cs 514, lecture 03may00: spline interpolation

We are now ready to use splines to recover (not necessarily perfectly) a function from some information about it, using splines.

To be specific, we choose some real interval I, some sequence $\tau = (\tau_1 < \cdots < \tau_n)$ in I, and consider the data

$$\Lambda'_\tau g := (g(\tau_i) : i = 1{:}n)$$

for some real-valued function g on I, i.e., some

$$g \in X := \mathbb{R}^I.$$

We also choose some linear subspace $F \subset X$, of dimension n, i.e., with some basis $V : \mathbb{R}^n \to F$, and assume that the Gramian matrix $\Lambda'_\tau V$ is invertible. Then we know that

$$f = P_\tau g := V(\Lambda'_\tau V)^{-1}\Lambda'_\tau g$$

is the unique element of F that matches the given information about g, i.e., satisfies

$$\Lambda'_\tau f = \Lambda'_\tau g.$$

The classical choice for F is $\Pi_{<n}$, the space of polynomials of degree $< n$. As discussed earlier, $\Lambda'_\tau V$ is invertible for any choice of basis V for this space. However, there are some problems, as the following calculations show.

Take, specifically, $I = [-1\,..\,1]$, and choose

$$\tau = (-1 = \tau_1 < \cdots < \tau_n = 1)$$

uniformly spaced (i.e., $\tau_i = -1 + 2(i-1)/(n-1)$, $i = 1{:}n$). Also choose the **Runge example**

$$g : x \mapsto \frac{1}{1 + (5x)^2}.$$

We can compute and plot the polynomial interpolant with the aid of the command `spapi` as follows (choosing $n = 15$, to be specific):

```
g = inline('1./(1+(5*x).^2)');
xx = linspace(-1,1,101); plot(xx,g(xx)), hold on
n = 15; tau = linspace(-1,1,n); pol = spapi(n,tau,g(tau));
fnplt(pol,'k')
```

Near the end points, this polynomial interpolant overshoots the function g by more than 7.

For comparison, we also construct and plot a cubic spline interpolant to the same data:

```
cs = spapi(4,tau,g(tau)); fnplt(cs,'r'), hold off
```

It agrees with g to graphic accuracy.

What is going wrong with the polynomial interpolant? The main difficulty appears to be that the polynomial interpolant can be much bigger than the function it interpolates.

To make this precise, we introduce the *size* or *norm* of a function:

$$\|g\| := \sup_{x\in I} |g(x)|, \qquad g \in X = \mathbb{R}^I.$$

Further, we introduce

$$\|P\| := \sup_{g\in X} \frac{\|Pg\|}{\|g\|}$$

as a measure of the worst-case scenario, i.e., the biggest possible ratio of the size of Pg compared to the size of g. The larger $\|P\|$, the bigger can be the difference between some g and its interpolant.

As it turns out, there is a convenient formula for $\|P\|$, namely

$$\|P\| = \sup_{x\in I} \sum_{j=1}^{n} |\ell_j(x)|,$$

with $[\ell_1, \ldots, \ell_n]$ the basis for F dual to Λ'_τ (also called the Lagrange basis), i.e.,

$$\ell_i(\tau_j) = \delta_{ij}.$$

This means that we can estimate $\|P\|$ quite closely in the following way, first for polynomial interpolation and then for cubic spline interpolation:

```
max(sum(abs(fnval(spapi(n,tau,eye(n)),xx))))
ans =  283.1809
max(sum(abs(fnval(spapi(4,tau,eye(n)),xx))))
ans =  1.9698
```

This shows a great difference. Moreover, one can work out that, as n increases, $\|P\|$ grows like $\exp(n/2)$ for the polynomial case, while, for cubic spline interpolation, it stays around 2 regardless of n.

This very different behavior is a main reason we use splines rather than just one polynomial when trying to recover a function from function values. But this is not the whole story.

Have a look at the *error* $g - Pg$ in our interpolant Pg to g. Since $Pg \in F$, $\|g - Pg\|$ cannot be any smaller than the **distance**

$$\operatorname{dist}(g, F) := \inf_{f\in F} \|g - f\|$$

of g from F. On the other hand, since $\|P\|$ is the supremum over all ratios $\|Pg\|/\|g\|$, we know that

$$\|Pg\| \le \|P\|\|g\|, \quad g \in X.$$

We also know that $Pf = f$ for all $f \in F$, and that P is linear, hence $g-Pg = g-f+Pf-Pg = g-f+P(f-g)$. Therefore,

$$\|g - Pg\| \le \|g - f\| + \|P\|\|f - g\| = (1 + \|P\|)\|g - f\|,$$

and this holds for every $f \in F$. By choosing f to make $\|g - f\|$ as small as possible, we obtain

Lebesgue's inequality. *For all $g \in X$,*

$$\operatorname{dist}(g, F) \le \|g - Pg\| \le (1 + \|P\|) \operatorname{dist}(g, F).$$

In particular, if our interpolation process P has a modest norm, then the interpolation error $\|g - Pg\|$ is within a modest factor of the best possible error.

This shows that there are two aspects to good recovery (by interpolation or otherwise):

(1) The recovery process should have a modest norm, and that is often easily achievable with splines (and not so easy with polynomials).

(2) The distance $\operatorname{dist}(g, F)$ of g from F should be small. This, too, is often easily achievable with splines, simply by making the knot spacing sufficiently small or otherwise placing the knots appropriately, while, for polynomials, the only possibility is to make the degree large, and that leads to many difficulties.