Statistics 840 HW8 handed out Lect 9 due lect14

See hw8b.txt for reading

1 Study the properties of GCV and the cubic smoothing spline by Monte Carlo Methods

The model:

$$y_i = f(t(i)) + \epsilon_i, i = 1, ..., n, t(i) \in [0, 1], \tag{1}$$

where the ϵ_i are independent, $\mathcal{N}(0, \sigma^2)$

1.1 Plot an Example

Select a curve f to represent 'truth' that you can represent by some simple mathematical formula (for example the curves in Craven and Wahba, or in fig 4.1 of my book), or, linear combinations of Gaussians, Beta functions or, ... It would be particularly interesting if they were based on some physical phenomenon ..these will be your 'truth functions' (f). To be well fitted by the cubic smoothing spline with GCV for the smoothing parameter, they should be smooth, have at most two or three peaks (for the sample sizes below), and not have VERY large second derivatives at the ends. For ease of comparison between student homeworks, scale f so that it is between 0 and 1. Fix n and let $t(i) = i/n, i = 1, 2, ...n$. Call a random number generator to get the ϵ_i and generate data $y_1, ...y_n$. Use this data to estimate f using a cubic smoothing spline with the GCV estimate for lambda. Plot f and the estimate $f_{\hat{\lambda}}$, $V(\lambda)$ and $R(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[f(t(i) - f_\lambda(t(i)]^2$ for this first replicate, and compute the Inefficiency $I = R(\hat{\lambda})//R(\lambda*)$, where $\hat{\lambda}$ is the GCV estimate of λ and $\lambda*$ is the optimal λ (minimizes $R(\lambda)$ for this data set.)

1.2 Simulation Study

Generate 20 replicates of this experiment (that is, use the same f, generate 20 sets of the ϵ_i, fit the spline, and record the inefficiency. Provide description of the 20 inefficiencies (use a box plot, histogram, or other method which will provide an informative picture of the inefficiencies). Determine $R(\hat{\lambda})$ for each replicate and save the 20 values of $R(\hat{\lambda})$.

Now that you have done this for a particular n and σ, repeat for 4 or 5 different n and two σ's. For n, use $n = 32, 64, 128, 256$ (and 512 if you can fit it onto the computer you're using without tying everything up). Scale f to be between 0 and 1. Then, for example, 5% noise would have $\sigma = .05$ (as a percentage of the range of f). Choose a small and a large value of σ. A small σ would be around 1 to 5% (say) while a large sigma might be somewhere around 5 to 25%.

Provide a summary of the 8 or 10 sets of inefficiencies - i. e. eight or ten histograms or box plots, or possibly tables, whatever will give a good idea of the properties of the method, from the point of view of inefficiency.

Plot log of the mean $R(\hat{\lambda})$ vs $\log n$, for each σ. For each sigma this should give a roughly straight line whose slope p will give you an idea of the convergence rate of the method. ($R \approx$ const n^{-p}, so $\log R \approx$ const $-p \log n$). Compare with the theoretical results in Section 4.5 of the book. (Note that what is being called $R(\lambda)$ here is called $T(\lambda)$ in the book.

Please label all plots and/or provide a journal-quality description of your experiments and results - by 'journal- quality description' I mean enough information about the cases you took etc. so that another person who was familiar with splines could repeat your experiment and get the same results.

2 Explore limits of GCV

There are several ways that GCV might fail to give good results.

1. sample size too small - this probably means fewer than 8 or 10 data points per local maximum
2. huge noise (like 50% or more of the signal, say)
3. errors more like roundoff than like observational error (like sigma = .0001%, for example.. -)
4. very large second derivatives at the boundaries
5. sharp spikes in one region while being very flat elsewhere
6. noise low frequency rather than 'white' - for example a low order autoregressive scheme with positive correlation between neighboring points
7. presence of large outliers (sometimes)
8. highly irregular data distribution

Pick one or more of these cases, or some other case that might defeat the method, and run a few examples to see what happens.

3 TPS convergence rates.

See if you can deduce convergence rates for the cross validated thin plate spline (TPS), either experimentally or theoretically. There is a TPS tutorial in the coursepage under R.