IAM Data Science Pre-Workshop
Linear Algebra Lecture #3

1 Eigenanalysis & diagonalization.

$\mathbb{A}$ $n \times n$ (square).

If $\underline{x} \neq \underline{0}$ and

$\mathbb{A}\underline{x} = \lambda \underline{x}$ ← if $\underline{x}$ is an eigenvector then so is $\alpha \underline{x}$, $\alpha \neq 0$ (only direction matters).

Then $\underline{x}$ is said to be an eigenvector of $\mathbb{A}$ and λ its corresponding eigenvalue.

Notes:
- $\underline{x} \neq \underline{0}$ but λ can be zero (in this case A is not invertible).
- λ, $\underline{x}$ can be complex. If $\mathbb{A}$ is real then complex eigenvalues and vectors come in complex conjugate pairs.
- Multiplication by $\mathbb{A}$ changes the size but not the direction of $\underline{x}$. Multiplication by any power of $\mathbb{A}$ is just scalar multiplication

$$\mathbb{A}^{l}\underline{x} = \lambda^{l}\underline{x}.$$

not true in general.

If $\mathbb{A}$ has n eigenvectors $\mathcal{C} = \{\underline{v}_1, \cdots \underline{v}_n\}$ such that the span of $\mathcal{C}$ is $\mathbb{R}^n$ (we say $\mathcal{C}$ is a basis for $\mathbb{R}^n$) then any $\underline{x}$ can be written (uniquely) as a linear combination of the eigenvectors.

$$\underline{x} = c_1 \underline{v_1} + c_2 \underline{v_2} + \cdots c_n \underline{v_n}$$

for some $\underline{c}$ that solves.

$$\underbrace{\begin{bmatrix} | & | & & | \\ \underline{v_1} & \underline{v_2} & \cdots & \underline{v_n} \\ | & | & & | \end{bmatrix}}_{\mathbb{V}} \underline{c} = \underline{x}.$$

or (symbolically) $\underline{c} = \mathbb{V}^{-1} \underline{x}$.

Then $\mathbb{A}^l \underline{x} = c_1 \lambda_1^l \underline{v_1} + c_2 \lambda_2^l \underline{v_2} + \cdots c_n \lambda_n^l \underline{v_n}$. (1)

Consider $l = 1$, we have

$$\mathbb{A}\,\underline{x} = \mathbb{V} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \underline{c}$$

$$= \mathbb{V} \mathbb{D} \mathbb{V}^{-1} \underline{x}$$

true for every $\underline{x}$, so $\mathbb{A} = \mathbb{V}\mathbb{D}\mathbb{V}^{-1}$.

and $\mathbb{A}^l = \mathbb{V} \mathbb{D}^l \mathbb{V}^{-1}$

Finding $\mathbb{A}^l \underline{x}$ using this formula gives insight and may be computationally cheaper if l is large.

→ computational examples. Obvious application is to Markov chains.

② Eigenanalysis for Symmetric Matrices.

If $\mathbb{A}$ is symmetric - that is $\mathbb{A} = \mathbb{A}^T$, (or $\mathbb{A} = \mathbb{A}^*$ if $\mathbb{A}$ is complex).

$\mathbb{A}^*$: conjugate transpose

Then $\mathbb{A}$ has only real eigenvalues, there is a basis $\mathcal{C}$ of eigenvectors, and $\mathcal{C}$ can be chosen so that the vectors are orthogonal to each other and of unit length.

$$\mathbb{V} = \begin{bmatrix} | & | & & | \\ \underline{V_1} & \underline{V_2} & \cdots & \underline{V_n} \\ | & | & & | \end{bmatrix} \qquad \underline{V_i} \cdot \underline{V_j} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j. \end{cases}$$

so $\mathbb{V}^T \mathbb{V} = \mathbb{I}$, and $\mathbb{V}^T = \mathbb{V}^{-1}$

If $\mathbb{A}$ has only a few (β) "large" eigenvalues, corresponding to $\lambda_1, \cdots \lambda_\beta$, then

$$\mathbb{A}^\ell \underline{x} \approx C_1 \lambda_1^\ell + C_2 \lambda_2^\ell + \cdots C_\beta \lambda_\beta^\ell$$

and $C_j = \underline{x} \cdot \underline{V_j}$

This is starting to look useful.

If $\mathbb{A}$ is invertible (no zero eigen values) then

$$\underbrace{\mathbb{A}^{-1} \underline{b}} = \mathbb{V} \mathbb{D}^{-1} \mathbb{V}^T$$

$\mathbb{D}^{-1}$: $\operatorname{diag}(\lambda_1^{-1}, \lambda_2^{-1}, \cdots \lambda_n^{-1})$.

shorthand for $\underline{x}$ that solves $\mathbb{A}\underline{x} = \underline{b}$.

condition number in $\|\cdot\|_2$ is the ratio of largest to smallest absolute value eigenvalues.

→ computational examples of eigenanalysis of symmetric matrices.

Consider the Finite Difference matrix D_2 in the periodic setting, N grid points.

The discrete Fourier vectors $\{\underline{F_\alpha}\}$ $\alpha = 0, \dots N-1$ are a basis for $\mathbb{R}^n$.

$$[\underline{F}_\alpha]_j = e^{2\pi i j \alpha / N}.$$

They are orthogonal in the complex sense

← conjugate

$$\sum_{j=1}^{n} F_{\alpha j} F^*_{\beta j} = \begin{cases} 0 & \text{if } \alpha \neq \beta \\ 1 & \text{if } \alpha = \beta. \end{cases}$$

They are not quite normalized, since they all have length N, not 1, but

$$F^{-1} = \frac{1}{N} F^*$$

↑ matrix of Fourier vectors in column.

The discrete Fourier vectors are the eigenvectors of D_2 (von Neumann analysis).

$$[D_2 \underline{F_\alpha}]_j = \frac{1}{h^2}\left(e^{2\pi i (j+1)\alpha/N} - 2e^{2\pi i j \alpha/N} + e^{2\pi i (j-1)\alpha/N}\right)$$

$$= \frac{1}{h^2}\left(e^{2\pi i \alpha/N} + e^{-2\pi i \alpha/N} - 2\right)[F_\alpha]_j$$

↑ eigenvalue (symbol) $\frac{2}{h^2}(\cos(\alpha h) - 1)$

This leads to a fast solver for the 1D problem discretization

$$-D_2 \underline{U} + \underline{U} = \underline{b}$$

$A = -D_2 + \mathbb{I}$ condition number $O(N^2)$.

$$\Rightarrow \quad \hat{\underline{b}} = F^{-1} \underline{b}.$$

then diagonal $\hat{U}_\alpha = \dfrac{\hat{b}_\alpha}{1 + \frac{2}{h^2}(1 - \cos(\alpha h))}$

then $\underline{U} = F \hat{\underline{U}}$.

possibly confusing.

<u>Notes</u>:
- Multiplication by F^{-1} is actually the Fourier transform, F the inverse transform
- where you put the $\frac{1}{N}$ and whether the fourier vectors have $\pm i$ is arbitrary.
- Multiplication by F and F^{-1} only takes $O(N \log N)$ operations, not $O(N^2)$, using the Fast Fourier Transform.
- Even so, this is not competitive in 1D (direct solver is great) but is great for higher dimensions.

→ computational implementation.

③ Singular Value Decomposition (SVD).

A $m \times n$

Consider $m \gg n$ for the next topic.

A :  A^{+} : 

assume A has full rank n.

$A^T A$ small matrix $n \times n$, symmetric, rank n, n positive eigenvalues $\lambda_1^2, \cdots \lambda_n^2$, unitary matrix V of eigenvectors ($n \times n$).

$A A^T$ large matrix $n \times n$, symmetric, rank n, same n positive eigenvalues $\lambda_1^2, \cdots \lambda_n^2$, $m-n$ zero eigenvalues, unitary matrix U of eigenvectors ($m \times m$).

SVD: $\underset{m \times n}{A} = \underset{m \times m}{U} \; \underset{m \times n}{D} \; \underset{n \times n}{V^T}$

$$\begin{array}{cc} & n \\ n & \left[\begin{array}{c} \ddots \\ \hline 0 \end{array}\right] \end{array} \leftarrow \mathrm{diag}(\lambda_1, \cdots \lambda_n).$$

If $\mathbb{A}$ is SPD, the SVD is just diagonalization.

If $\mathbb{A}$ is symmetric but with eigenvalues of both signs, the SVD is just diagonalization but with some sign changes in $\mathbb{U}$ & $\mathbb{V}$.

→ Computational examples.

There is a connection to least squares.

If we were considering

$$\mathbb{A}\underline{x} = \underline{b}.$$

and looking for the least squares solution, it could be found by

$$\underline{x} = \mathbb{V}\,\mathbb{D}^{+}\,\mathbb{U}^{T}$$

better conditioned than the formula we had before.

↑ Take the reciprocal of the nonzero entries of $\mathbb{D}$ & transpose.

Note: If $m<n$ A: ▭ This same formula gives a different kind of least squares solution: the one that has the smallest norm $\|x\|_2$.

(5) PCA (Principle Component Analysis)

Consider m data points in $\mathbb{R}^n$, $m \gg n$, $\{\underline{x}_i\}$ row vectors.

put them in an $m \times n$ matrix $\mathbb{X}$

Later, we'll need some of the eigenvectors of $X^T X$, the small $n \times n$ matrix, only the ones corresponding to the "large" eigenvalues. ← The $\mathbb{V}$ from SVD.

Want to represent these data with compression:

$$\underline{x}_i \approx \mathbb{D}\, \underline{c}_i \qquad \mathbb{D} \text{ for "decoder".}$$

↑

$n \times l$ matrix to be determined

$l << n$.

orthonormal columns, full rank l.

two goals: (1) compression of the data, with some loss of fidelity.

(2) The low dimensional $\underline{c}$ values can be used for classification of new $\underline{x}$

(A) Suppose $\mathbb{D}$ is given, want to choose $\underline{c}(\underline{x})$ to minimize

$$\| \underline{x} - \mathbb{D}\underline{c} \|^2$$

least squares problem!

$$\underbrace{\mathbb{D}^T \mathbb{D}}_{\mathbb{I}_l}\, \underline{c} = \mathbb{D}^T \underline{x}.$$

so $\underline{c} = \mathbb{D}^T \underline{x}$ easy!

(5) Now minimize the encoding error by choice of $\mathbb{D}$:

encoding error

$$\mathbb{D}\underline{c} - \underline{x} = \mathbb{D}\mathbb{D}^T\underline{x} - \underline{x} = (\mathbb{D}\mathbb{D}^T - I)\,\underline{x}.$$

minimize

$$\sum_i \left\| (\mathbb{D}\mathbb{D}^T - \mathbb{I}_n)\, \underline{x}_i \right\|^2$$

$n \times n$ but rank $l \ll n$.

subject to $\mathbb{D}^T\mathbb{D} = \mathbb{I}_l$.

This is a kind of constrained least squares problem.

Theory: take the columns of $\mathbb{D}$ to be the l eigenvectors of $\mathbb{X}^T\mathbb{X}$ with the largest eigenvalues.

(6) Suggested problems.

#1. Implement a spectral discretization of the 2D elliptic problem in lecture #1. Use FFT to make a fast solver.

#2. Implement a spectral discretization of

$$-(a(x)u')' + u = f(x)$$

$$a(x) = 1 + e^{\cos x}.$$

a) direct solve (A will be a full matrix

b) Conjugate gradient solver.

c) preconditioned CG solver using the constant coefficient problem on p.5 as the preconditioner.

Patrick should have some good PCA application questions.

I hope to see you all Monday!