DSA5206 Advanced Topics in Data Science

AY2024/25 Special Term 1 By Zhao Peiduo

Part 1

A dynamical process is a sequence of states indexed by time:

$$\{x(t) \in \mathcal{X} : t \in \mathcal{T}\},$$

where $\mathcal{T}$ is a set of time indices, which can be subsets of either $\mathbb{Z}$ (discrete) or $\mathbb{R}$ (continuous).
Model of a Dynamical Process: A mathematical description of how the states $x(t)$ depend on time t. Examples include:

- Explicit formula: $x(t) = \sin(t)$
- Differential equation: $\dot{x}(t) = f(t, x(t)) \quad (t \in \mathbb{R})$
- Difference equation: $x(t+1) = g(t, x(t)) \quad (t \in \mathbb{Z})$

Input-Output Systems:

$$\mathbf{x} = \{x(t) : t \in \mathbb{R}\}, \quad \mathbf{y} = F(\mathbf{x}), \quad y(t) = F_t(x), \quad t \in \mathcal{T}$$

Examples:

- Convolutional model: $y(t) = \int_{\mathbb{R}} \rho(t-s)x(s)ds$
- Time-delay model: $y(t) = x(t-\tau)$

First-Principles vs Empirical Models:

- First-principles: derived from physical laws (e.g., Newton's law: $F = G\frac{Mm}{r^2}$)
- Empirical: fit to data, e.g., $z(t) = \sum_{j=1}^{n} a_j e^{i\omega_j t}$

Empirical Model Classification:

- *Non-parametric:* no fixed model form, e.g., $y(t) = \sum_{s=0}^{\infty} \rho(s)x(t-s)$
- *Parametric:* specified structure, e.g.,

$$y(t) = \sum_{s=1}^{n} a(s)y(t-s) + \sum_{r=0}^{m} b(r)x(t-r)$$

- *Black-box:* no interpretation, only predictions
- *Grey-box:* incorporates some physical knowledge

Temporal Index Sets:

- Discrete: $\mathcal{T} \subset \mathbb{Z}$
- Continuous: $\mathcal{T} \subset \mathbb{R}$

$$y(t) = \int_{-\infty}^{t} \rho(t-s)x(s)ds \quad \rightarrow \quad y(t) = \sum_{s \le t-\delta} \rho(t-s\delta)x(s\delta)\delta$$

Linear Functions:

$$H(\alpha x_1 + \beta x_2) = \alpha H(x_1) + \beta H(x_2) \quad \forall \alpha, \beta \in \mathbb{R},\ x_1, x_2 \in \mathcal{X}$$

Linear vs Nonlinear Models:

- Linear differential: $\dot{x}(t) = Ax(t) + b(t)$
- Linear difference: $x(t+1) = Ax(t) + b(t)$
- Linear PDE: $\partial_t u(t,x) = \partial_x^2 u(t,x)$
- Convolution: $y(t) = \int_{\mathbb{R}} \rho(t-s)x(s)ds$

Time-Varying vs Time-Invariant:

$$(T_\tau x)(t) = x(t-\tau)$$

System is time-invariant if: $T_\tau F(x) = F(T_\tau x)$

Time-Invariant Systems

- $y(t) = \int_{-\infty}^{t} e^{-w(t-s)} x(s)\, ds$
  Reason: The kernel $e^{-w(t-s)}$ depends only on the time difference $t-s$.
- $y(t) = x(t-\tau)x(t-\tau)$
  Reason: The output depends only on delayed versions of the input; the delay is constant.
- $y(t) = x(t-\tau) + 3x(t-2\tau)$
  Reason: The output depends only on delayed versions of the input; the delay is constant.
- $y(t) = \alpha y(t-1) + \beta x(t-1)$
  Reason: Recursive equation with fixed delay; time-shifted input leads to time-shifted output.

Time-Variant Systems

- $y(t) = \int_{-\infty}^{t} \sin(ts)x(s)\, ds$
  Reason: The kernel $\sin(ts)$ explicitly depends on t, not only on $t-s$.

Other Model Classifications

- Single-scale vs multi-scale
- Deterministic vs stochastic
- Markovian vs non-Markovian
- Time-domain vs frequency-domain

System Identification

- Given data: $\{(x_i, y_i)\}_{i=1}^{N}$
- Objective: Find a function F such that $y_i \approx F(x_i)$ for all i

Focus: Discrete Linear Time-Invariant (LTI) Systems

- Discrete: $\mathcal{T} \subset \mathbb{Z}$
- Linear: $F(\alpha x_1 + \beta x_2) = \alpha F(x_1) + \beta F(x_2)$
- Time-invariant: $T_\tau F(x) = F(T_\tau x)$

Convolution Model (SISO, Discrete LTI)

$$y(t) = \sum_{s=-\infty}^{\infty} \rho(t-s)x(s) = \sum_{s=-\infty}^{\infty} \rho(s)x(t-s)$$

- $\rho(t)$: impulse response (IR) coefficients
- Interpretation: weighted sum over all past, present, and future values of $x(t)$

Why is ρ called IR Coefficients?

- Impulse input: $x(t) = \delta_{t,0} \Rightarrow y(t) = \rho(t)$

Delta Functions

- Kronecker delta: $\delta_{t,s} = \begin{cases} 1 & t = s \\ 0 & t \ne s \end{cases},\ t, s \in \mathbb{Z}$
- Dirac delta: $\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \ne 0 \end{cases},\ t \in \mathbb{R}$

Causality for Discrete LTI Systems

- Causal if $\rho(t) = 0 \quad \forall t < 0$
- Strictly causal if $\rho(t) = 0 \quad \forall t \le 0$

Example Causality Table

- $y(t) = \sum_{s=-\infty}^{t-1} x(s)$: Causal: True, Strictly Causal True
- $y(t) = x(t) - x(t-1)$: Causal True, Strictly Causal False
- $y(t) = x(t+1) - 2x(t) + x(t-1)$: Causal False, Strictly Causal False
- $y(t) = \sum_{s=-\infty}^{t-1} \left(\frac{1}{5}\right)^{t-s} x(s)$: Causal True, Strictly Causal True
- $y(t) = \frac{1}{2}y(t-1) + x(t-1)$: Causal True, Strictly Causal True

Stability – BIBO Stability

- $\|x\| := \sup_{t \in \mathcal{T}} |x(t)|$
- BIBO stable if:

$$\sup_{\|x\| \le 1} \|F(x)\| < \infty$$

BIBO Stability Theorem

- A discrete LTI system is BIBO stable if and only if:

$$\sum_{t \in \mathbb{Z}} |\rho(t)| < \infty$$

Difference Equations

- Difference equations provide a recursive relation between current output and past inputs/outputs.
- General form:

$$y(t) = \text{SomeFunction}[y(t-1), \ldots, y(t-\tau_1); x(t), x(t-1), \ldots, x(t-\tau_2)]$$

Causal LTI Systems as Difference Equations

$$y(t) = \sum_{s=1}^{\tau_1} a(s)y(t-s) + \sum_{s=0}^{\tau_2} b(s)x(t-s)$$

Terminology

- **Order:** τ_1, assuming $a(\tau_1) \neq 0$
- **Delay:** Smallest s such that $b(s) \neq 0$
- **Memory:** τ_2, assuming $b(\tau_2) \neq 0$

Difference Equation as Parametric Convolution

- Example: $y(t) = \alpha y(t-1) + \beta x(t-1),\ y(0) = 0$
- Equivalent convolution:

$$y(t) = \sum_{s=0}^{t-1} \beta \alpha^{t-s-1} x(s), \quad \rho(t) = \beta \alpha^{t-1}$$

Asymptotic Stability

- With $x(t) = 0\, \forall t$, the system:

$$y(t) = \sum_{s=1}^{\tau_1} a(s) y(t-s)$$

is asymptotically stable if

$$\lim_{t \to \infty} y(t) = 0 \quad \forall y(0) \in \mathbb{R}$$

State-Space Models (SSMs)

- Represent input-output relationships via internal (hidden) states.
- Avoid some limitations of convolution and difference models.
- General form (discrete):

$$h(t+1) = Ah(t) + Bx(t)$$
$$y(t) = Ch(t) + Dx(t)$$

- Dimensions:

$$x(t) \in \mathbb{R}^n, \quad y(t) \in \mathbb{R}^p, \quad h(t) \in \mathbb{R}^m$$
$$A \in \mathbb{R}^{m \times m}, B \in \mathbb{R}^{m \times n}, C \in \mathbb{R}^{p \times m}, D \in \mathbb{R}^{p \times n}$$

- Block matrix: $G = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$

Continuous-Time SSM

$$\dot{h}(t) = Ah(t) + Bx(t)$$
$$y(t) = Ch(t) + Dx(t)$$

- Continuous-time version of SSMs with $\mathcal{T} \subset \mathbb{R}$
- Properties: causal, linear, time-invariant

Why Use State-Space Models?

- Handle multi-dimensional inputs and outputs
- Explicitly model internal processes
- Support filtering, control, and identification
- Universality: can approximate any LTI system if $m \to \infty$

Limitations of State-Space Models

- Non-uniqueness of representation
- Performance depends on chosen representation
- Potential structural bias in modeling

Forms of State-Space Models (SSMs)

- SSMs are preserved under similarity transformation:

$$G = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad \longrightarrow \quad \tilde{G} = \begin{pmatrix} T^{-1}AT & T^{-1}B \\ CT & D \end{pmatrix}$$

- Canonical forms for SISO systems ($n = p = 1$):
  - Diagonal canonical form: A is diagonal
  - Observer canonical form: $C^\top = (1, 0, 0, \ldots, 0)$
  - Controller canonical form: $B^\top = (1, 0, 0, \ldots, 0)$

Definition (Minimal Realisation)

- A realization (A, B, C, D) is *minimal* if any other realization $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$ must satisfy:

$$\dim(\tilde{A}) \geq \dim(A)$$

Theorem (Stability of SSMs)

- An SSM is asymptotically stable if and only if all eigenvalues of A satisfy:

$$|\lambda| < 1$$

Definition of Estimation

- Estimation: inferring unobserved variable x from an information set y using a mapping and estimation criterion.
- Example:
  - Unobserved variable: x
  - Information set: y
  - Map: $y(t) = x + \varepsilon(t)$
  - Criterion: mean squared error
  - Estimator: $\frac{1}{T} \sum_{t=1}^{T} y(t)$

Goodness of Estimators

- Closeness to true parameter: $\mathbb{E}|\theta - \hat{\theta}|^2$
- Prediction error: $\mathbb{E}|\text{Prediction}(\theta) - \text{Prediction}(\hat{\theta})|^2$
- Bias/Consistency: $|\mathbb{E}\hat{\theta} - \theta|$
- Variance/Efficiency: $\mathbb{E}|\hat{\theta} - \mathbb{E}\hat{\theta}|^2$

The Parameter Estimation Problem

- Model: $y = F(x; \theta) \equiv F_\theta(x)$
- Goal: estimate θ from samples (x, y)
- Statistical version: $y \sim P(\cdot|x; \theta) \equiv P_\theta(\cdot|x)$

Least Squares Introduction

- Linear model: $y = \theta^T x + \varepsilon$
- Objective: minimize mean squared error:

$$L(\theta) = \mathbb{E}|y - \theta^T x|^2$$

- Empirical version:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} |y_i - \theta^T x_i|^2$$

Ordinary Least Squares Formula

- $X \in \mathbb{R}^{N \times n}$: matrix of inputs; $Y \in \mathbb{R}^{N \times p}$: output matrix
- OLS solution:

$$\hat{\theta} = (X^T X)^{-1} X^T Y$$

- Assumption: X has full column rank

Pseudo-Inverse

- If X does not have full column rank, find minimum norm solution:

$$\min_\theta |\theta|^2 \quad \text{subject to} \quad Y = X\theta$$

- Solution via pseudo-inverse: $\hat{\theta} = X^+ Y$

Measure of Goodness of Fit

- Residual error:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} |y_i - \theta^T x_i|^2$$

- R^2 coefficient:

$$R^2(\theta) = 1 - \frac{\sum_{i=1}^{N} |y_i - \theta^T x_i|^2}{\sum_{i=1}^{N} |y_i - \bar{y}|^2}$$

Weighted Least Squares

- Objective function:

$$L(\theta) = \text{Tr}\left[(Y - X\theta)^T W (Y - X\theta)\right]$$

- $W \in \mathbb{R}^{N \times N}$: symmetric positive definite weighting matrix
- Mahalanobis distance:

$$d(y, y') = (y - y')^T W (y - y')$$

- If $W = I$: ordinary least squares
- If $W = \text{cov}(Y)^{-1}$: generalized least squares

A Probabilistic View of Least Squares

- Model: $y \sim P(\cdot|x, \theta) = \mathcal{N}(\theta^T x, I)$
- Equivalently: $y = \theta^T x + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, I)$
- Least squares estimation is equivalent to Maximum Likelihood Estimation (MLE):

$$\max_\theta \sum_i \log P(y_i | x_i, \theta)$$

Moments of Distributions

- The n-th moment of P is:
$$[M_n(P)]_{i_1,\ldots,i_n} = \mathbb{E}_{y\sim P}[y_{i_1}\cdots y_{i_n}]$$
- Moment generating function:
$$M_P(s) = \mathbb{E}_{X\sim P}[e^{s^T X}]$$
- **Theorem (moments determine distributions):** If $M_n(P) = M_n(Q)$ for all $n \in \mathbb{N}$, then $P = Q$.

The Method of Moments

- For $p = 1$, match empirical and theoretical moments:
$$m_k(\theta) := \mathbb{E}_{y\sim P_\theta}[y^k] \approx \frac{1}{N}\sum_{i=1}^{N} y_i^k =: \hat{m}_k, \quad k = 1, 2, \ldots, K$$
- Estimation via:
$$\min_\theta \sum_{k=1}^{K} |m_k(\theta) - \hat{m}_k|^2$$

Advantages of the Method of Moments

- No need to know the full likelihood P_θ.
- Efficient when θ is low-dimensional.

The Identification Problem for SSMs

- Given input-output sequence (x, y), determine:
$$\text{order } m, \text{ matrices } A, B, C, D, \text{ and initial state } h(0) \in \mathbb{R}^m$$
- Note: matrices A, B, C, D are not unique for given x, y.

Can We Convert to a Regression Problem?

- Example SISO state space model:
$$h(t+1) = ah(t) + bx(t), \quad h(0) = 0$$
$$y(t) = ch(t) + dx(t)$$
- Express y in terms of x:
$$y(t) = \sum_{i=0}^{t-1} a^i bc\, x(t-i-1) + dx(t)$$
- Solving for a, b, c, d is non-convex and difficult.

Observability Problem

- Given SSM:
$$h(t+1) = Ah(t) + Bx(t)$$
$$y(t) = Ch(t) + Dx(t)$$
- Definition: The SSM (A, B, C, D) is observable if, for any input-output sequence (x, y), we can uniquely determine $h(0)$.

Observability Condition

- Define observability matrix:
$$O = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{m-1} \end{pmatrix} \in \mathbb{R}^{mp\times m}$$
- The SSM is observable if and only if O has full column rank m.

Controllability Problem

- Definition: The SSM (A, B, C, D) is controllable if for any target state $h^* \in \mathbb{R}^m$, there exists an input sequence x and time T such that $h(T) = h^*$.

Controllability Condition

- Define controllability matrix:
$$R = \begin{pmatrix} B & AB & \cdots & A^{m-1}B \end{pmatrix} \in \mathbb{R}^{m\times mn}$$
- The SSM is controllable if and only if R has full row rank m.

Duality Theorem

- (A, B, C, D) is observable $\iff$ $(A^\top, C^\top, B^\top, D^\top)$ is controllable.

Singular Value Decomposition (SVD)

- Any real matrix $M \in \mathbb{R}^{m\times n}$ can be written as:
$$M = U\Sigma V^\top$$
where:
  - $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$: orthogonal matrices
  - $\Sigma \in \mathbb{R}^{m\times n}$: diagonal matrix of singular values

Properties of SVD

- Singular values unique up to ordering.
- U and V are not unique.
- Singular values are square roots of eigenvalues of $M^T M$ or MM^T.
- Pseudo-inverse from SVD:
$$M^+ = V\Sigma^+ U^T \quad \text{where} \quad \Sigma^+_{ii} = \begin{cases} \frac{1}{\Sigma_{ii}} & \text{if } \Sigma_{ii} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$
- $\text{rank}(M)$ equals number of non-zero singular values.

Identification of SSMs: Key Steps

- Consider SSM:
$$h(t+1) = Ah(t) + Bx(t), \quad h(t) \in \mathbb{R}^m$$
$$y(t) = Ch(t) + Dx(t)$$
- 3 key steps:
  - Determination of order m
  - Estimation of matrices (A, B, C, D)
  - Estimation of states $\{h(t)\}$

Determination of Order

- Use observability matrix O:
$$m = \text{rank}(O)$$

Estimating Matrices (A, C)

- First p rows of O: $C = O[:p, :]$
- Next p rows: $CA = O[p:2p, :]$
- Compute A via:
$$A = (\text{pinv}(C))O[p:2p, :]$$

Estimating Matrices (B, D)

- From controllability matrix R:
$$R = \begin{pmatrix} B & AB & \cdots & A^{m-1}B \end{pmatrix}$$
- First p columns of R: $B = R[:, :p]$
- For D: from output equation $y(t) = Ch(t) + Dx(t)$.

Extended Observability and Controllability

- Extended matrices:
$$O_r = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{pmatrix}, \quad R_r = \begin{pmatrix} B & AB & \cdots & A^{r-1}B \end{pmatrix}$$
- Use $r \gg m$ for robustness.

Two Types of Methods

- Realisation method: from IR function $\rho(t)$ using SVD.
- Direct method: from input-output data using projection.

Realisation Method IR Function

- Assume $h(0) = 0$, then:
$$\rho(t) = \begin{cases} D & t = 0 \\ CA^{t-1}B & t > 0 \end{cases}$$
- So, $D = \rho(0)$ directly.

Hankel Matrix and Rank

- Build block Hankel matrix:

$$\mathcal{H} = \begin{pmatrix} \rho(1) & \rho(2) & \rho(3) & \cdots \\ \rho(2) & \rho(3) & \rho(4) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

- $\operatorname{rank}(\mathcal{H}) = m$, order of the SSM.
- Factorization:

$$\mathcal{H} = O_\infty R_\infty$$

- Input-output relation:

$$y(t) = Dx(t) + \mathcal{H}_1 x_-$$

  where $\mathcal{H}_1$ is the first block row.

Realisation Identification Algorithm

1. Construct Hankel matrix $\mathcal{H}_{l_1,l_2}$
2. Compute SVD:

$$\mathcal{H}_{l_1,l_2} = \begin{pmatrix} U_m & U_s \end{pmatrix} \begin{pmatrix} \Sigma_m & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_m^T \\ V_s^T \end{pmatrix}$$

3. Estimate:

$$O = U_m \Sigma_m^{1/2} T, \quad R = T^{-1} \Sigma_m^{1/2} V_m^T$$

Direct Method (Sketch)

$$y(t) = \sum_{s=0}^{t} \rho(t-s) x(s)$$

$$\begin{pmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+r-1) \end{pmatrix} = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{pmatrix} h(k) + G_r \begin{pmatrix} x(k) \\ x(k+1) \\ \vdots \\ x(k+r-1) \end{pmatrix}$$

$$Y = O_r H + G_r X$$

$$O_r = Y X^\perp \quad \text{with} \quad X X^\perp = 0$$

Subspace Identification (Up to Similarity)

$$O_r R_r = O_r T T^{-1} R_r$$

Identification of Nonlinear Systems
Two Approaches

- Linearize and proceed
- Function approximation via optimization

$$x(t+1) = f(x(t)), \quad x(t) \in \mathbb{R}^n$$

Linear Basis Models

$$f(x) \approx f_\theta(x) = \theta^\top \varphi(x) = \sum_{j=1}^{m} \theta_j \varphi_j(x)$$

$$\hat{\theta} = \arg\min_\theta \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=0}^{T-1} \left| x_i(t+1) - \theta^\top \varphi(x_i(t)) \right|^2$$

Issues

- Choice of dictionary
- Overfitting
- Curse of dimensionality ($m \propto \exp(n)$)
- Lack of physical interpretability

Regularisation Based on Sparsity

- Seek approximation: $f(x) \approx \theta^\top \varphi(x)$
- Enforce many $\theta_j = 0$ for parsimony $\to$ sparse representation

Advantages:

- Physical interpretation: $\theta_j \neq 0$ indicates relevant features.
- Better generalisation.
- More robust to noise.

Forms of Regularised Problems:

1. $\min_\theta L(\theta) + \lambda R(\theta)$
2. $\min_\theta L(\theta)$ s.t. $R(\theta) \leq \mu$
3. $\min_\theta R(\theta)$ s.t. $L(\theta) \leq \epsilon$

Ideal Regulariser:

$$R(\theta) = \|\theta\|_0 = \sum_{j=1}^{m} 1_{\theta_j \neq 0}$$

- Hard combinatorial optimisation
- Want convex alternatives

ℓ^p Balls:

$$B_p^m := \left\{ \theta \in \mathbb{R}^m : \|\theta\|_p \leq 1 \right\}, \quad \|\theta\|_p = \left(\sum_{j=1}^{m} |\theta_j|^p \right)^{1/p}$$

- Convex for $p \geq 1$

ℓ^p-Regularised Linear Regression:

$$\min_\theta \frac{1}{2N} \|Y - X\theta\|^2 \quad \text{s.t. } \|\theta\|_p \leq \mu$$

ℓ^1-Regularised (Lasso):

$$\min_\theta \frac{1}{2N} \|Y - X\theta\|^2 + \lambda \|\theta\|_1$$

Cyclic Coordinate Descent:

$$\theta_j = S_\lambda \left((X_j)^\top r^{(j)} \right), \quad r^{(j)} = y_i - \sum_{k \neq j} x_{ik} \theta_k$$

- Sequentially update $j = 1, 2, \ldots, m$
- Or use random coordinate descent

Sparse Identification of Dynamical Systems (SINDy):

- Identify $x(t+1) = f(x(t))$
- Use ℓ^1-regularised linear basis model: $f(x) \approx \theta^\top \varphi(x)$

Other Parameterisations of Dynamics:

- Neural networks for f
- Kernel-based methods
- Hamiltonian-type systems
- Dissipative-type systems
- Many others

Part 2

What is dimensionality reduction?

- Process of describing high-dimensional data by a lower-dimensional representation.
- Essential features of data are retained.

In the context of dynamical systems:

- Visualising dominant features driving dynamics.
- Reducing computational complexity of simulations.
- Studying key underlying dynamical mechanisms.
- Improving data-driven prediction and control.

Spatiotemporal data described by PDEs

$$\partial_t u = L(u)$$

where:

- $u = u(t, x)$ is solution at time $t \in \mathbb{R}$ and location $x \in \mathbb{R}^n$.
- ∂_t is time derivative.
- L is a general operator: may include spatial derivatives, nonlinear terms etc.

$$L(u) = F(x, u, \partial_x u, \partial_x^2 u, \ldots)$$

Discretisations of PDEs

- $u(t, \cdot)$ is infinite-dimensional.
- Discretise space into n grid points $\{x_1, \ldots, x_n\}$ where $x_{i+1} - x_i = \delta x \ll 1$.

$$u(t, \cdot) \approx (u(t, x_1), \ldots, u(t, x_n))^T \in \mathbb{R}^n$$

Finite differences: spatial derivatives

$$\partial_x u(t,x)\big|_{x_i} \approx \frac{u(t,x_{i+1}) - u(t,x_{i-1})}{2\delta x}$$

$$\partial_{xx} u(t,x)\big|_{x_i} \approx \frac{u(t,x_{i+1}) - 2u(t,x_i) + u(t,x_{i-1})}{\delta x^2}$$

- PDE converted to ODE system in $\mathbb{R}^n$.
- Accurate approximation requires $n \gg 1$: high-dimensional dynamics!

Separation of Variables (classic PDE method)

- Guess solution as product form: $\varphi(t,x) = v(t)w(x)$.
- Substitute into $\partial_t u = L(u)$ to obtain ODEs for $v(t)$ and $w(x)$.
- General solution as sum of products:

$$u(t,x) = \sum_{k=1}^{\infty} a_k v_k(t) w_k(x)$$

- Constants a_k determined from boundary/initial conditions.

Representing Functions by Truncated Expansions

- Separation of variables gives expansion:

$$u(t,x) = \sum_{k=1}^{\infty} a_k v_k(t) w_k(x)$$

- Both sides are infinite-dimensional.
- Under general conditions, we truncate:

$$u(t,x) \approx \sum_{k=1}^{N} a_k v_k(t) w_k(x)$$

- For large enough N, approximation is accurate.

Proper Orthogonal Decomposition (POD)

- Generalises Eckart-Young theorem (SVD) to spatio-temporal data.
- Use averaging over time domain.

Low Rank Approximation of Matrices

- Given $A \in \mathbb{C}^{m\times n}$, define:

$$e_r(A) = \min_{B\in\mathcal{M}_r} \|A - B\|_F$$

- Key questions:
  - What is A_r?
  - How does $e_r(A)$ behave?

Eckart-Young Theorem

- $e_r(A) = \left(\sum_{j=r+1}^{\min(m,n)} \sigma_j^2(A)\right)^{1/2}$
- Optimal approximation:

$$A_r = U\Sigma_r V^*$$

Back to Spatio-Temporal Data

- Decomposition:

$$u(t,x) = \sum_{k=1}^{K} a_k v_k(t) w_k(x)$$

- Discrete grid: Time: $t = t_1, \ldots, t_m$; Space: $x = x_1, \ldots, x_n$
- Matrix representation: $U_{ij} = u(t_i, x_j)$

Low-Rank Approximation Viewpoint

- Formula:

$$U = \sum_{k=1}^{K} a_k v_k w_k^T$$

- Optimal spatio-temporal decomposition = optimal low-rank approximation.

Time Domain as Averaging Set

- Time average:

$$\langle u\rangle = \frac{1}{T}\int_0^T u(t,\cdot)dt$$

- Correlation operator:

$$(R_w)(x) = \int_\Omega C(x,z)w(z)dz$$

Properties of POD

- POD bases: $\{w_k\}$
- Function space:

$$\mathcal{S} = \left\{\sum_{k=1}^{\infty} a_k w_k \;:\; \sum_k |a_k|^2 < \infty\right\}$$

- Approximation of any $f \in \mathcal{S}$ with arbitrary precision.
- Modes with nonzero eigenvalues reconstruct original data.
- In general, span of $\{w_k\}$ may not be dense in $L^2(\Omega)$.
- POD satisfies optimality property (Eckart-Young generalisation).

Theorem (Optimality of the POD)

- Let $u(t,\cdot) \in L^2(\Omega)$, $t \in [0,T]$, and $\{w_k\}$ be orthonormal eigenfunctions of the covariance operator.
- POD decomposition:

$$u(t,x) = \sum_k a_k v_k(t) w_k(x)$$

- Any other decomposition:

$$u(t,x) = \sum_k \tilde{a}_k \tilde{v}_k(t) \tilde{w}_k(x)$$

- Properties:
  - Temporal coefficients uncorrelated:

$$\frac{1}{T}\int_0^T v_k(t)v_\ell(t)dt = \delta_{k\ell}$$

    where λ_k is the k-th eigenvalue.
  - Optimality: for any $N \geq 1$,

$$\frac{1}{T}\int_0^T\int_\Omega \left|u(t,x) - \sum_{k=1}^{N} a_k v_k(t) w_k(x)\right|^2 dxdt$$

    is minimized.

The POD Algorithm

- Given snapshots:

$$u_i = (u(t_i,x_1), u(t_i,x_2), \ldots, u(t_i,x_n))^T \in \mathbb{C}^n$$

- Form matrix:

$$U = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix} \in \mathbb{C}^{m\times n}$$

- SVD:

$$U = V\Sigma W^H$$

- Truncate SVD for POD basis:

$$U \approx \sum_{k=1}^{r} s_k v_k w_k^H$$

Applications of POD

- Visualize dominant modes.
- Reduced models for complex dynamics.

POD: Optimality and Limitations

- POD minimizes:

$$e_r = \frac{1}{T}\int_0^T\int_\Omega \left|u(t,x) - \sum_{k=1}^{r} \sigma_k v_k(t) w_k(x)\right|^2 dxdt$$

- But: POD modes do not always evolve independently.

Dynamical Invariants

- Given PDE: $\partial_t u = L(u)$.
- POD extracts modes $v_1(t), w_1(x)$, but in general:

$$u(t,x) \not\approx a(t) w_1(x)$$

for any $a(t)$.

Dynamic Mode Decomposition (DMD)

- Spatio-temporal data:

$$U = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{m-1} \end{pmatrix} \in \mathbb{C}^{m\times n}$$

- Form input/output pairs:

$$X = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{m-2} \end{pmatrix}, \quad Y = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{m-1} \end{pmatrix}$$

- Fit linear model:

$$A = \arg\min_B \|Y - XB^T\|_F, \quad u_{t+1} \approx A u_t$$

- Decompose A: if diagonalizable:

$$u(t) = \sum_{j=1}^{n} c_j \lambda_j^t w_j$$

where $\{w_j\}$ are eigenvectors of A, λ_j eigenvalues.

Some Properties of the Dynamic Mode Decomposition (DMD)

- If $u(0) = c w_k$, then $u(t) = c\lambda_k^t w_k$ — the DMD modes evolve independently.
- More generally: let $\Gamma \subset \{1,2,\ldots,n\}$ and assume

$$u(0) \in \text{Span}\{w_j : j \in \Gamma\} \implies u(t) \in \text{Span}\{w_j : j \in \Gamma\} \text{ for all } t \geq 0.$$

- The DMD modes are not necessarily orthogonal and are not the same as the POD spatial modes.

Limitations of the DMD

- DMD fits a linear model $u(t+1) = Au(t)$.
- This is often poor for highly nonlinear systems.
- Question: Can we turn a nonlinear system into a linear one and apply DMD?

Linearisation by Taylor Expansions

$$u(t+1) = f(u(t)), \quad u(t) \in \mathbb{C}^n.$$

Let $u^* \in \mathbb{C}^n$ be an equilibrium point: $f(u^*) = u^*$.

$$f(u) \approx f(u^*) + Df(u^*)(u - u^*) + \mathcal{O}(|u-u^*|^2).$$

Defining $v(t) = u(t) - u^*$ gives approximate linear dynamics:

$$v(t+1) = Df(u^*)v(t).$$

Two Different Views of Nonlinear Systems

$$u(t+1) = f(u(t)), \quad u(t) \in \Omega \subset \mathbb{C}^n.$$

- State space view.
- Function space view.

The Space of Observables

- Define $\mathcal{H} \subset$ functions on Ω.
- For example: $L^2(\Omega) = \left\{\varphi : \Omega \to \mathbb{C}, \int_\Omega |\varphi(u)|^2 du < \infty\right\}$.
- We study evolution of observables $\varphi \in \mathcal{H}$.

Linearising the Dynamics in the Space of Observables

$$u(t+1) = f(u(t))$$
$$\varphi(t+1) = \mathcal{K}\varphi(t), \quad \mathcal{K} : L^p(\Omega) \to L^p(\Omega)$$
$$\mathcal{K}\varphi = \varphi \circ f.$$

Did We Really Simplify the Problem?

- Koopman operator $\mathcal{K}$ is infinite-dimensional and hard to compute.
- We've traded nonlinear finite-dimensional for linear infinite-dimensional dynamics.
- **Key point: dimensionality reduction is easier for linear systems.**

Koopman Operators and Nonlinear DMD
Koopman Invariant Subspace

- Recall: A subspace $\mathcal{V}$ of a vector space $\mathcal{H}$ is closed under addition and scalar multiplication:

$$\varphi_1, \varphi_2 \in \mathcal{V} \implies a\varphi_1 + b\varphi_2 \in \mathcal{V}, \quad a, b \in \mathbb{C}.$$

- $\mathcal{V} \subset \mathcal{H}$ is invariant under K if:

$$\varphi \in \mathcal{V} \implies K\varphi \in \mathcal{V}.$$

- Note: $\mathcal{V}$ can be much smaller than $\mathcal{H}$.

Parameterising a Koopman Invariant Subspace

- Use dictionary functions $\varphi_1, \ldots, \varphi_m \in \mathcal{H}$.
- Define: $\mathcal{V} = \text{span}\{\varphi_1, \ldots, \varphi_m\}$.
- Requirements:
  - $\mathcal{V}$ contains full-state observables: $\varphi_i(u) = u_i$.
  - $K\mathcal{V} \subset \mathcal{V}$.
- Approximate satisfaction is sufficient for numerics.

$$\varphi(t+1) = K\varphi(t) \equiv \varphi(t) \circ f.$$

We get a finite-dimensional linear system, and apply DMD to extract modes.
Extended DMD (EDMD) Algorithm

1. Dataset: $\{u^{(l)}(t)\}_{t=0,i=1}^{T-1,N}$, where $u^{(l)}(t) \in \mathbb{C}^n$.
2. Pick dictionary functions $\varphi_1, \ldots, \varphi_m$.
3. Form data matrices:
$$\Phi_{ij}(t) = \varphi_j(u^{(i)}(t)), \quad \Phi^+_{ij}(t) = \varphi_j(u^{(i)}(t+1)).$$
4. Solve:
$$\min_{K \in \mathbb{C}^{m\times m}} \frac{1}{N(T-1)} \sum_{t=0}^{T-2} \|\Phi^+(t) - \Phi(t)K^\top\|_F^2.$$
5. Compute eigen-decomposition of $K^\top$, yielding eigenvectors W_j and eigenvalues λ_j.
6. Koopman spectral decomposition:
$$\psi(u(t)) = \sum_{j=1}^{m} \lambda_j^t c_j W_j^\top \varphi_j(u(0)).$$

Manifolds

- A manifold locally resembles Euclidean space.
- Examples:
  - Subspaces of Euclidean space
  - The Earth
  - Rank-1 matrices, SPD matrices

Manifold Hypothesis

- Real-world high-dimensional data often lies on a lower-dimensional manifold.

A Difficulty: Non-linearity

- After assuming the manifold hypothesis, the challenge is identifying the manifold.
- Manifolds can be non-linear, curved, and complicated.

Two Types of Approaches

1. Transform data to apply linear methods:
   - Linear basis models
   - Koopman operator
2. Develop non-linear methods:
   - Neural networks
   - Heuristic methods

Review of PCA

- Data: $\{x_1, \dots, x_N\}$, $x_i \in \mathbb{R}^n$, zero-mean.
- Sample covariance:
$$S = \frac{1}{N} X^T X = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T$$
- Eigen-decomposition: $S = U \Sigma U^T$
- Principal components: columns of U
- Dimensionality reduction: $X \mapsto Z_r$ (first r columns of U)
- Reconstruction: $X_r = Z_r U_r^T$

The Dual Form of PCA

- Usually $N \gg n \gg r$.
- For high-dimensional data, use:
$$G = \frac{1}{N} X X^T \in \mathbb{R}^{N \times N}$$
- This gives the dual PCA formulation.

Kernel Methods

- Replace inner product $x_i^T x_j$ with kernel $k(x_i, x_j)$.
- Properties of kernel $k \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$:
  - Symmetry: $k(x_i, x_j) = k(x_j, x_i)$.
  - Positive semi-definiteness: $\sum_{i,j} c_i c_j k(x_i, x_j) \geq 0$.
- Applying dual PCA with kernel gives **Kernel PCA**.

Kernels and Infinite-Dimensional Feature Maps

- Every positive semi-definite kernel corresponds to a feature map:
$$\varphi \colon \mathbb{R}^n \to \mathcal{H}$$
- Inner product:
$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$$
- Kernel PCA lifts data to $\varphi(X)$ and applies linear PCA.

Locally Linear Embedding (LLE)

- Manifolds are locally flat (locally linear).
- Use local linear approximations to progressively build global embedding.
- LLE performs these local-to-global embeddings automatically.

Some Extensions of the LLE
Multidimensional Features

- Previously assumed each $x_i \in \mathbb{R}$.
- If $x_i \in \mathbb{R}^n$, then there will be n sets of weights.
- A common solution: combine weights into one set by averaging over all features.

Kernel LLE

- Construction of k-NN graph uses Euclidean distance.
- If Euclidean distance is not appropriate, replace with kernel-distance using:
$$\|\varphi(x_i) - \varphi(x_j)\|_{\mathcal{H}}^2 = R(x_i, x_i) - 2R(x_i, x_j) + R(x_j, x_j)$$

Motivation for Diffusion Maps

- Locality is key to manifold learning for non-linear dimensionality reduction.
- Key issues:
  - What is the right scale of locality?
  - How to deal with multi-scale structures?
  - How does local connectivity carve out the manifold?

Connecting Scales Through Dynamics

- Diffusion maps connect spatial scales by converting them into temporal scales of a dynamical process in space.

Preliminary: Markov Chains

- Discrete-time Markov chain: $X(0), X(1), \dots$
- Markov property:
$$\mathbb{P}[X(t+1) = j \mid X(t) = i, X(t-1), \dots] = \mathbb{P}[X(t+1) = j \mid X(t) = i]$$
- Transition matrix P defined by:
$$P_{ij} = \mathbb{P}[X(t+1) = j \mid X(t) = i]$$

Properties of Markov Chains

- Entries of P are non-negative, rows sum to 1 (stochastic matrix).
- t-step transition matrix is P^t.
- 1 is an eigenvalue of P, others have magnitude ≤ 1.
- Stationary distribution π: $\pi P = \pi$.
- If unique, Markov chain converges to π.

Markov Chain on Connectivity Graphs

- Dataset $\{x_1, \dots, x_N\}$, kernel $k \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$.
- Build Markov chain using:
$$P_{ij} \propto k(x_i, x_j)$$
- Transition probabilities encode closeness at different time scales.
- Eigenvectors of P give coordinates for dimensionality reduction.

Summary of Diffusion Map Algorithm

1. Build similarity matrix L (e.g., RBF kernel).
2. Form transition matrix $P = D^{-1} L$.
3. Compute spectral decomposition of P; use eigenvectors for embedding.