Multiresolution Analysis and Wavelets on Hierarchical Data Trees

Jianzhong Wang
Sam Houston State University
Huntsville, Texas USA

June 2, 2017

Workshop on Frame Theory and Sparse Representation for Complex Data
29 May - 2 June 2017, Singapore

Outline

- Introduction

Outline

- Introduction
- Construction of hierarchical data tree via data graph

Outline

- Introduction
- Construction of hierarchical data tree via data graph
- Construction of wavelet basis and frame on hierarchical data

Outline

- Introduction
- Construction of hierarchical data tree via data graph
- Construction of wavelet basis and frame on hierarchical data
- Wavelet representations of functions on data set

Section 1. Introduction

The connection of sensor locations in US

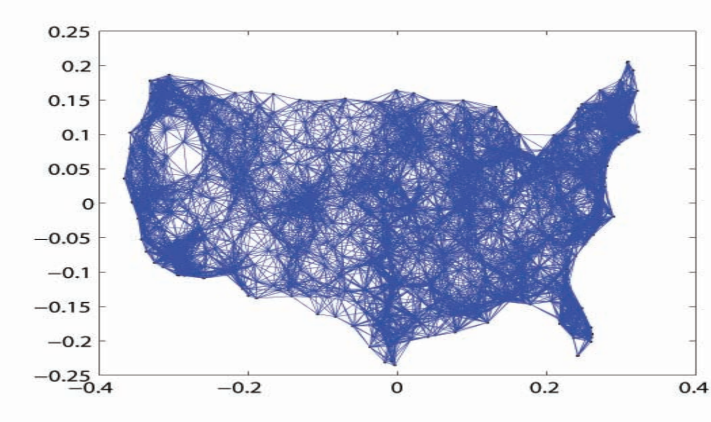


Figure: Sensor locations inferred for $n = 1055$ largest cities in the continental US. On average, each sensor estimated local distances to 18 neighbors, with measurements corrupted by 10% Gaussian noise. We assume that the locations in the figure is not known in prior. Only the distance of two locations within radius of 0.1 can be measured.

Weighted graph for a give data set

The geometric structure of a data set is given by the weighted graph on the data.

Weighted graph for a give data set

The geometric structure of a data set is given by the weighted graph on the data.

- Let $X \subset \mathbb{R}^D$ and $|X| = n$. A weighted graph on X is the triple $G = [X, E, W]$, where X is the node set, E is the edge set, and W is an $n \times n$ (sparse) weight matrix with $w_{i,j} = w_{j,i}$ and $\begin{cases} w_{i,j} = 0, & (x_i, x_j) \notin E, \\ w_{i,j} > 0, & (x_i, x_j) \in E \end{cases}.$

Weighted graph for a give data set

The geometric structure of a data set is given by the weighted graph on the data.

- Let $X \subset \mathbb{R}^D$ and $|X| = n$. A weighted graph on X is the triple $G = [X, E, W]$, where X is the node set, E is the edge set, and W is an $n \times n$ (sparse) weight matrix with $w_{i,j} = w_{j,i}$ and
$$\begin{cases} w_{i,j} = 0, & (x_i, x_j) \notin E, \\ w_{i,j} > 0, & (x_i, x_j) \in E \end{cases}.$$
- Example: $w_{i,j} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \quad (x_i, x_j) \in E.$

Weighted graph for a give data set

The geometric structure of a data set is given by the weighted graph on the data.

- Let $X \subset \mathbb{R}^D$ and $|X| = n$. A weighted graph on X is the triple $G = [X, E, W]$, where X is the node set, E is the edge set, and W is an $n \times n$ (sparse) weight matrix with $w_{i,j} = w_{j,i}$ and $\begin{cases} w_{i,j} = 0, & (x_i, x_j) \notin E, \\ w_{i,j} > 0, & (x_i, x_j) \in E \end{cases}.$
- Example: $w_{i,j} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \quad (x_i, x_j) \in E.$
- The weight matrix defines a metric on the graph G, which defines the kernel distance on X:

$$d_W^2(x_i, x_j) = w_{i,i} + w_{j,j} - 2w_{i,j}.$$

On a connected data graph $G = [X, E, W]$, the weight is given by a positive definite and symmetric kernel $k(x_i, x_j) = w_{i,j}$. Let $d(x) = \int_X k(x, y) d\mu(y)$.

On a connected data graph $G = [X, E, W]$, the weight is given by a positive definite and symmetric kernel $k(x_i, x_j) = w_{i,j}$. Let $d(x) = \int_X k(x, y) d\mu(y)$.

- Diffusion kernel: $\tilde{k}(x, y) = \frac{k(x,y)}{\sqrt{d(x)d(y)}} = \sum_{j=0}^{n-1} \lambda_j^2 \phi_j(x)\phi_j(y)$. Then $\lambda_0 = 1$ and $\lambda_1 < 1$.

On a connected data graph $G = [X, E, W]$, the weight is given by a positive definite and symmetric kernel $k(x_i, x_j) = w_{i,j}$. Let $d(x) = \int_X k(x, y) d\mu(y)$.

- Diffusion kernel: $\tilde{k}(x, y) = \frac{k(x,y)}{\sqrt{d(x)d(y)}} = \sum_{j=0}^{n-1} \lambda_j^2 \phi_j(x)\phi_j(y)$. Then $\lambda_0 = 1$ and $\lambda_1 < 1$.
- Diffusion map: It is defined as $\{\Phi_t\} : X \to l^2$ such that

$$\Phi_t(x) = [\lambda_1^t \phi_1(x), \cdots, \lambda_{n-1}^t \phi_{n-1}(x)]^T.$$

On a connected data graph $G = [X, E, W]$, the weight is given by a positive definite and symmetric kernel $k(x_i, x_j) = w_{i,j}$. Let $d(x) = \int_X k(x, y) d\mu(y)$.

- Diffusion kernel: $\tilde{k}(x, y) = \frac{k(x,y)}{\sqrt{d(x)d(y)}} = \sum_{j=0}^{n-1} \lambda_j^2 \phi_j(x)\phi_j(y)$. Then $\lambda_0 = 1$ and $\lambda_1 < 1$.
- Diffusion map: It is defined as $\{\Phi_t\} : X \to l^2$ such that

$$\Phi_t(x) = [\lambda_1^t \phi_1(x), \cdots, \lambda_{n-1}^t \phi_{n-1}(x)]^T.$$

- Diffusion distance: $d_{\tilde{k}^t}(x, y) = \|\Phi_t(x) - \Phi_t(y)\|$.

Ref. [Coifman and Maggioni, Diffusion Wavelets, 2006. Similar idea from Wilkinson (1965), Watkins (1982, 1991)]

Let $\mathcal{H} = L^2(X, \mu)$ be a Hilbert space of functions on (X, μ) and the diffusion operator on $\mathcal{H}$ be $(T^t f)(x) = \int_X \tilde{k}^t(x, y) f(y) d\mu(y)$.

Ref. [Coifman and Maggioni, Diffusion Wavelets, 2006. Similar idea from Wilkinson (1965), Watkins (1982, 1991)]

Let $\mathcal{H} = L^2(X, \mu)$ be a Hilbert space of functions on (X, μ) and the diffusion operator on $\mathcal{H}$ be $(T^t f)(x) = \int_X \tilde{k}^t(x, y) f(y) d\mu(y)$.

- Let $\epsilon > 0$ be sufficient small. A subspace $S \subset \mathcal{H}$ is called a ϵ-null space of T^t if $\|T^t f\| \leqslant \epsilon \|f\|$ for all $f \in S$. We denote it by $S = Nul_\epsilon(T^t)$.

Ref. [Coifman and Maggioni, Diffusion Wavelets, 2006. Similar idea from Wilkinson (1965), Watkins (1982, 1991)]

Let $\mathcal{H} = L^2(X, \mu)$ be a Hilbert space of functions on (X, μ) and the diffusion operator on $\mathcal{H}$ be $(T^t f)(x) = \int_X \tilde{k}^t(x, y) f(y) d\mu(y)$.

- Let $\epsilon > 0$ be sufficient small. A subspace $S \subset \mathcal{H}$ is called a ϵ-null space of T^t if $\|T^t f\| \leqslant \epsilon \|f\|$ for all $f \in S$. We denote it by $S = Nul_\epsilon(T^t)$.
- Let $V_0 = \mathcal{H}$, $V_j = T^{2^{j-1}}(\mathcal{H})$, and $n \in \mathbb{N}$ be the integer such that $V_n = Span(\mathbf{v}_0)$. Then

$$V_0 \supset V_1 \supset V_2 \supset \cdots \supset V_n$$

  is an MRA of $\mathcal{H}$ with respect to T.

Ref. [Coifman and Maggioni, Diffusion Wavelets, 2006. Similar idea from Wilkinson (1965), Watkins (1982, 1991)]

Let $\mathcal{H} = L^2(X, \mu)$ be a Hilbert space of functions on (X, μ) and the diffusion operator on $\mathcal{H}$ be $(T^t f)(x) = \int_X \tilde{k}^t(x, y) f(y) d\mu(y)$.

- Let $\epsilon > 0$ be sufficient small. A subspace $S \subset \mathcal{H}$ is called a ϵ-null space of T^t if $\|T^t f\| \leqslant \epsilon \|f\|$ for all $f \in S$. We denote it by $S = Nul_\epsilon(T^t)$.
- Let $V_0 = \mathcal{H}$, $V_j = T^{2^{j-1}}(\mathcal{H})$, and $n \in \mathbb{N}$ be the integer such that $V_n = Span(\mathbf{v}_0)$. Then

$$V_0 \supset V_1 \supset V_2 \supset \cdots \supset V_n$$

  is an MRA of $\mathcal{H}$ with respect to T.
- Let $V_j = V_{j+1} \oplus W_{j+1}$, $V_{j+1} \perp W_{j+1}$. An o.n. basis of W_j is call the diffusion wavelet basis of W_j.

Ref. [Coifman and Maggioni, Diffusion Wavelets, 2006. Similar idea from Wilkinson (1965), Watkins (1982, 1991)]

Let $\mathcal{H} = L^2(X,\mu)$ be a Hilbert space of functions on (X,μ) and the diffusion operator on $\mathcal{H}$ be $(T^t f)(x) = \int_X \tilde{k}^t(x,y) f(y) d\mu(y)$.

- Let $\epsilon > 0$ be sufficient small. A subspace $S \subset \mathcal{H}$ is called a ϵ-null space of T^t if $\|T^t f\| \leqslant \epsilon \|f\|$ for all $f \in S$. We denote it by $S = Nul_\epsilon(T^t)$.
- Let $V_0 = \mathcal{H}$, $V_j = T^{2^{j-1}}(\mathcal{H})$, and $n \in \mathbb{N}$ be the integer such that $V_n = Span(\mathbf{v}_0)$. Then

$$V_0 \supset V_1 \supset V_2 \supset \cdots \supset V_n$$

  is an MRA of $\mathcal{H}$ with respect to T.
- Let $V_j = V_{j+1} \oplus W_{j+1}$, $V_{j+1} \perp W_{j+1}$. An o.n. basis of W_j is call the diffusion wavelet basis of W_j.
- The all basis of W_j and $\mathbf{v}_0$ form a basis of $\mathcal{H}$.

Ref. [D.K. Hammond, P. Vandergheynst, R. Gribonval, 2011]

Let $\mathcal{L}$ be the graph Laplacian on G such that
$\mathcal{L} = \sum_{j=0}^{n-1} \lambda_j \chi_j(x)\chi_j(y)$, where $0 = \lambda_0 < \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_{n-1}$.

Ref. [D.K. Hammond, P. Vandergheynst, R. Gribonval, 2011]

Let $\mathcal{L}$ be the graph Laplacian on G such that
$\mathcal{L} = \sum_{j=0}^{n-1} \lambda_j \chi_j(x)\chi_j(y)$, where $0 = \lambda_0 < \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_{n-1}$.

- Let g be a function on $\mathbb{R}^+$. The wavelet operator is defined as $T_g = g(\mathcal{L}) = \sum_{j=0}^{n-1} g(\lambda_j)\chi_j(x)\chi_j(y)$.

Ref. [D.K. Hammond, P. Vandergheynst, R. Gribonval, 2011]

Let $\mathcal{L}$ be the graph Laplacian on G such that $\mathcal{L} = \sum_{j=0}^{n-1} \lambda_j \chi_j(x)\chi_j(y)$, where $0 = \lambda_0 < \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_{n-1}$.

- Let g be a function on $\mathbb{R}^+$. The wavelet operator is defined as $T_g = g(\mathcal{L}) = \sum_{j=0}^{n-1} g(\lambda_j)\chi_j(x)\chi_j(y)$.
- The spectral graph wavelet is defined as $\psi_{t,x}(y) = \sum_{j=0}^{n-1} g(t\lambda_j)\chi_j(x)\chi_j(y), \quad x \in X, t \geqslant 0.$

Ref. [D.K. Hammond, P. Vandergheynst, R. Gribonval, 2011]

Let $\mathcal{L}$ be the graph Laplacian on G such that $\mathcal{L} = \sum_{j=0}^{n-1} \lambda_j \chi_j(x)\chi_j(y)$, where $0 = \lambda_0 < \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_{n-1}$.

- Let g be a function on $\mathbb{R}^+$. The wavelet operator is defined as $T_g = g(\mathcal{L}) = \sum_{j=0}^{n-1} g(\lambda_j)\chi_j(x)\chi_j(y)$.
- The spectral graph wavelet is defined as $\psi_{t,x}(y) = \sum_{j=0}^{n-1} g(t\lambda_j)\chi_j(x)\chi_j(y), \quad x \in X, t \geqslant 0.$
- The wavelet transform of f is given by

$$W_f(t,x) = \langle \psi_{t,x}, f \rangle = \sum_{j=0}^{n-1} g(t\lambda_j)\chi_j(x) \sum_{y \in X} \chi_j(y) f(y)$$

Our Purpose

Constructing “traditional“ compact supported wavelets on data set

- Construction of MRA on the data via hierarchical tree

Constructing “traditional“ compact supported wavelets on data set

- Construction of MRA on the data via hierarchical tree
- Construction of compact supported wavelet basis and frame on data sets

Our Purpose

Constructing “traditional“ compact supported wavelets on data set

- Construction of MRA on the data via hierarchical tree
- Construction of compact supported wavelet basis and frame on data sets
- Development of pyramid algorithm for wavelet decomposition and recovering of functions on data

Section 2. Construction of hierarchical data tree via data graph

Data partition via normalized cut

We adopt the method proposed by [J. Shi and J. Malik, 2000]. Let $A, B, V \subset X$ s.t. $A \bigcap B = \emptyset, A \bigcup B = V$ and $A \subset V$. Define the cut of (A, B) (w.r.t. V) and the association of (A, V) as

$cut(A, B) = \sum_{a \in A, b \in B} k(a, b), \quad assoc(A, V) = \sum_{a \in A, v \in V} k(a, v)$

We adopt the method proposed by [J. Shi and J. Malik, 2000]. Let $A, B, V \subset X$ s.t. $A \bigcap B = \emptyset, A \bigcup B = V$ and $A \subset V$. Define the cut of (A, B) (w.r.t. V) and the association of (A, V) as
$cut(A, B) = \sum_{a \in A, b \in B} k(a, b), \quad assoc(A, V) = \sum_{a \in A, v \in V} k(a, v)$

Definition

The normalized cut of (A, B) (w.r.t. V) is the following number:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

Data partition via normalized cut

We adopt the method proposed by [J. Shi and J. Malik, 2000]. Let $A, B, V \subset X$ s.t. $A \bigcap B = \emptyset, A \bigcup B = V$ and $A \subset V$. Define the cut of (A, B) (w.r.t. V) and the association of (A, V) as
$cut(A, B) = \sum_{a \in A, b \in B} k(a, b), \quad assoc(A, V) = \sum_{a \in A, v \in V} k(a, v)$

Definition

The normalized cut of (A, B) (w.r.t. V) is the following number:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

- $Ncut(A, B)$ can be naturally extended to $Ncut(A_1, \cdots, A_k)$.

Data partition via normalized cut

We adopt the method proposed by [J. Shi and J. Malik, 2000]. Let $A, B, V \subset X$ s.t. $A \bigcap B = \emptyset, A \bigcup B = V$ and $A \subset V$. Define the cut of (A, B) (w.r.t. V) and the association of (A, V) as $cut(A, B) = \sum_{a \in A, b \in B} k(a, b), \quad assoc(A, V) = \sum_{a \in A, v \in V} k(a, v)$

Definition

The normalized cut of (A, B) (w.r.t. V) is the following number:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

- $Ncut(A, B)$ can be naturally extended to $Ncut(A_1, \cdots, A_k)$.
- The optimal k-partition of V is the solution:

$$(A_1, \cdots, A_k) = \arg\min Ncut(A_1, \cdots, A_k)$$

where $\bigcup_{j=1}^{k} A_j = V$ and $A_i \bigcap A_j = \emptyset$, if $i \neq j$.

The optimal k-partition is a NP problem, which can be relaxed to a problem of eigen-decomposition of $\mathcal{L}$ and approximatively solved as following:

The optimal k-partition is a NP problem, which can be relaxed to a problem of eigen-decomposition of $\mathcal{L}$ and approximatively solved as following:

- Obtaining a k-dimensional reduction $Y \subset \mathbb{R}^k$ of V

The optimal k-partition is a NP problem, which can be relaxed to a problem of eigen-decomposition of $\mathcal{L}$ and approximatively solved as following:

- Obtaining a k-dimensional reduction $Y \subset \mathbb{R}^k$ of V
- Making a k partition of Y using a clustering algorithm, say, the k-mean one.

The optimal k-partition is a NP problem, which can be relaxed to a problem of eigen-decomposition of $\mathcal{L}$ and approximatively solved as following:

- Obtaining a k-dimensional reduction $Y \subset \mathbb{R}^k$ of V
- Making a k partition of Y using a clustering algorithm, say, the k-mean one.
- Deriving the k-partition of V from the k-partition of Y.

The optimal k-partition is a NP problem, which can be relaxed to a problem of eigen-decomposition of $\mathcal{L}$ and approximatively solved as following:

- Obtaining a k-dimensional reduction $Y \subset \mathbb{R}^k$ of V
- Making a k partition of Y using a clustering algorithm, say, the k-mean one.
- Deriving the k-partition of V from the k-partition of Y.
- There are (self-tuning) algorithms for finding the optimal partition number k. [A. Zelnik-Manor and P. Perona, 2004.]

The optimal k-partition is a NP problem, which can be relaxed to a problem of eigen-decomposition of $\mathcal{L}$ and approximatively solved as following:

- Obtaining a k-dimensional reduction $Y \subset \mathbb{R}^k$ of V
- Making a k partition of Y using a clustering algorithm, say, the k-mean one.
- Deriving the k-partition of V from the k-partition of Y.
- There are (self-tuning) algorithms for finding the optimal partition number k. [A. Zelnik-Manor and P. Perona, 2004.]

Applying the partition algorithm recursively, we construct a *multi-layer partition*, in which the cluster number k can be varied for each subpartition.

Definition

Assume X has a L-layer partition s.t. $X = X_1^L = \bigcup_{j=1}^{n_{L-1}} X_j^{L-1}$, and for $1 \leqslant \ell \leqslant L$, $X_k^\ell = \bigcup X_j^{\ell-1}$. Define $S_\ell = \{X_1^\ell, \cdots, X_{n_\ell}^\ell\}, 1 \leqslant \ell \leqslant L$, and $S_0 = X$. Then the structure

$$S_L \lhd S_{L-1} \lhd \cdots \lhd S_1 \lhd S_0$$

is called a hierarchical data tree and denoted by $\mathcal{T}(X)$.

Definition

Assume X has a L-layer partition s.t. $X = X_1^L = \bigcup_{j=1}^{n_{L-1}} X_j^{L-1}$, and for $1 \leqslant \ell \leqslant L$, $X_k^\ell = \bigcup X_j^{\ell-1}$. Define $S_\ell = \{X_1^\ell, \cdots, X_{n_\ell}^\ell\}, 1 \leqslant \ell \leqslant L$, and $S_0 = X$. Then the structure

$$S_L \lhd S_{L-1} \lhd \cdots \lhd S_1 \lhd S_0$$

is called a hierarchical data tree and denoted by $\mathcal{T}(X)$.

- S_L is called the *roof* of the tree $\mathcal{T}(x)$, the points in X called the *leaves*, and a set in S_ℓ called a ℓ-level *folder* (or a ℓ*-level node*).

Definition

Assume X has a L-layer partition s.t. $X = X_1^L = \bigcup_{j=1}^{n_{L-1}} X_j^{L-1}$, and for $1 \leqslant \ell \leqslant L$, $X_k^\ell = \bigcup X_j^{\ell-1}$. Define $S_\ell = \{X_1^\ell, \cdots, X_{n_\ell}^\ell\}, 1 \leqslant \ell \leqslant L$, and $S_0 = X$. Then the structure

$$S_L \lhd S_{L-1} \lhd \cdots \lhd S_1 \lhd S_0$$

is called a hierarchical data tree and denoted by $\mathcal{T}(X)$.

- S_L is called the *roof* of the tree $\mathcal{T}(x)$, the points in X called the *leaves*, and a set in S_ℓ called a ℓ-level *folder* (or a ℓ*-level node*).
- The set X_j^k has a double identities: A subset of X and a k-level folder in the tree.

Definition

Assume X has a L-layer partition s.t. $X = X_1^L = \bigcup_{j=1}^{n_{L-1}} X_j^{L-1}$, and for $1 \leqslant \ell \leqslant L$, $X_k^\ell = \bigcup X_j^{\ell-1}$. Define $S_\ell = \{X_1^\ell, \cdots, X_{n_\ell}^\ell\}, 1 \leqslant \ell \leqslant L$, and $S_0 = X$. Then the structure

$$S_L \lhd S_{L-1} \lhd \cdots \lhd S_1 \lhd S_0$$

is called a hierarchical data tree and denoted by $\mathcal{T}(X)$.

- S_L is called the *roof* of the tree $\mathcal{T}(x)$, the points in X called the *leaves*, and a set in S_ℓ called a ℓ-level *folder* (or a ℓ-*level node*).
- The set X_j^k has a double identities: A subset of X and a k-level folder in the tree.
- We have $\bigcup_k (X_k^\ell) = X, |S_0| = |X| = n, |S_L| = 1$.

By the partition tree, we can construct the hierarchical date tree. We apply an ordering operator to sort the nodes at each level, from the root to the leaves.

- Full tree.

Deferent hierarchical data trees

By the partition tree, we can construct the hierarchical date tree. We apply an ordering operator to sort the nodes at each level, from the root to the leaves.

- Full tree.
- Tight balanced tree.

By the partition tree, we can construct the hierarchical date tree. We apply an ordering operator to sort the nodes at each level, from the root to the leaves.

- Full tree.
- Tight balanced tree.
- Balanced tree.

By the partition tree, we can construct the hierarchical date tree. We apply an ordering operator to sort the nodes at each level, from the root to the leaves.

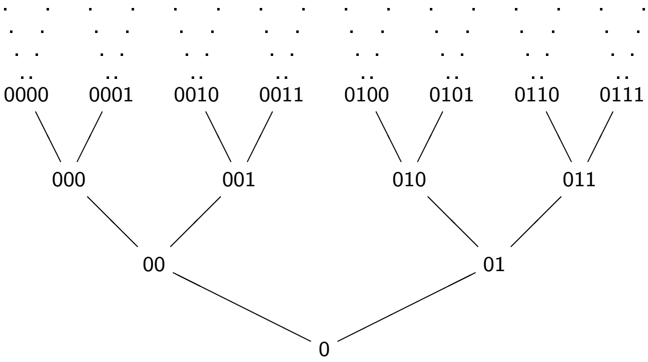
- Full tree.
- Tight balanced tree.
- Balanced tree.
  For all parent and child folders,
  $0 < \underline{B} \leqslant \frac{|\text{child folder}|}{|\text{parent folder}|} \leqslant \overline{B} < 1.$

By the partition tree, we can construct the hierarchical date tree. We apply an ordering operator to sort the nodes at each level, from the root to the leaves.

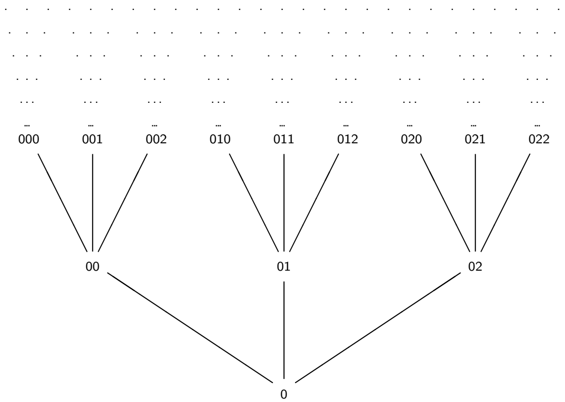
- Full tree.
- Tight balanced tree.
- Balanced tree.
  For all parent and child folders,
  $0 < \underline{B} \leqslant \frac{|\text{child folder}|}{|\text{parent folder}|} \leqslant \overline{B} < 1.$

For a balanced tree, the number of levels is $L = \bigcirc(\log n)$.

Full data tree I: Binary tree

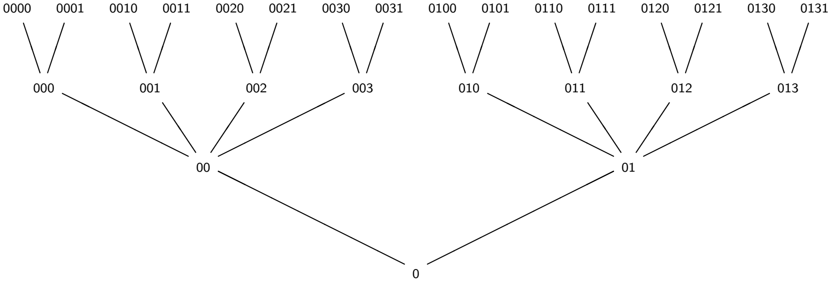


Binary full data tree



Ternary Full Tree

Tight balance tree



Tight Balanced Tree

Distance between folders

Let $A = \{a_1, a_2, \cdots, a_k\}$ and $B = \{b_1, b_2, \cdots, b_m\}$ be two folders at $(L-1)$ level.

- Define the $k \times m$ distance matrix $D(A, B) = [d_G(a_i, b_j)]$.

Distance between folders

Let $A = \{a_1, a_2, \cdots, a_k\}$ and $B = \{b_1, b_2, \cdots, b_m\}$ be two folders at $(L-1)$ level.

- Define the $k \times m$ distance matrix $D(A, B) = [d_G(a_i, b_j)]$.
- The average distance $d_a(A, B) = \|D(A, B)\|_F$.

Distance between folders

Let $A = \{a_1, a_2, \cdots, a_k\}$ and $B = \{b_1, b_2, \cdots, b_m\}$ be two folders at $(L-1)$ level.

- Define the $k \times m$ distance matrix $D(A, B) = [d_G(a_i, b_j)]$.
- The average distance $d_a(A, B) = \|D(A, B)\|_F$.
- The shortest distance $d_s(A, B) = \min d_G(a_i, b_j)$.

Distance between folders

Let $A = \{a_1, a_2, \cdots, a_k\}$ and $B = \{b_1, b_2, \cdots, b_m\}$ be two folders at $(L-1)$ level.

- Define the $k \times m$ distance matrix $D(A, B) = [d_G(a_i, b_j)]$.
- The average distance $d_a(A, B) = \|D(A, B)\|_F$.
- The shortest distance $d_s(A, B) = \min d_G(a_i, b_j)$.
- The longest distance $d_l(A, B) = \max d_G(a_i, a_j)$.

Distance between folders

Let $A = \{a_1, a_2, \cdots, a_k\}$ and $B = \{b_1, b_2, \cdots, b_m\}$ be two folders at $(L-1)$ level.

- Define the $k \times m$ distance matrix $D(A, B) = [d_G(a_i, b_j)]$.
- The average distance $d_a(A, B) = \|D(A, B)\|_F$.
- The shortest distance $d_s(A, B) = \min d_G(a_i, b_j)$.
- The longest distance $d_l(A, B) = \max d_G(a_i, a_j)$.

Let $A = \{a_1, a_2, \cdots, a_k\}$ and d a distance on A. Let π be an index permutation of $[1, \cdots, k]$. We call $\mathbf{a}_\pi = [a_{\pi(1)}, a_{\pi(2)}, \cdots, a_{\pi(k)}]$ a *stack* of A headed by $a_{\pi(1)}$, and call $\ell(\mathbf{a}_\pi) = \sum_{j=1}^{k-1} d(a_{\pi(j)}, a_{\pi(j+1)})$ the path length of $\mathbf{a}_\pi$. We denote the set of permutations (with the head l) by

$$\mathcal{P}_l = \{\pi; \quad \pi(1) = l\}.$$

Definition

A shortest-path sorting of A headed by a_l is a stack $\mathbf{a}_\pi$ that has the shortest path length among all pathes starting from a_l:

$$\mathbf{a}_\pi = \arg\min_{\pi \in P_l} \ell(\mathbf{a}_\pi).$$

Denote by A the folder set at a level. Let p be a probability function on A and Ω the sorted index set initialized to $\Omega = \emptyset$.

1. Set $\pi(1) = l$ and update $\Omega = \{l\}$.

Denote by A the folder set at a level. Let p be a probability function on A and Ω the sorted index set initialized to $\Omega = \emptyset$.

1. Set $\pi(1) = l$ and update $\Omega = \{l\}$.
2. After i steps, assume now $\Omega = \{\pi(1), \cdots, \pi(i)\}$. To find $\pi(i+1)$, from unsorted elements, pick up two nearest ones y_1 and y_2 of $a_{\pi(i)}$ and compute

$$q_i = \frac{1}{1 + \exp\left(\frac{d(a_i, y_1) - d(a_i, y_2)}{\alpha}\right)},$$

where $\alpha > 0$ is the *sorting parameter*. If $q_i < p_{\pi(i)}$, we select $a_{\pi(i+1)} = y_2$. Otherwise, select $a_{\pi(i+1)} = y_1$.

Denote by A the folder set at a level. Let p be a probability function on A and Ω the sorted index set initialized to $\Omega = \emptyset$.

1. Set $\pi(1) = l$ and update $\Omega = \{l\}$.
2. After i steps, assume now $\Omega = \{\pi(1), \cdots, \pi(i)\}$. To find $\pi(i+1)$, from unsorted elements, pick up two nearest ones y_1 and y_2 of $a_{\pi(i)}$ and compute

$$q_i = \frac{1}{1 + \exp\left(\frac{d(a_i, y_1) - d(a_i, y_2)}{\alpha}\right)},$$

   where $\alpha > 0$ is the *sorting parameter*. If $q_i < p_{\pi(i)}$, we select $a_{\pi(i+1)} = y_2$. Otherwise, select $a_{\pi(i+1)} = y_1$.
3. Update Ω, and repeat the step above. The algorithm is terminated when $|\Omega| = k$.

Denote by A the folder set at a level. Let p be a probability function on A and Ω the sorted index set initialized to $\Omega = \emptyset$.

1. Set $\pi(1) = l$ and update $\Omega = \{l\}$.
2. After i steps, assume now $\Omega = \{\pi(1), \cdots, \pi(i)\}$. To find $\pi(i+1)$, from unsorted elements, pick up two nearest ones y_1 and y_2 of $a_{\pi(i)}$ and compute

$$q_i = \frac{1}{1 + \exp\left(\frac{d(a_i,y_1)-d(a_i,y_2)}{\alpha}\right)},$$

   where $\alpha > 0$ is the *sorting parameter*. If $q_i < p_{\pi(i)}$, we select $a_{\pi(i+1)} = y_2$. Otherwise, select $a_{\pi(i+1)} = y_1$.
3. Update Ω, and repeat the step above. The algorithm is terminated when $|\Omega| = k$.
4. $\mathbf{a}_\pi$ is an approximative shortest path sorting of A headed by a_l.

Algorithm for building data tree

1. Input: A weighted graph $G = [X, W]$ on the data set X.
2. Construct the matrix $P = D^{-1}W$ and use a fast eigen-decomposition algorithm to find the largest k Left eigenvectors. To make sure that the gap between λ_k and λ_{k+1} is large.
3. Use a partition algorithm, e.g., k-mean, to make a partition of $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$.
4. On each subset X_j, repeat the processing above to partition it again up to L levels.
5. Smoothly order the folders at each level.

Data tree of a brain image

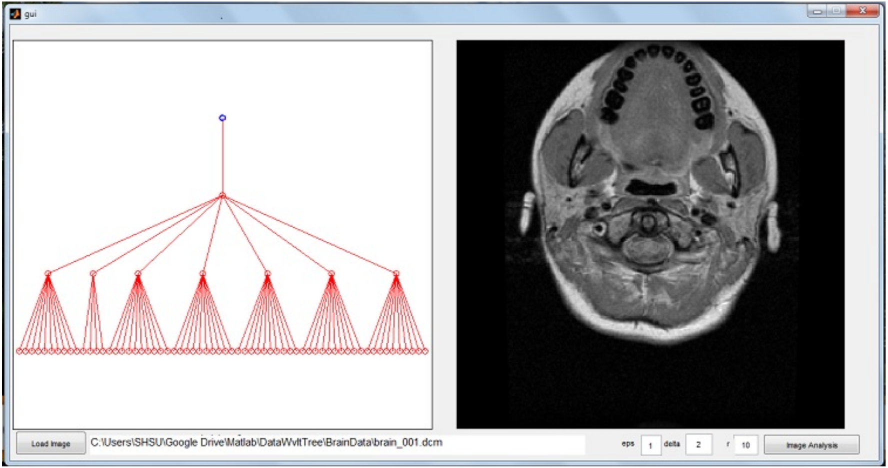


Figure: Data tree of a brain image

Section 3. Construction of hierarchical data tree via data graph

Definition

Let $\mathcal{H}_0 = \mathcal{H}(= L^2(X, d\mu))$ and
$\mathcal{H}_\ell = \{f \in \mathcal{H};\ f(x) = c_j, x \in X_j^\ell \in S_\ell\}$. The hierarchical tree $\mathcal{T}(X)$ derives the following MRA on $\mathcal{H}$:

$$\mathcal{H}_0 \supset \mathcal{H}_1 \cdots \supset \mathcal{H}_L$$

where $\dim(\mathcal{H}_\ell) = n_\ell(= |S_\ell|)$.

Definition

Let $\mathcal{H}_0 = \mathcal{H}(= L^2(X, d\mu))$ and $\mathcal{H}_\ell = \{f \in \mathcal{H};\ f(x) = c_j, x \in X_j^\ell \in S_\ell\}$. The hierarchical tree $\mathcal{T}(X)$ derives the following MRA on $\mathcal{H}$:

$$\mathcal{H}_0 \supset \mathcal{H}_1 \cdots \supset \mathcal{H}_L$$

where $\dim(\mathcal{H}_\ell) = n_\ell(= |S_\ell|)$. Let $\mathcal{W}_\ell \bigoplus \mathcal{H}_\ell = \mathcal{H}_{\ell-1}$ and $\mathcal{W}_\ell \perp \mathcal{H}_\ell$. Then $\mathcal{W}_\ell$ is a wavelet subspace of $\mathcal{H}$.

Definition

Let $\mathcal{H}_0 = \mathcal{H}(= L^2(X, d\mu))$ and $\mathcal{H}_\ell = \{f \in \mathcal{H};\ f(x) = c_j, x \in X_j^\ell \in S_\ell\}$. The hierarchical tree $\mathcal{T}(X)$ derives the following MRA on $\mathcal{H}$:

$$\mathcal{H}_0 \supset \mathcal{H}_1 \cdots \supset \mathcal{H}_L$$

where $\dim(\mathcal{H}_\ell) = n_\ell(= |S_\ell|)$. Let $\mathcal{W}_\ell \bigoplus \mathcal{H}_\ell = \mathcal{H}_{\ell-1}$ and $\mathcal{W}_\ell \perp \mathcal{H}_\ell$. Then $\mathcal{W}_\ell$ is a wavelet subspace of $\mathcal{H}$.

We have $\dim(\mathcal{W}_\ell) = m_\ell = |S_{\ell-1}| - |S_\ell|$, and

$$\mathcal{H} = \mathcal{H}_L \bigoplus \mathcal{W}_L \bigoplus \cdots \bigoplus \mathcal{W}_1.$$

1. In $L^2(X, dx)$, $\langle a, b\rangle = \sum_j a_j b_j$.
   In $L^2(X, d\mu)$, $\langle a, b\rangle_m = \sum_j a_j b_j m_j = \langle a, bm\rangle$.

1. In $L^2(X, dx)$, $\langle a, b \rangle = \sum_j a_j b_j$.
   In $L^2(X, d\mu)$, $\langle a, b \rangle_m = \sum_j a_j b_j m_j = \langle a, bm \rangle$.
2. Let $\{\eta_j\}_{j=1}^n$ be an o.n. basis of $L^2(X, dx)$. Then, setting $\tilde{\eta}_j = \eta_j / \sqrt{m}$, $\{\tilde{\eta}_j\}_{j=1}^n$ is an o.n. basis of $L^2(X, d\mu)$.

The relation between o.n. bases of $L^2(X, d\mu)$ and of $L^2(X, dx)$

1. In $L^2(X, dx)$, $\langle a, b\rangle = \sum_j a_j b_j$.
   In $L^2(X, d\mu)$, $\langle a, b\rangle_m = \sum_j a_j b_j m_j = \langle a, bm\rangle$.
2. Let $\{\eta_j\}_{j=1}^n$ be an o.n. basis of $L^2(X, dx)$. Then, setting $\tilde{\eta}_j = \eta_j/\sqrt{m}$, $\{\tilde{\eta}_j\}_{j=1}^n$ is an o.n. basis of $L^2(X, d\mu)$.
3. We may use o.n wavelet basis of $L^2(X, dx)$ to perform the o.n. wavelet decomposition and recovering for $f \in L^2(X, d\mu)$ by using the following formula:

$$\langle fm, \eta_j\rangle = \langle f, \tilde{\eta}_j\rangle_m.$$

The scaling functions and wavelet functions in $\mathcal{H} = L^2(X, dx)$ have the following properties:

Properties of scaling function and wavelets

- At the leaf level, the set of delta functions $\{\delta_x\}_{x\in X}$ is an o.n. basis of $\mathcal{H}$. Each $f \in \mathcal{H}$ has the decomposition $f = \sum_j f_j^0 \delta_{x_j}$, where $f_j^0 = f(x_j)$.

The scaling functions and wavelet functions in $\mathcal{H} = L^2(X, dx)$ have the following properties:

Properties of scaling function and wavelets

- At the leaf level, the set of delta functions $\{\delta_x\}_{x\in X}$ is an o.n. basis of $\mathcal{H}$. Each $f \in \mathcal{H}$ has the decomposition $f = \sum_j f_j^0 \delta_{x_j}$, where $f_j^0 = f(x_j)$.
- At Level ℓ, assume $\mathcal{S}_\ell = \{X_j^\ell\}_{j=1}^{n_\ell}$. Let
$$\phi_j^\ell(x) = \begin{cases} \frac{1}{\sqrt{|X_j^\ell|}}, & x \in X_j^\ell, \\ 0, & x \notin X_j^\ell. \end{cases}$$
Then $\{\phi_j^\ell\}_{j=1}^{n_\ell}$ is an o.n. basis of $\mathcal{H}_\ell$.

Hierarchical structure of wavelet basis on $\mathcal{H} = L^2(X, dx)$

The scaling functions and wavelet functions in $\mathcal{H} = L^2(X, dx)$ have the following properties:

Properties of scaling function and wavelets

- At the leaf level, the set of delta functions $\{\delta_x\}_{x \in X}$ is an o.n. basis of $\mathcal{H}$. Each $f \in \mathcal{H}$ has the decomposition $f = \sum_j f_j^0 \delta_{x_j}$, where $f_j^0 = f(x_j)$.
- At Level ℓ, assume $\mathcal{S}_\ell = \{X_j^\ell\}_{j=1}^{n_\ell}$. Let
$$\phi_j^\ell(x) = \begin{cases} \frac{1}{\sqrt{|X_j^\ell|}}, & x \in X_j^\ell, \\ 0, & x \notin X_j^\ell. \end{cases}$$
Then $\{\phi_j^\ell\}_{j=1}^{n_\ell}$ is an o.n. basis of $\mathcal{H}_\ell$.
- There is a wavelet basis $\{\psi_j^\ell\}_{j=1}^{m_\ell}$ of $\mathcal{W}_\ell$ such that each ψ_j^ℓ is locally supported and has at least one vanishing moment, i.e., there is $1 \leqslant s \leqslant m_\ell$, s.t. $\mathrm{supp}(\psi_j^\ell) \subset X_s^\ell$, and $\langle \psi_j^\ell, 1 \rangle = 0$.

By the properties of wavelets, we may construct the wavelet basis on $\mathcal{H}$ folder-by-folder. We denote by Y a folder at 1-level having k leaves: $Y = \{y_j\}_{j=1}^k$. Let $\phi_j^0 = \delta_{y_j}$. Then $\{\phi_j^0\}_{j=1}^k$ is an o.n. basis of $L^2(Y, dy)$. The spatial representation of $f \in L^2(Y, dy)$ is $f = \sum_{j=1}^k f_j \phi_j^0$. We denote by f the vector $[f_1, \cdots, f_k]^T$ too.

By the properties of wavelets, we may construct the wavelet basis on $\mathcal{H}$ folder-by-folder. We denote by Y a folder at 1-level having k leaves: $Y = \{y_j\}_{j=1}^k$. Let $\phi_j^0 = \delta_{y_j}$. Then $\{\phi_j^0\}_{j=1}^k$ is an o.n. basis of $L^2(Y, dy)$. The spatial representation of $f \in L^2(Y, dy)$ is $f = \sum_{j=1}^k f_j \phi_j^0$. We denote by f the vector $[f_1, \cdots, f_k]^T$ too.

Definition

An o.n. wavelet basis on $L^2(Y, dy)$ is a $k \times k$ o.g. matrix: $M = [\phi, \psi_1, \cdots, \psi_{k-1}]$, where the first column ϕ is a scaling function and others are wavelets. The wavelet transform of a function $f \in L^2(Y, dy)$ is given by $d = M^T f$ and the inverse wavelet transform given by in $f = Md$.

By MRA on $L^2(Y)$, we may construct the o.n. wavelet basis of $L^2(Y)$ using a pyramid algorithm.

Let the first layer Haar o.n. wavelet basis be represented as a $k \times k$ matrix $M_1 = [L_1, H_1]$, where $L_1 = [\phi_1^1, \cdots, \phi_{[k/2]}^1]$ contains scaling functions and $H = [\psi_1^1, \cdots, \psi_{[k+1]/2}^1]$ contains wavelets.

Let the first layer Haar o.n. wavelet basis be represented as a $k \times k$ matrix $M_1 = [L_1, H_1]$, where $L_1 = [\phi_1^1, \cdots, \phi_{[k/2]}^1]$ contains scaling functions and $H = [\psi_1^1, \cdots, \psi_{[k+1]/2}^1]$ contains wavelets.

Construction I: From 2 leaf scaling functions

$$[\phi_i^1, \psi_i^1] = [\phi_{2i-1}^0, \phi_{2i}^0] \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Let the first layer Haar o.n. wavelet basis be represented as a $k \times k$ matrix $M_1 = [L_1, H_1]$, where $L_1 = [\phi_1^1, \cdots, \phi_{[k/2]}^1]$ contains scaling functions and $H = [\psi_1^1, \cdots, \psi_{[k+1]/2}^1]$ contains wavelets.

Construction I: From 2 leaf scaling functions

$$[\phi_i^1, \psi_i^1] = [\phi_{2i-1}^0, \phi_{2i}^0] \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

When $k = 2s - 1$, we also need the following:

Construction II: From 3 leaf scaling functions

$$[\phi_{s-1}^1, \psi_{s-1}^1, \psi_s^1] = [\phi_{m-2}^0, \phi_{m-1}^0, \phi_m] \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{bmatrix}.$$

We now construct $(j+1)$-level scaling functions and wavelets from j-level scaling functions $\Phi_j = [\phi_1^j, \cdots, \phi_m^j]$. Write $s_i = |\,\mathrm{supp}(\phi_i^j)|$.

$[\phi_i^{j+1}, \psi_i^{j+1}] = [\phi_{2i-1}^j, \phi_{2i}^j] W_j^2$

$$W_j^2 = \frac{1}{\sqrt{s_{2i-1}+s_{i+2i}}} \begin{bmatrix} \sqrt{s_{2i-1}} & \sqrt{s_{2i}} \\ \sqrt{s_{2i}} & -\sqrt{s_{2i-1}} \end{bmatrix}$$

We now construct $(j+1)$-level scaling functions and wavelets from j-level scaling functions $\Phi_j = [\phi_1^j, \cdots, \phi_m^j]$. Write $s_i = |\operatorname{supp}(\phi_i^j)|$.

$[\phi_i^{j+1}, \psi_i^{j+1}] = [\phi_{2i-1}^j, \phi_{2i}^j] W_j^2$

$$W_j^2 = \frac{1}{\sqrt{s_{2i-1}+s_{i+2i}}} \begin{bmatrix} \sqrt{s_{2i-1}} & \sqrt{s_{2i}} \\ \sqrt{s_{2i}} & -\sqrt{s_{2i-1}} \end{bmatrix}$$

Wavelet transform: $c^{j+1} = (W_j^2)^T c^j, \; c^j = W_j^2 c^{j+1}$.

Multi-layer Haar o.n. wavelet basis in a folder

We now construct $(j+1)$-level scaling functions and wavelets from j-level scaling functions $\Phi_j = [\phi_1^j, \cdots, \phi_m^j]$. Write $s_i = |\,\mathrm{supp}(\phi_i^j)|$.

$$[\phi_i^{j+1}, \psi_i^{j+1}] = [\phi_{2i-1}^j, \phi_{2i}^j] W_j^2$$

$$W_j^2 = \frac{1}{\sqrt{s_{2i-1}+s_{i+2i}}} \begin{bmatrix} \sqrt{s_{2i-1}} & \sqrt{s_{2i}} \\ \sqrt{s_{2i}} & -\sqrt{s_{2i-1}} \end{bmatrix}$$

Wavelet transform: $c^{j+1} = (W_j^2)^T c^j, \; c^j = W_j^2 c^{j+1}$.

When $m = 2s-1$, set $h_m = s_{m-2} + s_{m-1} + s_m$.

$$[\phi_{s-1}^{j+1}, \psi_{s-1}^{j+1}, \psi_s^{j+1}] = [\phi_{m-2}^j, \phi_{m-1}^j, \phi_m^j] W_j^3$$

$$W_j^3 = \begin{bmatrix} \sqrt{\frac{s_{m-2}}{h_m}} & \sqrt{\frac{s_m}{s_{m-2}+s_m}} & \sqrt{\frac{s_{m-1}s_{m-2}}{h_m(s_{m-2}+s_m)}} \\ \sqrt{\frac{s_{m-1}}{h_m}} & 0 & -\sqrt{\frac{s_{m-2}+s_m}{h_m}} \\ \sqrt{\frac{s_m}{h_m}} & -\sqrt{\frac{s_{m-2}}{s_{m-2}+s_m}} & \sqrt{\frac{s_{m-1}s_m}{h_m(s_{m-2}+s_m)}} \end{bmatrix}.$$

We now construct $(j+1)$-level scaling functions and wavelets from j-level scaling functions $\Phi_j = [\phi_1^j, \cdots, \phi_m^j]$. Write $s_i = |\operatorname{supp}(\phi_i^j)|$.

$[\phi_i^{j+1}, \psi_i^{j+1}] = [\phi_{2i-1}^j, \phi_{2i}^j] W_j^2$

$$W_j^2 = \frac{1}{\sqrt{s_{2i-1}+s_{i+2i}}} \begin{bmatrix} \sqrt{s_{2i-1}} & \sqrt{s_{2i}} \\ \sqrt{s_{2i}} & -\sqrt{s_{2i-1}} \end{bmatrix}$$

Wavelet transform: $c^{j+1} = (W_j^2)^T c^j, \; c^j = W_j^2 c^{j+1}$.

When $m = 2s-1$, set $h_m = s_{m-2} + s_{m-1} + s_m$.

$[\phi_{s-1}^{j+1}, \psi_{s-1}^{j+1}, \psi_s^{j+1}] = [\phi_{m-2}^j, \phi_{m-1}^j, \phi_m^j] W_j^3$

$$W_j^3 = \begin{bmatrix} \sqrt{\frac{s_{m-2}}{h_m}} & \sqrt{\frac{s_m}{s_{m-2}+s_m}} & \sqrt{\frac{s_{m-1}s_{m-2}}{h_m(s_{m-2}+s_m)}} \\ \sqrt{\frac{s_{m-1}}{h_m}} & 0 & -\sqrt{\frac{s_{m-2}+s_m}{h_m}} \\ \sqrt{\frac{s_m}{h_m}} & -\sqrt{\frac{s_{m-2}}{s_{m-2}+s_m}} & \sqrt{\frac{s_{m-1}s_m}{h_m(s_{m-2}+s_m)}} \end{bmatrix}.$$

Wavelet transform: $c^{j+1} = (W_j^3)^T c^j, \; c^j = W_j^3 c^{j+1}$.

The construction of wavelets above can be applied to the whole tree. Assume that the Haar wavelet basis has been built up to Level ℓ, where $S_\ell = \{X_1^\ell, \cdots, X_{n_\ell}^\ell\}$. Therefore, in this basis, there are n_ℓ scaling functions: $\phi_j^{(\ell)} = \frac{1}{\sqrt{|X_j^\ell|}}\chi_{X_j^\ell}, 1 \leqslant j \leqslant n_\ell$. Let a wavelet on X_k^ℓ is denoted by $\psi_j^{(\ell,k)}$. (If it is at i-th layer and the layer level need to stress, then it is denoted by $\psi_{i,j}^{(\ell,k)}$.)

The construction of wavelets above can be applied to the whole tree. Assume that the Haar wavelet basis has been built up to Level ℓ, where $S_\ell = \{X_1^\ell, \cdots, X_{n_\ell}^\ell\}$. Therefore, in this basis, there are n_ℓ scaling functions: $\phi_j^{(\ell)} = \frac{1}{\sqrt{|X_j^\ell|}} \chi_{X_j^\ell}, 1 \leqslant j \leqslant n_\ell$. Let a wavelet on X_k^ℓ is denoted by $\psi_j^{(\ell,k)}$. (If it is at i-th layer and the layer level need to stress, then it is denoted by $\psi_{i,j}^{(\ell,k)}$.)

Let $X_1^{\ell+1} = \bigcup_{j=1}^k X_j^\ell, X_1^{\ell+1} \in \mathcal{S}_{\ell+1}$. We construct the $(\ell+1)$-layer wavelets on $X_1^{\ell+1}$ recursively.

- Initialize 0-layer wavelets as $\phi_{0,j}^{(\ell+1,1)} = \phi_j^{(\ell)}, 1 \leqslant j \leqslant k$.

The construction of wavelets above can be applied to the whole tree. Assume that the Haar wavelet basis has been built up to Level ℓ, where $S_\ell = \{X_1^\ell, \cdots, X_{n_\ell}^\ell\}$. Therefore, in this basis, there are n_ℓ scaling functions: $\phi_j^{(\ell)} = \frac{1}{\sqrt{|X_j^\ell|}}\chi_{X_j^\ell}, 1 \leqslant j \leqslant n_\ell$. Let a wavelet on X_k^ℓ is denoted by $\psi_j^{(\ell,k)}$. (If it is at i-th layer and the layer level need to stress, then it is denoted by $\psi_{i,j}^{(\ell,k)}$.)

Let $X_1^{\ell+1} = \bigcup_{j=1}^k X_j^\ell, X_1^{\ell+1} \in \mathcal{S}_{\ell+1}$. We construct the $(\ell+1)$-layer wavelets on $X_1^{\ell+1}$ recursively.

- Initialize 0-layer wavelets as $\phi_{0,j}^{(\ell+1,1)} = \phi_j^{(\ell)}, 1 \leqslant j \leqslant k$.
- When k is even, then apply

$$[\phi_{t+1,i}^{(\ell+1,1)}, \psi_{t+1,i}^{(\ell+1,1)}] = [\phi_{t,2i-1}^{(\ell+1,1)}, \phi_{t,2i}^{(\ell+1,1)}]W_j^2$$

The construction of wavelets above can be applied to the whole tree. Assume that the Haar wavelet basis has been built up to Level ℓ, where $S_\ell = \{X_1^\ell, \cdots, X_{n_\ell}^\ell\}$. Therefore, in this basis, there are n_ℓ scaling functions: $\phi_j^{(\ell)} = \frac{1}{\sqrt{|X_j^\ell|}}\chi_{X_j^\ell}, 1 \leqslant j \leqslant n_\ell$. Let a wavelet on X_k^ℓ is denoted by $\psi_j^{(\ell,k)}$. (If it is at i-th layer and the layer level need to stress, then it is denoted by $\psi_{i,j}^{(\ell,k)}$.)

Let $X_1^{\ell+1} = \bigcup_{j=1}^k X_j^\ell, X_1^{\ell+1} \in \mathcal{S}_{\ell+1}$. We construct the $(\ell+1)$-layer wavelets on $X_1^{\ell+1}$ recursively.

- Initialize 0-layer wavelets as $\phi_{0,j}^{(\ell+1,1)} = \phi_j^{(\ell)}, 1 \leqslant j \leqslant k$.
- When k is even, then apply

$$[\phi_{t+1,i}^{(\ell+1,1)}, \psi_{t+1,i}^{(\ell+1,1)}] = [\phi_{t,2i-1}^{(\ell+1,1)}, \phi_{t,2i}^{(\ell+1,1)}]W_j^2$$

- When $k = 2s - 1$, we apply following for the last block:

$$[\phi_{t+1,s-1}^{(\ell+1,1)}, \psi_{t+1,s-1}^{(\ell+1,1)}, \psi_{t+1,s}^{(\ell+1,1)}] = [\phi_{t,k-2}^{(\ell+1,1)}, \phi_{t,k-1}^{(\ell+1,1)}, \phi_{t,k}^{(\ell+1,1)}]W_j^3$$

Using the similar way, we also can construct a tight frame on the data tree $\mathcal{T}(X)$.

Motivation

- Tight frames have excellent localization.
- The redundance in the frames are very useful in data analysis and processing.
- Rich algorithms and methods for constructions of tight frames with boundaries are available in literature. Ref. [Chan, Riemenschneider, Shen, and Shen, 1998; Cai, Chan, Shen, and Shen, 1998; Daubechies, Han, Ron and Shen, 2003; Shen, 2010; ...].

The steps for constructing tight frame on a data tree

- Construction of tight frame within a folder.
- Construction of tight frame on the whole tree.

Tight frames on a folder (I)

To construct the wavelet tight frame within a folder, we employ the tight framelets on a space of finite sequence $[\mathbf{x}_1, \cdots, \mathbf{x}_N]$ with a certain boundary condition, say, symmetric one. [see Chan, Riemenschneider, Shen,and Shen, 2005]

To construct the wavelet tight frame within a folder, we employ the tight framelets on a space of finite sequence $[\mathbf{x}_1, \cdots, \mathbf{x}_N]$ with a certain boundary condition, say, symmetric one. [see Chan, Riemenschneider, Shen,and Shen, 2005]

- When $L = 3$, choose $h_0 = [1/4, 1/2, 1/4]$, $h_1 = [-1/4, 1/2, -1/4]$, $h_2 = [-\sqrt{2}/4, 0, \sqrt{2}/4]$ as the masks of the generators for the tight frame $[\phi, \psi_1, \psi_2]$.

To construct the wavelet tight frame within a folder, we employ the tight framelets on a space of finite sequence $[\mathbf{x}_1, \cdots, \mathbf{x}_N]$ with a certain boundary condition, say, symmetric one. [see Chan, Riemenschneider, Shen,and Shen, 2005]

- When $L = 3$, choose $h_0 = [1/4, 1/2, 1/4]$, $h_1 = [-1/4, 1/2, -1/4]$, $h_2 = [-\sqrt{2}/4, 0, \sqrt{2}/4]$ as the masks of the generators for the tight frame $[\phi, \psi_1, \psi_2]$.
- When $L = 4$, choose $h_0 = \frac{1}{8}[1, 2, 2, 2, 1]$, $h_1 = \frac{1}{8}[1, 0, 0, 0, -1]$, $h_2 = \frac{\sqrt{2}}{8}\cos\left(\frac{\pi}{8}\right)[1, \sqrt{2}, 0, -\sqrt{2}, -1]$, $h_3 = \frac{\sqrt{2}}{8}[\cos\left(\frac{\pi}{8}\right), -\sqrt{2}\sin\left(\frac{\pi}{8}\right), -2\sin\left(\frac{\pi}{8}\right), -\sqrt{2}\sin\left(\frac{\pi}{8}\right), \cos\left(\frac{\pi}{8}\right)]$ $h_4 = \frac{1}{8}[1, 0, -2, 0, 1]$, $h_5 = \frac{1}{8}[1, -2, 0, 2, -1]$, $h_6 = \frac{\sqrt{2}}{8}\sin\left(\frac{\pi}{8}\right)[1, -\sqrt{2}, 0, \sqrt{2}, -1]$, $h_7 = \frac{\sqrt{2}}{8}[\sin\left(\frac{\pi}{8}\right), -\sqrt{2}\cos\left(\frac{\pi}{8}\right), -2\cos\left(\frac{\pi}{8}\right), -\sqrt{2}\cos\left(\frac{\pi}{8}\right), \sin\left(\frac{\pi}{8}\right)]$

To construct the wavelet tight frame within a folder, we employ the tight framelets on a space of finite sequence $[\mathbf{x}_1, \cdots, \mathbf{x}_N]$ with a certain boundary condition, say, symmetric one. [see Chan, Riemenschneider, Shen,and Shen, 2005]

- When $L = 3$, choose $h_0 = [1/4, 1/2, 1/4]$, $h_1 = [-1/4, 1/2, -1/4]$, $h_2 = [-\sqrt{2}/4, 0, \sqrt{2}/4]$ as the masks of the generators for the tight frame $[\phi, \psi_1, \psi_2]$.
- When $L = 4$, choose $h_0 = \frac{1}{8}[1, 2, 2, 2, 1]$, $h_1 = \frac{1}{8}[1, 0, 0, 0, -1]$, $h_2 = \frac{\sqrt{2}}{8}\cos\left(\frac{\pi}{8}\right)[1, \sqrt{2}, 0, -\sqrt{2}, -1]$, $h_3 = \frac{\sqrt{2}}{8}[\cos\left(\frac{\pi}{8}\right), -\sqrt{2}\sin\left(\frac{\pi}{8}\right), -2\sin\left(\frac{\pi}{8}\right), -\sqrt{2}\sin\left(\frac{\pi}{8}\right), \cos\left(\frac{\pi}{8}\right)]$ $h_4 = \frac{1}{8}[1, 0, -2, 0, 1]$, $h_5 = \frac{1}{8}[1, -2, 0, 2, -1]$, $h_6 = \frac{\sqrt{2}}{8}\sin\left(\frac{\pi}{8}\right)[1, -\sqrt{2}, 0, \sqrt{2}, -1]$, $h_7 = \frac{\sqrt{2}}{8}[\sin\left(\frac{\pi}{8}\right), -\sqrt{2}\cos\left(\frac{\pi}{8}\right), -2\cos\left(\frac{\pi}{8}\right), -\sqrt{2}\cos\left(\frac{\pi}{8}\right), \sin\left(\frac{\pi}{8}\right)]$
- The boundary elements need to add.

- At a level ℓ, Assume the the coefficient sequence of scaling functions is $\mathbf{c} = [c_1, \cdots, c_N], N \geqslant 5$. When N is odd, we choose the framelets with $L = 3$ and when it is even, we choose them with $L = 4$.
- If $1 < N < 5$, then we use the Haar do construct the wavelet and scaling function.
- The result tight frame within the folder contains only one scaling function.

To decompose the data in a tree by tight frame, we introduce the following:

Definition

Let $\mathcal{T}(X)$ be a data tree on the space $(X, d\mu_0)$, where $d\mu_0 = m^{(0)}dx$ and $m^{(0)}$ is a measure function. Assume also $\mathcal{T}(X)$ has L levels: $S_L \lhd S_{L-1} \lhd \cdots \lhd S_1 \lhd S_0$. Then the measure function $m^{(\ell)}$ on $(S_\ell, d\mu_\ell)$ is defined as

$$m^{(\ell)}(X_k^\ell) = \sum_{X_j^{\ell-1} \subset X_k^\ell} m^{\ell-1}(X_j^{\ell-1}),$$

and the set $\{m^{(0)}, \cdots, m^{(L)}\}$ is called a hierarchical measures on the tree $\mathcal{T}(X)$.

An example of hierarchical measures

Example

Let $m^{(0)}$ be the uniform measure such that $m^{(0)}(x) = 1, x \in X$. Then $m^{(\ell)}(X_j^\ell) = |X_j^\ell|$. It can be normalized to pmf by setting $p^{(\ell)}(X_j^\ell) = \frac{|X_j^\ell|}{|X|}$.

1. Within each folder, construct the tight frame as described above.
2. For cross-level folders, we make the tight frame w.r.t. the measure m. Let $\{\eta_j\}_{j=1}^{n}$ be an tight frame of $L^2(X, dx)$. Write $\tilde{\eta}_j = \eta_j/\sqrt{m}$. Then $\{\tilde{\eta}_j\}_{j=1}^{n}$ is an tight of $L^2(X, d\mu)$.

1. Within each folder, construct the tight frame as described above.
2. For cross-level folders, we make the tight frame w.r.t. the measure m. Let $\{\eta_j\}_{j=1}^n$ be an tight frame of $L^2(X, dx)$. Write $\tilde{\eta}_j = \eta_j/\sqrt{m}$. Then $\{\tilde{\eta}_j\}_{j=1}^n$ is an tight of $L^2(X, d\mu)$.
3. To compute the coefficients of the tight frame on $L^2(X, d\mu)$, we use the formula:

$$\langle fm, \eta_j \rangle = \langle f, \tilde{\eta}_j \rangle_m.$$

Section 4. Wavelet representations of functions on data set

Why do we need wavelet representation?

1. It works on a wide-range of data sets and avoids to treat the high-dimensional data directly. (No curse of dimensionality).

Why do we need wavelet representation?

1. It works on a wide-range of data sets and avoids to treat the high-dimensional data directly. (No curse of dimensionality).
2. It needn't the spatial operators that work only on the data sets in $\mathbb{R}^D$.

Why do we need wavelet representation?

1. It works on a wide-range of data sets and avoids to treat the high-dimensional data directly. (No curse of dimensionality).
2. It needn't the spatial operators that work only on the data sets in $\mathbb{R}^D$.
3. It provides sparse representations of the functions such as compactly supported functions, piecewise constant functions, zero-moment functions, and so on.
4. The optimization models based on wavelets usually have simple structure and lead to a fast algorithm.

Let the data tree on X be given:

$$X = X_1^L \supset \{X_1^{L-1}, \cdots, X_{n_1}^{L-1}\} \supset \cdots \supset \{X_1^0, \cdots, X_n^0\},$$

where $X_j^0 = \{\mathbf{x}_j\}$. Assume that the wavelet o.n. basis or the tight wavelet frame is constructed. Let $f \in L^2(X)$. We may apply the classical Mallat's pyramid algorithm to compute the wavelet coefficients of f.

- As the initial, we set $\mathbf{c} = [c_1, \cdots, c_n] = [f(\mathbf{x}_1), \cdots, f(\mathbf{x}_n)]$. Then $f = \sum_{j=1}^n c_j \phi_j^0(x)$, where $\phi_j^0(\mathbf{x}_i) = \delta_{i,j}$

Let the data tree on X be given:

$$X = X_1^L \supset \{X_1^{L-1}, \cdots, X_{n_1}^{L-1}\} \supset \cdots \supset \{X_1^0, \cdots, X_n^0\},$$

where $X_j^0 = \{\mathbf{x}_j\}$. Assume that the wavelet o.n. basis or the tight wavelet frame is constructed. Let $f \in L^2(X)$. We may apply the classical Mallat's pyramid algorithm to compute the wavelet coefficients of f.

- As the initial, we set $\mathbf{c} = [c_1, \cdots, c_n] = [f(\mathbf{x}_1), \cdots, f(\mathbf{x}_n)]$. Then $f = \sum_{j=1}^n c_j \phi_j^0(x)$, where $\phi_j^0(\mathbf{x}_i) = \delta_{i,j}$
- At Level 1, assume that $X_1^1 = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ and the Haar o.n. basis is employed. Denote by $\mathbf{c}_1 = [c_1, \cdots, c_m]$. Then $\mathbf{c}_{1,1} = (\downarrow 2)\mathbf{c}_1 * h_0$, $\mathbf{d}_{1,1} = (\downarrow 2)\mathbf{c}_1 * h_1$ and $\mathbf{c}_{1,2} = (\downarrow 2)\mathbf{c}_{1,1} * h_0$, $\mathbf{d}_{1,2} = (\downarrow 2)\mathbf{c}_{1,1} * h_1$

Compute the wavelet coefficients via pyramid algorithm

Let the data tree on X be given:

$$X = X_1^L \supset \{X_1^{L-1}, \cdots, X_{n_1}^{L-1}\} \supset \cdots \supset \{X_1^0, \cdots, X_n^0\},$$

where $X_j^0 = \{\mathbf{x}_j\}$. Assume that the wavelet o.n. basis or the tight wavelet frame is constructed. Let $f \in L^2(X)$. We may apply the classical Mallat's pyramid algorithm to compute the wavelet coefficients of f.

- As the initial, we set $\mathbf{c} = [c_1, \cdots, c_n] = [f(\mathbf{x}_1), \cdots, f(\mathbf{x}_n)]$. Then $f = \sum_{j=1}^n c_j \phi_j^0(x)$, where $\phi_j^0(\mathbf{x}_i) = \delta_{i,j}$
- At Level 1, assume that $X_1^1 = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ and the Haar o.n. basis is employed. Denote by $\mathbf{c}_1 = [c_1, \cdots, c_m]$. Then $\mathbf{c}_{1,1} = (\downarrow 2)\mathbf{c}_1 * h_0$, $\mathbf{d}_{1,1} = (\downarrow 2)\mathbf{c}_1 * h_1$ and $\mathbf{c}_{1,2} = (\downarrow 2)\mathbf{c}_{1,1} * h_0$, $\mathbf{d}_{1,2} = (\downarrow 2)\mathbf{c}_{1,1} * h_1$
- The decompositions are repeated, say K_1 times, until $\mathbf{c}_{1,K_1}$ is reduced to a single value.

Let the data tree on X be given:

$$X = X_1^L \supset \{X_1^{L-1}, \cdots, X_{n_1}^{L-1}\} \supset \cdots \supset \{X_1^0, \cdots, X_n^0\},$$

where $X_j^0 = \{\mathbf{x}_j\}$. Assume that the wavelet o.n. basis or the tight wavelet frame is constructed. Let $f \in L^2(X)$. We may apply the classical Mallat's pyramid algorithm to compute the wavelet coefficients of f.

- As the initial, we set $\mathbf{c} = [c_1, \cdots, c_n] = [f(\mathbf{x}_1), \cdots, f(\mathbf{x}_n)]$. Then $f = \sum_{j=1}^n c_j \phi_j^0(x)$, where $\phi_j^0(\mathbf{x}_i) = \delta_{i,j}$
- At Level 1, assume that $X_1^1 = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ and the Haar o.n. basis is employed. Denote by $\mathbf{c}_1 = [c_1, \cdots, c_m]$. Then $\mathbf{c}_{1,1} = (\downarrow 2)\mathbf{c}_1 * h_0$, $\mathbf{d}_{1,1} = (\downarrow 2)\mathbf{c}_1 * h_1$ and $\mathbf{c}_{1,2} = (\downarrow 2)\mathbf{c}_{1,1} * h_0$, $\mathbf{d}_{1,2} = (\downarrow 2)\mathbf{c}_{1,1} * h_1$
- The decompositions are repeated, say K_1 times, until $\mathbf{c}_{1,K_1}$ is reduced to a single value.
- Repeat the steps above for $[c_{1,K_1}, \cdots, c_{n_{L-1},K_{n_{L-1}}}]$ and so on.

Remark

- The similar algorithm is available for tight wavelet frame too.

Remark

- The similar algorithm is available for tight wavelet frame too.
- The reconstruction of f from its wavelet coefficients is also similar to the classical pyramid algorithm.

Remark

- The similar algorithm is available for tight wavelet frame too.
- The reconstruction of f from its wavelet coefficients is also similar to the classical pyramid algorithm.
- In the wavelet representation $f = c_L\phi^L + \sum d_{\ell,k,j}\psi_{\ell,k,j}$, c_0 is the average of f: $c_0 = \frac{1}{\sqrt{n}}\sum_{j=1}^{n} f(\mathbf{x}_j)$. We denote by W_f for the vector of wavelet coefficients of f (excluding c_L).

Ref. [M. Gavish, B. Nadler, R.R. Coifman, 2010]

Definition

For each subset $S \subset X$, define $\rho(S) = |S|/|X|$. For $\mathbf{x}, \mathbf{y} \in X$, denote by $S(\mathbf{x}, \mathbf{y})$ the smallest folder in the tree $\mathcal{T}(X)$ that contains both $\mathbf{x}$ and $\mathbf{y}$. Then the tree distance of $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$d_{\mathcal{T}}(\mathbf{x}, \mathbf{y}) = \begin{cases} \rho(S(\mathbf{x}, \mathbf{y})), & \mathbf{x} \neq \mathbf{y}, \\ 0 & \mathbf{x} = \mathbf{y}. \end{cases}$$

For $0 < \alpha < 1$, a function $f \in L^2(X)$ is called α-Hölder continuous w.r.t. $\mathcal{T}$ (denoted by $f \in H^\alpha(\mathcal{T})$ if

$$|f(\mathbf{x}) - f(\mathbf{y})| \leqslant C d_{\mathcal{T}}^{\alpha}(\mathbf{x}, \mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y}, \in X.$$

Theorem

Assume $f \in \mathcal{H}^\alpha(\mathcal{T})$ *and* $\psi_j^{(\ell,k)}$ *is the wavelet at* ℓ*-level with* $\mathrm{supp}(\psi_j^{(\ell,k)}) \subset X_k^\ell$*. Then*

$$\langle f, \psi_j^{(\ell,k)} \rangle \leqslant C\rho(X_k^\ell)^{\alpha+1/2}.$$

On the other hand, if the inequality above holds for all wavelets $\psi_j^{(\ell,k)}$*, then* $f \in \mathcal{H}^\alpha(\mathcal{T})$*.*

Exponential decay of wavelet coefficients

Theorem

Assume $f \in \mathcal{H}^\alpha(\mathcal{T})$ *and* $\psi_j^{(\ell,k)}$ *is the wavelet at* ℓ*-level with* $\mathrm{supp}(\psi_j^{(\ell,k)}) \subset X_k^\ell$. *Then*

$$\langle f, \psi_j^{(\ell,k)} \rangle \leqslant C\rho(X_k^\ell)^{\alpha+1/2}.$$

On the other hand, if the inequality above holds for all wavelets $\psi_j^{(\ell,k)}$, *then* $f \in \mathcal{H}^\alpha(\mathcal{T})$.

Corollary

Let $\mathcal{T}(X)$ *be a balanced tree with the upper bound* $\overline{B}$. *Assume* $f \in H^\alpha(\mathcal{T})$ *and* $\psi_j^{(\ell,k)}$ *is the wavelet at* ℓ*-level with* $\mathrm{supp}(\psi_j^{(\ell,k)}) \subset X_k^\ell$. *Then*

$$\langle f, \psi_j^{(\ell,k)} \rangle \leqslant C\overline{B}^{(\alpha+1/2)(\ell-1)}.$$

Let f be a binary classification function: $X \to \{-1, 1\}$, which is known on the labeled set $S \subset X : f(\mathbf{x}) = y$. Then the classifier can be computed as the minimum of the following:

$$f = \arg\min_{f \in \mathcal{H}(\mathcal{T})} \sum_{\mathbf{x} \in S} \|f(\mathbf{x}) - y\|^2 + \lambda \|\mathbf{W}_f\|_1.$$

Let f be a binary classification function: $X \to \{-1, 1\}$, which is known on the labeled set $S \subset X : f(\mathbf{x}) = y$. Then the classifier can be computed as the minimum of the following:

$$f = \arg\min_{f \in \mathcal{H}(\mathcal{T})} \sum_{\mathbf{x} \in S} \|f(\mathbf{x}) - y\|^2 + \lambda \|\mathbf{W}_f\|_1.$$

We denote by M be the matrix representing the wavelet transform on X, by M^T the inverse wavelet transform matrix. Let $S = [\mathbf{x}_{j_1}, \cdots, \mathbf{x}_{j_s}]$ and $P_s = [\vec{e}_{j_1}; \cdots; \vec{e}_{j_s}]$ be the landmark extraction. Then the minimization problem above becomes the following:

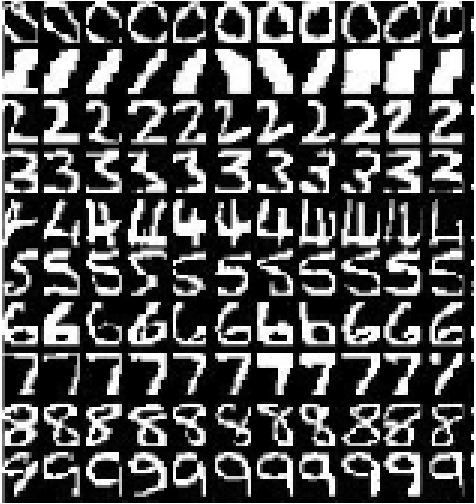
$$W_f = \arg\min_{W_f} (P_s M^T W_f - \mathbf{y})^T (P_s M^T W_f - \mathbf{y}) + \lambda \|\mathbf{W}_f\|_1,$$

which leads to a wavelet threshold algorithm [see Chui and Wang, 2007]

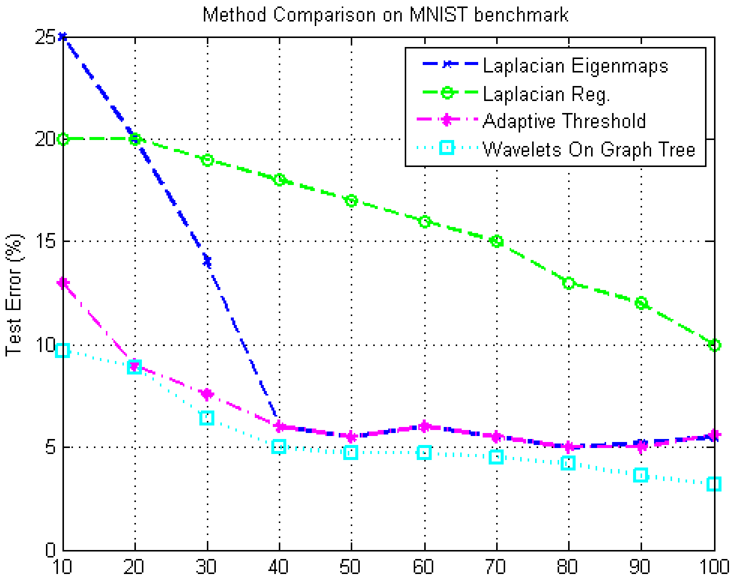
Let $g(\mathbf{x}) = f(\mathbf{x}) + n(\mathbf{x})$, where $n(\mathbf{x})$ is a noise on X. Then a simple denoising algorithm is given by

$$f = \arg\min_{f \in \mathcal{H}(\mathcal{T})} \sum_{\mathbf{x} \in X} \|\mathbf{W}_f - \mathbf{W}_g\|^2 + \lambda \|\mathbf{W}_f\|_1.$$

A set of test digits is given randomly. Only a small number of the test digits are labeled.

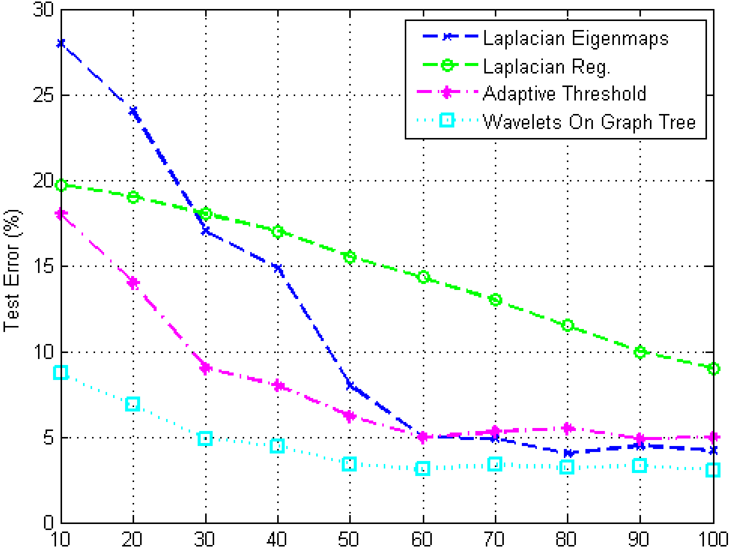


- We select 1000 handwritten digits at random from MNIST, where 200 samples are for each of the digits 8, 3, 4, 5, 7. Digits 8 were in a class, and others are in another class.
- We test the algorithm for the labeled set size $|S| = 10, 20, \cdots, 100$, that is, the label rates are from 1% to 10%.
- We compare our method with three others: **Laplacian Eigenvalues, Laplacian Regression**, and **Adaptive Threshold**. They do not employ graph tree structure, but are based on manifold learning.



Method Comparison on MNIST benchmark
Test Error (%)
0
5
10
15
20
25
10
20
30
40
50
60
70
80
90
100
Laplacian Eigenmaps
Laplacian Reg.
Adaptive Threshold
Wavelets On Graph Tree

- We select 1500 handwritten digits at random from USPS, where 150 samples are for each of the digits from 0 to 9. Digits 2 and 5 were in a class, and others are in another class.
- We test the algorithm for the labeled set size $|S| = 10, 20, \cdots, 100$, that is, the label rates are from about 0.67% to 6.67%.
- We again compare our method with three others: **Laplacian Eigenvalues, Laplacian Regression**, and **Adaptive Threshold**.


Laplacian Eigenmaps
Laplacian Reg.
Adaptive Threshold
Wavelets On Graph Tree
Test Error (%)
0
5
10
15
20
25
30
10
20
30
40
50
60
70
80
90
100

Experiment on USPS 1500 samples: Error rates (%) of different methods.

Method	$\|X_0\| = 10$	$\|X_0\| = 100$
1-NN	19.82	7.64
SVM	20.03	9.75
MVU + 1-NN	14.88	6.09
LEM + 1-NN	19.14	6.09
QC + CMN	13.61	6.36
Discrete Reg.	16.07	4.68
TSVM	25.20	9.77
SGT	25.36	6.80
Cluster-Kernel	19.41	9.68
Data-Dep. Reg.	17.96	5.10
LDS	17.57	4.96
Laplacian RLS	18.99	4.68
CHM (Normalized)	20.53	7.65
Graph-tree Wavelets	**8.21**	**3.47**

References

- Zuowei Shen, *Wavelet frames and image restorations*, Proc. of ICM, Vol IV,(2010).
- Raymond Chan, S. D. Riemenschneider, Lixin Shen, Zuowei Shen, *Tight Frame: An efficient way for high-resolution image reconstruction*, ACHA **17** (2004).
- R. Coifman and M. Maggioni, *Diffusion wavelets*, ACHA, (2006).
- D.K. Hammond and P. Vandergheynst and R. Gribonval, *Wavelets on graphs via spectral graph theory*, ACHA (2011).
- A. Ng, M. Jordan and Y. Weiss, *On spectral clustering: Analysis and an algorithm*, Advances in Neur. Infor. Proc. Sys. **14** (2001)
- J. Shi and J. Malik, *Normalized Cuts and Image Segmentation*, IEEE Trans. on Pattern Anal. and Machine Intel. **22** (2000)
- J. Wang, "Geometric Structure of Data and Dimensionality Reduction", Springer and Chinese High Edu. Press, 2012.

Last slide

THANK YOU !