Best Basis

(A first look)

With the LOT, we have a very flexible framework for partitioning the real line and analyzing each segment (local trig. series in each interval)

Natural question: Given a signal $f(t)$, what is the "best" partition?

To answer this question, we will restrict ourselves to dyadic partions and a very particular meaning of "best".

Dyadic Partions

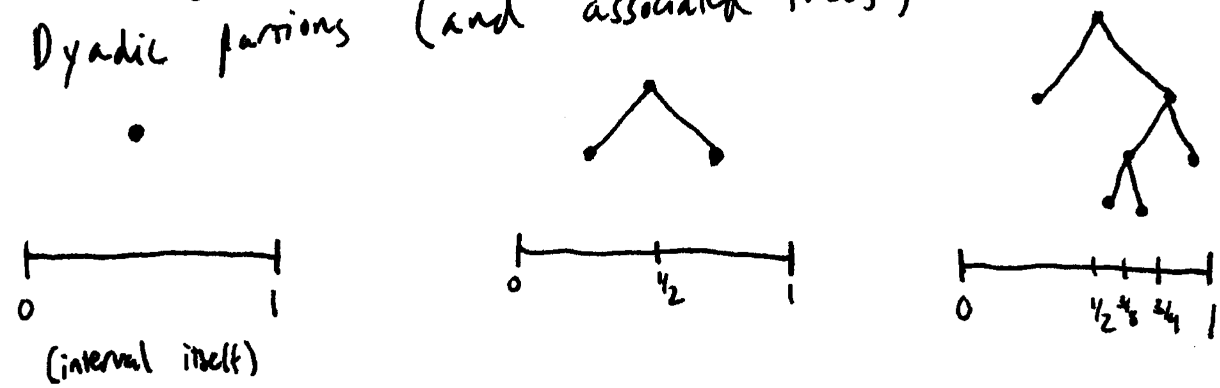
A dyadic partition of an interval is any partition we can generate by recursively splitting subintervals in half.

Associated with each dyadic partition is a binary tree.

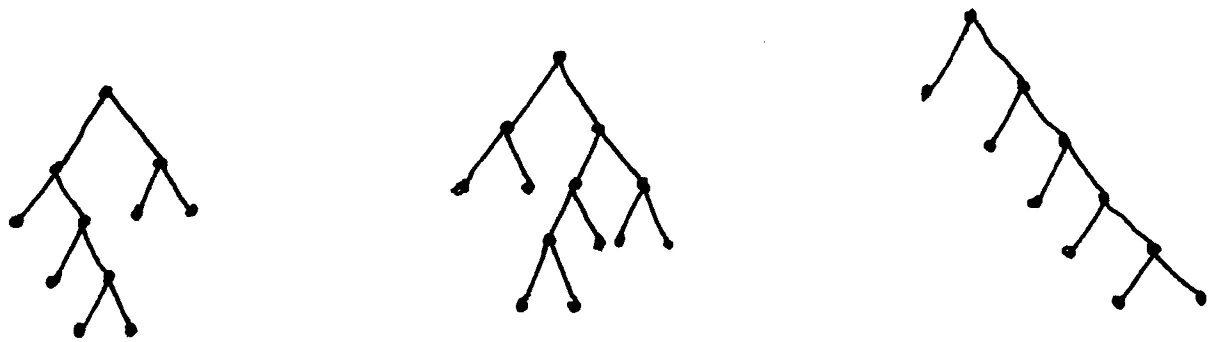
Examples.

Partioning the interval $[0,1]$

Dyadic partions (and associated trees)

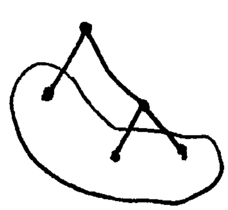


What do the partitions for these trees look like?



Note: To correspond to a dyadic partition each parent in the tree must have either 0 or 2 children.

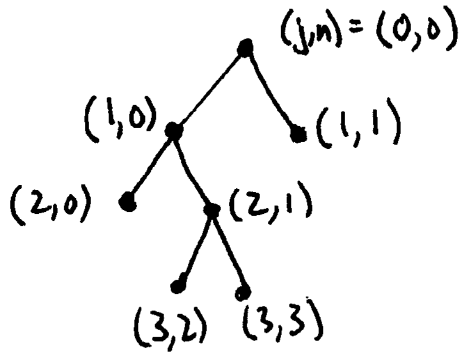
Nodes with 0 children are called leaves.



Each node in a dyadic tree can be indexed by a *scale* j and a *shift* n.

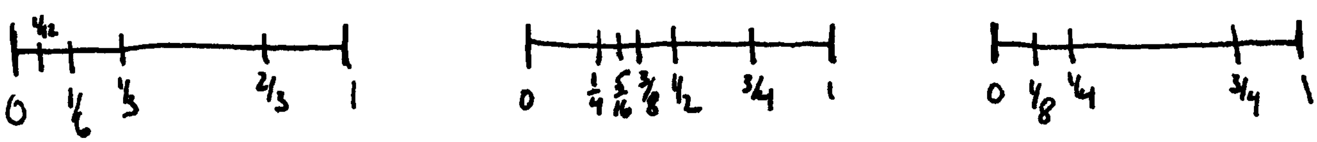
$$0 \le j \le J \qquad J = \text{maximum depth of tree}$$

$$0 \le n \le 2^j - 1$$



The *children* of a node indexed (j,n) are indexed by $(j+1, 2n)$ and $(j+1, 2n+1)$.

Exercise Which of these are *not* dyadic partions ?



Given a binary tree, the endpoints of the corresponding partition of $[0,1]$ are

$$a_{j,n} = n \cdot 2^{-j} \quad \text{for each } \underline{\text{leaf}} \ (j,n)$$

Along with a partition, each (valid) binary tree corresponds to a different orthogonal decomposition of $L_2([0,1])$.

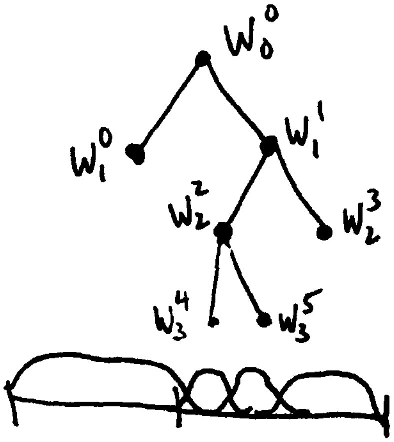
Take $\eta \leq 2^{-J-1}$

For a dyadic interval $[a_{j,n}, a_{j,n+1}] = [n \cdot 2^{-j}, (n+1) \cdot 2^{-j}]$ define the window

$$g_{j,n}(t) = \begin{cases} \beta\left(\frac{t - a_{j,n}}{\eta}\right) & t \in [a_{j,n} - \eta, a_{j,n} + \eta] \\ 1 & t \in [a_{j,n} + \eta, a_{j,n+1} - \eta] \\ \beta\left(\frac{a_{j,n+1} - t}{\eta}\right) & t \in [a_{j,n+1} - \eta, a_{j,n+1} + \eta] \\ 0 & \text{otherwise} \end{cases}$$

and the subspaces W_j^n as before

$\Big(f \in W_j^n \iff f(t)$ can be written $f(t) = g_{j,n}(t) \cdot h(t)$ where $h(t)$ is symmetric around $a_{j,n}$ and anti-symmetric around $a_{j,n+1} \Big)$



We can write

$$L_2([0,1]) = \bigoplus_{(j,n)\in \text{Leaves}(\Upsilon)} W_j^n$$

for any valid binary tree Υ.

Also, it is not hard to see that the children spaces W_{j+1}^{2n}, W_{j+1}^{2n+1} orthogonally decompose the parent space W_j^n. That is

$$W_j^n = W_{j+1}^{2n} \oplus W_{j+1}^{2n+1}$$

We know an orthobasis for each W_j^n, we will denote it

$$\mathcal{B}_j^n = \left\{ g_{n,j}(t) \cdot \sqrt{2^{j+1}} \cdot \cos\left[\pi\left(k+\tfrac{1}{2}\right) \cdot \frac{t-a_{n,j}}{2^{-j}}\right]\right\}_{k\in\mathbb{Z}}$$

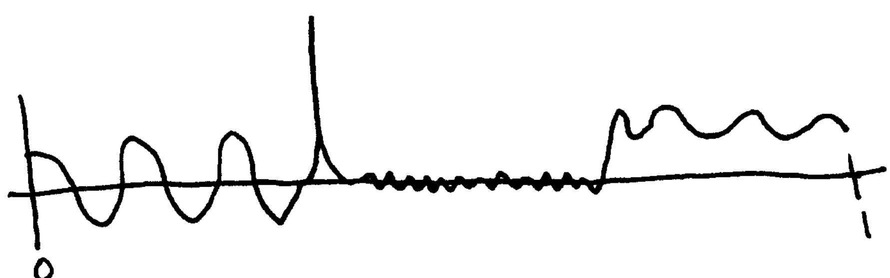
So given a valid binary tree $\mathcal{T}$, we have an orthobasis for $L_2([0,1])$

$$\underbrace{\mathcal{B}_{\mathcal{T}}}_{\text{"orthobasis corresponding to } \mathcal{T}\text{"}} = \bigcup_{(j,n)\in \text{Leaves}(\mathcal{T})} \mathcal{B}_j^n$$

Choosing a dyadic partition

How do we choose which dyadic partition is best for a given signal $f(t)$?

Suppose $f(t)$ looks like this:



What would a "good" partition look like (qualitatively)?

We want an orthobasis $\mathcal{B}_{\mathcal{P}}$ such that the decomposition

$$\{ \langle f, \theta \rangle \}_{\theta \in \mathcal{B}_{\mathcal{P}}}$$

is maximally <u>concentrated</u>.

(i.e. it gives the <u>simplest</u> description of $f(t)$).

To quantify this, we will assign a <u>cost</u> of a basis (partition) using an <u>entropy</u> <u>measure</u>:

$$\mathcal{E}(f, \mathcal{B}_{\mathcal{P}}) = - \sum_{\theta \in \mathcal{B}_{\mathcal{P}}} \frac{|\langle f, \theta \rangle|^2}{\|f\|_2^2} \cdot \log \left(\frac{|\langle f, \theta \rangle|^2}{\|f\|_2^2} \right)$$

$\mathcal{E}(f, \mathcal{B}_{\mathcal{P}})$ is minimum $(=0)$ when ____________

$\mathcal{E}(f, \mathcal{B}_{\mathcal{P}})$ is maximum when ____________

Since the W_j^n are $\perp$ for $(j,n) \in \text{Leaves}(\mathcal{T})$ we can write

$$\mathcal{E}(f, \mathcal{B}_{\mathcal{P}}) = \sum_{(j,n) \in \text{Leaves}(\mathcal{T})} \mathcal{E}(f, \mathcal{B}_j^n)$$

$$\mathcal{B}_j^n = - \sum_{\theta \in \mathcal{B}_j^n} \frac{|\langle f, \theta \rangle|^2}{\|f\|_2^2} \cdot \log \left(\frac{|\langle f, \theta \rangle|^2}{\|f\|_2^2} \right)$$

Associated with each node in $\mathcal{T}$ we have

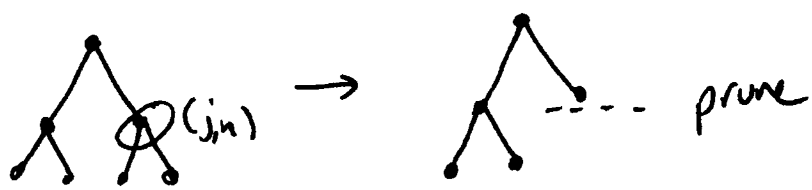
- an <u>index</u> (j,n) (scale, shift)
- a subspace W_j^n
- an orthobasis $\mathcal{B}_j^n$ (for W_j^n)
- a cost $\mathcal{C}(f, \mathcal{B}_j^n)$
  (given a signal $f(t)$)

The goal is given an $f(t)$, find the tree $\mathcal{T}$ that <u>minimizes the sum of the costs at the leaves</u>.

$$\min_{\mathcal{T}} \sum_{(j,n)\in \text{Leaves}(\mathcal{T})} \mathcal{C}(f, \mathcal{B}_j^n)$$

Of course, $\mathcal{E}(f,\mathcal{T})$ will change as we vary $\mathcal{T}$. But because the W_j^n are $\perp$ for $(j,n) \in \text{Leaves}(\mathcal{T})$, <u>collapsing</u>* a pair of children W_{j+1}^{2n} and W_{j+1}^{2n+1} into their parent W_j^n does not affect the contributions at the other leaves.

<u>Collapsing</u> or <u>pruning</u> a tree at node (j,n) means we use $\mathcal{B}_j^n$ to span W_j^n instead of $\mathcal{B}_{j+1}^{2n} \cup \mathcal{B}_{j+1}^{2n+1}$



This allows us to decide:
"Which is better, $\mathcal{B}_j^n$ or $\mathcal{B}_{j+1}^{2n} \cup \mathcal{B}_{j+1}^{2n+1}$?"
<u>independently</u> from decisions in other subtrees.

We can then use a fast <u>dynamic program</u> to find the <u>global</u> <u>minimum</u> of $\mathcal{E}(f, \mathcal{B}_{\mathcal{T}})$ over all valid dyadic trees $\mathcal{T}$.

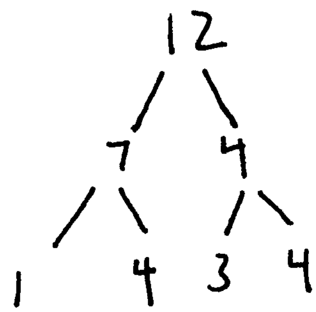
Example:

$J = 2$. Say $\mathcal{E}(f, B_0^0) = 12$

$\mathcal{E}(f, B_1^0) = 7 \quad \mathcal{E}(f, B_1^1) = 4$

$\mathcal{E}(f, B_2^0) = 1 \quad \mathcal{E}(f, B_2^1) = 4 \quad \mathcal{E}(f, B_2^2) = 3$

$\mathcal{E}(f, B_2^3) = 4$



What is the best partition?

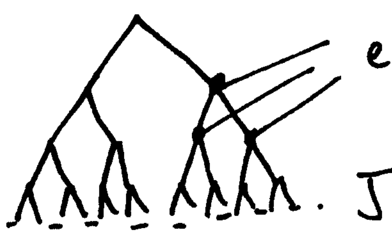
<u>Optimization Algorithm</u> (CART)

Maximum tree depth J

① Initialize the <u>score</u> S_j^n at each node on the tree with

$$S_j^n = \mathcal{C}(f, \mathcal{B}_j^n)$$

Initialize the tree $\mathcal{T}$ as a full binary tree out to level J



each node has a score S_j^n

② For $j = J-1, J-2, \ldots, 0$ and
$n = 0, 1, \ldots, 2^j - 1$ do

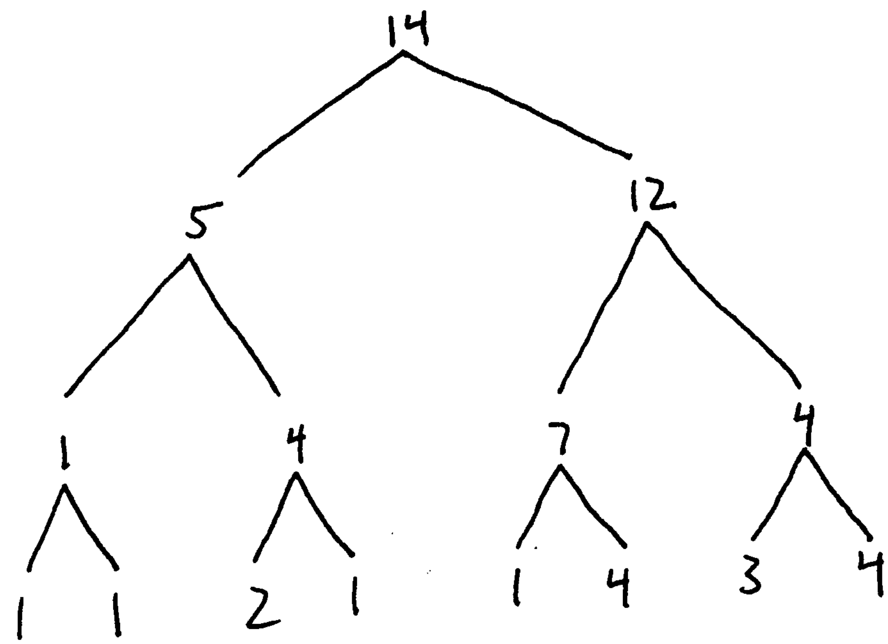
If $S_j^n \leq S_{j+1}^{2n} + S_{j+1}^{2n+1}$ then
prune tree at node (j,n)

else
set $S_j^n = S_{j+1}^n + S_{j+1}^{2n+1}$

After we have traveled all the way up the tree, $\mathcal{T}$ will correspond to the optimal cost/partition/basis

Example.

Suppose the $\mathcal{E}(f, B_j^n)$ are given by



Use CART to find the optimal tree / partition / basis.