Wavelets and Wavelet Regression

Jeffrey S. Morris

Department of Biostatistics

The University of Texas

MD Anderson Cancer Center

Rice University

2/25/2008

Nonparametric Regression

- Given a response vector $y=\{y_1, \ldots, y_N\}'$ with predictors $t=\{t_1, \ldots, t_N\}'$, a ***nonparametric regression model*** for y on t is given by

$$y_i = f(t_i) + e_i \qquad e_i \sim N(0, \sigma_e^2)$$

where the form of f is left unspecified.

- **Goal**: estimate f
- **Various approaches:**
  - ***Kernel*** methods
  - ***Spline***-based methods
  - ***Wavelet***-based methods

Nonparametric Regression: Kernel Estimators

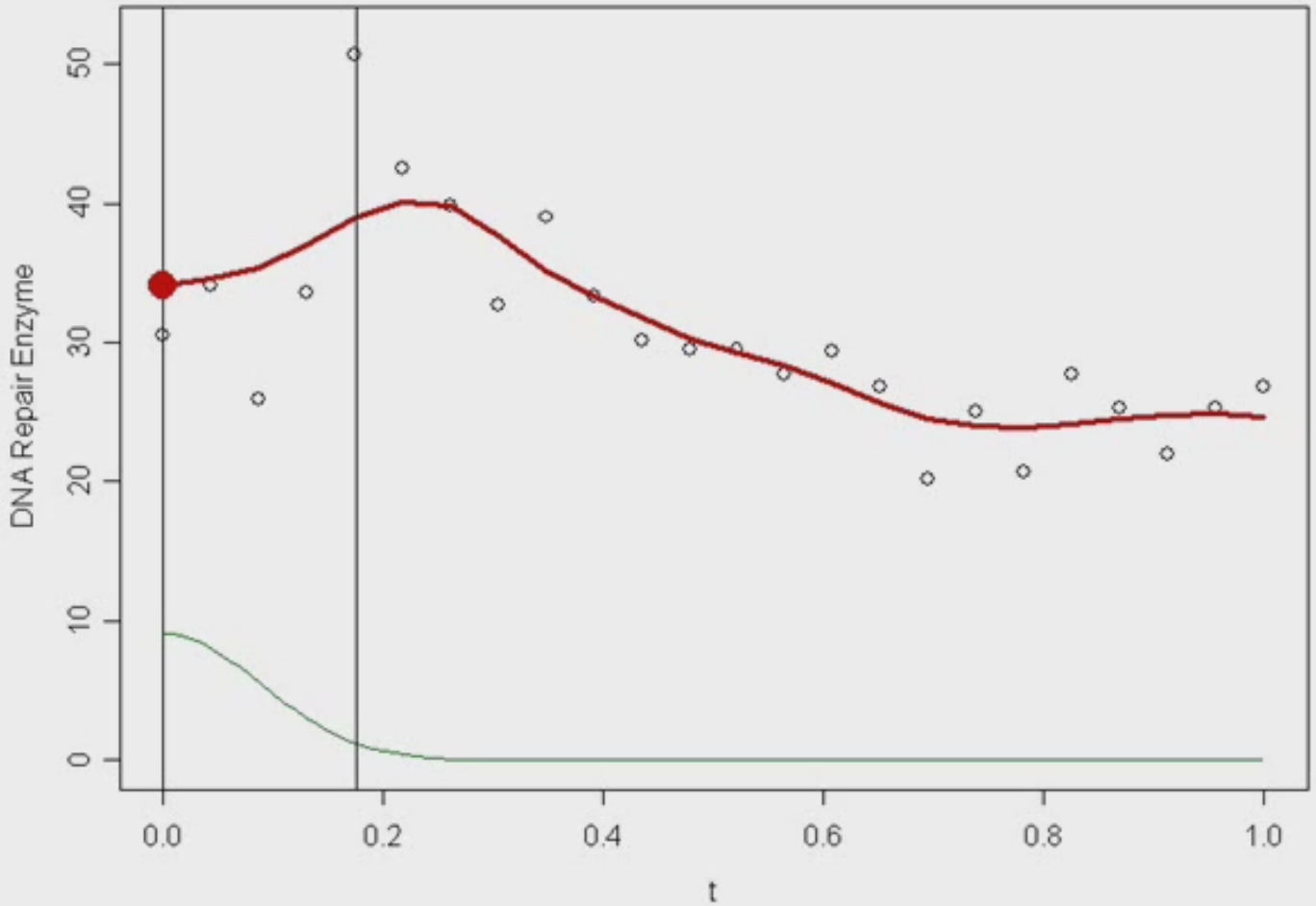
- **The idea behind kernel estimation is that simple parametric estimators (mean, linear, quadratic) are applied *locally*, with data points weighted according to a *kernel function* (bounded or decays to zero)**

Nadaraya/Watson *local mean* estimator:

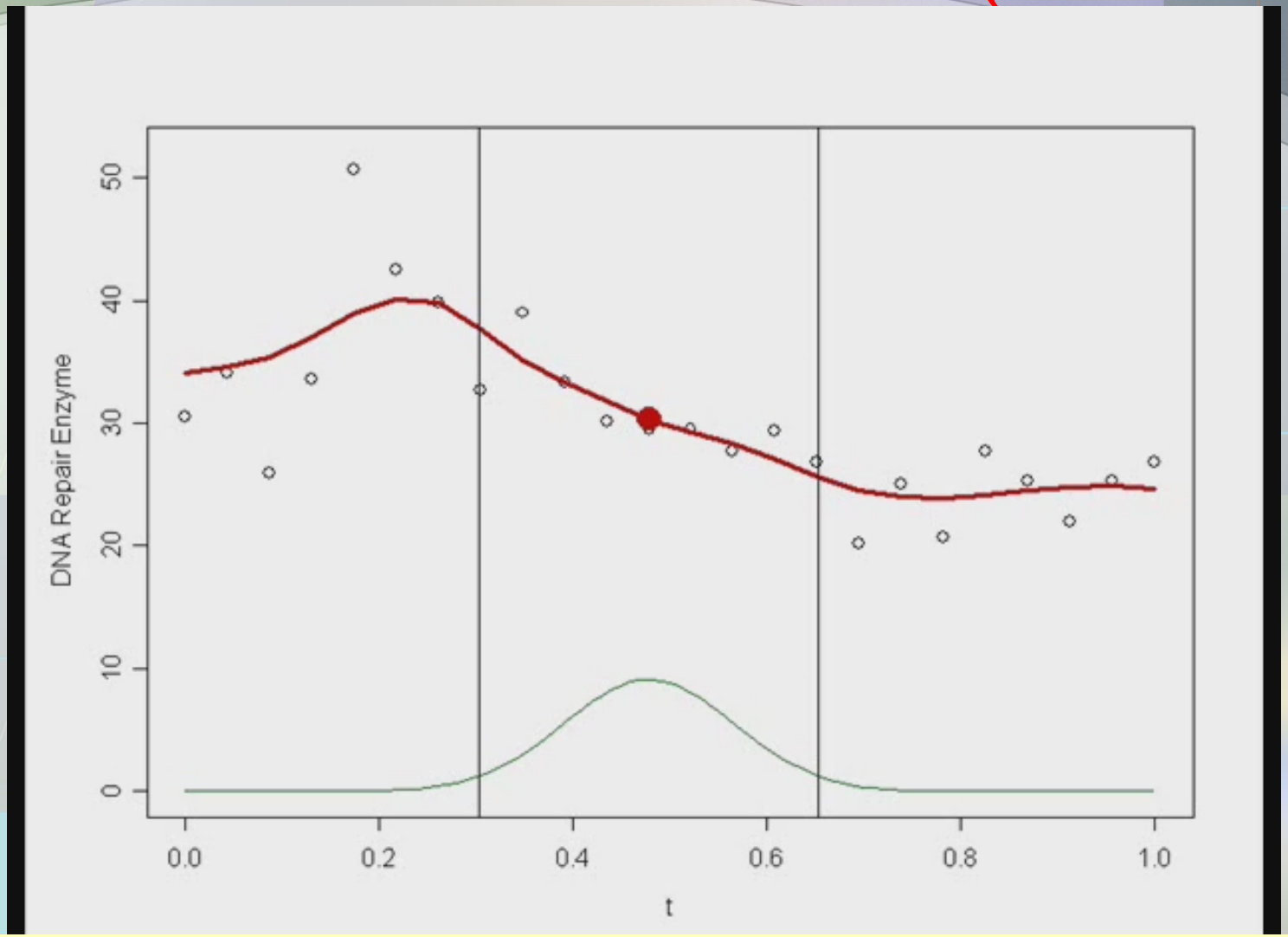
$$\hat{f}(t) = \frac{\sum_{i=1}^{N} w(t_i - t; h) y_i}{\sum_{i=1}^{N} w(t_i - t; h)}$$

- ***w* is a kernel function (e.g. Gaussian pdf)**
- ***h* is a *bandwidth* mitigating the trade-off between bias and variance (e.g. Gaussian standard dev.)**

Nonparametric Regression: N-W Local Mean Smoother



Nonparametric Regression:
N-W Local Mean Smoother (Bandwidth)



- **Can estimate *h* from data by cross validation (or GCV)**

Nonparametric Regression: Kernel Estimators

- **Can improve performance by replacing local mean with local regression fits (line/parabola)**
- **Fits line or parabola locally, with weights determined by kernel function**

***Local Linear* Smoother:**

$$\hat{f}(t) = \text{least squares solution to}$$

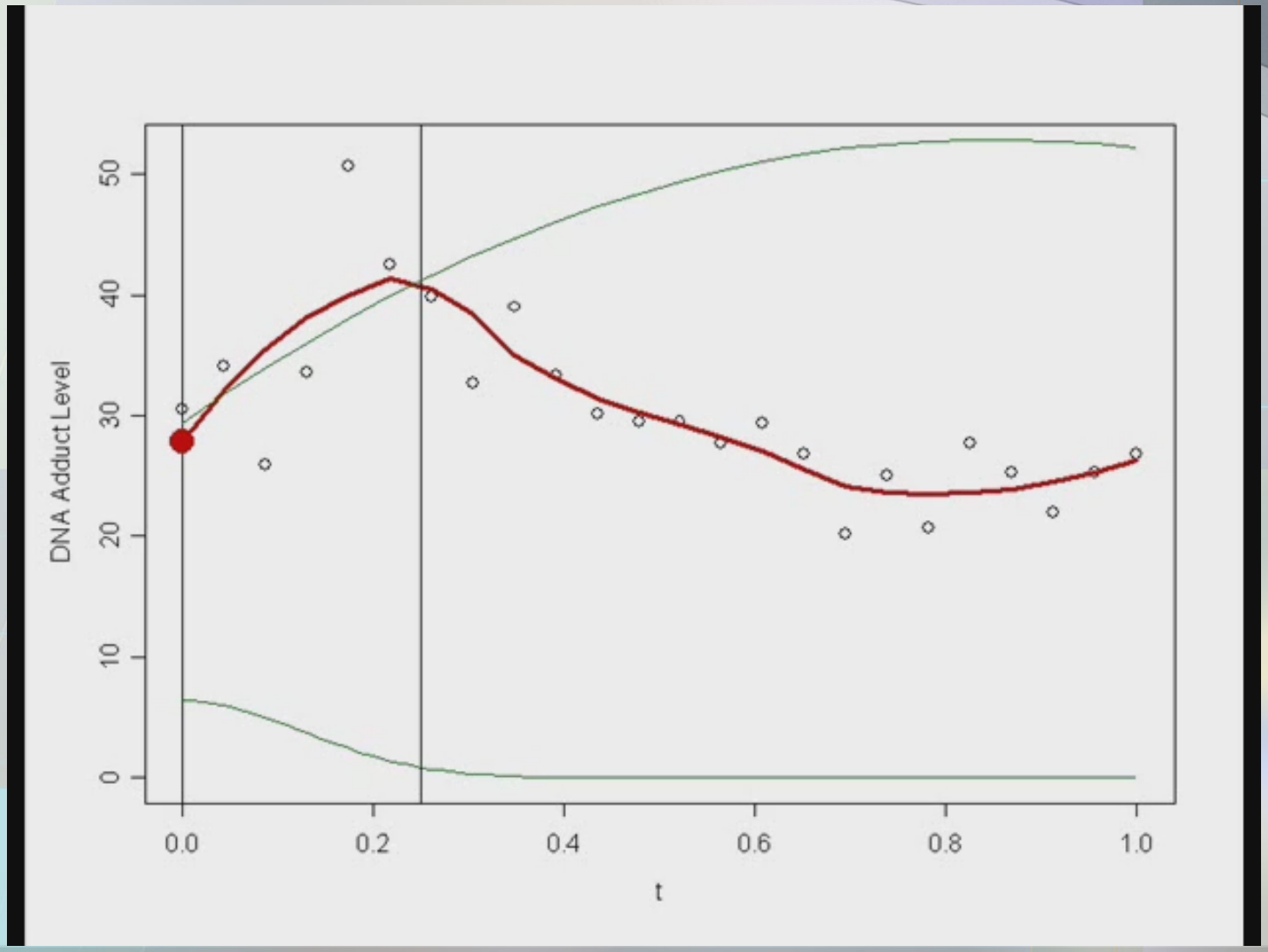
$$\min_{b_0, b_1} \sum_{i=1}^{N} [y_i - \{b_0 + b_1(t_i - t)\}]^2 w(t_i - t; h)$$

***Local Quadratic* Smoother:**

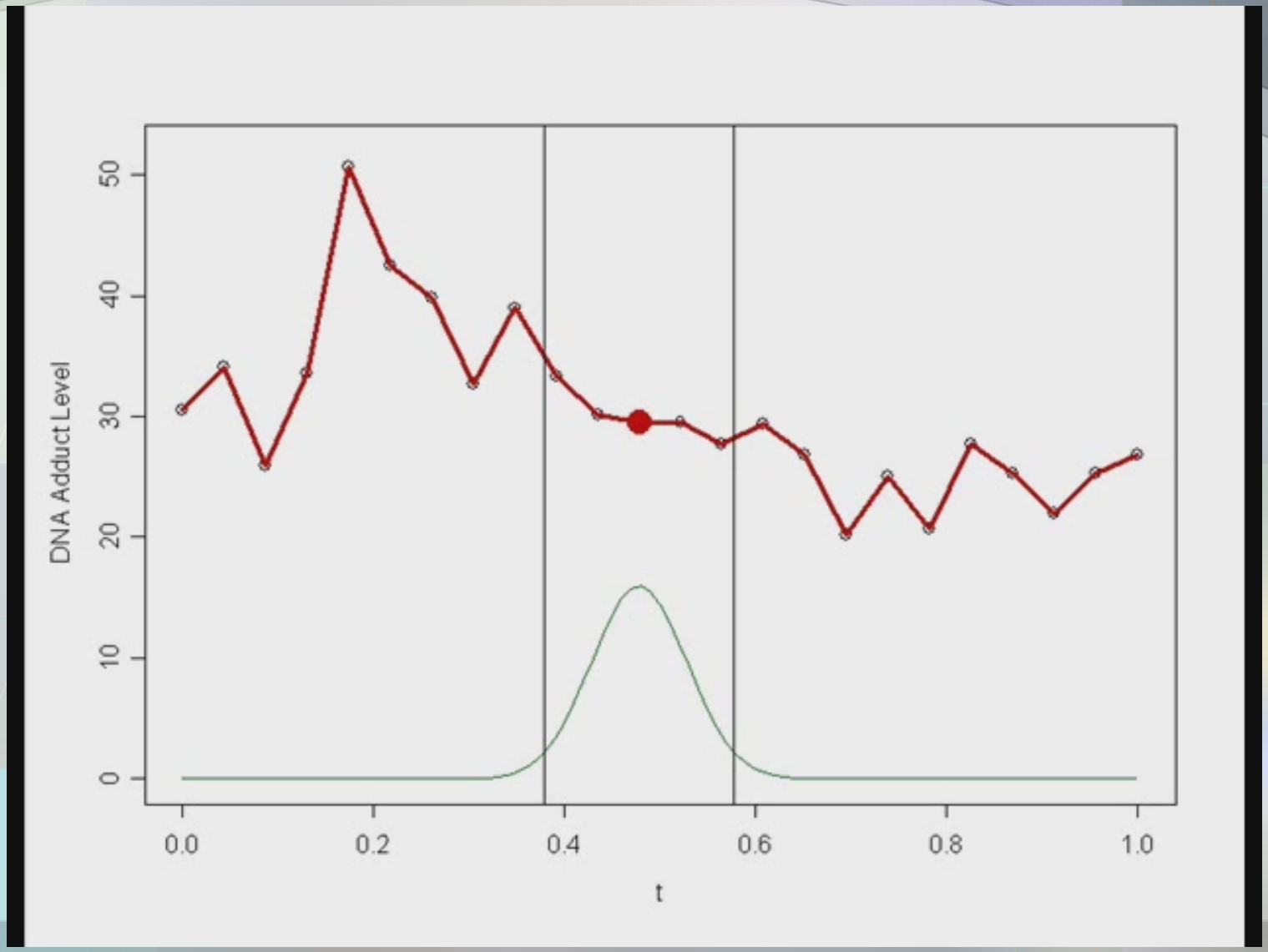
$$\hat{f}(t) = \text{least squares solution to}$$

$$\min_{b_0, b_1, b_2} \sum_{i=1}^{N} [y_i - \{b_0 + b_1(t_i - t) + b_2(t_i - t)^2\}]^2 w(t_i - t; h)$$

Nonparametric Regression: Local Quadratic Smoother



Nonparametric Regression:
Local Quadratic Smoother (Bandwidth)



Nonparametric Regression: Smoothing Splines

- **Another way to view the nonparametric regression problem is as *penalized regression*; minimize *L*:**

$$L = N^{-1}\sum_{i=1}^{N}\{y_i - f(t_i)\}^2 + \alpha\int f''(t)^2\,dt$$

- **It can be shown that this function is minimized by a *cubic smoothing spline* with knots at values t_i**
  - **–Piecewise cubic polynomial between knots**
  - **–Two continuous derivatives**
  - **–Third derivative: step function with change point at knots**
- **Smoothing parameter: α (can be estimated from data)**

Nonparametric Regression: Regression Splines

- **An easier-to-fit alternative to smoothing splines are *regression splines*, which involve polynomial regressions in intervals between chosen set of *p* knots $\kappa=\{\kappa_1, \kappa_2, \ldots, \kappa_p\}$:**

$$\text{minimize} \sum_{i=1}^{N} (y_i - Xb)^2$$

Cubic Regression : Spline

$$X = \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 & (t_1-\kappa_1)_+^3 & \cdots & (t_1-\kappa_p)_+^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_N & t_N^2 & t_N^3 & (t_N-\kappa_1)_+^3 & \cdots & (t_N-\kappa_p)_+^3 \end{bmatrix}$$

- **where $(t-\kappa)^3_+=(t-\kappa)^3$ if $(t-\kappa)>0$, 0 otherwise**
- **Knots frequently chosen to be at quantiles of *t***
  - **Can be shown that 7-8 knots sufficient for most smooth functions**

Nonparametric Regression: Penalized Regression Splines

- **Problem:** wiggly fits because no penalty (overfitting)
- **Solution:** Penalize Σb_j^2 for spline terms

  P-Splines (Eilers and Marx, Ruppert Wand and Carroll)

$$\text{minimize} \sum_{i=1}^{N} (y_i - Xb)^2 + \lambda^2 b' D b$$

- Where D is a diagonal matrix with 1's corresponding to the "spline" terms, and 0's to the "polynomial"
- **Smoothing parameter:** λ
- Solution is **ridge regression estimator**:

$$\hat{y} = X(X'X + \lambda^2 D)^{-1} X' y$$

Nonparametric Regression: Penalized Regression Splines

- **Can be represented by linear mixed model!!**

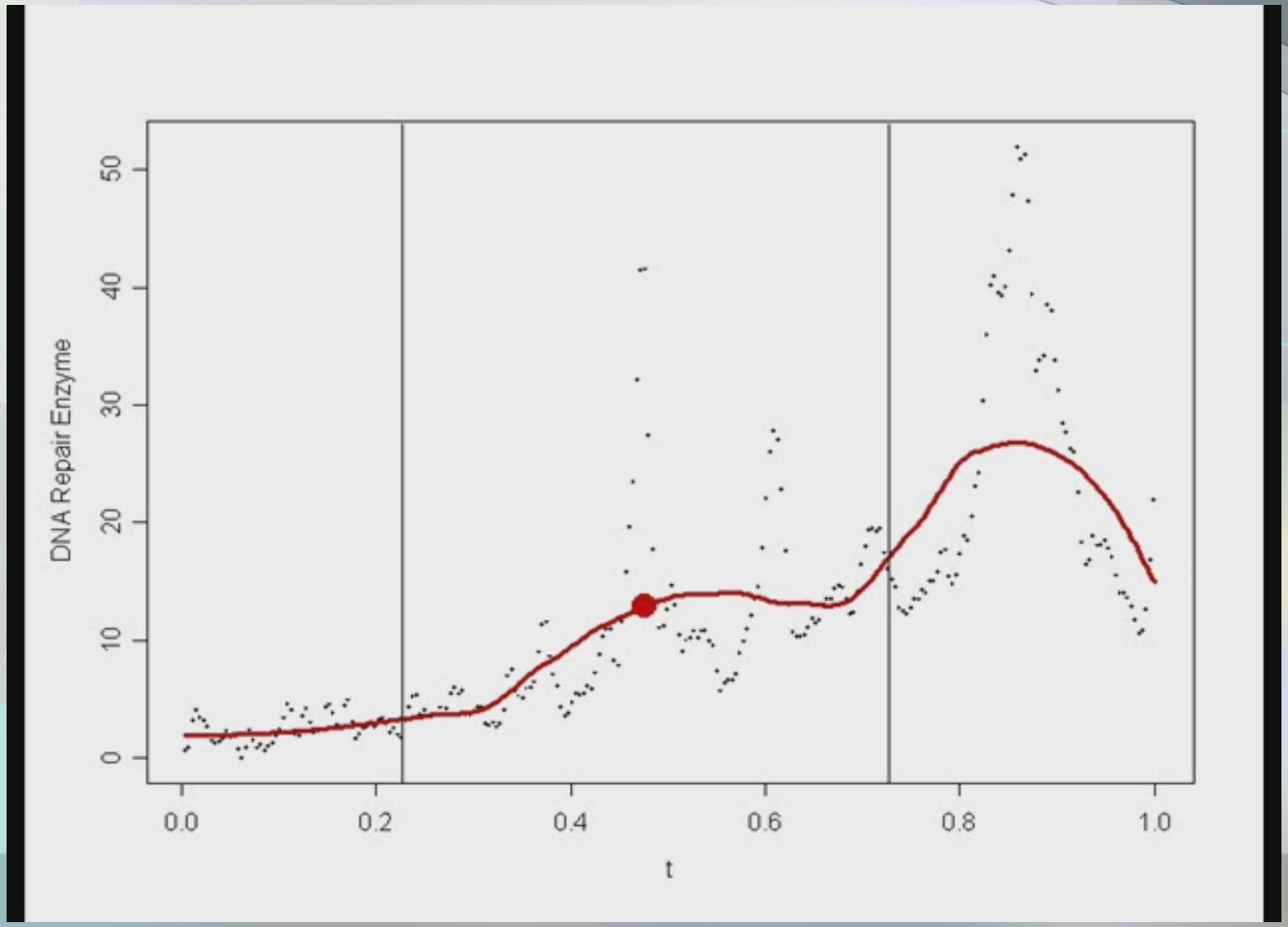
$$Y = Xb + Zu + e \qquad u \sim N(0, \lambda^2) \qquad e \sim N(0, \sigma_e^2)$$

$$X = \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & t_N & t_N^2 & t_N^3 \end{bmatrix} \qquad Z = \begin{bmatrix} (t_1 - \kappa_1)_+^3 & \cdots & (t_1 - \kappa_p)_+^3 \\ \vdots & \ddots & \vdots \\ (t_N - \kappa_1)_+^3 & \cdots & (t_N - \kappa_p)_+^3 \end{bmatrix}$$

- **Spline coefficient estimators are BLUPs from LMM!**
  - ***Linearly shrunken*** **towards zero according to relative sizes of λ^2 and σ_e^2 ; equivalent to mean zero Bayesian prior on u**
- **Smoothing parameter is variance component!**
  - **Estimated from data using REML**
- **Can also be fit as Bayesian hierarchical model**

Nonparametric Regression:

- **What if we have a spatially heterogeneous function?**



Nonparametric Regression

Spatially Adaptive Function Estimation

- **What if we have *spatially heterogeneous* data?**
  - **Different degrees of smoothness at different parts of curve**
- **We need *spatially adaptive* smoothers (regularization rather than smoothness)**
- **Kernels can be spatially adaptive**
  - **Spatially varying bandwidth**
- **Splines can be spatially adaptive**
  - **Vary knot locations (free knot splines)**
  - **Spatially varying smoothing parameter**
- **Wavelet Regression: has been shown to adjust optimally to varying smoothness (Fan, et al. 1993)**

Wavelets: Fourier Analysis

- **Any function *f* in $L^2[-\pi,\pi]$ can be represented by a *Fourier Series***

$$y(t) = \frac{a_0}{\sqrt{2\pi}} + \sum_{j=1}^{\infty}\left\{a_j \frac{\cos(jt)}{\sqrt{\pi}} + b_j \frac{\sin(jt)}{\sqrt{\pi}}\right\}$$

$$a_0 = \left\langle y, \frac{1}{\sqrt{2\pi}}\right\rangle = \int_{-\pi}^{\pi} y(t)\frac{1}{\sqrt{2\pi}}dt$$

$$a_j = \left\langle y, \frac{\cos(jt)}{\sqrt{\pi}}\right\rangle = \int_{-\pi}^{\pi} y(t)\frac{\cos(jt)}{\sqrt{\pi}}dt$$

$$b_j = \left\langle y, \frac{\sin(jt)}{\sqrt{\pi}}\right\rangle = \int_{-\pi}^{\pi} y(t)\frac{\sin(jt)}{\sqrt{\pi}}dt$$

- **Coefficients (a_j,b_j) describe behavior of the function at frequency j**
- $\{1/\sqrt{2\pi},\ 1/\sqrt{\pi}\cos(jt),\ 1/\sqrt{\pi}\sin(jt)\ j=1, \ldots, \infty\}$ **form a *complete orthonormal basis* for $L^2[-\pi,\pi]$**
  - $\langle f_j, f_k \rangle = I(j=k)$
- **This set of functions spans $L^2[-\pi,\pi]$**

Wavelets: Fourier Analysis

- **Fourier Series also have a complex representation:**

$$y(t) = \sum_{j=-\infty}^{\infty} d_j \psi_j(t) \text{ for } \psi_j(t) = \frac{e^{ijt}}{\sqrt{2\pi}} = \frac{\cos(jt)}{\sqrt{2\pi}} + i\frac{\sin(jt)}{\sqrt{2\pi}} \qquad d_j = \langle y, \overline{\psi_j} \rangle = \int_{-\pi}^{\pi} y(t) \frac{e^{-ijt}}{\sqrt{2\pi}} dt$$

- **$\{\psi_j, j=-\infty, \ldots, \infty\}$ form a complete orthonormal basis for $L^2[-\pi,\pi]$**
- **Discrete Fourier Analysis: For signal defined on an equally spaced grid of size T, a fast algorithm {FFT, $O(T\log T)$} is available for computing Fourier coefficients from the observed signal. (Transform data to frequency domain: $T \rightarrow T$)**
- **Fourier analysis useful for signal processing, but has important limitation: all location information lost in frequency domain**
  - **Useful for processing *stationary* signals, but not so much for *nonstationary***
  - **What if we had a set of basis functions that decomposed information in both the frequency and time domain? We do! *Wavelets*!!**

Wavelets

$$y(t) = \sum_{j,k \in \Im} d_{jk} \psi_{jk}(t)$$

$$\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k)$$

$$d_{jk} = \langle y, \psi_{jk} \rangle = \int_{\Re} y(t) \psi_{jk}(t) dt$$

- **Coefficient d_{jk} summarizes behavior of the function at scale (frequency) indexed by j, and location indexed by k.**
- **$\{\psi_{jk}, j,k=-\infty, \ldots, \infty\}$ form a complete orthonormal basis for $L^2(\Re)$**
- **The *mother wavelet basis function*, ψ, has several properties:**
  - $\int \psi(t)dt=0$, $\int \psi^2(t)dt=1$
  - **It has a corresponding *father wavelet* φ such that $\langle \varphi, \psi \rangle = \int \varphi(t)\psi(t)dt=0$ (also called a *scaling function*), $\int \varphi(t)dt=1$, $\int \varphi^2(t)dt=1$**
  - **It has compact or vanishing support (unlike Fourier bases)**
  - **It generates a *multiresolution analysis***

Wavelets: Multiresolution Analysis

- Let V_j, $j= \ldots, -2, -1, 0, 1, 2, \ldots$ be a sequence of function subspaces in $L^2(\Re)$. The collection of spaces $\{V_j, j \in \Im\}$ is called a ***multi-resolution analysis*** w/ scaling function φ if these conditions hold:
  - ***Orthonormal Basis***: The function φ belongs to V_0 and the set $\{\varphi(t-k), k \in \Im\}$ is an orthonormal basis for V_0.
  - ***Nested:*** $V_j \subset V_{j+1}$
  - ***Density:*** $\cup V_j = L^2(\Re)$
  - ***Separation:*** $\cap V_j = \{0\}$
  - ***Scaling:*** the function $f(t)$ belongs to V_j iff the function $f(2^j t)$ belongs to V_0

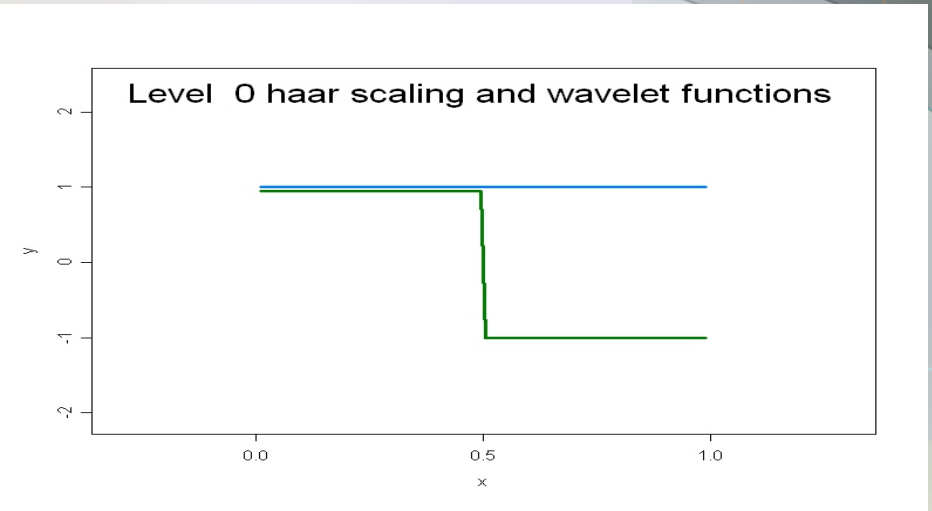
    Thus, $\{2^{j/2}\varphi(2^j t-k), k \in \Im\}$ is an orthonormal basis for V_j
- Note the following properties also hold:
  - W_j=space spanned by $\{2^{j/2}\psi(2^j t-k), k \in \Im\}$
  - $V_j \oplus W_j = V_{j+1}$, so one can think of W as "residual space"

Wavelets: Haar Wavelet Example

- **The simplest and oldest wavelet is the *Haar* wavelet**

$$\varphi(t) = I_{[0,1)}(t)$$

$$\psi(t) = \begin{cases} 1 \text{ for } t \in [0,0.5) \\ -1 \text{ for } t \in [0.5,1) \\ 0 \text{ otherwise} \end{cases}$$

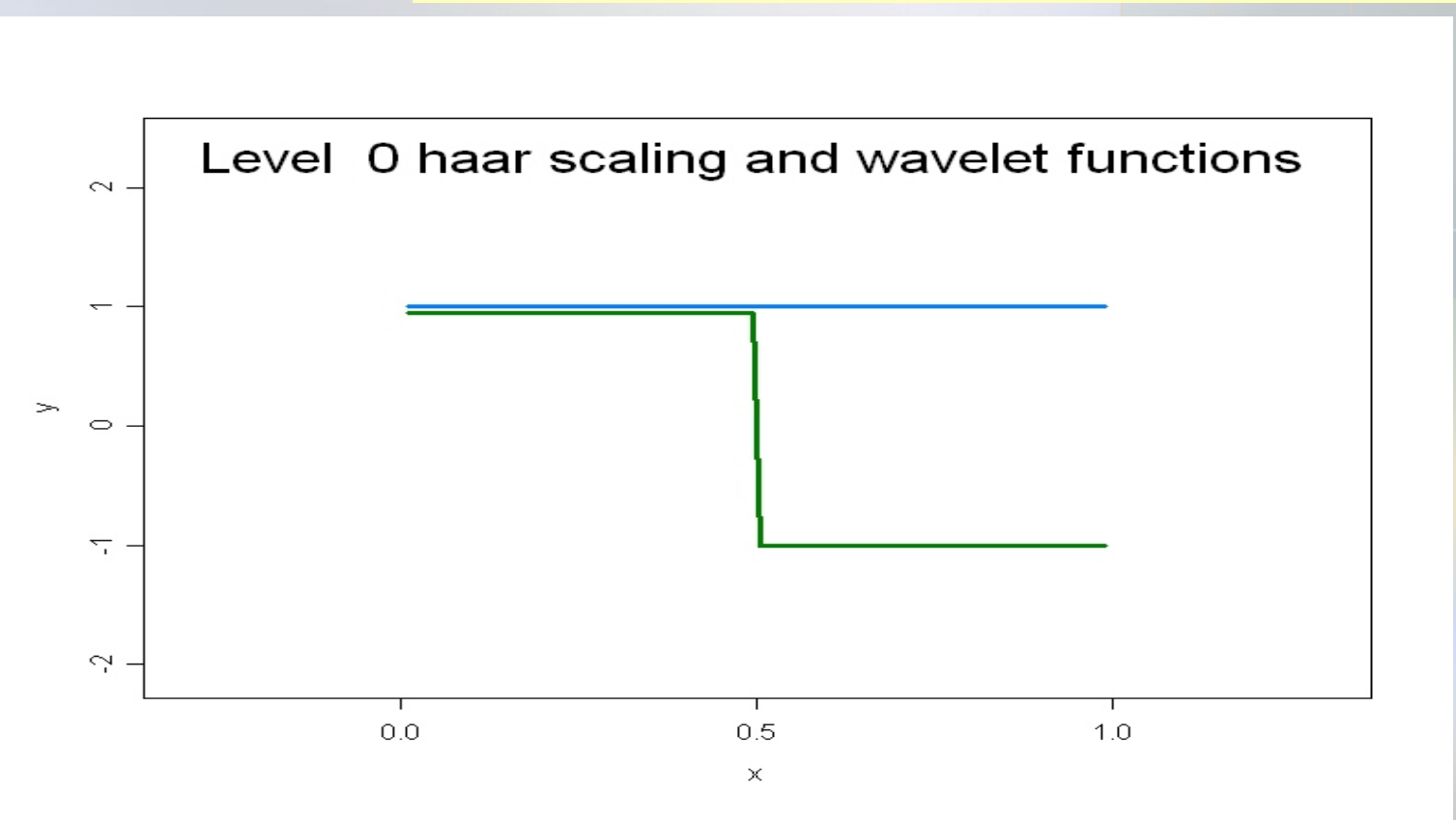


- $\int \varphi(t)\,\psi(t)dt=0$ $\int \varphi(t)dt=1$ $\int \varphi^2(t)dt=1$ $\int \psi(t)dt=0$ $\int \psi^2(t)dt=1$
- V_0=space spanned by $\{\varphi(t-k), k \in \mathfrak{I}\}$ (piecewise const, jumps *1*)
- W_0=space spanned by $\{\psi(t-k), k \in \mathfrak{I}\}$,
- V_1=space from $\{2^{1/2}\varphi(2t-k), k \in \mathfrak{I}\}$ (piecewise const, jumps ½)
- $V_0 \subset V_1$, $V_1 = V_0 \oplus W_0$, $\bigcup V_j = L^2(\Re)$

Wavelets: Haar Wavelet Example

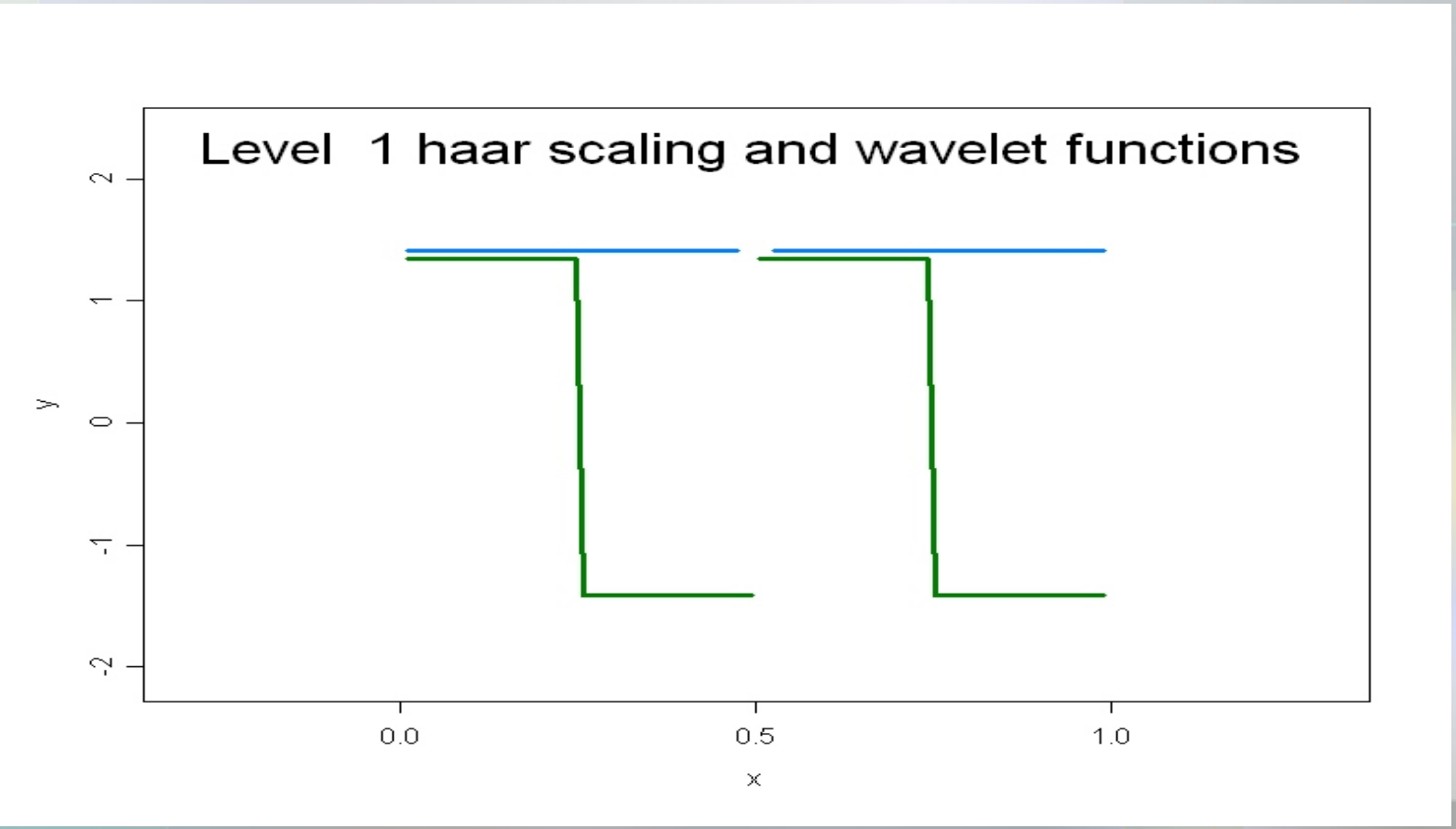
- **Consider** $L^2\{[0,1)\}$

$$\varphi_{00}(t) = I_{[0,1)}(t) \qquad \psi_{00}(t) = \begin{cases} 1 \text{ for } t \in [0,0.5) \\ -1 \text{ for } t \in [0.5,1) \\ 0 \text{ otherwise} \end{cases}$$



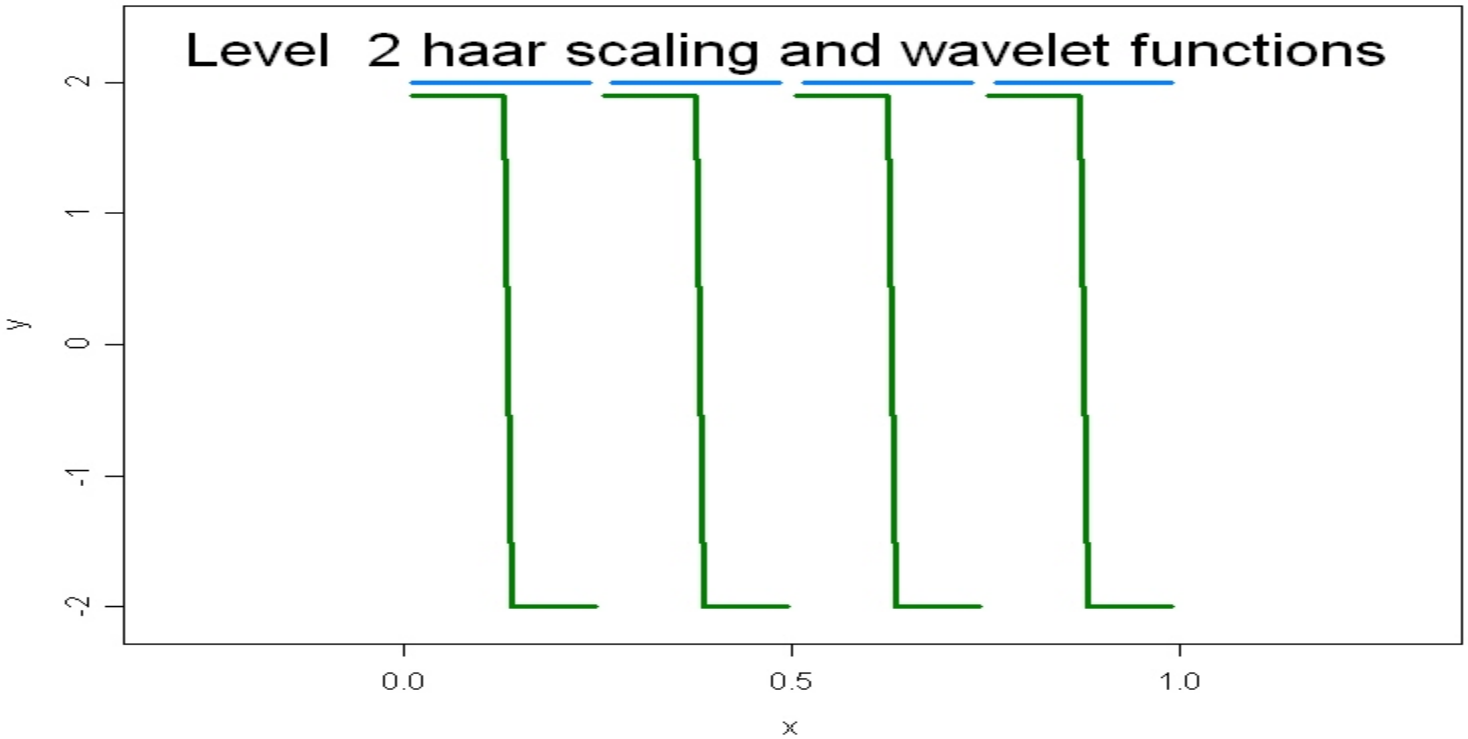
Wavelets: Haar Wavelet Example

$$\varphi_{11}(t) = 2^{1/2} I_{[0.5,1)}(t) \qquad \psi_{11}(t) = \begin{cases} 2^{1/2} \text{ for } t \in [0.5, 0.75) \\ -2^{1/2} \text{ for } t \in [0.75, 1.0) \\ 0 \text{ otherwise} \end{cases}$$



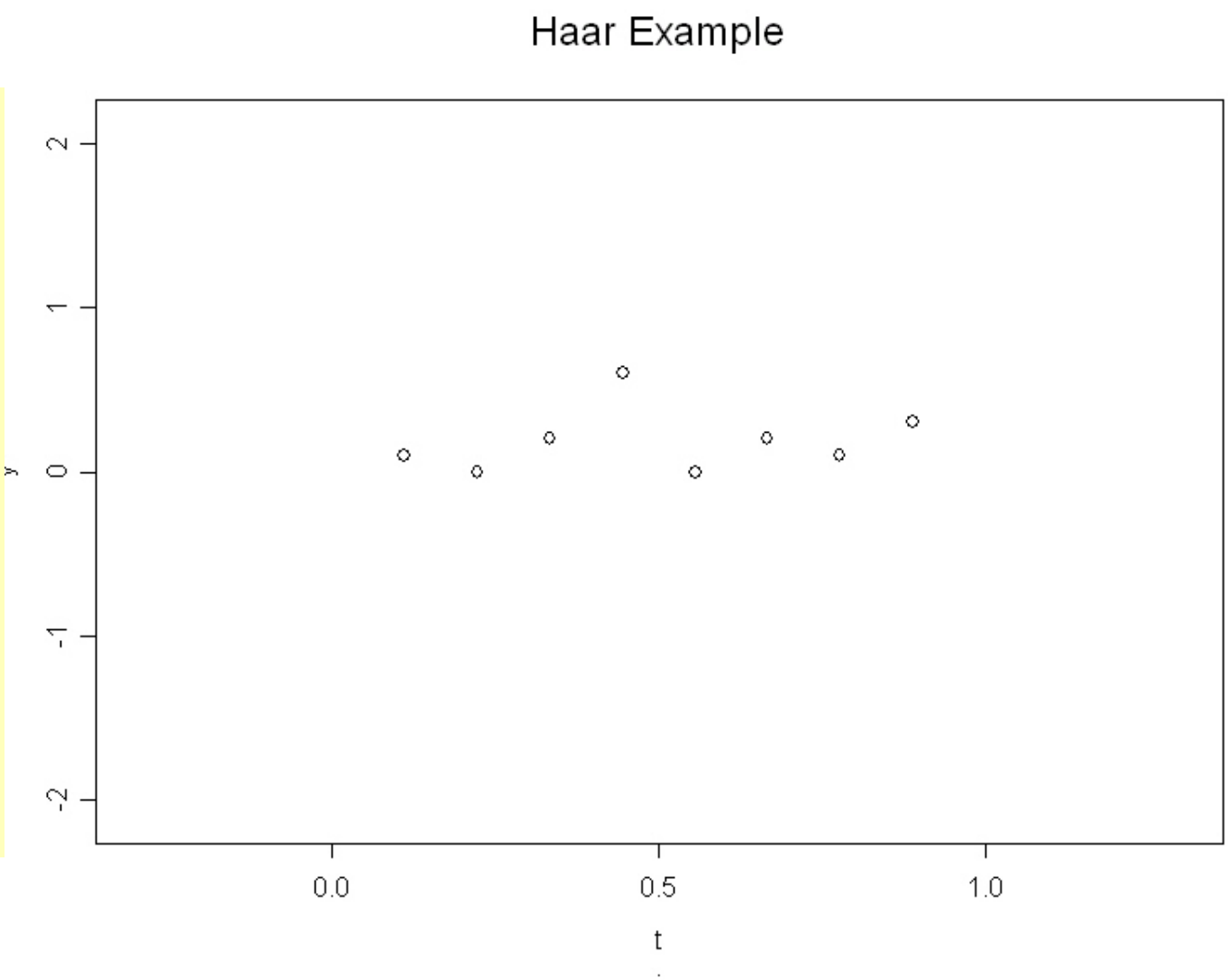
Wavelets: Haar Wavelet Example

$$\varphi_{23}(t) = I_{[0.75,1.00)}(t) \qquad \psi_{23}(t) = \begin{cases} 2 \text{ for } t \in [0.75, 0.875) \\ -2 \text{ for } t \in [0.875, 1) \\ 0 \text{ otherwise} \end{cases}$$



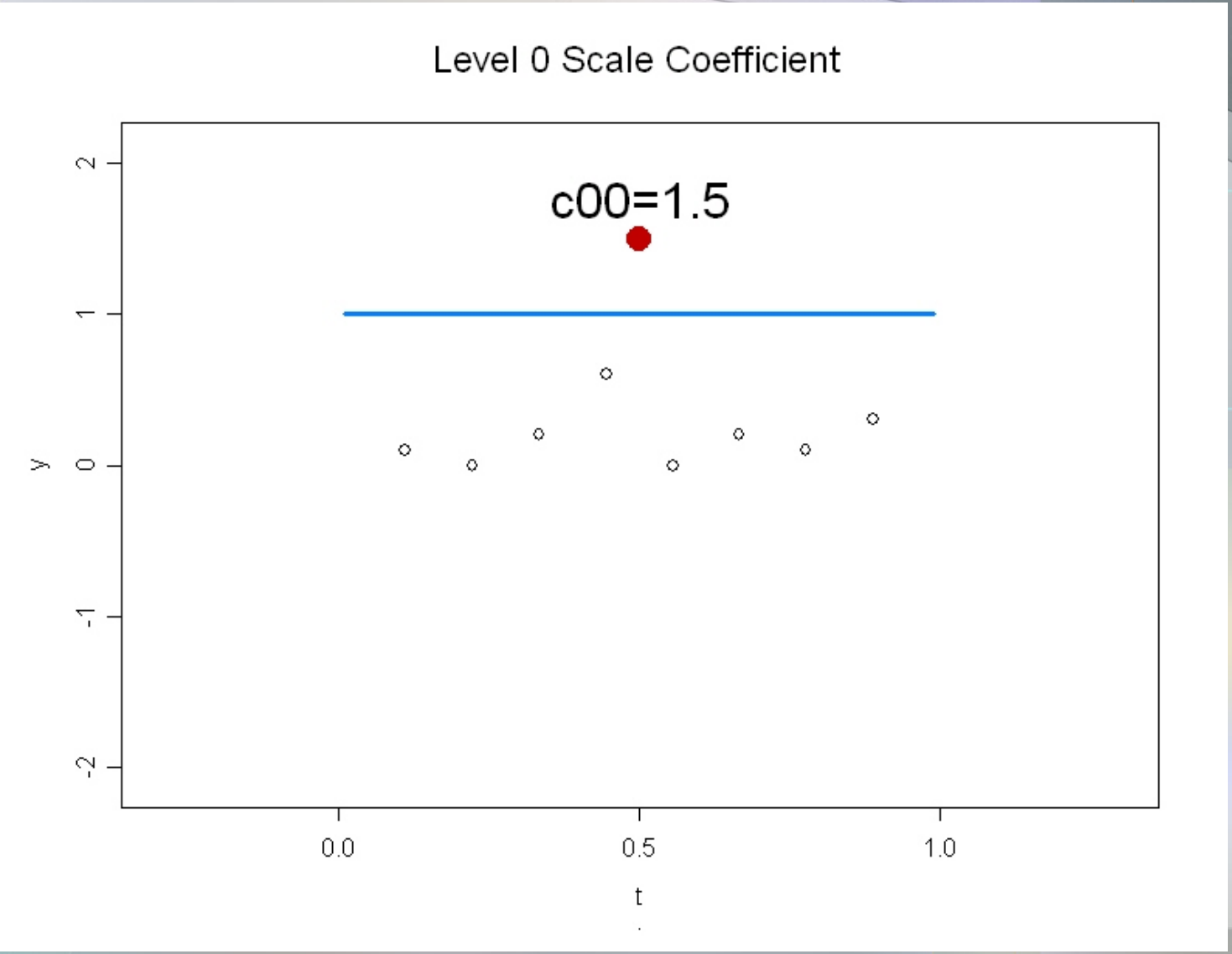
Wavelets: Haar Wavelet Example

$$y = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.2 \\ 0.6 \\ 0.0 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$$



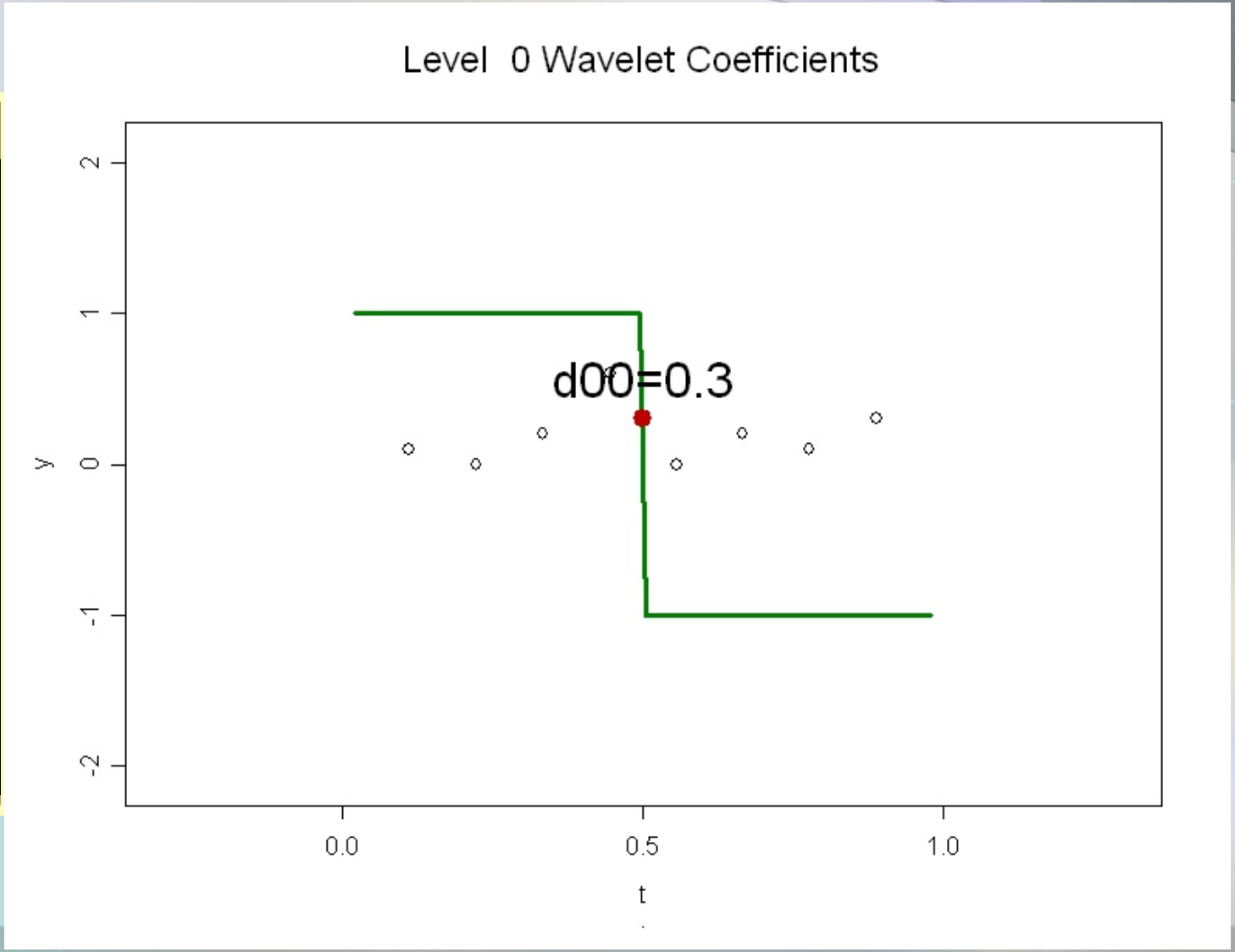
Wavelets: Haar Wavelet Example

$$y = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.2 \\ 0.6 \\ 0.0 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$$



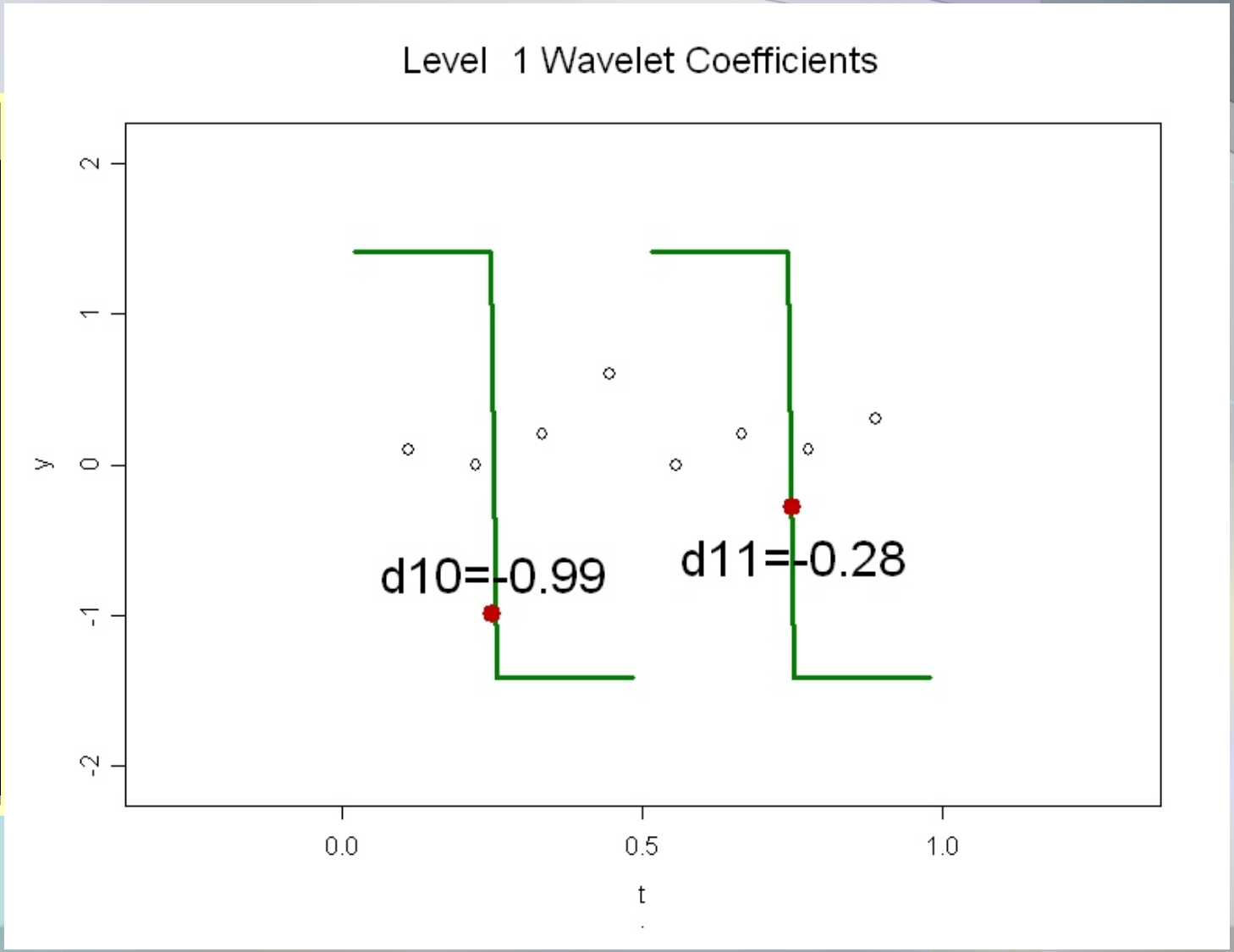
Wavelets: Haar Wavelet Example

$$y = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.2 \\ 0.6 \\ 0.0 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$$



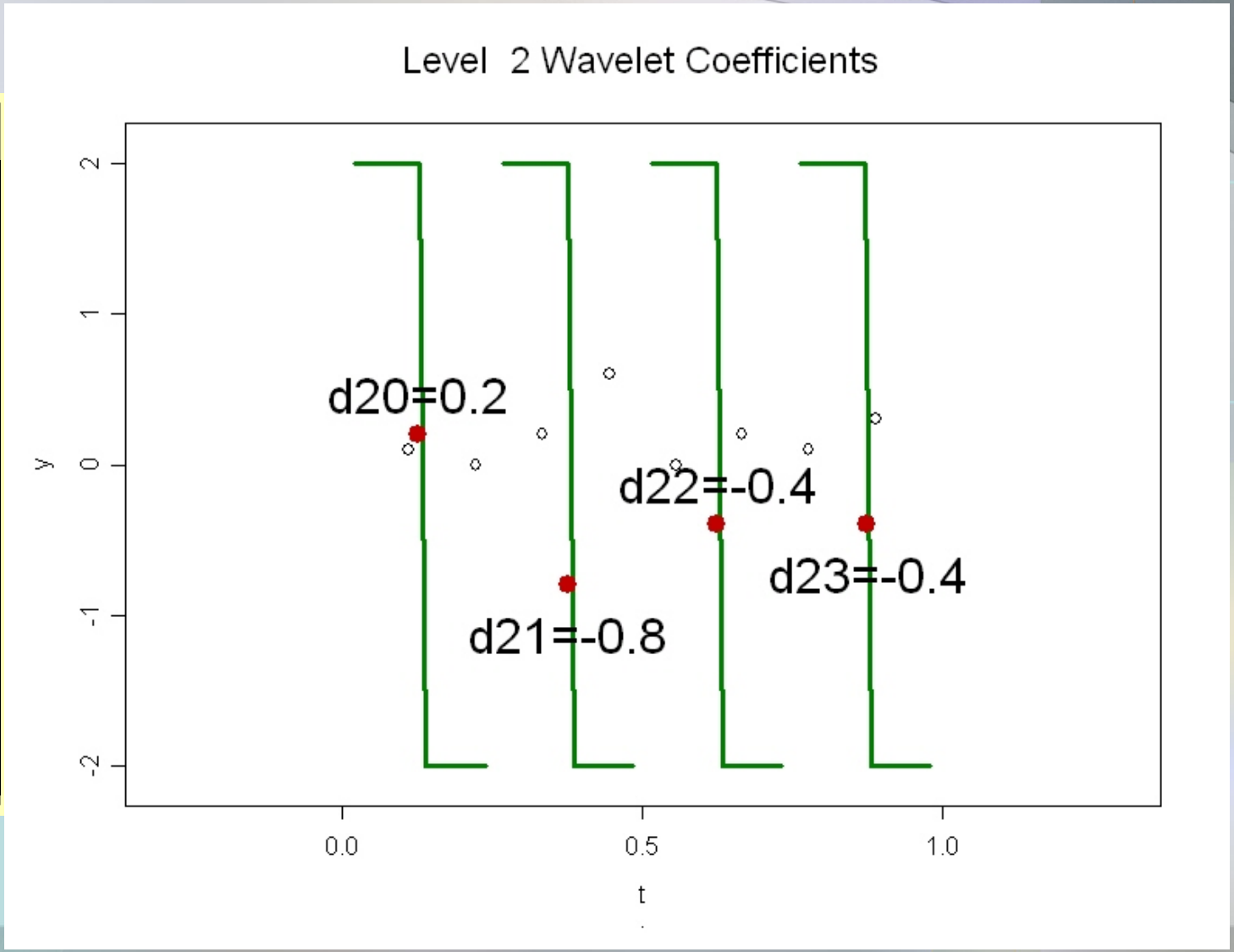
Wavelets: Haar Wavelet Example

$$y = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.2 \\ 0.6 \\ 0.0 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$$



Wavelets: Haar Wavelet Example

$$y = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.2 \\ 0.6 \\ 0.0 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$$



Wavelets: Haar Wavelet Example

$$y' = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.2 \\ 0.6 \\ 0.0 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix} \qquad d' = \begin{bmatrix} c_{00} \\ d_{00} \\ d_{10} \\ d_{11} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.3 \\ -0.99 \\ -0.28 \\ 0.2 \\ -0.8 \\ -0.4 \\ -0.4 \end{bmatrix}$$

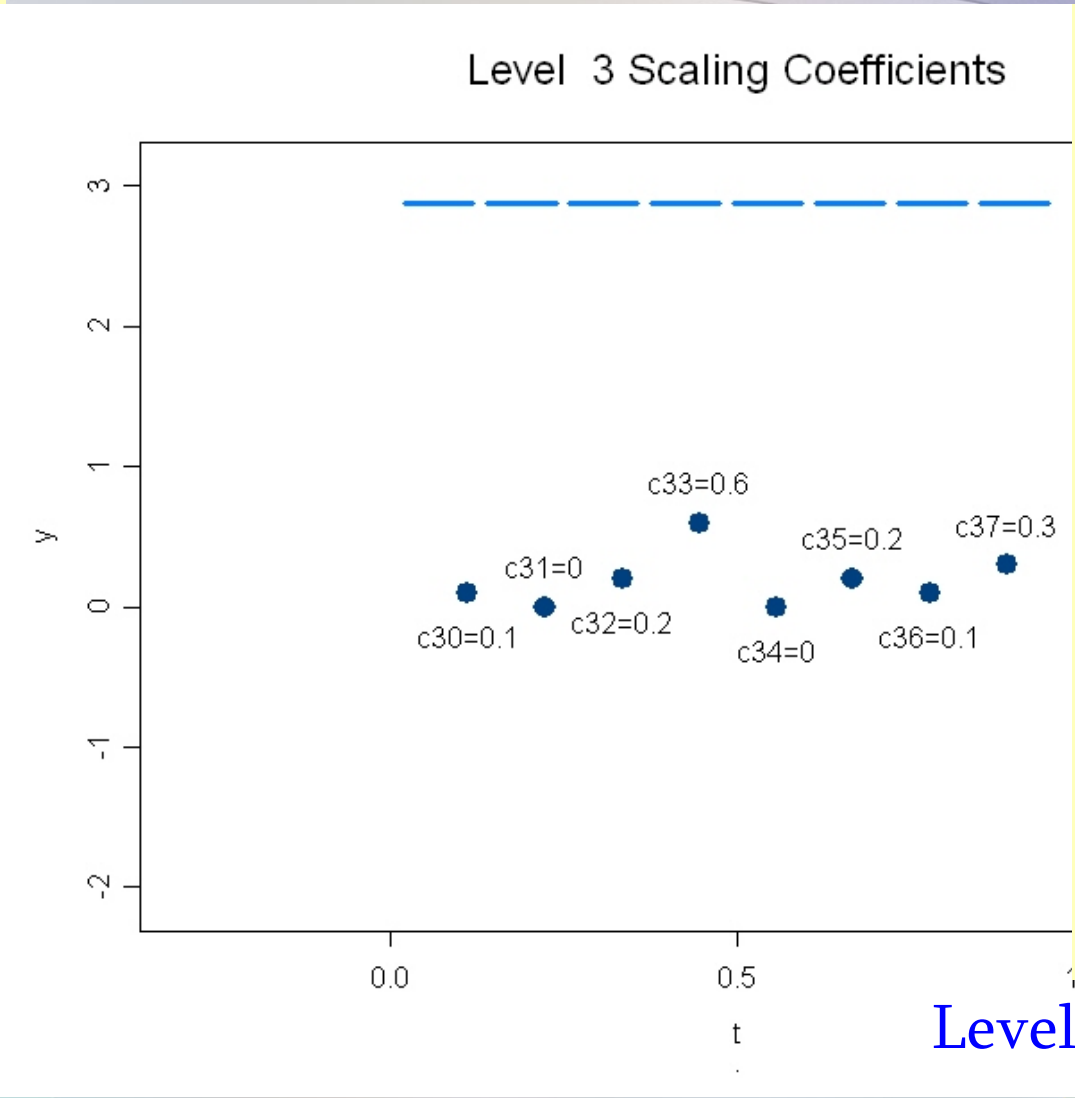
- **Full rank linear transformation from *data space* to *wavelet space***
- **Can be written *y=dW***

$$W' = \underbrace{\begin{bmatrix} \varphi_{00}(t_1) & \cdots & \varphi_{00}(t_8) \\ \psi_{00}(t_1) & \cdots & \psi_{00}(t_8) \\ \vdots & \ddots & \vdots \\ \psi_{23}(t_1) & \cdots & \psi_{23}(t_8) \end{bmatrix}}_{\text{Orthonormal Wavelet Transform Matrix}}$$

- ***W'W=WW'=I***
- **Thus *yW'=dWW'***
  - ***d=yW'***

Wavelets: Haar Wavelet Example

$$y = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.2 \\ 0.6 \\ 0.0 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$$

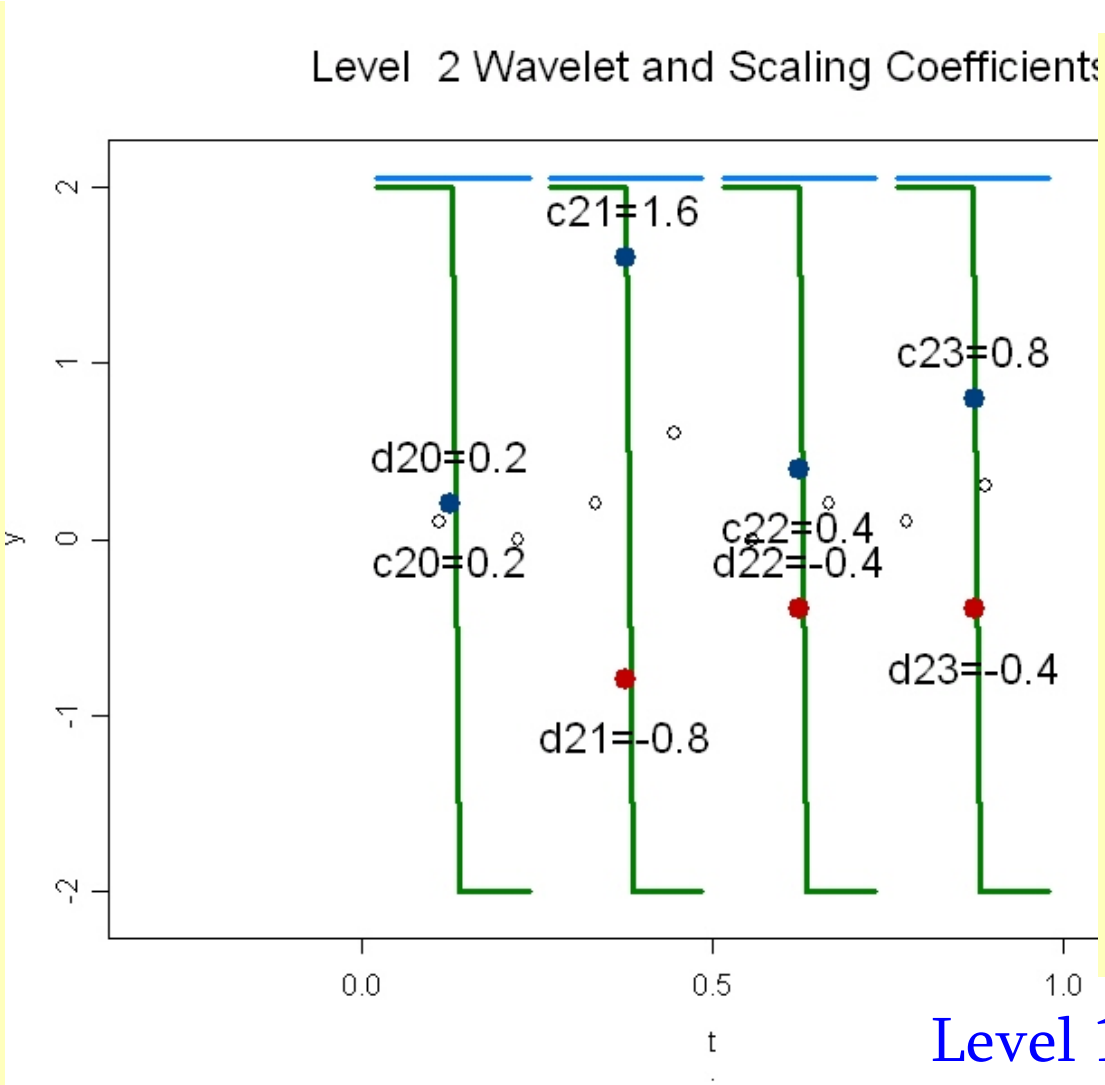


$$d(0) = \begin{bmatrix} c_{30} \\ c_{31} \\ c_{32} \\ c_{33} \\ c_{34} \\ c_{35} \\ c_{36} \\ c_{37} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.2 \\ 0.6 \\ 0.0 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$$

Level 0 Decomposition

Wavelets: Haar Wavelet Example

$$y = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.2 \\ 0.6 \\ 0.0 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$$

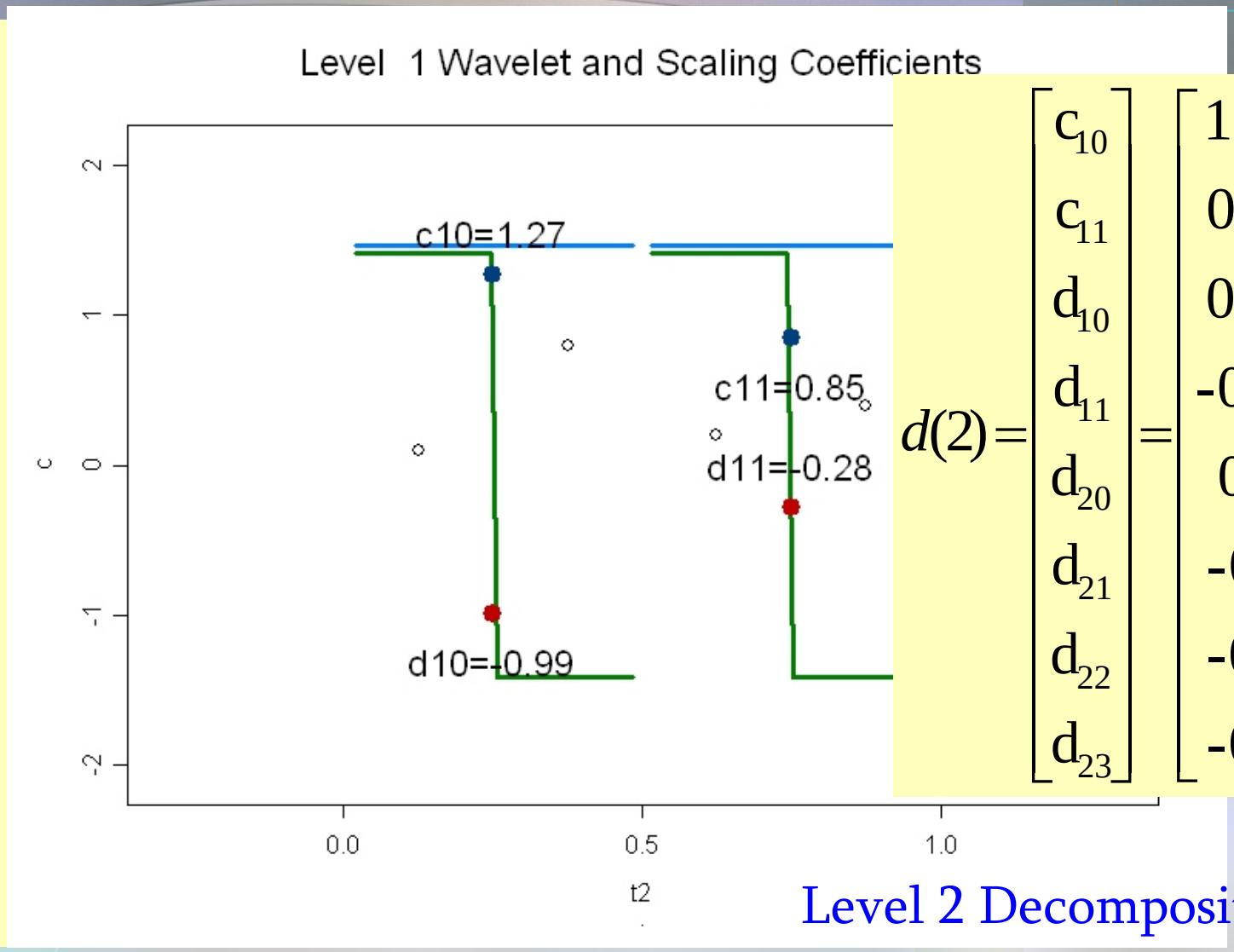


$$d(1) = \begin{bmatrix} c_{20} \\ c_{21} \\ c_{22} \\ c_{23} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix} = \begin{bmatrix} 0.2 \\ 1.6 \\ 0.4 \\ 0.8 \\ 0.2 \\ -0.8 \\ -0.4 \\ -0.4 \end{bmatrix}$$

Level 1 Decomposition

Wavelets: Haar Wavelet Example

$$y = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.2 \\ 0.6 \\ 0.0 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$$

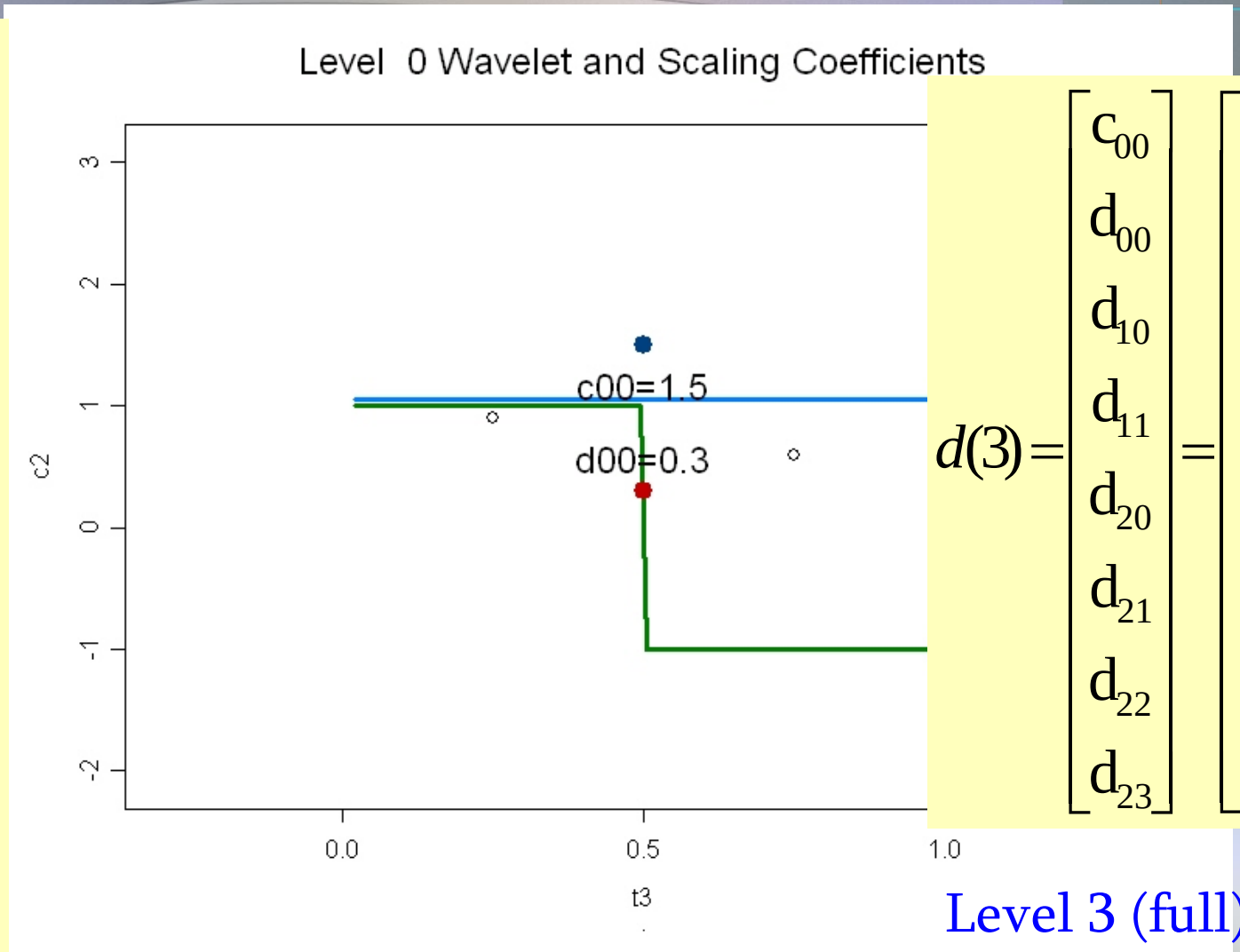


$$d(2) = \begin{bmatrix} c_{10} \\ c_{11} \\ d_{10} \\ d_{11} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix} = \begin{bmatrix} 1.27 \\ 0.85 \\ 0.99 \\ -0.28 \\ 0.2 \\ -0.8 \\ -0.4 \\ -0.4 \end{bmatrix}$$

Level 2 Decomposition

Wavelets: Haar Wavelet Example

$$y = \begin{bmatrix} 0.1 \\ 0.0 \\ 0.2 \\ 0.6 \\ 0.0 \\ 0.2 \\ 0.1 \\ 0.3 \end{bmatrix}$$

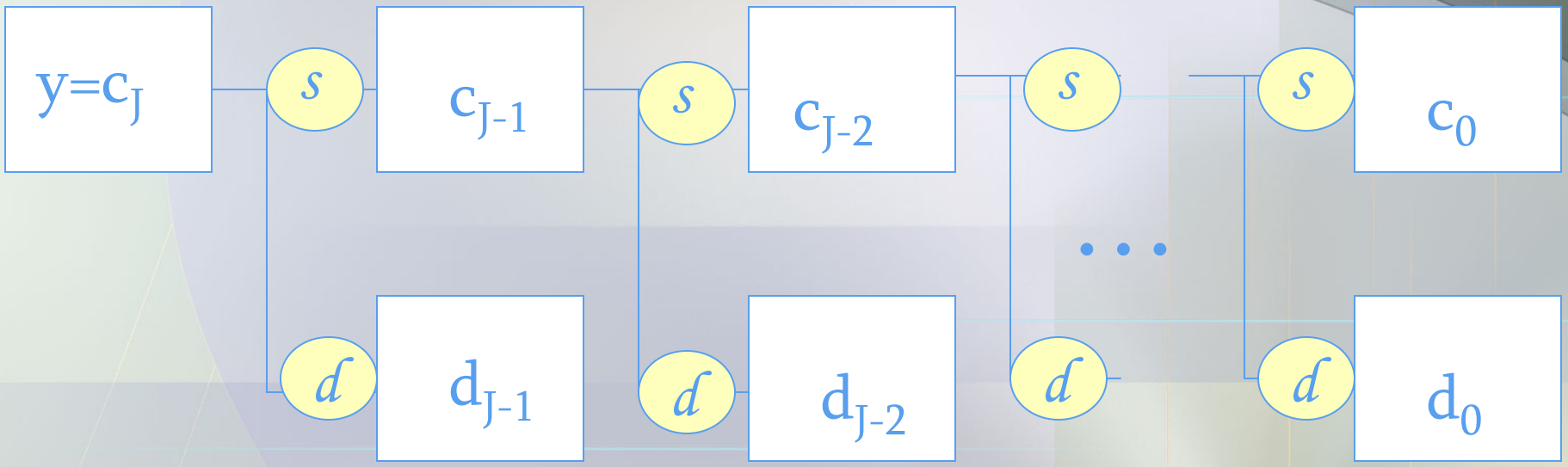


$$d(3) = \begin{bmatrix} c_{00} \\ d_{00} \\ d_{10} \\ d_{11} \\ d_{20} \\ d_{21} \\ d_{22} \\ d_{23} \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.3 \\ 0.99 \\ -0.28 \\ 0.2 \\ -0.8 \\ -0.4 \\ -0.4 \end{bmatrix}$$

Level 3 (full) Decomposition

Wavelets: DWT Pyramid Algorithm

- **Consider function *y(t)* observed on equally spaced grid size $T=2^J$**



Decomposition Level:

0 1 2 J (full)

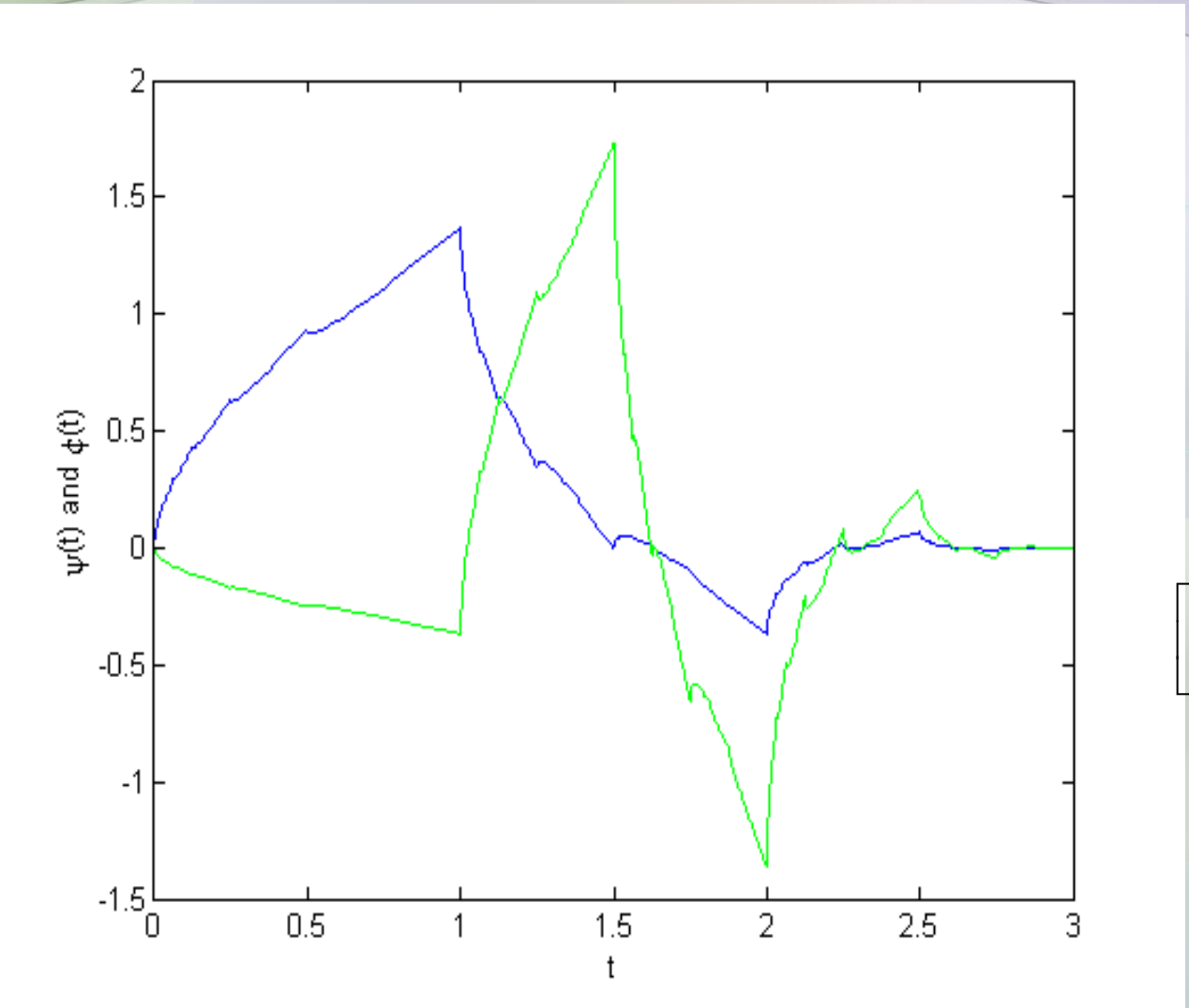
Wavelets: Filter Coefficients

- **Note**: Because of the special construction of the wavelet coefficients, each inner product only involves multiplication of a portion of the data vector by $[2^{-1/2}\ 2^{-1/2}]$ or $[2^{-1/2}\ -2^{-1/2}]$. These are called the Haar's ***filter coefficients***.
- Thus, the matrix W can be generated by convolutions of these filter coefficients within a banded structure. These allow this pyramid-based algorithm to run very quickly $O(T)$. (DWT/IDWT)

Wavelets: Daubechies Family

- **Properties of Haar wavelets:**
  - $\int \varphi(t)\,\psi(t)dt=0$ $\int \varphi(t)dt=\int \varphi^2(t)dt=1$ $\int \psi(t)dt=\int \psi^2(t)dt=0$, *MRA*
  - **Orthonormal transform (*y=dW* for orthogonal *W*)**
  - ***W* determined by two "filter coefficients": $[2^{-1/2}, 2^{-1/2}]$**
  - **Has compact support on *[0,1)***
  - **Vanishing "zeroth" moment:** $\int t^0 \psi(t)dt=\int \psi(t)dt=0$
- **Other wavelet bases:**
  - **Suppose we want wavelet with orthonormality, compact support, but also want vanishing 1st moment** $\int t\,\psi(t)dt=0$
  - **Is there a wavelet/scaling function pair with these properties?**

Wavelets: Daubechies Family



- **No closed form expression**
- **Determined recursively by just 4 filter coefficients**

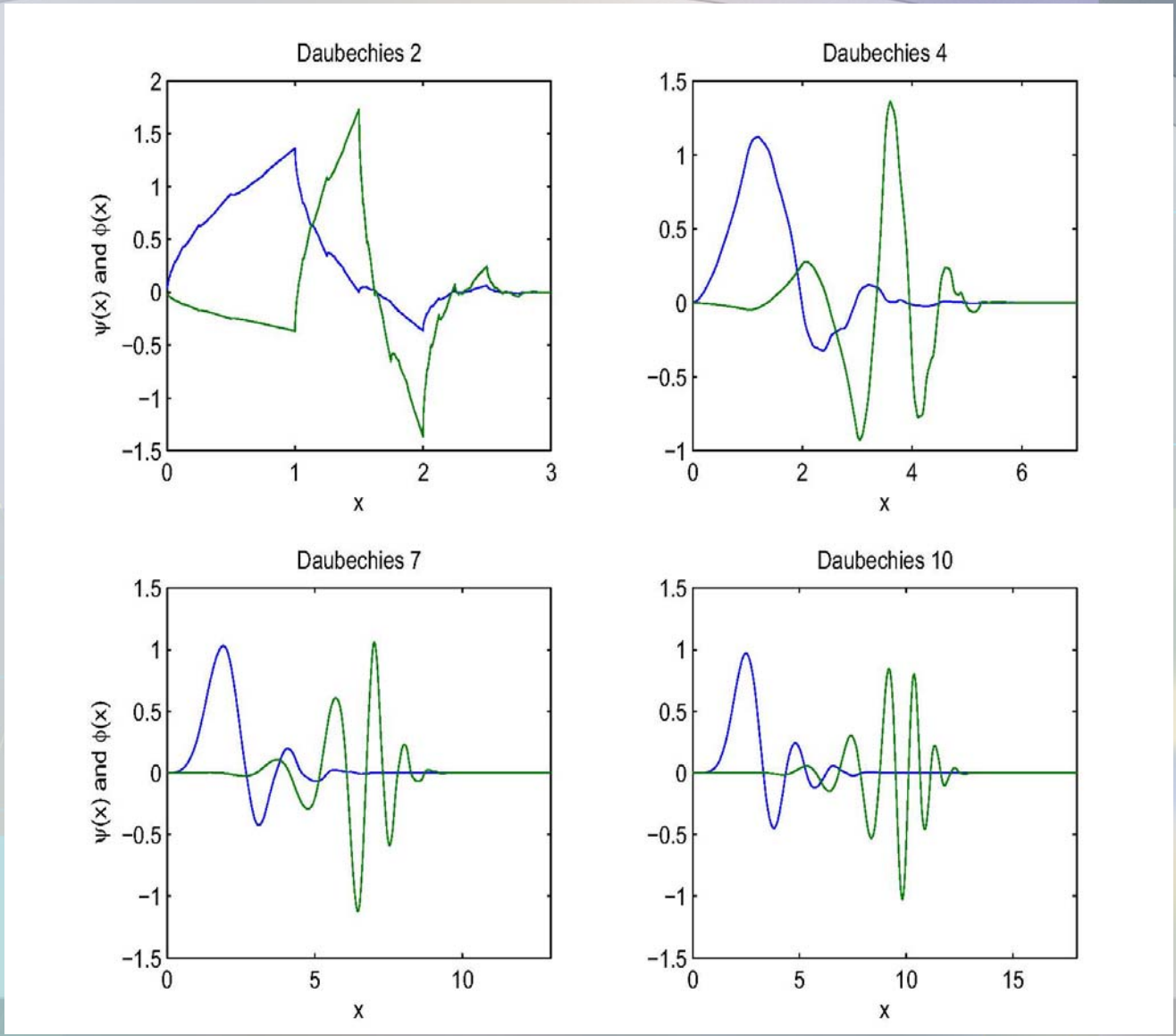
$$\left[\frac{1-\sqrt{3}}{4}, \frac{3-\sqrt{3}}{4}, \frac{3+\sqrt{3}}{4}, \frac{1+\sqrt{3}}{4}\right]$$

- **Compact support on [0,3)**

Wavelets: Daubechies Family

- **Daubechies family of wavelets (Daub 1988):**
- **Indexed by number of vanishing moments (N)**
  - **All Daub wavelets have the properties:**
    - $\int \varphi(t)\,\psi(t)dt=0$
    - $\int \varphi(t)dt=\int \varphi^2(t)dt=1$
    - $\int \psi(t)dt=\int \psi^2(t)dt=0$
    - **Multiresolution Analysis (MRA)**
    - **Orthonormal transform ($y=dW$ for orthogonal W)**
  - **Determined by $L=2N$ filter coefficients**
  - **Has compact support on $[0,2N-1)$**
  - **N vanishing moments: $\int t^a \psi(t)dt=0$ for $a=0, 1, 2,\ldots, N-1$**

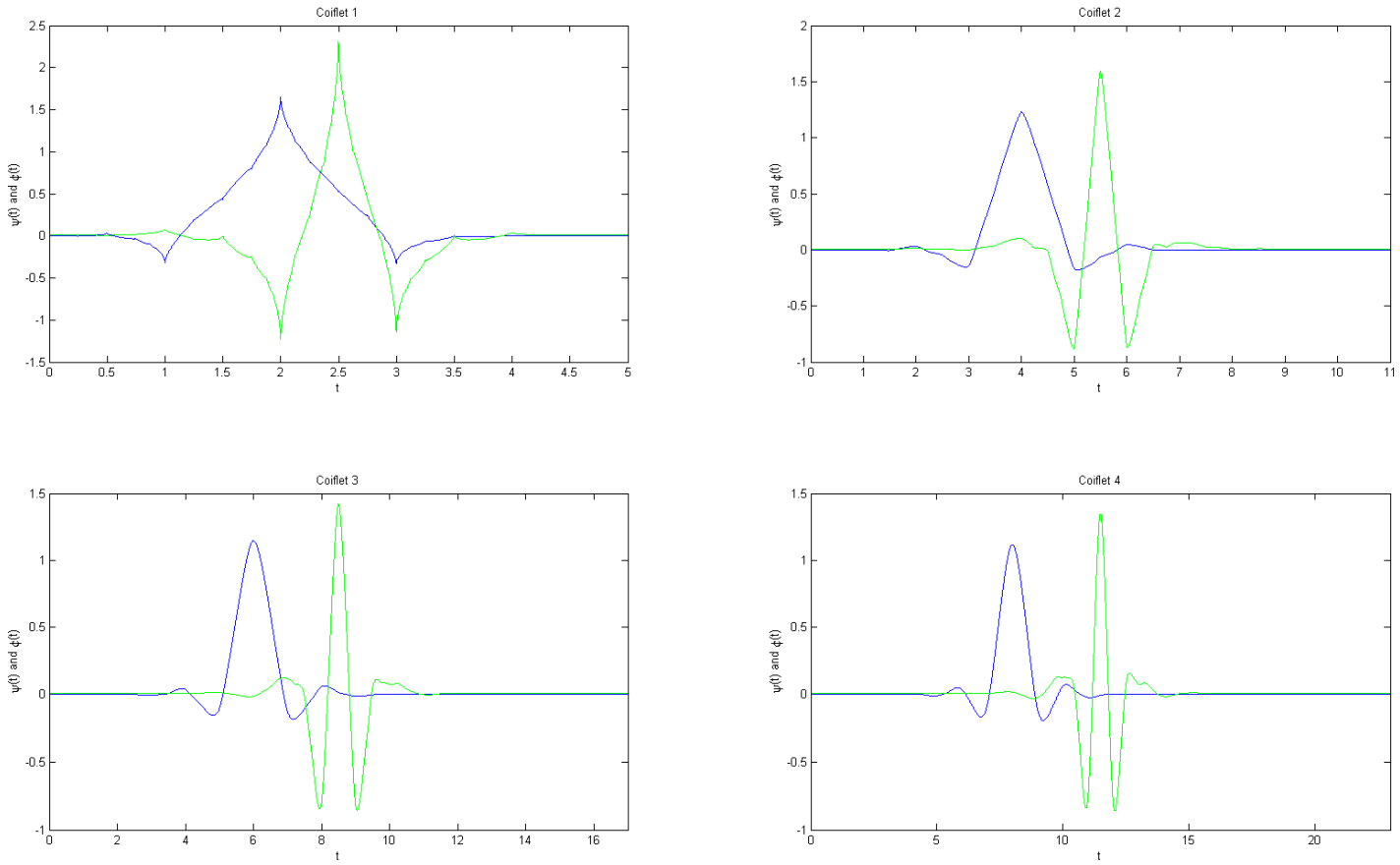
Wavelets: Daubechies Family



Wavelets: Coiflet Family

- **What if we also require L vanishing moments for the scaling function φ?**
- **Coiflet Family (Daubechies 1992, for R Coifman) (N)**
- **All Coiflet wavelets have the properties:**
  - $\int \varphi(t)\,\psi(t)dt=0$
  - $\int \varphi(t)dt=\int \varphi^2(t)dt=1$
  - $\int \psi(t)dt=\int \psi^2(t)dt=0$
  - **Multiresolution Analysis (MRA)**
  - **Orthonormal transform ($y=dW$ for orthogonal W)**
- **Completely determined by $L=6N$ filter coefficients**
- **Has compact support on $[0,6N-1)$**
- **Wavelets: $2N$ vanishing moments: $\int t^a \psi(t)dt=0$ for $a=0, 1, 2,\ldots, 2N-1$**
- **Scaling: $2N-1$ vanishing moments: $\int t^a \varphi(t)dt=0$ for $a=0, 1, 2,\ldots, 2N-2$**

Wavelets: Coiflet Family



Wavelets: Summary

$$y(t) = \sum_{j,k \in \Im} d_{jk} \psi_{jk}(t)$$

$$d_{jk} = \int y(t) \psi_{jk}(t) dt$$

$$\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k)$$

Linear Representation:

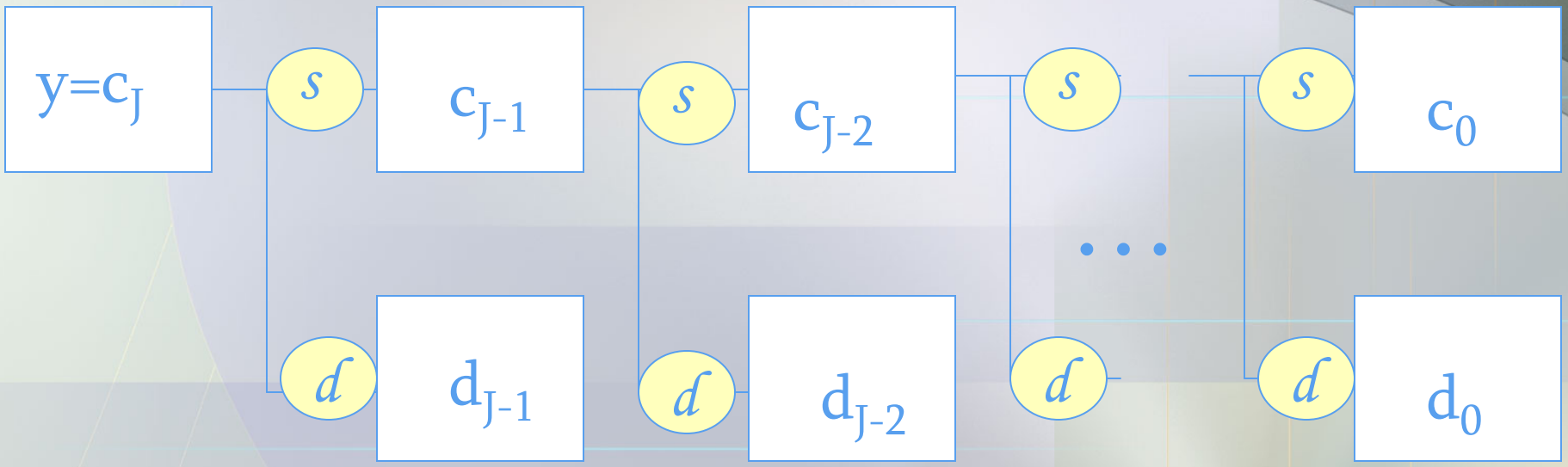
$$\overbrace{\mathbf{y}}^{1 \times T} = \underbrace{\mathbf{d}}_{1 \times T} \overbrace{\mathbf{W}}^{T \times T}$$

$$\overbrace{\mathbf{d}}^{1 \times T} = \underbrace{\mathbf{y}}_{1 \times T} \overbrace{\mathbf{W}'}^{T \times T}$$

- **Given *T*-vector y consisting of function sampled on equally-spaced grid, a pyramid-based algorithm for DWT (Mallat) can be used to obtain d, *T* –vector of wavelet coefficients for the given family, in *O(T)* operations (converse also true)**
- **We never need to compute or think about ψ or φ, since we can apply the DWT using only the wavelet family's filter coefficients**

Wavelets: DWT Pyramid Algorithm

- **Consider function *y(t)* observed on equally spaced grid size $T=2^J$**



Decomposition Level:

0 1 2 J (full)

- **DWT: Can be computed in *O(T)*, Also IDWT**
- **Note: The low pass and high pass filters, *s* and *d*, are determined solely by the wavelet's filter coefficients**

Wavelets: Practical Issues

- ***T* not power of *2*?**
  - **Adjustments to DWT possible (Percival&Walden 2000)**
- **Choice of wavelet basis :match data features**
  - **Piecewise constant (aCGH): Haar**
  - **Smoother functions: Daub#N with N higher**
  - **Symmetric features?: Symmlets or Coiflets**
  - **Tradeoff: smoothness vs. filter length**
- **Boundary correction issues (all but Haar)**
  - **Periodic, Pad with zeros, Reflection**
- ***J*=Number of levels of decomposition**
  - **Full decomposition: $J=\log_2(T)$**
  - **Choose J: $\text{floor}(K_{J-1}/2)>L-1$ (at least one coefficient in middle unaffected by boundary conditions, PW2000)**

Wavelets: Properties

- ***Very fast calculation*** **:** ***O(T)***
- ***Compact support*** **: good for spatially heterogeneous data**
- ***Signal compression*****: concentration on few *d***
- ***Orthogonality*****: distributes white noise across *d***
- ***Whitening*****: autocorrelation of wavelet coefficients tends to die away rapidly across *k*; d_{jk} approximately uncorrelated (Johnstone and Silverman 1997)**
- ***Time/frequency decomposition*****: key to adaptive smoothing**

Wavelets: Wavelet Regression

- **Wavelet Regression:**
  - **Row vector *y*: response on equally-spaced grid *t* (length *T*)**

$$\mathbf{y} = g(\mathbf{t}) + \varepsilon \qquad \varepsilon \sim N(0, \sigma^2 I_T)$$

1. **Project data into wavelet space using DWT.**

   d=yW' **where *W'* is the orthogonal DWT matrix**

Recall: $d=\{d_{jk}, j=1, \ldots, J; k=1, \ldots, K_j\}$

Wavelets: Wavelet Regression

- **Wavelet Regression:**
  - **Row vector *y*: response on equally-spaced grid *t* (length *T*)**

$$\mathbf{y}W' = g(\mathbf{t})W' + \varepsilon W' \qquad \varepsilon \sim N(0, \sigma^2 I_T)$$

1. **Project data into wavelet space using DWT.**

   d=yW' **where *W'* is the orthogonal DWT matrix**

Wavelets: Wavelet Regression

- **Wavelet Regression:**
  - **Row vector *y*: response on equally-spaced grid *t* (length *T*)**

$$\mathbf{d} = g(\mathbf{t})W' + \varepsilon W' \qquad \varepsilon \sim N(0, \sigma^2 I_T)$$

1. **Project data into wavelet space using DWT.**

   d=yW' **where *W'* is the orthogonal DWT matrix**

Wavelets: Wavelet Regression

- **Wavelet Regression:**
  - **Row vector *y*: response on equally-spaced grid *t* (length *T*)**

$$\mathbf{d} = \boldsymbol{\theta} + \varepsilon W' \qquad \varepsilon \sim N(0, \sigma^2 I_T)$$

1. **Project data into wavelet space using DWT.**

   d=yW' **where *W'* is the orthogonal DWT matrix**

Note: $\theta=\{\theta_{jk}: j=1, \ldots, J; k=1, \ldots, K_j\}$

By sparsity property, we expect few $|\theta_{jk}|>0$

Wavelets: Wavelet Regression

- **Wavelet Regression:**
  - **Row vector *y*: response on equally-spaced grid *t* (length *T*)**

$$\mathbf{d} = \boldsymbol{\theta} + \varepsilon^* \qquad \varepsilon \sim N(0, \sigma^2 I_T)$$

1. **Project data into wavelet space using DWT.**

   d=yW' **where *W'* is the orthogonal DWT matrix**

Note: $\varepsilon^* = \varepsilon W' \sim N(0, W\sigma^2 I_T W')$

$\sim N(0, \sigma^2 WW')$

$\sim N(0, \sigma^2 I_T)$ **since** $WW' = I_T$ **(orthogonality)**

Wavelets: Wavelet Regression

- **Wavelet Regression:**
  - **Row vector *y*: response on equally-spaced grid *t* (length *T*)**

$$\mathbf{d} = \boldsymbol{\theta} + \varepsilon^* \qquad \varepsilon^* \sim N(0, \sigma^2 I_T)$$

1. **Project data into wavelet space using DWT.**

   d=y W' **where *W'* is the orthogonal DWT matrix**

2. **Estimate** θ **by *hard thresholding* :** $\theta_{jk} = d_{jk} * I(|d_{jk}| > \delta\sigma^2)$

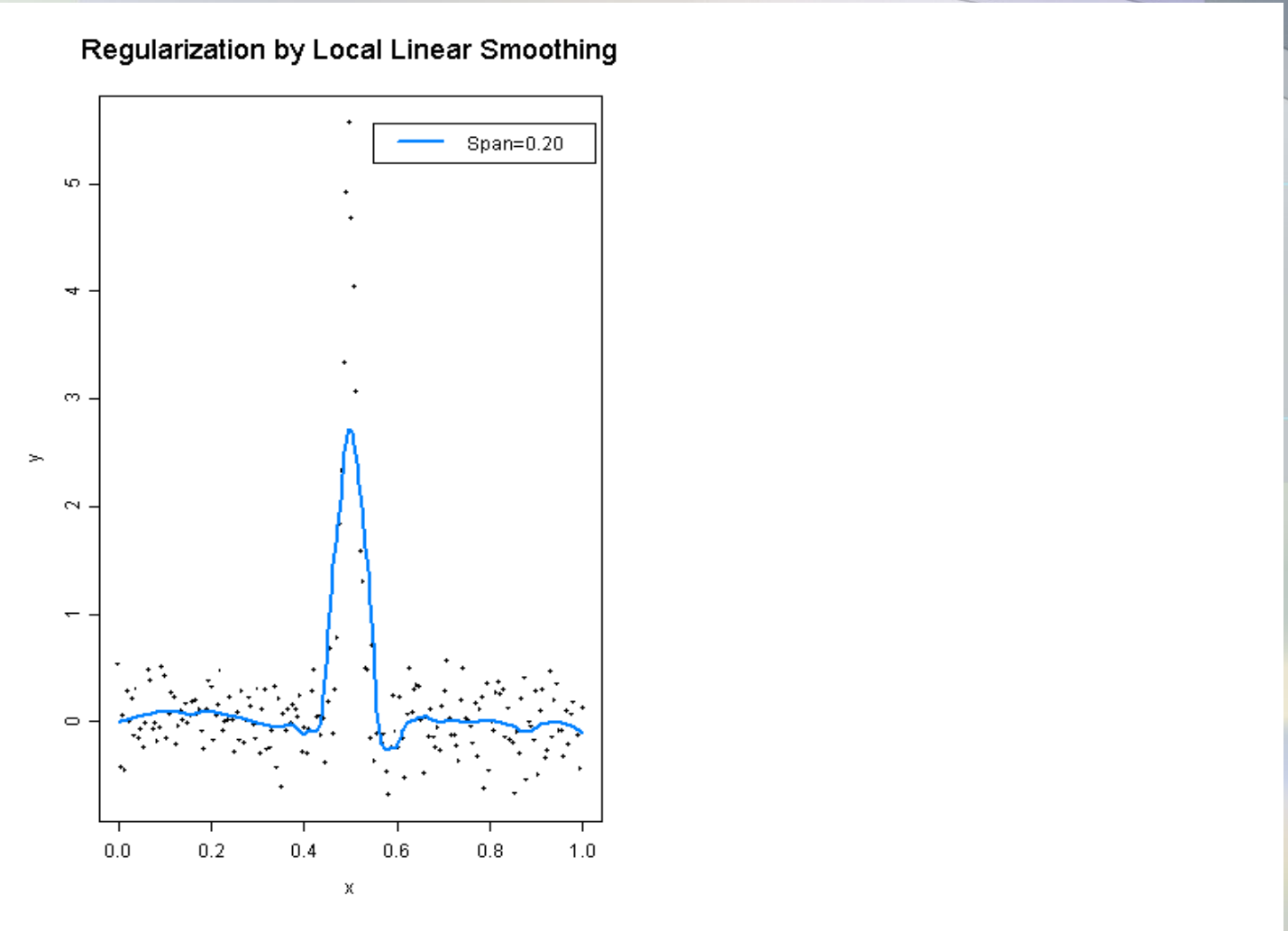
   $\delta = \sqrt{2\log(T)}$ is Universal threshold (DJ1994)

3. **Project back to data space using IDWT**

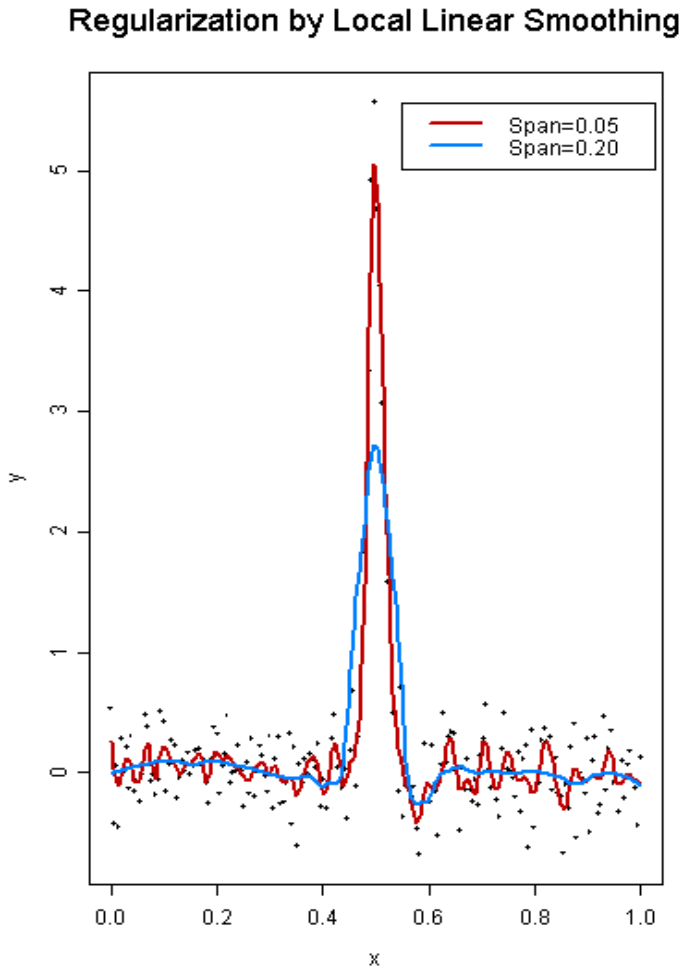
$$\hat{g}(t) = \hat{\boldsymbol{\theta}} W$$

- **Yields *adaptive regularized* nonparametric estimate of *g(t)*.**

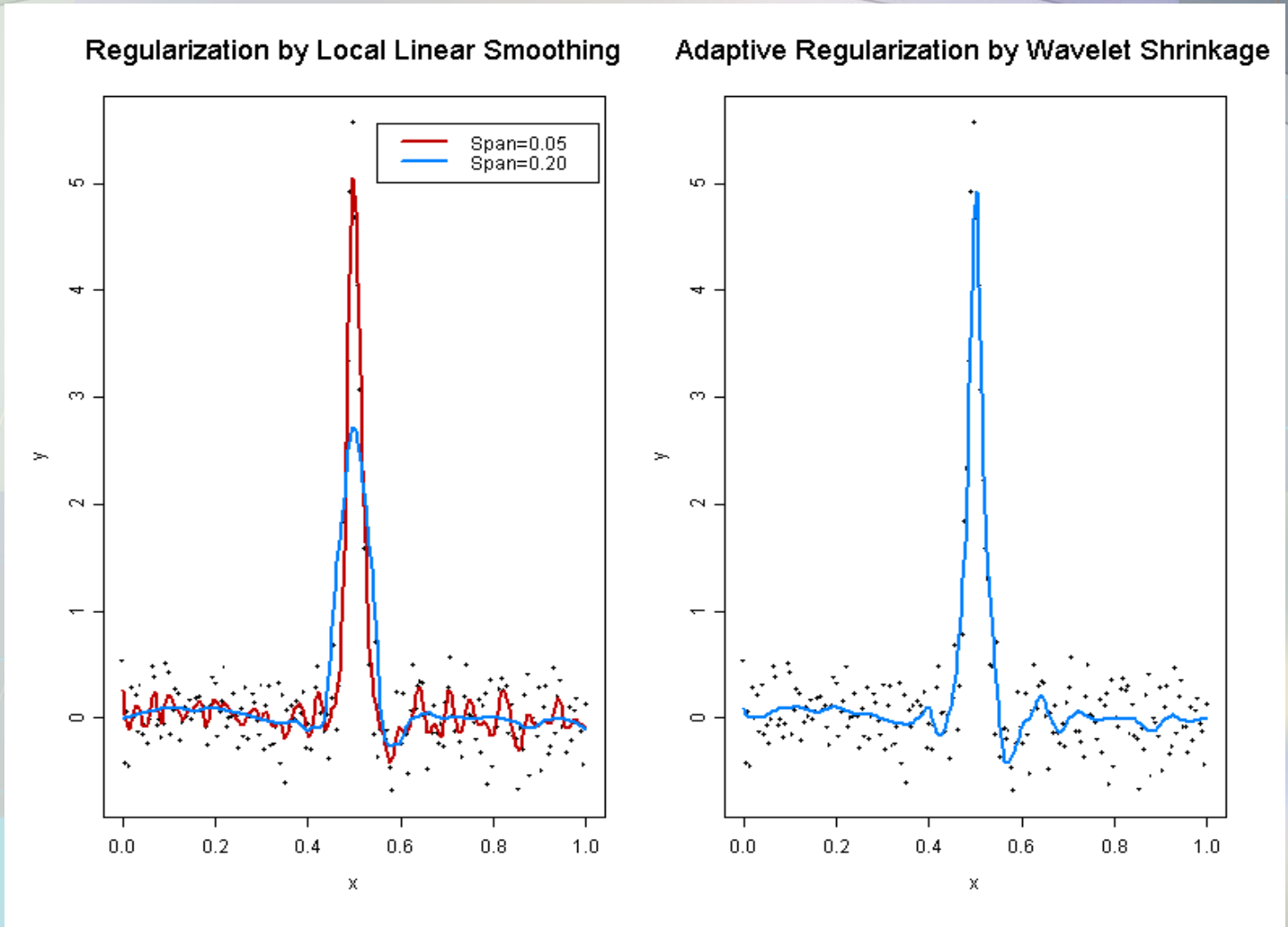
Wavelets: Wavelet Regression



Wavelets: Wavelet Regression



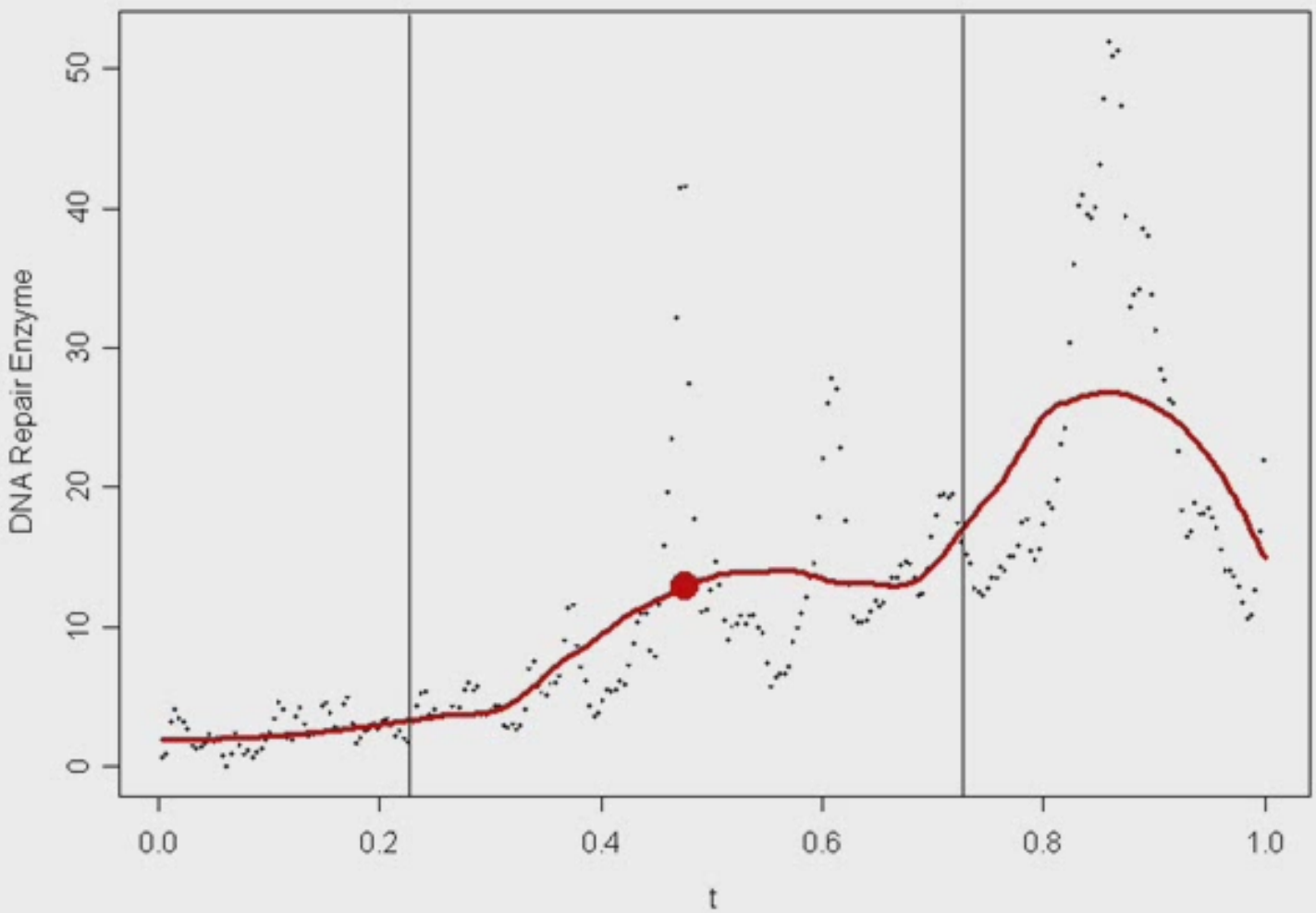
Wavelets: Wavelet Regression



Return

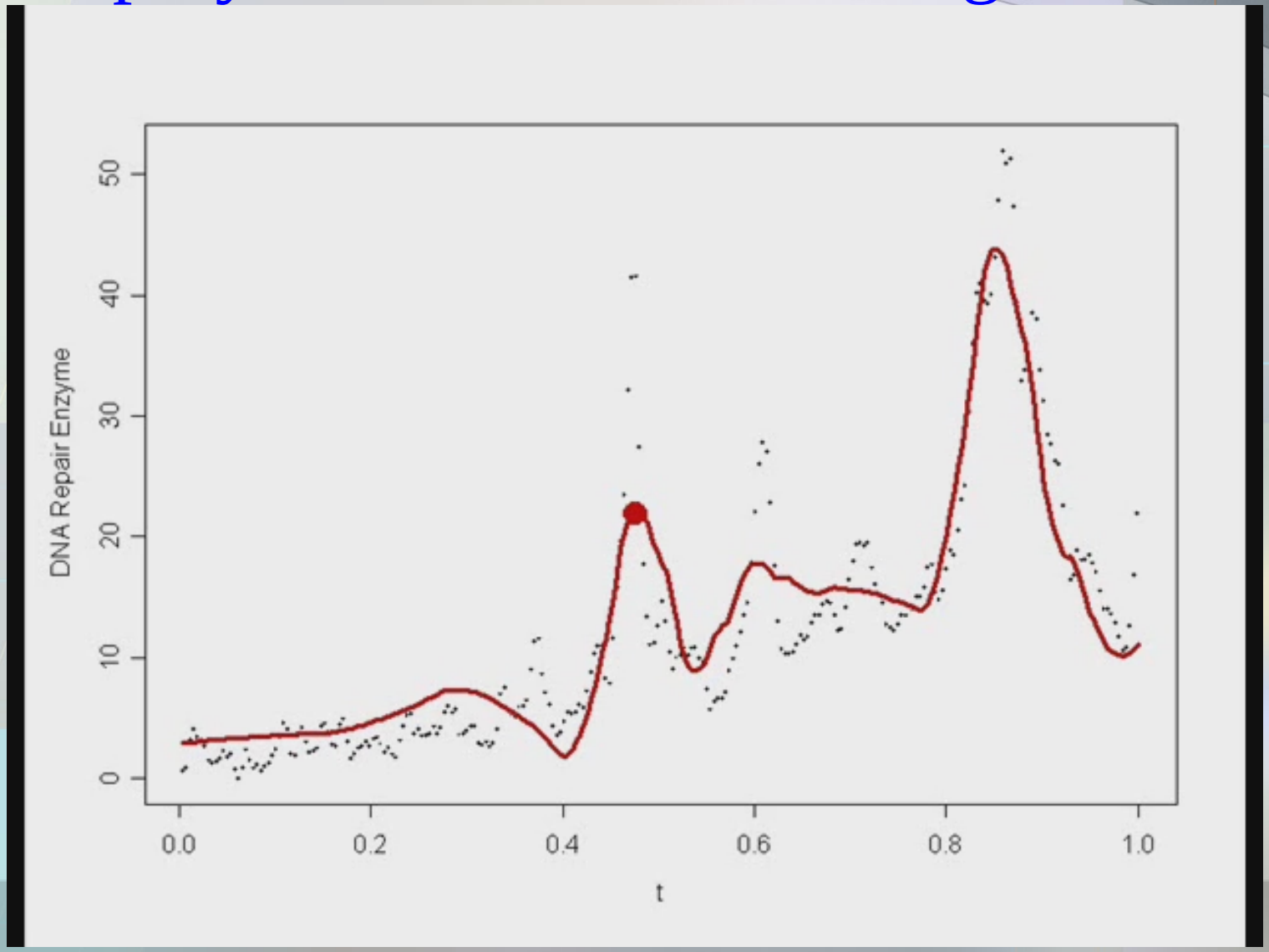
Wavelets: Wavelet Regression

Spiky function: Local Linear Regression



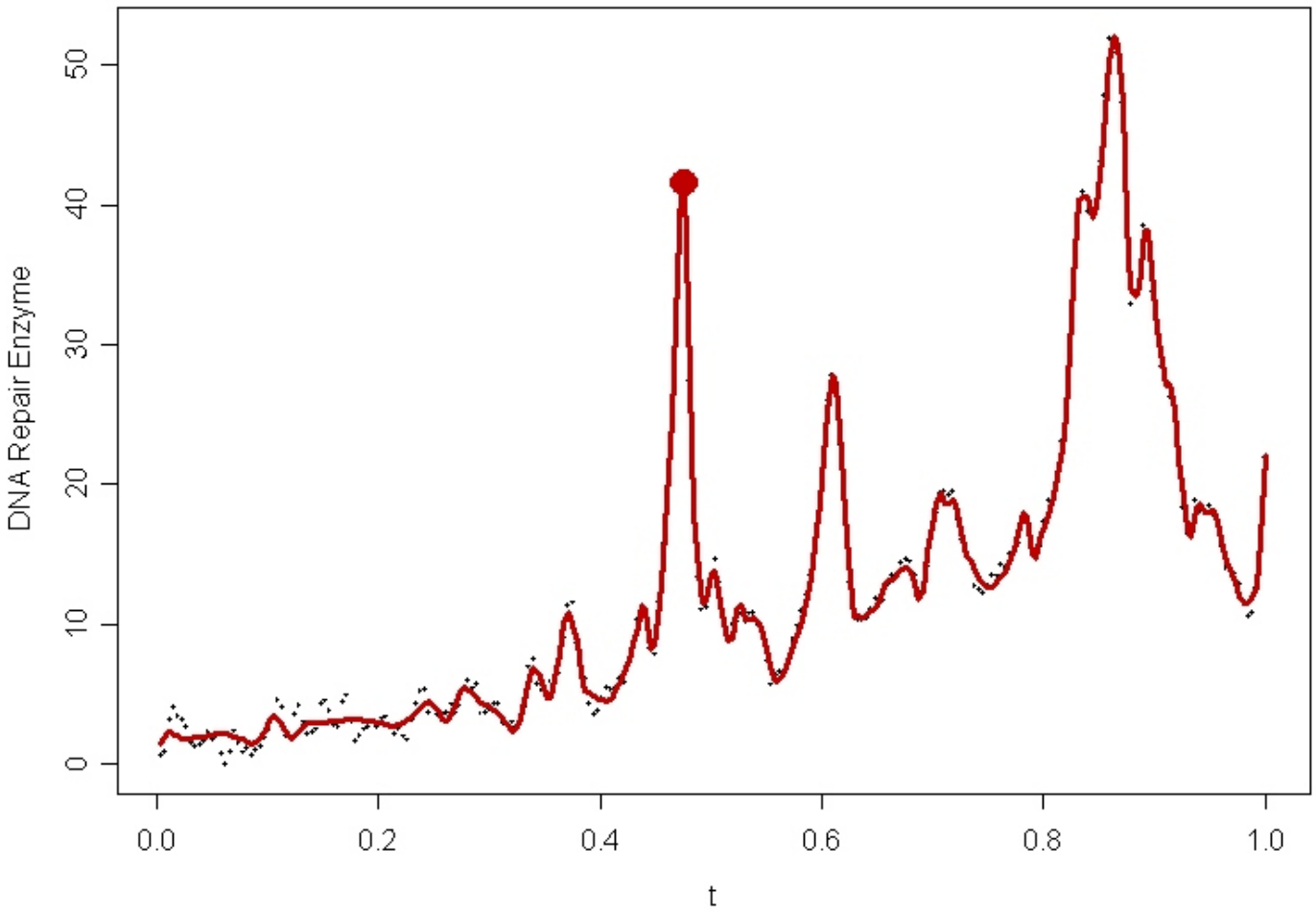
Wavelets: Wavelet Regression

Spiky function: Wavelet Regression



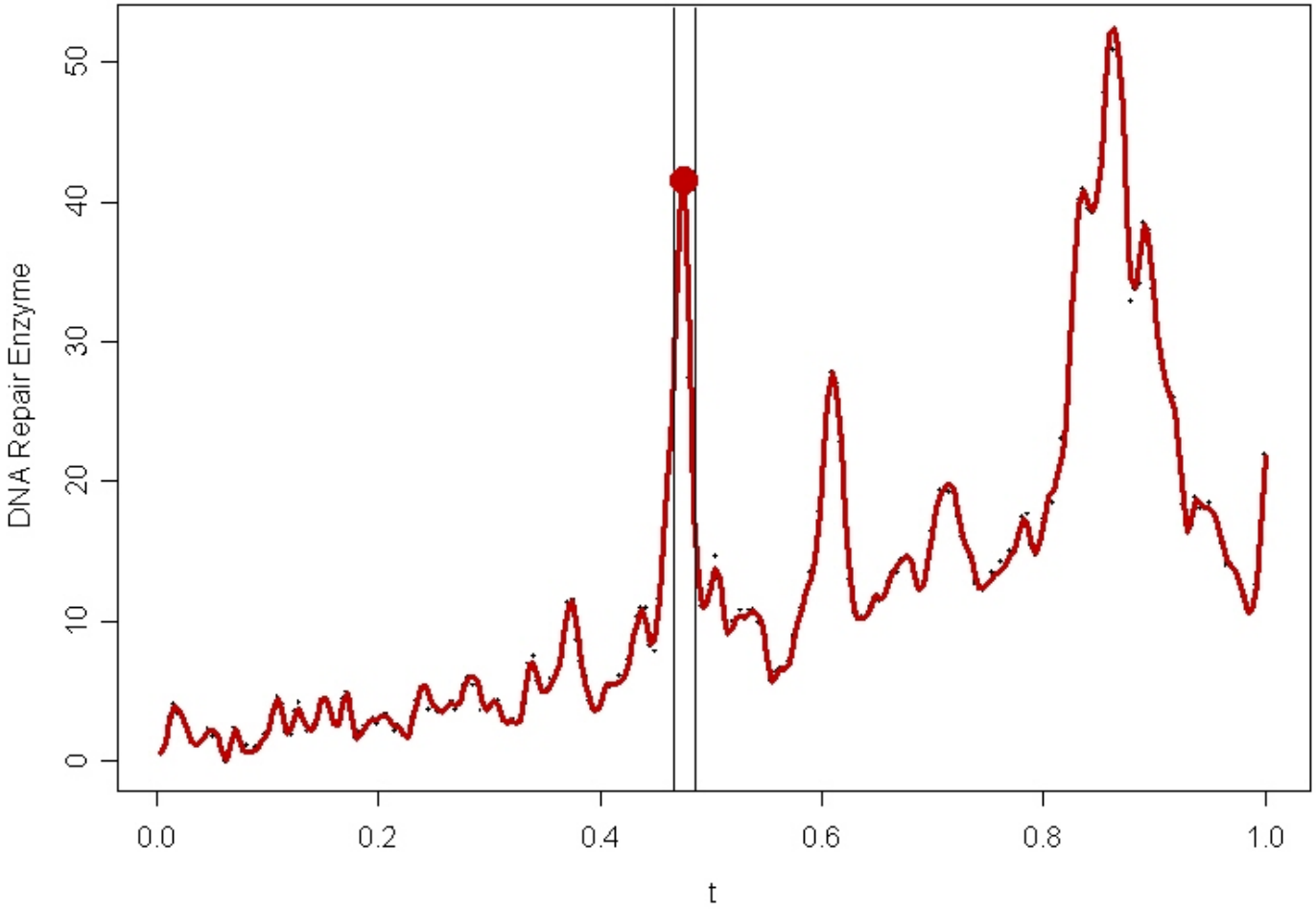
Wavelets: Wavelet Regression

Donoho and Johnstone's Universal threshold



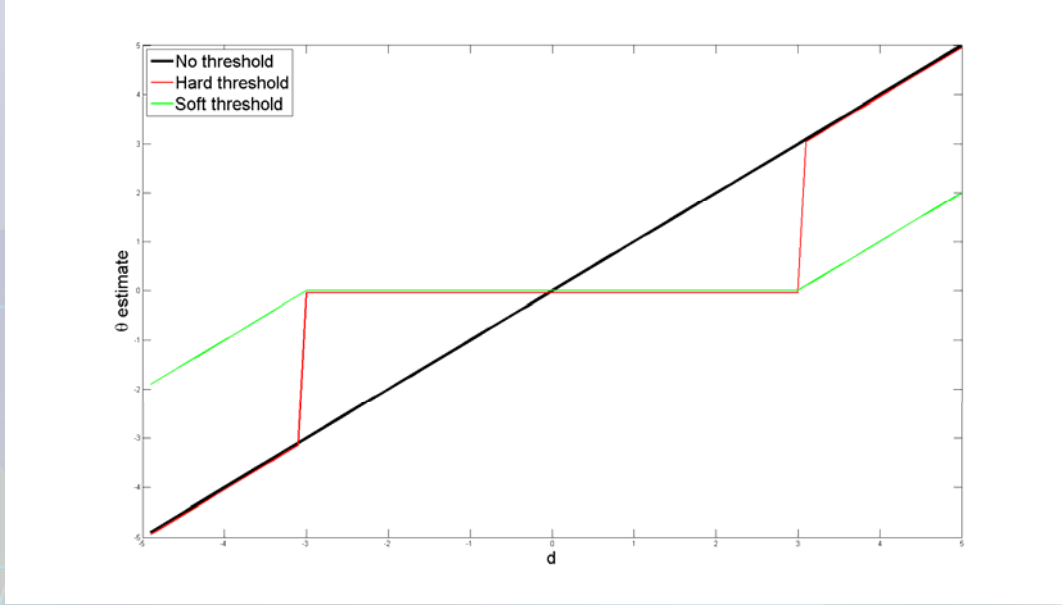
Wavelets: Wavelet Regression

Local Quadratic Regression: maximum *h* to preserve spike



Wavelets: Wavelet Regression

- **Hard vs. Soft Thresholding**
  - Hard thresholding: $\theta_{jk}=d_{jk}*I(|d_{jk}|>\delta\sigma^2)$
  - Soft thresholding: $\theta_{jk}=\mathrm{sgn}(d_{jk})\times(|d_{jk}|-\delta\sigma^2)^+$
  - Hard: preserves peak heights better at cost of smoothness



- **Universal vs. Level Dependent Thresholds**
  - Level dependent thresholds $\delta\sigma_j^2$ perform even better

Wavelets: Wavelet Regression

- **Bayesian hierarchical approach : Prior distribution on θ_{jk} can mimic the idea of thresholding**

$$d_{jk} = N(\theta_{jk}, \sigma^2) \qquad \theta_{jk} = \gamma_{jk} N(0, \tau_j^2 \sigma^2) + (1 - \gamma_{jk})\delta_0$$

$$\gamma_{jk} = \text{Bernoulli}(\pi_j)$$

- **τ_j^2 and π_j are regularization parameters**
  - **Can be specified, or estimated from data using ML in empirical Bayes fashion (Clyde and George 2000)**
- **Conjugate model (given σ^2), so closed form $(\theta_{jk}|d_{jk}, \sigma^2)$**

Wavelets: Wavelet Regression

$$(\theta_{jk} \mid d_{jk}, \sigma^2) = P_{jk} N\left(d_{jk} SF_j, \sigma^2 SF_j\right) + (1 - P_{jk}) I_0$$

$$SF_j = \frac{\tau_j^2}{\tau_j^2 + 1} = \text{"Linear shrinkage factor"}$$

$$P_{jk} = \Pr(\gamma_{jk} = 1 \mid d_{jk}) = \text{"Nonlinear shrinkage factor"}$$

$$\Pr\left(\gamma_{jk} = 1 \mid \mathbf{d}_{jk}\right) = \frac{O_{jk}}{O_{jk} + 1}, \qquad O_{jk} = \text{Posterior Odds}$$

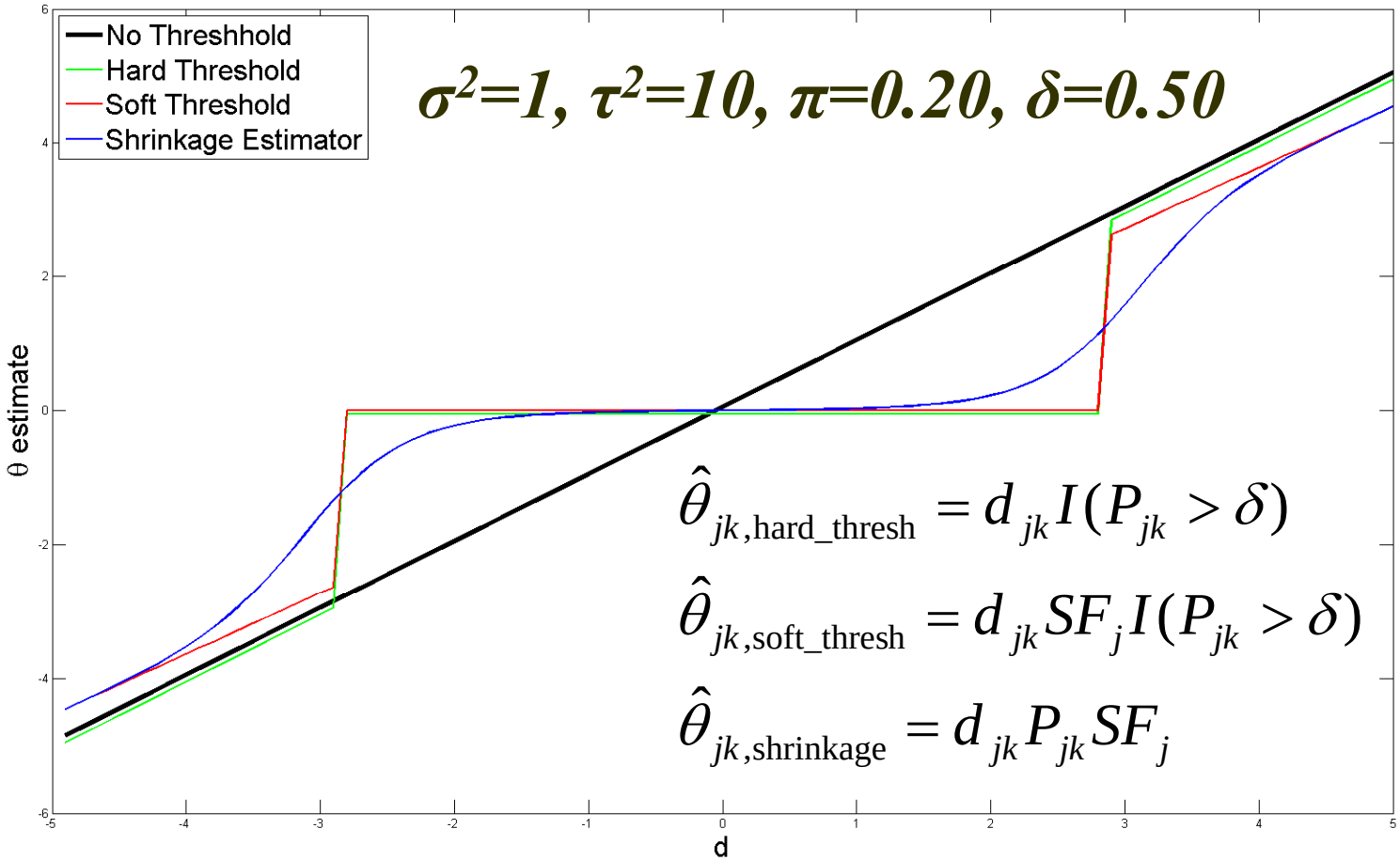
$$\underbrace{O_{jk}}_{\textbf{Posterior Odds}} = \underbrace{\left(\frac{\pi_j}{1 - \pi_j}\right)}_{\textbf{Prior Odds}} \underbrace{\left(1 + \tau_j^2\right)^{-1} \exp\left(\frac{d_{jk}^2 SF_j}{2\sigma^2}\right)}_{\textbf{Bayes Factor}}$$

$$\hat{\theta}_{jk,\text{hard\_thresh}} = d_{jk} I(P_{jk} > \delta)$$

$$\hat{\theta}_{jk,\text{soft\_thresh}} = d_{jk} SF_j I(P_{jk} > \delta)$$

$$\hat{\theta}_{jk,\text{shrinkage}} = d_{jk} P_{jk} SF_j$$

Wavelets: Wavelet Regression

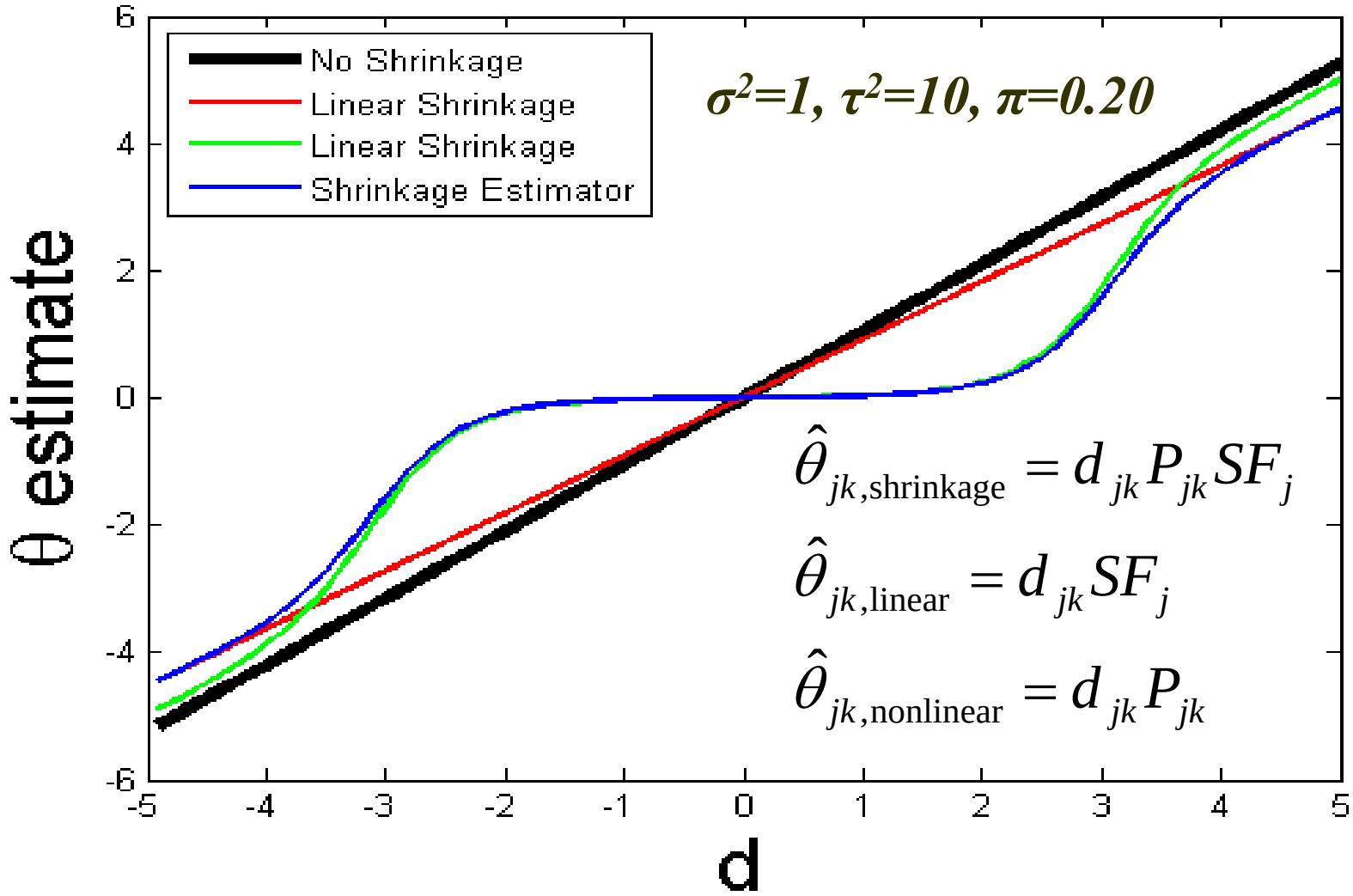


$$\hat{\theta}_{jk,\text{hard\_thresh}} = d_{jk} I(P_{jk} > \delta)$$

$$\hat{\theta}_{jk,\text{soft\_thresh}} = d_{jk} SF_j I(P_{jk} > \delta)$$

$$\hat{\theta}_{jk,\text{shrinkage}} = d_{jk} P_{jk} SF_j$$

Wavelets: Wavelet Regression

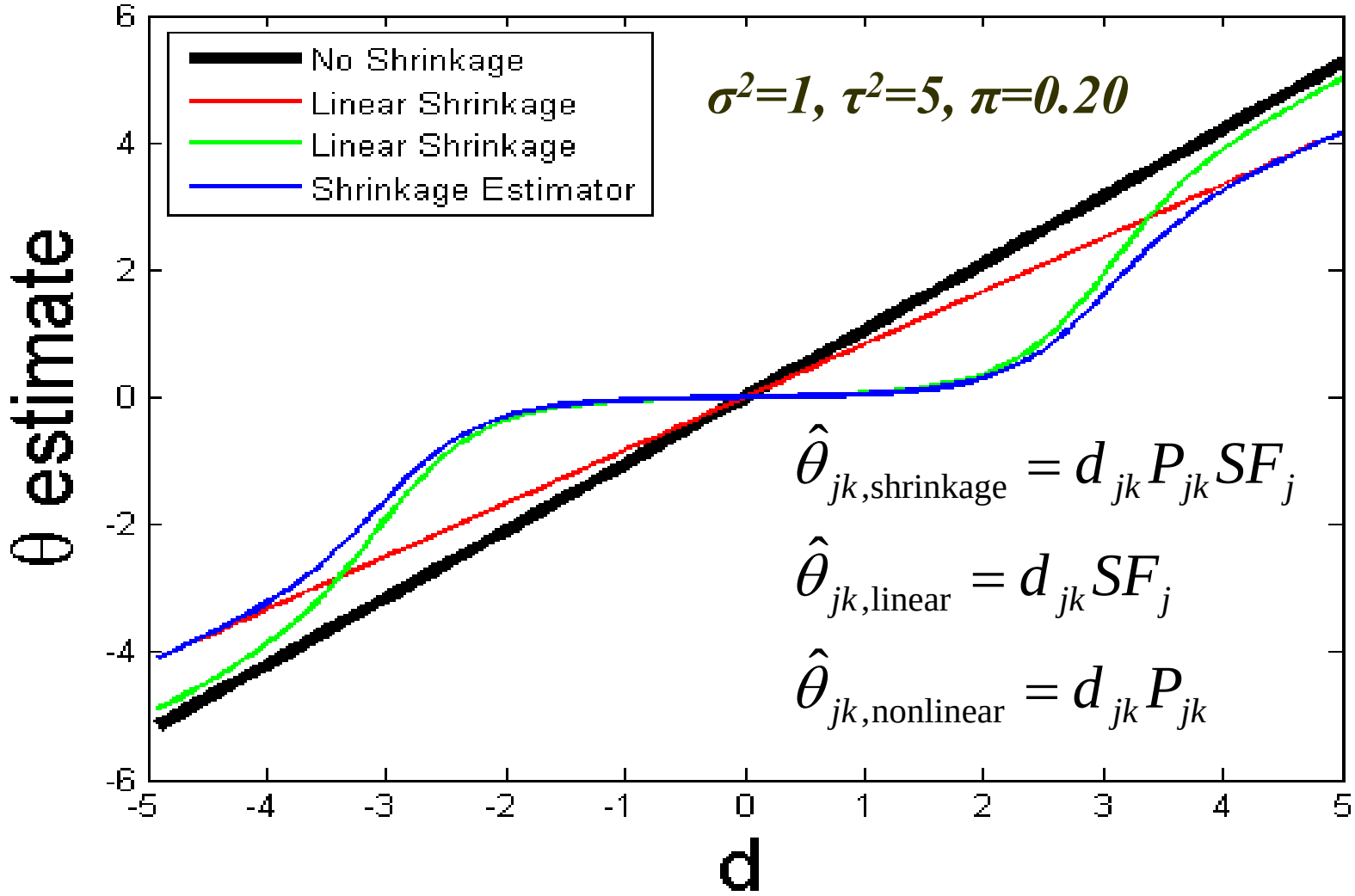


$$\hat{\theta}_{jk,\text{shrinkage}} = d_{jk} P_{jk} SF_j$$

$$\hat{\theta}_{jk,\text{linear}} = d_{jk} SF_j$$

$$\hat{\theta}_{jk,\text{nonlinear}} = d_{jk} P_{jk}$$

Wavelets: Wavelet Regression

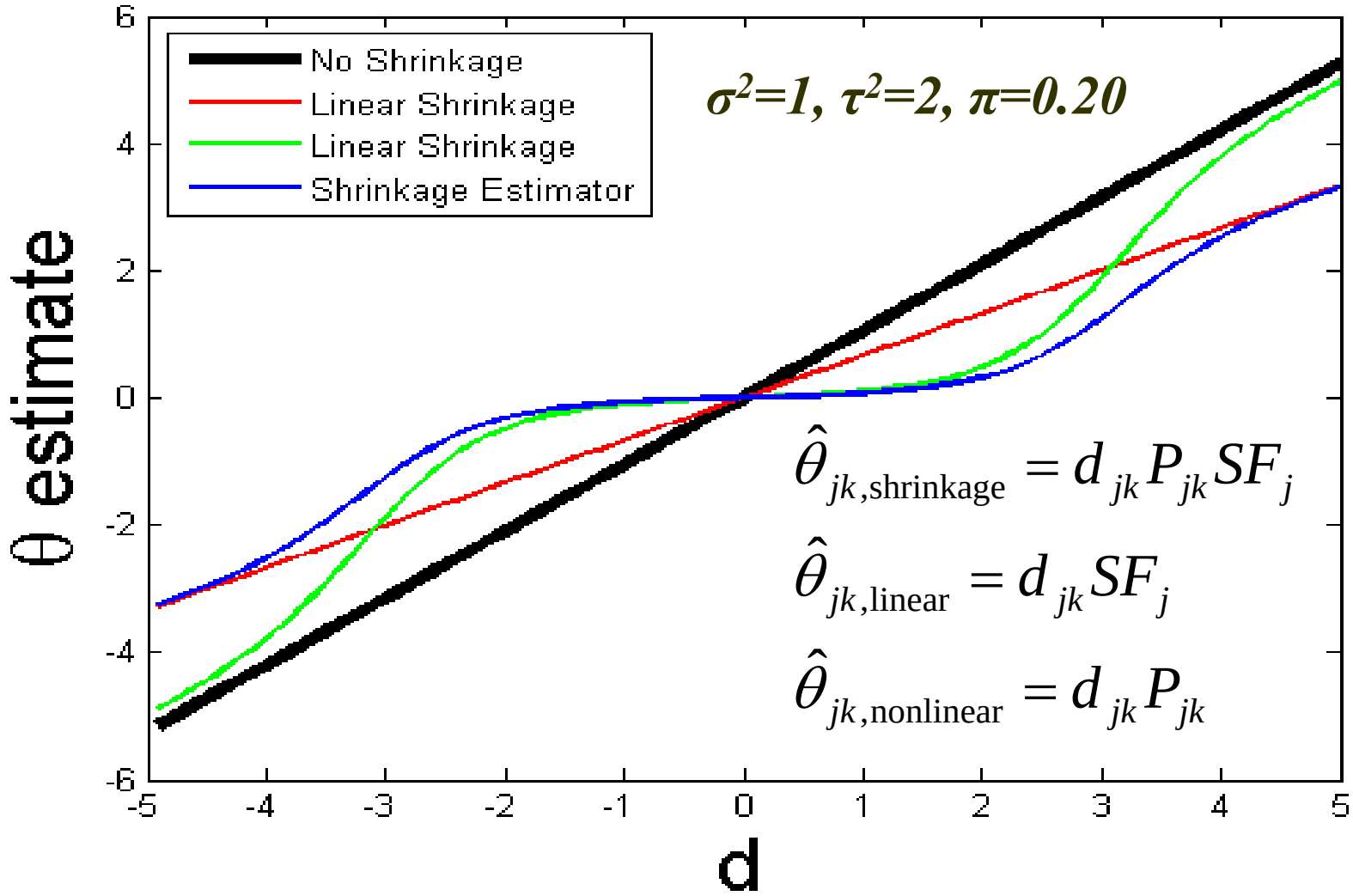


$$\hat{\theta}_{jk,\text{shrinkage}} = d_{jk} P_{jk} SF_j$$

$$\hat{\theta}_{jk,\text{linear}} = d_{jk} SF_j$$

$$\hat{\theta}_{jk,\text{nonlinear}} = d_{jk} P_{jk}$$

Wavelets: Wavelet Regression



$$\hat{\theta}_{jk,\text{shrinkage}} = d_{jk} P_{jk} SF_j$$

$$\hat{\theta}_{jk,\text{linear}} = d_{jk} SF_j$$

$$\hat{\theta}_{jk,\text{nonlinear}} = d_{jk} P_{jk}$$