BIOS 825: Basis Expansions and Regularization

Alexander McLain

September 27th, 2021

Introduction

- So far, the methods that we've used (whether regression or classification) have assumed some linear function $\boldsymbol{X}'\beta$.
- Linear function can always be viewed as first-ordered Taylor approximations on non-linear functions $f(X)$.
- In this section, we explore non-linear functions $h_m(\boldsymbol{X}) : \mathbb{R}^p \to \mathbb{R}$ for $m = 1, \ldots, M$ and use the model

$$f(\boldsymbol{X}) = \sum_{m=1}^{M} \beta_m h_m(\boldsymbol{X})$$

- What are some examples of h_m we might consider?

Basis expansions

- The $h_m(\boldsymbol{X})$ we'll consider are *basis expansions*.
- A set of vectors $\boldsymbol{B}$ is called a **basis** of a vector space $\boldsymbol{V}$ if every element in $\boldsymbol{V}$ can be written as a linear combination of elements of $\boldsymbol{B}$.
- Every continuous function in the function space can be represented as a linear combination of **basis functions**.

Basis expansions

- There are many types of **basis expansions**: *piecewise-polynomials*, *splines* and *wavelets* are a few.
- Let $\mathcal{D}$ denote the *dictionary* of methods under consideration.
- How can we choose the basis function from $\mathcal{D}$?
  1. Restriction methods
  2. Selection methods
  3. Regularization methods

Piecewise-polynomials and splines

- For the moment we'll consider one-dimensional $\boldsymbol{X}$ (we could have multiple X's but were only considering one at a time).
- Piecwise constant with 2 knots

$$h_1(X) = I(X < \xi_1) \quad h_2(X) = I(\xi_1 \leq X < \xi_2) \quad h_3(X) = I(X > \xi_2)$$

- Continuous piecewise linear version

$$h_1(X) = 1 \quad h_2(X) = X \quad h_3(X) = (X - \xi_1)_+ \quad h_4(X) = (X - \xi_2)_+$$

Piecewise Cubic Polynomials

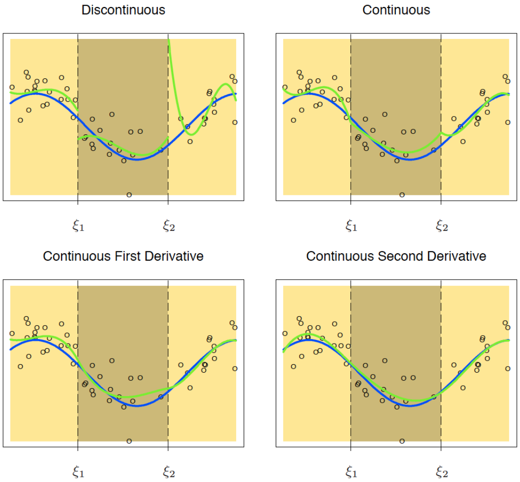

Figure: From ESL (online version).

Cubic splines

- Having a continuous second derivative leads to *cubic spline*
  - (note: continuous second derivative functions are smooth to the naked eye, going beyond second doesn't make them look more smooth)
- For our example we have

$$h_1(X) = 1 \quad h_3(X) = X^2 \quad h_5(X) = (X - \xi_1)_+^3$$

$$h_2(X) = X \quad h_4(X) = X^3 \quad h_6(X) = (X - \xi_2)_+^3$$

- The number of parameters is: (3 regions)$\times$(4 parameters per region) - (2 knots)$\times$(3 constraints per knot) $= 6$.

order-M splines

- An order-M spline (or a *piecewise-polynomial of order M*) has continuous derivatives to up to order $M-2$ (for cubic spline $M=4$).
- With knots at ξ_j for $j=1,\ldots,K$ we have

$$\begin{aligned} h_j(X) &= X^{j-1} \quad \text{for } j=1,\ldots,M \\ h_{M+\ell}(X) &= (X-\xi_\ell)_+^{M-1}, \quad \text{for } \ell=1,\ldots,K \end{aligned}$$

- These **fixed knot splines** are known as *regression splines*
- The knots can be chosen, for example, by the expression `bs(x,df=7)` which generates a basis matrix of cubic-spline functions evaluated at the N observations in `x`, with the $7-3=4$ interior knots at the appropriate percentiles of `x` (20, 40, 60 and 80th.).

Natural Cubic Splines

- Polynomials are erratic towards the boundary of the data, not to mention when results are extrapolated beyond the range of the data.
- This can be exacerbated with piecewise-polynomial splines.
- *Natural cubic splines* add the constraint that the function is linear beyond the boundary of the data.
- This constraint decreases the *degrees of freedom* of the model (i.e., we could add more interior knots) and adds stability to the model at and beyond the boundary.
  - it also may add bias, but some bias would be preferred since we don't have much data in this region.

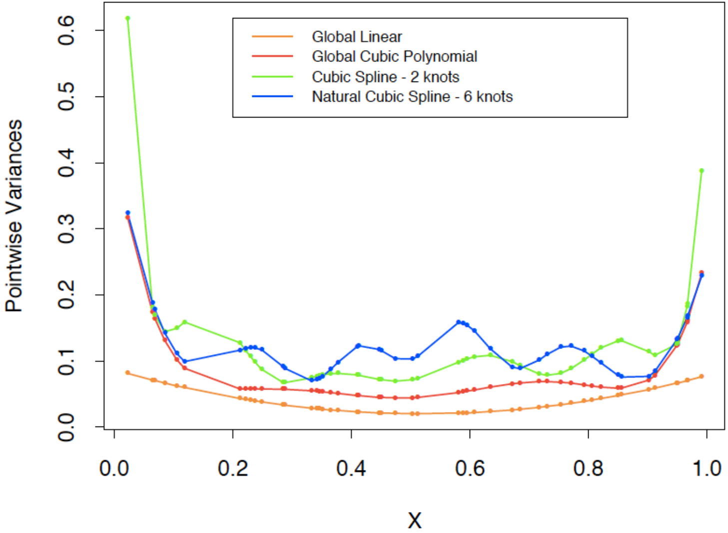

Figure: Pointwise variance of different spline methods. From ESL (online version) page 145.

Natural Cubic Splines

- Natural Cubic Splines with K knots are represented by K basis functions.
- They can be derived by reducing the basis of cubic spline by imposing the boundary constraints.
- In this case they take the form

$$N_1(X) = 1, \quad N_2(X) = X, \quad N_{k+2}(X) = d_k(X) - d_{K-1}(X),$$

where

$$d_k(X) = \frac{(X - \xi_k)^3_+ - (X - \xi_K)^3_+}{\xi_K - \xi_k}.$$

- Note that each of these basis functions has zero 2nd and 3rd derivative outside the boundary knots

Cubic B-Splines

- The previous splines we've discussed are good to learn from since they have relatively simple forms.
- In practice, (by far) the most used spline is the B-spline.
- Let $B_{i,m}(x)$ be the ith B-spline basis function of order m with knot sequence τ.
- The B-splines are defined recursively as $B_{i,1} = I(\tau_i \leq x < \tau_{i+1})$ for $i = 1, \ldots, K + 2M - 1$ and

$$B_{i,m}(x) = \frac{x - \tau_i}{\tau_{i+m-1} - \tau_i} B_{i,m-1}(x) + \frac{\tau_{i+m} - x}{\tau_{i+m} - \tau_{i+1}} B_{i+1,m-1}(x)$$

- The most common is the $m = 4$ cubic B-splines. There are also natural cubic B-splines.

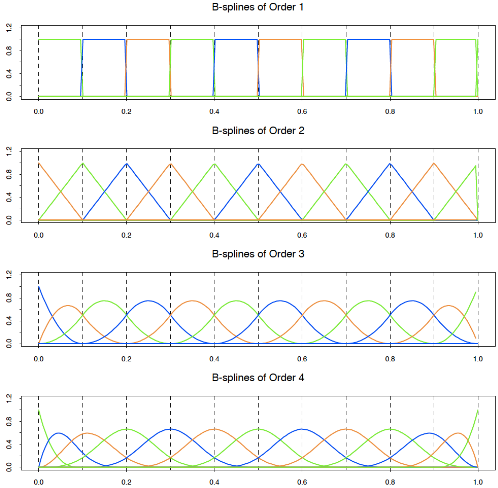

Figure: B-splines or varying order. From ESL (online version) page 188.

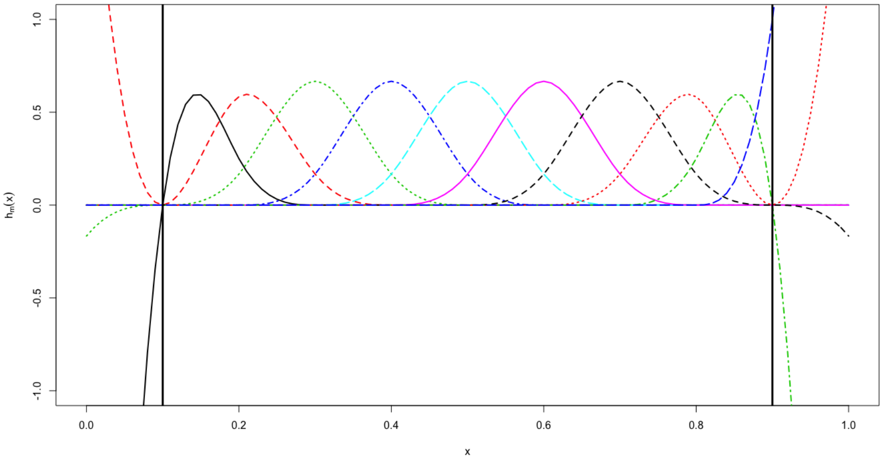

Figure: Cubic B-splines with $df = 10$ and boundary knots at 0.1 and 0.9.

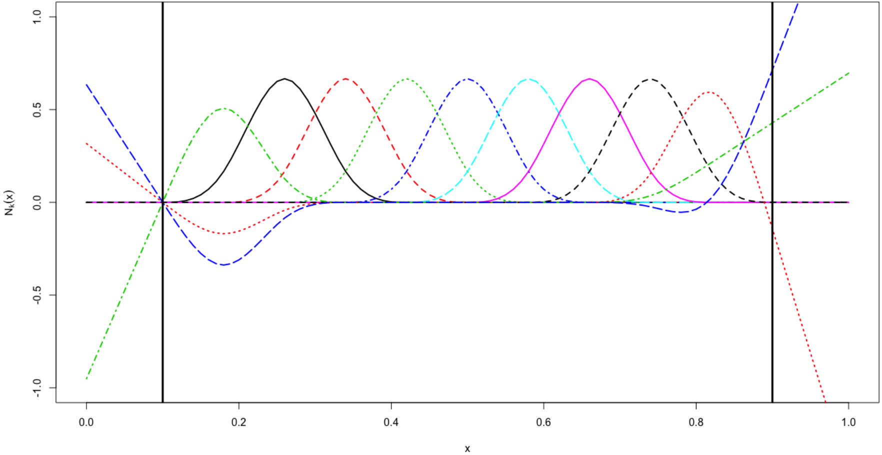

Figure: Natural cubic B-splines with $df = 10$ and boundary knots at 0.1 and 0.9.

Introduction

- The previous slides we had to choose the number and location of the interior knots of the spline function.
- To avoid the knot selection issue we can use the penalized RSS

$$RSS(f,\lambda)=\sum_{i=1}^{N}\{y_i-f(x_i)\}^2+\lambda\int\{f''(t)\}^2dt$$

  where λ is the *smothing parameter*
- Some special cases:
  - $\lambda=0$: f interpolates the data
  - $\lambda=\infty$: reverts to the LS fit wrt x with no second derivative

Smoothing Spline Solution

- Interestingly, it can be shown that $RSS(f, \lambda)$ has an explicit, finite-dimensional, unique minimizer which is a *natural cubic spline* with knots at **the unique values of the x_i for $i = 1, \ldots, N$.**
- While having (potentially) N knots might seem crazy the penalty term will shrink the spline coefficients to zero.
- We write the solution to this model as

$$f(x) = \sum_{i=1}^{N} N_j(x)\theta_j$$

where $N_j(x)$ represent the basis functions of this family of natural splines.

Smoothing Spline Solution

- We can re-write the penalized RSS as

$$RSS(f, \lambda) = (\boldsymbol{y} - \boldsymbol{N}\theta)'(\boldsymbol{y} - \boldsymbol{N}\theta) + \lambda\theta'\Omega_N\theta$$

  where $\boldsymbol{N}_{ij} = N_j(x_i)$ and $\{\Omega_N\}_{jk} = \int N_j''(t)N_k''(t)dt$.
- Note the similarities of this RSS and ridge regression. If fact, this is known as a *generalized ridge regression*
- The solution is given by

$$\hat{\theta} = (\boldsymbol{N}'\boldsymbol{N} + \lambda\Omega_N)^{-1}\boldsymbol{N}'\boldsymbol{y}$$

Smoothing matrix

- The fitted model is then $\hat{f} = \boldsymbol{N}'\hat{\theta}$ or

$$\begin{aligned} \hat{f} &= (\boldsymbol{N}'\boldsymbol{N} + \lambda\Omega_N)^{-1}\boldsymbol{N}'\boldsymbol{y} \\ &= \boldsymbol{S}_\lambda \boldsymbol{y} \end{aligned}$$

  where $\boldsymbol{S}_\lambda$ is known as the smoother matrix.
- Note that given λ this is a *linear smoother* in that the fitted values are a linear combination of the y_i.

Decomposing the smoothing matrix

- The eigen-decomposition of $\boldsymbol{S}_\lambda$ is

$$\boldsymbol{S}_\lambda = \sum_{k=1}^{N} \rho_k(\lambda)\boldsymbol{u}_k\boldsymbol{u}_k'$$

with

$$\rho_k(\lambda) = \frac{1}{1+\lambda d_k}$$

where d_k are the eigenvalues of $\boldsymbol{K}$ and $\boldsymbol{S}_\lambda = (\boldsymbol{I} + \lambda\boldsymbol{K})^{-1}$.

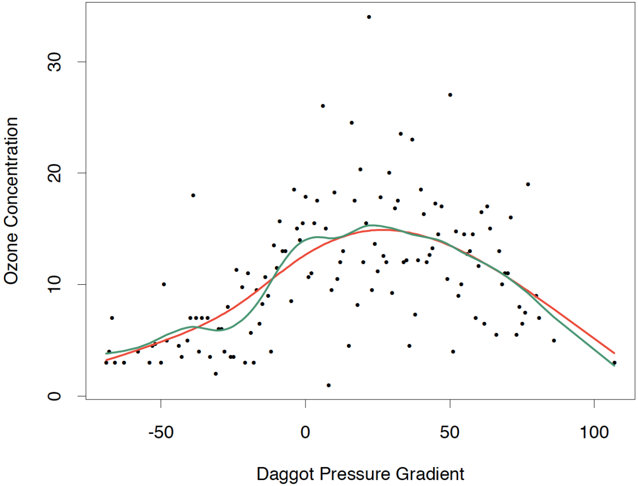

Figure: From ESL (online version) page 155.

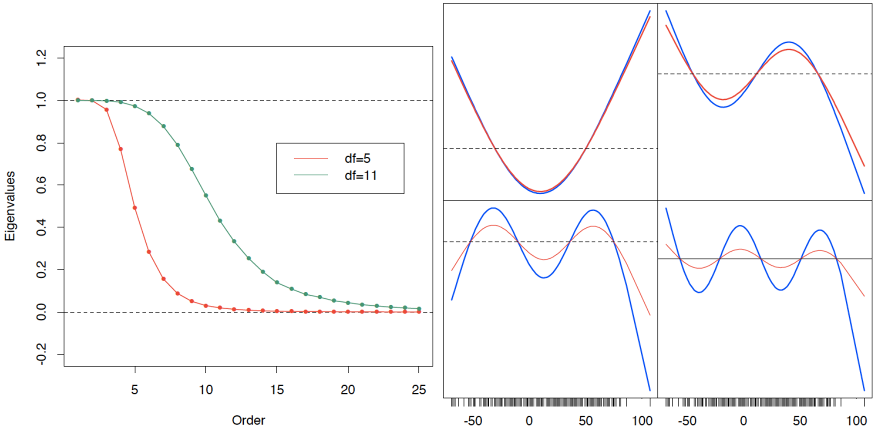

Figure: Eigenvalues on the left and the 3rd to 6th eigenvectors on the right. From ESL (online version) page 155.

Decomposing the smoothing matrix

Some facts this decomposition implies

- The first two eigenvalues are 1, meaning they have no shrinkage and $d_1 = d_2 = 0$. This is always true of the first two-dimensions, which correspond to the linear portions of x.
- The sequence of $\boldsymbol{u}_k$, ordered by decreasing $\rho_k(\lambda)$, appear to increase in complexity. Indeed, they have the zero-crossing behavior of polynomials of increasing degree.
- $\boldsymbol{S}_\lambda \boldsymbol{u}_k = \rho_k(\lambda)\boldsymbol{u}_k$, thus each of the eigenvectors are shrunk by the smoothing spline such that the higher the complexity, the more shrinkage they have.

Selecting λ via degrees of freedom

- Similar to before, we can use the smoother matrix to estimate the *effective degrees of freedom* via

$$df_\lambda = \text{trace}(\boldsymbol{S}_\lambda) = \sum_{k=1}^{N} \rho_k(\lambda)$$

- This gives us a way to specify the amount of smoothing in terms of the degrees of freedom.
- For example, we can set λ by solving $df_\lambda = 12$ for λ (df_λ will be monotone in λ). Some packages will do this for us.

Selecting λ

- Other guidelines for selecting λ are minimizing the *EPE*

$$
\begin{aligned}
EPE(\hat{f}_\lambda) &= E\{Y - \hat{f}_\lambda(X)\}^2 \\
&= \mathrm{Var}(Y) + E\left[\mathrm{Bias}^2\{\hat{f}_\lambda(X)\} + \mathrm{Var}\{\hat{f}_\lambda(X)\}\right] \\
&= \sigma^2 + \mathrm{MSE}(\hat{f}_\lambda)
\end{aligned}
$$

where since $\hat{f}_\lambda$ is linear $\mathrm{Cov}(\hat{f}_\lambda) = \boldsymbol{S}_\lambda \mathrm{Cov}(\boldsymbol{y})\boldsymbol{S}'_\lambda$ and $\mathrm{Bias}(\hat{f}_\lambda) = f - E(\boldsymbol{S}_\lambda \boldsymbol{y}) = f - \boldsymbol{S}_\lambda f$.

Introduction

- So far we have focused on one-dimensional spline models. Each of the approaches have multidimensional analogs.
- Suppose $X \in \mathbb{R}^2$ and we have univariate function $h_{1j}(X_1)$ for $j = 1, \ldots, M_1$ and $h_{2j}(X_2)$ for $j = 1, \ldots, M_2$.
- Then the $M_1 \times M_2$ dimensional tensor product basis defined by

$$g_{jk}(X) = h_{1j}(X_1)h_{2k}(X_2) \; j = 1, \ldots, M_1 \;\; k = 1, \ldots, M_2$$

gives the two-dimensional function

$$g(X) = \sum_{j=1}^{M_1} \sum_{k=1}^{M_2} \theta_{jk} g_{jk}(X).$$

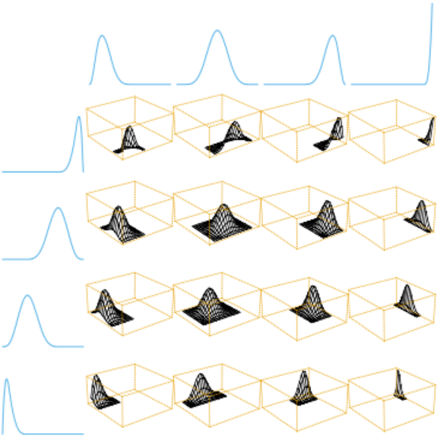

FIGURE 5.10. *A tensor product basis of B-splines, showing some selected pairs. Each two-dimensional function is the tensor product of the corresponding one dimensional marginals.*

Figure: A tensor product of basis of B-splines. From ESL (online version) page 163.

Penalization

- The above can be generalized to d dimensions, but the number of parameters grows exponentially (curse of dimensionality).
- Tensor basis functions can again be fitted with a penalized RSS

$$\sum_{i=1}^{N}\{y_i - f(x_i)\}^2 + \lambda J[f]$$

where J is a penalty functional for the f, which is no longer univariate.

Penalization

- If $d = 2$ we could use the following

$$J[f] = \int\int \left[\left(\frac{\partial f(x)}{\partial x_1^2} \right)^2 + \left(\frac{\partial f(x)}{\partial x_1 \partial x_2} \right)^2 + 2 \left(\frac{\partial f(x)}{\partial x_2^2} \right)^2 \right] dx_1 dx_2$$

- Using this penalty leads to a smooth two-dimensional surface, known as a *thin-plate spline*.
- Some special cases:
  - $\lambda = 0$: f approaches an interpolation the data
  - $\lambda = \infty$: reverts to the LS plane wrt x_1, x_2 with no second derivative
  - for intermediate values of λ, the solution can be represented as a linear expansion of basis functions

Penalization Solution

- For $0 < \lambda < \infty$, the solution has the form

$$f(x) = \beta_0 + \boldsymbol{\beta}'\boldsymbol{x} + \sum_{j=1}^{N} \alpha_j h_j(\boldsymbol{x})$$

  where $h_j(\boldsymbol{x}) = ||\boldsymbol{x} - \boldsymbol{x}_j||^2 \log ||\boldsymbol{x} - \boldsymbol{x}_j||$.
- These h_j are known as *radial basis functions*
- The $\boldsymbol{\alpha}$ vector can be found using the **generalized ridge regression** solution discussed in the previous notes.

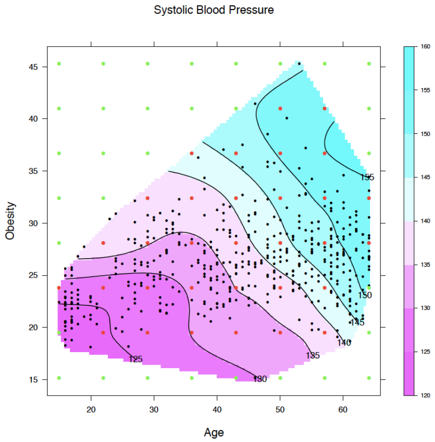

Figure: Example of a contour plot from a fitted tensor spline model. From ESL (online version) page 166.

General thin-plate splines

- The above solution can be generalized to d dimensions (note J would have to be generalized).
- In the two dimensional case the computational complexity is $O(N^3)$.
- Even in the penalized case, we can usually get away with fewer then N knots (reducing the computational complexity).
- Example

Introduction and motivation

- The previous used a few spline bases function to represent smooth functions.
- Wavelets can be used to represent a mostly flat function with a few isolated bumps.
- Wavelet basis can be thought of as bumpy basis functions.
- They differ from previous basis in that they have both *time and frequency* localization

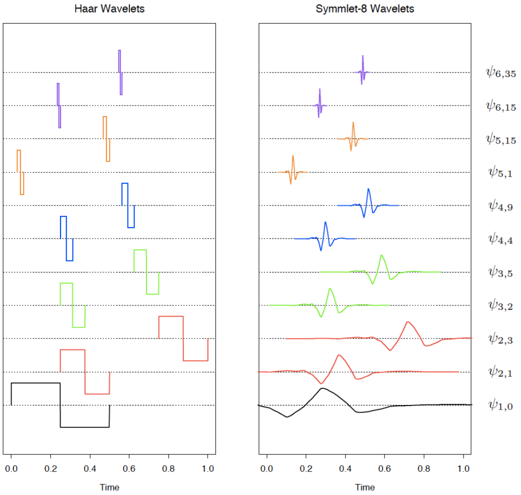

Figure: Harr and Symmlet-8 wavelets. From ESL (online version) page 175.

Details

- We'll not discuss the mathematical formulation of the wavelet basis (see section 5.9.1 in ESL if your interested).
- Some things to note:
  - At each scale the wavelets are packed in side-by-side to completely fill the x-axis (not all are shown in the figure above).
  - The spaces or orthogonal, and all the basis functions are orthonormal.
  - There are 2^j elements at level j,
  - If there are $N = 2^J$ unique x values, then level J is the maximum level we can use.

Adaptive Wavelet Filtering

- For a response $\boldsymbol{y}$ let $\boldsymbol{W}$ be the $N \times N$ orthonormal wavelet basis matrix evaluated at the N uniformly spaced observations ($N = 2^J$).
- The *wavelet transformation* of $\boldsymbol{y}$ is $\boldsymbol{y}^* = \boldsymbol{W}'\boldsymbol{y}$ and is the full least squares regression coefficient.
- A popular method for adaptive wavelet fitting is known as SURE shrinkage (Stein Unbiased Risk Estimation, Donoho and Johnstone (1994)).
- This starts with the lasso criteria

$$\min_{\theta} ||\boldsymbol{y} - \boldsymbol{W}\theta||_2^2 + 2\lambda||\theta||_1,$$

then since $\boldsymbol{W}$ is orthonormal the solution is

$$\hat{\theta}_j = \text{sign}(y_j^*)(|y_j^*| - \lambda)_+$$

Adaptive Wavelet Filtering

- A simple choice for λ is $\lambda = \sigma\sqrt{2\log N}$ where σ is an estimate of the standard deviation of the residuals.
    - Note that $\sigma\sqrt{2\log N}$ is the maximum absolute value of N Gaussian variables.
- The interesting thing about wavelets is that coefficients represent characteristics of the signal **localized in time** (the basis functions at each level are translations of each other) and **localized in frequency.**
- Modern image compression is often performed using two-dimensional wavelet representations.

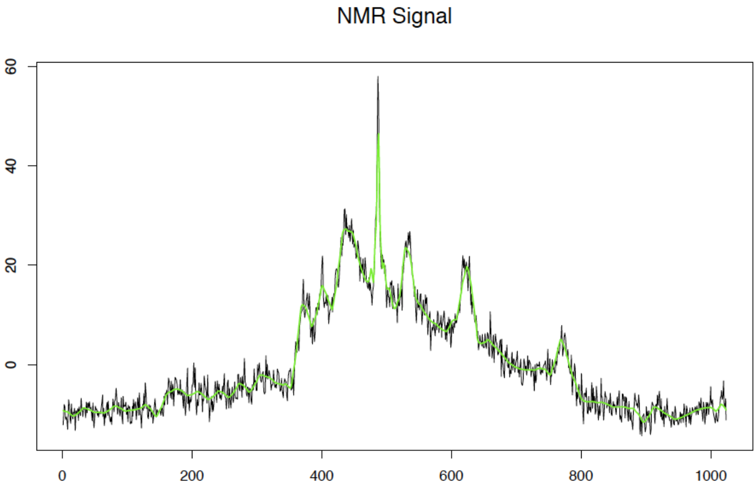

Figure: Truncated and untruncated coefficients of Symmlet-8 wavelets used to de-noise on a nuclear magnetic resonance signal. From ESL (online version) page 175.

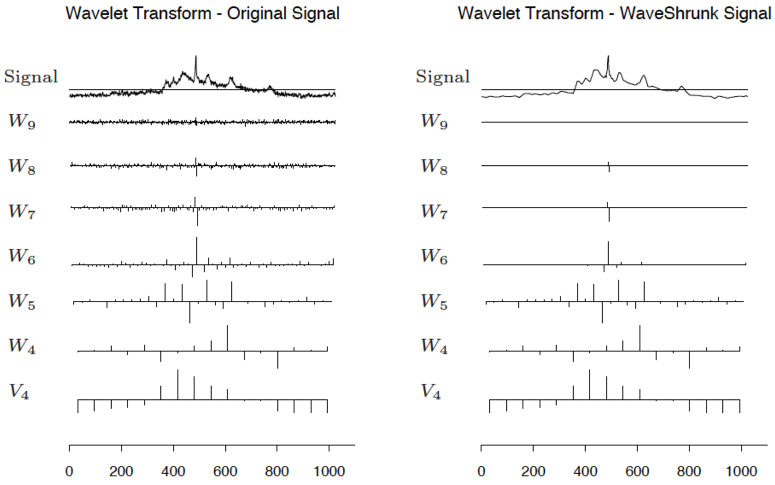

Figure: Example of using Symmlet-8 wavelets to de-noise on a nuclear magnetic resonance signal. From ESL (online version) page 175.

Summary: Pros of Basis Expansion

- **Capture Non-linear Relationships:** Basis expansion can allow linear models to capture non-linear relationships.
- **Flexibility:** They offer the flexibility to fit a wide range of data shapes, from simple linear trends to more intricate patterns.
- **Interpretability:** Despite the transformation, some basis expansions (like polynomial expansions) can remain interpretable.
- **Computational Simplicity:** For some methods (e.g., polynomial regression), the basis expansion does not significantly increase the computational cost.
- **Enhanced Performance:** By capturing non-linearities or interactions, basis expansion can lead to models with improved predictive performance compared to linear models without expansion.

Summary: Cons of Basis Expansion

- **Overfitting**.
- **Loss of Interpretability:** While some expansions are interpretable, others (like certain spline functions or radial basis functions) can make the model more challenging to understand.
- **Multicollinearity:** Especially in polynomial regression, higher-order terms can be highly correlated with lower-order terms.
- **Feature Selection:** Deciding which features to expand and to what degree (e.g., deciding the degree for polynomial expansion) can be challenging.
- **Boundary Effects:** Some basis expansions, like splines, can behave unpredictably near the boundaries of the data.

Summary

- Basis expansion can enhance the flexibility and performance of models by allowing them to capture non-linear relationships, and
- It's essential to be aware of the potential pitfalls and to use techniques like cross-validation or penalization to avoid overfitting.

Up next:

- Here, we talked about expanding the predictor space.
- Next, we'll talk about reducing the predictor space.