MAT 271: Applied & Computational Harmonic Analysis
Lecture 19: *A Library of Orthonormal Bases and Adapted Signal Analysis*

Naoki Saito

Department of Mathematics
University of California, Davis

March 11, 2014

Outline

Outline

Our Basic Setup

- Consider an ensemble of N 1D discrete signals, $\boldsymbol{x}_m \in \mathbb{R}^n$, $m = 1, \ldots, N$; we then form the data matrix $X \in \mathbb{R}^{n \times N}$ consisting those signals as column vectors.
- For the notational convenience, let $\boldsymbol{x}_m = \left(x_{0,m}, x_{1,m}, \ldots, x_{n-1,m}\right)^{\mathsf{T}}$.
- There are many tasks given X, such as joint compression; classifying them into a set of groups in a supervised or unsupervised manner (classification vs clustering), ...
- In order to perform such tasks efficiently, it is a good idea to use a basis that is *adapted* to a given task and to the signal ensemble.
- Once such a basis is selected, we can expand each $\boldsymbol{x}_m$ relative to the basis and analyze the coefficients/coordinates for the given task.

Our Basic Setup

- Consider an ensemble of N 1D discrete signals, $\boldsymbol{x}_m \in \mathbb{R}^n$, $m = 1, \ldots, N$; we then form the data matrix $X \in \mathbb{R}^{n \times N}$ consisting those signals as column vectors.
- For the notational convenience, let $\boldsymbol{x}_m = \left(x_{0,m}, x_{1,m}, \ldots, x_{n-1,m}\right)^{\mathsf{T}}$.
- There are many tasks given X, such as joint compression; classifying them into a set of groups in a supervised or unsupervised manner (classification vs clustering), ...
- In order to perform such tasks efficiently, it is a good idea to use a basis that is *adapted* to a given task and to the signal ensemble.
- Once such a basis is selected, we can expand each $\boldsymbol{x}_m$ relative to the basis and analyze the coefficients/coordinates for the given task.

Our Basic Setup

- Consider an ensemble of N 1D discrete signals, $\boldsymbol{x}_m \in \mathbb{R}^n$, $m = 1, \ldots, N$; we then form the data matrix $X \in \mathbb{R}^{n \times N}$ consisting those signals as column vectors.
- For the notational convenience, let $\boldsymbol{x}_m = \left(x_{0,m}, x_{1,m}, \ldots, x_{n-1,m}\right)^{\mathsf{T}}$.
- There are many tasks given X, such as joint compression; classifying them into a set of groups in a supervised or unsupervised manner (classification vs clustering), ...
- In order to perform such tasks efficiently, it is a good idea to use a basis that is *adapted* to a given task and to the signal ensemble.
- Once such a basis is selected, we can expand each $\boldsymbol{x}_m$ relative to the basis and analyze the coefficients/coordinates for the given task.

Our Basic Setup

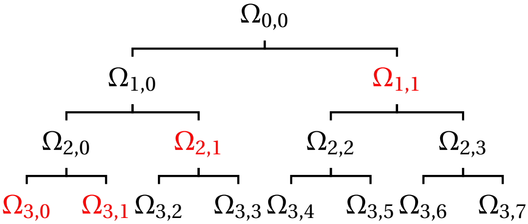
- Consider an ensemble of N 1D discrete signals, $\boldsymbol{x}_m \in \mathbb{R}^n$, $m = 1, \ldots, N$; we then form the data matrix $X \in \mathbb{R}^{n \times N}$ consisting those signals as column vectors.
- For the notational convenience, let $\boldsymbol{x}_m = \left(x_{0,m}, x_{1,m}, \ldots, x_{n-1,m}\right)^{\mathsf{T}}$.
- There are many tasks given X, such as joint compression; classifying them into a set of groups in a supervised or unsupervised manner (classification vs clustering), ...
- In order to perform such tasks efficiently, it is a good idea to use a basis that is *adapted* to a given task and to the signal ensemble.
- Once such a basis is selected, we can expand each $\boldsymbol{x}_m$ relative to the basis and analyze the coefficients/coordinates for the given task.

Our Basic Setup

- Consider an ensemble of N 1D discrete signals, $\boldsymbol{x}_m \in \mathbb{R}^n$, $m = 1, \ldots, N$; we then form the data matrix $X \in \mathbb{R}^{n \times N}$ consisting those signals as column vectors.
- For the notational convenience, let $\boldsymbol{x}_m = \left(x_{0,m}, x_{1,m}, \ldots, x_{n-1,m}\right)^{\mathsf{T}}$.
- There are many tasks given X, such as joint compression; classifying them into a set of groups in a supervised or unsupervised manner (classification vs clustering), ...
- In order to perform such tasks efficiently, it is a good idea to use a basis that is *adapted* to a given task and to the signal ensemble.
- Once such a basis is selected, we can expand each $\boldsymbol{x}_m$ relative to the basis and analyze the coefficients/coordinates for the given task.

A Library of Orthonormal Bases

A library of orthonormal bases consists of *dictionaries of orthonormal bases*: each dictionary is a *binary tree* whose nodes are subspaces of $\Omega_{0,0} = \mathbb{R}^n$ with different time-frequency localization characteristics.



Basis Dictionaries in a Library

- Examples of dictionaries include:
  - *Wavelet Packet Bases*
  - *Block Discrete Cosine Bases*
  - Local Trigonometric/Fourier Bases
- It costs $O(n[\log n]^p)$ to generate a dictionary for a signal of length n ($p = 1$ for wavelet packets, $p = 2$ for BDCT/LTB).
- Each dictionary may contain up to $n(1 + \log_2 n)$ basis vectors and more than $2^{n/2}$ possible orthonormal bases.
- How to select the best possible basis for the problem at hand is a key issue.

Basis Dictionaries in a Library

- Examples of dictionaries include:
  - *Wavelet Packet Bases*
  - *Block Discrete Cosine Bases*
  - Local Trigonometric/Fourier Bases
- It costs $O(n[\log n]^p)$ to generate a dictionary for a signal of length n ($p = 1$ for wavelet packets, $p = 2$ for BDCT/LTB).
- Each dictionary may contain up to $n(1 + \log_2 n)$ basis vectors and more than $2^{n/2}$ possible orthonormal bases.
- How to select the best possible basis for the problem at hand is a key issue.

Basis Dictionaries in a Library

- Examples of dictionaries include:
  - *Wavelet Packet Bases*
  - *Block Discrete Cosine Bases*
  - Local Trigonometric/Fourier Bases
- It costs $O(n[\log n]^p)$ to generate a dictionary for a signal of length n ($p = 1$ for wavelet packets, $p = 2$ for BDCT/LTB).
- Each dictionary may contain up to $n(1 + \log_2 n)$ basis vectors and more than $2^{n/2}$ possible orthonormal bases.
- How to select the best possible basis for the problem at hand is a key issue.

Basis Dictionaries in a Library

- Examples of dictionaries include:
  - *Wavelet Packet Bases*
  - *Block Discrete Cosine Bases*
  - Local Trigonometric/Fourier Bases
- It costs $O(n[\log n]^p)$ to generate a dictionary for a signal of length n ($p=1$ for wavelet packets, $p=2$ for BDCT/LTB).
- Each dictionary may contain up to $n(1+\log_2 n)$ basis vectors and more than $2^{n/2}$ possible orthonormal bases.
- How to select the best possible basis for the problem at hand is a key issue.

Basis Dictionaries in a Library

- Examples of dictionaries include:
  - *Wavelet Packet Bases*
  - *Block Discrete Cosine Bases*
  - Local Trigonometric/Fourier Bases
- It costs $O(n[\log n]^p)$ to generate a dictionary for a signal of length n ($p = 1$ for wavelet packets, $p = 2$ for BDCT/LTB).
- Each dictionary may contain up to $n(1 + \log_2 n)$ basis vectors and more than $2^{n/2}$ possible orthonormal bases.
- How to select the best possible basis for the problem at hand is a key issue.

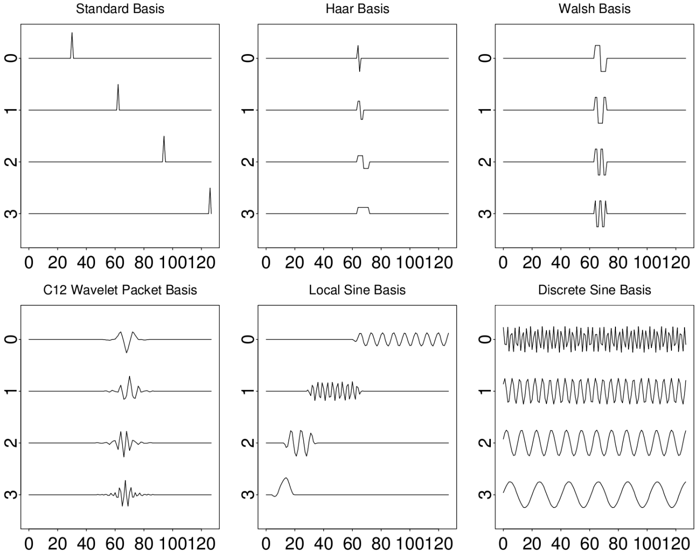
Basis Dictionaries in a Library

- Examples of dictionaries include:
  - *Wavelet Packet Bases*
  - *Block Discrete Cosine Bases*
  - Local Trigonometric/Fourier Bases
- It costs $O(n[\log n]^p)$ to generate a dictionary for a signal of length n ($p = 1$ for wavelet packets, $p = 2$ for BDCT/LTB).
- Each dictionary may contain up to $n(1 + \log_2 n)$ basis vectors and more than $2^{n/2}$ possible orthonormal bases.
- How to select the best possible basis for the problem at hand is a key issue.

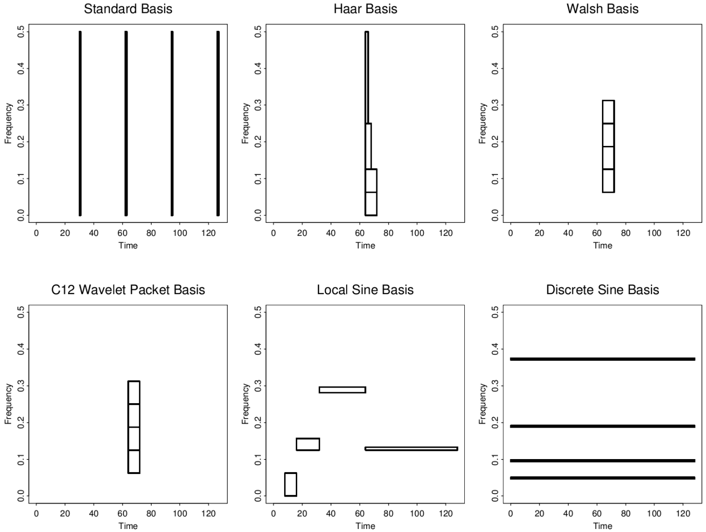
Basis Dictionaries in a Library

- Examples of dictionaries include:
  - *Wavelet Packet Bases*
  - *Block Discrete Cosine Bases*
  - Local Trigonometric/Fourier Bases
- It costs $O(n[\log n]^p)$ to generate a dictionary for a signal of length n ($p = 1$ for wavelet packets, $p = 2$ for BDCT/LTB).
- Each dictionary may contain up to $n(1 + \log_2 n)$ basis vectors and more than $2^{n/2}$ possible orthonormal bases.
- How to select the best possible basis for the problem at hand is a key issue.

Example of Local Basis Functions



Time-Frequency Characteristics



Outline

A Wavelet Packet Dictionary

- View $\Omega_{0,0}$ as the basic space V_0 of Multiresolution Analysis
- A pair of filters $\{H, G\}$ consisting of convolution with the CMF coefficients $\{h_\ell\}$, $\{g_\ell\}$, and subsequent subsampling, are applied to each $x_m \in \Omega_{0,0}$, $m = 1, \ldots, N$.
- As usual, we need to pay attention to the boundary treatment of the signals (e.g., need to do even reflection at the boundary or periodization).
- $H x_m \in \Omega_{1,0} = V_1$ while $G x_m \in \Omega_{1,1} = W_1$. Hence, $\Omega_{0,0} = \Omega_{1,0} \oplus \Omega_{1,1}$.
- In the case of the *Discrete Wavelet Transform*, we iterate this filtering operations only on the *lower frequency* subspaces, i.e., $\Omega_{j-1,0} = \Omega_{j,0} \oplus \Omega_{j,1}$, $j = 1, \ldots, J (\leq \log_2 n)$. The high frequency subspaces $\Omega_{j,1} = W_j$, $j = 1, \ldots, J$ are kept intact once they are generated.
- A *Wavelet Packet Dictionary* also iterates the above filtering procedure on the *higher frequency* subspaces, i.e., $\Omega_{j-1,k} = \Omega_{j,2k} \oplus \Omega_{j,2k+1}$, $j = 1, \ldots, J$, $k = 0, \ldots, 2^j - 1$, are generated.

A Wavelet Packet Dictionary

- View $\Omega_{0,0}$ as the basic space V_0 of Multiresolution Analysis
- A pair of filters $\{H, G\}$ consisting of convolution with the CMF coefficients $\{h_\ell\}$, $\{g_\ell\}$, and subsequent subsampling, are applied to each $\boldsymbol{x}_m \in \Omega_{0,0}$, $m = 1, \ldots, N$.
- As usual, we need to pay attention to the boundary treatment of the signals (e.g., need to do even reflection at the boundary or periodization).
- $H\boldsymbol{x}_m \in \Omega_{1,0} = V_1$ while $G\boldsymbol{x}_m \in \Omega_{1,1} = W_1$. Hence, $\Omega_{0,0} = \Omega_{1,0} \oplus \Omega_{1,1}$.
- In the case of the *Discrete Wavelet Transform*, we iterate this filtering operations only on the *lower frequency* subspaces, i.e., $\Omega_{j-1,0} = \Omega_{j,0} \oplus \Omega_{j,1}$, $j = 1, \ldots, J (\le \log_2 n)$. The high frequency subspaces $\Omega_{j,1} = W_j$, $j = 1, \ldots, J$ are kept intact once they are generated.
- A *Wavelet Packet Dictionary* also iterates the above filtering procedure on the *higher frequency* subspaces, i.e., $\Omega_{j-1,k} = \Omega_{j,2k} \oplus \Omega_{j,2k+1}$, $j = 1, \ldots, J$, $k = 0, \ldots, 2^j - 1$, are generated.

A Wavelet Packet Dictionary

- View $\Omega_{0,0}$ as the basic space V_0 of Multiresolution Analysis
- A pair of filters $\{H, G\}$ consisting of convolution with the CMF coefficients $\{h_\ell\}$, $\{g_\ell\}$, and subsequent subsampling, are applied to each $\boldsymbol{x}_m \in \Omega_{0,0},\ m = 1, \ldots, N$.
- As usual, we need to pay attention to the boundary treatment of the signals (e.g., need to do even reflection at the boundary or periodization).
- $H\boldsymbol{x}_m \in \Omega_{1,0} = V_1$ while $G\boldsymbol{x}_m \in \Omega_{1,1} = W_1$. Hence, $\Omega_{0,0} = \Omega_{1,0} \oplus \Omega_{1,1}$.
- In the case of the *Discrete Wavelet Transform*, we iterate this filtering operations only on the *lower frequency* subspaces, i.e., $\Omega_{j-1,0} = \Omega_{j,0} \oplus \Omega_{j,1},\ j = 1, \ldots, J (\le \log_2 n)$. The high frequency subspaces $\Omega_{j,1} = W_j,\ j = 1, \ldots, J$ are kept intact once they are generated.
- A *Wavelet Packet Dictionary* also iterates the above filtering procedure on the *higher frequency* subspaces, i.e., $\Omega_{j-1,k} = \Omega_{j,2k} \oplus \Omega_{j,2k+1},\ j = 1, \ldots, J,\ k = 0, \ldots, 2^j - 1$, are generated.

A Wavelet Packet Dictionary

- View $\Omega_{0,0}$ as the basic space V_0 of Multiresolution Analysis
- A pair of filters $\{H, G\}$ consisting of convolution with the CMF coefficients $\{h_\ell\}$, $\{g_\ell\}$, and subsequent subsampling, are applied to each $\boldsymbol{x}_m \in \Omega_{0,0}$, $m = 1, \ldots, N$.
- As usual, we need to pay attention to the boundary treatment of the signals (e.g., need to do even reflection at the boundary or periodization).
- $H\boldsymbol{x}_m \in \Omega_{1,0} = V_1$ while $G\boldsymbol{x}_m \in \Omega_{1,1} = W_1$. Hence, $\Omega_{0,0} = \Omega_{1,0} \oplus \Omega_{1,1}$.
- In the case of the *Discrete Wavelet Transform*, we iterate this filtering operations only on the *lower frequency* subspaces, i.e., $\Omega_{j-1,0} = \Omega_{j,0} \oplus \Omega_{j,1}$, $j = 1, \ldots, J (\le \log_2 n)$. The high frequency subspaces $\Omega_{j,1} = W_j$, $j = 1, \ldots, J$ are kept intact once they are generated.
- A *Wavelet Packet Dictionary* also iterates the above filtering procedure on the *higher frequency* subspaces, i.e., $\Omega_{j-1,k} = \Omega_{j,2k} \oplus \Omega_{j,2k+1}$, $j = 1, \ldots, J$, $k = 0, \ldots, 2^j - 1$, are generated.

A Wavelet Packet Dictionary

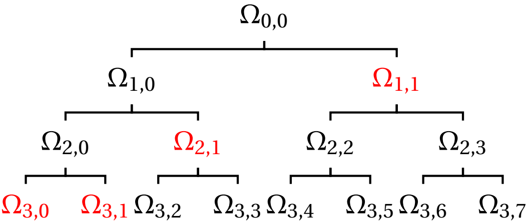
- View $\Omega_{0,0}$ as the basic space V_0 of Multiresolution Analysis
- A pair of filters $\{H, G\}$ consisting of convolution with the CMF coefficients $\{h_\ell\}$, $\{g_\ell\}$, and subsequent subsampling, are applied to each $\boldsymbol{x}_m \in \Omega_{0,0}$, $m = 1, \ldots, N$.
- As usual, we need to pay attention to the boundary treatment of the signals (e.g., need to do even reflection at the boundary or periodization).
- $H\boldsymbol{x}_m \in \Omega_{1,0} = V_1$ while $G\boldsymbol{x}_m \in \Omega_{1,1} = W_1$. Hence, $\Omega_{0,0} = \Omega_{1,0} \oplus \Omega_{1,1}$.
- In the case of the *Discrete Wavelet Transform*, we iterate this filtering operations only on the *lower frequency* subspaces, i.e., $\Omega_{j-1,0} = \Omega_{j,0} \oplus \Omega_{j,1}$, $j = 1, \ldots, J (\le \log_2 n)$. The high frequency subspaces $\Omega_{j,1} = W_j$, $j = 1, \ldots, J$ are kept intact once they are generated.
- A *Wavelet Packet Dictionary* also iterates the above filtering procedure on the *higher frequency* subspaces, i.e., $\Omega_{j-1,k} = \Omega_{j,2k} \oplus \Omega_{j,2k+1}$, $j = 1, \ldots, J$, $k = 0, \ldots, 2^j - 1$, are generated.

A Wavelet Packet Dictionary

- View $\Omega_{0,0}$ as the basic space V_0 of Multiresolution Analysis
- A pair of filters $\{H,G\}$ consisting of convolution with the CMF coefficients $\{h_\ell\}$, $\{g_\ell\}$, and subsequent subsampling, are applied to each $\boldsymbol{x}_m \in \Omega_{0,0}$, $m = 1,\dots,N$.
- As usual, we need to pay attention to the boundary treatment of the signals (e.g., need to do even reflection at the boundary or periodization).
- $H\boldsymbol{x}_m \in \Omega_{1,0} = V_1$ while $G\boldsymbol{x}_m \in \Omega_{1,1} = W_1$. Hence, $\Omega_{0,0} = \Omega_{1,0} \oplus \Omega_{1,1}$.
- In the case of the *Discrete Wavelet Transform*, we iterate this filtering operations only on the *lower frequency* subspaces, i.e., $\Omega_{j-1,0} = \Omega_{j,0} \oplus \Omega_{j,1}$, $j = 1,\dots,J(\le \log_2 n)$. The high frequency subspaces $\Omega_{j,1} = W_j$, $j = 1,\dots,J$ are kept intact once they are generated.
- A *Wavelet Packet Dictionary* also iterates the above filtering procedure on the *higher frequency* subspaces, i.e., $\Omega_{j-1,k} = \Omega_{j,2k} \oplus \Omega_{j,2k+1}$, $j = 1,\dots,J$, $k = 0,\dots,2^j - 1$, are generated.

A Wavelet Packet Dictionary ...

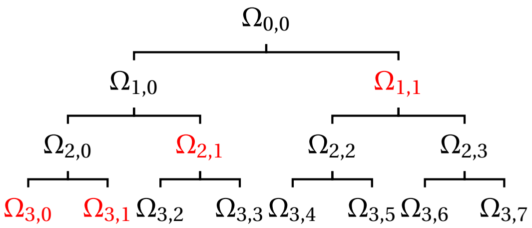
This generates a *complete binary tree* of subspaces $\{\Omega_{j,k}\}$ with $\dim \Omega_{j,k} = \frac{n}{2^j}$, $j = 0, \ldots, J$, $k = 0, \ldots, 2^j - 1$.



- Of course, in the above tree, we set $J = 3$.
- The red part forms the *wavelet basis*.
- The cost of expanding an input signal x_m into this binary tree of subspaces is $O(nJ)$, which can be easily understood by the repeated applications of the filtering operations at each level $j = 0, \ldots, J - 1$. Hence, the overall cost for the whole data matrix is $O(NnJ)$.

A Wavelet Packet Dictionary ...

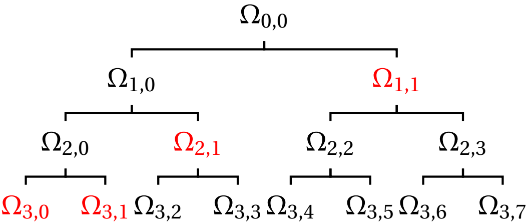
This generates a *complete binary tree* of subspaces $\{\Omega_{j,k}\}$ with $\dim \Omega_{j,k} = \frac{n}{2^j}$, $j = 0, \ldots, J$, $k = 0, \ldots, 2^j - 1$.



- Of course, in the above tree, we set $J = 3$.
- The red part forms the *wavelet basis*.
- The cost of expanding an input signal x_m into this binary tree of subspaces is $O(nJ)$, which can be easily understood by the repeated applications of the filtering operations at each level $j = 0, \ldots, J - 1$. Hence, the overall cost for the whole data matrix is $O(NnJ)$.

A Wavelet Packet Dictionary . . .

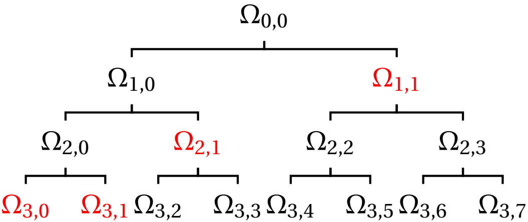
This generates a *complete binary tree* of subspaces $\{\Omega_{j,k}\}$ with $\dim \Omega_{j,k} = \frac{n}{2^j}$, $j = 0, \ldots, J$, $k = 0, \ldots, 2^j - 1$.



- Of course, in the above tree, we set $J = 3$.
- The red part forms the *wavelet basis*.
- The cost of expanding an input signal x_m into this binary tree of subspaces is $O(nJ)$, which can be easily understood by the repeated applications of the filtering operations at each level $j = 0, \ldots, J - 1$. Hence, the overall cost for the whole data matrix is $O(NnJ)$.

A Wavelet Packet Dictionary . . .

This generates a *complete binary tree* of subspaces $\{\Omega_{j,k}\}$ with $\dim \Omega_{j,k} = \frac{n}{2^j}$, $j = 0, \ldots, J$, $k = 0, \ldots, 2^j - 1$.



- Of course, in the above tree, we set $J = 3$.
- The red part forms the *wavelet basis*.
- The cost of expanding an input signal $\boldsymbol{x}_m$ into this binary tree of subspaces is $O(nJ)$, which can be easily understood by the repeated applications of the filtering operations at each level $j = 0, \ldots, J-1$. Hence, the overall cost for the whole data matrix is $O(NnJ)$.

Outline

The Block DCT Dictionary

- Again let $\Omega_{0,0} = \mathbb{R}^n$.
- Split each signal in X into two halves, i.e.,

$$\boldsymbol{x}_m = \boldsymbol{\chi}_{[0,\frac{n}{2}-1]} .* \boldsymbol{x}_m + \boldsymbol{\chi}_{[\frac{n}{2},n-1]} .* \boldsymbol{x}_m,$$

  where $.*$ is the element-wise multiplication (following the MATLAB convention), and $\chi_{[n_1,n_2]}$ is the indicator function (on integer grids):

$$\chi_{[n_1,n_2]}(i) := \begin{cases} 1 & \text{if } n_1 \le i \le n_2; \\ 0 & \text{otherwise.} \end{cases}$$

- Hence, $\dim \Omega_{1,0} = \dim \Omega_{1,1} = n/2$. We now apply the *DCT Type II* for length $n/2$ in those two half size signals. The cost is $O\left(2 \times \frac{n}{2} \log_2 \frac{n}{2}\right) \approx O(n \log_2 n)$.
- We repeat this splitting procedure recursively to generate the binary tree of subspaces $\{\Omega_{j,k}\}$, $j = 0, \ldots, J$, $k = 0, \ldots, 2^j - 1$ with $\dim \Omega_{j,k} = \frac{n}{2^j}$.
- Note that we are *splitting the spatial (or time) axis* instead of the frequency axis in the wavelet packet case.

The Block DCT Dictionary

- Again let $\Omega_{0,0} = \mathbb{R}^n$.
- Split each signal in X into two halves, i.e.,

$$\boldsymbol{x}_m = \boldsymbol{\chi}_{[0,\frac{n}{2}-1]} .* \boldsymbol{x}_m + \boldsymbol{\chi}_{[\frac{n}{2},n-1]} .* \boldsymbol{x}_m,$$

where $.*$ is the element-wise multiplication (following the MATLAB convention), and $\boldsymbol{\chi}_{[n_1,n_2]}$ is the indicator function (on integer grids):

$$\chi_{[n_1,n_2]}(i) := \begin{cases} 1 & \text{if } n_1 \le i \le n_2; \\ 0 & \text{otherwise.} \end{cases}$$

- Hence, $\dim \Omega_{1,0} = \dim \Omega_{1,1} = n/2$. We now apply the *DCT Type II* for length $n/2$ in those two half size signals. The cost is $O\left(2 \times \frac{n}{2} \log_2 \frac{n}{2}\right) \approx O(n \log_2 n)$.
- We repeat this splitting procedure recursively to generate the binary tree of subspaces $\{\Omega_{j,k}\}$, $j = 0, \ldots, J$, $k = 0, \ldots, 2^j - 1$ with $\dim \Omega_{j,k} = \frac{n}{2^j}$.
- Note that we are *splitting the spatial (or time) axis* instead of the frequency axis in the wavelet packet case.

The Block DCT Dictionary

- Again let $\Omega_{0,0} = \mathbb{R}^n$.
- Split each signal in X into two halves, i.e.,

$$\boldsymbol{x}_m = \boldsymbol{\chi}_{[0,\frac{n}{2}-1]} .* \boldsymbol{x}_m + \boldsymbol{\chi}_{[\frac{n}{2},n-1]} .* \boldsymbol{x}_m,$$

where $.*$ is the element-wise multiplication (following the MATLAB convention), and $\boldsymbol{\chi}_{[n_1,n_2]}$ is the indicator function (on integer grids):

$$\chi_{[n_1,n_2]}(i) := \begin{cases} 1 & \text{if } n_1 \le i \le n_2; \\ 0 & \text{otherwise.} \end{cases}$$

- Hence, $\dim \Omega_{1,0} = \dim \Omega_{1,1} = n/2$. We now apply the *DCT Type II* for length $n/2$ in those two half size signals. The cost is $O\left(2 \times \frac{n}{2} \log_2 \frac{n}{2}\right) \approx O(n \log_2 n)$.
- We repeat this splitting procedure recursively to generate the binary tree of subspaces $\{\Omega_{j,k}\}$, $j = 0,\ldots,J$, $k = 0,\ldots,2^j-1$ with $\dim \Omega_{j,k} = \frac{n}{2^j}$.
- Note that we are *splitting the spatial (or time) axis* instead of the frequency axis in the wavelet packet case.

The Block DCT Dictionary

- Again let $\Omega_{0,0} = \mathbb{R}^n$.
- Split each signal in X into two halves, i.e.,

$$\boldsymbol{x}_m = \boldsymbol{\chi}_{[0,\frac{n}{2}-1]} .* \boldsymbol{x}_m + \boldsymbol{\chi}_{[\frac{n}{2},n-1]} .* \boldsymbol{x}_m,$$

where $.*$ is the element-wise multiplication (following the MATLAB convention), and $\boldsymbol{\chi}_{[n_1,n_2]}$ is the indicator function (on integer grids):

$$\chi_{[n_1,n_2]}(i) := \begin{cases} 1 & \text{if } n_1 \le i \le n_2; \\ 0 & \text{otherwise.} \end{cases}$$

- Hence, $\dim \Omega_{1,0} = \dim \Omega_{1,1} = n/2$. We now apply the *DCT Type II* for length $n/2$ in those two half size signals. The cost is $O\left(2 \times \frac{n}{2} \log_2 \frac{n}{2}\right) \approx O(n \log_2 n)$.
- We repeat this splitting procedure recursively to generate the binary tree of subspaces $\{\Omega_{j,k}\}$, $j = 0, \ldots, J$, $k = 0, \ldots, 2^j - 1$ with $\dim \Omega_{j,k} = \frac{n}{2^j}$.
- Note that we are *splitting the spatial (or time) axis* instead of the frequency axis in the wavelet packet case.

The Block DCT Dictionary

- Again let $\Omega_{0,0} = \mathbb{R}^n$.
- Split each signal in X into two halves, i.e.,

$$\boldsymbol{x}_m = \boldsymbol{\chi}_{[0,\frac{n}{2}-1]} .* \boldsymbol{x}_m + \boldsymbol{\chi}_{[\frac{n}{2},n-1]} .* \boldsymbol{x}_m,$$

where $.*$ is the element-wise multiplication (following the MATLAB convention), and $\boldsymbol{\chi}_{[n_1,n_2]}$ is the indicator function (on integer grids):

$$\chi_{[n_1,n_2]}(i) := \begin{cases} 1 & \text{if } n_1 \le i \le n_2; \\ 0 & \text{otherwise.} \end{cases}$$

- Hence, $\dim \Omega_{1,0} = \dim \Omega_{1,1} = n/2$. We now apply the *DCT Type II* for length $n/2$ in those two half size signals. The cost is $O\left(2 \times \frac{n}{2} \log_2 \frac{n}{2}\right) \approx O(n \log_2 n)$.
- We repeat this splitting procedure recursively to generate the binary tree of subspaces $\{\Omega_{j,k}\}$, $j = 0, \ldots, J$, $k = 0, \ldots, 2^j - 1$ with $\dim \Omega_{j,k} = \frac{n}{2^j}$.
- Note that we are *splitting the spatial (or time) axis* instead of the frequency axis in the wavelet packet case.

The Block DCT Dictionary ...

- Note the DCT-II treats the boundary with *even reflection automatically*, i.e., a brutal cut by $\chi_{\left[\frac{kn}{2^j}, \frac{(k+1)n}{2^j}-1\right]}$ for the signals in $\Omega_{j,k}$ does not create artifitial discontinuities around the boundary points $x_{\frac{kn}{2^j},m}$, $x_{\frac{(k+1)n}{2^j},m}$.
- The total computational cost of expanding x_m into this BDCT dictionary is $O(nJ\log_2 n) \le O(n[\log_2 n]^2)$; hence for the whole data matrix X, it costs at most $O(Nn[\log_2 n]^2)$.
- The *local cosine transform* dictionary, originally developed by R. R. Coifman and Y. Meyer, uses the smoother cutoff functions instead of $\chi_{\left[\frac{kn}{2^j}, \frac{(k+1)n}{2^j}-1\right]}$, followed by DCT type IV, not by DCT type II.
- Unfortunately, a good implementation is not straightforward, and the advantage of using the smoother cutoff functions has not been drastic. However, I would recommend you to read the nicer implementation and discussions by Lars Villemoes [6].

The Block DCT Dictionary ...

- Note the DCT-II treats the boundary with *even reflection automatically*, i.e., a brutal cut by $\chi_{\left[\frac{kn}{2^j},\frac{(k+1)n}{2^j}-1\right]}$ for the signals in $\Omega_{j,k}$ does not create artifitial discontinuities around the boundary points $x_{\frac{kn}{2^j},m}$, $x_{\frac{(k+1)n}{2^j},m}$.
- The total computational cost of expanding $\boldsymbol{x}_m$ into this BDCT dictionary is $O(nJ\log_2 n) \le O(n[\log_2 n]^2)$; hence for the whole data matrix X, it costs at most $O(Nn[\log_2 n]^2)$.
- The *local cosine transform* dictionary, originally developed by R. R. Coifman and Y. Meyer, uses the smoother cutoff functions instead of $\chi_{\left[\frac{kn}{2^j},\frac{(k+1)n}{2^j}-1\right]}$, followed by DCT type IV, not by DCT type II.
- Unfortunately, a good implementation is not straightforward, and the advantage of using the smoother cutoff functions has not been drastic. However, I would recommend you to read the nicer implementation and discussions by Lars Villemoes [6].

The Block DCT Dictionary . . .

- Note the DCT-II treats the boundary with *even reflection automatically*, i.e., a brutal cut by $\chi_{\left[\frac{kn}{2^j}, \frac{(k+1)n}{2^j}-1\right]}$ for the signals in $\Omega_{j,k}$ does not create artifitial discontinuities around the boundary points $x_{\frac{kn}{2^j},m}$, $x_{\frac{(k+1)n}{2^j},m}$.
- The total computational cost of expanding $\boldsymbol{x}_m$ into this BDCT dictionary is $O(nJ\log_2 n) \le O(n[\log_2 n]^2)$; hence for the whole data matrix X, it costs at most $O(Nn[\log_2 n]^2)$.
- The *local cosine transform* dictionary, originally developed by R. R. Coifman and Y. Meyer, uses the smoother cutoff functions instead of $\chi_{\left[\frac{kn}{2^j}, \frac{(k+1)n}{2^j}-1\right]}$, followed by DCT type IV, not by DCT type II.
- Unfortunately, a good implementation is not straightforward, and the advantage of using the smoother cutoff functions has not been drastic. However, I would recommend you to read the nicer implementation and discussions by Lars Villemoes [6].

The Block DCT Dictionary . . .

- Note the DCT-II treats the boundary with *even reflection automatically*, i.e., a brutal cut by $\chi_{\left[\frac{kn}{2^j},\frac{(k+1)n}{2^j}-1\right]}$ for the signals in $\Omega_{j,k}$ does not create artifitial discontinuities around the boundary points $x_{\frac{kn}{2^j},m}$, $x_{\frac{(k+1)n}{2^j},m}$.
- The total computational cost of expanding $\boldsymbol{x}_m$ into this BDCT dictionary is $O(nJ\log_2 n) \leq O(n[\log_2 n]^2)$; hence for the whole data matrix X, it costs at most $O(Nn[\log_2 n]^2)$.
- The *local cosine transform* dictionary, originally developed by R. R. Coifman and Y. Meyer, uses the smoother cutoff functions instead of $\chi_{\left[\frac{kn}{2^j},\frac{(k+1)n}{2^j}-1\right]}$, followed by DCT type IV, not by DCT type II.
- Unfortunately, a good implementation is not straightforward, and the advantage of using the smoother cutoff functions has not been drastic. However, I would recommend you to read the nicer implementation and discussions by Lars Villemoes [6].

Outline

How Many ONBs in a Dictionary?

- We can associate $\Omega_{j,k}$ as the dyadic interval $I_{j,k} := \left[\frac{k}{2^j}, \frac{k+1}{2^j}\right) \subset [0,1) =: I,\ j = 0, \ldots, J,\ k = 0, \ldots, 2^j - 1.$
- Let the orthonormal basis of $\Omega_{j,k}$ generated by these hierarchical operations be $\{\boldsymbol{\psi}_{j,k,\ell}\}_{\ell=0}^{\frac{n}{2^j}-1}$, where $\boldsymbol{\psi}_{j,k,\ell} \in \mathbb{R}^n$.
- A family of dyadic subintervals $\mathscr{I}$ is said to be a *disjoint cover* of I if $\bigcup_{I_{j,k} \in \mathscr{I}} I_{j,k} = I$ and $I_{j,k} \cap I_{j',k'} = \emptyset$ for $(j,k) \neq (j',k')$.
- (Coifman & Wickerhauser 1992): If $\mathscr{I}$ is a disjoint cover of I, then the collection of basis vectors $\{\boldsymbol{\psi}_{j,k,\ell}\}$ where (j,k) are chosen such that $I_{j,k} \in \mathscr{I}$, and $\ell = 0, \ldots, \frac{n}{2^j} - 1$, form an ONB of $\Omega_{0,0}$.
- Hence, every disjoint dyadic cover of I corresponds to an ONB for $\Omega_{0,0}$. Then, the number of possible ONBs in this binary tree is equal to the number of possible disjoint dyadic covers of I. How many such covers?

How Many ONBs in a Dictionary?

- We can associate $\Omega_{j,k}$ as the dyadic interval $I_{j,k} := \left[\frac{k}{2^j}, \frac{k+1}{2^j}\right) \subset [0,1) =: I,\ j = 0,\ldots,J,\ k = 0,\ldots,2^j - 1.$
- Let the orthonormal basis of $\Omega_{j,k}$ generated by these hierarchical operations be $\{\boldsymbol{\psi}_{j,k,\ell}\}_{\ell=0}^{\frac{n}{2^j}-1}$, where $\boldsymbol{\psi}_{j,k,\ell} \in \mathbb{R}^n$.
- A family of dyadic subintervals $\mathscr{I}$ is said to be a *disjoint cover* of I if $\bigcup_{I_{j,k} \in \mathscr{I}} I_{j,k} = I$ and $I_{j,k} \cap I_{j',k'} = \emptyset$ for $(j,k) \neq (j',k')$.
- (Coifman & Wickerhauser 1992): If $\mathscr{I}$ is a disjoint cover of I, then the collection of basis vectors $\{\boldsymbol{\psi}_{j,k,\ell}\}$ where (j,k) are chosen such that $I_{j,k} \in \mathscr{I}$, and $\ell = 0,\ldots,\frac{n}{2^j} - 1$, form an ONB of $\Omega_{0,0}$.
- Hence, every disjoint dyadic cover of I corresponds to an ONB for $\Omega_{0,0}$. Then, the number of possible ONBs in this binary tree is equal to the number of possible disjoint dyadic covers of I. How many such covers?

How Many ONBs in a Dictionary?

- We can associate $\Omega_{j,k}$ as the dyadic interval $I_{j,k} := \left[\frac{k}{2^j}, \frac{k+1}{2^j}\right) \subset [0,1) =: I,\ j = 0, \ldots, J,\ k = 0, \ldots, 2^j - 1.$
- Let the orthonormal basis of $\Omega_{j,k}$ generated by these hierarchical operations be $\{\boldsymbol{\psi}_{j,k,\ell}\}_{\ell=0}^{\frac{n}{2^j}-1}$, where $\boldsymbol{\psi}_{j,k,\ell} \in \mathbb{R}^n$.
- A family of dyadic subintervals $\mathscr{I}$ is said to be a *disjoint cover* of I if $\bigcup_{I_{j,k} \in \mathscr{I}} I_{j,k} = I$ and $I_{j,k} \cap I_{j',k'} = \emptyset$ for $(j,k) \neq (j',k')$.
- (Coifman & Wickerhauser 1992): If $\mathscr{I}$ is a disjoint cover of I, then the collection of basis vectors $\{\boldsymbol{\psi}_{j,k,\ell}\}$ where (j,k) are chosen such that $I_{j,k} \in \mathscr{I}$, and $\ell = 0, \ldots, \frac{n}{2^j} - 1$, form an ONB of $\Omega_{0,0}$.
- Hence, every disjoint dyadic cover of I corresponds to an ONB for $\Omega_{0,0}$. Then, the number of possible ONBs in this binary tree is equal to the number of possible disjoint dyadic covers of I. How many such covers?

How Many ONBs in a Dictionary?

- We can associate $\Omega_{j,k}$ as the dyadic interval $I_{j,k} := \left[\frac{k}{2^j}, \frac{k+1}{2^j}\right) \subset [0,1) =: I,\ j = 0,\ldots,J,\ k = 0,\ldots,2^j - 1.$
- Let the orthonormal basis of $\Omega_{j,k}$ generated by these hierarchical operations be $\{\boldsymbol{\psi}_{j,k,\ell}\}_{\ell=0}^{\frac{n}{2^j}-1}$, where $\boldsymbol{\psi}_{j,k,\ell} \in \mathbb{R}^n$.
- A family of dyadic subintervals $\mathscr{I}$ is said to be a *disjoint cover* of I if $\bigcup_{I_{j,k}\in\mathscr{I}} I_{j,k} = I$ and $I_{j,k} \cap I_{j',k'} = \emptyset$ for $(j,k) \neq (j',k')$.
- (Coifman & Wickerhauser 1992): If $\mathscr{I}$ is a disjoint cover of I, then the collection of basis vectors $\{\boldsymbol{\psi}_{j,k,\ell}\}$ where (j,k) are chosen such that $I_{j,k} \in \mathscr{I}$, and $\ell = 0,\ldots,\frac{n}{2^j} - 1$, form an ONB of $\Omega_{0,0}$.
- Hence, every disjoint dyadic cover of I corresponds to an ONB for $\Omega_{0,0}$. Then, the number of possible ONBs in this binary tree is equal to the number of possible disjoint dyadic covers of I. How many such covers?

How Many ONBs in a Dictionary?

- We can associate $\Omega_{j,k}$ as the dyadic interval $I_{j,k} := \left[\frac{k}{2^j}, \frac{k+1}{2^j}\right) \subset [0,1) =: I,\ j = 0,\ldots,J,\ k = 0,\ldots,2^j - 1.$
- Let the orthonormal basis of $\Omega_{j,k}$ generated by these hierarchical operations be $\{\boldsymbol{\psi}_{j,k,\ell}\}_{\ell=0}^{\frac{n}{2^j}-1}$, where $\boldsymbol{\psi}_{j,k,\ell} \in \mathbb{R}^n$.
- A family of dyadic subintervals $\mathscr{I}$ is said to be a *disjoint cover* of I if $\bigcup_{I_{j,k} \in \mathscr{I}} I_{j,k} = I$ and $I_{j,k} \cap I_{j',k'} = \emptyset$ for $(j,k) \neq (j',k')$.
- (Coifman & Wickerhauser 1992): If $\mathscr{I}$ is a disjoint cover of I, then the collection of basis vectors $\{\boldsymbol{\psi}_{j,k,\ell}\}$ where (j,k) are chosen such that $I_{j,k} \in \mathscr{I}$, and $\ell = 0,\ldots,\frac{n}{2^j} - 1$, form an ONB of $\Omega_{0,0}$.
- Hence, every disjoint dyadic cover of I corresponds to an ONB for $\Omega_{0,0}$. Then, the number of possible ONBs in this binary tree is equal to the number of possible disjoint dyadic covers of I. How many such covers?

How Many ONBs in a Dictionary?

- Let J be the deepest level of decomposition so that the j ranges $0 \le j \le J$ (i.e., $J+1$ levels).
- Let A_J be the number of ONBs in the tree with J+1 levels.
- We do induction on J for A_J. Clearly $A_0 = 1$, $A_1 = 2$.
- Let's relate A_{J+1} and A_J. Consider the binary tree of $J+2$ levels. Then, $\Omega_{1,0}$ and below forms a binary tree with J+1 levels so as $\Omega_{1,1}$ and below. By assumption, there are A_J ONBs for each $\Omega_{1,k}$, $k = 0, 1$. Hence, we have

$$A_{J+1} = 1 + A_J^2,$$

  where 1 comes from choosing the canonical basis at $\Omega_{0,0}$.
- One can show that $A_J > 2^{n/2}$.
- This sequence is cataloged as A003095 in the *On-line Encyclopedia of Integer Sequences* by Neil J. A. Sloane. For example, A_J for $J = 0, 1, 2, 3, 4, 5, 6, 7 \ldots$, are: $1, 2, 5, 26, 677, 458330, 210066388901, \ldots$

How Many ONBs in a Dictionary?

- Let J be the deepest level of decomposition so that the j ranges $0 \le j \le J$ (i.e., $J+1$ levels).
- Let A_J be the number of ONBs in the tree with J+1 levels.
- We do induction on J for A_J. Clearly $A_0 = 1$, $A_1 = 2$.
- Let's relate A_{J+1} and A_J. Consider the binary tree of $J+2$ levels. Then, $\Omega_{1,0}$ and below forms a binary tree with J+1 levels so as $\Omega_{1,1}$ and below. By assumption, there are A_J ONBs for each $\Omega_{1,k}$, $k = 0,1$. Hence, we have

$$A_{J+1} = 1 + A_J^2,$$

  where 1 comes from choosing the canonical basis at $\Omega_{0,0}$.
- One can show that $A_J > 2^{n/2}$.
- This sequence is cataloged as A003095 in the *On-line Encyclopedia of Integer Sequences* by Neil J. A. Sloane. For example, A_J for $J = 0,1,2,3,4,5,6,7\ldots$, are: $1,2,5,26,677,458330,210066388901,\ldots$

How Many ONBs in a Dictionary?

- Let J be the deepest level of decomposition so that the j ranges $0 \leq j \leq J$ (i.e., $J+1$ levels).
- Let A_J be the number of ONBs in the tree with J+1 levels.
- We do induction on J for A_J. Clearly $A_0 = 1$, $A_1 = 2$.
- Let's relate A_{J+1} and A_J. Consider the binary tree of $J+2$ levels. Then, $\Omega_{1,0}$ and below forms a binary tree with J+1 levels so as $\Omega_{1,1}$ and below. By assumption, there are A_J ONBs for each $\Omega_{1,k}$, $k = 0,1$. Hence, we have

$$A_{J+1} = 1 + A_J^2,$$

  where 1 comes from choosing the canonical basis at $\Omega_{0,0}$.
- One can show that $A_J > 2^{n/2}$.
- This sequence is cataloged as A003095 in the *On-line Encyclopedia of Integer Sequences* by Neil J. A. Sloane. For example, A_J for $J = 0,1,2,3,4,5,6,7\ldots$, are: $1,2,5,26,677,458330,210066388901,\ldots$

How Many ONBs in a Dictionary?

- Let J be the deepest level of decomposition so that the j ranges $0 \le j \le J$ (i.e., $J+1$ levels).
- Let A_J be the number of ONBs in the tree with J+1 levels.
- We do induction on J for A_J. Clearly $A_0 = 1$, $A_1 = 2$.
- Let's relate A_{J+1} and A_J. Consider the binary tree of $J+2$ levels. Then, $\Omega_{1,0}$ and below forms a binary tree with J+1 levels so as $\Omega_{1,1}$ and below. By assumption, there are A_J ONBs for each $\Omega_{1,k}$, $k = 0, 1$. Hence, we have

$$A_{J+1} = 1 + A_J^2,$$

  where 1 comes from choosing the canonical basis at $\Omega_{0,0}$.
- One can show that $A_J > 2^{n/2}$.
- This sequence is cataloged as A003095 in the *On-line Encyclopedia of Integer Sequences* by Neil J. A. Sloane. For example, A_J for $J = 0, 1, 2, 3, 4, 5, 6, 7 \ldots$, are: $1, 2, 5, 26, 677, 458330, 210066388901, \ldots$

How Many ONBs in a Dictionary?

- Let J be the deepest level of decomposition so that the j ranges $0 \le j \le J$ (i.e., $J+1$ levels).
- Let A_J be the number of ONBs in the tree with J+1 levels.
- We do induction on J for A_J. Clearly $A_0 = 1$, $A_1 = 2$.
- Let's relate A_{J+1} and A_J. Consider the binary tree of $J+2$ levels. Then, $\Omega_{1,0}$ and below forms a binary tree with J+1 levels so as $\Omega_{1,1}$ and below. By assumption, there are A_J ONBs for each $\Omega_{1,k}$, $k = 0, 1$. Hence, we have

$$A_{J+1} = 1 + A_J^2,$$

  where 1 comes from choosing the canonical basis at $\Omega_{0,0}$.
- One can show that $A_J > 2^{n/2}$.
- This sequence is cataloged as A003095 in the *On-line Encyclopedia of Integer Sequences* by Neil J. A. Sloane. For example, A_J for $J = 0, 1, 2, 3, 4, 5, 6, 7 \ldots$, are: $1, 2, 5, 26, 677, 458330, 210066388901, \ldots$

How Many ONBs in a Dictionary?

- Let J be the deepest level of decomposition so that the j ranges $0 \le j \le J$ (i.e., $J+1$ levels).
- Let A_J be the number of ONBs in the tree with J+1 levels.
- We do induction on J for A_J. Clearly $A_0 = 1$, $A_1 = 2$.
- Let's relate A_{J+1} and A_J. Consider the binary tree of $J+2$ levels. Then, $\Omega_{1,0}$ and below forms a binary tree with J+1 levels so as $\Omega_{1,1}$ and below. By assumption, there are A_J ONBs for each $\Omega_{1,k}$, $k = 0, 1$. Hence, we have

$$A_{J+1} = 1 + A_J^2,$$

  where 1 comes from choosing the canonical basis at $\Omega_{0,0}$.
- One can show that $A_J > 2^{n/2}$.
- This sequence is cataloged as A003095 in the *On-line Encyclopedia of Integer Sequences* by Neil J. A. Sloane. For example, A_J for $J = 0, 1, 2, 3, 4, 5, 6, 7 \ldots$, are: $1, 2, 5, 26, 677, 458330, 210066388901, \ldots$

Outline

- Let $\mathscr{D}$ be a orthonormal basis dictionary, i.e., the collection of all the basis vectors in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0 \le j \le J; 0 \le k \le 2^j - 1}$, which can be written as $\mathscr{D} := \{\boldsymbol{\psi}_{j,k,\ell}\}_{0 \le j \le J; 0 \le k \le 2^j - 1; 0 \le \ell \le \frac{n}{2^j} - 1}$.
- In this dictionary, there are nJ basis vectors.
- $\mathscr{D}$ can also be written as $\mathscr{D} = \{B_i\}_{1 \le i \le A_J}$, where $B_i \in \mathbb{R}^{n \times n}$ is an ONB contained in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0 \le j \le J; 0 \le k \le 2^j - 1}$, and A_J is the number of possible ONBs contained in this binary tree, which could be *huge*, certainly, $A_J > 2^{n/2}$.
- Let $\mathscr{M}(B_i)$ be a *measure of efficacy* of B_i w.r.t. a given data matrix X for a task given at hand.
- Then the *best basis* w.r.t. $\mathscr{M}$ among $\mathscr{D}$ for X is:

$$\Psi = \Psi(X; \mathscr{D}) = \arg\max_{B_i \in \mathscr{D}} \mathscr{M}\left(B_i^{\mathsf{T}} X\right).$$

- Finally, if a library $\mathscr{L}$ consists of multiple dictionaries, then the overall best basis can be obtained by

$$\Psi(X; \mathscr{L}) = \arg\max_{\mathscr{D} \in \mathscr{L}} \mathscr{M}\left(\Psi(X; \mathscr{D})^{\mathsf{T}} X\right).$$

- Let $\mathscr{D}$ be a orthonormal basis dictionary, i.e., the collection of all the basis vectors in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0\le j\le J; 0\le k\le 2^j-1}$, which can be written as $\mathscr{D} := \{\boldsymbol{\psi}_{j,k,\ell}\}_{0\le j\le J; 0\le k\le 2^j-1; 0\le \ell\le \frac{n}{2^j}-1}$.
- In this dictionary, there are nJ basis vectors.
- $\mathscr{D}$ can also be written as $\mathscr{D} = \{B_i\}_{1\le i\le A_J}$, where $B_i \in \mathbb{R}^{n\times n}$ is an ONB contained in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0\le j\le J; 0\le k\le 2^j-1}$, and A_J is the number of possible ONBs contained in this binary tree, which could be *huge*, certainly, $A_J > 2^{n/2}$.
- Let $\mathscr{M}(B_i)$ be a *measure of efficacy* of B_i w.r.t. a given data matrix X for a task given at hand.
- Then the *best basis* w.r.t. $\mathscr{M}$ among $\mathscr{D}$ for X is:

$$\Psi = \Psi(X;\mathscr{D}) = \arg\max_{B_i\in\mathscr{D}} \mathscr{M}\left(B_i^{\mathsf{T}} X\right).$$

- Finally, if a library $\mathscr{L}$ consists of multiple dictionaries, then the overall best basis can be obtained by

$$\Psi(X;\mathscr{L}) = \arg\max_{\mathscr{D}\in\mathscr{L}} \mathscr{M}\left(\Psi(X;\mathscr{D})^{\mathsf{T}} X\right).$$

- Let $\mathscr{D}$ be a orthonormal basis dictionary, i.e., the collection of all the basis vectors in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0\le j\le J;0\le k\le 2^j-1}$, which can be written as $\mathscr{D} := \{\boldsymbol{\psi}_{j,k,\ell}\}_{0\le j\le J;0\le k\le 2^j-1;0\le \ell\le \frac{n}{2^j}-1}$.
- In this dictionary, there are nJ basis vectors.
- $\mathscr{D}$ can also be written as $\mathscr{D} = \{B_i\}_{1\le i\le A_J}$, where $B_i \in \mathbb{R}^{n\times n}$ is an ONB contained in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0\le j\le J;0\le k\le 2^j-1}$, and A_J is the number of possible ONBs contained in this binary tree, which could be *huge*, certainly, $A_J > 2^{n/2}$.
- Let $\mathscr{M}(B_i)$ be a *measure of efficacy* of B_i w.r.t. a given data matrix X for a task given at hand.
- Then the *best basis* w.r.t. $\mathscr{M}$ among $\mathscr{D}$ for X is:

$$\Psi = \Psi(X;\mathscr{D}) = \arg\max_{B_i\in\mathscr{D}} \mathscr{M}\left(B_i^{\top}X\right).$$

- Finally, if a library $\mathscr{L}$ consists of multiple dictionaries, then the overall best basis can be obtained by

$$\Psi(X;\mathscr{L}) = \arg\max_{\mathscr{D}\in\mathscr{L}} \mathscr{M}\left(\Psi(X;\mathscr{D})^{\top}X\right).$$

- Let $\mathscr{D}$ be a orthonormal basis dictionary, i.e., the collection of all the basis vectors in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0\le j\le J; 0\le k\le 2^j-1}$, which can be written as $\mathscr{D} := \{\boldsymbol{\psi}_{j,k,\ell}\}_{0\le j\le J; 0\le k\le 2^j-1; 0\le \ell\le \frac{n}{2^j}-1}$.
- In this dictionary, there are nJ basis vectors.
- $\mathscr{D}$ can also be written as $\mathscr{D} = \{B_i\}_{1\le i\le A_J}$, where $B_i \in \mathbb{R}^{n\times n}$ is an ONB contained in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0\le j\le J; 0\le k\le 2^j-1}$, and A_J is the number of possible ONBs contained in this binary tree, which could be *huge*, certainly, $A_J > 2^{n/2}$.
- Let $\mathscr{M}(B_i)$ be a *measure of efficacy* of B_i w.r.t. a given data matrix X for a task given at hand.
- Then the *best basis* w.r.t. $\mathscr{M}$ among $\mathscr{D}$ for X is:

$$\Psi = \Psi(X;\mathscr{D}) = \arg\max_{B_i\in\mathscr{D}} \mathscr{M}\left(B_i^{\top} X\right).$$

- Finally, if a library $\mathscr{L}$ consists of multiple dictionaries, then the overall best basis can be obtained by

$$\Psi(X;\mathscr{L}) = \arg\max_{\mathscr{D}\in\mathscr{L}} \mathscr{M}\left(\Psi(X;\mathscr{D})^{\top} X\right).$$

- Let $\mathscr{D}$ be a orthonormal basis dictionary, i.e., the collection of all the basis vectors in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0\le j\le J;0\le k\le 2^j-1}$, which can be written as $\mathscr{D} := \{\boldsymbol{\psi}_{j,k,\ell}\}_{0\le j\le J;0\le k\le 2^j-1;0\le \ell\le \frac{n}{2^j}-1}$.
- In this dictionary, there are nJ basis vectors.
- $\mathscr{D}$ can also be written as $\mathscr{D} = \{B_i\}_{1\le i\le A_J}$, where $B_i \in \mathbb{R}^{n\times n}$ is an ONB contained in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0\le j\le J;0\le k\le 2^j-1}$, and A_J is the number of possible ONBs contained in this binary tree, which could be *huge*, certainly, $A_J > 2^{n/2}$.
- Let $\mathscr{M}(B_i)$ be a *measure of efficacy* of B_i w.r.t. a given data matrix X for a task given at hand.
- Then the *best basis* w.r.t. $\mathscr{M}$ among $\mathscr{D}$ for X is:

$$\Psi = \Psi(X;\mathscr{D}) = \arg\max_{B_i\in\mathscr{D}} \mathscr{M}\left(B_i^{\mathsf{T}} X\right).$$

- Finally, if a library $\mathscr{L}$ consists of multiple dictionaries, then the overall best basis can be obtained by

$$\Psi(X;\mathscr{L}) = \arg\max_{\mathscr{D}\in\mathscr{L}} \mathscr{M}\left(\Psi(X;\mathscr{D})^{\mathsf{T}} X\right).$$

- Let $\mathscr{D}$ be a orthonormal basis dictionary, i.e., the collection of all the basis vectors in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0\le j\le J;0\le k\le 2^j-1}$, which can be written as $\mathscr{D} := \{\boldsymbol{\psi}_{j,k,\ell}\}_{0\le j\le J;0\le k\le 2^j-1;0\le \ell\le \frac{n}{2^j}-1}$.
- In this dictionary, there are nJ basis vectors.
- $\mathscr{D}$ can also be written as $\mathscr{D} = \{B_i\}_{1\le i\le A_J}$, where $B_i \in \mathbb{R}^{n\times n}$ is an ONB contained in the binary tree of subspaces of $\{\Omega_{j,k}\}_{0\le j\le J;0\le k\le 2^j-1}$, and A_J is the number of possible ONBs contained in this binary tree, which could be *huge*, certainly, $A_J > 2^{n/2}$.
- Let $\mathscr{M}(B_i)$ be a *measure of efficacy* of B_i w.r.t. a given data matrix X for a task given at hand.
- Then the *best basis* w.r.t. $\mathscr{M}$ among $\mathscr{D}$ for X is:

$$\Psi = \Psi(X;\mathscr{D}) = \arg\max_{B_i\in\mathscr{D}} \mathscr{M}\left(B_i^{\mathsf{T}} X\right).$$

- Finally, if a library $\mathscr{L}$ consists of multiple dictionaries, then the overall best basis can be obtained by

$$\Psi(X;\mathscr{L}) = \arg\max_{\mathscr{D}\in\mathscr{L}} \mathscr{M}\left(\Psi(X;\mathscr{D})^{\mathsf{T}} X\right).$$

Outline

The Best Basis Selection Algorithm

Let $B_{j,k} := \left(\boldsymbol{\psi}_{j,k,0}, \ldots, \boldsymbol{\psi}_{j,k,\frac{n}{2^j}-1}\right) \in \mathbb{R}^{n \times \frac{n}{2^j}}$ be the ONB of $\Omega_{j,k}$. Then, *the best basis algorithm* of Coifman-Wickerhauser (1992) proceeds as follows:

Step 0: Choose a dictionary of orthonormal bases $\mathcal{D}$ (i.e., specify the CMF for a wavelet packet dictionary or decide to use either the BDCT dictionary or the LCT dictionary) and specify the maximum depth of decomposition J and the measure of efficacy $\mathcal{M}$.

Step 1: Expand the columns of the data matrix X, into the dictionary $\mathcal{D}$ and obtain coefficients $\left\{B_{j,k}^{\mathsf{T}} X\right\}_{0 \le j \le J; 0 \le k \le 2^j - 1}$.

Step 2: Set $\Psi_{J,k} := B_{J,k}$ for $k = 0, \ldots, 2^J - 1$ (i.e., start from the *bottom*)

Step 3: Determine the best subspace basis $\Psi_{j,k}$ for $j = J-1, \ldots, 0$, $k = 0, \ldots, 2^j - 1$ (i.e., from bottom to top) by

$$\Psi_{j,k} = \begin{cases} B_{j,k} & \text{if } \mathcal{M}\left(B_{j,k}^{\mathsf{T}} X\right) \ge \mathcal{M}\left(\Psi_{j+1,2k}^{\mathsf{T}} X \cup \Psi_{j+1,2k+1}^{\mathsf{T}} X\right), \\ \Psi_{j+1,2k} \oplus \Psi_{j+1,2k+1} & \text{otherwise.} \end{cases} \quad (1)$$

The Best Basis Selection Algorithm

Let $B_{j,k} := \left(\boldsymbol{\psi}_{j,k,0}, \ldots, \boldsymbol{\psi}_{j,k,\frac{n}{2^j}-1}\right) \in \mathbb{R}^{n \times \frac{n}{2^j}}$ be the ONB of $\Omega_{j,k}$. Then, *the best basis algorithm* of Coifman-Wickerhauser (1992) proceeds as follows:

Step 0: Choose a dictionary of orthonormal bases $\mathcal{D}$ (i.e., specify the CMF for a wavelet packet dictionary or decide to use either the BDCT dictionary or the LCT dictionary) and specify the maximum depth of decomposition J and the measure of efficacy $\mathscr{M}$.

Step 1: Expand the columns of the data matrix X, into the dictionary $\mathcal{D}$ and obtain coefficients $\left\{B^{\mathsf{T}}_{j,k} X\right\}_{0 \le j \le J; 0 \le k \le 2^j - 1}$.

Step 2: Set $\Psi_{J,k} := B_{J,k}$ for $k = 0, \ldots, 2^J - 1$ (i.e., start from the *bottom*)

Step 3: Determine the best subspace basis $\Psi_{j,k}$ for $j = J-1, \ldots, 0,\ k = 0, \ldots, 2^j - 1$ (i.e., from bottom to top) by

$$\Psi_{j,k} = \begin{cases} B_{j,k} & \text{if } \mathscr{M}\left(B^{\mathsf{T}}_{j,k} X\right) \ge \mathscr{M}\left(\Psi^{\mathsf{T}}_{j+1,2k} X \cup \Psi^{\mathsf{T}}_{j+1,2k+1} X\right), \\ \Psi_{j+1,2k} \oplus \Psi_{j+1,2k+1} & \text{otherwise.} \end{cases} \quad (1)$$

The Best Basis Selection Algorithm

Let $B_{j,k} := \left(\boldsymbol{\psi}_{j,k,0}, \ldots, \boldsymbol{\psi}_{j,k,\frac{n}{2^j}-1}\right) \in \mathbb{R}^{n \times \frac{n}{2^j}}$ be the ONB of $\Omega_{j,k}$. Then, *the best basis algorithm* of Coifman-Wickerhauser (1992) proceeds as follows:

Step 0: Choose a dictionary of orthonormal bases $\mathcal{D}$ (i.e., specify the CMF for a wavelet packet dictionary or decide to use either the BDCT dictionary or the LCT dictionary) and specify the maximum depth of decomposition J and the measure of efficacy $\mathscr{M}$.

Step 1: Expand the columns of the data matrix X, into the dictionary $\mathcal{D}$ and obtain coefficients $\left\{B^{\mathsf{T}}_{j,k} X\right\}_{0 \le j \le J; 0 \le k \le 2^j - 1}$.

Step 2: Set $\Psi_{J,k} := B_{J,k}$ for $k = 0, \ldots, 2^J - 1$ (i.e., start from the *bottom*)

Step 3: Determine the best subspace basis $\Psi_{j,k}$ for $j = J-1, \ldots, 0$, $k = 0, \ldots, 2^j - 1$ (i.e., from bottom to top) by

$$\Psi_{j,k} = \begin{cases} B_{j,k} & \text{if } \mathscr{M}\left(B^{\mathsf{T}}_{j,k} X\right) \ge \mathscr{M}\left(\Psi^{\mathsf{T}}_{j+1,2k} X \cup \Psi^{\mathsf{T}}_{j+1,2k+1} X\right), \\ \Psi_{j+1,2k} \oplus \Psi_{j+1,2k+1} & \text{otherwise.} \end{cases} \quad (1)$$

The Best Basis Selection Algorithm

Let $B_{j,k} := \left(\boldsymbol{\psi}_{j,k,0}, \ldots, \boldsymbol{\psi}_{j,k,\frac{n}{2^j}-1}\right) \in \mathbb{R}^{n \times \frac{n}{2^j}}$ be the ONB of $\Omega_{j,k}$. Then, *the best basis algorithm* of Coifman-Wickerhauser (1992) proceeds as follows:

Step 0: Choose a dictionary of orthonormal bases $\mathcal{D}$ (i.e., specify the CMF for a wavelet packet dictionary or decide to use either the BDCT dictionary or the LCT dictionary) and specify the maximum depth of decomposition J and the measure of efficacy $\mathscr{M}$.

Step 1: Expand the columns of the data matrix X, into the dictionary $\mathcal{D}$ and obtain coefficients $\left\{B_{j,k}^{\mathsf{T}} X\right\}_{0 \le j \le J; 0 \le k \le 2^j - 1}$.

Step 2: Set $\Psi_{J,k} := B_{J,k}$ for $k = 0, \ldots, 2^J - 1$ (i.e., start from the *bottom*)

Step 3: Determine the best subspace basis $\Psi_{j,k}$ for $j = J-1, \ldots, 0,\ k = 0, \ldots, 2^j - 1$ (i.e., from bottom to top) by

$$\Psi_{j,k} = \begin{cases} B_{j,k} & \text{if } \mathscr{M}\left(B_{j,k}^{\mathsf{T}} X\right) \ge \mathscr{M}\left(\Psi_{j+1,2k}^{\mathsf{T}} X \cup \Psi_{j+1,2k+1}^{\mathsf{T}} X\right), \\ \Psi_{j+1,2k} \oplus \Psi_{j+1,2k+1} & \text{otherwise.} \end{cases} \tag{1}$$

The Best Basis Selection Algorithm

Let $B_{j,k} := \left(\boldsymbol{\psi}_{j,k,0}, \ldots, \boldsymbol{\psi}_{j,k,\frac{n}{2^j}-1}\right) \in \mathbb{R}^{n \times \frac{n}{2^j}}$ be the ONB of $\Omega_{j,k}$. Then, *the best basis algorithm* of Coifman-Wickerhauser (1992) proceeds as follows:

- Step 0: Choose a dictionary of orthonormal bases $\mathcal{D}$ (i.e., specify the CMF for a wavelet packet dictionary or decide to use either the BDCT dictionary or the LCT dictionary) and specify the maximum depth of decomposition J and the measure of efficacy $\mathscr{M}$.
- Step 1: Expand the columns of the data matrix X, into the dictionary $\mathcal{D}$ and obtain coefficients $\left\{B_{j,k}^{\mathsf{T}} X\right\}_{0 \le j \le J; 0 \le k \le 2^j - 1}$.
- Step 2: Set $\Psi_{J,k} := B_{J,k}$ for $k = 0, \ldots, 2^J - 1$ (i.e., start from the *bottom*)
- Step 3: Determine the best subspace basis $\Psi_{j,k}$ for $j = J-1, \ldots, 0,\ k = 0, \ldots, 2^j - 1$ (i.e., from bottom to top) by

$$\Psi_{j,k} = \begin{cases} B_{j,k} & \text{if } \mathscr{M}\left(B_{j,k}^{\mathsf{T}} X\right) \ge \mathscr{M}\left(\Psi_{j+1,2k}^{\mathsf{T}} X \cup \Psi_{j+1,2k+1}^{\mathsf{T}} X\right), \\ \Psi_{j+1,2k} \oplus \Psi_{j+1,2k+1} & \text{otherwise.} \end{cases} \tag{1}$$

- To make this algorithm fast, the measure of efficacy $\mathscr{M}$ should be *additive*:

Definition

A map $\mathscr{M}$ from sequences $\{x_i\}$ to $\mathbb{R}$ is said to be *additive* if $\mathscr{M}(0) = 0$ and $\mathscr{M}(\{x_i\}) = \sum_i \mathscr{M}(x_i)$.

- Thus, if $\mathscr{M}$ is additive, then Eq. (1) simplifies to

$$\mathscr{M}\left(\Psi^{\mathsf{T}}_{j+1,2k}X \cup \Psi^{\mathsf{T}}_{j+1,2k+1}X\right) = \mathscr{M}\left(\Psi^{\mathsf{T}}_{j+1,2k}X\right) + \mathscr{M}\left(\Psi^{\mathsf{T}}_{j+1,2k+1}X\right).$$

- This implies that a simple addition suffices instead of computing the efficacy of the union of the nodes.
- In fact, the cost of selecting the best basis Ψ for an additive measure $\mathscr{M}$ given all the expansion coefficients of X in $\mathscr{D}$ is $O(n)$ while the cost of expanding all the columns of X into $\mathscr{D}$ costs at most $O(Nn[\log_2 n]^p)$, $p = 1$ for a wavelet packet dictionary and $p = 2$ for the BDCT/LCT dictionaries.

- To make this algorithm fast, the measure of efficacy $\mathscr{M}$ should be *additive*:

Definition

A map $\mathscr{M}$ from sequences $\{x_i\}$ to $\mathbb{R}$ is said to be *additive* if $\mathscr{M}(0) = 0$ and $\mathscr{M}(\{x_i\}) = \sum_i \mathscr{M}(x_i)$.

- Thus, if $\mathscr{M}$ is additive, then Eq. (1) simplifies to

$$\mathscr{M}\left(\Psi_{j+1,2k}^{\mathsf{T}} X \cup \Psi_{j+1,2k+1}^{\mathsf{T}} X\right) = \mathscr{M}\left(\Psi_{j+1,2k}^{\mathsf{T}} X\right) + \mathscr{M}\left(\Psi_{j+1,2k+1}^{\mathsf{T}} X\right).$$

- This implies that a simple addition suffices instead of computing the efficacy of the union of the nodes.
- In fact, the cost of selecting the best basis Ψ for an additive measure $\mathscr{M}$ given all the expansion coefficients of X in $\mathscr{D}$ is $O(n)$ while the cost of expanding all the columns of X into $\mathscr{D}$ costs at most $O(Nn[\log_2 n]^p)$, $p = 1$ for a wavelet packet dictionary and $p = 2$ for the BDCT/LCT dictionaries.

- To make this algorithm fast, the measure of efficacy $\mathscr{M}$ should be *additive*:

Definition

A map $\mathscr{M}$ from sequences $\{x_i\}$ to $\mathbb{R}$ is said to be *additive* if $\mathscr{M}(0) = 0$ and $\mathscr{M}(\{x_i\}) = \sum_i \mathscr{M}(x_i)$.

- Thus, if $\mathscr{M}$ is additive, then Eq. (1) simplifies to

$$\mathscr{M}\left(\Psi_{j+1,2k}^{\mathsf{T}} X \cup \Psi_{j+1,2k+1}^{\mathsf{T}} X\right) = \mathscr{M}\left(\Psi_{j+1,2k}^{\mathsf{T}} X\right) + \mathscr{M}\left(\Psi_{j+1,2k+1}^{\mathsf{T}} X\right).$$

- This implies that a simple addition suffices instead of computing the efficacy of the union of the nodes.
- In fact, the cost of selecting the best basis Ψ for an additive measure $\mathscr{M}$ given all the expansion coefficients of X in $\mathscr{D}$ is $O(n)$ while the cost of expanding all the columns of X into $\mathscr{D}$ costs at most $O(Nn[\log_2 n]^p)$, $p = 1$ for a wavelet packet dictionary and $p = 2$ for the BDCT/LCT dictionaries.

- To make this algorithm fast, the measure of efficacy $\mathscr{M}$ should be *additive*:

Definition

A map $\mathscr{M}$ from sequences $\{x_i\}$ to $\mathbb{R}$ is said to be *additive* if $\mathscr{M}(0) = 0$ and $\mathscr{M}(\{x_i\}) = \sum_i \mathscr{M}(x_i)$.

- Thus, if $\mathscr{M}$ is additive, then Eq. (1) simplifies to

$$\mathscr{M}\left(\Psi_{j+1,2k}^{\mathsf{T}} X \cup \Psi_{j+1,2k+1}^{\mathsf{T}} X\right) = \mathscr{M}\left(\Psi_{j+1,2k}^{\mathsf{T}} X\right) + \mathscr{M}\left(\Psi_{j+1,2k+1}^{\mathsf{T}} X\right).$$

- This implies that a simple addition suffices instead of computing the efficacy of the union of the nodes.
- In fact, the cost of selecting the best basis Ψ for an additive measure $\mathscr{M}$ given all the expansion coefficients of X in $\mathscr{D}$ is $O(n)$ while the cost of expanding all the columns of X into $\mathscr{D}$ costs at most $O(Nn[\log_2 n]^p)$, $p = 1$ for a wavelet packet dictionary and $p = 2$ for the BDCT/LCT dictionaries.

- To make this algorithm fast, the measure of efficacy $\mathscr{M}$ should be *additive*:

Definition

A map $\mathscr{M}$ from sequences $\{x_i\}$ to $\mathbb{R}$ is said to be *additive* if $\mathscr{M}(0) = 0$ and $\mathscr{M}(\{x_i\}) = \sum_i \mathscr{M}(x_i)$.

- Thus, if $\mathscr{M}$ is additive, then Eq. (1) simplifies to

$$\mathscr{M}\left(\Psi^{\mathsf{T}}_{j+1,2k}X \cup \Psi^{\mathsf{T}}_{j+1,2k+1}X\right) = \mathscr{M}\left(\Psi^{\mathsf{T}}_{j+1,2k}X\right) + \mathscr{M}\left(\Psi^{\mathsf{T}}_{j+1,2k+1}X\right).$$

- This implies that a simple addition suffices instead of computing the efficacy of the union of the nodes.
- In fact, the cost of selecting the best basis Ψ for an additive measure $\mathscr{M}$ given all the expansion coefficients of X in $\mathscr{D}$ is $O(n)$ while the cost of expanding all the columns of X into $\mathscr{D}$ costs at most $O(Nn[\log_2 n]^p)$, $p = 1$ for a wavelet packet dictionary and $p = 2$ for the BDCT/LCT dictionaries.

- Now, the name of the game is how to define $\mathscr{M}$ for a given task.
- For example, in the case of efficient approximation, one may want to find a basis among $\mathscr{D}$ that most *sparsifies* the data matrix X on average.
- In that case, a possible choice of $\mathscr{M}$ is:

$$\mathscr{M}\left(B_{j,k}^{\mathsf{T}} X\right) = -\frac{1}{N} \sum_{m=1}^{N} \left\| B_{j,k}^{\mathsf{T}} \boldsymbol{x}_m \right\|_p^p, \quad 0 < p \leq 1.$$

  negative of the average sparsity of X measured by the ℓ^p-norm.
- There are many many possibilities of $\mathscr{M}$ depending on the task such as clustering, classification, regression, ... See [4], [5] for more examples.

- Now, the name of the game is how to define $\mathscr{M}$ for a given task.
- For example, in the case of efficient approximation, one may want to find a basis among $\mathscr{D}$ that most *sparsifies* the data matrix X on average.
- In that case, a possible choice of $\mathscr{M}$ is:

$$\mathscr{M}\left(B_{j,k}^{\mathsf{T}} X\right) = -\frac{1}{N}\sum_{m=1}^{N}\left\| B_{j,k}^{\mathsf{T}} \boldsymbol{x}_m \right\|_p^p, \quad 0 < p \leq 1.$$

  negative of the average sparsity of X measured by the ℓ^p-norm.
- There are many many possibilities of $\mathscr{M}$ depending on the task such as clustering, classification, regression, ... See [4], [5] for more examples.

- Now, the name of the game is how to define $\mathscr{M}$ for a given task.
- For example, in the case of efficient approximation, one may want to find a basis among $\mathscr{D}$ that most *sparsifies* the data matrix X on average.
- In that case, a possible choice of $\mathscr{M}$ is:

$$\mathscr{M}\left(B_{j,k}^{\mathsf{T}}X\right) = -\frac{1}{N}\sum_{m=1}^{N}\left\|B_{j,k}^{\mathsf{T}}\boldsymbol{x}_m\right\|_p^p, \quad 0 < p \leq 1.$$

negative of the average sparsity of X measured by the ℓ^p-norm.

- There are many many possibilities of $\mathscr{M}$ depending on the task such as clustering, classification, regression, ... See [4], [5] for more examples.

- Now, the name of the game is how to define $\mathscr{M}$ for a given task.
- For example, in the case of efficient approximation, one may want to find a basis among $\mathscr{D}$ that most *sparsifies* the data matrix X on average.
- In that case, a possible choice of $\mathscr{M}$ is:

$$\mathscr{M}\left(B_{j,k}^{\mathsf{T}}X\right) = -\frac{1}{N}\sum_{m=1}^{N}\left\|B_{j,k}^{\mathsf{T}}\boldsymbol{x}_m\right\|_p^p, \quad 0 < p \leq 1.$$

  negative of the average sparsity of X measured by the ℓ^p-norm.
- There are many many possibilities of $\mathscr{M}$ depending on the task such as clustering, classification, regression, ... See [4], [5] for more examples.

Outline

An Example: Efficient Approximation of Geophysical Acoustic Waveforms

- Objective: Efficiently approximate the acoustic waveforms recorded in a borehole propagated through sandstone layers in the subsurface.
- We have 201 such waveforms each of which has $n = 256$ time samples.
- First, randomly split this set of waveforms into the training and test datasets. The training dataset consists of $N = 101$ waveforms while the test dataset contains the remaining 100 waveforms.
- Compare the performance of the global DCT, KLB, and the JBB (Joint Best Basis for the whole training dataset) using the local cosine dictionary.

An Example: Efficient Approximation of Geophysical Acoustic Waveforms

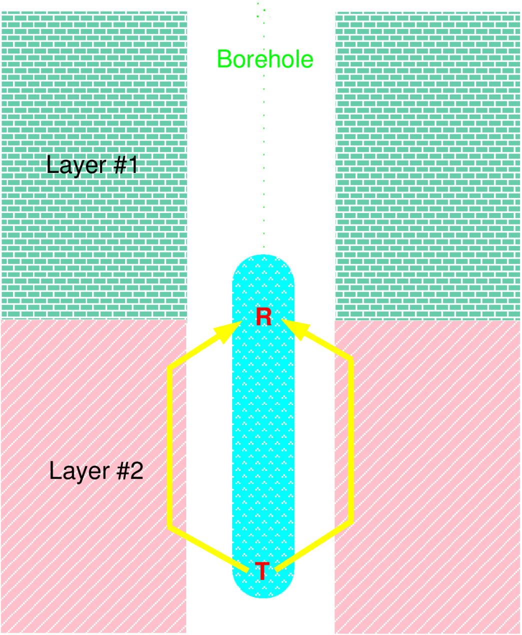
- Objective: Efficiently approximate the acoustic waveforms recorded in a borehole propagated through sandstone layers in the subsurface.
- We have 201 such waveforms each of which has $n = 256$ time samples.
- First, randomly split this set of waveforms into the training and test datasets. The training dataset consists of $N = 101$ waveforms while the test dataset contains the remaining 100 waveforms.
- Compare the performance of the global DCT, KLB, and the JBB (Joint Best Basis for the whole training dataset) using the local cosine dictionary.

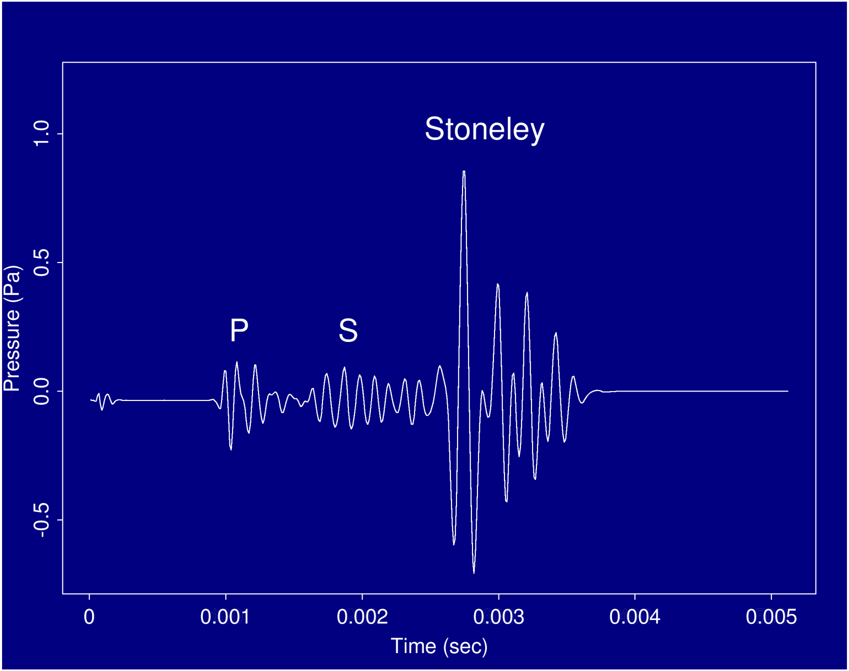
An Example: Efficient Approximation of Geophysical Acoustic Waveforms

- Objective: Efficiently approximate the acoustic waveforms recorded in a borehole propagated through sandstone layers in the subsurface.
- We have 201 such waveforms each of which has $n = 256$ time samples.
- First, randomly split this set of waveforms into the training and test datasets. The training dataset consists of $N = 101$ waveforms while the test dataset contains the remaining 100 waveforms.
- Compare the performance of the global DCT, KLB, and the JBB (Joint Best Basis for the whole training dataset) using the local cosine dictionary.

An Example: Efficient Approximation of Geophysical Acoustic Waveforms

- Objective: Efficiently approximate the acoustic waveforms recorded in a borehole propagated through sandstone layers in the subsurface.
- We have 201 such waveforms each of which has $n = 256$ time samples.
- First, randomly split this set of waveforms into the training and test datasets. The training dataset consists of $N = 101$ waveforms while the test dataset contains the remaining 100 waveforms.
- Compare the performance of the global DCT, KLB, and the JBB (Joint Best Basis for the whole training dataset) using the local cosine dictionary.


Borehole
Layer #1
R
Layer #2
T


Stoneley
P
S
Pressure (Pa)
1.0
0.5
0.0
-0.5
0
0.001
0.002
0.003
0.004
0.005
Time (sec)

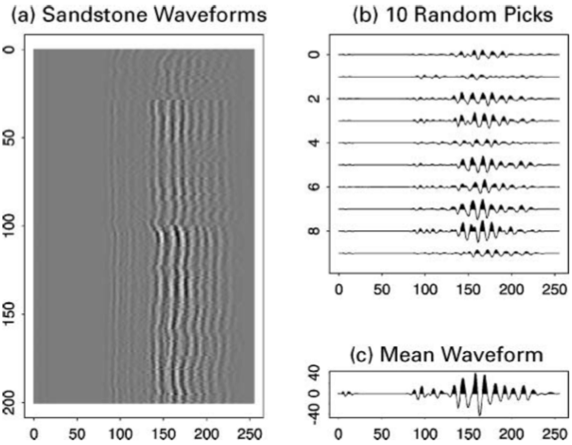


Fig. 2. The acoustic waveforms propagated through sandstone layers: (a) Original 201 waveforms displayed as gray scale images. The horizontal axis represents time samples (with sampling rate 10 μs). (b) Ten waveforms randomly selected from the 201 waveforms are displayed as wiggles (the positive parts are painted in black). (c) The mean waveform of the training dataset consisting of 101 randomly picked waveforms.

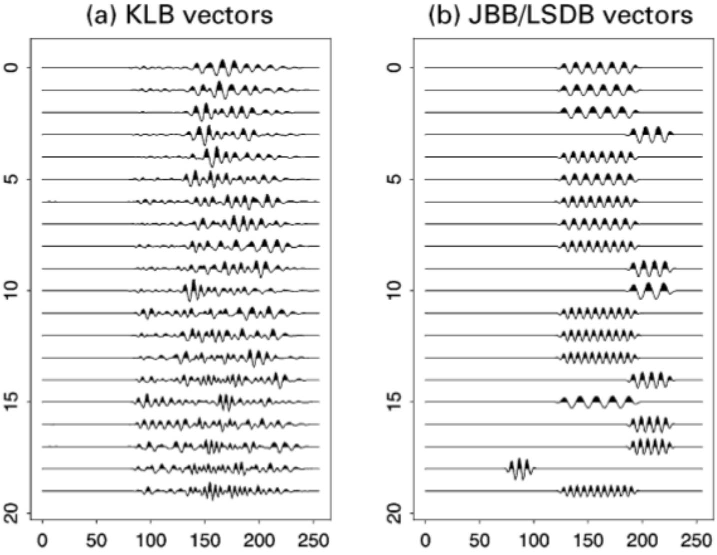


Fig. 3. (a) Top 20 KLB vectors. (b) Top 20 JBB/LSDB vectors. The basis vectors are sorted in the energy-decreasing order.

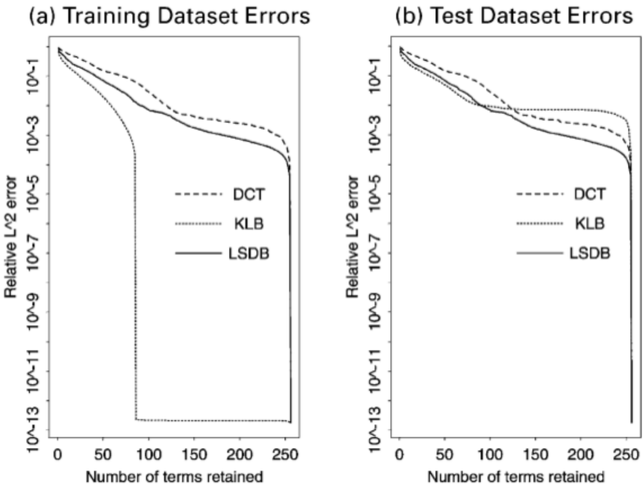


Fig. 4. Relative ℓ^2 approximation errors of the geophysical acoustic waveforms using DCT, KLB, LSDB plotted as functions of the number of terms used for approximation: (a) average errors over all the training signals; (b) average errors over all the test signals.

Observations

- For the training dataset, the KLB approximation was perfect. In fact, the KLB approximation with 86 terms already reached the relative ℓ^2 error of 2.425×10^{-13} on average.
- The same KLB approximates the test dataset better than the JBB only up to 89 terms. If we try to have more accuracy by increasing the number of terms, it got worse than the JBB approximation.
- This implies that these geophysical acoustic waveforms *do not obey the multivariate Gaussian distribution*, and the sample mean and the covariance matrices computed from the training dataset were not enough to capture the statistics of the test dataset.
- On the other hand, the JBB and global DCT approximations are quite *consistent* for both the training and the test datasets.
- The *locality* of the basis functions of the JBB clearly gave a better performance than the global DCT basis functions.

Observations

- For the training dataset, the KLB approximation was perfect. In fact, the KLB approximation with 86 terms already reached the relative ℓ^2 error of 2.425×10^{-13} on average.
- The same KLB approximates the test dataset better than the JBB only up to 89 terms. If we try to have more accuracy by increasing the number of terms, it got worse than the JBB approximation.
- This implies that these geophysical acoustic waveforms *do not obey the multivariate Gaussian distribution*, and the sample mean and the covariance matrices computed from the training dataset were not enough to capture the statistics of the test dataset.
- On the other hand, the JBB and global DCT approximations are quite *consistent* for both the training and the test datasets.
- The *locality* of the basis functions of the JBB clearly gave a better performance than the global DCT basis functions.

Observations

- For the training dataset, the KLB approximation was perfect. In fact, the KLB approximation with 86 terms already reached the relative ℓ^2 error of 2.425×10^{-13} on average.
- The same KLB approximates the test dataset better than the JBB only up to 89 terms. If we try to have more accuracy by increasing the number of terms, it got worse than the JBB approximation.
- This implies that these geophysical acoustic waveforms *do not obey the multivariate Gaussian distribution*, and the sample mean and the covariance matrices computed from the training dataset were not enough to capture the statistics of the test dataset.
- On the other hand, the JBB and global DCT approximations are quite *consistent* for both the training and the test datasets.
- The *locality* of the basis functions of the JBB clearly gave a better performance than the global DCT basis functions.

Observations

- For the training dataset, the KLB approximation was perfect. In fact, the KLB approximation with 86 terms already reached the relative ℓ^2 error of 2.425×10^{-13} on average.
- The same KLB approximates the test dataset better than the JBB only up to 89 terms. If we try to have more accuracy by increasing the number of terms, it got worse than the JBB approximation.
- This implies that these geophysical acoustic waveforms *do not obey the multivariate Gaussian distribution*, and the sample mean and the covariance matrices computed from the training dataset were not enough to capture the statistics of the test dataset.
- On the other hand, the JBB and global DCT approximations are quite *consistent* for both the training and the test datasets.
- The *locality* of the basis functions of the JBB clearly gave a better performance than the global DCT basis functions.

Observations

- For the training dataset, the KLB approximation was perfect. In fact, the KLB approximation with 86 terms already reached the relative ℓ^2 error of 2.425×10^{-13} on average.
- The same KLB approximates the test dataset better than the JBB only up to 89 terms. If we try to have more accuracy by increasing the number of terms, it got worse than the JBB approximation.
- This implies that these geophysical acoustic waveforms *do not obey the multivariate Gaussian distribution*, and the sample mean and the covariance matrices computed from the training dataset were not enough to capture the statistics of the test dataset.
- On the other hand, the JBB and global DCT approximations are quite *consistent* for both the training and the test datasets.
- The *locality* of the basis functions of the JBB clearly gave a better performance than the global DCT basis functions.

Outline

More Dictionaries

- Straightforward generalizations of wavelet packets and BDCT/LCT dictionaries via tensor products cannot always sparsify images or signals in higher dimensions due to the lack of *orientation selectivities*.
- Hence, many new dictionaries have been developed for images based on mathematical modeling of images, e.g., curvelets (Candès-Donoho); contourlets (Do-Vetterli); bandlets (LePennec-Mallat-Peyré); …dual-tree $\mathbb{C}$WT (Selesnick-Baraniuk-Kingsbury); shearlets (Kutyniok-Labate);
- On the other hand, data-driven dictionaries have become quite popular, e.g., Sparse coding (Olshausen-Field); K-SVD (Aharon-Elad-Bruckstein); …
- See [3] for their nice review on all of the above dictionaries.

More Dictionaries

- Straightforward generalizations of wavelet packets and BDCT/LCT dictionaries via tensor products cannot always sparsify images or signals in higher dimensions due to the lack of *orientation selectivities*.
- Hence, many new dictionaries have been developed for images based on mathematical modeling of images, e.g., curvelets (Candès-Donoho); contourlets (Do-Vetterli); bandlets (LePennec-Mallat-Peyré); ... dual-tree $\mathbb{C}$WT (Selesnick-Baraniuk-Kingsbury); shearlets (Kutyniok-Labate);
- On the other hand, data-driven dictionaries have become quite popular, e.g., Sparse coding (Olshausen-Field); K-SVD (Aharon-Elad-Bruckstein); ...
- See [3] for their nice review on all of the above dictionaries.

More Dictionaries

- Straightforward generalizations of wavelet packets and BDCT/LCT dictionaries via tensor products cannot always sparsify images or signals in higher dimensions due to the lack of *orientation selectivities*.
- Hence, many new dictionaries have been developed for images based on mathematical modeling of images, e.g., curvelets (Candès-Donoho); contourlets (Do-Vetterli); bandlets (LePennec-Mallat-Peyré); ... dual-tree $\mathbb{C}$WT (Selesnick-Baraniuk-Kingsbury); shearlets (Kutyniok-Labate);
- On the other hand, data-driven dictionaries have become quite popular, e.g., Sparse coding (Olshausen-Field); K-SVD (Aharon-Elad-Bruckstein); ...
- See [3] for their nice review on all of the above dictionaries.

More Dictionaries

- Straightforward generalizations of wavelet packets and BDCT/LCT dictionaries via tensor products cannot always sparsify images or signals in higher dimensions due to the lack of *orientation selectivities*.
- Hence, many new dictionaries have been developed for images based on mathematical modeling of images, e.g., curvelets (Candès-Donoho); contourlets (Do-Vetterli); bandlets (LePennec-Mallat-Peyré); ... dual-tree $\mathbb{C}$WT (Selesnick-Baraniuk-Kingsbury); shearlets (Kutyniok-Labate);
- On the other hand, data-driven dictionaries have become quite popular, e.g., Sparse coding (Olshausen-Field); K-SVD (Aharon-Elad-Bruckstein); ...
- See [3] for their nice review on all of the above dictionaries.

Outline

For more information about the library and dictionaries of ONBs, the best-basis algorithm and their variants, and many applications, see, e.g., [1]; [2, Chap. 8]; [4];[5]; [6]; [7, Chap. 4, 7, 8]; For the recent review on many more dictionaries, see [3].

[1] R. R. Coifman and M. V. Wickerhauser, *Entropy-based algorithms for best basis selection*, IEEE Trans. Inform. Theory, 38 (1992), pp. 713–719.

[2] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, Burlington, MA, third ed., 2009.

[3] R. Rubinstein, A. M. Bruckstein, and M. Elad, *Dictionaries for sparse representation modeling*, Proc. IEEE, 98 (2010), pp. 1045–1057.

[4] N. Saito, *Local feature extraction and its applications using a library of bases*, in Topics in Analysis and Its Applications: Selected Theses, R. Coifman, ed., World Scientific Pub. Co., Singapore, 2000, pp. 269–451.

[5] ———, *Image approximation and modeling via least statistically dependent bases*, Pattern Recognition, 34 (2001), pp. 1765–1784.

[6] L. F. Villemoes, *Adapted bases of time-frequency local cosines*, Applied and Computational Harmonic Analysis, 10 (2001), pp. 139–162.

[7] M. V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, A K Peters, Ltd., Wellesley, MA, 1994.