cs 514, lecture 5apr02: recap re data maps, column maps, recovery, projectors; start on B-splines

Recall the setup so far. We are hoping to recover an element f of the ls F from information $\Lambda' f = (\lambda_i f : i \in I)$ about it, with each λ_i a lfnl, hence

$$\Lambda' : F \to \mathbb{F}^I$$

a linear map. We assume that Λ' is 1-1.

In order to avoid any difficulties with identifying $\Lambda'(F) = \operatorname{ran} \Lambda'$, we only looked at the finite-dimensional case:

$$\Lambda' f = (\lambda_i : i = 1, \ldots, n),$$

and assumed that Λ' is onto.

Then F is necessarily n-dimensional, hence has a basis consisting of n elements, or, equivalently, there is a reconstruction map, or column map,

$$W : \mathbb{F}^n \to F : a \mapsto \sum_{j=1}^{n} w_j a(j) =: [w_1, \ldots, w_n]a$$

that is invertible.

Under that assumption, the Gram matrix $\Lambda' W = (\lambda_i w_j : i, j = 1, \ldots, n)$ is invertible, and the formula

$$f = W(\Lambda' W)^{-1} \Lambda' f, \qquad \forall f \in F,$$

provides the complete reconstruction of $f \in F$ from the information $\Lambda' f$ about it.

We observed that this amounts to nothing more than a change of basis: Since $\Lambda' : F \to \mathbb{F}^n$ is invertible, its inverse, $V := (\Lambda')^{-1}$, is a linear map, hence necessarily of the form

$$V : \mathbb{F}^n \to F : a \mapsto \sum_{j=1}^{n} v_j a(j) =: [v_1, \ldots, v_n]a,$$

and, necessarily,

$$\mathrm{id} = \Lambda' V = (\lambda_i v_j : i, j = 1, \ldots, n)$$

the identity matrix, i.e., the two sequences (λ_i) and (v_j) are biorthonormal and, correspondingly,

$$f = V \Lambda' f = \sum_{j=1}^{n} v_j \lambda_j f, \qquad \forall f \in F.$$

In fact, necessarily

$$V = W(\Lambda' W)^{-1}.$$

Hence, our reconstruction consisted of the following: obtain, from the coordinates $\Lambda' f$ of f wrto the basis V, the coordinates $(\Lambda' W)^{-1} \Lambda' f$ of f wrto the basis W.

Nevertheless, we counted this a success to the extent that the coordinates of f wrto the basis W, more readily than $\Lambda' f$, provide the information about f we are after.

This led us to the basic question: *What information about $f \in F$ is readily obtainable from its coordinates wrto a given basis W for F?*

We are going to look in detail into this question for the case that F is a spline space, and the basis in question is its B-spline basis.

But before starting that discussion, here are two more general observations, at this more abstract level, that will be important.

1. **linear projectors**: Extend our setup slightly, by assuming that our linear space F is actually a linear subspace of some linear space X. A concrete example would be

$$F = \Pi_{<n} \subset C(\mathbb{R}) =: X.$$

Correspondingly, assume that our data map, Λ', is actually defined on all of X. For our earlier concrete example,

$$\Lambda' f = (f(\tau_i) : i = 1, \ldots, n),$$

this is obviously true (provided $\tau_i \in \mathbb{R}$, all i). Then we can construct

$$Pg := W(\Lambda' W)^{-1} \Lambda' g$$

for every $g \in X$, but, while $g = Pg$ whenever $g \in F$, this need not hold for every $g \in X$. In fact, since necessarily $Pg \in F$ (why?), it follows that

$$g = Pg \quad \Longleftrightarrow \quad g \in F.$$

In other words,

$$\operatorname{ran} P = F, \quad PP = P.$$

A linear map with the property $P^2 = P$ is called **idempotent** or a **projector**. So, P is a linear projector onto F.

What is the relationship between $g \in X$ and Pg? Since

$$\Lambda' Pg = \Lambda' W(\Lambda' W)^{-1} \Lambda' g = \Lambda' g,$$

we recognize Pg as the *unique element of F that agrees with g on Λ'* and therefore call it the unique **interpolant from F to the data $\Lambda' g$**. The most commonly used approximation schemes are all of this form, with the only difference being the choice of the data map, i.e., the choice of the information to be matched.

By considering g from a superspace for F, we cannot hope for lossless reconstruction; still, it is reconstruction of a sort.

2. Earlier in this course, a situation more special and more general was considered. The ls F was usually infinite-dimensional, but a Hilbert space. But, even when when F was finite-dimensional, the data map, while 1-1, was not assumed to be onto. This provided some extra flexibility, but it is important to realize what was lost.

If $\Lambda' = [\lambda_1, \ldots, \lambda_n]' : F \to \mathbb{F}^n$ is not onto, then

$$m := \dim F = \dim \Lambda'(F) < n.$$

In particular, our basis W for F now is a map from $\mathbb{F}^m$ rather $\mathbb{F}^n$, hence, offhand, not helpful for the reconstruction. Rather, we now must choose some column map $U : \mathbb{F}^n \to F$, necessarily not 1-1, so that

$$f = U\Lambda' f, \quad \forall f \in F,$$

but it is not obvious how to do this. If U is such a map, then, certainly, the Gramian $\Lambda' U$ must be the identity on $\operatorname{ran} \Lambda'$, but that is far from having it be the identity matrix.

In contrast, if all we know is that Λ' is some data map to $\mathbb{F}^n$ and that $F = \operatorname{ran} W$ for some reconstruction map $W : \mathbb{F}^n \to X$, then just the fact that the Gramian $\Lambda' W$, which is square, is either 1-1 or onto tells us already that it must be invertible, and Λ' must be onto even when restricted to F, and W must be a basis for F, and we get the earlier reconstruction formula for all $f \in F$.

A standard sufficient condition for a square matrix to be invertible is for the matrix to be triangular with nonzero diagonal entries.

Before tackling splines, a quick discussion of a basis V for $F := \Pi_{<n}$ that, in my experience, is a 'good' basis since the coordinates $V^{-1} f$ of $f \in F$ are relatively close to standard information about f. The basis in question is the **Newton basis**:

$$V = (v_j := \prod_{j<k\leq n} (\cdot - \tau_k) : j = 1, \ldots, n),$$

with $\tau_2, \ldots, \tau_n$ arbitrary points.

If all the τ_j are 0, we get the power basis as a special case.

Evaluation of the 'Newton form' Va is via **nested multiplication** (see the notes for 3apr). Here it is, in `Matlab`:

```
function vals = nestmult(x,tau,a)
%NESTMULT value at x of the polynomial in Newton form
%        p(x) = sum_j a(j) prod_{k>j}(x-tau(k))
% with both A and TAU n-vectors (TAU(1) is ignored).
% VALS is of the same size as X, with VALS(i,j,...) containing the value of p
% at X(i,j,...).
vals = repmat(a(1),size(x));
for j=2:length(a)
   vals = a(j) + (x-tau(j)).*vals;
end
```

The 'rows' λ_i of the corresponding analysis map $[\lambda_1, \ldots, \lambda_n]' := \Lambda' := V^{-1}$ are 'divided differences'.

B-splines defined

The traditional definition of a B-spline is in terms of the divided difference of the truncated power function

$$()^j_+ : x \mapsto x^j_+ := \begin{cases} x^j, & x \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

However, since I did not take the time to introduce divided differences, I'll introduce B-splines instead 'out of the blue' via their recurrence, with the B-spline of order 1 the starting point for that recurrence.

Definition. *The* **B-spline with knots** a**,** b *is the characteristic function of the half-open interval* $[a \,..\, b)$*, i.e.,*

$$B(x|a,b) := \chi_{[a..b)}(x) = \begin{cases} 1, & a \leq x < b; \\ 0, & \textit{otherwise} \end{cases}.$$

Observations

- $B(\cdot|a,b)$ is piecewise constant, with a breaks a and b (and nowhere else), in case $a < b$; in particular, it is positive on the interval $[a \,..\, b)$, and is zero elsewhere.
- $B(\cdot|a,b)$ is the zero function in case $a \geq b$.
- $B(\cdot|a,b)$ is **right-continuous**, i.e.,

$$B(x|a,b) = B(x^+|a,b) := \lim_{h \downarrow 0} B(x+h|a,b).$$

The choice of making the B-spline right-continuous is just that, a choice, since some choice has to be made, and other choices would have been possible, e.g., the more symmetric choice

$$f(x) := (f(x-) + f(x+))/2.$$

The particular choice made is connected to the ppform, a standard way to represent piecewise polynomials.

If now

$$\mathbf{t} := (\cdots \leq t_i \leq t_{i+1} \leq \cdots)$$

is a given nondecreasing sequence, we associate with it the sequence

$$B_i = B_{i,1,\mathbf{t}} := B(\cdot|t_i, t_{i+1}), \quad \forall i,$$

of first-order B-splines, and find that this sequence is a **partition of unity**, i.e.,

$$\sum_i B_i(x) = 1, \quad \inf_i t_i < x < \sup_i t_i.$$

Further, if $t_i < t_{i+1}$ for all i, then $(B_i : i)$ provides a 'basis' for the space

$$\Pi_{<1,\mathbf{t}}$$

of all (right-continuous) piecewise constant functions with possible breaks at the t_i's and nowhere else. Here, I have put 'basis' in quotes since I don't exclude the possibility that the sequence $\mathbf{t}$ is infinite or even bi-infinite, hence correspondingly consider the infinite or even bi-infinite sum

$$f = \sum_i a(i)B_i : x \mapsto \sum_i a(i)B_i(x) = \begin{cases} a(j), & t_j \le x < t_{j+1}; \\ 0, & \text{otherwise}, \end{cases}$$

i.e., as a pointwise sum.

In practice, one usually only considers finite sequences $\mathbf{t}$, but it is convenient to permit $\mathbf{t}$ to be nonfinite. In fact, in your consideration of B-splines so far, the sequence $\mathbf{t}$ was always bi-infinite, namely the sequence $\mathbf{t} = \mathbb{Z}$.

Definition. *With* $\mathbf{t}$ *continuing to denote a nondecreasing sequence,*

$$B_i := B_{i,2} := B_{i,2,\mathbf{t}} := B(\cdot|t_i, t_{i+1}, t_{i+2})x \mapsto \begin{cases} \frac{x-t_i}{t_{i+1}-t_i}, & t_i \le x < t_{i+1} \\ \frac{t_{i+2}-x}{t_{i+2}-t_{i+1}}, & t_{i+1} \le x < t_{i+2} \\ 0, & \textit{otherwise}, \end{cases}$$

is the **B-spline with knots** t_i, t_{i+1}, t_{i+2}.

Observations

- $B_{i,2} = B(\cdot|t_i, t_{i+1}, t_{i+2})$ is piecewise linear, with breaks at t_i, t_{i+1}, t_{i+2}; it is positive on $(t_i \mathinner{..} t_{i+2})$ and is zero off $[t_i \mathinner{..} t_{i+2}]$.
- $B_{i,2}$ is continuous from the right and is continuous at any of its breaks if and only if that break appears only once the the sequence $\mathbf{t}$.

It is still true that

$$\sum_i B_{i,2}(x) = 1, \quad \inf_i t_i < x < \sup_i t_i,$$

except for some possible trouble near the first or last t_i, if there is one. It is a little bit harder, though, to describe precisely the span of $(B_{i,2} : i)$.

Now the following observation: *With* $\omega_{i,2}$ *the linear polynomial*

$$\omega_{i,2} : x \mapsto \frac{x - t_i}{t_{i+1} - t_i}, \quad \forall i,$$

we have

$$B_{i,2} = \omega_{i,2}B_{i,1} + (1 - \omega_{i+1,2})B_{i+1,1}.$$

Thus $B_{2,i}$ fits the following **recursive definition** of the B-spline.

Definition. *With* $\mathbf{t}$ *a nondecreasing sequence and* k *a positive integer, the B-spline with knots* $t_i, \ldots, t_{i+k}$ *is*

$$B(\cdot|t_i, \ldots, t_{i+k}) := B_{i,k} := B_{i,k,\mathbf{t}} := \begin{cases} \chi_{[t_i \ldots t_{i+1})}, & k = 1; \\ \omega_{i,k}B_{i,k-1} + (1 - \omega_{i+1,k})B_{i+1,k-1}, & k > 1, \end{cases}$$

where $\omega_{i,k}$ *is the linear polynomial*

$$\omega_{i,k} : x \mapsto \frac{x - t_i}{t_{i+k-1} - t_i}.$$

Use the `MATLAB Spline Toolbox` command `bspligui` to become more familiar with just how $B_{i,k,\mathbf{t}}$ depends on its $k + 1$ knots.

Also, marvel at the fact that the recurrence relation takes two functions of a certain smoothness, multiplies each of them by some linear polynomial, and then adds the resulting products and, in this way, obtains a function with higher smoothness than either summand.

The reference on B-splines and such that I favor (naturally) is my book *A practical guide to splines*, Revised edition, Springer-Verlag, 2001. A (picturesque but proofless) overview is provided in the article posted on my cs514 webpage.

