Wavelets and Approximation

Ronald DeVore

University of South Carolina

Why Approximation?

- Image Processing

Why Approximation?

- Image Processing
- Image given by a function $f(x)$, $x \in \Omega := [0,1]^2$

Why Approximation?

- Image Processing
- Image given by a function $f(x)$, $x \in \Omega := [0,1]^2$
- Digitized Image: Pixel values: $p_I := \frac{1}{|I|} \int_I f(x)\,dx$

The Following are Forms of Approximation

- Compression: Approximate f by a simpler function $A(f)$ which can be encoded with relatively few bits

The Following are Forms of Approximation

- Compression: Approximate f by a simpler function $A(f)$ which can be encoded with relatively few bits
- Denoising: Given noise corrupted pixels $\bar{p}_I$, approximate f

The Following are Forms of Approximation

- Compression: Approximate f by a simpler function $A(f)$ which can be encoded with relatively few bits
- Denoising: Given noise corrupted pixels $\bar{p}_I$, approximate f
- Deblurring: Given a blurred image $P(f)$ approximate f

Goals of This Talk

- What are the advantages/disadvantages of using wavelets in Approximation

Goals of This Talk

- What are the advantages/disadvantages of using wavelets in Approximation
- Types of wavelet approximation: linear and nonlinear

Goals of This Talk

- What are the advantages/disadvantages of using wavelets in Approximation
- Types of wavelet approximation: linear and nonlinear
- Compare wavelets with other methods of approximation

Goals of This Talk

- What are the advantages/disadvantages of using wavelets in Approximation
- Types of wavelet approximation: linear and nonlinear
- Compare wavelets with other methods of approximation
- Future Directions

Univariate Approximation

- $f(x)$, $x \in [0, 1]$

Univariate Approximation

- $f(x)$, $x \in [0, 1]$
- Approximate f by piecewise constants

Univariate Approximation

- $f(x)$, $x \in [0, 1]$
- Approximate f by piecewise constants
- Types of approximation: Linear, Nonlinear

Univariate Approximation

- $f(x)$, $x \in [0, 1]$
- Approximate f by piecewise constants
- Types of approximation: Linear, Nonlinear
- Select a metric/norm $\|\cdot\|$ to measure distortion

$$\|f\|_{L_p(\Omega)} := \left(\int_\Omega |f(x)|^p \, dx\right)^{1/p}, \quad 0 < p < \infty$$

Univariate Approximation

- $f(x)$, $x \in [0,1]$
- Approximate f by piecewise constants
- Types of approximation: Linear, Nonlinear
- Select a metric/norm $\|\cdot\|$ to measure distortion

$$\|f\|_{L_p(\Omega)} := \left(\int_\Omega |f(x)|^p \, dx\right)^{1/p}, \quad 0 < p < \infty$$

- $\|f\|_{L_\infty} := \sup_{x\in\Omega} |f(x)|, \quad p = \infty$

Linear approximation

- Divide $[0, 1]$ into n equal length intervals

Linear approximation

- Divide $[0, 1]$ into n equal length intervals
- $\Pi_n := \{[\frac{k}{n}, \frac{k+1}{n}), \quad k = 0, \ldots, n-1\}$

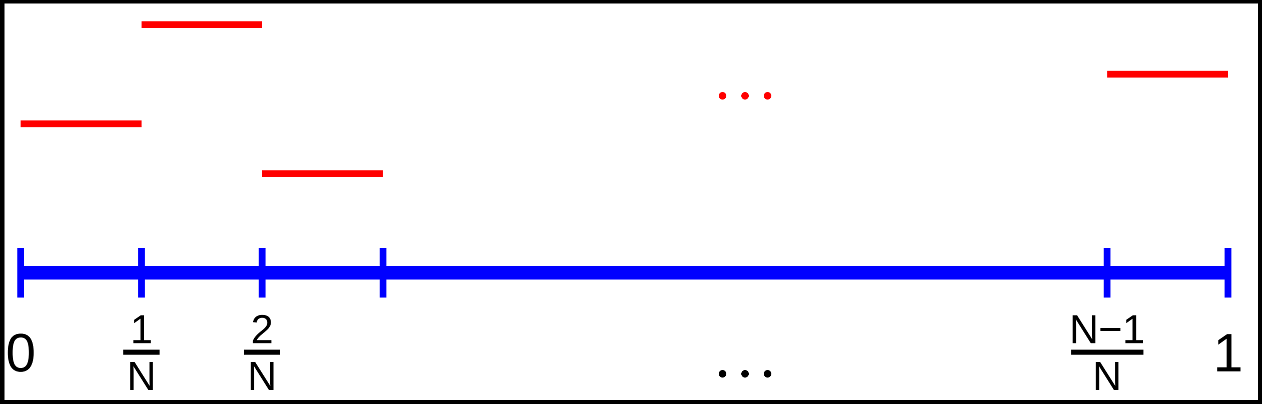
Linear approximation

- Divide $[0, 1]$ into n equal length intervals
- $\Pi_n := \{[\frac{k}{n}, \frac{k+1}{n}), \quad k = 0, \ldots, n-1\}$
- $\mathcal{S}_n := \{S : S \; is \; constant \; on \; each \; I \in \Pi_n\}$

Linear approximation

- Divide $[0,1]$ into n equal length intervals
- $\Pi_n := \{[\frac{k}{n}, \frac{k+1}{n}), \quad k = 0, \dots, n-1\}$
- $\mathcal{S}_n := \{S : S \; is \; constant \; on \; each \; I \in \Pi_n\}$
- $\mathcal{S}_n$ is a linear space

Typical function in $\mathcal{S}_n$



Linear approximation

- Divide $[0,1]$ into n equal length intervals
- $\Pi_n := \{[\frac{k}{n}, \frac{k+1}{n}), \quad k = 0, \ldots, n-1\}$
- $\mathcal{S}_n := \{S : S\ is\ constant\ on\ each\ I \in \Pi_n\}$
- $\mathcal{S}_n$ is a linear space
- Given $f \in L_p[0,1]$, define error

$$E_n(f)_p := \inf_{S \in \mathcal{S}_n} \|f - S\|_{L_p}$$

Nonlinear approximation: Free Knot Splines

- Divide $[0,1]$ into n intervals $I_i = [\xi_i, \xi_{i+1}]$, $0 = \xi_0 < \xi_1 < \cdots < \xi_n = 1$

Nonlinear approximation: Free Knot Splines

- Divide $[0, 1]$ into n intervals $I_i = [\xi_i, \xi_{i+1}]$, $0 = \xi_0 < \xi_1 < \cdots < \xi_n = 1$
- $\Sigma_n := \{S : S \textit{ is constant on each } I_i, i = 1, \ldots, n\}$

Nonlinear approximation: Free Knot Splines

- Divide $[0, 1]$ into n intervals $I_i = [\xi_i, \xi_{i+1}]$, $0 = \xi_0 < \xi_1 < \cdots < \xi_n = 1$
- $\Sigma_n := \{S : S \textit{ is constant on each } I_i, i = 1, \ldots, n\}$
- Σ_n is not a linear space: $S_1, S_2 \in \Sigma_n$ but $S_1 + S_2 \in \Sigma_{2n}$

Nonlinear approximation: Free Knot Splines

- Divide $[0,1]$ into n intervals $I_i = [\xi_i, \xi_{i+1}]$, $0 = \xi_0 < \xi_1 < \cdots < \xi_n = 1$
- $\Sigma_n := \{S : S \textit{ is constant on each } I_i, i = 1, \ldots, n\}$
- Σ_n is not a linear space: $S_1, S_2 \in \Sigma_n$ but $S_1 + S_2 \in \Sigma_{2n}$
- Given $f \in L_p[0,1]$, define error

$$\sigma_n(f)_p := \inf_{S \in \Sigma_n} \|f - S\|_{L_p}$$

Nonlinear Approximation: Adaptive

- For each interval I, $E(I)$ the local L_p error in approximating f by constants

Nonlinear Approximation: Adaptive

- For each interval I, $E(I)$ the local L_p error in approximating f by constants
- Given error tolerance $\epsilon > 0$ generate partition $\mathcal{P}_\epsilon$ such that $E(I) \leq \epsilon$ for all $I \in \mathcal{P}_\epsilon$

Nonlinear Approximation: Adaptive

- For each interval I, $E(I)$ the local L_p error in approximating f by constants
- Given error tolerance $\epsilon > 0$ generate partition $\mathcal{P}_\epsilon$ such that $E(I) \leq \epsilon$ for all $I \in \mathcal{P}_\epsilon$
- I is good if $E(I) \leq \epsilon$

Nonlinear Approximation: Adaptive

- For each interval I, $E(I)$ the local L_p error in approximating f by constants
- Given error tolerance $\epsilon > 0$ generate partition $\mathcal{P}_\epsilon$ such that $E(I) \leq \epsilon$ for all $I \in \mathcal{P}_\epsilon$
- I is good if $E(I) \leq \epsilon$
- I is bad if $E(I) > \epsilon$

Nonlinear Approximation: Adaptive

- Initially if $E([0,1]) \leq \epsilon$ then algorithm terminates and $\mathcal{P}_\epsilon := \{[0,1]\}$

Nonlinear Approximation: Adaptive

- Initially if $E([0,1]) \leq \epsilon$ then algorithm terminates and $\mathcal{P}_\epsilon := \{[0,1]\}$
- if $E([0,1]) > \epsilon$ then set $\mathcal{B}_\epsilon := \{[0,1]\}$ and $\mathcal{G}_\epsilon := \emptyset$

Nonlinear Approximation: Adaptive

- Initially if $E([0,1]) \leq \epsilon$ then algorithm terminates and $\mathcal{P}_\epsilon := \{[0,1]\}$
- if $E([0,1]) > \epsilon$ then set $\mathcal{B}_\epsilon := \{[0,1]\}$ and $\mathcal{G}_\epsilon := \emptyset$
- Recursion: For each $I \in \mathcal{B}_\epsilon$ put child J of I in $\mathcal{G}_\epsilon$ if it is good, put it in $\mathcal{B}_\epsilon$ if it is bad. Remove I from $\mathcal{B}_\epsilon$

Nonlinear Approximation: Adaptive

- Initially if $E([0,1]) \leq \epsilon$ then algorithm terminates and $\mathcal{P}_\epsilon := \{[0,1]\}$
- if $E([0,1]) > \epsilon$ then set $\mathcal{B}_\epsilon := \{[0,1]\}$ and $\mathcal{G}_\epsilon := \emptyset$
- Recursion: For each $I \in \mathcal{B}_\epsilon$ put child J of I in $\mathcal{G}_\epsilon$ if it is good, put it in $\mathcal{B}_\epsilon$ if it is bad. Remove I from $\mathcal{B}_\epsilon$
- Stop when $\mathcal{B}_\epsilon = \emptyset$, $\mathcal{P}_\epsilon := \mathcal{G}_\epsilon$, $N_\epsilon := \#(\mathcal{P}_\epsilon)$

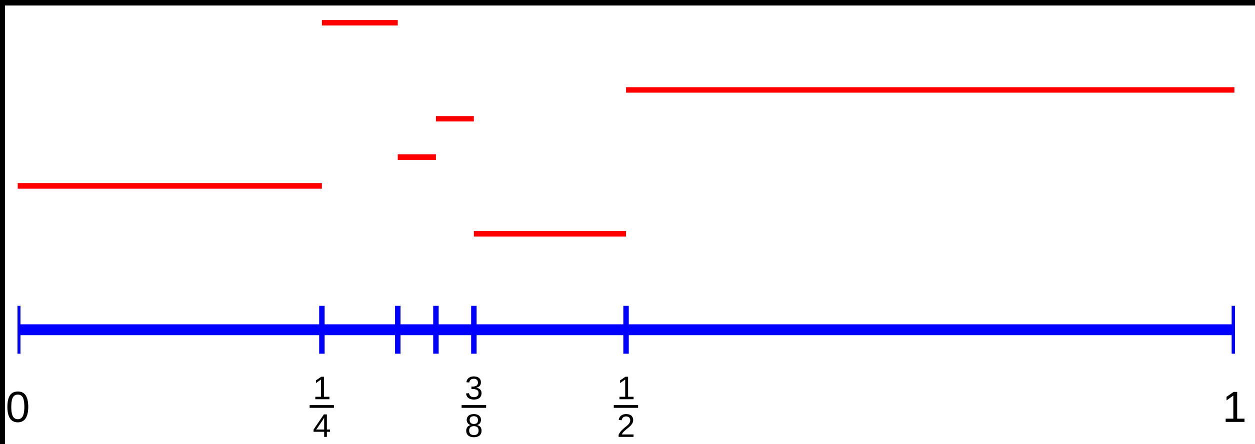
Nonlinear Approximation: Adaptive (continued)

- $A_{N_\epsilon}(f)$ best approximation to f by piecewise constants on $\mathcal{P}_\epsilon$

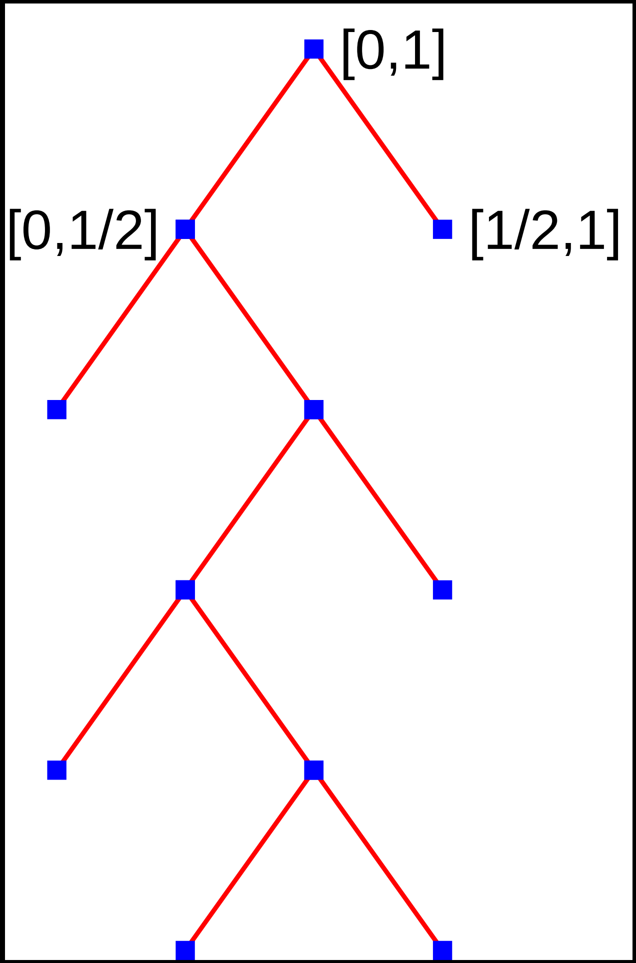
Nonlinear Approximation: Adaptive (continued)

- $A_{N_\epsilon}(f)$ best approximation to f by piecewise constants on $\mathcal{P}_\epsilon$
- $a_n(f)_p := \inf\{\epsilon : N_\epsilon \leq n\}$

Adaptively generated partition

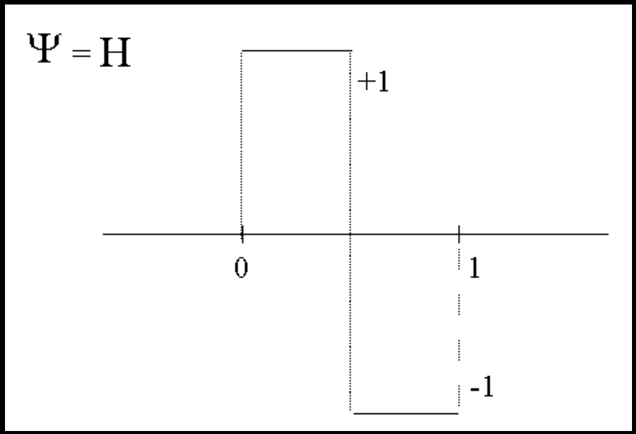


Tree associated to adaptive partition



Wavelets: Haar Wavelet

$$H(x) := \begin{cases} -1, & x \in [0, 1/2) \\ +1, & x \in [1/2, 1] \end{cases},$$



Wavelets: Haar Basis

- $H_I(x) := 2^{j/2}H(2^jx - k)$, $I = [k2^{-j}, (k+1)2^{-j}]$

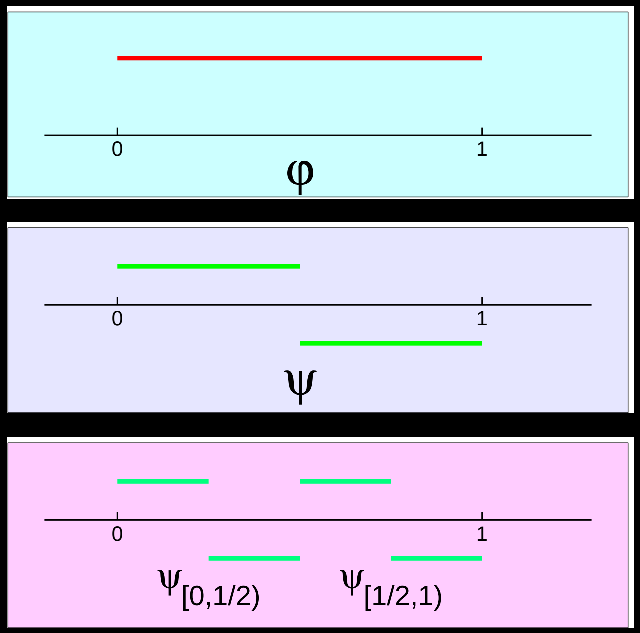
Wavelets: Haar Basis

- $H_I(x) := 2^{j/2} H(2^j x - k)$, $I = [k2^{-j}, (k+1)2^{-j}]$
- $\mathcal{D}_+ := \{I \in \mathcal{D} : |I| \leq 1\}$

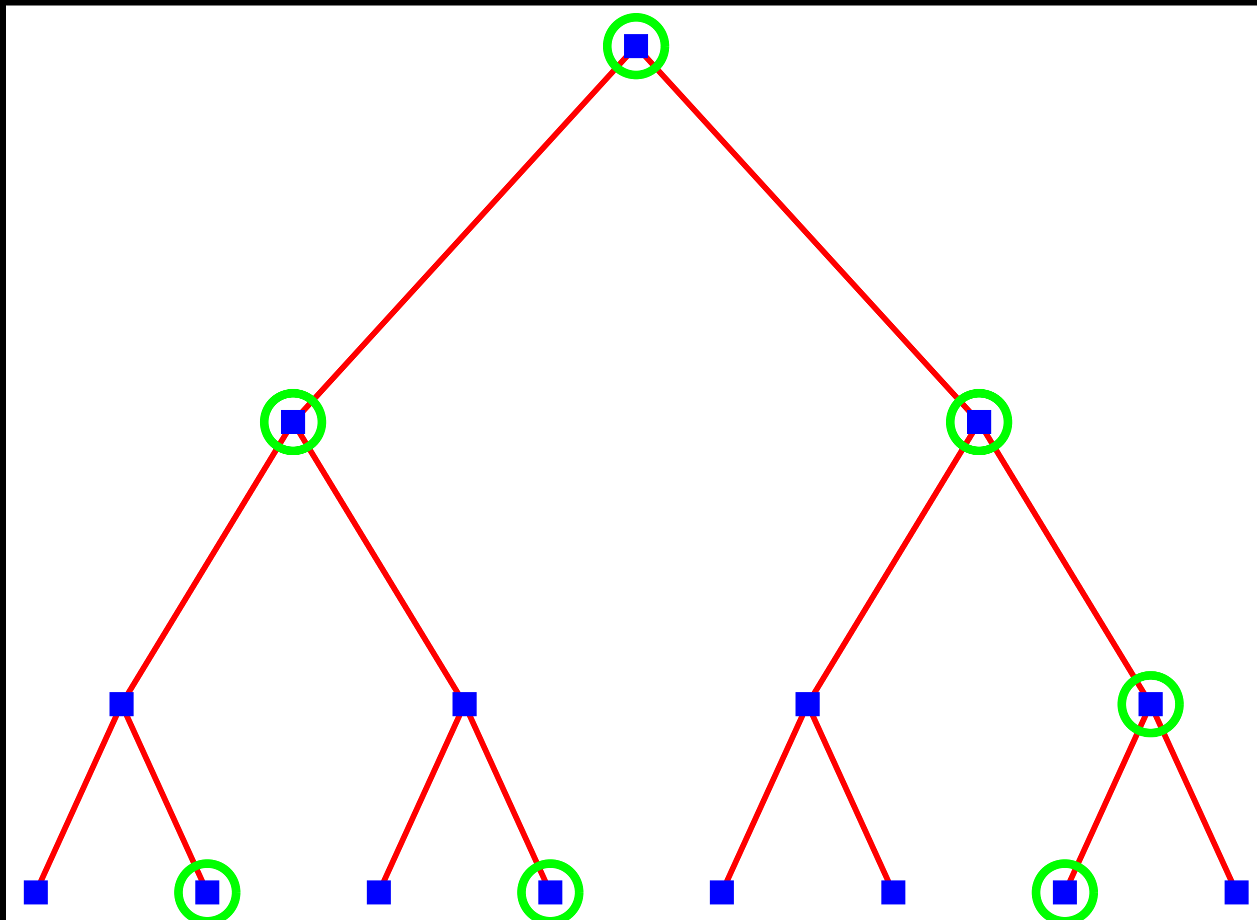
Wavelets: Haar Basis

- $H_I(x) := 2^{j/2} H(2^j x - k)$, $I = [k2^{-j}, (k+1)2^{-j}]$
- $\mathcal{D}_+ := \{I \in \mathcal{D} : |I| \leq 1\}$
- $\{\chi_{[0,1]}\} \cup \{H_I\}_{I \in \mathcal{D}_+}$ is a complete orthonormal system in $L_2[0,1]$

Haar Basis



Wavelet tree



Linear Approximation with Haar Basis

- Natural ordering of dyadic intervals

Linear Approximation with Haar Basis

- Natural ordering of dyadic intervals
- X_n span of first n Haar functions

Linear Approximation with Haar Basis

- Natural ordering of dyadic intervals
- X_n span of first n Haar functions
- X_n is a linear space

Linear Approximation with Haar Basis

- Natural ordering of dyadic intervals
- X_n span of first n Haar functions
- X_n is a linear space
- $E_n^w(f)_p := \inf_{g \in X_n} \|f - g\|_{L_p[0,1]}$

Linear Approximation with Haar Basis

- Natural ordering of dyadic intervals
- X_n span of first n Haar functions
- X_n is a linear space
- $E_n^w(f)_p := \inf_{g \in X_n} \|f - g\|_{L_p[0,1]}$
- This is linear approximation because X_n is a linear space

Nonlinear Approximation with Haar Basis

- n-term approximation

Nonlinear Approximation with Haar Basis

- n-term approximation
- $\Sigma_n := \{S = \sum_{I \in \Lambda} c_I H_I : \ \#(\Lambda) \leq n\}$

Nonlinear Approximation with Haar Basis

- n-term approximation
- $\Sigma_n := \{S = \sum_{I \in \Lambda} c_I H_I : \ \#(\Lambda) \leq n\}$
- Σ_n is not a linear space

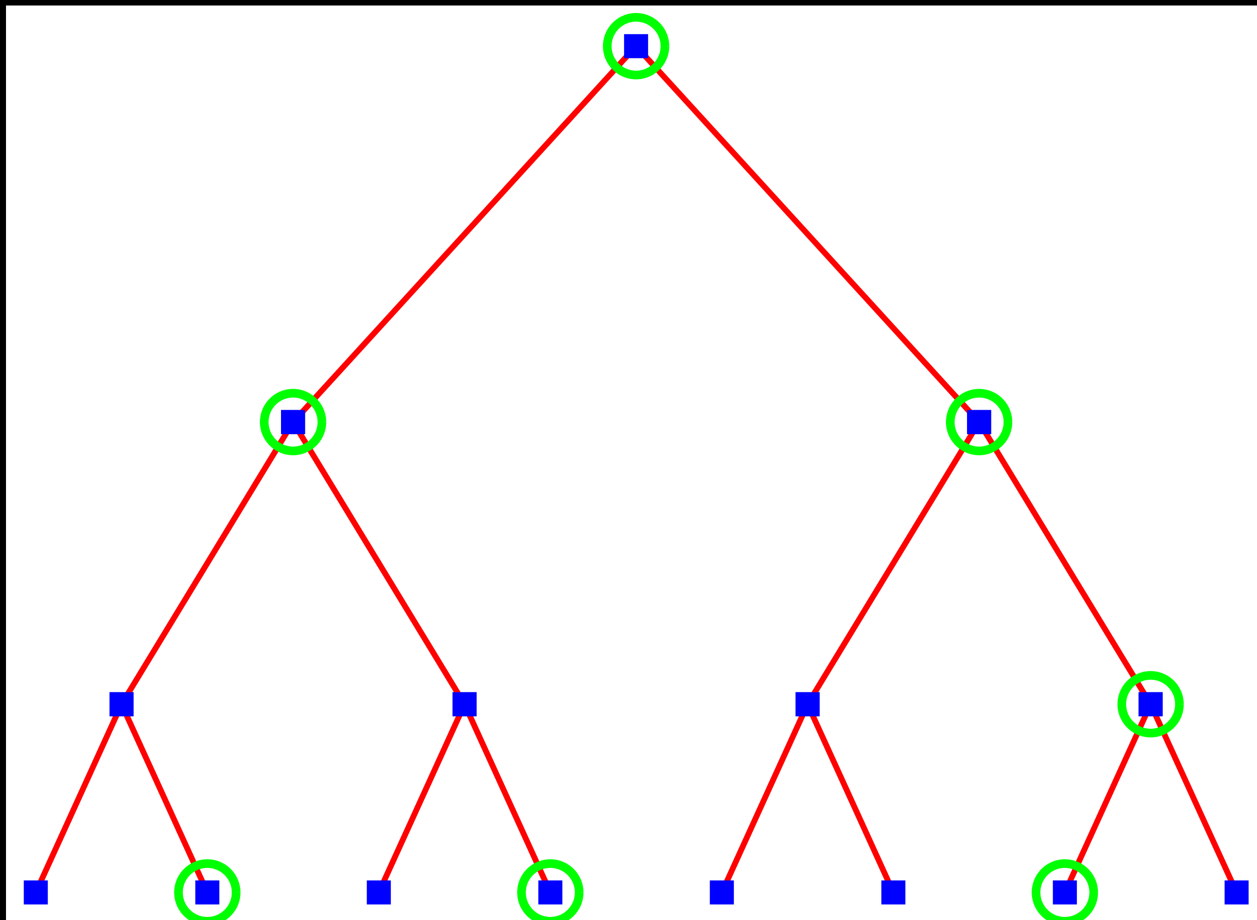
Nonlinear Approximation with Haar Basis

- n-term approximation
- $\Sigma_n := \{S = \sum_{I \in \Lambda} c_I H_I : \#(\Lambda) \leq n\}$
- Σ_n is not a linear space
- $\sigma_n^w(f)_p := \inf_{S \in \Sigma_n} \|f - S\|_{L_p[0,1]}$

Nonlinear Approximation with Haar Basis

- n-term approximation
- $\Sigma_n := \{S = \sum_{I \in \Lambda} c_I H_I : \#(\Lambda) \leq n\}$
- Σ_n is not a linear space
- $\sigma_n^w(f)_p := \inf_{S \in \Sigma_n} \|f - S\|_{L_p[0,1]}$
- This is nonlinear approximation because decisions are made dependent on f

n-term approximation



Tree Approximation with Haar Basis

- Require that the wavelet positions chosen in the approximation lie on a tree with n-nodes

Tree Approximation with Haar Basis

- Require that the wavelet positions chosen in the approximation lie on a tree with n-nodes
- $\Sigma_n^t := \{S = \sum_{I \in \Lambda} c_I H_I : \Lambda \; a \; tree \; \#(\Lambda) \le n\}$

Tree Approximation with Haar Basis

- Require that the wavelet positions chosen in the approximation lie on a tree with n-nodes
- $\Sigma_n^t := \{S = \sum_{I \in \Lambda} c_I H_I : \ \Lambda \ a \ tree \ \#(\Lambda) \leq n\}$
- $\sigma_n^t(f)_p := \inf_{S \in \Sigma_n^t} \|f - S\|_{L_p[0,1]}$

Comparison of these different types of approximation

- Approximation classes: $\alpha > 0$ define $\mathcal{A}^\alpha(L_p,\ linear\ splines)$ as the set of all $f \in L_p[0,1]$ such that

$$E_n(f)_p \leq Cn^{-\alpha}, \quad n \geq 1$$

Comparison of these different types of approximation

- Approximation classes: $\alpha > 0$ define $\mathcal{A}^\alpha(L_p,\ linear\ splines)$ as the set of all $f \in L_p[0,1]$ such that

$$E_n(f)_p \leq Cn^{-\alpha}, \quad n \geq 1$$

- Similarly define $\mathcal{A}^\alpha(L_p)$ for the other forms of approximation

Comparison of these different types of approximation

- Approximation classes: $\alpha > 0$ define $\mathcal{A}^\alpha(L_p,\ linear\ splines)$ as the set of all $f \in L_p[0,1]$ such that

$$E_n(f)_p \leq Cn^{-\alpha}, \quad n \geq 1$$

- Similarly define $\mathcal{A}^\alpha(L_p)$ for the other forms of approximation
- $\mathcal{A}^s_q(L_p)$ finer scaling: same approximation order s

Approximation Classes for Linear Approximation

- Fix the L_p space to measure error

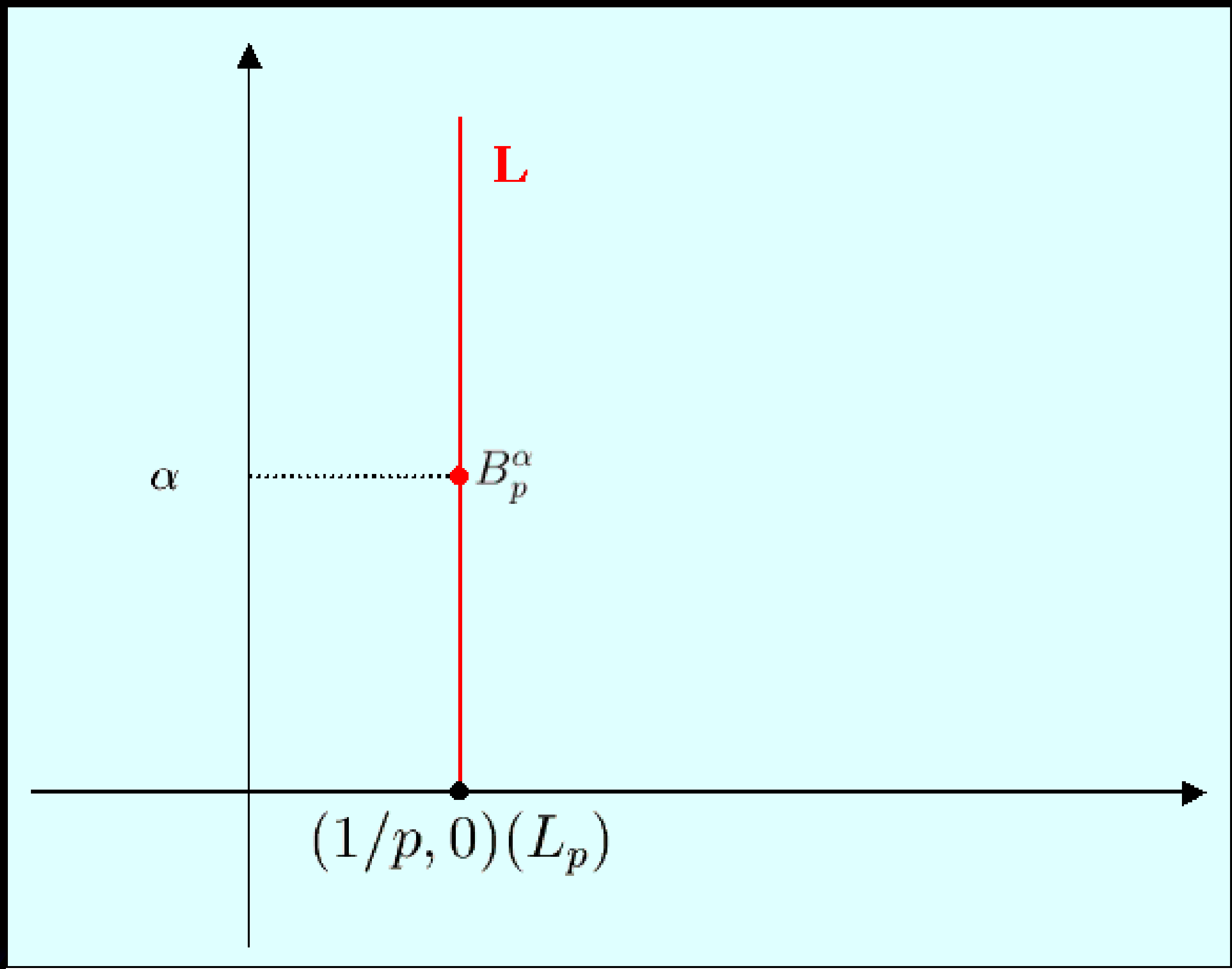
Approximation Classes for Linear Approximation

- Fix the L_p space to measure error
- $A^s(L_p,\ linear) = B^s_\infty(L_p)$

Approximation Classes for Linear Approximation

- Fix the L_p space to measure error
- $A^s(L_p,\ linear) = B^s_\infty(L_p)$
- Proved by Scherer +

Linear Approximation: $\mathcal{A}^s_\infty(L_p)$ Besov space of smoothness s



Approximation Classes for free knot splines and n-term Approximation

- Fix the L_p space to measure error

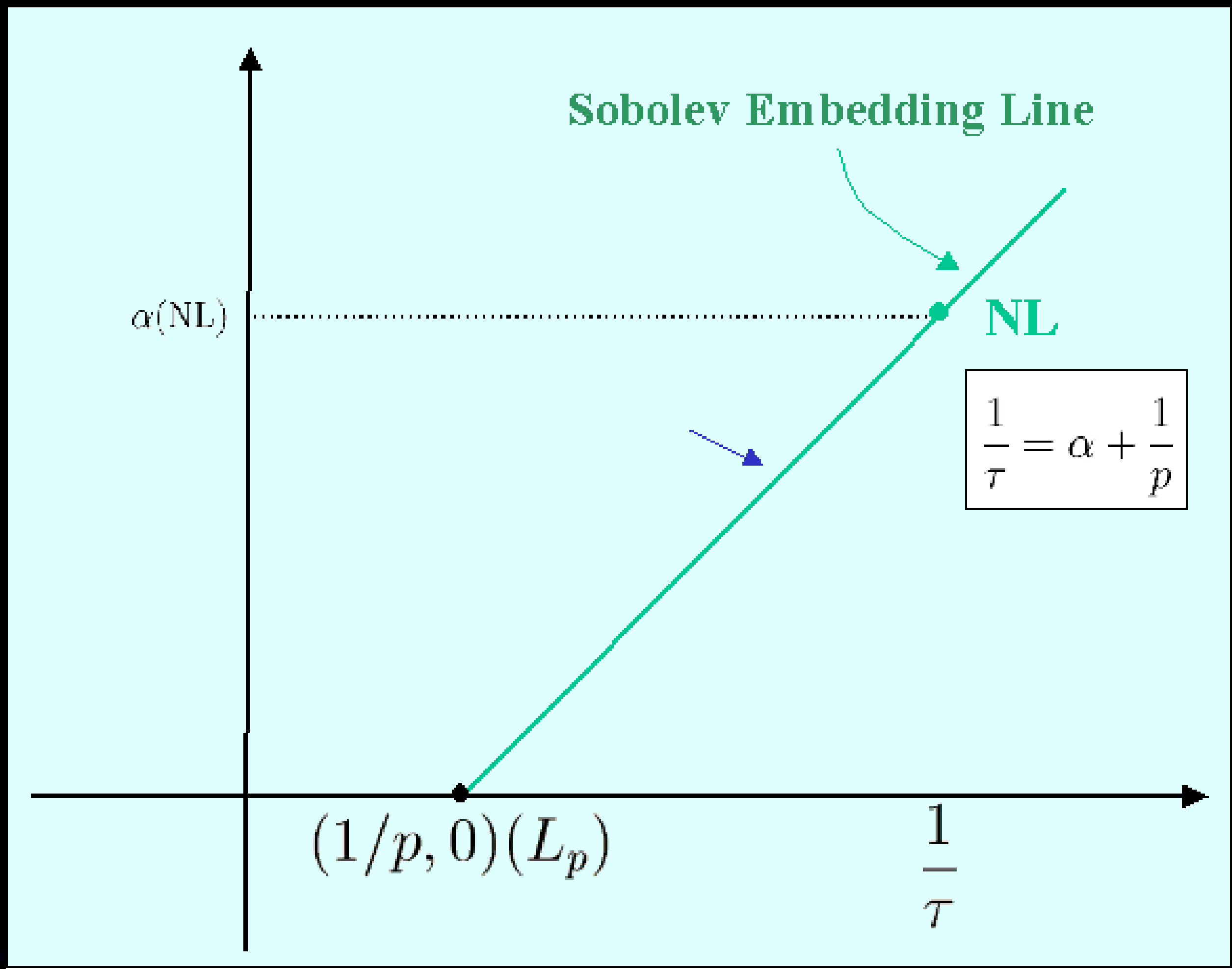
Approximation Classes for free knot splines and n-term Approximation

- Fix the L_p space to measure error
- $A^s_\tau(L_p,\ nonlinear) = B^s_\tau(L_\tau)$, $\frac{1}{\tau} = s + \frac{1}{p}$

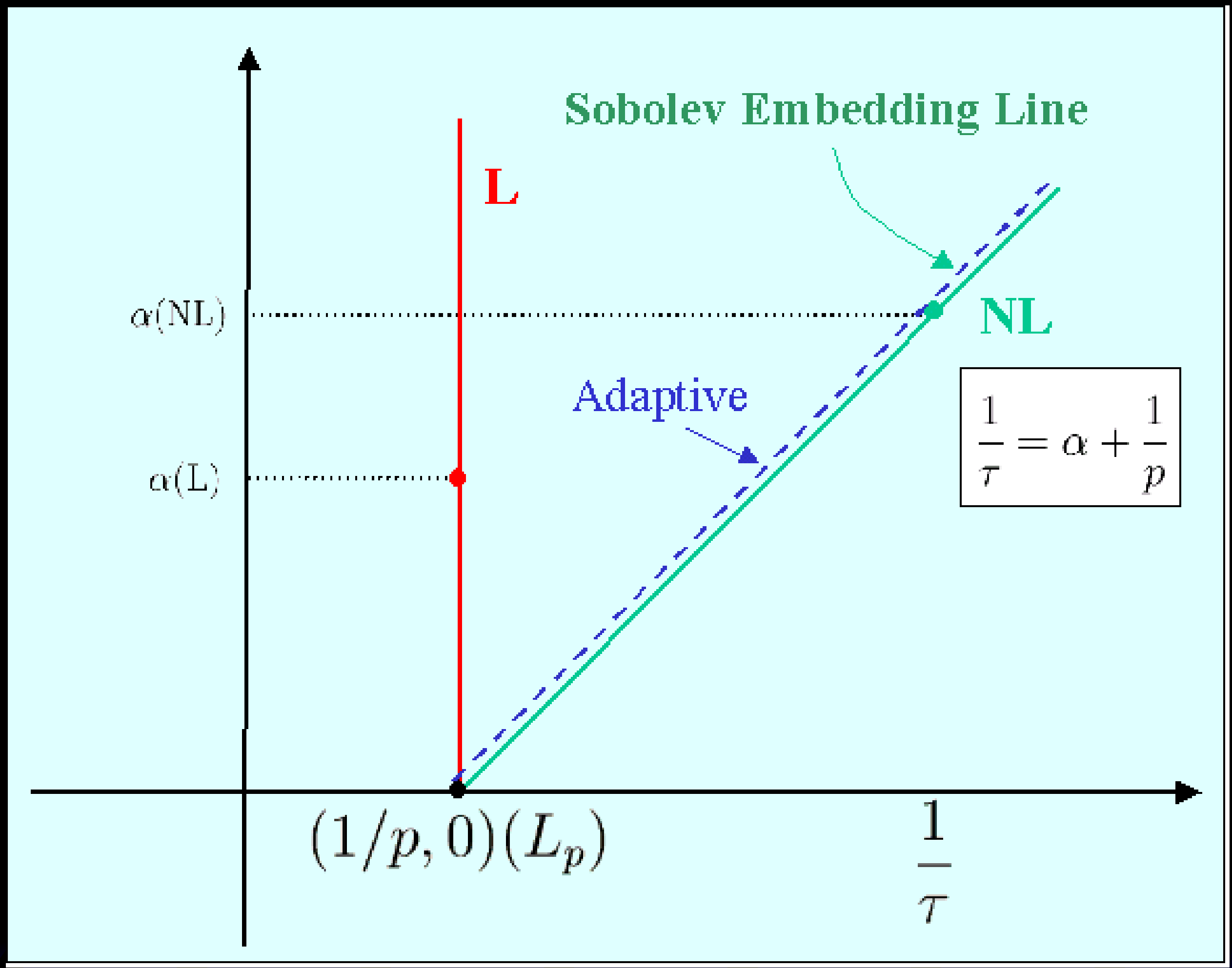
Approximation Classes for free knot splines and n-term Approximation

- Fix the L_p space to measure error
- $A^s_\tau(L_p,\ nonlinear) = B^s_\tau(L_\tau)$, $\frac{1}{\tau} = s + \frac{1}{p}$
- Petrushev, DeVore-Popov (splines); DeVore-Jawerth-Popov (wavelets)

Approximation class for n-term approximation



Adaptive approximation



Example: Approximation in L_∞

- $p = \infty$ approximation order $O(n^{-1})$

Example: Approximation in L_∞

- $p = \infty$ approximation order $O(n^{-1})$
- Linear approximation $f' \in L_\infty$

Example: Approximation in L_∞

- $p = \infty$ approximation order $O(n^{-1})$
- Linear approximation $f' \in L_\infty$
- Nonlinear approximation (free knot splines and n-term wavelet) $f' \in L_1$

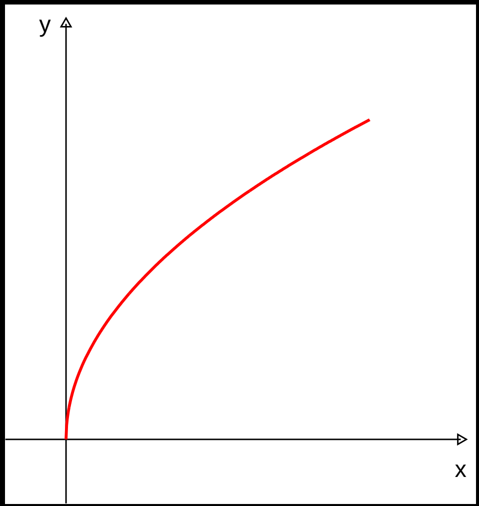
Example: Approximation in L_∞

- $p = \infty$ approximation order $O(n^{-1})$
- Linear approximation $f' \in L_\infty$
- Nonlinear approximation (free knot splines and n-term wavelet) $f' \in L_1$
- Adaptive approximation $f' \in LlogL$: for example $f' \in L_p$ for some $p > 1$

Example: Approximation in L_∞

- $p = \infty$ approximation order $O(n^{-1})$
- Linear approximation $f' \in L_\infty$
- Nonlinear approximation (free knot splines and n-term wavelet) $f' \in L_1$
- Adaptive approximation $f' \in LlogL$: for example $f' \in L_p$ for some $p > 1$
- Tree approximation $f' \in L_p$ for some $p > 1$

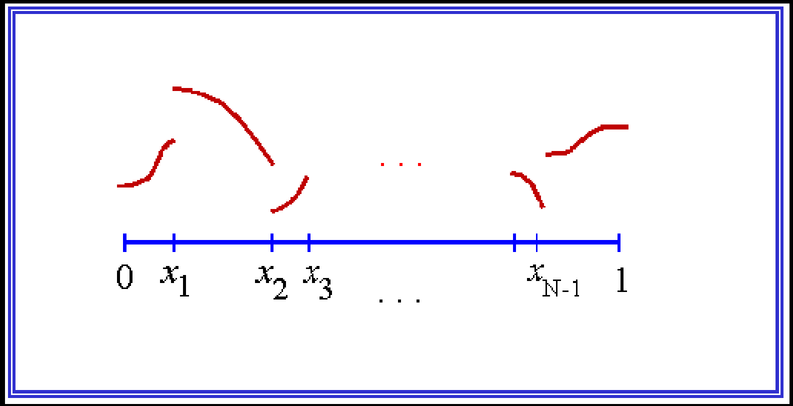
Example: $f(x) = x^\alpha$, $\alpha > -1/p$



$$E_n(f)_p \approx Cn^{-(\alpha+1/p)} \quad \sigma_n(f)_p \leq Cn^{-1}$$

Break points/ wavelets concentrate near singularity at 0

Example: piecewise smooth



$$E_n(f)_p \geq Cn^{-1/p} \quad \sigma_n(f)_p \leq Cn^{-1}$$

Breakpoints/wavelets concentrate near singularities

Multidimensional results

- Linear approximation as before except for curse of dimensionality

Multidimensional results

- Linear approximation as before except for curse of dimensionality
- $\mathcal{A}^{\alpha/d}(L_p) = B^{\alpha}_{\infty}(L_p)$

Multidimensional results

- Linear approximation as before except for curse of dimensionality
- $\mathcal{A}^{\alpha/d}(L_p) = B^{\alpha}_{\infty}(L_p)$
- Nonlinear Approximation?

Multidimensional results

- Linear approximation as before except for curse of dimensionality
- $\mathcal{A}^{\alpha/d}(L_p) = B^{\alpha}_{\infty}(L_p)$
- Nonlinear Approximation?
- No analogue of free knot splines: general triangulation

Multidimensional results

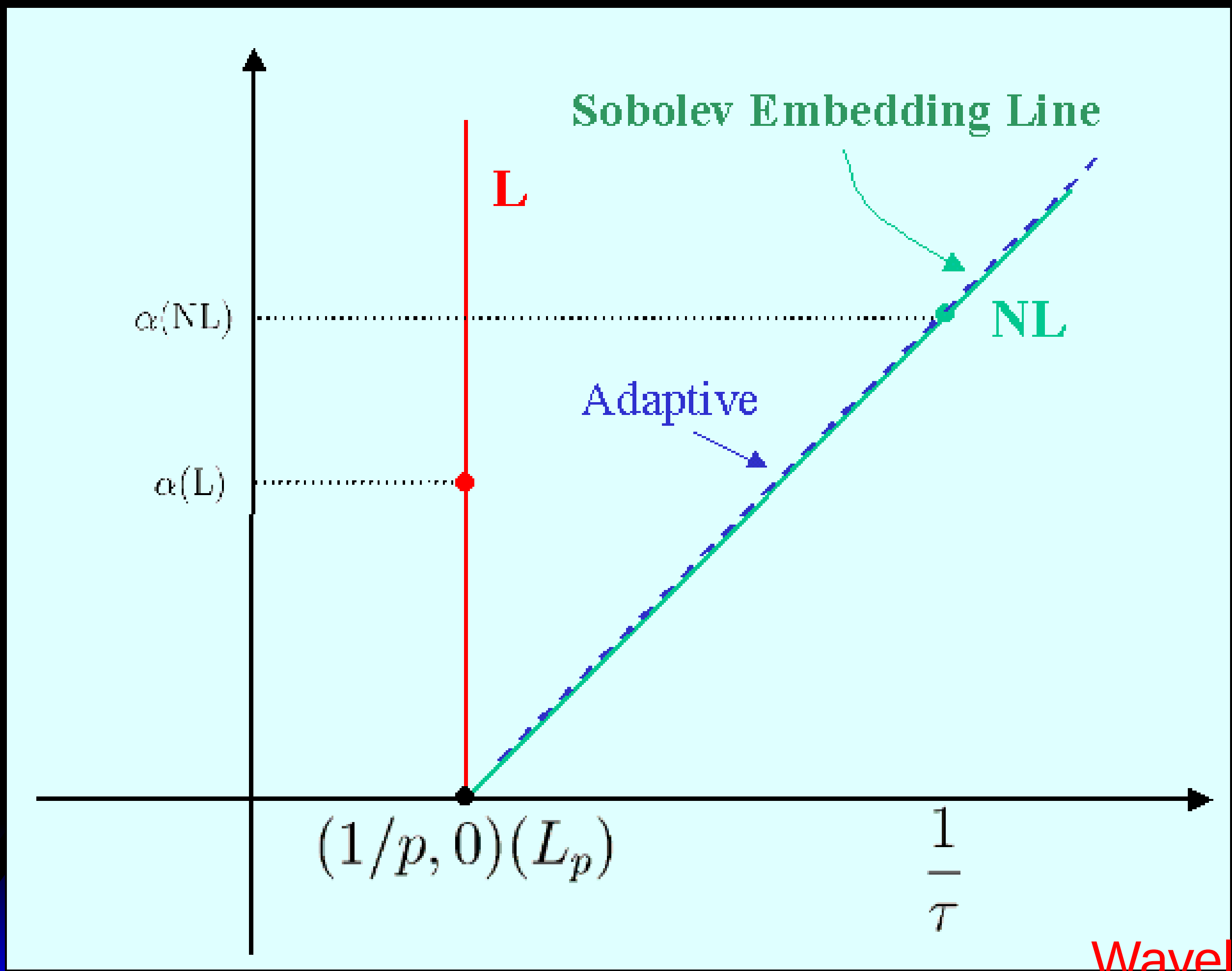
- Linear approximation as before except for curse of dimensionality
- $\mathcal{A}^{\alpha/d}(L_p) = B^{\alpha}_{\infty}(L_p)$
- Nonlinear Approximation?
- No analogue of free knot splines: general triangulation
- n-term approximation using multivariate wavelets the same

Multidimensional results

- Linear approximation as before except for curse of dimensionality
- $\mathcal{A}^{\alpha/d}(L_p) = B^{\alpha}_{\infty}(L_p)$
- Nonlinear Approximation?
- No analogue of free knot splines: general triangulation
- n-term approximation using multivariate wavelets the same
- $\mathcal{A}^{\alpha/d}_{\tau}(wavelets, L_p) = B^{\alpha}_{\tau}(L_p)$, $\frac{1}{\tau} = \frac{\alpha}{d} + \frac{1}{p}$

Multivariate Nonlinear Approximation Classes

Now $\frac{1}{\tau} = \frac{\alpha}{d} + \frac{1}{p}$



Wavelet advantages

- Fix L_p to measure error $1 < p < \infty$

Wavelet advantages

- Fix L_p to measure error $1 < p < \infty$
- Given wavelet decomposition $f = \sum_\lambda a_\lambda(f)\psi_\lambda$

Wavelet advantages

- Fix L_p to measure error $1 < p < \infty$
- Given wavelet decomposition $f = \sum_\lambda a_\lambda(f)\psi_\lambda$
- Choose Λ_n set of indices of the n largest

$$\{\lambda : \|a_\lambda(f)\psi_\lambda\|_{L_p}\}$$

Wavelet advantages

- Fix L_p to measure error $1 < p < \infty$
- Given wavelet decomposition $f = \sum_\lambda a_\lambda(f)\psi_\lambda$
- Choose Λ_n set of indices of the n largest

$$\{\lambda : \|a_\lambda(f)\psi_\lambda\|_{L_p}\}$$

- $\|f - \sum_{\lambda\in\Lambda_n} a_\lambda(f)\psi_\lambda\|_{L_p} \leq C_p\sigma_n(f)_p$

Wavelet advantages

- Fix L_p to measure error $1 < p < \infty$
- Given wavelet decomposition $f = \sum_\lambda a_\lambda(f)\psi_\lambda$
- Choose Λ_n set of indices of the n largest

$$\{\lambda : \|a_\lambda(f)\psi_\lambda\|_{L_p}\}$$

- $\|f - \sum_{\lambda\in\Lambda_n} a_\lambda(f)\psi_\lambda\|_{L_p} \leq C_p\sigma_n(f)_p$
- Wavelet basis is greedy

Wavelet advantages

- Fix L_p to measure error $1 < p < \infty$
- Given wavelet decomposition $f = \sum_\lambda a_\lambda(f)\psi_\lambda$
- Choose Λ_n set of indices of the n largest

$$\{\lambda : \|a_\lambda(f)\psi_\lambda\|_{L_p}\}$$

- $\|f - \sum_{\lambda\in\Lambda_n} a_\lambda(f)\psi_\lambda\|_{L_p} \leq C_p\sigma_n(f)_p$
- Wavelet basis is greedy
- Simple thresholding is a near best strategy

Further Wavelet advantages

- Function spaces are characterized by wavelet coefficients in a simple way

Further Wavelet advantages

- Function spaces are characterized by wavelet coefficients in a simple way
- n-term approximations of images align themselves on trees

Further Wavelet advantages

- Function spaces are characterized by wavelet coefficients in a simple way
- n-term approximations of images align themselves on trees
- Other correlations in positions of big wavelet coefficients

Remarks on Encoding

- Tree with n nodes can be encoded with n bits

Remarks on Encoding

- Tree with n nodes can be encoded with n bits
- Cohen-Dahmen-Daubechies-DeVore (Cohen-Daubechies-Gulleryuz-Orchard) If $f \in B^\alpha_\tau(L_\tau)$, $\frac{1}{\tau} < \frac{\alpha}{2} + \frac{1}{p}$ then f can be approximated to accuracy $C\|f\|_{B^\alpha_\tau(L_\tau)} n^{-\alpha/2}$ and the approximant can be encoded with n bits

Remarks on Wavelet encoders

- EZW, Said-Pearlman, HMM, etc.

Remarks on Wavelet encoders

- EZW, Said-Pearlman, HMM, etc.
- Bits to quantize coefficients

Remarks on Wavelet encoders

- EZW, Said-Pearlman, HMM, etc.
- Bits to quantize coefficients
- Bits to encode positions of wavelets

Remarks on Wavelet encoders

- EZW, Said-Pearlman, HMM, etc.
- Bits to quantize coefficients
- Bits to encode positions of wavelets
- number of quantization bits determined by decay of rearranged wavelet coefficients: the α in Besov regularity

Remarks on Wavelet encoders

- EZW, Said-Pearlman, HMM, etc.
- Bits to quantize coefficients
- Bits to encode positions of wavelets
- number of quantization bits determined by decay of rearranged wavelet coefficients: the α in Besov regularity
- If n coefficients are taken then need at least (and at most) n bits to achieve distortion $n^{-\alpha}$

Remarks on Wavelet encoders

- n-bits to encode a tree with n nodes

Remarks on Wavelet encoders

- n-bits to encode a tree with n nodes
- Better information on correlations may improve on this n

Remarks on Wavelet encoders

- n-bits to encode a tree with n nodes
- Better information on correlations may improve on this n
- Statistical correlations improve on C in estimate $Cn^{=\alpha}$ but not on α

Remarks on Wavelet encoders

- n-bits to encode a tree with n nodes
- Better information on correlations may improve on this n
- Statistical correlations improve on C in estimate $Cn^{=\alpha}$ but not on α
- Besov smoothness is exact predictor of asymptotic performance of wavelet encoders

Remarks on Wavelet encoders

- n-bits to encode a tree with n nodes
- Better information on correlations may improve on this n
- Statistical correlations improve on C in estimate $Cn^{=\alpha}$ but not on α
- Besov smoothness is exact predictor of asymptotic performance of wavelet encoders
- Models vs. Encoders

Remarks on Wavelet encoders

- n-bits to encode a tree with n nodes
- Better information on correlations may improve on this n
- Statistical correlations improve on C in estimate $Cn^{=\alpha}$ but not on α
- Besov smoothness is exact predictor of asymptotic performance of wavelet encoders
- Models vs. Encoders

Remarks on Wavelet encoders

- n-bits to encode a tree with n nodes
- Better information on correlations may improve on this n
- Statistical correlations improve on C in estimate $Cn^{=\alpha}$ but not on α
- Besov smoothness is exact predictor of asymptotic performance of wavelet encoders
- Models vs. Encoders

Remarks on Wavelet encoders

- n-bits to encode a tree with n nodes
- Better information on correlations may improve on this n
- Statistical correlations improve on C in estimate $Cn^{-\alpha}$ but not on α
- Besov smoothness is exact predictor of asymptotic performance of wavelet encoders
- Models vs. Encoders

Drawbacks to wavelets in multidimensions

- Wavelets isotropic and oriented to coordinate axis

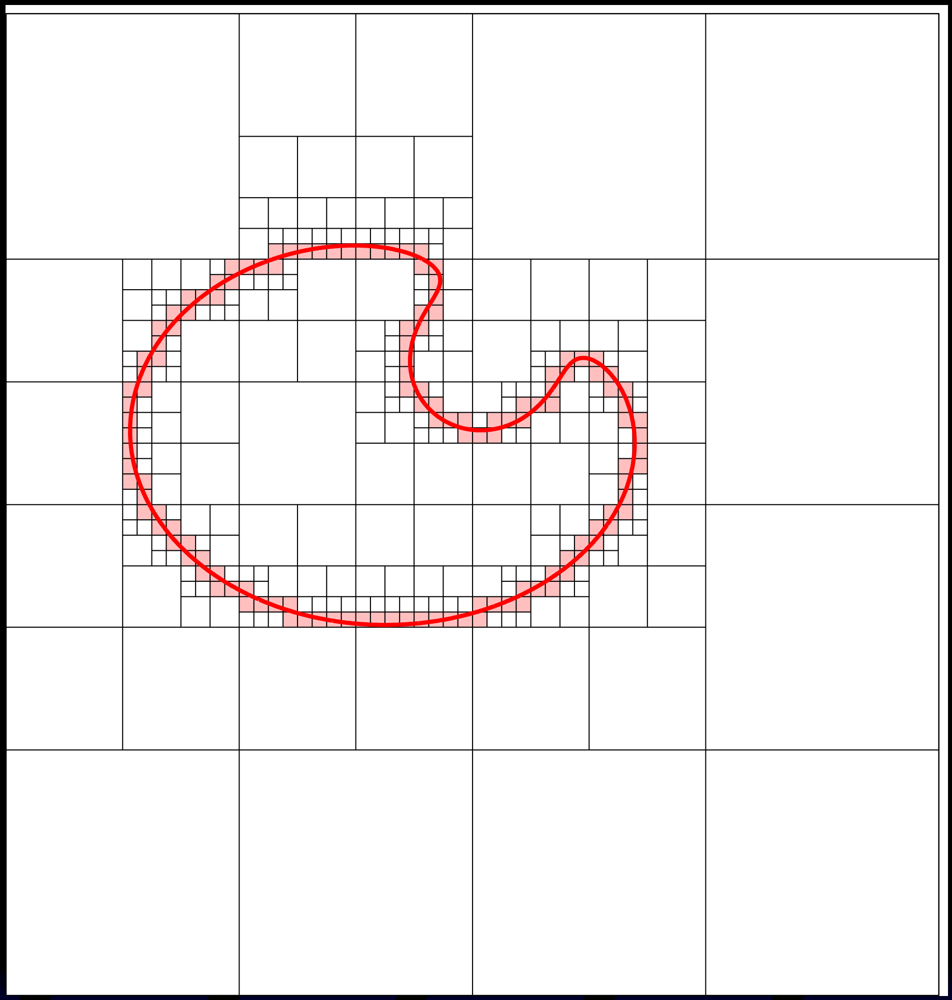
Drawbacks to wavelets in multidimensions

- Wavelets isotropic and oriented to coordinate axis
- Wavelets handle point singularities well

Drawbacks to wavelets in multidimensions

- Wavelets isotropic and oriented to coordinate axis
- Wavelets handle point singularities well
- Wavelets do not handle singularities along curves

Horizon approximation



Possible remedies

- New systems: ridgelets, wedgelets, curvelets, ...

Possible remedies

- New systems: ridgelets, wedgelets, curvelets, ...
- Candes and Donoho

Possible remedies

- New systems: ridgelets, wedgelets, curvelets, ...
- Candes and Donoho
- Frames and redundant systems

Example: Wedgelets

- To each dyadic square I with $|I| = 2^{-j}$ associate a family of wedgelets

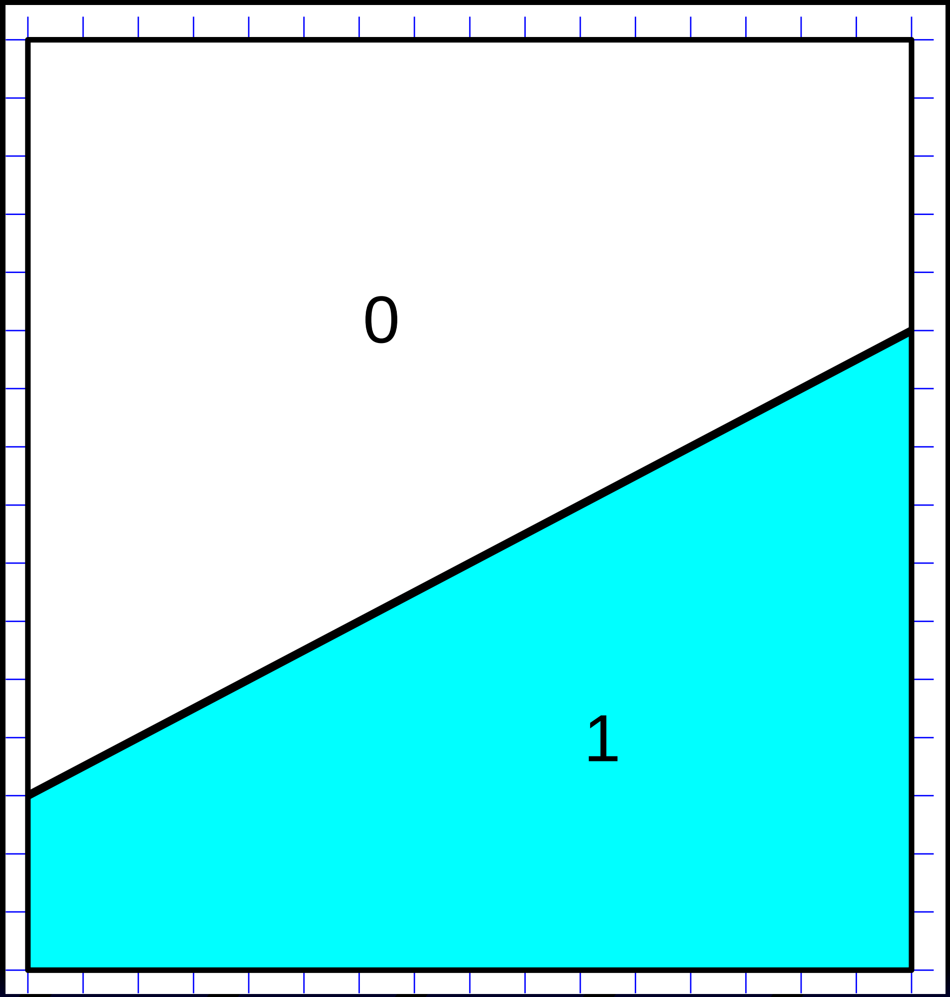
Example: Wedgelets

- To each dyadic square I with $|I| = 2^{-j}$ associate a family of wedgelets
- A wedgelet is a piecewise constant function taking the values $0, 1$

Example: Wedgelets

- To each dyadic square I with $|I| = 2^{-j}$ associate a family of wedgelets
- A wedgelet is a piecewise constant function taking the values $0, 1$
- Demarcation is given by a line connecting grid points on boundary of I with spacing 2^{-2j}

Picture of a wedgelet



Horizon functions: piecewise constants

- Any horizon function with C^2 boundary can be approximated in L_2 to error n^{-1} using n wedgelets

Horizon functions: piecewise constants

- Any horizon function with C^2 boundary can be approximated in L_2 to error n^{-1} using n wedgelets
- By comparison wavelets give error $n^{-1/2}$: $f \in \mathrm{BV} \subset \mathcal{A}^1(\mathrm{nonlinear}, \mathrm{L}_2)$. Fourier gives error $n^{-1/4}$: $f \in B^{1/2}_{\infty}(L_2) = \mathcal{A}^{1/2}(Fourier, L_2)$

General orthonormal systems

- Let $\mathcal{B} := \{b_\nu\}$ be any orthonormal basis

General orthonormal systems

- Let $\mathcal{B} := \{b_\nu\}$ be any orthonormal basis
- $f = \sum_\nu c_\nu(f) b_\nu$

General orthonormal systems

- Let $\mathcal{B} := \{b_\nu\}$ be any orthonormal basis
- $f = \sum_\nu c_\nu(f) b_\nu$
- Simple characterization of $\mathcal{A}^s(L_2)$

$$f \in \mathcal{A}^s(L_2, \mathcal{B}) \longleftrightarrow (c_\nu(f)) \in w\ell_\tau, \quad \frac{1}{\tau} = s + \frac{1}{2}$$

General orthonormal systems

- Let $\mathcal{B} := \{b_\nu\}$ be any orthonormal basis
- $f = \sum_\nu c_\nu(f) b_\nu$
- Simple characterization of $\mathcal{A}^s(L_2)$

$$f \in \mathcal{A}^s(L_2, \mathcal{B}) \longleftrightarrow (c_\nu(f)) \in w\ell_\tau, \quad \frac{1}{\tau} = s + \frac{1}{2}$$

- $w\ell_\tau$ consists of all sequences (c_ν) such that decreasing rearrangement (c_n^*) satisfies

$$c_n^* \leq C n^{-1/\tau}$$

Cannot compare systems

- $\mathcal{B}_1, \mathcal{B}_2$ two orthonormal systems

Cannot compare systems

- $\mathcal{B}_1, \mathcal{B}_2$ two orthonormal systems
- Generally speaking $\mathcal{A}^s(L_p, \mathcal{B}_1)$ and $\mathcal{A}^s(L_p, \mathcal{B}_2)$ cant be compared

Cannot compare systems

- $\mathcal{B}_1, \mathcal{B}_2$ two orthonormal systems
- Generally speaking $\mathcal{A}^s(L_p, \mathcal{B}_1)$ and $\mathcal{A}^s(L_p, \mathcal{B}_2)$ cant be compared
- Fourier basis $\mathcal{A}^1(Fourier,\ L_2)$ characterized by

$$\hat{f}^*(n) \leq Cn^{-3/2}$$

Cannot compare systems

- $\mathcal{B}_1, \mathcal{B}_2$ two orthonormal systems
- Generally speaking $\mathcal{A}^s(L_p, \mathcal{B}_1)$ and $\mathcal{A}^s(L_p, \mathcal{B}_2)$ cant be compared
- Fourier basis $\mathcal{A}^1(Fourier,\ L_2)$ characterized by

$$\hat{f}^*(n) \leq Cn^{-3/2}$$

- Wavelets characterized by same condition for wavelet coefficients.

Cannot compare systems

- $\mathcal{B}_1, \mathcal{B}_2$ two orthonormal systems
- Generally speaking $\mathcal{A}^s(L_p, \mathcal{B}_1)$ and $\mathcal{A}^s(L_p, \mathcal{B}_2)$ cant be compared
- Fourier basis $\mathcal{A}^1(Fourier,\ L_2)$ characterized by

$$\hat{f}^*(n) \leq Cn^{-3/2}$$

- Wavelets characterized by same condition for wavelet coefficients.
- Always some functions which prefer one basis over another

Dictionary approximation

- Can we put together different systems?

Dictionary approximation

- Can we put together different systems?
- A dictionary is a collection D of functions with norm 1.

Dictionary approximation

- Can we put together different systems?
- A dictionary is a collection D of functions with norm 1.
- Example: D is a union of several orthonormal bases

Dictionary approximation

- Can we put together different systems?
- A dictionary is a collection D of functions with norm 1.
- Example: D is a union of several orthonormal bases
- $\sigma_n(f, D)_p$ is error in n-term approximation using elements of dictionary

Greedy Algorithms

- How can we generate good approximations from a dictionary?

Greedy Algorithms

- How can we generate good approximations from a dictionary?
- Greedy Algorithm : Given $f \in L_2$:
  1. Choose $g_1 \in D$ such that $|\langle f, g_1 \rangle|$ maximum over $g \in D$
  2. $A_1 := \langle f, g_1 \rangle g_1$, $f_1 = f - A_1$ residual
  3. Repeat with f replaced by f_1.....

Greedy Algorithms

- How can we generate good approximations from a dictionary?
- Greedy Algorithm : Given $f \in L_2$:
  1. Choose $g_1 \in D$ such that $|\langle f, g_1 \rangle|$ maximum over $g \in D$
  2. $A_1 := \langle f, g_1 \rangle g_1$, $f_1 = f - A_1$ residual
  3. Repeat with f replaced by f_1.....
- Greedy Algorithm converges (L. Jones)

Convergence Rates for Greedy Algorithms

- $\mathcal{F}_\tau := \{ f = \sum_{g\in\mathcal{D}} c_g g : \sum_{g\in\mathcal{D}} |c_g|^\tau < \infty \}$

Convergence Rates for Greedy Algorithms

- $\mathcal{F}_\tau := \{\ f = \sum_{g\in\mathcal{D}} c_g g:\ \sum_{g\in\mathcal{D}} |c_g|^\tau < \infty\}$
- (DeVore-Temlyakov) $f \in \mathcal{F}_1$ then $\|f_n\| \le C n^{-1/6}$

Convergence Rates for Greedy Algorithms

- $\mathcal{F}_\tau := \{ f = \sum_{g \in \mathcal{D}} c_g g : \sum_{g \in \mathcal{D}} |c_g|^\tau < \infty \}$
- (DeVore-Temlyakov) $f \in \mathcal{F}_1$ then $\|f_n\| \leq Cn^{-1/6}$
- Modifications of Greedy Algorithm have better convergence

Convergence Rates for Greedy Algorithms

- $\mathcal{F}_\tau := \{\, f = \sum_{g\in\mathcal{D}} c_g g : \ \sum_{g\in\mathcal{D}} |c_g|^\tau < \infty\}$
- (DeVore-Temlyakov) $f \in \mathcal{F}_1$ then $\|f_n\| \leq C n^{-1/6}$
- Modifications of Greedy Algorithm have better convergence
- Example $f \in \mathcal{F}_1$ then decay is $O(n^{-1/2})$ for modified algorithm (Maurey +)

Convergence Rates for Greedy Algorithms

- $\mathcal{F}_\tau := \{ f = \sum_{g\in\mathcal{D}} c_g g : \sum_{g\in\mathcal{D}} |c_g|^\tau < \infty\}$
- (DeVore-Temlyakov) $f \in \mathcal{F}_1$ then $\|f_n\| \leq Cn^{-1/6}$
- Modifications of Greedy Algorithm have better convergence
- Example $f \in \mathcal{F}_1$ then decay is $O(n^{-1/2})$ for modified algorithm (Maurey +)
- Greedy Algorithms expensive to implement

Example: Spline-Fourier

- Given n, σ_n piecewise Fourier

Example: Spline-Fourier

- Given n, σ_n piecewise Fourier
- Partition $[0, 1]$ into m intervals I_k, $k = 1, \ldots, m$

Example: Spline-Fourier

- Given n, σ_n piecewise Fourier
- Partition $[0, 1]$ into m intervals I_k, $k = 1, \ldots, m$
- On each interval use n_k complex exponentials e^{ijx}

Example: Spline-Fourier

- Given n, σ_n piecewise Fourier
- Partition $[0,1]$ into m intervals I_k, $k = 1, \ldots, m$
- On each interval use n_k complex exponentials e^{ijx}
- Total investment $n = \sum_{k=1}^{m} n_k$

Example: Spline-Fourier

- Given n, σ_n piecewise Fourier
- Partition $[0, 1]$ into m intervals I_k, $k = 1, \ldots, m$
- On each interval use n_k complex exponentials e^{ijx}
- Total investment $n = \sum_{k=1}^{m} n_k$
- $\sigma_n(f)_p$ error in best approximation by elements from Σ_n

Example: Spline-Fourier

- Given n, σ_n piecewise Fourier
- Partition $[0, 1]$ into m intervals I_k, $k = 1, \ldots, m$
- On each interval use n_k complex exponentials e^{ijx}
- Total investment $n = \sum_{k=1}^{m} n_k$
- $\sigma_n(f)_p$ error in best approximation by elements from Σ_n
- Characterize approximation classes for this type of approximation?

Another Idea for merging systems

- Baraniuk-Romberg-Wakin idea for merging wedgelets and wavelets

Another Idea for merging systems

- Baraniuk-Romberg-Wakin idea for merging wedgelets and wavelets
- Ornate wavelet tree with label on how wavelet coefficients to be computed

Another Idea for merging systems

- Baraniuk-Romberg-Wakin idea for merging wedgelets and wavelets
- Ornate wavelet tree with label on how wavelet coefficients to be computed
- interior nodes compute wavelet coefficients in standard way

BRW Continued

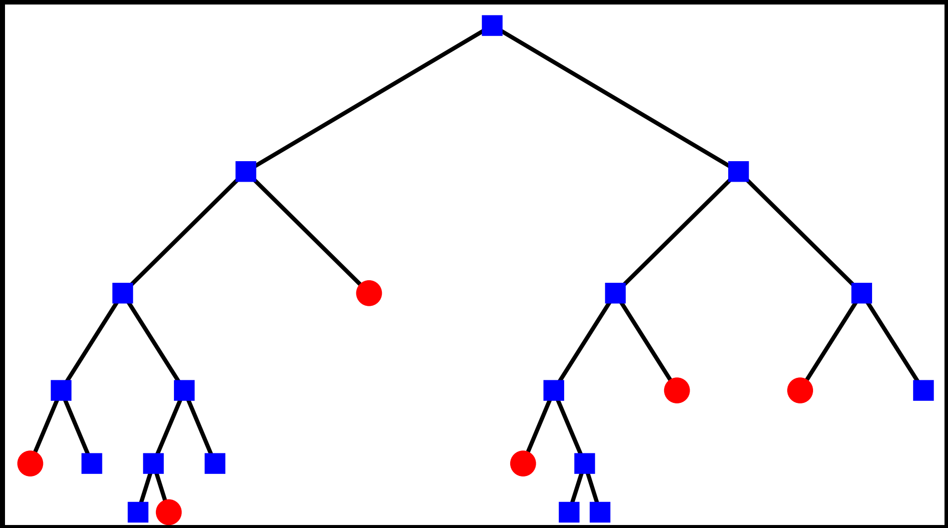
- If leave (final node) of tree is ornated with a wavelet then wavelet coefficients for all nodes below leave are given value zero

BRW Continued

- If leave (final node) of tree is ornated with a wavelet then wavelet coefficients for all nodes below leave are given value zero
- If leave is ornated with wedgelet, all coefficients below this node are computed as wavelet coefficients of that wedgelet (wedgeprint)

Wedgelet-wavelet tree

red = Wedgelets   blue = Wavelets



Encoding Tree

- Given a terminal node I we have a fixed number of options to encode the tree T_I with root I

Encoding Tree

- Given a terminal node I we have a fixed number of options to encode the tree T_I with root I
- To each terminal node associate error $e(n, I)$ for best encoding of the tree T_I using n bits

Encoding Tree

- Given a terminal node I we have a fixed number of options to encode the tree T_I with root I
- To each terminal node associate error $e(n, I)$ for best encoding of the tree T_I using n bits
- Similarly have an error $e(n, I)$ for enconding the wavelet coefficient for interior nodes using n nodes

Encoding Tree: continued

- Given bit budget N there is a best tree and bit allocation n_I

Encoding Tree: continued

- Given bit budget N there is a best tree and bit allocation n_I
- Can we dynamically find best tree and best bit allocation