ECE7252
Statistical Learning for Signal Processing

Lectures 19-20: Basis Expansion

Chin-Hui Lee

School of Electrical and Computer Engineering

Georgia Institute of Technology

Atlanta, GA 30332, USA

chl@ece.gatech.edu

Overview

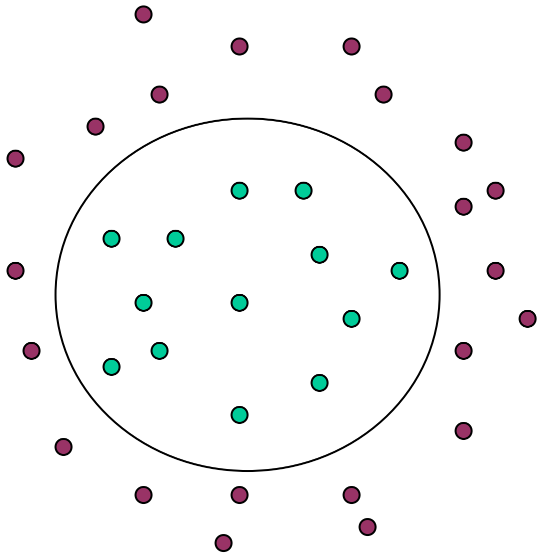
- Basis expansion
- Splines
- (Natural) cubic splines
- Smoothing splines
- Nonparametric logistic regression
- Multidimensional splines
- Wavelets

Moving Beyond Linearity

Let’s take a look at the following classification example:

Class purple: $x_1^2 + x_2^2 < r^2$

Class cyan: $x_1^2 + x_2^2 > r^2$



Observe it is not possible to separate the two classes of input (x_1, x_2) by a linear boundary; however they are separable by the circle $x_1^2 + x_2^2 = r^2$

Moving Beyond Linearity: Not So Fast!

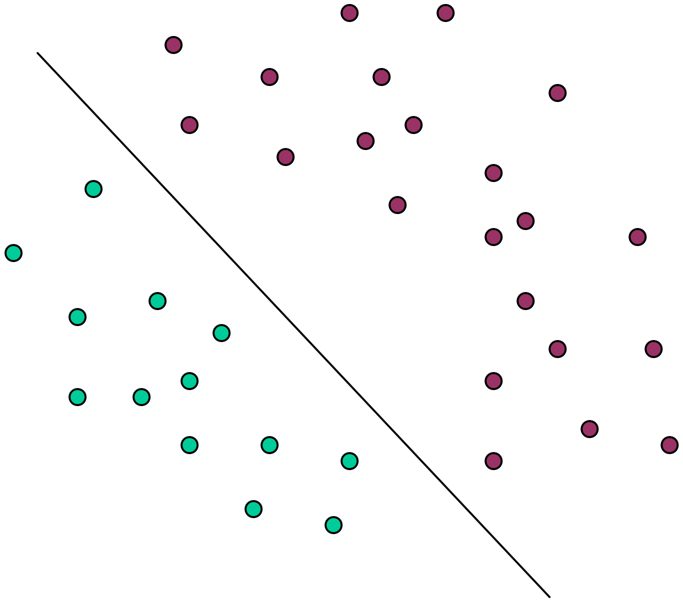
A simple trick for the previous example still allows us to stay within the linear model. Let us make this transformation:

$$X_1 = x_1^2$$

$$X_2 = x_2^2$$

Class purple: $X_1 + X_2 < r^2$

Class cyan: $X_1 + X_2 > r^2$



In the transformed input space (X_1, X_2) the

Class boundary is linear: $X_1 + X_2 = r^2$

Linear Basis Expansions

Original input space: $X=(X_1, \ldots, X_p)$

Transformed input space: $h_1(X), h_2(X), \ldots, h_M(X)$

Linear model in the transformed input space: $f(X)=\sum_{m=1}^{M}\beta_m h_m(X)$

This model is known as a linear basis expansion

An example where the original space is transformed into a polynomial space:

$$h_1(X)=X_1^2, \quad h_2(X)=\sqrt{2}X_1X_2, \quad h_3(X)=X_2^2$$

h(x)'s are called basis functions

Example Basis Functions

- Original space: $h_m(X) = X_m$, m=1, 2, …, p
- Polynomial expansions: $h_m(X) = (X_j)^2$ or $X_j X_k$
- Non-linear transformation: $h_m(X) = \log(X_j)$, $\mathrm{sqrt}(X_j)$, etc.
- Indicator region: $h_m(X) = I(L_m <= X_k <= U_m)$

Linear Basis Expansion

Linear regression

x_1	x_2	x_3	y
1	-3	6	12
…	…	…	…

True model: $y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$

Question: How to find $\hat{f}$?

Answer: Apply linear regression to obtain: $\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3$

Linear Basis Expansion (Cont.)

Nonlinear model

x_1	x_2	x_3	y
1	-3	-1	12
…	…	…	…

True model:

$$y = \beta_1 x_1 x_2 + \beta_2 x_2 e^{x_3} + \beta_3 \sin x_3 + \beta_4 x_1^2 + \varepsilon$$

Question: How to find $\hat{f}$?

Answer: A) Introduce new variables

$$u_1 = x_1 x_2, u_2 = x_2 e^{x_3},$$

$$u_3 = \sin x_3, u_4 = x_1^2$$

Linear Basis Expansion (Cont.)

Nonlinear model

B) Transform the data set

u_1	u_2	u_3	u_4	y
-3	-1.1	-0.84	1	12
…	…	…		…

True model:

$$y = \beta_1 u_1 + \beta_2 u_2 + \beta_3 u_3 + \beta_4 u_4 + \varepsilon$$

C) Apply linear regression and obtain: $\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4$

Linear Basis Expansion (Cont.)

Conclusion on linear basis expansion:

- Now we know how to fit any model of the type

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

- In other words, we model a *linear basis expansion* in *X*
- *Example:* If the model is known to be nonlinear, but the exact form is unknown, try to introduce interaction (problem: no. of variables grows exponentially)

$$f(X) = \beta_1 X_1 + \ldots + \beta_p X_p + \beta_{11} X_1^2 + \beta_{12} X_1 X_2 + \ldots$$

Piecewise Polynomials

Assume X is one-dimesional

- **Def.** Assume the domain of $X=[a,b]$ is split into intervals $[a, \xi_1], [\xi_1, \xi_2], \ldots, [\xi_n, b]$. Then $f(X)$ is piecewise polynomial if $f(X)$ is represented by separate polynomial in each interval.
- Note The points $\xi_1, \ldots, \xi_n$ are called *knots*

Piecewise Constant Model

The basis functions are:

$$h_1(X) = I(X < \xi_1), \quad h_2(X) = I(\xi_1 \le X < \xi_2), \quad h_3(X) = I(\xi_2 \le X)$$

Note that the input *X* is 1D

The linear basis expansion model:

$$f(X) = \beta_1 h_1(X) + \beta_2 h_2(X) + \beta_3 h_3(X)$$

The estimated coefficients are respective mean of data within mth region: $\hat{\beta}_m = \bar{Y}_m$

Piecewise Linear Model

The basis functions are:

$$h_1(X) = I(X < \xi_1)$$
$$h_2(X) = I(\xi_1 \leq X < \xi_2)$$
$$h_3(X) = I(\xi_2 \leq X)$$
$$h_4(X) = I(X < \xi_1)X$$
$$h_5(X) = I(\xi_1 \leq X < \xi_2)X$$
$$h_6(X) = I(\xi_2 \leq X)X$$

The linear basis expansion model:

$$f(X) = \beta_1 h_1(X) + \beta_2 h_2(X) + \beta_3 h_3(X) + \beta_4 h_4(X) + \beta_5 h_5(X) + \beta_6 h_6(X)$$

The coefficients in this case can be estimated by fitting straight lines in the respective regions – think of fitting three straight lines in the three regions – there is no interdependency among these three lines

Continuous Piecewise Linear Model

The basis functions are:

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

where $$(X - \xi_1)_+ = \begin{cases} X - \xi_1, \text{ if } X > \xi_1 \\ 0, \qquad \text{otherwise} \end{cases}$$

This model imposes continuity in the fits between the regions

The same model can be obtained if we impose two continuity constraints at the knot points along with the six basis functions on page 8 of this lecture:

$$f(\xi_1^-) = f(\xi_1^+), \text{ i.e., } \beta_1 + \xi_1\beta_4 = \beta_2 + \xi_1\beta_5$$
$$f(\xi_2^-) = f(\xi_2^+), \text{ i.e., } \beta_2 + \xi_2\beta_5 = \beta_3 + \xi_2\beta_6$$

So that the effective number of parameters is four = six – number of constraints

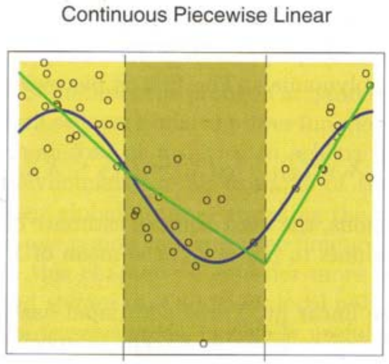
Piecewise Polynomials (Cont.)

- *Example.* Continuous piecewise linear function
- Alternative A. Introduce linear functions on each interval + restrictions.

$$\begin{cases} y_1 = \alpha_1 x + \beta_1 \\ y_2 = \alpha_2 x + \beta_2 \\ y_3 = \alpha_3 x + \beta_3 \end{cases}$$

$$\begin{cases} y_1(\xi_1) = y_2(\xi_1) \\ y_2(\xi_2) = y_3(\xi_2) \end{cases}$$

- (4 free param.)

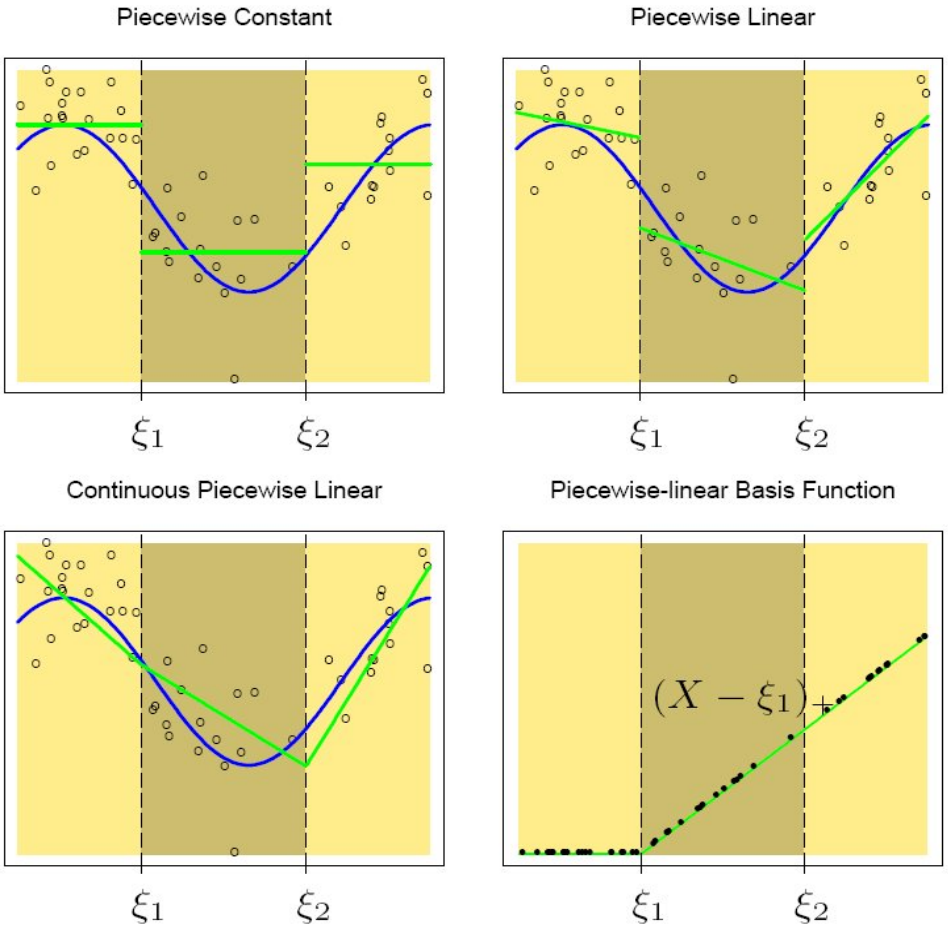

Continuous Piecewise Linear

- Alternative B. Use basis incorporating constraints(4 free param)

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

- Theorem. The given formulations are equivalent

Piecewise Models

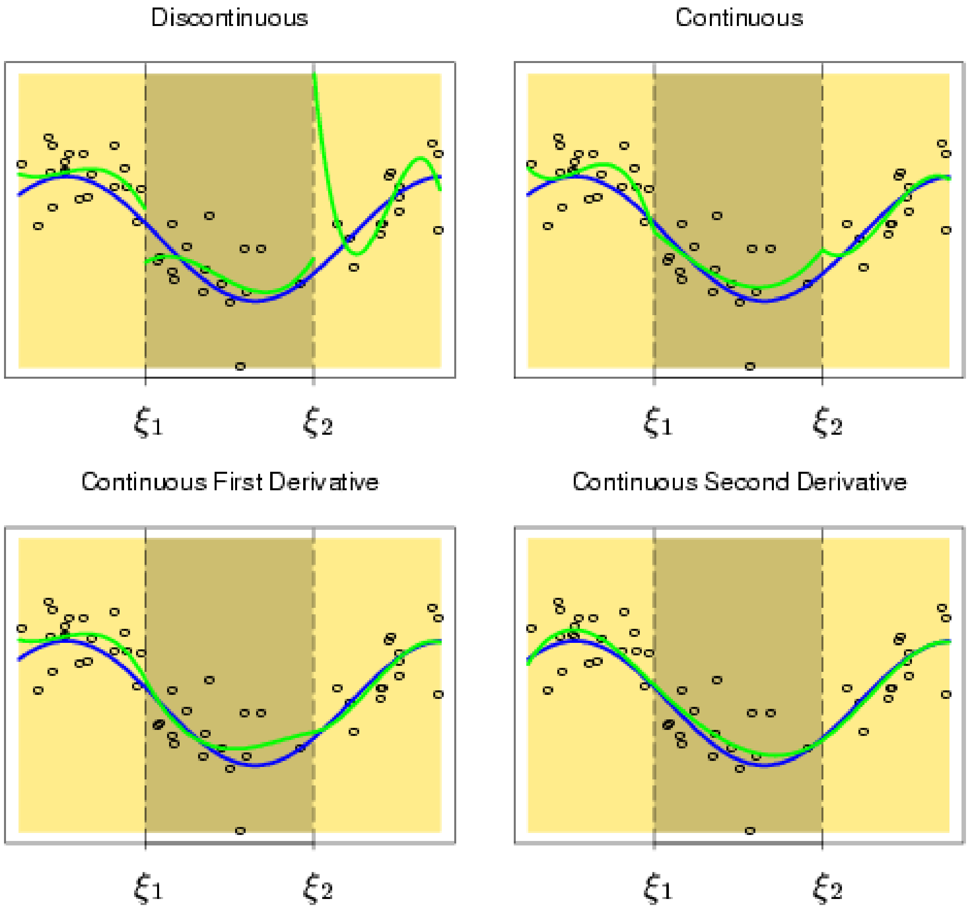


Piecewise Cubic Models

- Why stop at piecewise linear models?

- Increase smoothness by higher order polynomials

- Bottom right panel: cubic spline



Splines

- **Def:** A piecewise polynomial is called *order-M spline* if it has continuous derivatives up to order *M-1* at the knots
- Alternative: An order-*M* spline is a function which can be represented by basis functions (*K* = # of knots)

$$h_j(X) = X^{j-1},\ j = 1\ldots M$$

$$h_{M+l}(X) = (X - \xi_l)_+^{M-1},\ l = 1\ldots K$$

- Theorem. The definitions above are equivalent
- **Def:** Order-4 spline is called *cubic spline (look at basis and compare the number of free parameters)*
- Note. Cubic splines: knot-discontinuity is not visible

Cubic Spline

- Fit cubic polynomial in each region: 3 types of constraints:
  1. Zero order continuity (i.e., continuity of the curve) at knot points
  2. First order continuity (i.e., has first derivative) at knot points
  3. Second order continuity (i.e., has second derivative) at knot points
- The model fits a smooth curve to the data

Basis functions:

$$h_1(X) = 1$$

$$h_2(X) = X$$

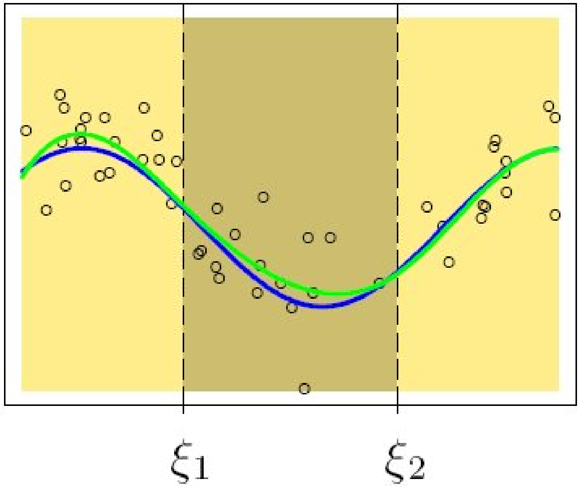
$$h_3(X) = X^2$$

$$h_4(X) = X^3$$

$$h_5(X) = (X - \xi_1)_+^3$$

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

Continuous Second Derivative



Cubic Spline: Number of Parameters

A cubic spline can be represented as piecewise cubic polynomials in each region with zero, first, and second order continuities at knot points

$$f(X) = \begin{cases} a_1 X^3 + b_1 X^2 + c_1 X + d_1, & X \le \xi_1 \\ a_2 X^3 + b_2 X^2 + c_2 X + d_2, & \xi_1 \le X \le \xi_2 \\ a_3 X^3 + b_3 X^2 + c_3 X + d_3, & \xi_2 \le X \end{cases}$$

Cubic Spline… (Cont.)

- Constraints for the first knot point:

1. Continuity: $(a_1 - a_2)\xi_1^3 + (b_1 - b_2)\xi_1^2 + (c_1 - c_2)\xi_1 + (d_2 - d_1) = 0$
2. 1st Derivative exists: $3(a_1 - a_2)\xi_1^2 + 2(b_1 - b_2)\xi_1 + (c_1 - c_2) = 0$
3. 2nd Derivative exists: $6(a_1 - a_2)\xi_1 + 2(b_1 - b_2) = 0$

- There are three more equations for the second knot points

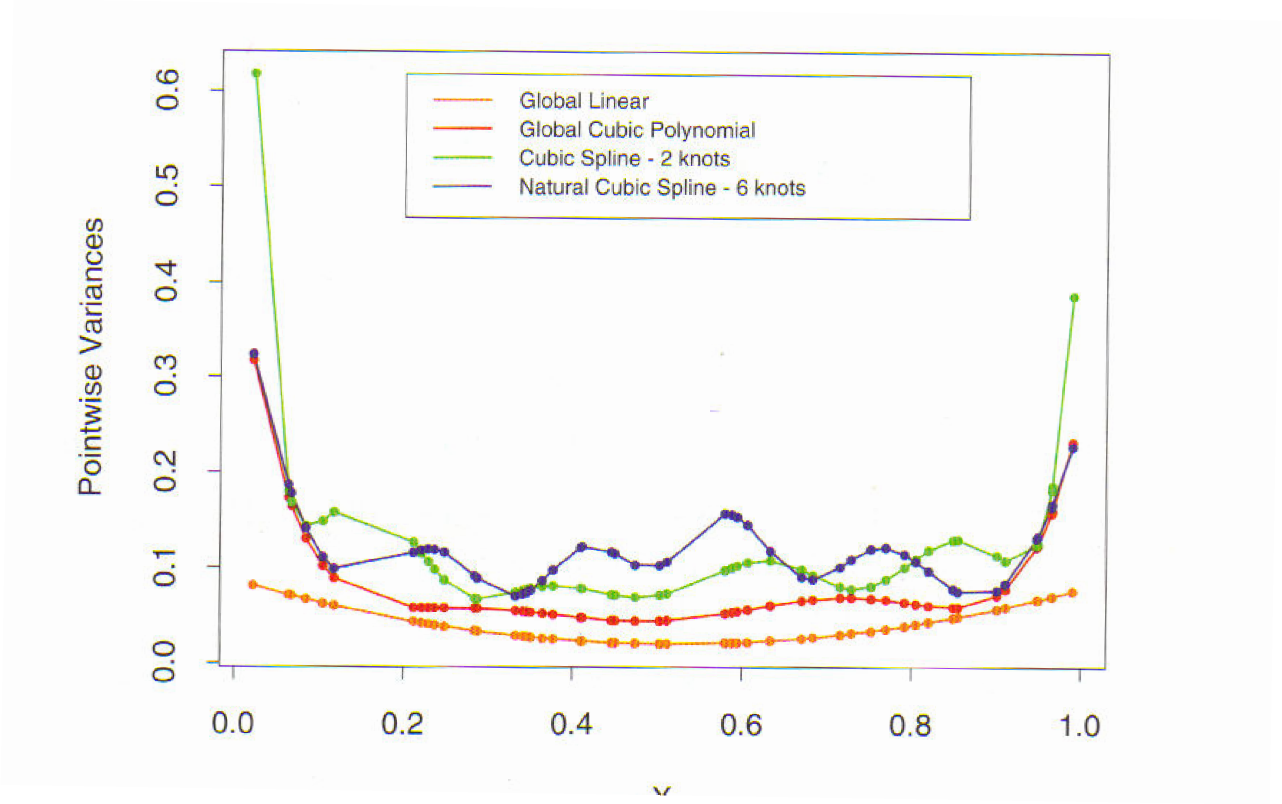
Essentially this shows that effectively there are 6 parameters:
6 = (3 regions)(4 parameters per region) – (2 knots)(3 constraints per knot)

Cubic Spline… (Cont.)

- There is seldom any need to go beyond (i.e., toward higher order) a cubic spline curves
- One needs to select the number of knot points and their placement
- Knot placement can be done by the observation data fixing the number of basis functions
- Note that increasing number of basis functions decreases the square bias and increases the variance (why?)
- Variances near the two boundary knots are high (why?)

Problem with Splines

Problem with spline fitting – boundary effects



Natural Cubic Splines

- In order to reduce the variance of cubic spline near the boundary knots, natural cubic spline places some rigidity in the model – the function is linear beyond the two boundary knots (2nd and 3rd derivatives are zero)
- This frees up some degrees of freedom in the cubic spline (i.e., it is not as wild near the boundaries now)
- There are *K* basis functions for *K* knots:

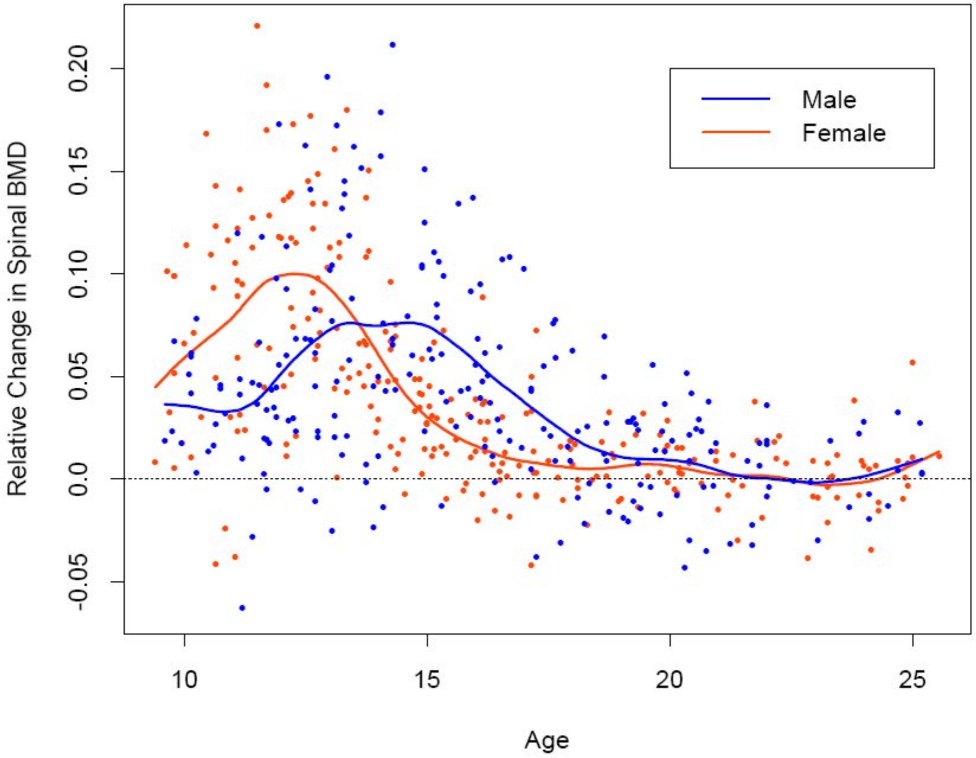
$$N_1(X) = 1$$

$$N_2(X) = X$$

$$N_{k+2}(X) = d_k(x) - d_{K-1}(X)$$

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

Example of Natural Spline Fitting



Bone mineral density (BMD) for males and females versus ages. The fits reinforce the evidence that the growth spurt for females precedes that for males by about 2 years. λ=0.00022 (chosen by cross validation)

Computing Spline

- Points to ponder
  - Number of knots
  - Placement of the knots

- There are three techniques where knots are automatically decided
  - Least Squares fitting
  - Smoothing splines
  - Logistic regression

Smoothing Spline

- Do not even bother about the selection of knots – choose all training (unique) points as knots – maximum possible knots
- However, this choice make the spline over-parameterized: variance will increase
- To shrink the variance some coefficients are set to zero by a penalty term (we have seen this before in ridge regression): intuitively this means you are making spline model a bit rigid
- One way to achieve this rigidity is to penalize the second derivative – the spline will not move wildly now!

Penalty functional: $J(f) = \int f''(t)^2 \, dt$

Smoothing Spline… (Cont.)

- Given *N* points in the training data: $\{(x_i, y_i)\}_{i=1}^{N}$

We need to find the function *f* among all functions with two continuous derivatives that minimizes the 'penalized' RSS

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

λ is the smoothing parameter. The first term measures the closeness of fit or data fidelity, while the second penalizes curvature in the function (regularization)

$\lambda \rightarrow 0$ *f(x)* can be any function that interpolates the data

$\lambda \rightarrow \infty$ least squares fit

It can be shown that $RSS(f,\lambda)$ minimized by a natural cubic spline!

Smoothing Spline… (Cont.)

The minimizer, a natural cubic spline, has knots at the unique x_i, $i=1\ldots N$. So we can write the minimizer as:

$$f(x) = \sum_{i=1}^{N} N_i(x)\theta_i$$

where $N_i(X)$ are basis functions. Note that this is a linear basis expansion. So we can apply least square techniques to find out the parameters θ_i's

Ex. $$RSS(f,\lambda) = (y - \mathbf{N}\theta)^T (y - \mathbf{N}\theta)^T + \lambda\theta^T \Omega_N \theta$$

where $\{\mathbf{N}\}_{ij} = N_j(x_i)$ and $\{\Omega_N\}_{ij} = \int N_i''(t) N_j''(t)\,dt$

Smoothing Splines (Cont.)

Solution to (SS):

- Theorem: Function *f* is a natural cubic spline with knots at unique values of x_i (NOTE: *N* knots!)

$$f(x) = \sum_{j=1}^{N} N_j(x)\theta_j = \eta(x)^T \Theta$$

$$RSS(\Theta, \lambda) = (\mathbf{y} - \mathbf{N}\Theta)^T (\mathbf{y} - \mathbf{N}\Theta) + \lambda \Theta^T \Omega_N \Theta$$

$$\{\mathbf{N}\}_{ij} = N_j(x_i) \quad \{\Omega_N\}_{ij} = \int N_i''(t) N_j''(t) dt$$

$$\hat{\Theta} = \left(\mathbf{N}^T \mathbf{N} + \lambda \Omega_N\right)^{-1} \mathbf{N}^T \mathbf{y}$$

Smoothing Splines (Cont.)

Properties of smoothing splines

- Fitted function: $\hat{f} = \mathbf{N}\left(\mathbf{N}^T\mathbf{N} + \lambda\Omega_N\right)^{-1}\mathbf{N}^T\mathbf{y} = \mathbf{S}_\lambda\mathbf{y}$
- If we take a few basis functions *M*<<*N* and compute $\mathbf{B}_\xi$ at *ξ*

$$\hat{f} = \mathbf{B}_\xi\left(\mathbf{B}_\xi{}^T\mathbf{B}_\xi\right)^{-1}\mathbf{B}_\xi{}^T\mathbf{y} = \mathbf{H}_\xi\mathbf{y}$$

- $\mathbf{H}_\xi$ is a linear projection operator
- Both symmetric, positive definite, different ranks
- Shrinking nature of $\mathbf{S}_\lambda$:

$$\mathbf{H}_\xi\mathbf{H}_\xi = \mathbf{H}_\xi, \quad \mathbf{S}_\lambda\mathbf{S}_\lambda \le \mathbf{S}_\lambda$$

Smoothing Splines (Cont.)

Properties of smoothing splines

- $M = trace(\mathbf{H}_\xi)$ defines the dimension of the basis functions (degrees of freedom)
- By analogy, effective degrees of freedom for the smoother matrix $\mathbf{S}_\lambda$ is

$$df_\lambda = trace(\mathbf{S}_\lambda)$$

-> A way to select λ

Smoothing Splines (Cont.)

Eigenvalue decomposition

- Rewriting in Reinsch form (show): $\mathbf{S}_\lambda = \left(\mathbf{I} + \lambda\mathbf{K}\right)^{-1}$

K is a *penalty matrix*

- The eigen-decomposition is (show):

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

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

- Note: d_k and $\mathbf{u}_k$ are respective eigenvalues and eigenvectors of **K**

Smoothing Splines (Cont.)

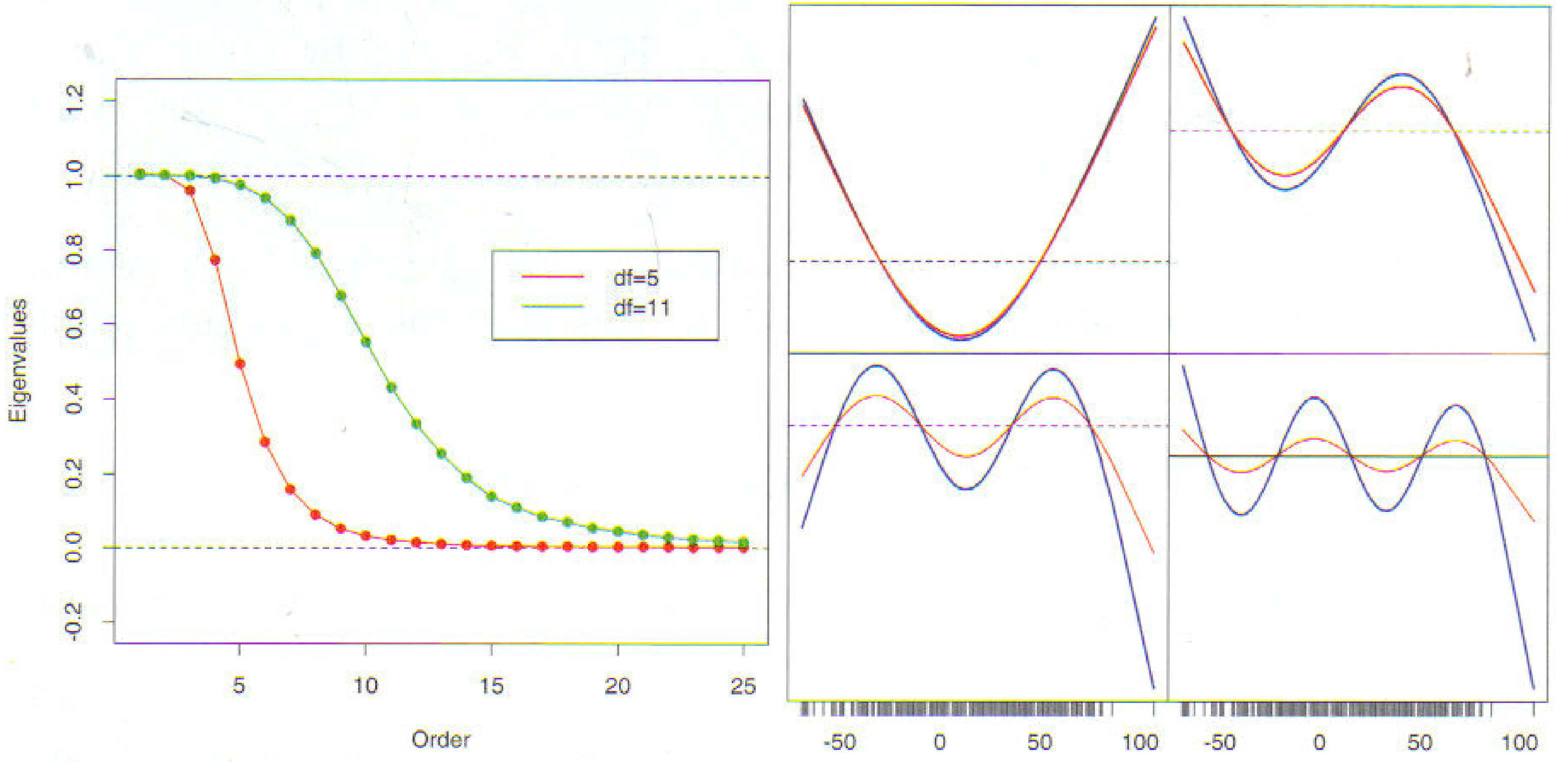
Eigenvalue decomposition – conclusions

$$\mathbf{S}_{\lambda}\mathbf{y} = \sum_{k=1}^{N} \mathbf{u}_k \rho_k(\lambda) \langle \mathbf{u}_k^T, \mathbf{y} \rangle$$

- Smoothing spline decomposes vector **y** with respect to basis of eigenvectors and shrinks respective contributions
- The eigenvectors ordered by *ρ* increase in complexity. The higher the complexity, the more the respective contribution is shrunk

Smoothing Splines (Cont.)

Eigenvalue decomposition – conclusions

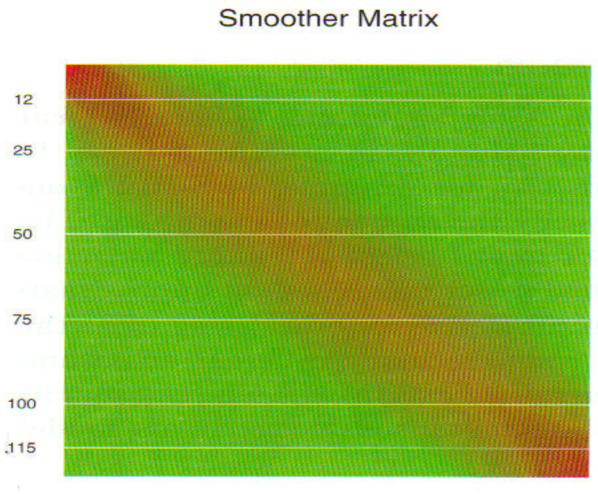


Smoothing Splines (Cont.)

Eigenvalue decomposition – conclusions

- Eigenvalues are reverse functions of *λ*. The higher *λ*, the higher penalization
- Smoother matrix is has banded nature -> local fitting method

$$df_{\lambda} = trace(\mathbf{S}_{\lambda}) = \sum_{k=1}^{N} \frac{1}{1+\lambda d_k}$$



Smoothing Splines (Cont.)

- **How does one fit splines in practice?**
- Reinsch form: $\mathbf{S}_\lambda = (\mathbf{I} + \lambda\mathbf{K})^{-1}$
- Theorem. If ***f*** is natural cubic spline with values at knots ***f*** and second derivative γ at knots then

$$Q^T\mathbf{f} = R\gamma$$

where Q & R are band matrices, dependent on ξ only

- Theorem. $K = QR^{-1}Q^T$

Smoothing Splines (Cont.)

Reinsch algorithm (show…)

- Evaluate $Q^T\mathbf{y}$
- Compute $R+\lambda Q^TQ$ and find Cholesky decomposition (in linear time!)
- Solve matrix equation (in linear time!)
- Obtain $\mathbf{f}=\mathbf{y}-\lambda Q\gamma$

Selection of Smoothing Parameters

Fixing the degrees of freedom

$$df_\lambda = trace(\mathbf{S}_\lambda) = \sum_{k=1}^{N} \frac{1}{1+\lambda d_k}$$

- If we fix df_λ then we can find λ by solving the equation numerically
- It is not difficult to solve since the function is monotonic
- One could try two different df_λ and choose one based on F-tests, residual plots etc.

Nonparametric Logistic Regression

Logistic regression model $$\log \frac{\Pr(Y=1 \mid X=x)}{\Pr(Y=0 \mid X=x)} = f(X)$$

- Note: *X* is one-dimensional but "what is *f(x)*"?
- Linear -> ordinary logistic regression (Chapter 4)
- Enough smoothness -> nonparametric logistic regression (splines+others)
- Other choices are possible

Nonparametric Logistic Regression (Cont.)

Problem formulation:

- Minimize penalized log-likelihood

$$\min l_p(f,\lambda)=l_u(f,\lambda)-\frac{1}{2}\lambda\int\{f''(t)\}^2 dt$$

- Good news: Solution is still a natural cubic spline
- Bad news: There is no analytic expression of that spline function

Nonparametric Logistic Regression (Cont.)

How to proceed?

- Use Newton-Rapson to compute spline numerically, i.e

Compute $\nabla l_p = \frac{\partial l_p(\Theta)}{\partial \Theta}, \nabla^2 l_p = \frac{\partial^2 l_p(\Theta)}{\partial \Theta \partial \Theta^T}$ (analytically)

- Compute Newton direction using current parameter and derivative information
- Compute new values of parameters using old values and update formula

$$\Theta^{new} = \Theta^{old} - \left(\nabla^2 l_p\right)^{-1} \nabla l_p$$

Multidimensional Splines

How to fit data smoothly in higher dimensions?

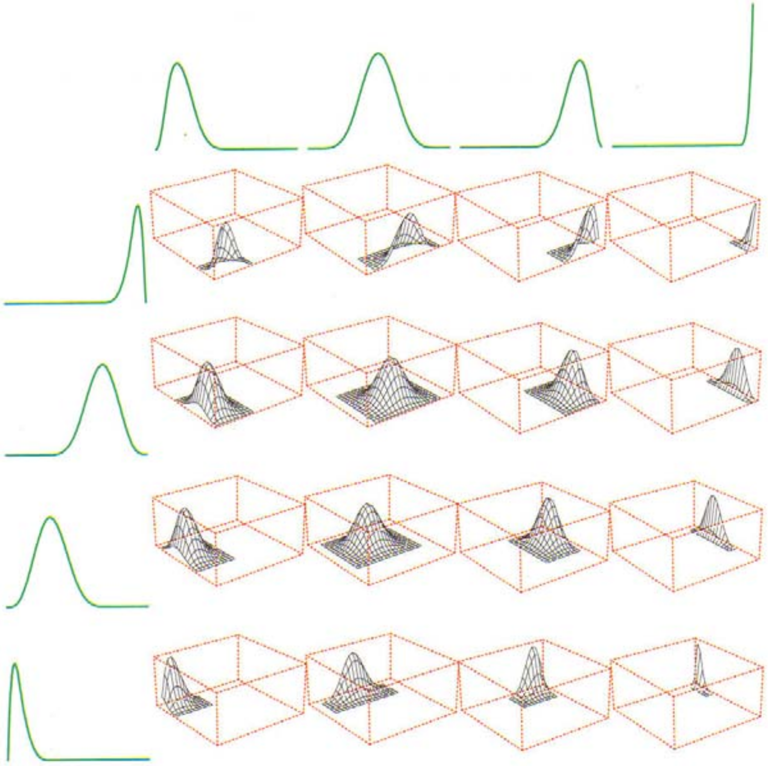
- Use basis of one dimensional functions and produce basis by tens

$$g_{jk}(X) = h_{1j}(X_1)h_{2k}(X_2),$$

$$g(X) = \sum\sum \theta_{jk} g_{jk}(X)$$

- <u>Problem</u>: Exponential

growth of basis with dim



Multidimensional Splines (Cont.)

How to fit data smoothly in higher dimensions?

Alternative: Formulate a new problem

$$\min \sum_i \left(y_i - f\left(x_i\right)\right)^2 + \lambda J\left[f\right]$$

- The solution is *thin-plate splines*
- The similar properties for λ=0
- The solution in 2 dimension is essentially sum of radial basis functions

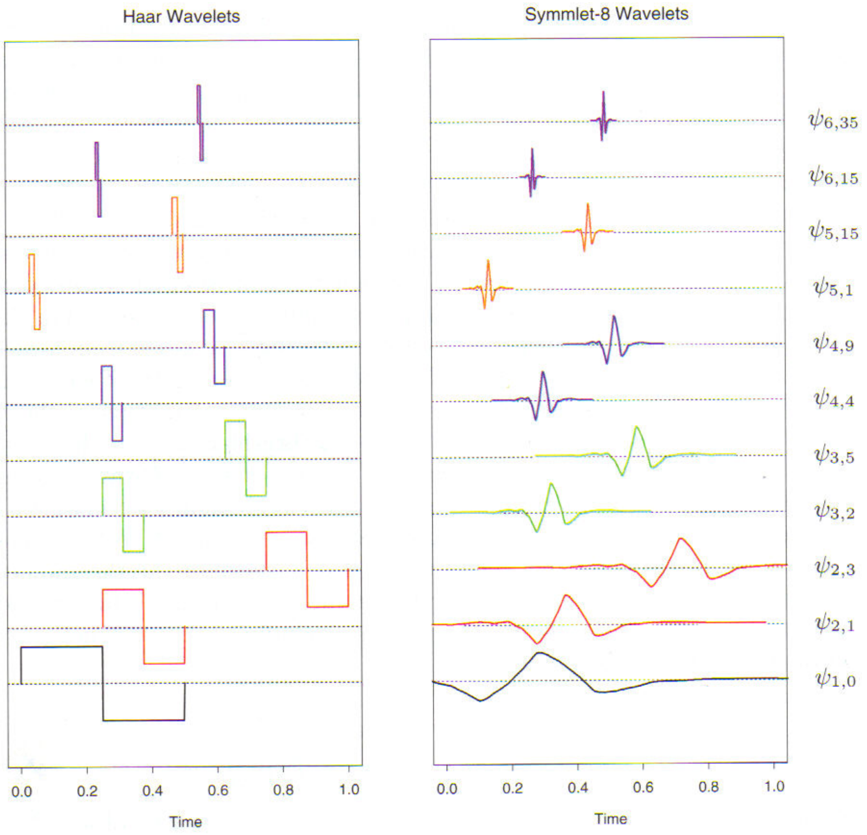
$$f\left(x\right) = \beta_0 + \beta^T x + \sum \alpha_j \eta\left(\left\|x - x_j\right\|\right)$$

Wavelets

- The idea: to fit bumpy function by removing noise
- Application area: signal processing, compression
- How it works: The function is represented in the basis of bumpy functions. The small coefficients are filtered

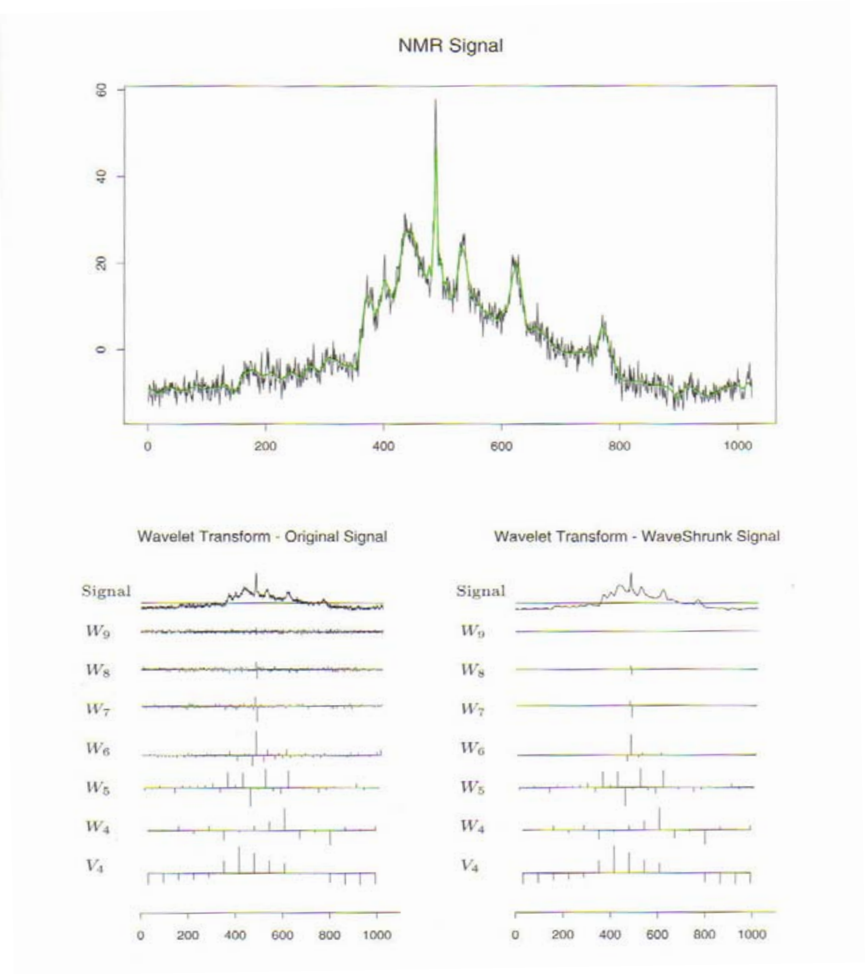
Wavelets (Cont.)

Basis functions (Haar Wavelets, Symmlet-8 Wavelets)



Wavelets (Cont.)

An Example



Summary

- Today's Class
  - Basis Expansion (Chapter 5)
- Next Classes
  - Model Selection
- Quiz 2: April 2
- Reading Assignments
  - HTF, Chapters 6 & 5