In this section of the course, we will focus on problems of the form

$$y = Ax$$

y: $M \times 1$ vector; A: $M \times N$ matrix; x: $N \times 1$ vector

We observe y, and want to find x.

This is called a linear inverse problem, and is a fundamental concept in engineering and applied science.

We will be concerned with all of the cases $M > N$, $M = N$, $M < N$.

Conceptually, you can think of y as a set of measurements of x:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} \langle a_1, x \rangle \\ \langle a_2, x \rangle \\ \vdots \\ \langle a_M, x \rangle \end{bmatrix}$$

$a_i = i^{\text{th}}$ row of A

Example: Deconvolution

Given $y[n] = h[n] * x[n]$, find $x[n]$

Applications are manifold:

- image deblurring
- seismology
- channel equalization in digital communications

In matrix vector form we have

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[M-1] \end{bmatrix} = \begin{bmatrix} h[0] & 0 & 0 & \cdots & & \\ h[1] & h[0] & 0 & \cdots & & \\ h[2] & h[1] & h[0] & 0 & \cdots & \\ \vdots & & & & & \\ h[K] & h[K-1] & \cdots & & & \\ 0 & h[K] & \cdots & & & \\ \vdots & & & & & \\ 0 & & & & & h[K] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$$

Here we have $M > N$

What does the matrix look like for circular convolution?

Discretizing linear inverse problems

In many real-world applications, the signal or image we are measuring is a function of a continuous variable (or variables for images). Of course, if we are going to reconstruct the signal/image on a computer, our answer will ultimately be discrete. In this module, we discuss a general way to discretize linear inverse problems using a basis representation. We will start with the particular example of 2D tomography ("reconstruction from projections"), but the framework will be easy to generalize.

The Radon Transform

In the 2D tomographic reconstruction problem, the image $f(s,t)$ we wish to acquire is sampled using line integrals. We can parameterize a line $\vec{\ell}$ using an offset r and an angle θ as shown below: The line $\vec{\ell}$ is the set of points obeying a linear constraint:

$$\vec{\ell} = \{(s,t) \ : \ s\cos\theta + t\sin\theta = r\}$$

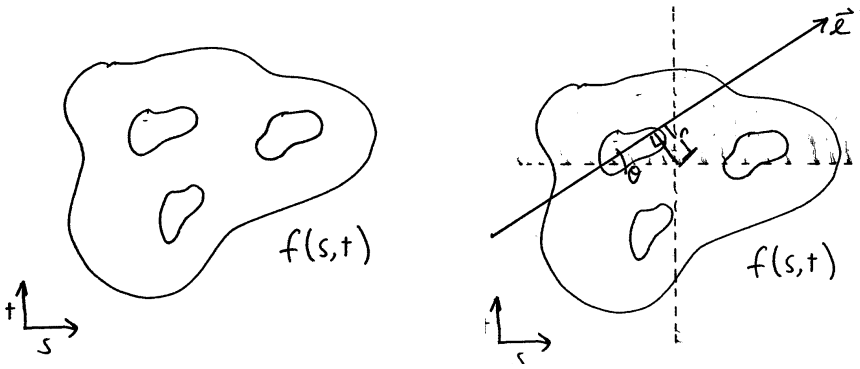
The integral of $f(s,t)$ along $\vec{\ell}$ is given by

$$R_{r,\theta}[f] = \begin{cases} \int f\left(\frac{r-t\sin\theta}{\cos\theta}, t\right) dt & |\theta| \leq \pi/4 \\ \int f\left(s, \frac{r-s\cos\theta}{\sin\theta}\right) ds & \pi/4 < |\theta| \leq \pi/2 \end{cases}$$

Of course, these expressions are equal to one another except when $\theta = 0, \pi/2$. Note also that the measurements are unique only over a range of π, as $R_{r,\theta+\pi}[f] = R_{-r,\theta}[f]$. It is sometimes convenient to write the line integral as a 2D integral of $f(s,t)$ against a *delta ridge*:

$$R_{r,\theta}[f] = \int\int f(s,t)\delta(s\cos\theta + t\sin\theta - r)\ ds\ dt, \qquad (1)$$

where $\delta(\cdot)$ is the Dirac delta function.



The collection of all such line integrals $\{R_{r,\theta}[x],\ \theta \in [0,\pi],\ r \in \mathbb{R}\}$ is called the *Radon transform* of $f(s,t)$. The radon transform is itself a continuous function of two variables. The figure below show an illustrative example: on the left, we see $R_{r,\theta}$ of a test image as a function of r for two different fixed values of θ. On the right is the collection $R_{r,\theta}$ as a function of both r and θ.

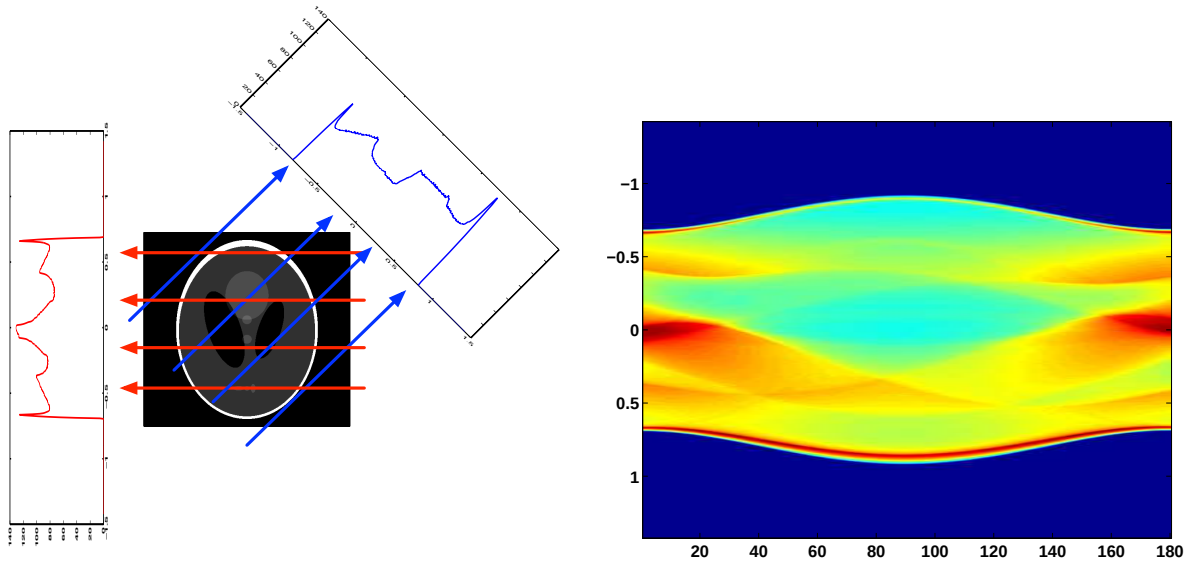


Figure 1: *Left: $R_{r,\pi/4}[f]$ and $R_{r,\pi}[f]$ as a function of r, where $f(s,t)$ is the Shepp-Logan phantom. Right: The Radon transform of the phantom. The rows are indexed by r and the columns by θ (in degrees).*

Reconstruction from a discrete set of line integrals

Given measurements y_m, $m = 1, \ldots, M$ corresponding to line integrals at different different offsets r_m and angles θ_m (i.e. a finite set of samples of the Radon transform), which have possibly been corrupted by noise, we would like to estimate the underlying image $f(s,t)$. If the measurements are dense in (r,θ) space, the natural approach to this problem is to use filtered backprojection. Our focus here will be setting this problem up as finite linear inverse problem

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \text{noise}, \quad \mathbf{y} \in \mathbb{R}^M, \mathbf{x} \in \mathbb{R}^N$$

so that it can be attacked with the general set of tools for solving such problems (e.g. least-squares).

We start by choosing a finite-dimensional space V in which to perform the reconstruction that comes equipped with a set of N basis vectors $\{\psi_\gamma(s,t)\}$. We will use the general index $\gamma \in \Gamma$ where Γ is a set of size N as, depending on the basis, it may be convenient to index the basis in different ways (i.e. by integers, pairs of integers over the same range, pairs of integers over different ranges, etc.).

For example, if $f(s,t)$ is non-zero only for $(s,t) \in [0,1]^2$, we might take our reconstruction space V to be the set of all "pixellated" images — images that are piecewise-constant on squares of side length $1/n$ for some integer n. A natural basis for this space is the set of indicator functions on these squares:

$$\psi_{j,k}(s,t) = \begin{cases} 1 & s \in [j/n, (j+1)/n],\ t \in [k/n, (k+1)/n] \\ 0 & \text{otherwise} \end{cases}$$

Using our general index notation, we can write any $f(s,t) \in V$ as

$$f(s,t) = \sum_{\gamma \in \Gamma} x_\gamma \psi_\gamma(s,t),$$

where $\Gamma = \{(j,k) \ : \ j,k = 0,1,\ldots,n-1\}$ with size $N = n^2$, and the $x_\gamma \in \mathbb{R}$ are the basis expansion coefficients, which are, in this case, the pixel values. (Another natural basis for V would be the two-dimensional Haar basis we encountered earlier). The point is that knowing the discrete set of coefficients x_γ is the same as knowing the continuous-space function $f(s,t)$.

We can also write the measurements of an $f(s,t) \in V$ in terms of the basis functions:

$$\begin{aligned} y_m &= R_{r_m,\theta_m}\left[\sum_{\gamma\in\Gamma} x_\gamma \psi_\gamma(s,t)\right] \\ &= \sum_{\gamma\in\Gamma} x_\gamma R_{r_m,\theta_m}\left[\psi_\gamma(s,t)\right] \qquad \text{(since } R_{r,\theta}[\cdot] \text{ is linear)} \\ &= \sum_{\gamma\in\Gamma} A_{m,\gamma} x_\gamma \qquad \text{where } A_{m,\gamma} = R_{r_m,\theta_m}\left[\psi_\gamma(s,t)\right], \end{aligned}$$

which can be written in more compact form as

$$\mathbf{y} = \mathbf{A}\mathbf{x}. \tag{2}$$

The entries of the $M \times N$ matrix $\mathbf{A}$ contain the results of each of the M measurements functionals $R_{r_m,\theta_m}[\cdot]$ applied to each of the N basis functions $\psi_\gamma(s,t)$, the N-vector $\mathbf{x}$ contains the expansion coefficients for $f(s,t)$ in the basis $\{\psi_\gamma\}$, and $\mathbf{y}$ contains the M measurements. This is illustrated in the figure below. As we can see, not too many of the $\vec{\ell}_m$ pass through a given pixel, meaning that the matrix $\mathbf{A}$ will be very sparsely populated.

Of course, the true underlying image will in general not lie in the chosen finite-dimensional subspace V. This means that even when there is no measurement noise, there will still be some inherent error in our calculations. But solving (2) will in some sense find the member of V that best explains the measurements that have been observed. If the true image can be closely approximated by a

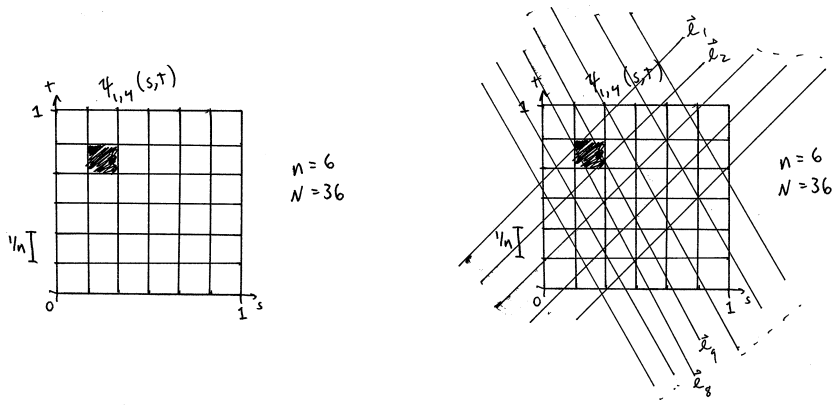


Figure 2: *Left: A sketch of one of the basis functions $\psi_\gamma(s,t)$ from the discussion above. Right: The entries of $\mathbf{A}$ in the column indexed by γ will be the result of measuring the basis function $\psi_\gamma(s,t)$: $A_{m,\gamma} = R_{r_m,\theta_m}[\psi_\gamma]$.*

member of V, then we will not lose much through this discretization. A major consideration in choosing the space V is how well we can use it to approximate images we expect to encounter.

General linear operators

The technique above can be very naturally generalized to different kinds of measurement operators that map signals of a continuous variable(s) into $\mathbb{R}^M$. In general, the measurements y_m will consist of inner products of $f(s,t)$ against different "test functions" $\phi_m(s,t)$:

$$y_m = \langle f, \phi_m \rangle = \int \int f(s,t)\phi_m(s,t)\, ds\, dt.$$

In the tomography example above, we took

$$\phi_m(s,t) = \delta(s\cos\theta_m + t\sin\theta_m - r_m)$$

(see (1) above). Just as before, for $f(s,t) \in V$ we can write

$$\begin{aligned} y_m &= \langle \sum_{\gamma\in\Gamma} x_\gamma \psi_\gamma, \phi_m \rangle \\ &= \sum_{\gamma\in\Gamma} x_\gamma \langle \psi_\gamma, \phi_m \rangle, \end{aligned}$$

and so

$$\mathbf{y} = \mathbf{A}\mathbf{x},$$

where the $M \times N$ matrix $\mathbf{A}$ has entries

$$A_{m,\gamma} = \langle \psi_\gamma, \phi_m \rangle = \int \int \psi_\gamma(s,t)\phi_m(s,t)\, ds\, dt.$$

It is worth noting that since the entries of $\mathbf{A}$ do not depend on f, they can be pre-computed.

In problems such as these

① M & N can be huge, creating all kinds of numerical challenges

② The data are usually noisy.
That is, we actually observe

$$y = Ax + e$$

(e → error)

So whatever we do, we want to be stable in the presence of noise.

③ Even if $M = N$ (or $M > N$), the problem can be severely ill-posed
(A^{-1} cannot be calculated accurately)

The key concept we will use to understand these problems is the singular value decomposition.

There are two basic cases:

- underdetermined: $M < N$

  → A is "short and fat"

$$[\,y\,] = [\quad A \quad]\,[\,x\,]$$

→ we are trying to estimate a lot of unknowns from a few observations

This problem is always ill-posed. There will be many possible x from which we need to choose.

- overdetermined: $M > N$

  → A is "tall & thin"

$$[\,y\,] = [\,A\,]\,[\,x\,]$$

→ we have a lot of observations to estimate a small number of unknowns.

→ there will be some "averaging" (and hence noise reduction) in our estimate of X

→ In general, as M gets bigger, we will be more & more robust against noise.

It can often happen that even in the overdetermined (or $M=N$) case, the problem will be ill-posed

i.e. just because A is square doesn't mean it's invertible.

Again, the effectiveness with which we can solve these problems depends on the <u>singular values</u> of A.

Eigenvalue decompositions of Symmetric Matrices

In this section, A is an $N \times N$ matrix (square) that is symmetric or Hermitian.

Definition: If A is real valued, then we call A <u>symmetric</u> if $A^T = A$

Example:

$$A = \begin{bmatrix} 1 & 3 & 7 \\ 3 & -5 & -2 \\ 7 & -2 & 6 \end{bmatrix}$$

$A_{ij} = A_{ji}$

Definition: If A is complex valued, then we call A <u>Hermitian</u> if $A^H = A$.

Recall that to compute A^H, we exchange rows & columns then conjugate all the entries

Example:

$$A = \begin{bmatrix} 1 & 3+j2 & 1-j3 \\ 3-j2 & -5 & 2+j4 \\ 1+j3 & 2-j4 & 6 \end{bmatrix}$$

$A_{ij} = \overline{A_{ji}}$

Of course, if A is real valued, then $A^H = A^T$.

In the end, we will be interested in the general case of non-symmetric and even non-square matrices, but symmetric/Hermitian matrices will be the starting point of our study — they play a fundamental role in "least-squares" and our analysis will prove <u>very useful</u> later on.

Like almost everywhere in applied linear algebra, the eigenvalues & eigenvectors of a matrix will come up all over in our conversation.

Definition: An eigenvector of an $N \times N$ matrix A is a vector v such that

$$Av = \lambda v$$

for some $\lambda \in \mathbb{C}$. The scalar λ is called the eigenvalue associated with v.

A is called diagonalizable if it has N linearly independent eigenvectors $v_1, \dots, v_N$.

In this case, we can write

$$Av_1 = \lambda_1 v_1$$
$$Av_2 = \lambda_2 v_2$$
$$\vdots$$
$$Av_N = \lambda_N v_N$$

as

$$A\begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_N \\ | & | & & | \end{bmatrix} = \underbrace{\begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_N \\ | & | & & | \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix}}_{\Lambda}$$

(Diagonal matrix)

i.e.

$$AV = V\Lambda$$

$$\Rightarrow \quad A = V\Lambda V^{-1}$$

(V^{-1} exists since $v_1, \dots, v_N$ are linearly independent)

Recall that even if A is real-valued, in general its eigenvectors & eigenvalues can be complex valued.

But if A is symmetric/Hermitian, it also has four very special properties:

Property 1: For all complex-valued vectors x

$x^H A x$ is a real number.

The argument for this is simple. Since $x^H A x$ is just a scalar, its Hermitian transpose is just its conjugate. We have

$$\overline{x^H A x} = (x^H A x)^H = x^H A^H x^{HH} = x^H A x$$

(since $A^H = A$ and $x^{HH} = x$)

Since $x^H A x$ is equal to its conjugate, it must be real.

Property 2: Every eigenvalue of A is real-valued.

Given Prop. 1, the argument here is also simple.

If v is an eigenvector of A, then

(*) $$Av = \lambda v$$

and

$$v^H A v = \lambda v^H v$$

Since $v^H v = \|v\|_2^2$, we know $v^H v$ on the right hand side is real. Property 1 tells us the left hand side is real. Thus λ must be real.

Notice that re-examining (*) also gives us

Property 2a: If A is real-valued (and $A = A^T$) then all eigenvectors of A are real-valued.

Property 3: Eigenvectors of A with different eigenvalues are orthogonal.

Let v_1, v_2 be eigenvectors of A with

$$Av_1 = \lambda_1 v_1 \qquad \lambda_1 \neq \lambda_2$$
$$Av_2 = \lambda_2 v_1$$

Then

$$V_2^H A V_1 = \lambda_1 V_2^H V_1$$

$$V_1^H A V_2 = \lambda_2 V_1^H V_2 \quad \text{(take conjugate of both sides)}$$

$$V_2^H A V_1 = \lambda_2 V_2^H V_1$$

So

$$\lambda_1 V_2^H V_1 = \lambda_2 V_2^H V_1$$

If $\lambda_1 \neq \lambda_2$, this is only possible when $V_2^H V_1 = 0$.

If there are two linearly independent vectors V_1, V_2 with the same eigenvalue

$$AV_1 = \lambda V_1$$
$$AV_2 = \lambda V_2$$

then they need not be orthogonal. In this case everything in $\text{span}\{V_1, V_2\}$ is also an eigenvector, as

$$\begin{aligned} A(\alpha V_1 + \beta V_2) &= \alpha A V_1 + \beta A V_2 \\ &= \alpha \lambda V_1 + \beta \lambda V_2 \\ &= \lambda(\alpha V_1 + \beta V_2) \end{aligned}$$

Thus there exist two orthogonal vectors V_1' & V_2' (which you can find by applying Gram-Schmidt to $\{V_1, V_2\}$)

whose span it the same as span$\{v_1, v_2\}$ and have

$$Av_1' = \lambda v_1'$$
$$Av_2' = \lambda v_2' \quad , \quad v_1'^H v_2' = 0$$

It should be clear that a similar statement is true if there are p linearly independent eigenvectors which share the same eigenvalue.

Property 4: We can write

$$A = V\Lambda V^H \quad \text{(when } A \text{ is complex-valued Hermitian)}$$

or

$$A = V\Lambda V^T \quad \text{(when } A \text{ is real-valued symmetric)}$$

where V is an orthonormal matrix

$$V^H V = VV^H = I \quad \text{or} \quad V^T V = VV^T = I$$

and Λ is diagonal:

$$\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix}$$ ← eigenvalues of A

If A has N distinct eigenvalues, this follows directly from Property 3, as there will be N $\perp$ vectors $v_1, \ldots, v_N$ with

$$Av_1 = \lambda_1 v_1$$
$$Av_2 = \lambda_2 v_2$$
$$\vdots$$
$$Av_N = \lambda_N v_N$$

$$v_i^H v_j = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$$

which we can write more compactly as

$$A\left[v_1 \middle| v_2 \middle| \cdots \middle| v_N\right] = \left[v_1 \middle| v_2 \middle| \cdots \middle| v_N\right] \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix}$$

$$AV = V\Lambda$$

$$\Rightarrow A = V\Lambda V^H \quad \text{since } VV^H = I.$$

For the more general case, where there are repeated eigenvalues, it takes a little more work to show that A has N linearly independent (and also orthogonal) eigenvectors — work which we will skip.

In either case, for symmetric (Hermitian) A we can write

$$A = V\Lambda V^T \quad (\text{or } A = V\Lambda V^H).$$

10h

One way to think about V is as a transform which greatly simplifies the action of A.

$$Ax = V\Lambda V^T = (\text{inverse } V \text{ transform}) \cdot (\text{pointwise multiplication}) \cdot (V \text{ transform}) \cdot x$$

From here on out we will focus on real-valued A, but almost everything we say can be extended in an obvious way to complex valued A.

Def: We will call symmetric A positive definite if $\lambda_n > 0$ for $n = 1, \ldots, N$

We will abbreviate this as sym+def.

We call symmetric A non-negative definite if $\lambda_n \geq 0$.

sym+def A are invertible, and obey

$$x^T A x > 0 \quad \text{for all } x \in \mathbb{R}^N$$

Example:

$$A = \begin{bmatrix} 3/2 & 1/2 \\ 1/2 & 3/2 \end{bmatrix}$$

Check that $A = V\Lambda V^T$

where $V = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$ and $\Lambda = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

Sketch the action of A on $x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Calculating $\alpha = V^T x = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ 1 \end{bmatrix}$

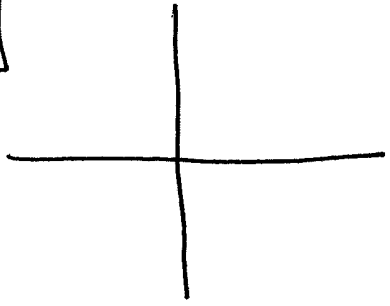
tells us that

$$x = \frac{-1}{\sqrt{2}} v_1 + \frac{1}{\sqrt{2}} v_2$$

Stretch the v_1 component by 1 & the v_2 component by 2 to get

$$Ax = \frac{-1}{\sqrt{2}} v_1 + \sqrt{2}\, v_2$$

$$= \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ 3/2 \end{bmatrix}$$



A convenient way to write A is

$$A = \sum_{k=1}^{N} \lambda_k v_k v_k^T$$

In fact, any function of A can be written in this manner.

Sylvester's Matrix Theorem

If A is symmetric w/ eigen-decomposition $\{\lambda_k, v_k\}$, then for any function f

$$f(A) = \sum_{k=1}^{N} f(\lambda_k) v_k v_k^T$$

Special cases of interest:

$$A = \sum_{k=1}^{N} \lambda_k v_k v_k^T = V \Lambda V^T$$

$$A^{-1} = \sum_{k=1}^{N} \frac{1}{\lambda_k} v_k v_k^T = V \Lambda^{-1} V^T$$

$$A^{1/2} = \sum_{k=1}^{N} \sqrt{\lambda_k}\, v_k v_k^T = V \Lambda^{1/2} V^T$$

$$e^A = \sum_{k=1}^{N} e^{\lambda_k} v_k v_k^T$$

There is also a variational expression for the largest and smallest eigenvalues for sym+def A

$$\max_{x \in \mathbb{R}^N} \frac{x^T A x}{\|x\|_2^2} = \max_{\substack{x \in \mathbb{R}^n \\ \|x\|_2 = 1}} x^T A x = \lambda_1 \quad (\text{when } x \propto v_1)$$

$$\min_{x \in \mathbb{R}^N} \frac{x^T A x}{\|x\|_2^2} = \min_{\substack{x \in \mathbb{R}^n \\ \|x\|_2 = 1}} x^T A x = \lambda_N \quad (\text{when } x \propto v_N)$$

Solving systems of symmetric +def equations

Again, we are interested in

$$y = A x$$

(y: $N \times 1$; A: $N \times N$ sym+def; x: $N \times 1$)

To find

$$x = A^{-1} y$$

we use

$$x = V \Lambda^{-1} V^T y$$

$$= (\text{inverse } V\text{-transform}) \cdot (\text{diagonal weighting}) \cdot (V \text{ transform}) \cdot y$$

(point-wise mult.)

We can also write

$$x = A^{-1}y = \sum_{k=1}^{N} \frac{1}{\lambda_k} \langle y, v_k \rangle \cdot v_k$$

$\Rightarrow$ given the eigenvalue decomposition, solving

$$y = Ax$$

is easy (as easy as matrix-vector multiplication)

Now suppose there is some observation error:

$$y = Ax + \underbrace{e}_{\text{some unknown error } e \in \mathbb{R}^N}$$

We reconstruct as before, by applying A^{-1} to y:

$$\tilde{x} = A^{-1}y = A^{-1}(Ax + e)$$

$$= x + \underbrace{A^{-1}e}_{\text{reconstruction error } \Sigma_x}$$

Questions:

① What is the largest eigenvalue of A^{-1}?

(2) What is an upper bound on the reconstruction error?

$$\mathcal{E}_x = \tilde{x} - x$$

$$\|\mathcal{E}_x\|_2^2 = \|A^{-1}e\|_2^2 = e^T (A^{-1})^T A^{-1} e$$

$$\leq$$

(3) What is the smallest eigenvalue of A^{-1}?

(4) What is a lower bound on the reconstruction error?

In the end, we have

$$\frac{1}{\lambda_1^2}\|e\|_2^2 \leq \|\mathcal{E}_x\|_2^2 \leq \frac{1}{\lambda_N^2}\|e\|_2^2$$

Average Reconstruction Error

The maximum & minimum eigenvalues give us the "best case" & "worst case" errors

$$\tilde{x} = A^{-1}y = A^{-1}(Ax + e)$$

$$= x + A^{-1}e \quad \Rightarrow \quad \tilde{x} - x = V\Lambda^{-1}V^T e$$

$$\underbrace{\frac{1}{\lambda_1^2}\cdot \|e\|_2^2}_{\text{best-case error, occurs when } e \in \text{span}\{v_1\}} \leq \|\tilde{x} - x\|_2^2 \leq \underbrace{\frac{1}{\lambda_N^2}\|e\|_2^2}_{\text{worst-case error, occurs when } e \in \text{span}\{v_N\}}$$

We can also get an "average case" reconstruction error when e is <u>generic</u> (i.e. random)

Our model for random noise is that the entries of e are iid Gaussian:

$$e[n] \sim \text{Normal}(0, \sigma^2) \quad n = 1, \ldots, N$$

$$E[e[n]e[l]] = \begin{cases} \sigma^2 & n = l \\ 0 & n \neq l \end{cases}$$

↑ uncorrelated + Gaussian ⇒ independent

Q: What is the mean-energy of e?

$$E\|e\|_2^2 = E\left[\sum_{n=1}^{N} |e[n]|^2\right]$$

$$= \sum_{n=1}^{N} E\left[|e[n]|^2\right]$$

$$= \sum_{n=1}^{N} \sigma^2$$

$$= N\sigma^2$$

We are really interested in

$$E\|\tilde{x} - x\|_2^2 = E\|A^{-1}e\|_2^2$$

↑ average reconstruction error

We can get a nice expression for this using the eigenvalue decomposition

$$A = V\Lambda V^T \iff A^{-1} = V\Lambda^{-1}V^T$$

Just plug-and-chug:

$$E\|A^{-1}e\|_2^2 = E\|V\Lambda^{-1}V^T e\|_2^2$$

$$= E\langle V\Lambda^{-1}V^T e, V\Lambda^{-1}V^T e\rangle$$

$$= E\langle \Lambda^{-1}V^T e, V^T V\Lambda^{-1}V^T e\rangle$$

$$= E\langle \Lambda^{-1}V^T e, \Lambda^{-1}V^T e\rangle \quad \text{since } V^T V = I$$

$$= E\left[\sum_{n=1}^{N} \left|\tfrac{1}{\lambda_n}\langle v_n, e\rangle\right|^2\right]$$

$$= \sum_{n=1}^{N} \frac{1}{\lambda_n^2} \cdot E\left[|\langle v_n, e\rangle|^2\right]$$

Note that

$$|\langle v_n, e\rangle|^2 = \left(\sum_{m=1}^{N} v_n[m]\cdot e[m]\right)^2$$

$$= \sum_{m=1}^{N}\sum_{\ell=1}^{N} v_n[m]\, v_n[\ell]\, e[m]\, e[\ell]$$

So

$$E\left[|\langle v_n, e\rangle|^2\right] = \sum_{m=1}^{N}\sum_{\ell=1}^{N} v_n[m]\,v_n[\ell]\cdot \underbrace{E\left[e[m]\,e[\ell]\right]}_{=0 \text{ unless } m=\ell}$$

$$= \sum_{m=1}^{N} |v_n[m]|^2 \cdot \sigma^2$$

$$= \sigma^2 \quad \text{since} \quad \|v_n\|_2^2 = 1$$

Thus

$$E\|A^{-1}e\|_2^2 = \sigma^2 \cdot \sum_{n=1}^{N} 1/\lambda_n^2$$

$$= \underbrace{N\sigma^2}_{E\|e\|_2^2} \cdot \underbrace{\left(\frac{1}{N}\sum_{n=1}^{N} 1/\lambda_n^2\right)}_{\text{average of } 1/\text{eigenvalues}^2}$$

The intuition here is that a random vector is spread out more or less equally over the basis $\{v_i\}$ rather than being concentrated in Span $\{v_1\}$ or span $\{v_n\}$.

16d

Summary of symtdef reconstruction:

Observe

$$y = Ax + e$$

$y \in \mathbb{R}^N$, A: $N \times N$ symmetric positive definite, $x \in \mathbb{R}^N$, error $e \in \mathbb{R}^N$

Reconstruct

$$\tilde{x} = A^{-1} y = x + A^{-1} e$$

Best/Worst case reconstruction errors:

$$\frac{1}{\lambda_1^2} \cdot \|e\|_2^2 \leq \|\tilde{x} - x\|_2^2 \leq \frac{1}{\lambda_N^2} \cdot \|e\|_2^2$$

Average reconstruction error:

$$e[n] \sim \text{Normal}(0, \sigma^2),\ e[n] \text{ iid}$$

$$E\|\tilde{x} - x\|_2^2 = \sigma^2 \cdot \sum_{n=1}^{N} 1/\lambda_n^2$$

16e

The Singular Value Decomposition (SVD)

We are interested in more than just sym+def matrices. But the eigenvalue-decompositions discussed above will play a major role in solving

$$\underset{M\times 1}{y} = \underset{M\times N}{A}\ \underset{N\times 1}{x}$$

The SVD expresses a $M\times N$ matrix A in a similar $\perp$-diagonal-$\perp$ format

$$\underset{M\times N}{A} = \underset{M\times p}{U}\ \underset{p\times p}{\Sigma}\ \underset{p\times N}{V^T}$$

p = rank of A (= # linearly indep. columns if $M \geq N$, or # linearly indep. rows if $M < N$)

If A is "full rank" then

$$p = \min(M, N)$$

The columns of U are $\perp$, and the rows of V^T (columns of V) are $\perp$:

$$U^T U = I \quad \text{(but in general } UU^T \neq I\text{)}$$
$$V^T V = I \quad \text{(but in general } VV^T \neq I\text{)}$$

Σ is a diagonal matrix

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots \\ 0 & \sigma_2 & \ddots \\ \vdots & & \sigma_p \end{bmatrix}$$

with $\sigma_k \geq 0$ (see below).

We can also write A as

$$A = \sum_{k=1}^{p} \sigma_k \underbrace{u_k}_{m\times 1} \underbrace{v_k^T}_{1\times n}$$

- The vectors $u_1, \ldots, u_p$ are the eigenvectors of AA^T that have non-zero eigenvalues

$$AA^T = U\Sigma \underbrace{V^T V}_{=I} \Sigma U^T = U\Sigma^2 U^T$$

$= $ eigen-decomposition of AA^T $(\lambda_k = \sigma_k^2)$
(with the columns of U removed that have a corresponding eigenvalue of zero)

$u_1, \ldots, u_p \in \mathbb{R}^m$

- The vectors $v_1, \ldots, v_p \in \mathbb{R}^N$ are the eigenvectors of A^TA that have non-zero eigenvalues

$$A^TA = V\Sigma U^TU\Sigma V^T = V\Sigma^2V^T$$

= eigendecomposition of A^TA
(with the columns of V removed that have a zero eigenvalue)

- The number of non-zero eigenvalues will be the same for AA^T and A^TA (and in fact, they are the same)

- The eigenvalues of AA^T (and A^TA) are the <u>squares</u> of the singular values of A.

The Least-Squares Problem

We will use the SVD to solve

$$\underset{M\times 1}{y} = \underset{M\times N}{A}\,\underset{N\times 1}{x}$$

in such a way that

1. When there is a unique solution, we find it.
2. When there is no solution, we return something reasonable.
3. When there are many (infinite) solutions, we choose one in a "smart" way.

Our framework revolves around minimizing the length (norm) of the residual

$$r = y - Ax$$

First, note that we can write any $x \in \mathbb{R}^N$ as

$$x = V\alpha + V_0\alpha_0$$

for $\alpha = V^T x$, $\alpha_0 = V_0^T x$

The V and V_0 come from the SVD of A:

$$A = U \Sigma V^T$$

- $A^T A v_k = \sigma_k^2 v_k$ with $k = 1, \ldots, p$

  $A^T A v_k = 0$ $k = p+1, \ldots, N$

  $V = [v_1 | \cdots | v_p]$ $V_0 = [v_{p+1} | \cdots | v_N]$

  $= \perp$-basis for the Null space of V^T (non-unique)

- $A A^T u_k = \sigma_k^2 u_k$ $k = 1, \ldots, p$

  $A A^T u_k = 0$ $k = p+1, \ldots, M$

  $U = [u_1 | \cdots | u_p]$ $U_0 = [u_{p+1} | \cdots | u_N]$

  $= \perp$-basis for the Null space of U^T (non-unique)

So finding

$$X = V\alpha + V_0 \alpha_0$$

is the same as finding the pair α, α_0

Similarly. $y = U\beta + V_0 \beta_0$

with $\beta = U^T y$ $\beta_0 = U_0^T y$

So the residual $r = y - Ax$ becomes

$$r = U\beta + U_0\beta_0 - U\Sigma V^T(V\alpha + V_0\alpha_0)$$

$$= U\beta + U_0\beta_0 - U\Sigma\alpha$$

since $V^TV = I$, $V^TV_0 = 0$

$$= U_0\beta_0 + U(\beta - \Sigma\alpha)$$

We want to choose α to minimize $\|r\|_2^2$

$$\|r\|_2^2 = \langle U_0\beta_0 + U(\beta - \Sigma\alpha), U_0\beta_0 + U(\beta - \Sigma\alpha)\rangle$$

$$= \langle U_0\beta_0, U_0\beta_0\rangle - 2\langle U_0\beta_0, U(\beta - \Sigma\alpha)\rangle + \langle U(\beta - \Sigma\alpha), U(\beta - \Sigma\alpha)\rangle$$

$$= \|\beta_0\|_2^2 + \|\beta - \Sigma\alpha\|_2^2$$

since $U_0^TU_0 = I$, $U^TU = I$, $U^TU_0 = 0$

β_0 is fixed (it's simply $\beta_0 = U_0^Ty$), so we choose α so that the second term is minimized

Clearly, $\hat{\alpha} = \Sigma^{-1}\beta$ does the trick.

And so finally

$$\boxed{\hat{x} = V\hat{\alpha} = V\Sigma^{-1}\beta = V\Sigma^{-1}U^T y}$$

Properties of the solution:

① When $y \in \text{span}\{u_1, \ldots, u_M\}$
we will have $\beta_0 = U_0^T y = \underline{0}$
and so the residual $r = \underline{0}$
(there is at least 1 exact solution)

② Note that if $p < N$, the solution is not unique.
In this case, V_0 has at least one column.
Any part of x in the span of V_0 is not seen by A, since

$$AV_0\alpha_0 = U\Sigma V^T V_0 \alpha_0 = 0 \quad \text{since } V^T V_0 = 0$$

As such,

$$x' = \hat{x} + V_0\alpha_0$$

for <u>any</u> $\alpha_0 \in \mathbb{R}^{N-p}$ will have the same residual (since $Ax' = A\hat{x}$)

In this case, we usually choose the x' with smallest norm

Since

$$\|x'\|_2^2 = \|V\Sigma^{-1}U^T y\|_2^2 + \|\alpha_0\|_2^2$$

(check at home), this simply means set $\alpha_0 = \underline{0}$.

So $\hat{x} = V\Sigma^{-1}U^T y$ has our desired properties:

1. When $y = Ax$ has a unique solution, it must be $\hat{x}$.

2. When an exact solution is not possible, it is the minimizer to

$$\min_{x \in \mathbb{R}^n} \|y - Ax\|_2^2$$

3. When there are many (infinite) solutions it the one with smallest norm. In this case it is the minimizer to

$$\min_{x \in \mathbb{R}^n} \|x\|_2^2 \quad \text{s.t. } Ax = y$$

The matrix

$$A^\dagger = V\Sigma^{-1}U^T$$

is called the pseudo-inverse of A.

(More on this next time.)