.}, \quad (3)$$

となる。式 (3) を \mathbf{h}_p で偏微分し、その結果を $\mathbf{0}$ とおいて解くと、方程式

$$\mathbf{R}\mathbf{h}_p = \mathbf{r}, \quad (4)$$

が得られる。ここで、 $r_{ij} := \sum_{t=1}^N x_{t-i} x_{t-j}$ 、 $\mathbf{r} := [r_{01}, \dots, r_{0p}]^T$ 、 $(\mathbf{R})_{ij} = r_{ij}$ ($i, j = 1, \dots, p$) とおいている。

信号系列を弱定常と仮定すると \mathbf{R} は Toeplitz 行列になる。この行列の構造に着目することで、式 (4) を時間計算量 $\mathcal{O}(p^2)$ で解く Levinson–Durbin(LD) アルゴリズム (アルゴリズム 1) が導かれる。アルゴリズム 1 において、 \mathcal{E}_k は k 次の残差分散と呼ばれ、 k 次の係数 \mathbf{h}_k^* を用いたときの残差の分散と一致する。

アルゴリズム 1 Levinson–Durbin (LD) アルゴリズム

入力: 係数次数 p , 自己相関系列 r_k ($k = 0, \dots, p$)

出力: 係数ベクトル \mathbf{h}_p^*

```

1:  $(\mathbf{h}_1^*)_1 \leftarrow -r_1/r_0$ 
2:  $\mathcal{E}_1 \leftarrow r_0 + r_1(\mathbf{h}_1^*)_1$ 
3: for  $k = 2, \dots, p$  do
4:    $\lambda_k \leftarrow -\frac{\sum_{i=1}^k r_{k+1-i}(\mathbf{h}_{k-1}^*)_i}{\mathcal{E}_{k-1}}$ 
5:    $\mathcal{E}_k \leftarrow (1 - \lambda_k^2)\mathcal{E}_{k-1}$ 
6:    $\mathbf{u}_k \leftarrow [1, (\mathbf{h}_{k-1}^*)_1, \dots, (\mathbf{h}_{k-1}^*)_{k-1}, 0]^T$ 
7:    $\mathbf{v}_k \leftarrow [0, (\mathbf{h}_{k-1}^*)_{k-1}, \dots, (\mathbf{h}_{k-1}^*)_1, 1]^T$ 
8:    $\mathbf{h}_k^* \leftarrow \mathbf{u}_k + \lambda_k \mathbf{v}_k$ 
9: end for
10: return  $\mathbf{h}_p^*$ 

```

$$\mathcal{E}_k = \mathbb{E} \left[\left(x_t - \mathbf{x}_{t-1}^T \mathbf{h}_k^* \right)^2 \right]. \quad (5)$$

2.2 再帰的 Golomb–Rice 符号とその平均符号長

正整数 x , 符号化パラメータ k に対する Golomb–Rice 符号 [7] の手続きは、

(1) $\lfloor x/2^k \rfloor$ 回 0 を出力し、続けて 1 を出力

(2) $x \bmod 2^k$ を k (bit) の 2 進数で出力

となる。任意の整数 x に対しても、たとえば以下の写像

$$f(x) = \begin{cases} 2x & (x \geq 0) \\ -2x + 1 & (x < 0) \end{cases}, \quad (6)$$

により正整数と全単射を構成でき、任意の整数を符号化できる。再帰的 Golomb–Rice (RGR) 符号 [10] は、

(1) $x < 2^{k_1}$ ならば x に対して k_1 を符号化パラメータとして Golomb–Rice 符号を適用して終了

(2) 0 を 1 回出力し、 $x - 2^{k_1}$ に対して k_2 を符号化パラメータとして Golomb–Rice 符号を適用

で与えられる。RGR 符号の長さ $L_{\text{RGR}}(x, k_1, k_2)$ は

$$L_{\text{RGR}}(x, k_1, k_2) = \begin{cases} k_1 + 1 & x < 2^{k_1} \\ 2 + k_2 + \left\lfloor \frac{x - 2^{k_1}}{2^{k_2}} \right\rfloor & x \geq 2^{k_1} \end{cases}, \quad (7)$$

となり、これを k_1, k_2 を変えてグラフにすると図 2 になる。

x が $\rho \in (0, 1]$ をパラメータとする幾何分布 $P(x) = \rho(1 - \rho)^x$ にしたかうとき、RGR 符号の平均符号長は

$$\mathbb{E}[L_{\text{RGR}}(x, k_1, k_2)] = (1 + k_1) \left\{ 1 - (1 - \rho)^{2^{k_1}} \right\} + \left\{ 1 + k_2 + \frac{1}{1 - (1 - \rho)^{2^{k_2}}} \right\} (1 - \rho)^{2^{k_1}}, \quad (8)$$

で与えられる [15]。式 (8) より $\mathbb{E}[L_{\text{RGR}}(x, k_1, k_2)]$ は k_1, k_2 に関して凸であり解析的に最適化可能である。この観察から得られたパラメータ設定方策をアルゴリズム 2 に示す。

2.3 SVR

Hinge 損失の元で定式化される回帰モデルに SVR[14],

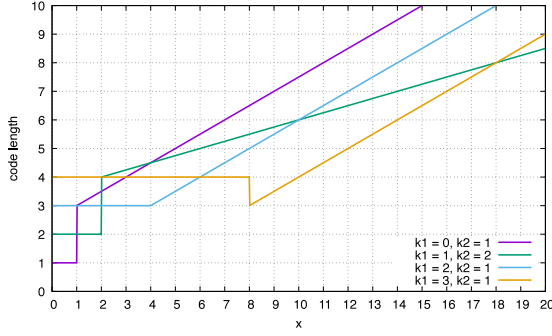


図 2 k_1, k_2 を変えたときの再帰的 Golomb-Rice 符号長. 簡単のため $x \geq 2^{k_1}$ における床関数 $\lfloor \cdot \rfloor$ を除去している.

アルゴリズム 2 RGR 符号のパラメータ設定方策 [15]

- 1: ρ を最尤推定: $\rho \leftarrow \frac{1}{1+E[x]}$
- 2: k_2 を計算: $k_2 \leftarrow \max \left\{ 0, \left\lceil \log_2 \left[\frac{\log(s^*)}{\log(1-\rho)} \right] \right\rceil \right\}$, $s^* \approx 0.512762951443$
- 3: k_1 を計算: $k_1 \leftarrow k_2 + 1$

[16] がある. SVR では, 以下のラグランジュ関数

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^N x_i \alpha_i - \varepsilon \sum_{i=1}^N |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j), \quad (9)$$

の最大化を議論する. ここで, $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_N]^T$ は双対変数ベクトル, ε はマージン, $K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i)^T \boldsymbol{\phi}(\mathbf{x}_j)$ はカーネル関数 ($\boldsymbol{\phi}$ は特徴写像) である.

SVR では, 係数 \mathbf{h}_p と双対変数は以下の関係式 (双対表現) で対応付けられる.

$$\mathbf{h}_p = \sum_{i=1}^N \alpha_i \boldsymbol{\phi}(\mathbf{x}_i), \quad (10)$$

また, 表記の簡単化のため, 以下の計画行列 (design matrix) \mathbf{X} を定義する.

$$\mathbf{X} := [\boldsymbol{\phi}(\mathbf{x}_1), \boldsymbol{\phi}(\mathbf{x}_2), \dots, \boldsymbol{\phi}(\mathbf{x}_N)]^T, \quad (11)$$

これにより, $(\mathbf{G})_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ なる Gram 行列は $\mathbf{G} = \mathbf{X}\mathbf{X}^T$ と書ける. さらに双対表現は $\mathbf{h}_p = \mathbf{X}^T \boldsymbol{\alpha}$, 回帰による予測を並べたベクトルは以下で表せる.

$$\mathbf{X}\mathbf{h}_p = \mathbf{X}\mathbf{X}^T \boldsymbol{\alpha} = \mathbf{G}\boldsymbol{\alpha}. \quad (12)$$

式 (9) の双対問題を解くために, SVM の文脈では SMO (Sequential minimal optimization) [17] が一般に使われる. 一方, $-L(\boldsymbol{\alpha})$ の最小化とすれば $\boldsymbol{\alpha}$ の L1 ノルム正則化を含む問題と解釈でき, 座標降下法 [18], [19], 近接勾配法に基づく Iterative shrinkage thresholding algorithm [18], [20] を適用できる.

3. 提案手法

3.1 SVR による FIR システム同定

アルゴリズム 2 を使用してパラメータ設定を行う場合,

RGR 符号長は (定数加算と床関数を除き) $\varepsilon = 2^{k_1}$ とした Hinge 損失に一致し, SVR を適用できる. しかし, 2.3 節で挙げた既存手法は双対変数 $\boldsymbol{\alpha}$ の解を求める手法であり適用が困難である. なぜなら, 音声信号処理の文脈では入力サンプル数 N は 10^3 以上に設定され Gram 行列 \mathbf{G} が大きくなるためである. 一方, ロスレス音声コーデックでは係数次数 p は N に比べ一般に小さい. そこで, 双対変数 $\boldsymbol{\alpha}$ について解く代わりに係数 \mathbf{h}_p の解を直接求める手法を提案する.

入力ベクトルを $\mathbf{x}_{\text{in}} := [x_1, x_2, \dots, x_N]^T$ として, (9) 式を $\boldsymbol{\alpha}$ で偏微分 (劣微分) すると,

$$\frac{\partial}{\partial \boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = \boldsymbol{\nu} - \varepsilon \text{sign}(\boldsymbol{\alpha}), \quad (13)$$

となる. ここで, $\boldsymbol{\nu} := \mathbf{x}_{\text{in}} - \mathbf{G}\boldsymbol{\alpha} = \mathbf{x}_{\text{in}} - \mathbf{X}\mathbf{h}_p$ は残差ベクトル, sign は符号関数で

$$\text{sign}(x) := \begin{cases} 1 & x > 0 \\ [-1, 1] & x = 0 \\ -1 & x < 0 \end{cases}, \quad (14)$$

と定義される ($x = 0$ で多値をとる). 勾配を要素ごとに見ると, $|\nu_i| \geq \varepsilon$ のとき x_i はサポートベクトルとなり α_i と残差 ν_i の符号が一致し, $|\nu_i| < \varepsilon$ のとき x_i はサポートベクトルにならず $\alpha_i = 0$ となる [16]. これより,

$$\frac{\partial}{\partial \alpha_i} L(\boldsymbol{\alpha}) = \begin{cases} \nu_i - \varepsilon \text{sign}(\nu_i) & |\nu_i| \geq \varepsilon \\ [\nu_i - \varepsilon, \nu_i + \varepsilon] & |\nu_i| < \varepsilon \end{cases} \quad (15)$$

$$\ni S_\varepsilon(\nu_i), \quad (16)$$

となる. ここで, $S_\varepsilon(\cdot)$ は以下で定義される軟しきい値作用素 (soft-thresholding operator) である.

$$S_\varepsilon(x) = \begin{cases} x - \varepsilon & x \geq \varepsilon \\ 0 & |x| < \varepsilon \\ x + \varepsilon & x \leq -\varepsilon \end{cases}, \quad (17)$$

従って勾配は

$$\frac{\partial}{\partial \boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = S_\varepsilon(\boldsymbol{\nu}), \quad (18)$$

としてよい. 次に, この勾配を主変数の勾配に変換する. 連鎖律より,

$$\frac{\partial}{\partial \mathbf{h}_p} L(\boldsymbol{\alpha}) = \left(\frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{h}_p} \right)^T \frac{\partial}{\partial \boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = \mathbf{X}^T \mathbf{G}^{-1} S_\varepsilon(\boldsymbol{\nu}), \quad (19)$$

ここで, $\frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{h}_p}$ は双対表現により求めている. 現在の問題設定では N が大きいため \mathbf{G}^{-1} の消去を考える. 勾配に対して式変形を加えると,

$$\frac{\partial}{\partial \mathbf{h}_p} L(\boldsymbol{\alpha}) = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X} \mathbf{X}^T \mathbf{G}^{-1} S_\varepsilon(\boldsymbol{\nu}) \quad (20)$$

$$= \mathbf{R}^{-1} \mathbf{X}^T S_\varepsilon(\boldsymbol{\nu}), \quad (21)$$

アルゴリズム 3 SVR による FIR システム同定

入力: 入力信号ベクトル \mathbf{x}_{in} , 係数次数 p , マージン幅 ε , Ridge 正則化パラメータ μ

出力: 解係数

- 1: 係数 \mathbf{h}_p を LD アルゴリズムで初期化, $\mathbf{R}_\mu \leftarrow \mathbf{X}^T \mathbf{X} + \mu \mathbf{I}$
- 2: 残差ベクトルの計算: $\boldsymbol{\nu} \leftarrow \mathbf{x}_{in} - \mathbf{X} \mathbf{h}_p$
- 3: 係数更新: $\mathbf{h}_p \leftarrow \mathbf{h}_p + \mathbf{R}_\mu^{-1} \mathbf{X}^T S_\varepsilon(\boldsymbol{\nu})$
- 4: 収束判定を行い, 収束していなければ 2. に戻る
- 5: return \mathbf{h}_p

となる. ここで, $\mathbf{R} = \mathbf{X}^T \mathbf{X}$ は自己相関行列である. さらに, \mathbf{G} に Ridge(L2) 正則化を導入できる. $\mathbf{G}_\mu := \mathbf{G} + \mu \mathbf{I}$ とおくと, 双対表現から,

$$\mathbf{X}^T \mathbf{G}_\mu \boldsymbol{\alpha} = (\mathbf{R} + \mu \mathbf{I}) \mathbf{h}_p =: \mathbf{R}_\mu \mathbf{h}_p, \quad (22)$$

だから, \mathbf{R} の Ridge 正則化と同等になる. また勾配は,

$$\frac{\partial}{\partial \mathbf{h}_p} L(\boldsymbol{\alpha}) = \mathbf{X}^T \mathbf{G}_\mu^{-1} S_\varepsilon(\boldsymbol{\nu}) = \mathbf{R}_\mu^{-1} \mathbf{X}^T S_\varepsilon(\boldsymbol{\nu}), \quad (23)$$

となるため, \mathbf{R} を \mathbf{R}_μ に置き換えればよい.

(23) 式を勾配とした係数の学習則をアルゴリズム 3 に示す. 3 行目の係数更新では, 方程式 $\mathbf{R}_\mu \mathbf{v} = \mathbf{X}^T S_\varepsilon(\boldsymbol{\nu})$ の解 \mathbf{v} を勾配として使える. \mathbf{R}_μ は一度 Cholesky 分解しておけば繰り返しの度に分解結果を使いまわせるため高速である. なお, $\varepsilon = 0$ のとき本手法は共分散法に一致する.

3.2 符号長に基づく次数選択

残差分散 \mathcal{E}_p は次数 p の増加と共に単調減少する一方, 係数領域サイズが増大するため適切な次数選択が重要である. ここでは符号長に基づく次数選択手法を提案する. 既存研究 [21] はエントロピーを基準にしているが, 本手法は RGR 符号の平均符号長に基づく点が異なる.

残差の分布は Laplace 分布で近似できる [4] ことから, 残差 ν_t は Laplace 分布 $\text{Laplace}(0, b)$ にしたがうと仮定する. 本分布の分散は $2b^2$ だから, 残差分散 (式 (5)) とは $b = \sqrt{\mathcal{E}_p/2}$ の関係がある. また,

$$\mathbb{E}[|\nu_t|] = \int_{-\infty}^{\infty} |x| \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right) dx = b, \quad (24)$$

から, 絶対値平均を

$$\mathbb{E}[|\nu_t|] = \sqrt{\mathcal{E}_p/2}, \quad (25)$$

で推定できる. この結果と平均符号長 (式 (8)) の関係を用いることで, p 次の係数を使用したときの符号長 L_p を

$$L_p = \mathbb{E}[L_{\text{RGR}}(|\nu_t|, k_1^*, k_2^*)] N + pB, \quad (26)$$

で推定できる. ここで, k_1^*, k_2^* はアルゴリズム 2 で求めたパラメータ, N はサンプル数, B は係数の量子化ビット数である. これより最適な次数 p^* は

$$p^* = \underset{p}{\operatorname{argmin}} L_p, \quad (27)$$

で求められる. ただし, 符号化では残差を非負整数に変換するため振幅を合わせる必要がある. たとえば式 (6) を用いる場合は, 絶対値平均を 2 倍することで近似する.

3.3 コーデック構成

提案手法を組み込んだロスレス音声コーデック SRLA のエンコーダとデコーダの構成をそれぞれ図 3,4 に示す.

3.3.1 チャンネル間相関除去

各チャンネルを統計的独立に近づけるため, チャンネル間の相関を除去する.

$$M = \frac{L+R}{2}, \quad S = L - R, \quad (28)$$

ここで, L, R, M, S はそれぞれ左, 右, Mid, Side チャンネルの音声を表す. Mid/Side, 左/Side, 右/Side チャンネルから左/右チャンネルを復元できるため, この組み合わせの中で平均符号長 (式 (8)) を評価して最も短い組み合わせを選択する. デコーダでは

$$L = M + \frac{S}{2}, \quad R = M - \frac{S}{2}, \quad (29)$$

によって左/右チャンネルを復元する.

3.3.2 プリエンファシス/デエンファシス

プリエンファシスは \mathbf{R} の条件数を改善する [22] 効果があり, 次の 1 次の FIR フィルタとして表される.

$$y_n = x_n - \eta x_{n-1}, \quad (30)$$

ここで, x_n, y_n はそれぞれ入力と出力の信号系列, $\eta \approx 1$ はフィルタ係数である. デエンファシスは式 (30) を x_n について解いた式を用いる.

3.3.3 LPC 予測/LPC 合成

アルゴリズム 3 によって LPC 係数を計算し, 予測を行う. 最適な係数次数は式 (27) にしたがって設定する.

3.3.4 マルチプレクサ/デマルチプレクサ

マルチプレクサは符号化した残差と係数などの付加情報を 1 本のビット列に結合して出力する. デマルチプレクサはビット列から残差と付加情報に分解する.

4. 計算機実験

4.1 LD アルゴリズムとの比較

提案手法を Python3 で実装し, LD アルゴリズムとの比較を RWC 音楽データセット [23] 音源での平均 Hinge 損失の計測を通じて行った. 実験では, 音声フレームサイズは 4096 とし, 係数次数とマージン幅を変えている. 実験に使用した音源は, 「交響曲 第 94 番 ト長調 「驚愕」 第 1 楽章」, 「Wasting Time」, 「Jive (Piano Solo)」, 「永遠のレプリカ」, 「ちょうちょう」の左チャンネルの 10-20 秒間である.

次数とマージン幅を変えた際の結果を図 5 に示す. 図 5

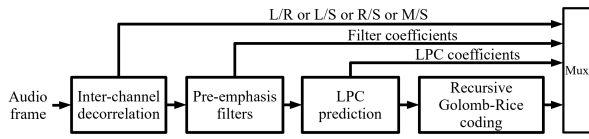


図 3 SRLA のエンコーダ構成

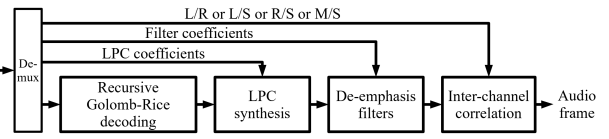


図 4 SRLA のデコーダ構成

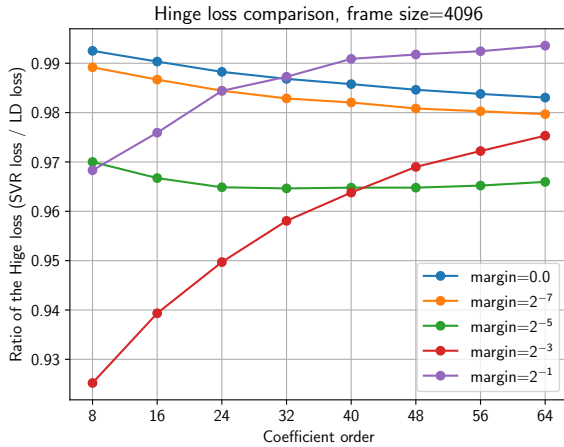


図 5 Hinge 損失の比較結果

表 1 コーデック性能比較に用いたデータセットの概要

カテゴリ	識別子 (カテゴリの接尾辞)	トラック数	合計サイズ
クラシック	RWC-MDB-C-2001(M01-M04)	34	2157.0 (MB)
音楽ジャンル	RWC-MDB-G-2001(M01-M04)	48	1862.3 (MB)
ジャズ	RWC-MDB-J-2001(M01-M04)	50	2260.0 (MB)
ポピュラー音楽	RWC-MDB-P-2001(M01-M04)	64	2587.9 (MB)
著作権切れ音楽	RWC-MDB-R-2001(M01)	15	331.9 (MB)
合計	-	211	9199.0 (MB)

では、LD アルゴリズムとの平均 Hinge 損失の比を計測している。比が 1 未満であれば提案手法の平均 Hinge 損失が小さいことを意味する。結果より、提案手法がより小さい Hinge 損失を与えていることが確かめられた。

4.2 コーデック性能評価

3.3 節のコーデック SRLA を C 言語で実装し、既存コーデックとの性能比較実験を行った。成果物は MIT ライセンスの元公開している [24]。音楽データとして RWC 音楽データセット [23] の一部 (表 1) を使用した。評価尺度として圧縮率と処理速度を以下で定義する。

$$\text{圧縮率} = \frac{\text{圧縮後サイズ (byte)}}{\text{wav ファイルサイズ (byte)}} \times 100, \quad (31)$$

$$\text{処理速度} = \frac{\text{処理時間 (sec)}}{\text{wav ファイルの長さ (sec)}} \times 100, \quad (32)$$

比較に用いた既存のコーデックと設定を表 2 に示す。SRLA のコーデックの設定は、音声フレームサイズは 4096、プリエンファシスの適用回数は 2、最大次数は 32、手法は表 3 に示す。表 3 の次数選択において”網羅探索”は、最大次数までの全ての次数で LD アルゴリズムで係数を求め予測を行い、平均符号長 (式 (8)) が最小となる次数を選択してい

表 2 コーデック設定

コーデック	バージョン	エンコードオプション
FLAC	1.3.2	-5, -8
WavPack	5.4.0	デフォルト, -h, -hh
TTA	2.3	デフォルト
Monkey's Audio	6.14	-c1000, -c2000, -c3000
MPEG4-ALS	RM23[25]	デフォルト
SRLA	0.0.1	-m 0, -m 1, -m 2, -m 3, -m 4, -m 5

表 3 SRLA (Version.0.0.1) のエンコードオプション

オプション	次数選択	係数設定	オプション	次数選択	係数設定
-m 0	最大固定	Algo.1	-m 3	(27) 式	Algo.3
-m 1	最大固定	Algo.3	-m 4	網羅探索	Algo.1
-m 2	(27) 式	Algo.1	-m 5	網羅探索	Algo.3

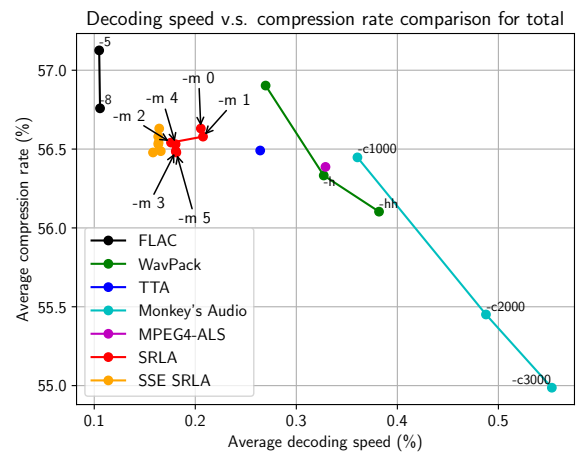


図 6 デコード速度に対する圧縮率の比較

る。実験は、CPU は Intel[®] Core™ i9-11900H、RAM は 32GB の Windows 11 OS 上で行った。SRLA は Microsoft Visual Studio 2022 の Release 設定でビルドした。

デコード速度に対する圧縮率の比較を図 6 に、エンコード速度の比較を表 4 に示す。なお、Intel[®] SSE 4.1 までの命令を使用してデコードを高速化した結果を”SSE SRLA”で示している。結果より、SRLA は FLAC よりも優れた圧縮率を示し、デコード速度は FLAC に次いで高速だった。アルゴリズム 3 は LD アルゴリズムより圧縮率は 0.05% 程度優れ、SSE の使用によりデコード速度は高速化した。また、次数選択を行うことでより小さい次数が選択され、デコード速度が改善した。とくに式 (27) による次数選択により、網羅探索に比べ僅かな圧縮率悪化と引き換えにエンコード速度の改善が得られた。一方、SRLA はエンコード速度が最も遅かった。

表 4 エンコード速度 ((32) 式) 比較

コーデック	FLAC -5	FLAC -8	WavPack	WavPack -h	WavPack -hh	TTA	MPEG4-ALS	Monkey's -c1000
平均	0.18	0.39	0.28	0.37	0.46	0.23	0.73	0.24
コーデック	Monkey's -c2000	Monkey's -c3000	SRLA -m 0	SRLA -m 1	SRLA -m 2	SRLA -m 3	SRLA -m 4	SRLA -m 5
平均	0.35	0.40	1.18	8.22	1.24	6.15	3.08	7.96

5. 考察

LD アルゴリズムとの比較では、マージン幅が小さい場合は係数次数の増加に伴い改善傾向が見られた。一方マージン幅が大きい場合は次数を高めると改善率が悪化した。マージン幅が広がると $S_\varepsilon(\nu) \approx 0$ となり勾配が消失し、初期値の LD アルゴリズムの係数からの変化が小さくなっていると考えられる。

エンコード処理は、残差の計算 (アルゴリズム 3 の 2 行目) が主な負荷を占めていた。これは、マージン幅 ε を網羅探索しており、探索の度にアルゴリズム 3 が実行されているのが原因である。アルゴリズム実行前に最適な ε の推定を行うことが重要と見ている。

6. 結論

本稿では SVR に基づく FIR システム同定手法を提案し、LD アルゴリズムとの比較実験とコーデック実装を示した。

将来の課題として手法の拡張を検討している。本研究では特徴写像を恒等写像 $\phi(\mathbf{x}) = \mathbf{x}$, かつカーネル関数を標準内積に限定していたが、非線形な特徴写像をとることでより予測精度が高いシステムを同定できる可能性がある。また本手法の収束レートを評価することや、手法を FLAC に組み込み圧縮率改善を示すことも重要な課題である。

参考文献

- [1] Konstantinides, K.: An introduction to super audio CD and DVD-audio, *IEEE signal processing magazine*, Vol. 20, No. 4, pp. 71–82 (2003).
- [2] Moriya, T., Harada, N., Kamamoto, Y. and Sekigawa, H.: MPEG-4 ALS international standard for lossless audio coding, *NTT Technical Review*, Vol. 4, No. 8, pp. 40–45 (2006).
- [3] Hans, M. and Schafer, R. W.: Lossless compression of digital audio, *IEEE Signal processing magazine*, Vol. 18, No. 4, pp. 21–32 (2001).
- [4] Robinson, T.: SHORTEN: Simple lossless and near-lossless waveform compression (1994).
- [5] Coalson, J. and Xiph Org Foundation: FLAC - Free Lossless Audio Codec, <https://xiph.org/flac/>.
- [6] Liebchen, T.: MPEG-4 ALS-the standard for lossless audio coding, *the Journal of the Acoustical Society of Korea*, Vol. 28, No. 7, pp. 618–629 (2009).
- [7] Rice, R. F.: Practical universal noiseless coding, *Applications of Digital Image Processing III*, Vol. 207, International Society for Optics and Photonics, pp. 247–267 (1979).
- [8] Bryant, D.: WavPack Audio Compression, <http://www.wavpack.com>.
- [9] Tau Software: TTA Lossless Audio Codec - True Audio Compressor Algorithms, http://tausoft.org/wiki/True_Audio_Codec_Overview.
- [10] Salomon, D.: *Data compression - The Complete Reference, 4th Edition*. (2007).
- [11] Levinson, N.: The Wiener (Root Mean Square) Error Criterion in Filter Design and Prediction, *Journal of Mathematics and Physics*, Vol. 25, pp. 261–278 (1946).
- [12] Durbin, J.: The fitting of time-series models, *Reu, Intern&. Statist. Inst. 28: 229*, Vol. 249, pp. 201–233 (1959).
- [13] Kameoka, H., Kamamoto, Y., Harada, N. and Moriya, T.: A Linear Predictive Coding Algorithm Minimizing the Golomb-Rice Code Length of the Residual Signal, *The Transactions of the Institute of Electronics, Information and Communication Engineers. A*, Vol. 91, pp. 1017–1025 (2008).
- [14] Vapnik, V. N.: The Nature of Statistical Learning Theory, *Statistics for Engineering and Information Science* (2000).
- [15] Mineo, T. and Shouno, H.: A lossless audio codec based on hierarchical residual prediction, *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 123–130 (2022).
- [16] Bishop, C. M.: *Pattern Recognition and Machine Learning*, Springer (2006).
- [17] Platt, J.: Sequential minimal optimization: A fast algorithm for training support vector machines (1998).
- [18] Elad, M.: スパースモデリング: 11/10 ノルム最小化の基礎理論と画像処理への応用, 共立出版 (2016).
- [19] 川野秀一, 松井秀俊, 廣瀬慧: スパース推定法による統計モデリング, 共立出版 (2018).
- [20] 永原正章: スパースモデリング: 基礎から動的システムへの応用, コロナ社 (2017).
- [21] Kamamoto, Y., Moriya, T., Harada, N. and Hiwasaki, Y.: Low-complexity PARCOR coefficient quantization and prediction order estimation designed for entropy coding of prediction residuals, *Acoustical Science and Technology*, Vol. 34, No. 2, pp. 105–112 (2013).
- [22] Kabal, P.: Ill-conditioning and bandwidth expansion in linear prediction of speech, *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*. (2003).
- [23] Goto, M., Hashiguchi, H., Nishimura, T. and Oka, R.: RWC Music Database: Popular, Classical and Jazz Music Databases., *Ismir*, Vol. 2, pp. 287–288 (2002).
- [24] aikiriao: SRLA, <https://github.com/aikiriao/SRLA>.
- [25] Technische Universität Berlin: Forschung: MPEG-4 Audio Lossless Coding (ALS), https://www.nue.tu-berlin.de/menue/research/research_topic/compression_and_transmission/mpeg_4_audio_lossless_coding_als/parameter/en/#c230252.