Lecture notes on linear dynamical systems for neuroscience*

Il Memming Park and Matthew Dowling

November 17, 2022

1 Motivation

Our aim is to understand complex spatio-temporal phenomena that involve the nervous system. In the physical space, interaction among entities generates collective phenomena: there are many neurons, multiple subregions of a single neuron, multiple brain areas, and even multiple brains that are socially interacting. Various measurement technologies allow access to many pixels/voxels imaged, many electrodes that record electrical activities, and many magnetic field sensors. In the time domain, neural activity of interest in neuroscience span time scales from microseconds to years. Only through time, the neural systems can process information, store information, adapt to new context, learn, and generate meaningful behavior. Both for the practical data analysis, and for the theoretical study of the neural system, we often face the grand challenge of making sense of high-dimensional spatio-temporal dynamics. If we want to "understand" how the brain works, we need some intuition on high-dimensional spatio-temporal dynamics.

This may seem daunting at first, but we can stand on the shoulders of giants who developed various conceptual tools. In addition, fortunately, anyone can learn the mathematical intuition for making precise statements on those concepts. In this week, we will learn versatile and extremely useful framework of linear dynamical systems. There are two key intuitions covered within two subjects in mathematics.

Learning Objective 1

Intuition on high-dimensional spaces can be gained by first studying **linear algebra**.

Learning Objective 2

Intuition on lawful temporal changes can be gained by first studying **linear dynamical systems**.

Learning them benefits you because of their immense practicality. For example, data analysis methods such as principal components analysis (PCA), filtering, fourier transform, and linear regression are much faster than corresponding nonlinear extensions. The recent boost in deep neural networks is fueled by the fast linear operations implemented on modern computer architectures (GPGPU-computing).

Saves you time and the planet

Linear systems are computationally *fast* and can be made numerically stable.

As we will see, with linear dynamical systems theory, we can understand when systems shows stable behavior, and how quickly it forgets the past. It may be surprising that there are a finite categories of linear dynamical systems if we are only concerned with their asymptotic behavior.

*Available online: `https://github.com/memming/linear-algebra-and-dynamics-lectures`

Easy asymptotic theory

We can understand *all* possible (finite-dimensional) linear dynamical systems.

This allows us to theoretically understand roughly what is going to happen in systems at longer time scales thanks to powerful first-order approximation techniques.

Linearize (locally)

Even complex nonlinear systems usually have a meaningful linear component.

Of course, there are exceptions and limitations, however, there are extensions and generalizations that typically incorporates some form of linearity.

2 Discrete-time Linear Dynamical System

In most aspects discrete-time and continuous-time linear dynamical systems are analogous. Continuous-time systems require more abstract mathematical skills, namely calculus and its friends, while discrete-time systems have lower barriers since solutions can be straightforwardly computed. Discrete-time systems can be implemented with simple loops on computers. A simple *Euler integration* scheme "tightly" connects the two approaches, and allows most of the practical intuitions to be transferred.

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x) \quad \Longleftrightarrow \quad x(t+\Delta) = x(t) + \Delta \cdot f(x) \tag{1}$$

for sufficiently small Δ. We will implicitly take discrete time steps of size Δ. Now, let us dive into discrete-time systems!

2.1 Memory traces in a 1-dimensional linear dynamical system

Consider an abstract leaky membrane potential model:

$$v(t+1) = a \cdot v(t) + I(t) \tag{2}$$

where $t = 0, 1, \ldots$ represents discrete time steps, $v(t) \in \mathbb{R}$ is the membrane potential, $a \in \mathbb{R}$ is a constant, and $I(t) \in \mathbb{R}$ is a current entering the neuron.

1. Given an initial condition $v(0)$ and the external current $I(t)$ for $t = 0, 1, \ldots$, write the general form of $v(t)$ using the summation notation (e.g. $\sum_{s=0}^{t}$). (Hint: try writing out $v(1), v(2), v(3)$ first and generalize.)

2. Sketch the solution over time from $t = 0$ to $t = 10$ for $a = 0.9$, $a = 1$, and $a = 2$ given $v(0) = 0$, $I(2) = 1$, and $I(t) = 0$ otherwise. Which one resembles the behavior of a neuronal membrane?

3. What is the value of a for an input pulse $I(0) \neq 0$ to decay to 10% of it original magnitude in 10 time steps (assume $v(0) = 0$ and $I(t > 0) = 0$)?

2.2 Linear network model

As we shall see later when we go to continuous time, $0 < a < 1$ is related to the time constant. The membrane potential time constant of a neuron is not very flexible; typically in the order of 10 milliseconds. Having many independent neurons does not help with the situation of forgetful membrane. But maybe we can create a network of (non-spiking) neurons that has more memory!

Consider a network of n linear neurons where each neuron's activity (analogous to membrane potential or firing rate) $x_i(t)$ evolves over time $t = 0, 1, 2, \ldots$ as,

$$x_i(t+1) = w_{i,i}x_i(t) + \sum_{j \neq i} w_{i,j}x_j(t) + b_i I(t) \tag{3}$$

where $w_{i,i} \in \mathbb{R}$ plays the role of a in (2), and $w_{i,j}$ is the influence of j-th neuron's activity directly onto i-th neuron's activity. Note that there is no synaptic dynamics in this super simplified model. Despite its simplicity, it is no doubt a very useful theoretical model of neural dynamics [3–5].

Let's write this in matrix form. Define the neural activity vector $\mathbf{x}(t)$, the connectivity matrix W, and the input gain vector $\mathbf{b}$.

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \qquad W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{4}$$

Note that $W \in \mathbb{R}^{n \times n}$ is a square matrix, and $\mathbf{x}(t), \mathbf{b} \in \mathbb{R}^{n \times 1}$ are column vectors.

4. Write the matrix form difference equation for (3).

5. Write out the general form for $\mathbf{x}(t)$ using matrix power.

Sure, these are solutions, but not in a very transparent form. We desire more insights! First, let us recall some linear algebra to help us.

Matrix multiplication

Let A be an $n \times k$ matrix, and B be an $k \times m$ matrix. The i-th row and j-th column of A is denoted $A_{i,j}$. The matrix product $C = AB$ is $n \times m$, and its entries are given by,

$$\underset{(n\times m)}{\mathbf{C}} = \underset{(n\times k)}{\mathbf{A}} \underset{(k\times m)}{\mathbf{B}} \qquad C_{i,j} = (AB)_{i,j} = \sum_k A_{i,k}B_{k,j} \tag{5}$$

Note that matrix products don't commute, i.e., $AB \neq BA$.

What does *matrix multiplication* do? Let us try out with some special matrices.

2.3 Some special matrices as connectivity

Diagonal matrices are non-zero only on the diagonal entries.

$$\text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \tag{6}$$

6. If $W = \text{diag}(\lambda_1, \ldots, \lambda_n)$, what is the general form for $\mathbf{x}(t)$?

Identity matrices are **diagonal matrices** with ones on the diagonal (zero otherwise):

$$I = \text{diag}(1, 1, \ldots, 1) = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} \tag{7}$$

7. If $W = I$, what is the general form for $\mathbf{x}(t)$?

Cross diagonal matrices are non-zero only on the anti-diagonal entries.

$$C = \begin{bmatrix} 0 & \cdots & 0 & c_1 \\ \vdots & & \ddots & \vdots \\ 0 & c_{n-1} & \cdots & 0 \\ c_n & 0 & \cdots & 0 \end{bmatrix} \tag{8}$$

8. What is the general form for $\mathbf{x}(t)$ for the following cross-diagonal weight matrix?

$$W = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{9}$$

Strictly upper triangular matrices have the following zero entry pattern (X marks arbitrary value, example only shown in 3×3):

$$\begin{bmatrix} 0 & X & X \\ 0 & 0 & X \\ 0 & 0 & 0 \end{bmatrix} \tag{10}$$

9. What is the general form for $\mathbf{x}(t)$ for the following strictly upper triangular weight matrix?

$$W = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{11}$$

10. Since (3) is causal $\mathbf{x}(t)$ only depends on the past input, i.e., $I(s)$ for $s < t$. How much memory of the past does $\mathbf{x}(t)$ have?
    (a) for (11)

    (b) for (7)

2.4 Orthonormal and Unitary Matrices

Recall that the Euclidean vector space $\mathbb{R}^n$ can be spanned by n linearly independent basis vectors. In particular, one can choose a set of unit length vectors that are orthogonal to each other. We can write this in linear algebra terms.

Transpose of a matrix (or vector) is defined as $(A^\top)_{i,j} = A_{j,i}$.

Two vectors $\mathbf{u}$ and $\mathbf{v}$ are **orthogonal** if $\mathbf{u}^\top \mathbf{v} = 0$.

$\mathbf{v} \in \mathbb{R}^{n \times 1}$ is a unit vector if $\|\mathbf{v}\| = \sqrt{\mathbf{v}^\top \mathbf{v}} = 1$, i.e., it is of unit length.

Let the collection of vectors $(\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n)$ form an *orthonormal basis* of $\mathbb{R}^n$, that is, (1) any vector $\mathbf{x} \in \mathbb{R}^{n \times 1}$ can be expressed as $\mathbf{x} = \sum_{i=1}^n \alpha_i \mathbf{u}_i$ where α_i are scalar, (2) if $i \neq j$, then $\mathbf{u}_i$ and $\mathbf{u}_j$ are orthogonal, and (3) $\mathbf{u}_i$ is a unit vector.

An **orthonormal matrix** is a real-valued square matrix where the collection of column vectors form an orthonormal basis.

$$U = \begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_n \\ | & & | \end{bmatrix} \in \mathbb{R}^{n \times n} \tag{12}$$

11. Show that if U is orthonormal then $U^\top U = I$.

> **Fun Fact:** For square matrices A and B, if $AB = I$ then $BA = I$. B is the *matrix inverse* of A, that is, $B = A^{-1}$. (They are inverses of each other.)

12. Show that if U is orthonormal then $UU^\top = I$.

A complex-valued square matrix $U \in \mathbb{C}^{n\times n}$ is **unitary**, if $UU^\dagger = I$ where † denotes conjugate transpose operation. Orthonormal matrices are unitary.

13. Show that multiplying a unitary matrix to a column vector preserves its norm, i.e., it is an *isometric* transformation.

14. Let W in (3) be orthonormal. In the absence of input, i.e., $I(t) = 0$, what can you say about $\mathbf{x}(t)$ given $\mathbf{x}(0)$? (geometrically)

A **permutation matrix** is a matrix where each row and column has a single unity and zero otherwise. (Why is it called a permutation matrix?)

15. Show that a permutation matrix is a unitary matrix.

16. The following **rotation matrix** is a unitary matrix. Multiplying vectors in $\mathbb{R}^2$ results in a counter-clock rotation by θ.

$$W = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{13}$$

(a) Sketch out the general form of $\mathbf{x}(t)$ when $I(t) = 0$ and $\theta = \pi/4$.

(b) How is the dynamics qualitatively different when θ is a rational multiple of π or not?

17. Orthonormal matrices can also "flip" or reflect vectors with respect to axes. Consider

$$W = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{14}$$

(a) Show that W is unitary.

(b) What is the general form of $\mathbf{x}(t)$ when $I(t) = 0$?

18. Let U be a unitary matrix. Show that unitary matrix preserves the inner product structure, i.e., $\mathbf{x}^\dagger \mathbf{y} = (U\mathbf{x})^\dagger (U\mathbf{y})$.

Unitary or Orthonormal Matrix

It can rotate, permute, and reflect vectors while preserving their norm and inner product structure.

2.5 Spectral Decomposition

Now we reach the first and arguably one of the most important matrix decompositions.

A matrix is **symmetric** if $A = A^\top$. A complex matrix is **Hermitian** (or self-adjoint) if $A = A^\dagger$. Note that only square matrices can be symmetric or Hermitian.

Spectral theorem for Hermitian matrices

Any *real symmetric square (or Hermitian) matrix* A can be decomposed into the following form,

$$A = U\Lambda U^\dagger \qquad \text{(spectral decomposition)} \tag{15}$$

where $U = \begin{bmatrix}\mathbf{u}_1 & \cdots & \mathbf{u}_n\end{bmatrix}$ is an orthonormal (or unitary) matrix, and Λ is a *real*-valued diagonal matrix.

19. Let the connectivity matrix W be real symmetric. This means every pair of neurons in the network reciprocally activates each other with the same strength. Use the spectral decomposition (15) to answer the following questions.

    (a) What is WW?

    (b) What is W^n?

    (c) What is the general form of $\mathbf{u}_i{}^\top \mathbf{x}(t)$ in the absence of input ($I(t) = 0$) and with initial condition $\mathbf{x}(0) = \mathbf{u}_i$?

    (d) What is the general form of $\mathbf{x}(t)$ in the absence of input ($I(t) = 0$)?

    (e) Can linear neural networks with symmetric weight matrices oscillate without input?

    (f) What is the general form of $\mathbf{x}(t)$ with input?

Eigenvalues and eigenvectors

Given a matirx A, solutions to the following eigenvalue equation

$$A\mathbf{x}_i = \lambda_i \mathbf{x}_i, \tag{16}$$

are called the (right) eigenvectors ($\mathbf{x}_i$) and eigenvalues (λ_i) of A.

20. Given a spectral decomposition of $A = U\Lambda U^\dagger$, show that $\mathbf{u}_i$ is an eigenvector with corresponding eigenvalue $\Lambda_{i,i}$ of A.

As you have just seen, the spectral decomposition makes things easy. But are there other matrices that also allow such decomposition?

Normal matrices

A matrix W is called **normal**, if it commutes with its (conjugate)-transpose.

$$W^\dagger W = W W^\dagger$$

21. Determine if the following non-symmetric matrices are normal.

(a)

$$\begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix} \tag{17}$$

(b)

$$\begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix} \tag{18}$$

The class of normal matrices is pretty big, and includes diagonal, unitary, symmetric, and skew-symmetric ($A = -A^\top$) matrices. Normal matrices are exactly those that have *a complete set of* orthonormal *eigenvectors*. The spectral theorem extends to the class of normal matrices, but the eigenvalues may be complex-valued.

Spectral theorem for Normal matrices

Any *normal matrix* A can be decomposed into the following form,

$$A = U\Lambda U^\dagger \qquad \text{(spectral decomposition)} \tag{19}$$

where U is a unitary matrix, and Λ is a *complex*-valued diagonal matrix.

22. Show that if a normal matrix W is real-valued, then its complex eigenvalues come in complex conjugate pairs.

23. Show that the eigenvalues of a rotation matrix are of the form $e^{\pm i\theta} = \cos(\theta) \pm i\sin(\theta)$.

$$W_\theta = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad U = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \tag{20}$$

24. Show that if A is normal, so is $A + \sigma I$ for any $\sigma \in \mathbb{C}$.

In some sense, linear dynamics with normal matrices are easy to understand now. In the following section, we will attempt to understand the other cases.

2.6 Singular Value Decomposition

Another very powerful matrix decomposition that applies to any matrix is the SVD.

SVD

Any rectangular matrix $A \in \mathbb{C}^{m\times n}$ can be factored into the following form,

$$A = U\Sigma V^\dagger \qquad \text{(singular value decomposition)} \tag{21}$$

where $U \in \mathbb{C}^{m\times m}$ is a unitary matrix, $\Sigma \in \mathbb{R}^{m\times n}$ is diagonal matrix with non-negative values and $V \in \mathbb{C}^{n\times n}$ is a unitary matrix.

The diagonal entries in Σ are called the **singular values** of A, and the columns of U and V are called the left and right singular vectors. SVD provides a nice interpretation of matrix multiplication in general as a sequence of unitary transformation, scaling, and another (potentially different) unitary transformation.

25. Show the relationship between singular vectors and singular values of A and the eigenvectors of the square matrices $AA^\dagger$ and $A^\dagger A$.

26. Find a singular value decomposition of the following square matrix:

$$W_s = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{22}$$

27. Write the general form of $\mathbf{x}(t)$ in the absence of input using the SVD of W. Do the dynamics decouple into nice independent modes?

Rank of a matrix is the number of non-zero singular values.

28. Consider a rank-1 neural network with $W = \mathbf{u}\mathbf{v}^\top$. (Why is it rank-1?)

$$\mathbf{u} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad W = \tag{23}$$

Describe its time evolution $\mathbf{x}(t)$.

Frobenius norm of a matrix A is defined as $\|A\|_F = \sqrt{\sum_{i,j} |A_{i,j}|^2}$.

> **Low-rank approximation (Eckart–Young) theorem (practical):** The best low-rank approximation of a matrix (in terms of Frobenius norm) is given by dropping the smallest singular values.

29. According to the Eckart–Young theorem, replacing the smallest singular values with 0 is a good way to achieve low-rank approximation of a matrix. For example, SVD of the following matrix shows two rank-1 components.

$$W_c = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} = 4 \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{24}$$

$$= 4 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{25}$$

The best rank-1 approximation is the component with singular value 4.

$$\tilde{W}_c = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{26}$$

Compare the two linear neural networks with weight matrix W_c and $\tilde{W}_c$, correspondingly.

2.7 Reading assignment for Day 1

Read Goldman [5] with the following questions in mind:

30. What is the form of recurrent weight matrix corresponding to the sequential activation network in Fig 1? Consider the problem in discrete time (3).

31. Take an orthonormal matrix U and left multiply to the weight matrix obtained from above. Can you rewrite the dynamics equation in a new coordiate system $\mathbf{y}(t) = U^\top \mathbf{x}(t)$? How does this relate to Fig 3C?

32. How does the eigenvector represent feedback only onto themselves and do not interact with each other?

2.8 Matrix Algebra Cheatsheet

Fun Fact: Every linear transformation is a matrix multiplication.

Matrix Fun Facts

Matrix multiplication: Vector inner product and outer products are special cases:

$$\mathbf{u}^\top \mathbf{v} = \begin{bmatrix} u_1, u_2, \cdots, u_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \sum_k u_k v_k \qquad \textit{inner product} \tag{27}$$

$$\mathbf{u}\mathbf{v}^\top = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \begin{bmatrix} v_1, v_2, \cdots, v_m \end{bmatrix} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & \cdots & u_1 v_m \\ u_2 v_1 & u_2 v_2 & \cdots & u_2 v_m \\ \vdots & \vdots & \ddots & \vdots \\ u_n v_1 & u_n v_2 & \cdots & u_n v_m \end{bmatrix} \qquad \textit{outer product} \tag{28}$$

Matrix outer product stacks:

$$\begin{bmatrix} | \\ \mathbf{u}_1 \\ | \end{bmatrix} \begin{bmatrix} — \mathbf{v}_1 — \end{bmatrix} + \begin{bmatrix} | \\ \mathbf{u}_2 \\ | \end{bmatrix} \begin{bmatrix} — \mathbf{v}_2 — \end{bmatrix} = \begin{bmatrix} | & | \\ \mathbf{u}_1 & \mathbf{u}_2 \\ | & | \end{bmatrix} \begin{bmatrix} — \mathbf{v}_1 — \\ — \mathbf{v}_2 — \end{bmatrix} \tag{29}$$

Matrix times diagonal matrix scales column vectors:

$$\begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_n \\ | & & | \end{bmatrix} \cdot \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix} = \begin{bmatrix} | & & | \\ d_1 \mathbf{u}_1 & \cdots & d_n \mathbf{u}_n \\ | & & | \end{bmatrix} \tag{30}$$

$$\begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix} \cdot \begin{bmatrix} — \mathbf{v}_1 — \\ \vdots \\ — \mathbf{v}_n — \end{bmatrix} = \begin{bmatrix} — d_1 \mathbf{v}_1 — \\ \vdots \\ — d_n \mathbf{v}_n — \end{bmatrix} \tag{31}$$

Fun Fact: $(AB)^\top = B^\top A^\top$.

3 Non-normal Linear Time Invariant System

Let us continue to develop more intuition on the memory properties of linear "recurrent" neural networks of the form (3). As discussed in Goldman [5], some linear network models can be viewed as feedforward, while others can be feedback. This distinction is very important for the memory properties of the system. The traditional tool for analyzing memory traces in a neural system is to use the eigenvalues (also known as eigenspectrum). In this section, we will still stick with discrete time.

33. **Similarity transform and change of basis**: Let M be any invertible matrix. Consider a linear system over discrete time, $\mathbf{x}_{t+1} = A\mathbf{x}_t$. A change of variable $\mathbf{y}_t = M\mathbf{x}_t$ is a linear trnasformation, or a change of basis. Show that $B = M^{-1}AM$ is the new dynamics matrix for $\mathbf{y}_t$. (B is said to be *"similar"* to A)

34. Consider a square matrix A. Let $B = M^{-1}AM$ be a similarity transform of A. Show that similar matrices A and B have the same eigenvalues.

3.1 Diagonalization

Non-normal matrices

$$W^\top W \neq WW^\top$$

The spectral theorem is nice because it decouples each dimension, but can we extend such analysis to non-normal matrices?

Diagonalization

If a square matrix has n linearly independent (not necessarily orthogonal) eigenvectors, we can diagonalize them using its eigenvectors.

$$A = S\Lambda S^{-1} \qquad \text{(diagonalization)} \tag{32}$$

where Λ is a diagonal matrix of complex numbers.

35. Verify that the columns of S are (right) eigenvectors of A.

As long as the set of eigenvectors span the whole space, we can use (32) in reverse to construct matrices with specific eigenvectors and eigenvalues. For example, take $\mathbf{v}_1 = [1, 0]^\top$, $\mathbf{v}_2 = [1, 1]^\top/\sqrt{2}$, $\lambda_1 = 1$, and $\lambda_2 = 2$. Let $S = [\mathbf{v}_1, \mathbf{v}_2]$ and $\Lambda = \text{diag}(\lambda_1, \lambda_2)$. We get the following non-normal matrix:

$$W_a = S\Lambda S^{-1} = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} \tag{33}$$

Let W be diagonizable, and $\mathbf{x}(t+1) = W\mathbf{x}(t)$. Let $\mathbf{v}_i$ be eigenvectors of W.

$$\mathbf{x}(t) = (S\Lambda S^{-1})^k\mathbf{x}(0) = (S\Lambda S^{-1})\cdots(S\Lambda S^{-1})\mathbf{x}(0) = S\begin{bmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k \end{bmatrix} S^{-1}\mathbf{x}(0) \tag{34}$$

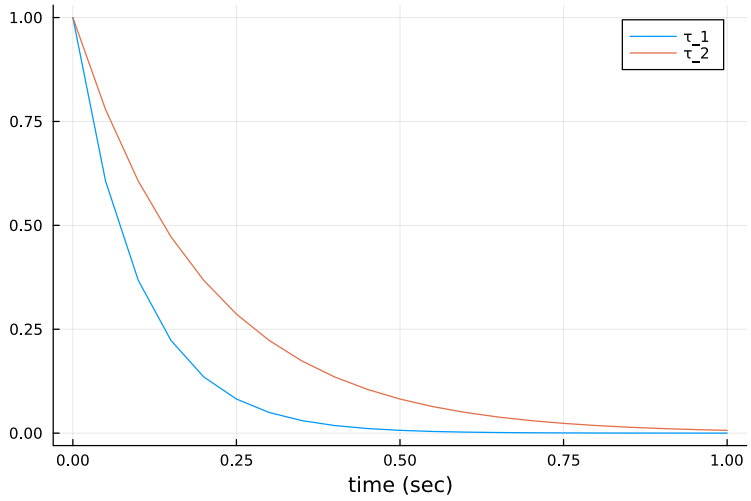
36. Show that $\mathbf{x}(t) = S\lambda_i^t$ if initialized at the eigenvector, i.e., $\mathbf{x}(0) = \mathbf{v}_i$.

    This means dynamics decomposed onto the eigenmodes (spanned by $\mathbf{v}_i$) behave independently and geometrically. This is almost as nice as the spectrally decomposed normal matrix case. They just need to be combined back with S which is not an *orthnormal* basis.

3.2 Difference of exponentials is usually not an exponential

37. Let $\tau_1 = 0.1$ and $\tau_2 = 0.2$, sketch the following difference of exponential decay (in continuous time):

$$f(t) = e^{-t/\tau_1} - e^{-t/\tau_2} \tag{35}$$



38. Explain how two eigenmodes that independently exhibit exponential (geometric) decay can manifest in seemingly transient amplification in a different reference frame (e.g. a single neuron activity).

$$\begin{aligned} e^{-t/\tau_1} - e^{-t/\tau_2} &= e^{-t/\tau_1}(1 - e^{-t/\tau_2}e^{+t/\tau_1}) = e^{-t/\tau_1}(1 - e^{-t/\tau_2+t/\tau_1}) && (36) \\ &= e^{-t/\tau_1}\left(1 - \left(1 + \left(\frac{t}{\tau_1} - \frac{t}{\tau_2}\right) + (\text{higher order terms})\right)\right) \\ &= \left(\frac{1}{\tau_2} - \frac{1}{\tau_1}\right) te^{-t/\tau_1} + (\text{higher order terms}) && (37) \end{aligned}$$

3.3 Purely feedforward networks

If there are no loops in the graph of the neural circuit, the neural activity of a neuron at time t will never influence itself later (at least within the circuit). Let us call such neural circuits **purely feedforward**.

39. Consider a network of 3 neurons where neuron 1 excites neuron 2 and neuron 2 excites neuron 3? Draw the neural circuit graph. What is the corresponding connectivity matrix?

40. Show that the connectivity matrix of any purely feedforward network can be written as a strictly triangular form (by choosing a certain ordering of neurons; recall that reordering neurons is just a multiplication by a permutation matrix).

3.4 Neural computation as a system

Given an input signal $u(t)$, a **system** outputs a corresponding output signal $y(t)$:

$$\{y(s), s = 0, 1, \ldots\} = H\{u(s), s = 0, 1, \ldots\} \tag{38}$$

where $H : (\mathbb{N} \to \mathbb{R}) \to (\mathbb{N} \to \mathbb{R})$ is an operator that denote the action of a system. Note that this is not necessarily an instantaneous mapping, but rather maps a time series to a time series. For simplicity, we will abuse the notation a little bit and simply write,

$$y(t) = H\{u(t)\}. \tag{39}$$

Typically we consider only systems that are *causal*, that is, output is produced through a causal mechanism and does not depend on the future input nor output. A causal system can capture the essence of biophysical computation at various scales–a transformation of signals over time. This transformation may also depend on the internal state of the neural system in general. However, for the simplicity, let us consider systems without an internal state.

A system is said to be **time-invariant** if its response to a certain signal does not depend on (absolute) time.

$$H\{u(t + \tau)\} = y(t + \tau) \tag{40}$$

which shows that time shift of the input of the system is the same as the output shifted by the same amount of time. Note that if a system is changing over time, this is not necessarily true; for example, with time evolution of the unobserved internal state of the system induced by synaptic plasticity or neuromodulators.

A system is said to be **linear**, if linear combination of input leads to corresponding linear combination of outputs,

$$H\{a \cdot u_1(t) + b \cdot u_2(t)\} = a \cdot y_1(t) + b \cdot y_2(t) \tag{41}$$

where $y_i(t) = H\{u_i(t)\}$ for $i = 1, 2$. Most neural systems are nonlinear, but linear analysis often provides a powerful baseline of understanding.

41. Let $y(t) = u(t - 1)$ be a system. It *delays* the input by 1 time step. For example, this delay could represent conduction delay of an axon. Verify that this system is causal, linear, and time-invariant.

42. Let $y(t) = \frac{1}{3}(u(t) + u(t - 1) + u(t - 2))$ be a system. It averages the input of the most recent 3 time steps, producing a smoothed output. Verify that this *moving average* system is causal, linear, and time-invariant.

A linear neural network can be seen as a system. For the simple case where it takes a scalar time series $u(t)$ as input and produces a scalar time series $y(t)$ as output, we can use,

$$\mathbf{x}(t+1) = W\mathbf{x}(t) + \mathbf{b}^\top u(t) \tag{42}$$

$$y(t) = \mathbf{c}^\top \mathbf{x}(t) \tag{43}$$

where $\mathbf{b}$ and $\mathbf{c}$ are instantaneous linear maps to and from the vector of neural activities $\mathbf{x}(t)$.

43. Show that a purely linear neural network is a causal LTI system. Draw neural circuit diagrams that corresponds to questions 41 and 42.

A linear time-invariant (LTI) system is fully characterized by its *impulse response*, defined by the output of the system to a unit pulse input at time $t = 0$ (delta function; corresponds to the Dirac delta function in continuous time).

$$h(t) \triangleq H\{\delta(t)\} \qquad \text{(impulse response)} \tag{44}$$

$$\delta(t) = \begin{cases} 1 & t = 0 \\ 0 & t > 0 \end{cases} \tag{45}$$

44. What is the impulse response of the system $y(t) = u(t-2)$? Sketch it out over time.

Convolution

Given two signals $x(t)$ and $y(t)$ for $t \in \mathbb{Z}$, we define convolution as

$$(x * y)(t) = \sum_{k=-\infty}^{\infty} x(k)y(t-k) \tag{46}$$

45. Given an LTI system with impulse response $h(t)$, show that the output of the system is a convolution of the input with the impulse response, i.e.,

$$y(t) = (u * h)(t) \tag{47}$$

When a system's impulse response is restricted to a finite number of non-zero elements, it is said to have a **finite impulse response (FIR)**. If there are infinite number of non-zero elements, then the system has an **infinite impulse response (IIR)**.

46. Show that a purely feedforward linear network produces a finite impulse response.

47. Show that a linear network with a diagonal matrix produces an infinite impulse response.

LTI systems are used as **filters** in signal processing tools for smoothing, denoising, sharpening, removing line noise, isolating signals in a particular frequency bands, and so on. The *frequency response* of an LTI is characterized by the Fourier transform of the impulse response. If you want to study this subject further, we recommend reading a textbook on "digital signal processing" or "signals and systems" (electrical engineering curriculum).

3.5 Coordinate transformed feedforward networks

So purely feedforward system is an FIR LTI, and also the connectivity weights are strictly upper (or lower) triangular.

48. Consider the following 2-neuron feedforward system:

$$\mathbf{x}(t+1) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 & 1 \end{bmatrix} u(t) \tag{48}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(t) \tag{49}$$

Let us do a change of variables, $\mathbf{z}(t) = M\mathbf{x}(t)$ with an invertible matrix, say a unitary matrix

$$M = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{50}$$

(a) What is the resulting neural network system defined with new neurons $z_1(t)$ and $z_2(t)$?

(b) Is the weight matrix strictly upper triangular for $\mathbf{z}$? Can it be made so through permutation?

(c) Does the impulse response change?

(d) Draw the neural circuit diagram of neurons $\mathbf{z}$ and compare to that of $\mathbf{x}$.

3.6 Schur decomposition

As we saw from 48, systems with finite impulse response do not necessarily have strictly triangular form. Can we reverse the construction of 48 to determine if a system is purely feedforward in disguise? There's another matrix decomposition that can answer this question.

Schur decomposition

Any square matrix (real- or complex-valued) A can be decomposed into the following form,

$$A = UTU^\dagger \qquad \text{(schur decomposition)} \tag{51}$$

where U is a unitary matrix, and T is an upper triangular matrix.

The Schur decomposition can be numerically found cheaply. If the resulting triangular matrix T has no non-zero diagonal element, that is, strictly triangular, then it is a feedforward system in disguise!

4 Continuous-time Linear Dynamical Systems

Physical systems are unarguably well described by changes through continuous time. In continuous time, we can harness the power of calculus and differential equations to make powerful statements about dynamical systems without relying on "crude" discretization of time with a small time step.

4.1 Scope

We want to study homogenuous differential equations with constant coefficients that can be written succintly in matrix form as

$$\dot{\mathbf{x}}(t) = F\mathbf{x}(t) \tag{52}$$

4.2 First order linear differential equations

Consider the following first order homogenuous linear ODE with constant coefficients

$$\dot{x}(t) + ax(t) = 0 \tag{53}$$
$$x(0) = x_0 \tag{54}$$

To solve the differential equation means to find the function $x(t)$ which passes through x_0 at time $t = 0$ and has a derivative, which irregardless of time, is the function evaluation scaled by a. Lets begin by making the formal rearrangement

$$\frac{dx}{x} = -a\,dt \tag{55}$$
$$\implies x(t) = c_1 \exp(-at) \tag{56}$$

If we were not given the initial condition, $x(0) = k$, this is as far as we could go in determining a solution. Fortunately, we can solve for c_1 by using the initial condition, so that $c_1 = x_0$. Thus, the full solution to the differential equation is just $x(t) = x_0 \exp(-at)$.

Remark: The system this differential equation describes is not very interesting. If $a < 0$ then $x(t) \to \infty$, whereas if $a > 0$ then $x(t) \to 0$.

Remark: If $\phi(t) = e^{-at}$,

$$x(t) = \phi(t)x(0) \tag{57}$$
$$x(t + \tau) = \phi(\tau)x(t) \tag{58}$$

49. Draw the phase portrait for this first order system.

4.3 N-th order linear differential equations

Let us consider a more interesting example where we consider the solution to an N-th order linear differential equation with constant coefficients given by

$$\sum_{i=0}^{N} a_i x^{(i)}(t) = 0 \tag{59}$$

$$x^{(i)}(0) = k_i \qquad i = 0, \ldots, N-1 \tag{60}$$

without loss of generality $a_N = 1$. If we define

$$\mathbf{x}(t) = \begin{bmatrix} x(t) & x^{(1)}(t) & \cdots & x^{(N-1)} \end{bmatrix}^\top \tag{61}$$

then we can transform this N-th order equation into a first order one below

$$\dot{\mathbf{x}}(t) = F\mathbf{x}(t) \tag{62}$$

$$\mathbf{x}(0) = \mathbf{k} \tag{63}$$

by overloading the differential operator so that $\dot{\mathbf{x}}(t) = \begin{bmatrix} x^{(1)}(t) & x^{(2)}(t) & \cdots & x^{(N)} \end{bmatrix}^\top$, and setting

$$F = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_1 & -a_2 & -a_3 & \cdots & -a_{N-1} \end{pmatrix} \tag{64}$$

Remark: A matrix in this form is said to be a companion form matrix. They arise when we transform an N-th order differential equation into a first order system as we did above.

Remark: Not every matrix is similar to a companion form. A matrix $A \in \mathbb{R}^{n\times n}$ can be written in companion form if it is not singular (i.e. full rank).

4.4 Fundamental matrix solution

How can we actually solve this matrix differential equation? That is, how can we find the vector that satisfies Eq. (62) and the associated initial condition? To answer this question, we have to take a slight detour and discuss the *fundamental matrix solution*

Lets begin by assuming that we can find the solution to Eq. (62), then if we denote the N solutions as $\phi_1(t), \phi_2(t), \ldots, \phi_N(t)$ then we can write

$$a_0\phi_1(t) + a_1\phi_1^{(1)}(t) + \cdots + \phi_1^{(N)}(t) = 0 \tag{65}$$

$$a_0\phi_2(t) + a_1\phi_2^{(1)}(t) + \cdots + \phi_2^{(N)}(t) = 0 \tag{66}$$

$$\vdots \tag{67}$$

$$a_0\phi_N(t) + a_1\phi_N^{(1)}(t) + \cdots + \phi_N^{(N)}(t) = 0 \tag{68}$$

which if take $\Phi_i(t) = \begin{bmatrix} \phi_i(t) & \phi_i^{(1)}(t) & \cdots & \phi_i^{(N)}(t) \end{bmatrix}^\top$, and set

$$\mathbf{\Phi}(t,0) = \begin{pmatrix} | & | & & | \\ \Phi_1(t) & \Phi_2(t) & \cdots & \Phi_N(t) \\ | & | & & | \end{pmatrix} \tag{69}$$

then we can write that

$$\dot{\mathbf{\Phi}}(t,0) = F\mathbf{\Phi}(t,0) \tag{70}$$

Now, if we were to make an *ansatz* and guess that the solution is $\mathbf{x}(t) = \mathbf{\Phi}(t,0)\mathbf{x}(0)$ then we can easily verify whether or not this is true

$$\dot{\mathbf{x}}(t) = F\mathbf{x}(t) \tag{71}$$
$$\implies \dot{\mathbf{\Phi}}(t,0)\mathbf{x}(0) = F\mathbf{\Phi}(t,0)\mathbf{x}(0) \tag{72}$$
$$\implies F\mathbf{\Phi}(t,0)\mathbf{x}(0) = F\mathbf{\Phi}(t,0)\mathbf{x}(0) \tag{73}$$

which also implies that $\mathbf{\Phi}(0,0) = I$.

Fundamental Solution

Theorem 1 (Series representation of fundamental solution)**.** *The series of matrices, $M_0, M_1, M_2, \ldots$, defined recursively as*

$$M_0 = I \tag{74}$$
$$M_k = I + F\int_0^t M_{k-1}(\sigma)d\sigma \tag{75}$$

converges uniformly to $\Phi(t,0)$

If we expand the recursion above (which we could do with the aid of induction), we see something interesting

$$\mathbf{\Phi}(t,0) = I + Ft + F^2\frac{t^2}{2!} + F^3\frac{t^3}{3!} + \cdots \tag{76}$$

If F were a scalar, then this would exactly be the series representation of $\exp(Ft)$. In fact, this infinite series is exactly how we define the *matrix exponential*.

Properties of the matrix exponential

Diagonal matrices: If $D = \text{diag}(d_1, d_2, \ldots, d_N)$ then $\exp(D) = \text{diag}(e^{d_1}, e^{d_2}, \ldots, e^{d_N})$.

Commuting matrices: If A and B commute (i.e., $AB = BA$), then we have that $\exp(A+B) = \exp(A)\exp(B) = \exp(B)\exp(A)$.

Diagonalizable matrices: If $A = SDS^{-1}$ where D is diagonal, then $\exp(A) = S\exp(D)S^{-1}$.

Inverse: $\exp(A)\exp(-A) = I$

50. Prove the property about commuting matrices.

avoid this common pitfall

Matrix exponential is not the exponential of each element of a matrix! Numerical software libraries will have both operations similar name (e.g. `exp` vs `expm` in MATLAB)

In many cases $\mathbf{\Phi}(t,0)$ is unwieldly to actually compute. However, we are fortunate in that we are working with differential equations of constant coefficients; in this particular case,

$$\mathbf{\Phi}(t,0) = \exp(Ft) \tag{77}$$

which we can verify satisfies both $\mathbf{\Phi}(0,0) = I$ as well as

$$\frac{d}{dt}\exp(Ft) = \frac{d}{dt}\left(I + Ft + F^2\frac{t^2}{2!} + F^3\frac{t^3}{3!}\cdots\right) \tag{78}$$

$$= F + F^2 t + F^3\frac{t^2}{2!} + \cdots \tag{79}$$

$$= F\exp(Ft) \tag{80}$$

4.5 Jordan form

Since the matrix exponential is defined in terms of an infinite series, we are left wondering whether it can be computed analytically. For this, we have to introduce the concept of a *Jordan matrix*. A matrix $J \in \mathbb{R}^{M\times M}$ is said to be a Jordan matrix if it takes the form

$$J = \begin{pmatrix} \lambda & 1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \lambda \end{pmatrix} \tag{81}$$

which means we could write that $J = \lambda I + N$; this matrix N is said to be nilpotent.

Nilpotent matrices

A matrix N is said to be nilpotent if there exists an integer k such that $N^k = 0$

51. Is the following matrix nilpotent? Support your claim either way.

$$Q = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \tag{82}$$

As we have shown, N and λI commute. Combining the first two properties of the matrix exponential, this allows us to write that $\exp(Jt) = \exp(\lambda t)\exp(Nt)$. However, we still have more work to do as we have not yet computed $\exp(Nt)$. Recalling that N is nilpotent, we can write

$$\exp(Nt) = \sum_{i=0}^{\infty} N^i \frac{t^i}{i!} \tag{83}$$

$$= \sum_{i=0}^{M-1} N^i \frac{t^i}{i!} \tag{84}$$

$$= \begin{pmatrix} 1 & t & \frac{t^2}{2!} & \frac{t^3}{3!} & \cdots & \frac{t^{M-1}}{(M-1)!} \\ 0 & 1 & t & \frac{t^2}{2!} & \cdots & \frac{t^{M-2}}{(M-2)!} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{85}$$

Great! Now we have the ability to easily calculate the matrix exponential of any Jordan form matrix. Now we may ask, what about the matrix exponential of more general matrices? As it turns out, any matrix can be written in Jordan canonical form.

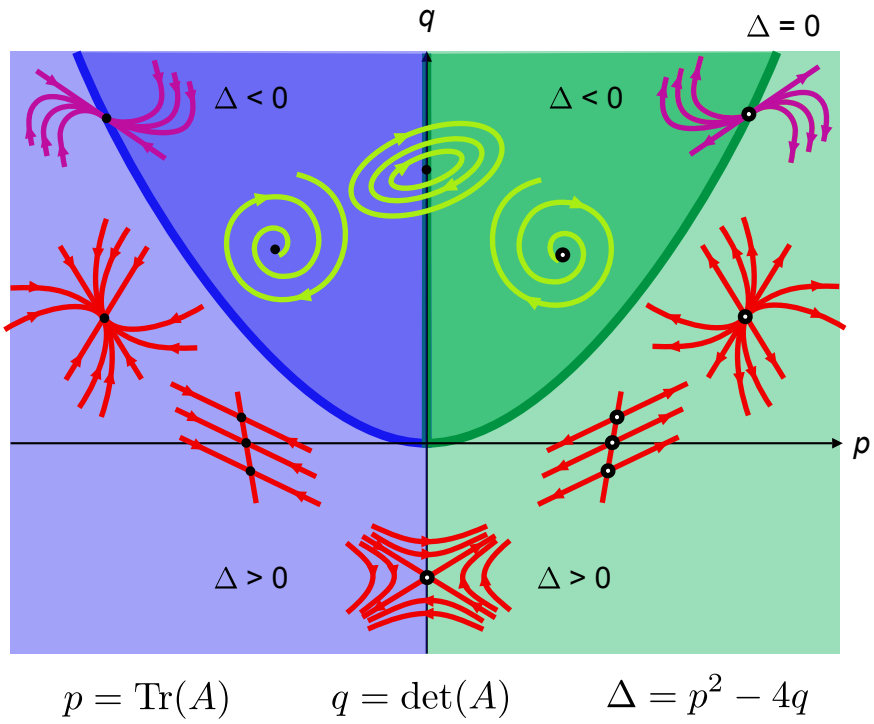


Figure 1: Different classes of two dimensional dynamical systems characterized by their trace and determinant. Note that the trace is the sum of eigenvalues, and the determinant is the product of eigenvalues. (source)

Jordan normal form

Definition of Jordan Normal Form: We say a matrix $J \in \mathbb{R}^{N\times N}$ is in Jordan normal form if it can be written as follows:

$$J = \begin{pmatrix} J_1 & 0 & \cdots & 0 \\ 0 & J_2 & \cdots & 0 \\ 0 & 0 & \cdots & J_K \end{pmatrix} \tag{86}$$

where each J_i is a Jordan block matrix, i.e.

$$J_i = \begin{pmatrix} \lambda_i & 1 & 0 & \cdots & 0 \\ 0 & \lambda_i & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_i \end{pmatrix} \tag{87}$$

The **Jordan decomposition** of a matrix, A, is a factorization that *almost* diagonalizes A. In fact, *any* square matrix has a Jordan decomposition, which means it can be factorized as

$$A = TJT^{-1} \tag{88}$$

where J is in Jordan normal form, and the columns of T contain the generalized eigenvectors of A.

4.6 Algebraic and geometric multiplicity

The **characteristic polynomial** of a matrix, A, is defined as

$$p_A(x) = \det(xI - A) \tag{89}$$

If A is non-singular, with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_s$, and associated multiplicities $m_1, m_2, \ldots, m_s$, then its characterisitc polynomial will be of the form

$$p_A(x) = (x - \lambda_1)^{m_1}(x - \lambda_2)^{m_2} \cdots (x - \lambda_s)^{m_s} \tag{90}$$

if we have already converted our matrix into Jordan form then finding the characteristic polynomial is a rudimentary excercise. The best way to see this is through a motivating example

From Jordan form to characteristic polynomial

Ex.1 Say we have a matrix A and we have factorized it into its Jordan decomposition TJT^{-1} where $J = \text{diag}(J_1, J_2, J_3)$ which we may sometimes write as $J = J_1 \bigoplus J_2 \bigoplus J_3$

$$J_1 = \begin{pmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{pmatrix} \qquad J_2 = \begin{pmatrix} \lambda_2 \end{pmatrix} \qquad J_3 = \begin{pmatrix} \lambda_3 & 1 & 0 \\ 0 & \lambda_3 & 1 \\ 0 & 0 & \lambda_3 \end{pmatrix} \tag{91}$$

then the characteristic polynomail of A will be

$$p_A(x) = (x - \lambda_1)^2(x - \lambda_2)(x - \lambda_3)^3 \tag{92}$$

Ex.2 Take a matrix A again, and let its Jordan decomposition be $J = J_1 \bigoplus J_2 \bigoplus J_3 \bigoplus J_4$ where

$$J_1 = \begin{pmatrix} \lambda_1 \end{pmatrix} \qquad J_2 = \begin{pmatrix} \lambda_1 \end{pmatrix} \qquad J_3 = \begin{pmatrix} \lambda_2 \end{pmatrix} \qquad J_4 = \begin{pmatrix} \lambda_3 & 1 & 0 \\ 0 & \lambda_3 & 1 \\ 0 & 0 & \lambda_3 \end{pmatrix} \tag{93}$$

in which case the characteristic polynomial is

$$p_A(x) = (x - \lambda_1)^2(x - \lambda_2)(x - \lambda_3)^3 \tag{94}$$

Although the two matrices we looked at have different Jordan decompositions, they have the same characteristic polynomial.

Lets overload notation as we have already done where we take the polynomial expression before i.e.

$$p(x) = \sum c_i x^i \tag{95}$$

so that if we *plug* a matrix into the argument then we would write

$$p(A) = \sum c_i A^i \tag{96}$$

It so happens that if $p(x)$ is the characteristic polynomial of A, then $p(A) = 0$ [6]. Is there a polynomial of lesser order than the characteristic polynomial, lets call it $\chi(x)$, that $\chi(A) = 0$? If there is, we call this polynomial the **minimal polynomial**.

Minimal polynomials

Looking at **Ex.1** from earlier, we have that the minimal polynomial is just the characteristic polynomial, so that $p(x) = \chi(x)$.

However, the matrix from **Ex.2** has a minimal polynomial that does not coincide with its characteristic polynomial. In fact, its minimal polynomial is

$$\chi(x) = (x - \lambda_1)(x - \lambda_2)(x - \lambda_3)^3 \tag{97}$$

5 Nonlinear systems that are really linear

5.1 Limitations of Linear Dynamical Systems

Linear dynamical system (LDS) is fundamentally inappropriate for describing neural dynamics, because the asymptotic behavior of LDS can only support a single isolated fixed point, oscillating center, or a linear subspace of zero flow (continuous attractor), while most tasks require characteristics such as multiple attractors, stable limit cycles, and saddles when the task duration can be *arbitrarily long*. In other words, the topological stability structures expressible by LDS are too limited.

52. Sketch the phase portrait of some 2D dynamical system that can stably sustain in one of two "states", i.e., *bistable*.

However, in practice, even well trained experimental subjects only perform tasks for a limited amount of time. The existence of an upper bound in duration introduces a potential identifiability problem in inferring the underlying dynamics as neural computation, especially if the following conjecture is true.

Universal latent LDS conjecture

A latent LDS model can approximate the behavior of any dynamical system $\dot{\mathbf{x}} = f(\mathbf{x})$ arbitrarily well for a finite duration.

where by latent LDS model is of the form

$$\dot{\mathbf{z}}(t) = A\mathbf{z}(t) \tag{98}$$
$$\mathbf{x}(t) = C\mathbf{z}(t) \tag{99}$$

with matrices A, and C.

53. (10 points) Prove (constructively) or disprove the LDS conjecture.

5.2 Linearization theorems

Let us consider a (time-invariant and autonomous) nonlinear dynamical system:

$$\dot{x} = f(x) \tag{100}$$

where $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^n$. Much of this nonlinear dynamical system can be understood piece by piece using our knowledge of linear dynamical systems.

Linearize that beast!

Most nonlinear dynamical systems are locally a linear dynamical system, but strange non-local things can happen.

In particular, the dynamics around equilibria (fixed points) are topologically conjugate to that of the linearization. Let $\bar{x}$ be a fixed point of the dynamics, that is, $f(\bar{x}) = 0$. Note that there can be several or even infinitely many fixed points. Let $[Df]_{i,j}(\bar{x}) = \frac{\partial f_i}{\partial x_j}\big|_{\bar{x}}$ be the Jacobian (matrix), i.e., linearization of the dynamics at $\bar{x}$:

$$\dot{x} = Df(\bar{x})(x - \bar{x}) \tag{101}$$

Theorem 2 (Hartman-Grobman)**.** *If $Df(\bar{x})$ has no zero or purely imaginary eigenvalues then there is a homeomorphism h defined on some neighborhood U of $\bar{x}$ in $\mathbb{R}^n$ locally taking obits of the nonlinear flow ϕ_t to those of the linear flow.*

As the theorem states, linearlization around the fixed point fails if there are eigenvalues with zero real part. They are typically rare in practice, but of great interest to nonlinear dynamics analysis and important for bifurcation theory. Fixed points that guarantee linearlization are called **hyperbolic** fixed points. For hyperbolic fixed points, the stability of nonlinear system is equivalent to that of the linearized system.

For example, the following 2D nonlinear system,

$$\dot{x} = x^3 + y \tag{102}$$
$$\dot{y} = 3x \tag{103}$$

has a fixed point at $[0, 0]$, and is locally diffeomorphic to a saddle (Fig. 2).

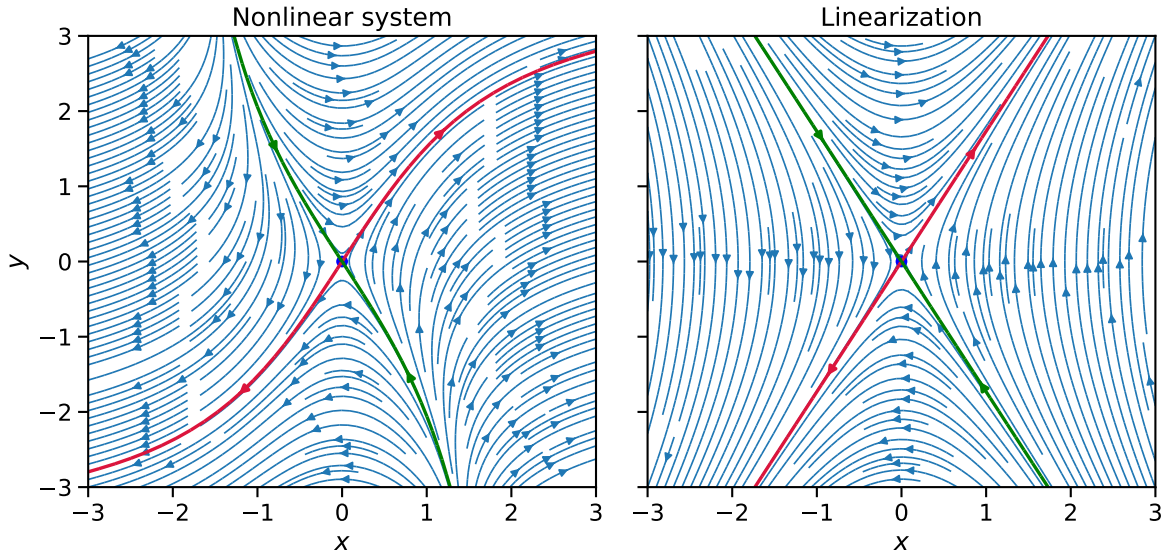


Figure 2: Linearization of the system described in (102).

Away from the fixed points, the flow of a linear or nonlinear dynamical system is topologically conjugate to a boring constant flow:

Theorem 3 (Rectification)**.** *If $p \in \mathbb{R}^n$ and $f(p) \neq 0$, then there are open sets $U, V \in \mathbb{R}^n$ with $p \in U$ and a diffeomorphism $g : U \to V$ such that the differential equation in the new coordinates, that is, the differential equation,*

$$\dot{y} = Dg(g^{-1}(y))f(g^{-1}(y)),$$

is given by $(\dot{y}_1, \ldots, \dot{y}_n) = (1, 0, 0, \ldots, 0)$.

See Chicone [2, Lemma 1.120].

Of course, a patchwork of linear dynamics can create some interesting systems. In neural data analysis, locally linear dynamical system model [12] and recurrent switching linear dynamical systems type models utilize this [7, 8].

5.3 Kernel methods

When linear methods fail due to nonlinearity, you can always try linear methods to a *nonlinear representation*. In this section, we will deviate from dynamical systems, but connect back in the next section. Let us consider the problem of linear regression from a vector explanatory variable $\mathbf{x} \in \mathbb{R}^d$ to a scalar dependent variable $y \in \mathbb{R}$, such that the square

error is minimized. Given a (training) dataset of n data points, $\{(\mathbf{x}_i, y_i)\}$, the least squares solution to the linear regression finds the best slope $\mathbf{w} \in \mathbb{R}^n$ that minimizes the following loss function:

$$L(w) = \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 \tag{104}$$

The result of linear regression is fitting the best slope $\mathbf{w}$ for a line.

But what if the relationship between $\mathbf{x}$ and y are not very linear? One potential solution is to map the explanatory variables to a higher dimensional space. For example, we can choose to double the explanatory space of least squares regression appending the squared value for each dimension,

$$\mathbf{x}' = \Phi(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^2 \end{bmatrix} \in \mathbb{R}^{d^2} \tag{105}$$

We will still be able to minimize the same loss function:

$$L(w) = \sum_{i=1}^{n} (y_i - \mathbf{w}'^\top \mathbf{x}'_i)^2 \tag{106}$$

but now the weights $\mathbf{w}'$ are also in $\mathbb{R}^{d^2}$. The result of such regression is now in a quadratic form:

$$y = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + w_{n+1} x_1^2 + \cdots + w_{2n} x_n^2 \tag{107}$$

but we only needed to solve a linear regression problem in the space of $\Phi(\mathbf{x})$. This quadratic regression is just a linear regression in disguise!

We present two important remarks without proof: (1) the optimal weight vector that minimizes (106) is a linear combination of data, that is,

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i). \tag{108}$$

for some α_i. This is known as the representer theorem, which allows to represent the solution as n number of α_i instead of d dimensional entries of $\mathbf{x}$. For a review of the subject, we recommend [11, Ch 4.2]. (2) in finding the optimal weight vector and for making predictions, the weight vector always interacts with $\Phi(\mathbf{x})$ through an inner product form. These two conditions holds for a number of important linear methods: Ridge regression, principal components analysis (PCA), support vector machine (SVM), canonical components analysis (CCA), least mean squares, spectral clustering, gaussian processes.

To make a very powerful map from $\mathbf{x}$ to y, we just need to use a map $\Phi(\mathbf{x})$ that carries all sorts of nonlinear representations of $\mathbf{x}$. Thanks to the representer theorem, even if the dimensionality of $\Phi(\mathbf{x})$ is very large, we only need n variables to represent the optimal solution. This means now we are almost ready to jump to infinite dimensions. The only problem is computing an inner product between $\Phi(\mathbf{x})$ and the weight vector $\mathbf{w}'$.

Inner product $\langle \mathbf{x} \mid \mathbf{y} \rangle$ has three properties it must satisfy:

$$\langle \mathbf{x} \mid \mathbf{y} \rangle = \langle \mathbf{y} \mid \mathbf{x} \rangle \qquad \text{(symmetric)} \tag{109}$$

$$\langle \mathbf{x} \mid \mathbf{x} \rangle \geq 0 \qquad \text{(positive semi-definite)} \tag{110}$$

$$\langle a_1 \mathbf{x}_1 + a_2 \mathbf{x}_2 \mid \mathbf{y} \rangle = a_1 \langle \mathbf{x}_1 \mid \mathbf{x} \rangle + a_2 \langle \mathbf{x}_2 \mid \mathbf{x} \rangle \qquad \text{(linear)} \tag{111}$$

A **kernel** function is a symmetric positive semi-definite function in the input space $\mathfrak{X}$.

$$k : \mathfrak{X} \times \mathfrak{X} \to \mathbb{R} \tag{112}$$

That is, for $x, y \in \mathfrak{X}$,

$$k(x,y) = k(y,x) \qquad \text{(symmetric)} \tag{113}$$

$$\sum_i \sum_j \alpha_i \alpha_j k(x_i, x_j) \geq 0 \qquad \text{(positive semi-definite)} \tag{114}$$

By the Moore-Aronzajn theorem[1], there exist a unique space with the inner product defined by a kernel, a so-called *reproducting kernel Hilbert space (RKHS)*.

Kernel Trick!

Implicitly map data to an infinite-dimensional space by using a powerful **kernel**. Apply a suitable kernelized linear method.

Some kernels can be shown to be *universal*, in the sense that they can approximate almost any nonlinear map arbitrarily closely [11].

Kernels not only extends linear methods to solve nonlinear problems, once we define kernels on *any* input space, we can apply linear methods on that space. There are graph kernels, sequence kernels, and even kernels defined on neural spike trains [9, 10].

5.4 Koopman operators

Can we take a similar approach to neural dynamics?

Acknowledgements

Special thanks to Ayesha Vermani and Ábel Ságodi.

References

[1] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950. ISSN 0002-9947.

[2] Carmen Chicone. *Ordinary Differential Equations with Applications*. Springer Science & Business Media, September 2006. ISBN 9780387357942.

[3] Shaul Druckmann and Dmitri B Chklovskii. Neuronal circuits underlying persistent representations despite time varying activity. *Current biology: CB*, 22(22):2095–2103, November 2012. ISSN 0960-9822, 1879-0445. doi: 10.1016/j.cub.2012.08.058.

[4] Surya Ganguli, Dongsung Huh, and Haim Sompolinsky. Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences*, 105(48):18970–18975, December 2008. ISSN 1091-6490. doi: 10.1073/pnas.0804451105.

[5] Mark S Goldman. Memory without feedback in a neural network. *Neuron*, 61(4):621–634, February 2009. ISSN 0896-6273, 1097-4199. doi: 10.1016/j.neuron.2008.12.012.

[6] Morris Hirsch and Stephen Smale. Differential equations, dynamical systems, and linear algebra (pure and applied mathematics, vol. 60). 1974.

[7] Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 914–922, Fort Lauderdale, FL, USA, 2017. PMLR.

[8] Josue Nassar, Scott W. Linderman, Monica Bugallo, and Il Memming Park. Tree-structured recurrent switching linear dynamical systems for multi-scale modeling. In *International Conference on Learning Representations (ICLR)*, November 2019.

[9] Il Memming Park, Sohan Seth, Murali Rao, and José C. Príncipe. Strictly positive definite spike train kernels for point process divergences. *Neural Computation*, 24(8):2223–2250, August 2012. doi: 10.1162/NECO_a_00309.

[10] Il Memming Park, Sohan Seth, Antonio R. C. Paiva, Lin Li, and Jose C. Principe. Kernel methods on spike train space for neuroscience: a tutorial. *IEEE Signal Processing Magazine*, 30(4):149–160, July 2013. ISSN 1053-5888. doi: 10.1109/msp.2013.2251072.

[11] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels : support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press, 2002. URL `http://www.worldcat.org/oclc/48970254`.

[12] Yuan Zhao and Il Memming Park. Interpretable nonlinear dynamic modeling of neural trajectories. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.