DATA-DRIVEN DYNAMICAL SYSTEMS

systems like $\dot{x} = f(x,t)$ $\begin{cases} t \text{ dep} \rightarrow \text{non-autonomous} \\ \text{not} \rightarrow \text{autonomous} \end{cases}$

there might be $\begin{cases} u & \text{control input} \\ \beta & \text{parameters} \end{cases}$ $\Rightarrow$ $f(x,t,u;\beta) = \dot{x}$

state, time, actuation (control), parameters

dynamics (vector field)

DYNAMICAL SYSTEM (anatomy)

Writing the dynamics in terms of first principles is complicated (non-starter) in a lot of fields. $\rightarrow$ you don't write an equation for brain!

FIRST PRINCIPLES $\rightarrow$ DATA DRIVEN we have a lot of data!

OBS linear systems are completely understood & characterized: $\dot{x} = Ax$ ✓
so we will care of non linear f

OBS the dynamics is not known

(non linear f, dynamics not known:) challenging

$\oplus$ $\begin{cases} \text{high dimensionality} & \text{noise, stochastics} & \text{uncertainty} \\ \text{chaos, transients} & \text{multiscale structures} & \end{cases}$

Uses of models

1. Future state prediction
2. Design / optimization
3. Control
4. Interpretability & physical intuition

Techniques

- Regression: linear, sparse
- Genetic programming
- Neural networks & Deep learning

EX for nonlinear f $\rightarrow$ find coordinate transformation $\varphi(x)$: $\frac{d}{dt}\varphi(x) = \lambda\varphi(x)$
so the new coordinate has linear dynamics (Koopman analysis)

Anatomy of dynamical systems → already seen

Discrete-time dynamical systems

always, but going back? not always!

$\dot{x} = f(x(t))$ continuous time $\Rightarrow$ discrete time $x_{k+1} = F(x_k)$

these systems are more generic, k can not be time

We can sample x in time: $x_k = x(k\Delta t)$, $x_{k+1} = \underbrace{x_k + \int_{k\Delta t}^{(k+1)\Delta t} f(x(\tau))\, d\tau}_{\text{FLOW MAP} \quad F_{\Delta t}}$

EX it is hard to go discrete $\Rightarrow$ continuous for population data (seasons, ...)

OBS approximately

$\dot{x} = f(x) \quad \Rightarrow \quad \dfrac{x_{k+1} - x_k}{\Delta t} \simeq f(x_k)$ (Forward Euler integrator)

$x_{k+1} = x_k + \Delta t\, f(x_k)$

approximates the Flow Map

EX logistic map $x_{k+1} = \beta x_k (1 - x_k)$ $\beta \in [0, 4]$ $x_0 = 0.5$

population model carrying capacity factor

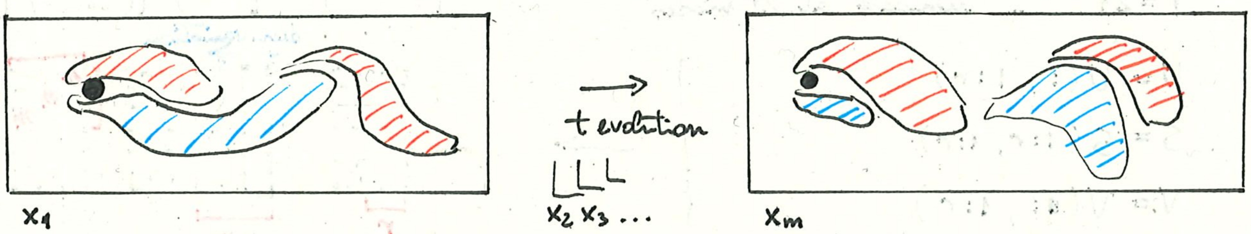
Dynamic model decomposition (DMD)

introduced by Schmid in fluid dynamics, extended to Koopman theory later on

gives a coupled system of spatial-temporal modes

is a purely data-driven method

EX movie of flow around a cylinder, to study vorticity



we reshape the frames as big column vectors & create the data as

$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_{m-1} \\ | & | & & | \end{bmatrix} \qquad X' = \begin{bmatrix} | & | & & | \\ x_2 & x_3 & \dots & x_m \\ | & | & & | \end{bmatrix}$$

OBS tipically the matrices are skinny, i.e. few t measurements, lots of spatial

DMD finds a best fit linear operator $A = X'X^+$ $(X' \approx AX)$ to make a future prediction $x_{k+1} = A x_k$.

OBS actually we don't want to compute A, but get the dominant eigenvalues & eigenvectors of A ↑ this is huge!

↳ these can be reshaped as eigen-fluid-flows!

To compute DMD:

columns of U and V are the singular ~~values~~ vectors of X (singular vectors) or POD modes

→ singular value decomposition $\quad X = U \Sigma V^* \qquad X' = A \cdot U \Sigma V^*$

diagonal of Σ are the singular values of X

we do not compute A from this!

instead we project A on the dominant singular vector $U^* A U = \tilde{A}$

which can be computed as $\tilde{A} = U^* \cdot X' V \Sigma^{-1}$

OBS it can be proven that non-zero eigenvalues of $\tilde{A}$ are the same of A

→ eigenv* $\quad \tilde{A} W = W \Lambda \qquad \Lambda \rightarrow$ eigenvalues $\quad W \rightarrow$ eigenvectors

amplitude of modes (init.)

→ DMD modes: $\quad \phi = AU \cdot W = X' V \Sigma^{-1} \cdot W \quad \Rightarrow$ prediction: $\hat{X}(k\Delta t) = \phi \Lambda^t b_0$

[Tu, JCD 2014]

notes on DMD code

economy

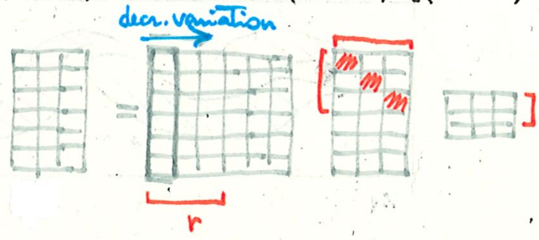
```
[U,S,V] = svd(X,'econ')
```

$$X = USV^*$$

$$\begin{array}{c} \\ n \end{array}\overset{m}{\left[\begin{pmatrix} X \end{pmatrix}\right]} = \begin{array}{c} \\ n \end{array}\overset{n}{\left[\begin{pmatrix} U \end{pmatrix}\right]} \begin{array}{c} \\ n \end{array}\overset{m}{\left[\begin{pmatrix} \Sigma \end{pmatrix}\right]} \begin{array}{c} \\ m \end{array}\overset{m}{\left[\begin{pmatrix} V^* \end{pmatrix}\right]}$$

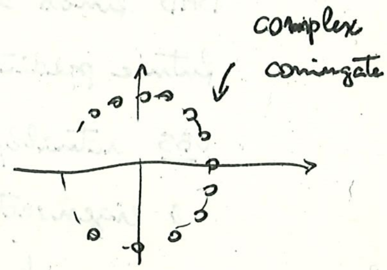
```
r=21        % truncate at 21 modes

U = U(:,1:r)
S = S(1:r,1:r)
V = V(:,1:r)
Atilde = U'·X'·V·inv(S)
[W,eigs] = eig(Atilde)
Phi = X'·V·inv(S)·W
```

→




```
% plot DMD modes
  reshape(real(φ(:,i)))
  reshape(img(φ(:,i)))
```

```
% plot DMD spectrum
  scatter(real(diag(eigs)), img(diag(eigs)))
```



Sparse DMD

DMD requires a large amount of data to collect & process

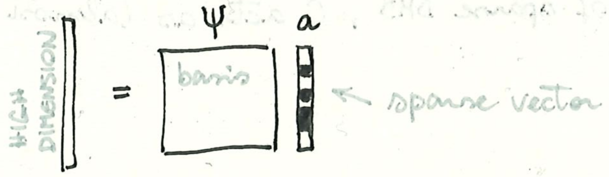
Goal: reconstruct full DMD from heavily subsampled measurements

EX particle image velocimetry (PIV) has limited bandwidth

OBS **Reconstruction by Compressed Sensing**

It turns out that some high dimensional signals $\underline{x}$ appear sparse in an appropriate transformed basis (compression). Instead of collecting all data $\underline{x}$, the idea of sparse sampling is to collect many smaller initial measurements $\underline{y}$ and infer the sparse coefficients in the transformed basis.

$\underline{x} = \Psi \, \underline{a}$



low dim. measures $\boxed{\underline{y}}^{\,IN} = \Phi \, \underline{x} = \Phi \Psi \underline{a} = \Theta \boxed{\underline{a}}^{\,OUT}$

random measurement matrix (incoherent wrt transform basis Ψ)

To reconstruct: minimize $\|a\|_1 \;:\; y = \Theta a$

Before we had to do this by brute-force combinations.

Therefore:

COMPRESSED SENSING DMD

$$\begin{tikzcd} \cancel{X,\ X'} \arrow[r, "DMD"] \arrow[d, "compress"] & \Lambda_x,\ \phi_x \\ Y\ \ Y' \arrow[r, "compressed\ DMD"] & \Lambda_y,\ \phi_y \arrow[u, "\ell_1\ min"'] \end{tikzcd}$$

ACCESS ONLY TO SPARSE DATA

the eigenvalues are close within machine precision!
BUT ℓ_1 minimization is expensive!

ACCESS TO FULL DATA ($\Rightarrow$ opti.)

if we have access to full dimensional data $\rightarrow$

$$\begin{tikzcd} X\ \ X' \arrow[d, "compress"] & \Lambda_x,\ \phi_x \\ Y\ \ Y' \arrow[r, "compressed\ DMD"] & \Lambda_y,\ \phi_y \arrow[u, "NO\ min!"'] \end{tikzcd}$$

use the formula

$\tilde{\Phi}_x = X' V_Y \Sigma_Y^{-1} W_Y$

COMPRESSED DMD

$\rightarrow$ the SVD is done in the compressed space!

Why compressed DMD works ?

PROP1 DMD is invariant to right unitary transformations

↑ because SVD is so

⇒ swapping (the same way) columns of X & X' does not change DMD

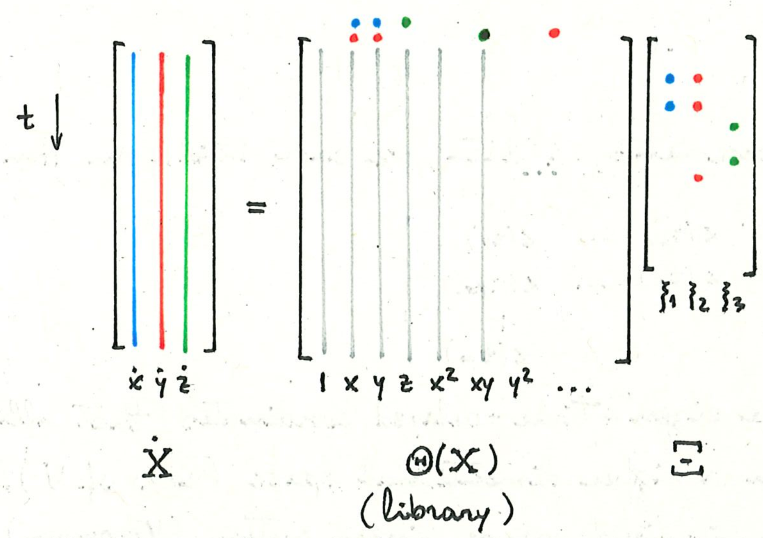
PROP2 DMD is (mostly) invariant to left unitary transformations

↑ swap rows

⇒ DMD of X, X' is related to any unitary transformations of data, like discrete FT

In our case C is not unitary (it is a heavy-compression) matrix. But from the theory of sparse DMD, C acts as (almost) unitary on Ψ (sparsifying basis)

Sparse Identification of Nonlinear Dynamics (SINDy)



this is a GENERALIZED LINEAR REGRESSION

Target: find the sparse set of Θ columns which describes the dynamics of data.

↓

We need a sparse optimization algorithm $\forall$ $\dot X$ column.

Data SINDy most of the time requires relatively clean data, sampled quickly. There have been advancements that relax some of those constraints.

→ works with noisy data (if you have plenty), you could compute derivatives, for example, with TV diff (or any other noise-proof algo)

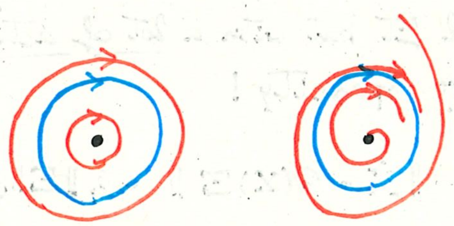
→ there is a "weaker" formulation of SINDy : integral SINDy

$$\int \left[\dot X\right] = \int \left[\ \Theta(X)\ \right]\left[\ \right]$$

the integration of data allows to account for larger amounts of noise

→ Θ must be well-conditioned ⇒ more data that excites transients in dyn. sys.

Ex



$\dot x = \omega y$
$\dot y = -\omega x$

$\dot x = \mu x + \omega y + A x(x^2+y^2)$
$\dot y = -\omega x + \mu y + A y(x^2+y^2)$

(cubic Hopf)

 → not disambiguating (same behav.)

 → disambiguating! (take more initial conditions)

⇕

EXCITE REGIONS OF PARAMETER SPACE

Time Delay Coordinates

If you have access to only few measurements, or data, we can build an Henkel matrix

$$H = \begin{bmatrix} x(t_1) & x(t_2) & \cdots & x(t_p) \\ x(t_2) & x(t_3) & \cdots & x(t_{p+1}) \\ \vdots & & & \vdots \\ & & & x(t_m) \end{bmatrix}$$

and performing a SVD we discover eigen-time-delayed coordinates that allow you to embed the x measurement in a higher dimentional space (rows of V). Applying SINDy to these coordinates we find sparse linear system (Koopman).

SINDy with control

You can extend the candidate library Θ to include functions of a control input u (and cross terms). This helps to disambiguate internal dynamics and the effect of actuation & control.

$$\Theta(X) \rightarrow \Theta(X, u)$$

The same reasoning can be applied to parameters: $\Theta(X) \rightarrow \Theta(X, \mu)$

Optimization algorithm

$\dot{X} = \Theta(X)\,\Xi$, take pseudo inverse of Θ → least square regression

⇒ good model fit but with a lot of active terms

WE WANT sparsity!

OBS we don't use LASSO typically $\|\dot{X} - \Theta(X)\Xi\| + \lambda \|\Xi\|_0$ (not usually the best)

→ SEQUENTIAL THRESHOLD LEAST SQUARES

- init with a least squares regression & hard-threshold small coefficients
- iteratively apply least squares ~~on~~ remaining coff. + thresholding, ...

OBS Sparse Relaxed Regularized Regression (SR3, give it a look!)

OBS you can force some prior knowledge in the optimization algo

KOOPMAN OPERATOR THEORY

Let us focus on nonlinear dynamics.

Bernard Koopman in 1931 introduced this formalism. He showed that Hilbert space of all possible measurements of a dynamical system can be described by a ∞-dim linear operator that evolves those measurements forward in time.
Von Neumann & Birkhoff used Koopman's operator as a missing piece in their theories.

In the modern perspective, measurement is synonim of data.

In dynamics: $\frac{d}{dt}\underline{x} = f(\underline{x}) \Rightarrow F_t(\underline{x}(t_0)) = \underline{x}(t_0+t) = \underline{x}(t_0) + \int_{t_0}^{t_0+t} f(\underline{x}(\tau))\,d\tau$

$\Downarrow$

$\underline{x}_{k+1} = F_t(\underline{x}_k)$

DISCRETE TIME-UPDATE

(this is the geometric Poincaré perspective, to look for separators, invariant manifolds, attractors, ...)

very classic!

In Koopman analysis we talk about measurements g of the systems, ($g \in$ Hilbert sp.) such that

$$K_t\, g = g \circ F_t$$

K is ∞-dim and linear

K_t advances all possible measurements one step in the future

$\Updownarrow$

$$K_t\, g(\underline{x}_k) = g(F_t(\underline{x}_k)) = g(\underline{x}_{k+1}) \qquad \text{(discrete-time update)}$$

TARGET: find a close finite-dim approximation of this ∞-dim operator

We want to find eigenfunctions of such operator, to span a Koopman invariant subspace:

$$g = \sum_{k=1}^{\infty} a_k y_k \rightarrow g = \sum_{k=1}^{m} a_k y_{s_k}, \quad Kg = \sum_{k=1}^{m} \beta_k y_{s_1} \qquad \text{\& } K \text{ is a matrix}$$

EX: nonlinear dynamics

$$\begin{cases} \dot{x}_1 = \mu x_1 \\ \dot{x}_2 = \lambda(x_2 - x_1^2) \end{cases}$$

$\Rightarrow$ Koopman linear system

$$\frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \mu & & \\ & \lambda & -\lambda \\ & & 2\mu \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad \text{for} \quad \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \end{bmatrix}$$

$\Rightarrow$ eigen-observables $\quad \varphi_\alpha(\underline{x}) = \xi_\alpha \underline{y}(\underline{x})$, where $\xi_\alpha K = \alpha \xi_\alpha$

$$\varphi_\mu = x_1, \quad \varphi_\lambda = x_2 - b x_1^2, \quad \text{with } b = \frac{\lambda}{\lambda - 2\mu}$$

EX $\frac{d}{dt} x = x^2$ has no easy closure! we can't represent it on finite dim! (Poincaré)

Polynomials might be bad basis!

Koopman eigenfunctions define invariant subspaces

Starting from $\frac{d}{dt} \underline{x} = f(\underline{x})$ (nonlinear dynamics), we want to find the new coordinates $\varphi(\underline{x})$ such that $\frac{d}{dt}\varphi(\underline{x}) = \lambda \varphi(\underline{x})$ (linear dynamical system).
From chain rule $\frac{d}{dt}\varphi(\underline{x}) = \nabla\varphi(\underline{x}) \cdot \dot{\underline{x}} \overset{①}{=} \nabla\varphi(\underline{x}) \cdot f(\underline{x})$

$$\Rightarrow \quad \nabla\varphi(\underline{x}) \cdot f(\underline{x}) = \lambda\, \varphi(\underline{x})$$

GENERATOR FOR KOOPMAN EIGENFUNCTIONS

ES the previous EX is solved by using Laurent series $\Rightarrow \varphi(\underline{x}) = e^{-1/x}$.

But what do we do when we don't have analytical solutions, but we have data?

DMD! $\rightarrow$ $x_{k+1} = A\, x_k$

looks a lot like the Koopman operator
then we want to find its eigenfunctions

$\Rightarrow$ extended DMD : add nonlinear measurements in $\underline{x}$

but what about overfitting them?

There is a way to prevent overfitting using sparsity.

We expand $\varphi(\underline{x})$ to a library of functions $\Theta(\underline{x}) = [\theta_1(\underline{x}), \theta_2(\underline{x}), \dots, \theta_p(\underline{x})]$

$$\varphi(\underline{x}) \simeq \sum_{k=1}^{p} \theta_k(\underline{x})\, \xi_k = \Theta(\underline{x})\, \underline{\xi}$$ where $\underline{\xi}$ is sparse.

We evaluate on data: $\Theta(X) =$ $\Gamma(X, \dot{X}) =$

$$\begin{bmatrix} \theta_1(\underline{x}_1) & \theta_2(\underline{x}_1) & \dots & \theta_p(\underline{x}_1) \\ \theta_1(\underline{x}_2) & \theta_2(\underline{x}_2) & \dots & \theta_p(\underline{x}_2) \\ \vdots & & \ddots & \vdots \\ \theta_1(\underline{x}_m) & \theta_2(\underline{x}_m) & \dots & \theta_p(\underline{x}_m) \end{bmatrix} \qquad \begin{bmatrix} \nabla\theta_1(\underline{x}_1)\cdot\dot{\underline{x}}_1 & \dots & \nabla\theta_p(\underline{x}_1)\cdot\dot{\underline{x}}_1 \\ \nabla\theta_1(\underline{x}_2)\cdot\dot{\underline{x}}_2 & \dots & \nabla\theta_p(\underline{x}_2)\cdot\dot{\underline{x}}_2 \\ \nabla\theta_1(\underline{x}_m)\cdot\dot{\underline{x}}_m & \dots & \nabla\theta_p(\underline{x}_m)\cdot\dot{\underline{x}}_m \end{bmatrix}$$

Then we solve with a sparse regressor $\boxed{\left(\lambda\Theta(\underline{X}) - \Gamma(\underline{X},\dot{\underline{X}})\right)\cdot\underline{\xi} = \underline{0}}$.

This is hard to do, you need to know λ ahead of time. It happens that eigenfunctions that are lightly damped eigenfunctions (i.e. $Re\lambda \simeq 0$) are the easiest to pull out with this regression.

$\downarrow$

EX: Hamiltonian and other conserved quantities

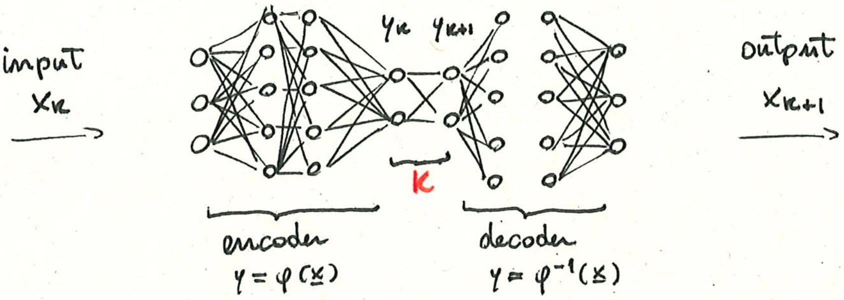
WORKFLOW

System of unknown dynamics $\dot{\underline{x}} = f(\underline{x})$ $\xrightarrow{\text{DATA}}$ PDE for Koopman eigenfunction $\left(\lambda \underbrace{\left[\quad\right]}_{\Theta(x)} - \underbrace{\left[\quad\right]}_{\Gamma(x,x')}\right)\underbrace{\left[\quad\right]}_{\xi} = \underline{0}$ $\longrightarrow$ Eigenfunctions $\varphi(x) \simeq \Theta(x)\xi$, $\lambda \to 0$

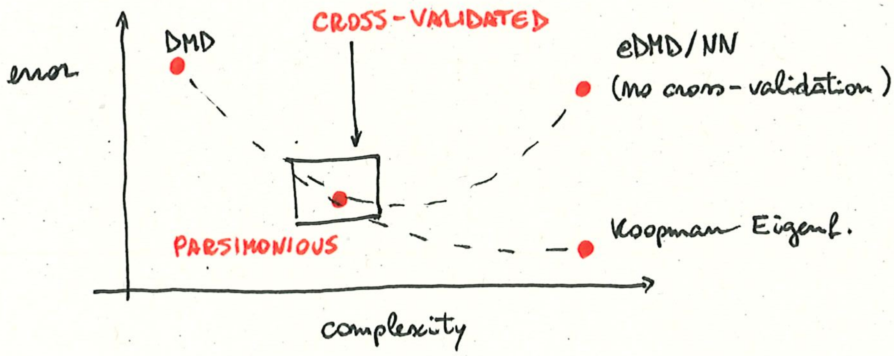
EX if the sparse regression gives $\xi = \left[\frac{2}{3}, -\frac{2}{3}, -\frac{1}{3}\right]^T$ for $\Theta(\underline{x}) = [x^2\ y^2\ x^4]$

$$\Rightarrow \varphi(x) = [x^2\ y^2\ x^4]\begin{bmatrix} 2/3 \\ -2/3 \\ -1/3 \end{bmatrix} \qquad \text{Truth:} \quad H = -\frac{1}{2}x^2 + \frac{1}{2}y^2 + \frac{1}{4}x^4$$

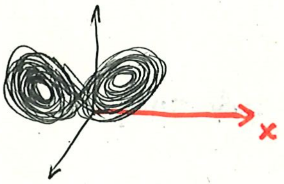
OBS there are also NN to find Koopman eigenfunctions



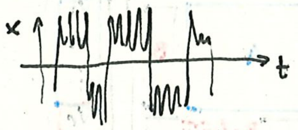
OBS



Hankel Alternative View of Koopman (HAVOK) analysis



take one measure of dyn sys.



$\longrightarrow$

build the time-delay coord. matrix

$$H = \begin{bmatrix} x(t_1) & x(t_2) & \cdots & x(t_p) \\ x(t_2) & x(t_3) & \cdots & x(t_{p+1}) \\ \vdots & \vdots & \ddots & \vdots \\ x(t_q) & x(t_{q+1}) & \cdots & x(t_m) \end{bmatrix}$$

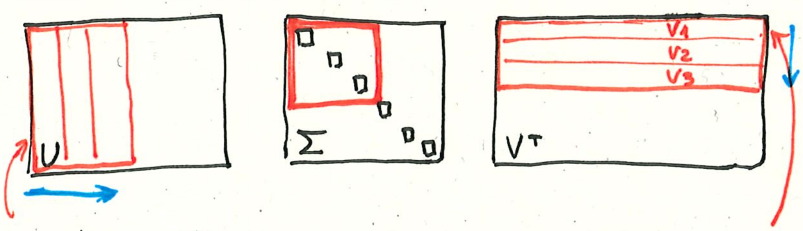
HANKEL MATRIX

The Hankel matrix is built using time delayed copies of a 1d coordinate, stacking upon rows.

$\downarrow$ SVD

Then compute the SVD

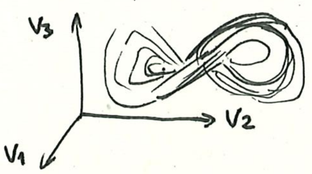
H =



hierarchically sorted to capture most variance

most important eigen-time-series that is most self-similar to the measurem. timeseries x anyway it is shifted

We can plot the timeseries from the columns of V :



" a similar attractor "

This is all pretty classic.

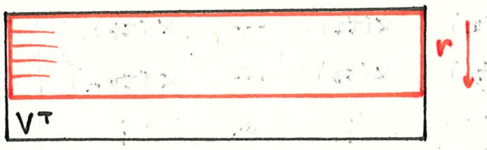
Let us take H and rewrite it in terms of Koopman operator iterations.

$$H = \begin{bmatrix} x(t_1) & Kx(t_1) & & K^{p-1}x(t_1) \\ Kx(t_1) & K^2x(t_1) & & \\ \vdots & & \ddots & \vdots \\ K^{q-1}x(t_1) & \cdots & & K^{m-1}x(t_1) \end{bmatrix}$$

The first columns of U are the most important to express columns in H.

Those first <u>r columns of U</u> is a measurement subspace in U-coordinates. ~~The H reconstructed by those columns are~~ Hitting them with K we stay in the same subspace.

Now we build a regression model on the eigen-time-delayed coordinates.



$$\leftrightarrow \quad \frac{d}{dt}\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_r \end{bmatrix} = \left[\begin{array}{c|c} A & B \\ \hline \text{Bad fit} & \end{array}\right] \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_r \end{bmatrix}$$

apply SINDy

Result: the last value has a bad fitting