Smoothing with Roughness Penalties

1. Why do we use roughness penalties?

- Controlling smoothness by limiting the number of basis functions is discontinuous; roughness penalties allow continuous control over smoothness.
- We want to be able to define “smooth” in ways that are appropriate to our problems.
  - We may want a smooth derivative rather than just a smooth function.
  - What is smooth in one situation is not smooth in another. Smoothness has to be defined differently for periodic functions, for example.
- We find that roughness penalty smoothing gives better results.
- Roughness penalties are connected to fitting data by a differential equation; they are models for process dynamics.

2. Defining smoothness

We have two competing objectives:

1. Fit the data well; keep bias low.
2. Keep the fit smooth so as to
   - filter out noise
   - get better estimates of derivatives

$$Mean\ squared\ error = Bias^2 + Sampling\ Variance$$

We can often greatly reduce MSE by trading a little bias off against a lot of sampling variance.

Quantifying roughness

- **The classic:**

$$\mathsf{PEN}_2(x) = \int [D^2 x(s)]^2 \, ds \ .$$

$[D^2 x(s)]^2$ measures the *curvature* in x at s. This penalty measures total curvature.

- **Curvature in acceleration:**

$$\mathsf{PEN}_4(x) = \int [D^4 x(s)]^2 \, ds$$

- These two penalties also define what we mean by "smooth"; any function that has zero penalty is "hyper–smooth." A straight line for the classic, a cubic polynomial for the acceleration penalty.

Harmonic acceleration

- If the process is periodic, it is natural to think of a $constant + sinusoid$ as “hyper–smooth”.
- This suggests that we use

$$\mathsf{PEN}_H(x) = \int [D^3 x(s) + \omega^2 Dx(s)]^2 \, ds$$

where $2\pi/\omega$ is the period.

- The functions $1, sin(\omega t)$, and $\cos(\omega t)$ all have zero penalties, as does any linear combination of them.
- Writing

$$Lx(s) = D^3 x(s) + \omega^2 Dx(s)$$

we have

$$\mathsf{PEN}_H(x) = \int [Lx(s)]^2 \, ds$$

Some questions to think about

- Can we think of other *differential operators L* that might be useful?
- If we have a small number of "hyper–smooth" functions in mind, can we find a differential operator L that will assign zero penalty to them?
- Can use the data themselves to tell us something about the right differential operator L?

3. Penalized least squares estimation

- – $\mathbf{y}$ is the n-vector of data y_j to be smoothed.
  – $\mathbf{t}$ is the n-vector of values of t_j.
  – $\mathbf{W}$ is a symmetric positive definite weight matrix.
  – $x(\mathbf{t})$ is the n-vector of fitted values.
- The penalized least squares criterion is

$$\mathsf{PENSSE}_\lambda(x|\mathbf{y}) = [\mathbf{y} - x(\mathbf{t})]'\mathbf{W}[\mathbf{y} - x(\mathbf{t})] + \lambda\,\mathsf{PEN}(x)\ ,$$

- *Smoothing parameter* λ controls the amount of roughness.
  – As $\lambda \to 0$, roughness matters less and less, and $x(t)$ fits the data better and better.
  – As $\lambda \to \infty$, roughness matters more and more, and $x(t)$ becomes more and more “hyper–smooth.”

- $x(t)$ has the basis function expansion

$$x(t) = \sum_k^K c_k \phi_k(t) = \mathbf{c}'\boldsymbol{\phi}(t)$$

- For the classic penalty,

$$\begin{aligned} \text{PEN}_2(x) &= \int [D^2 \mathbf{c}'\boldsymbol{\phi}(t)]^2 \, dt \\ &= \int [D^2 \mathbf{c}'\boldsymbol{\phi}(t)][D^2 \boldsymbol{\phi}'(t)\mathbf{c}] \, dt \\ &= \mathbf{c}' \int [D^2 \boldsymbol{\phi}(t)][D^2 \boldsymbol{\phi}'(t)] \, dt \, \mathbf{c} \\ &= \mathbf{c}'\mathbf{R}\mathbf{c} \end{aligned} \tag{1}$$

- The order K *roughness penalty matrix* $\mathbf{R}$ is

$$\mathbf{R} = \int [D^2 \boldsymbol{\phi}(t)][D^2 \boldsymbol{\phi}'(t)] \, dt = \int (D^2 \boldsymbol{\phi})(D^2 \boldsymbol{\phi}')$$

The roughness penalized estimates for c and y

- $\boldsymbol{\Phi}$ is the n by K matrix of basis function values $\phi_k(t_j)$.
- The penalized least squares criterion becomes

$$\text{PENSSE}(y|c) = (\mathbf{y} - \boldsymbol{\Phi}\mathbf{c})'\mathbf{W}(\mathbf{y} - \boldsymbol{\Phi}\mathbf{c}) + \lambda\mathbf{c}'\mathbf{R}\mathbf{c}\ .$$

- This is quadratic in $\mathbf{c}$, and is minimized by

$$\hat{\mathbf{c}} = \left(\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi} + \lambda\mathbf{R}\right)^{-1}\boldsymbol{\Phi}'\mathbf{W}\mathbf{y}\ .$$

- The data-fitting vector $\hat{\mathbf{y}} = x(\mathbf{t})$ is

$$\hat{\mathbf{y}} = \boldsymbol{\Phi}\left(\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi} + \lambda\mathbf{R}\right)^{-1}\boldsymbol{\Phi}'\mathbf{W}\mathbf{y} = \mathbf{S}_{\phi,\lambda}\mathbf{y}\ ,$$

- Smoothing matrix $\mathbf{S}_{\phi,\lambda}$ maps the data into the fit.

Equivalent degrees of freedom $df(\lambda)$

- It is useful to compare a fit using a roughness penalty to one using a fixed number of basis functions.
- A measure of the "degrees of freedom" in a roughness penalized fit is

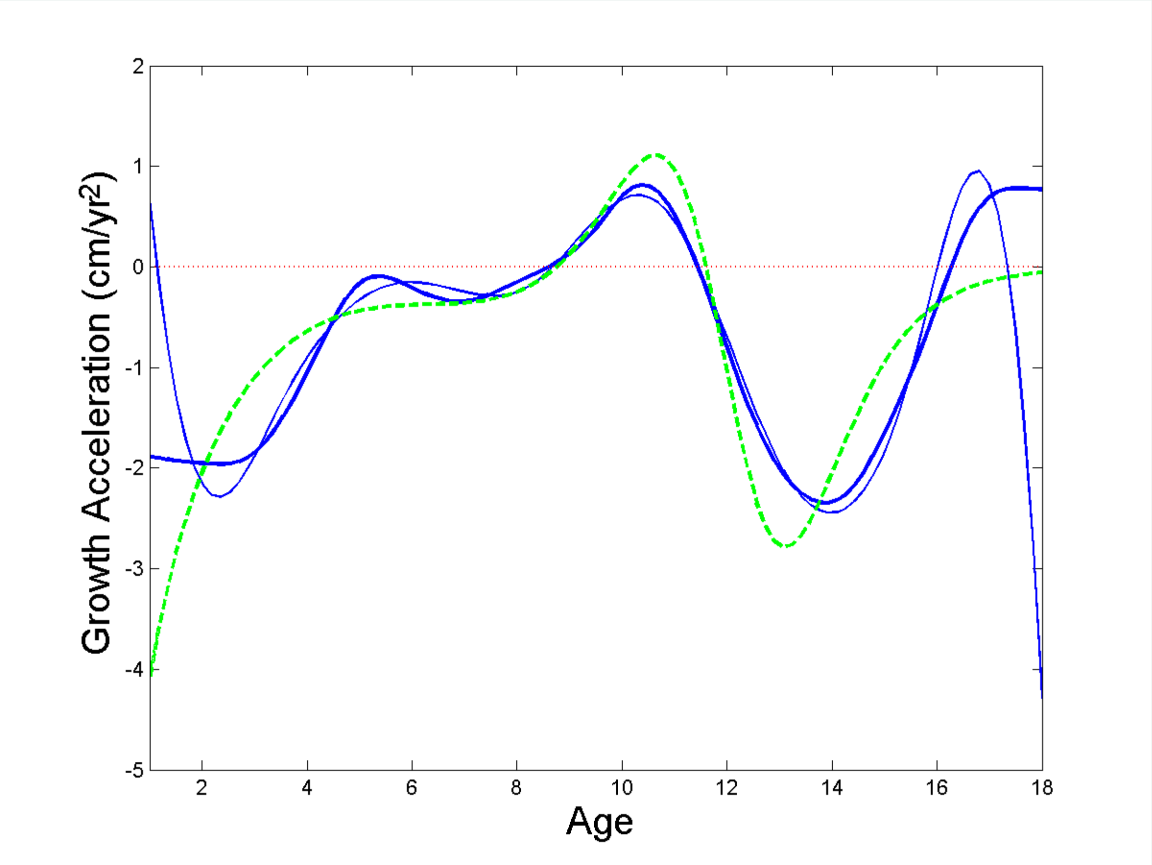
$$df(\lambda) = \text{trace}\, \mathbf{S}_{\phi,\lambda}$$

- This corresponds to the number of basis functions K in an un–penalized fit.

4. Spline Smoothing

- The term “smoothing spline” has come to mean the following procedure:
  - Use natural or B-spline basis functions.
  - Place a knot at each data point t_j.
  - Use a penalty on $D^2 x$.
- However, we find that
  - We can often achieve the same results by just using a number K of basis functions that is “large” relative to the resolution of the data.
  - We certainly want to be able to play with alternative roughness penalties.
  - Other basis functions systems are also desirable.

Two estimates of an acceleration curve.



5. Choosing smoothing parameter λ

Cross–validation for choosing the smoothing parameter λ

- In cross–validation,we
  - set aside a subset of data, the *validation sample*
  - call the balance of the data the *training sample*
  - fit the model to the training sample
  - assess fit to the validation sample
  - choose the λ value that gives the best fit

- We can also, for a sequence of values of λ,
  - set aside each observation (t_j, y_j) in turn
  - fit the data with the rest of the sample,
  - sum fits to the left out values to get a *cross–validated error sum of squares* $\mathtt{CV}(\lambda)$.
  - select the λ value that minimizes $\mathtt{CV}(\lambda)$.

Generalized cross–validation for choosing the smoothing parameter λ

- Cross-validation is time-consuming, and tends too often to under–smooth the data.
- The generalized cross-validation criterion is

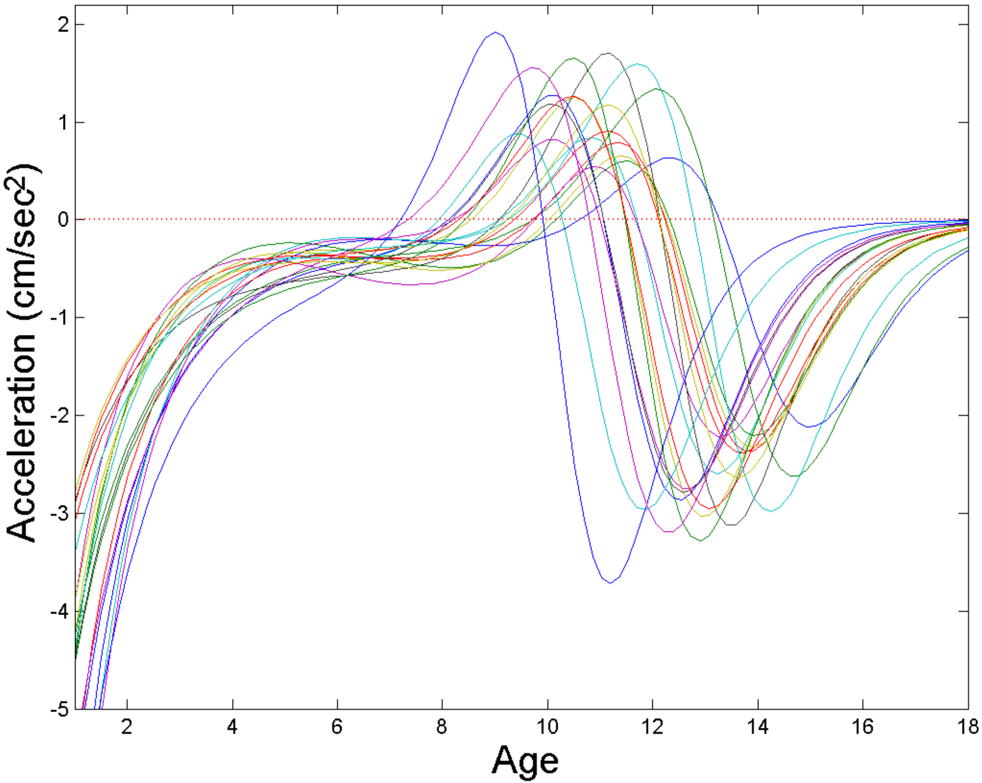
$$GCV(\lambda) = (\frac{n}{n - df(\lambda)})(\frac{\text{SSE}}{n - df(\lambda)})$$

  where df is the equivalent degrees of freedom of the smoothing operator.
- The right factor is just the unbiassed estimate s_e^2 of residual variance familiar in regression analysis.
- The left factor further "discounts" this measure further to allow for the influence of optimizing with respect to λ.

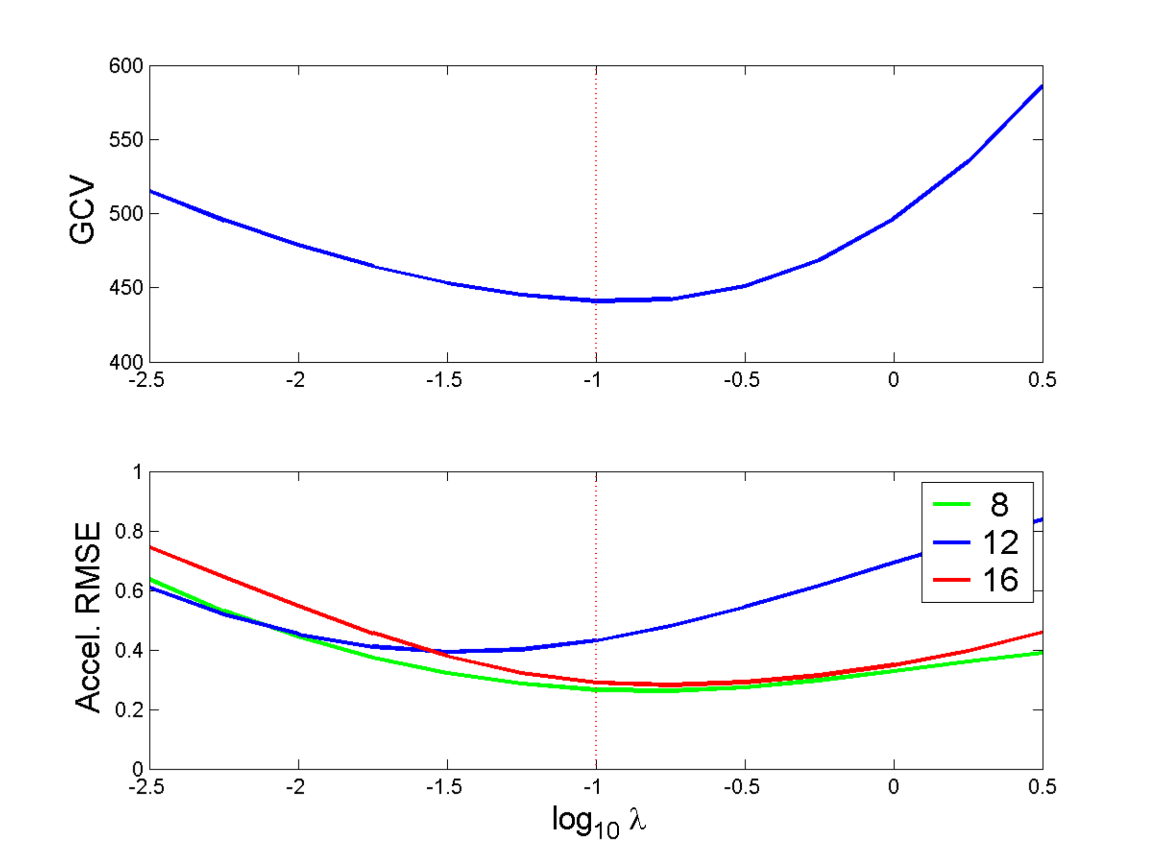
6. A simulation study

- How does GCV work in a simulated data example?
- A parametric growth model by Pierre Jolicoeur at the Université de Montréal offers a nice test problem.
- We simulate 1000 samples, each observation being a random sample from realistic Jolicoeur models plus realistic error.
- We smooth using a range of values of λ, and note the value giving the best value of GCV.
- How well do we estimate the Jolicoeur acceleration curves?

20 Jolicoeur acceleration curves



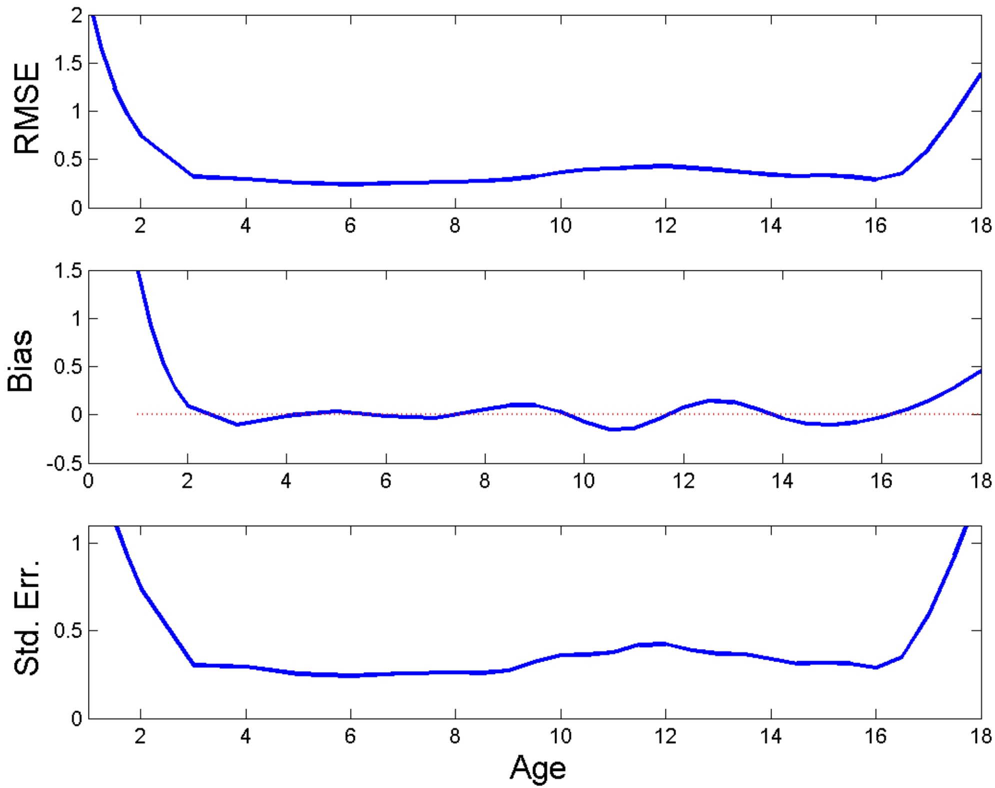
GCV and Root-Mean-Squared-Error



What we see

- In the top panel, GCV favors $\lambda = 0.1$.
- This is about right for optimal MSE for ages 8 and 16, but less smoothing would be better for age 12, in the middle of the pubertal growth spurt.
- One smoothing parameter value does not work best for all ages, but
- The value chosen by GCV certainly does a fine job.

RMSE, Bias, and Standard Error



What we see

- The performance of the spline smoothing estimate deteriorates badly at the extremes.
- The sharp curvature at the pubertal growth spurt also causes some problems.
- Except at the extremes and PGS, the bias is negligible.
- The standard error is about the same as RMSE.
- Would we do better at the extremes if the smooth respected monotonicity?

7. Confidence limits

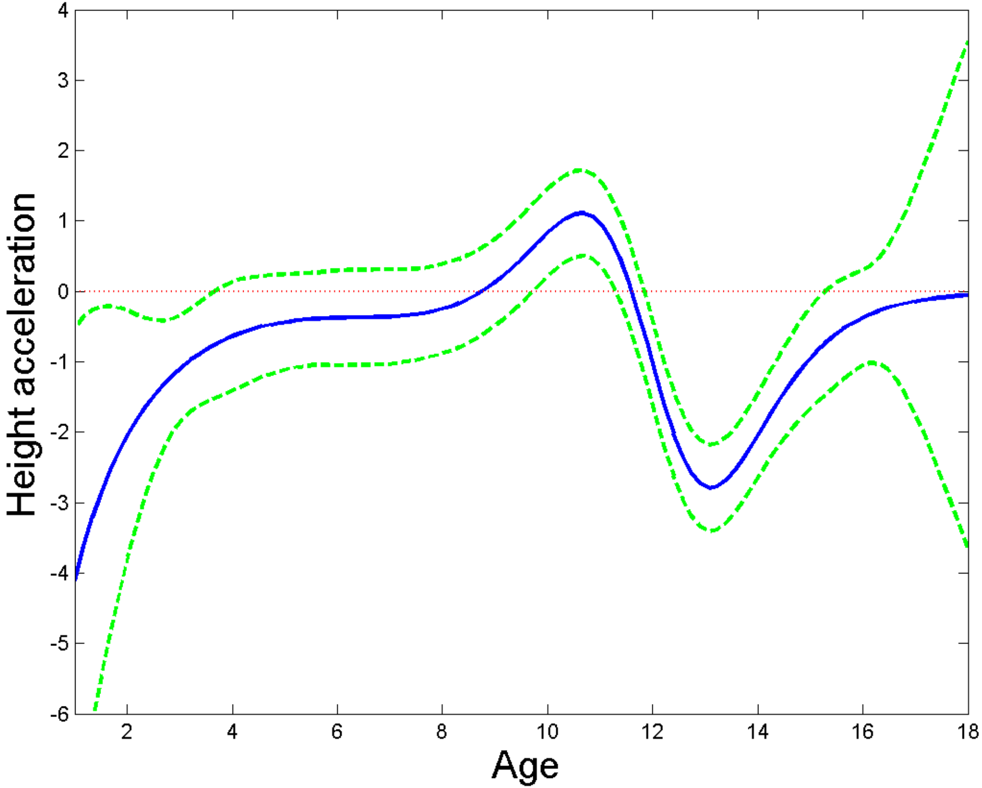
- Because the mapping from data $\mathbf{y}$ to the coefficient vector $\mathbf{c}$ is linear, it is a simple matter to work out the standard error of any linear functional of a curve defined by $\mathbf{c}$.

- The variance of a quantity $\rho(x)$ associated with linear mapping $\mathbf{M}$ from $\hat{\mathbf{c}}$ to $\hat{\rho}(x)$ is

$$\mathtt{Var}[\hat{\rho}(x)] = \mathbf{M}\mathbf{S}_{\phi,\lambda}\mathbf{\Sigma}_e\mathbf{S}_{\phi,\lambda}\mathbf{M}'$$

- Simple, that is, if we can get a good estimate of the variance-covariance matrix $\mathbf{\Sigma}_e$ of the residual vector.

95% point–wise confidence limits for growth acceleration



8. Summary

- Roughness penalization, also called *regularization*, is a flexible and effective way to ensure that an estimated function is "smooth."
- We can tailor the definition of "smooth" to our needs.
- The roughness penalty idea extends to any type of *functional parameter* that we want to estimate from the data.
- Roughness penalties are one of the main ways in which we exploit the smoothness that we assume in the process generating the data.
- "Roughness" is like *energy* in physics; roughness requires energy to produce, and smoothness implies limited energy.
- Where we imagine that the amount of energy behind the data is limited, it is natural to assume smoothness.