Reconstruction, or

how to recover a function from information about it

By now, you are used to the idea of the **analysis operator** aka **data map**. This is a map of the form

$$\Lambda' : F \to \mathbb{F}^I : f \mapsto (\lambda_i(f) : i \in I)$$

i.e., from functions to numerical sequences.

So far, F has been an inner-product space and, correspondingly, we have thought of the λ_i as elements of F itself, i.e., $\lambda_i(f) = \langle f, \lambda_i \rangle$. Correspondingly, we have used Λ^* to denote the corresponding data map:

$$\Lambda^* : F \to \mathbb{F}^I : f \mapsto (\langle f, \lambda_i \rangle : i \in I).$$

But now, I want to look at more general data maps, and using Λ' rather than Λ^* is meant to signal that. Along with this, I will merely assume that F is a *linear space*, with the scalar field $\mathbb{F}$ being either $\mathbb{R}$ or $\mathbb{C}$. As a quite concrete example, you might take

$$F = \Pi := \{t \mapsto \sum_{j=1}^{n} a(j) t^{j-1} : a(j) \in \mathbb{F}, n \in \mathbb{N}\},$$

the linear space of all $\mathbb{F}$-valued polynomials on the real line. Further, I'll take each λ_i to be a **linear functional** on F, meaning a scalar-valued map on F that is *linear*. To recall, having λ_i be linear means that

$$\lambda_i(f + \alpha g) = \lambda_i f + \alpha \lambda_i g, \quad \forall f, g \in F, \ \forall \alpha \in \mathbb{F}.$$

This is certainly true for the functionals $f \mapsto \langle f, \lambda_i \rangle$ used so far. But it is also true for the following common functionals when, e.g., $F = \Pi$ (and mentioned already earlier in the course):

- the evaluation functional:
$$\delta_t : f \mapsto f(t);$$
- the derivative evaluation functional:
$$\delta_t D^r : f \mapsto f^{(r)}(t);$$
- the weighted average functional:
$$f \mapsto \int_s^t w(u) f(u) \, \mathrm{d}u,$$
with w a nonnegative function with $\int_s^t w(u) \, \mathrm{d}u = 1$.

On the other hand, numerical information like

$$f \mapsto \|f\|_\infty := \sup_t |f(t)|$$

or

$$f \mapsto \sup_{a \le t \le b} f(t)$$

or

$$f \mapsto \#\{t \in [a \,..\, b] : f(t) = 0\},$$

while possibly very useful, is not a *linear* functional of f, and so is excluded from the λ_i that describe our data map Λ'.

With this assumption, Λ' itself is a linear map, i.e.,

$$\Lambda'(f + \alpha g) = \Lambda' f + \alpha \Lambda' g, \quad \forall f, g \in F, \ \forall \alpha \in \mathbb{F}.$$

This linearity is going to be essential to what is to follow.

Basic Question. *Given $\Lambda' f$, what can one say about f?*

I will deal with this question mostly in its most extreme form:

Problem 1. *Given $\Lambda' f$, tell me f.*

In other words, I am looking for *perfect recovery* of f from the numerical information $\Lambda' f$ about it. This puts certain demands on our data map Λ'.

Demand 1. *Λ' must be* **1-1**, *i.e., $f \neq g \Longrightarrow \Lambda' f \neq \Lambda' g$.*

Indeed, if $\Lambda' g = \Lambda' h$ for some $g \neq h$, then we have no hope of recovering g or h from $\Lambda' g = \Lambda' h$ since Λ' fails to distinguish between the two. Worse than that, once this happens for some pair g, h, then we are unable to recover any $f \in F$ from its numerical information $\Lambda' f$.

For, since Λ' is linear, we conclude that $\Lambda'(g-h) = 0$. Since, by assumption, $g - h \neq 0$, this says that *the kernel of Λ' is nontrivial*, meaning that

$$\ker \Lambda' := \{f \in F : \Lambda' f = 0\} \neq \{0\}.$$

And that is bad news since

$$\Lambda'(f+k) = \Lambda' f + \Lambda' k = \Lambda' f, \qquad \forall k \in \ker \Lambda',\ f \in F.$$

Hence, if there is some pair $g \neq h$ mapped by Λ' to the same numerical information, then, for every $f \in F$, there is $g \neq f$ that looks the same as f as far as Λ' can tell.

Put positively, *Λ' is 1-1 if and only if $\ker \Lambda'$ is* **trivial**, meaning that $\ker \Lambda' = \{0\}$, i.e., if and only if only 0 is mapped to 0. (For sure, any linear map must map 0 to 0.)

Demand 2. *We must have a description of*

$$\operatorname{ran} \Lambda' := \Lambda'(F) = \{\Lambda' f : f \in F\}.$$

For sure, someone has supplied us with what is claimed to be $\Lambda' f = (\lambda_i f : i \in I)$, but really all we hold is a numerical sequence $(a(i) : i \in I) \in \mathbb{F}^I$. E.g., suppose we are forced to round these values as we enter them into the computer; how can we be certain that, for these slightly perturbed values $\tilde{a} = (\tilde{a}(i) : i \in I)$, there is some $\tilde{f} \approx f$ with $\Lambda' \tilde{f} = \tilde{a}$?

Since Λ' is a linear map, we know that $\operatorname{ran} \Lambda'$ is a linear subspace of $\mathbb{F}^I$, but we need to know more than that.

Demand 2 may be very hard to meet in general. For example, if $F = \Pi$, then

$$\Lambda' : F \to \mathbb{R}^{\mathbb{N}} : f \mapsto (f(n) : n \in \mathbb{N})$$

is certainly 1-1 (why???), but not every $a \in \mathbb{R}^{\mathbb{N}}$ is of the form $(f(n) : n \in \mathbb{N})$ for some polynomial f. E.g., the sequence $(1, 0, 0, \ldots)$ is not (why???). Yet it is offhand tricky to identify all the a that actually are of the form $(f(n) : n \in \mathbb{N})$ for some polynomial f. In fact, the only situation I know where it is at all easy to describe $\operatorname{ran} \Lambda'$ occurs when we are dealing with a *finite* amount of information, i.e., with finitely many linear functionals:

$$\Lambda' : F \to \mathbb{F}^n : f \mapsto (\lambda_i f : i = 1, \ldots, n),$$

say.

In this setting, Linear Algebra furnishes the basic formula

$$\dim \ker \Lambda' + \dim \operatorname{ran} \Lambda' = \dim F.$$

From Demand 1, we know that $\ker \Lambda' = \{0\}$, i.e., $\dim \ker \Lambda' = 0$. Hence we know that $\dim \operatorname{ran} \Lambda' = \dim F$. We also know that $\operatorname{ran} \Lambda'$ is a linear subspace of the n-dimensional linear space $\mathbb{F}^n$, hence conclude that $\dim F \leq n$ *with equality if and only if* $\operatorname{ran} \Lambda' = \mathbb{F}^n$.

So, for the time being, we change Demand 2 to the following, which is *not* satisfied by $F = \Pi$, but might be satisfied, e.g., when

$$F = \Pi_{<n} := \{t \mapsto \sum_{j=1}^{n} a(j)t^{j-1} : a(j) \in \mathbb{F}\}.$$

Demand 2'. $\Lambda' : F \to \mathbb{F}^n$ *and* $\dim F = n$.

Then we know that $\Lambda' : F \to \mathbb{F}^n$ is 1-1 *and* onto, i.e., invertible. This means that there is a (unique) map $V : \mathbb{F}^n \to F$, necessarily linear, so that

$$\Lambda' V = \mathrm{id}_{\mathbb{F}^n}, \quad \textit{and} \quad V\Lambda' = \mathrm{id}_F. \tag{1}$$

More than that, we only have to check one of these conditions in order to know that both conditions hold. In particular, $\Lambda' V = \mathrm{id}_{\mathbb{F}^n}$ implies that

$$V\Lambda' f = f, \quad \forall f \in F.$$

In other words, we have completely solved the recovery problem.

Except, given Λ', just how do we construct V?

Well, what exactly does a linear map $V : \mathbb{F}^n \to F$ from the **coordinate space** $\mathbb{F}^n$ to our linear space F look like?

Let

$$\mathbf{i}_j := (\delta_{jk} : k = 1, \ldots, n) = (\underbrace{0, \ldots, 0}_{j-1 \text{ terms}}, 1, 0, \ldots) \in \mathbb{F}^n$$

be the jth coordinate vector in $\mathbb{F}^n$. Then, for any $a \in \mathbb{F}^n$, $a = \sum_j a(j)\mathbf{i}_j$. Therefore,

$$Va = V(a(1)v_1 + \cdots + a(n)v_n) = a(1)V\mathbf{i}_1 + \cdots + a(n)V\mathbf{i}_n.$$

Or, with the definition

$$v_j := V\mathbf{i}_j, \qquad j = 1, \ldots, n,$$

we have

$$Va = a(1)v_1 + \cdots + a(n)v_n. \tag{2}$$

In other words, *once we know* $v_j = V\mathbf{i}_j$ *for all* j*, we know* V *completely in the sense that we can compute* Va *for any* a *by (2).*

There is a special case of this well-known to you: Take $F = \mathbb{F}^m$, i.e., also F is a coordinate space. In that case, each v_j is an m-vector, and we are used to think of V as the $m \times n$-matrix

$$V = [v_1, \ldots, v_n] = \begin{bmatrix} v_1(1) & \cdots & v_n(1) \\ \vdots & \ldots & \vdots \\ v_1(m) & \cdots & v_n(m) \end{bmatrix}$$

since

$$Va = \begin{bmatrix} v_1(1) & \cdots & v_n(1) \\ \vdots & \ldots & \vdots \\ v_1(m) & \cdots & v_n(m) \end{bmatrix} \begin{bmatrix} a(1) \\ \vdots \\ a(n) \end{bmatrix} = \begin{bmatrix} v_1(1) \\ \vdots \\ v_1(m) \end{bmatrix} a(1) + \cdots + \begin{bmatrix} v_n(1) \\ \vdots \\ v_n(m) \end{bmatrix} a(n) \tag{3}$$

is precisely the product of the matrix V with the 1-column matrix $[a]$.

For this reason, I will also denote the linear map $V : \mathbb{F}^n \to F : a \to \sum_j v_j a(j)$ by $[v_1, \ldots, v_n]$, even when F is not a coordinate space, and even refer to v_j as its j-th **column**. Such a V is called a **synthesis operator** or **reconstruction map**, as it constructs an element of F from numerical information. I will also call it a **column map**, but that term, along with the notation $[v_1, \ldots, v_n]$, is completely nonstandard.

With this notation in hand, let's now look again at the two conditions (1) that characterize the inverse of the data map Λ' we started with. The first one says that $\Lambda' V = \mathrm{id}_{\mathbb{F}^n}$, but what exactly is $\Lambda' V$?

If you have trouble understanding a particular map, apply it to a typical element in its domain and see what you get:

$$\begin{aligned}(\Lambda' V)a = \Lambda'(Va) &= \Lambda'(v_1 a(1) + \cdots + v_n a(n)) \\ &= \Lambda'(v_1)a(1) + \cdots + \Lambda'(v_n)a(n) = [\Lambda' v_1, \ldots, \Lambda' v_n]a\end{aligned}$$

while

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

In other words,

$$\Lambda' V = (\lambda_i v_j) = \begin{bmatrix} \lambda_1 v_1 & \cdots & \lambda_1 v_n \\ \vdots & \ldots & \vdots \\ \lambda_n v_1 & \cdots & \lambda_n v_n \end{bmatrix}.$$

This matrix is often called the **Gramian** of the two sequences (λ_i) and (v_j). Our condition $\Lambda' V = \mathrm{id}_{\mathbb{F}^n}$ says that this matrix is to be the identity matrix, i.e.,

$$\lambda_i v_j = \begin{cases} 1, & i = j; \\ 0, & \text{otherwise.} \end{cases}$$

In other words, the two sequences (λ_i) and (v_j) should be **bi-orthonormal**.

Under this condition, we know that $V\Lambda' = \mathrm{id}_F$. Again, we work out the details by applying $V\Lambda'$ to some $f \in F$:

$$(V\Lambda')f = V(\Lambda' f) = V(\lambda_i f : i = 1, \ldots, n) = v_1 \lambda_1 f + \cdots + v_n \lambda_n f,$$

and this we know to equal f, i.e., our recovery problem has the solution

$$f = \sum_j v_j \lambda_j f = \sum_j (\lambda_j f) v_j, \quad \forall f \in F.$$

Example $F = \Pi_{<n}$, $\Lambda' : f \mapsto (f(\tau_i) : i = 1, \ldots, n)$ for some n-set $\{\tau_1, \ldots, \tau_n\}$. In this case, Lagrange has provided us with the functions

$$\ell_j : t \mapsto \prod_{i \neq j} \frac{t - \tau_i}{\tau_j - \tau_i}, \quad j = 1, \ldots, n. \tag{4}$$

Each of these is a product of $n - 1$ linear factors, hence in F. Also,

$$\ell_j(\tau_k) = \begin{cases} 1, & j = k; \\ 0, & \text{otherwise.} \end{cases}$$

In other words $V := [\ell_1, \ldots, \ell_n]$ maps into $\Pi_{<n}$ and satisfies:

$$\Lambda'[\ell_1, \ldots, \ell_n] = [\Lambda' \ell_1, \ldots, \Lambda' \ell_n] = [\mathbf{i}_1, \ldots, \mathbf{i}_n] = \mathrm{id}_{\mathbb{F}^n}.$$

Therefore

$$f = \sum_{j=1}^{n} f(\tau_j) \ell_j, \qquad \forall f \in \Pi_{<n}.$$

$\square$

But, without Lagrange's help, how would we have known to make up these ℓ_j? What exactly would we do when F is not just $\Pi_{<n}$?

Well, Demand 2' included the requirement that $\dim F = n$ and this means, by definition, that F has some basis consisting of n terms. To recall: the sequence $(w_1, \ldots, w_n)$ being a basis for F means that every $f \in F$ can be written in exactly one way in the form

$$f = a(1)w_1 + \cdots + a(n)w_n$$

for some choice of $a \in \mathbb{F}^n$. In other words, the column map

$$W := [w_1, \ldots, w_n] : \mathbb{F}^n \to F : a \mapsto [w_1, \ldots, w_n]a$$

is 1-1 and onto, i.e., invertible. Since Λ' is also invertible, it follows that the Gramian *matrix* $\Lambda' W$ must also be invertible, hence

$$V := W(\Lambda' W)^{-1} \tag{5}$$

is a well-defined linear map from $\mathbb{F}^n$ to F, hence a column map to F, and

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

Therefore, this V must be our sought-for synthesis operator. With this,

$$f = Wb = w_1 b(1) + \cdots + w_n b(n), \quad \text{with } b := (\Lambda' W)^{-1}(\Lambda' f), \quad \forall f \in F.$$

Example, continued For $F = \Pi_{<n}$, the 'natural' basis is the **power basis**, i.e., the sequence $(w_1, \ldots, w_n)$ with

$$w_j := (\,)^{j-1} : t \mapsto t^{j-1}, \quad \text{all } j.$$

With $\Lambda' f = (f(\tau_j) : j = 1, \ldots, n)$ as before, the matrix

$$\Lambda' W = (\tau_i^{j-1} : i, j = 1, \ldots, n)$$

is the well-known **Vandermonde** matrix. We recover $f \in \Pi_{<n}$ from $(f(\tau_j) : j = 1, \ldots, n) = \Lambda' f$ as

$$f(t) = \sum_{j=1}^{n} b(j) t^{j-1}, \quad b := (\Lambda' W)^{-1}(f(\tau_j) : j = 1, \ldots, n).$$

For a check of that perhaps mysterious equation (5), let's take $n = 2$. Then

$$\Lambda' W = \begin{bmatrix} 1 & \tau_1 \\ 1 & \tau_2 \end{bmatrix}, \quad \text{hence, e.g., by Cramer's rule} \quad (\Lambda' W)^{-1} = \begin{bmatrix} \tau_2 & -\tau_1 \\ -1 & 1 \end{bmatrix} / (\tau_2 - \tau_1).$$

Therefore,

$$\begin{aligned} V = W(\Lambda' W)^{-1} &= [(\,)^0, (\,)^1] \begin{bmatrix} \tau_2 & -\tau_1 \\ -1 & 1 \end{bmatrix} / (\tau_2 - \tau_1) \\ &= [(\tau_2 - (\,)^1)/(\tau_2 - \tau_1), (-\tau_1 + (\,)^1)/(\tau_2 - \tau_1)] = [\ell_1, \ell_2]. \end{aligned}$$

□

With this, our long journey of recovery is finished, at least for the case of a finite amount of linear information. But here is a final point to notice. Also V is an invertible column map to F, hence its columns form a basis for F. We started off with the numerical information $\Lambda' f$. But since, by the very construction of V,

$$f = \sum_j (\lambda_j f) v_j, \quad \forall f \in F,$$

our initial information $\Lambda' f$ is nothing but the coordinates for f wrto the basis map V. From this, we ended up writing f in terms of the basis $(w_1, \ldots, w_n)$. In other words, all we accomplished was a *change of basis*, obtaining the coordinates for f wrto W from its coordinates wrto V. This leads me to the final question:

WHY BOTHER???

The answer has to be that there is some advantage in knowing the coordinates of f wrto W over knowing its coordinates wrto V. This leads to the modified basic question:

Basic Question modified. *What information about $f \in F$ is readily available from its coordinates wrto to a given basis for F?*

To be sure, once we know the coordinates of f wrto any particular basis, we know in principle everything about f. But the whole point of a particular representation is that it might provide certain information particularly readily.

==================== did not cover the rest of this =================

Example 1. $F = \Pi_0$, $V = [v]$ with $v = ()^0 : t \mapsto 1$.
If we know that $f = a(1)v$, then $a(1) = f(t)$ for any t, but also $a(t) = \int_s^{s+1} f(t)\,\mathrm{d}t$ for any s.

Example 2. $F = \Pi_1$, $V = [v_1, v_2]$ with $v_j := ()^{j-1}$, i.e., the power basis.
If we know that $f = a(1)v_1 + a(2)v_2$, then we know $f(0) = a(1)$, $f'(0) = a(2)$, but also $a(2) = (f(s) - f(t))/(s - t)$ for any $s \neq t$. Even $f(t) = a(1) + ta(2)$ is almost immediately available.

Example 3: the power form. $F = \Pi_{<n}$, $V = [v_1, \ldots, v_n]$ with $v_j := ()^{j-1}$, all j, i.e., the power basis.
If we know that

$$f = Va = a(1) + a(2)t + a(3)t^2 + \cdots + a(n)t^{n-1},$$

then we know immediately

$$f^{(j-1)}(0)/(j-1)! = a(j), \quad j = 1, \ldots, n.$$

We can find $f(t)$ pretty quickly by **Nested Multiplication**:

$$f(t) = \underbrace{a(1) + t(\underbrace{a(2) + \cdots + t(\underbrace{a(n-2) + t(\underbrace{a(n-1) + t\,\underbrace{a(n)}_{=:b(n)}}_{a(n-1)+tb(n)=:b(n-1)}}_{a(n-2)+tb(n-1)=:b(n-2)})\cdots}_{a(2)+tb(3)=:b(2)})}_{a(1)+tb(2)=:b(1)}.$$

We can find f' almost immediately:

$$f'(t) = a(2) + 2a(3)t + \cdots + (n-1)a(n)t^{n-2},$$

and also the indefinite integral $t \mapsto \int_0^t f(s)\,\mathrm{d}s$ is very easily obtainable. Contrast this with the next example.

Example 4: the Lagrange form. $F = \Pi_{<n}$, $V = [\ell_1, \ldots, \ell_n]$ with ℓ_j, $j = 1, \ldots, n$, the Lagrange polynomials for the points $\tau_1, \ldots, \tau_n$ as given in (4).

If we know that

$$f = a(1)\ell_1 + \cdots + a(n)\ell_n, \tag{6}$$

then we know immediately

$$f(\tau_j) = a(j), \quad j = 1, \ldots, n.$$

However, to obtain from this coordinate vector a the value of f at some point t other than $t = \tau_1, \ldots, \tau_n$ is a bit more involved. E.g., if we let

$$\omega(t) := (t - \tau_1) \cdots (t - \tau_n),$$

then

$$\ell_j(t) = \frac{\omega(t)}{(t - \tau_j)\omega'(\tau_j)},$$

with $\omega'(\tau_j)$ a convenient short-hand for $\prod_{i \neq j}(\tau_i - \tau_j)$, therefore

$$f(t) = \omega(t) \sum_{j=1}^{n} \frac{a(j)}{(t - \tau_j)\omega'(\tau_j)}.$$

For this, it seems more efficient to have in hand the coordinate vector

$$b := (a(j)/\omega'(\tau_j) : j = 1, \ldots, n),$$

corresponding to the basis $[\prod_{i \neq j}(t - \tau_i) : j = 1, \ldots, n]$, but then it is a bit more work to extract $f(\tau_i)$ from that coordinate vector b.

Things get worse if we are interested in f'. Straightforward termwise differentiation of (6) leads to $n(n-1)$ terms. And what about obtaining the indefinite integral $t \mapsto \int_0^t f(s)\,\mathrm{d}s$?

By comparing the Lagrange form with the power form, we begin to appreciate that there can be a real gain in something as basic as a change of basis. It also illustrates the importance of the Basic Question modified.

Actually, there is a subtle difference between the Basic Question and the Basic Question modified. The former starts off with some data map Λ' and asks what information about $f \in F$ is (readily) obtainable from $\Lambda' f$. The latter starts off with some basis W for F and wonders about information about $f \in F$ obtainable from its coordinates wrto W. For this to make sense, we must have available these coordinates or, at least, have a way to obtain these coordinates for any given $f \in F$.

But that is a problem we have already solved. After all, in order to deal with f computationally, we must have some unambiguous description for f available. The easiest such description to work with is $\Lambda' f$ for some data map that is 1-1 on F. Getting from this the coordinates of f wrto the given basis W is just a change of basis. In particular, if Λ' maps F *onto* $\mathbb{F}^n$, then it must be invertible and, as we saw before,

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

i.e., the coordinates of $f \in F$ wrto W are provided by the n-vector $a := (\Lambda' W)^{-1} \Lambda' f$, i.e., by the solution to the linear system

$$\Lambda' W ? = \Lambda' f.$$

This indicates the following very nice pay-off of this entire discussion.

Suppose that F is some linear subspace of some linear space X and that our data map Λ' is defined on all of X. Then, for arbitrary $g \in X$,

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

is the unique element in f that agrees with g 'at' Λ', i.e., for which $\Lambda' f = \Lambda' g$. We call this Pg the **interpolant from F to g wrto the data (map) Λ'.**

