PRE-PROCESSING DESIGN FOR MULTIWAVELET FILTERS USING NEURAL NETWORKS

RYUICHI ASHINO
Division of Mathematical Sciences,
Osaka Kyoiku University,
Kashiwara, Osaka 582-8582, Japan
E-mail: ashino@cc.osaka-kyoiku.ac.jp

AKIRA MORIMOTO
Division of Information Science,
Osaka Kyoiku University,
Kashiwara, Osaka 582-8582, Japan
E-mail: morimoto@cc.osaka-kyoiku.ac.jp

Dedicate to Professor Michihiro NAGASE on the occasion of his 60th birthday

A pre-processing design for multiwavelet filters using neural networks is proposed. Various numerical experiments are presented and a comparison between a pre-processing using neural networks and a pre-processing solving linear system method is given. This pre-processing using neural networks performs well for approximation with large number of terms and converges fast.

1. Introduction

Since wavelets are solutions of multiscale equations, they can hardly be studied as mathematical objects and in the applications without the use of computers. This paper is no exception. Multiwavelets consist of several scaling functions and wavelets. It is believed that multiwavelets are ideally suited to multichannel signals like color images which are two-dimensional

*This research was partially supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Scientific Research (C), 13640171(2001-2002)
three-channel signals and stereo audio signals which are one-dimensional two-channel signals. Multiscaling functions and multiwavelets can simultaneously have orthogonality, linear phase, symmetry and compact support. This situation cannot occur with real uniscaling functions and real unwavelets.

Using fractal interpolation, Geronimo, Hardin, and Massopust\(^{2}\) constructed a real-valued one-dimensional symmetric two-scaling function with short support, and Donavan, Geronimo, Hardin, and Massopust\(^{3}\) constructed a corresponding real-valued one-dimensional two-wavelet (DGHM) with short support. Strang and Strela\(^{4}\) and Strang and Strela\(^{5}\) used matrix methods in the time domain to construct the DGHM two-wavelet and a nonsymmetric pair, respectively. Assuming sufficiently many vanishing moments of the scaling functions, Ashino and Kametani\(^{6}\) introduced \(r\)-regular multiwavelets in \(L^2(\mathbb{R}^n)\) and proved a general existence theorem, following Meyer’s general existence theorem (see Meyer\(^{1}\), Theorem 2 of Section 3.6 and Proposition 4 of Section 3.7). Jia and Shen\(^{7}\) investigated multiresolution on the basis of shift-invariant spaces, proved a general existence theorem and gave examples to illustrate the general theory. Using results of Lawton\(^{8}\) on complex-valued filters, Cooklev\(^{9}\) and Cooklev et al.\(^{10}\) obtained one-dimensional perfect-reconstruction two-filter banks given by a pair of analyzing and synthesizing orthogonal linear-phase two-channel multiwavelet filters. Plonka\(^{11}\), Cohen, Daubechies and Plonka\(^{12}\), Plonka and Strela\(^{13}\), Shen\(^{14}\), Strela\(^{15}\), and many others, have obtained important results on the existence, regularity, orthogonality and symmetry of multiwavelets. Definitions and properties of multiwavelets, filters and filter banks can be found, for instance, in Ashino, Nagase, and Vaillancourt\(^{16}\) and Zheng\(^{17}\).

To start with multiwavelet filterings, we need to get scaling coefficients at high resolution. In the case of multiwavelets constructed by \(d\)-multiscaling functions, there are \(d\)-channel to input for each sample of data, because frequently used multiwavelets have multiscaling functions with similar support widths. For fast algorithms of multiwavelet filters, a given data needs to be pre-processed into \(d\)-input for reducing their sizes. In the scalar case, that is, unwavelet case, samples of a given function are used as the scaling function, because the scaling function is close to a constant multiple of a translate of the delta function at very high resolution. But in the vector case, that is, multiwavelet case, simply using nearby samples as the scaling functions is a bad choice, because each of the \(d\)-scaling functions may not be close to a constant multiple of a translate of the delta functions.
even at very high resolution. By these reasons, data samples need to be
pre-processed to produce reasonable values of the expansion coefficients for
multiscaling functions at the highest scale. This kind of pre-processing is
also called a prefilter. Prefilters have been designed based on interpolation
Xia, Geronimo, Hardin and Suter\textsuperscript{18}, approximation Hardin and Roach\textsuperscript{19},
and orthogonal projection Vrhel and Aldroubi\textsuperscript{20}.

In this paper, we propose a pre-processing using neural networks, which
can be adaptively changed to each data. Our pre-processing is efficient when
many approximation coefficients are used. Our results are the following.

Theorem 1.

(i) In the case of high resolution approximations, our pre-processing
performs much better in accuracy than the method of numerical integra-
tion.

(ii) In the case of low resolution approximations, our pre-processing per-
forms better in accuracy by one digit than the method of numerical
integration, but the method of inverse matrix cheaper than our pre-
processing.

(iii) In the case of large data, our pre-processing saves memory compar-
ing to the conjugate gradient method at the same computing cost.

2. Multiwavelets

Notation 1. We will use the following notation.

- Given a function $f \in L^2(\mathbb{R})$ and given $j \in \mathbb{Z}$ and $k \in \mathbb{Z}$, we let
  $f_{jk}(x)$ denote the scaled and shifted function
  
  $f_{jk}(x) = 2^{j/2} f(2^j x - k)$.

- Given $d$ functions $f^1, \ldots, f^d \in L^2(\mathbb{R})$, we let $F$ denote the vector
  of functions $F = [f^1, \ldots, f^d] \in L^2(\mathbb{R})^d$. Further, $F_{jk}$ is the vector
  of scaled and shifted functions $F_{jk} = [f^1_{jk}, \ldots, f^d_{jk}]$.

- $D = \{1, \ldots, d\}$ for a positive integer $d$.

- $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$ is the set of natural numbers including zero.

- $\langle \cdot, \cdot \rangle$ denotes the inner product on $L^2(\mathbb{R})$.

A multiwavelet function, or multiwavelet $\Psi := \{\psi^1, \ldots, \psi^d\} \in (L^2(\mathbb{R}))^d$
is constructed from a multiresolution analysis $\{V_j\}_{j \in \mathbb{Z}}$ and a multiscaling
function $\Phi := \{\phi^1, \ldots, \phi^d\} \in (V_0)^d$, which can be found in Meyer\textsuperscript{1} for the
uniwavelet case. The components $\psi^\delta$ and the components $\varphi^\delta$ are also called as multiwavelets and multiscaling functions, respectively. Denote

$$H_k := \langle \Phi(x), \Phi(2x - k) \rangle_{L^2(\mathbb{R})}$$

$$= \left[ \langle \varphi^\delta(x), \varphi^\eta(2x - k) \rangle_{L^2(\mathbb{R})} \right]_{(\delta, \eta) \in D \times D} \in \mathbb{C}^{d \times d},$$

$$M_0(\xi) := \sum_{k \in \mathbb{Z}} H_k e^{-ik\xi} \in L^2([0, 2\pi])^{d \times d},$$

then we have the dilation equation and its Fourier transform:

$$\Phi(x) = 2 \sum_{k \in \mathbb{Z}} H_k \Phi(2x - k), \quad \hat{\Phi}(\xi) = M_0(\xi/2) \hat{\Phi}(\xi/2),$$

where $\hat{\Phi}(\xi) := \langle \varphi^\xi, \cdots, \varphi^\xi \rangle \in L^2(\mathbb{R})^d$. It is known that if we choose $M_1(\xi)$ such that

$$M(\xi) := \begin{bmatrix} M_0(\xi) & M_0(\xi + \pi) \\ M_1(\xi) & M_1(\xi + \pi) \end{bmatrix}$$

is an unitary matrix for almost all $\xi \in [0, 2\pi]$, then the multiwavelet function $\Psi$ is given by the wavelet equation or by its Fourier transform:

$$\Psi(x) = 2 \sum_{k \in \mathbb{Z}} G_k \Phi(2x - k), \quad \hat{\Psi}(\xi) = M_1(\xi/2) \hat{\Phi}(\xi/2),$$

where $G_k$ are the Fourier coefficients of $M_1(\xi)$, that is,

$$M_1(\xi) = \sum_{k \in \mathbb{Z}} G_k e^{-ik\xi} \in L^2([0, 2\pi])^{d \times d}.$$

Then, the system $\{\psi^\delta jk(x) := 2^{j/2} \psi^\delta(2^j x - k)\}_{\delta \in D, j \in \mathbb{Z}, k \in \mathbb{Z}}$ forms an orthonormal basis for $L^2(\mathbb{R})$.

3. Approximation using multiscaling functions

Hereafter we only deal with the real-valued case and assume that the number of multiscaling functions is two, that is, $d = 2$.

Our problem is to find the best approximation for $f \in L^2(\mathbb{R})$ by $V_j$. As each element $S_j \in V_j$ is represented as

$$S_j(x) = \sum_{k} C^1_{j,k} \varphi^1(2^j x - k) + \sum_{k} C^2_{j,k} \varphi^2(2^j x - k),$$

our problem becomes to find coefficients that minimize the integral

$$\int_{\mathbb{R}} |f(x) - S_j(x)|^2 \, dx.$$
When $\Phi = [\varphi_1(x), \varphi_2(x)]$ is an orthonormal multiscaling function, the best approximation is given by

$$C_{1,j,k} = 2^j \int f(x) \varphi_1(2^j x - k) \, dx, \quad C_{2,j,k} = 2^j \int f(x) \varphi_2(2^j x - k) \, dx,$$

which can be calculated by numerical integration.

When a given data consists of equally spaced samples, the integral (2) can be approximated by the sampling width $\delta$ multiple of

$$E\{C_{1,j,k}, C_{2,j,k}\} = \sum_n |f(x_n) - S_j(x_n)|^2.$$

We expect that the least square solution to (3) is an accurate approximation at the point $x_n$.

4. Multiwavelet neural networks

Assume that each summation of $\varphi_1(2^j x - k)$ and $\varphi_2(2^j x - k)$ in (1) contains $L$ terms. Consider three layers neural networks as Figure 1. Its input is $x$ and its output is $S_j(x)$. Then, the backpropagation learning method gives the least square solution to (3). This neural networks is called a multiwavelet neural networks.

![Figure 1. Multiwavelet neural networks.](image)

References


