Dynamic Mode Decomposition (DMD)

Abstract

Dynamic mode decomposition (DMD) is a dimensionality reduction algorithm developed by Peter Schmid in 2008. Given a time series of data, DMD computes a set of modes each of which is associated with a fixed oscillation frequency and decay/growth rate. For linear systems in particular, these modes and frequencies are analogous to the normal modes of the system, but more generally, they are approximations of the modes and eigenvalues of the composition operator (also called the Koopman operator). Due to the intrinsic temporal behaviors associated with each mode, DMD differs from dimensionality reduction methods such as principal component analysis (PCA), which computes orthogonal modes that lack predetermined temporal behaviors. Because its modes are not orthogonal, DMD-based representations can be less parsimonious than those generated by PCA. However, they can also be more physically meaningful because each mode is associated with a damped (or driven) sinusoidal behavior in time.

I. Lecture

[Book link]: Dynamic Mode Decomposition: Theory and Applications
[Book link]: Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems
[Youtube link]: Dynamic Mode Decomposition (Theory)
[Youtube link]: Dynamic Mode Decomposition (Code; Matlab)

II. Theory

Dynamical process is formulated as follows:

$$\frac{d\vec{\mathbf{x}}}{dt} = f(\vec{\mathbf{x}}, t, \mu),$$

where $\vec{\mathbf{x}}$ defines a measurements, t is a time, μ is a parametrical dependence, and f indicates a system. Since the system f is too complex and/or combined as well as nonlinear, it is not clear the system f what is. In other words, we do not the system f. A lot of data $\vec{\mathbf{x}}$ is measured from the system f although the system f is not clear, the complex dynamical system f can be approximated as follows:

$$\frac{d\vec{\mathbf{x}}}{dt} \approx A\vec{\mathbf{x}}$$

where A defines a linear dynamical system which is a low-rank structure.

When the linear dynamical system A is formulated as **differential equation**:

$$\frac{d\vec{\mathbf{x}}}{dt} = A\vec{\mathbf{x}}, \qquad x \in \mathbb{R}^n, \qquad n \gg 1,$$

the differential equation with the linear dynamical system A can be easily solved, then its general solution is **exponential solution** defined as:

$$\vec{\mathbf{x}} = \vec{\mathbf{v}}e^{\lambda t}$$

where $\vec{\mathbf{v}}$ and λ are eigen vectors and eigen values of the linear system A, respectively. The problem of finding the eigen vectors $\vec{\mathbf{v}}$ and the eigen values λ is a eigen value problem defined as:

$$\lambda\vec{\mathbf{v}} = A\vec{\mathbf{v}}.$$

The eigen values λ and the eigen vectors $\vec{\mathbf{v}}$ are found by solving the equations (called **characteristic function**) below:

$$\det|A - \lambda \mathrm{I}| = \vec{\mathbf{0}},$$

$$(\mathrm{A} - \lambda_j \mathrm{I})\vec{\mathbf{v}}_j = \vec{\mathbf{0}}.$$

TO FIT A GENERAL DMD EQUATION FORM, THE NOTATION OF EIGEN VECTORS (v) IS CHANGED TO EIGEN FUNCTION (ϕ).

An exact solution of the differential equation is represented as:

$$\vec{\mathbf{x}} = \sum_{j=1}^{n} b_j \phi_j e^{\lambda_j t}.$$

Finally, the exact solution of the original dynamic system f is formulated by the above expression, which preserve the time dynamic of t. Now, we know that how can express the exact solution $\vec{\mathbf{x}}$ from the linear dynamical system A. However, we DO NOT know that how can express the linear dynamical system A. In here, we will show that how can the expression is driven. Let we can measure $\mathrm{x}_j = \vec{\mathbf{x}}(t_j)$ at any time point of j. We make big matrix concatenating the data from 1^{st} snapshot to $(m-1)^{th}$ snapshot.

$$\bar{X} = \begin{bmatrix} \\ \mathrm{x}_1 & \mathrm{x}_2 & \cdots & \mathrm{x}_{m-1} \\ \\ \end{bmatrix}.$$

Another matrix shifted by 1 time step is defined as:

$$\bar{X}' = \begin{bmatrix} \\ \mathrm{x}_2 & \mathrm{x}_3 & \cdots & \mathrm{x}_m \\ \\ \end{bmatrix}.$$

It is a ‘data-driven way‘ to get this system.

1. WE ONLY TAKE THE MEASUREMENTS.

2. WE DO NOT THE DYNAMIC SOLVER OF THE SYSTEM.

3. WE RECALL BUILD THE MODEL.

Our objective is to build a linear dynamical system A fitted with $\frac{d\vec{\mathbf{x}}}{dt} = A\vec{\mathbf{x}}$. The linear dynamical system A takes the data $\vec{\mathbf{x}}$ from current state $(j-1)$ to future state (j). Therefore, the linear dynamical system A is satisfied with the relationship below:

$$\bar{X}' = A\bar{X},$$

where $\bar{X}'$ and $\bar{X}$ are the future state of $\bar{X}$ and the current state, respectively. The linear dynamical system A can be extracted using a pseudo inverse $\bar{X}^\dagger$ of $\bar{X}$:

$$A = \bar{X}'\bar{X}^\dagger.$$

We easily think about that the linear dynamical system A perform a least-square fitting from the current state $\bar{X}$ to the future state $\bar{X}'$. This method is called by **EXACT DMD**.

III. Algorithm

A. Singular Value Decomposition (SVD)

Let $\bar{X} \in \mathbb{R}^{n\times(m-1)}$ is dataset of a current state, its SVD is represented as:

$$\bar{X} = U\Sigma V^*.$$

The dimensions of each matrix are defined as:

$$U \in \mathbb{R}^{n\times n},$$

$$\Sigma \in \mathbb{R}^{n\times(m-1)},$$

$$V \in \mathbb{R}^{(m-1)\times(m-1)}.$$

In general, it is difficult to calculate the algorithm because the dimensions of the data $\bar{X}$ are too large. Fortunately, since all systems measuring $\bar{X}$ has a low-rank structure, ‘rank-r truncation‘ is applied to the SVD:

$$\bar{X} = U_r\Sigma_r V_r^*.$$

The truncated dimensions are defined as:

$$U_r \in \mathbb{R}^{n\times r},$$

$$\Sigma_r \in \mathbb{R}^{r\times r},$$

$$V_r \in \mathbb{R}^{(m-1)\times r}.$$

B. *Linear dynamical system A*

A linear dynamical system $A \in \mathbb{R}^{n \times n}$ is defined as:

$$A_{n\times n} = \bar{X}'\bar{X}^{\dagger},$$

where $\bar{X}^{\dagger}$ defines a pseudo-inverse of $\bar{X}$. Since $\bar{X}$ was decomposed by SVD, the pseudo-inverse can be easily calculated as below:

$$\bar{X}^{\dagger} = V_r \Sigma_r^{-1} U_r^*.$$

Then, the linear dynamical system $A_{n\times n}$ can be reformulated by feeding the pseudo-inverse $\bar{X}^{\dagger}$:

$$A_{n\times n} = \bar{X}' V_r \Sigma_r^{-1} U_r^*.$$

Although the linear dynamical system $A_{n\times n}$ was calculated, still the linear dynamical system $A_{n\times n}$ is too huge. To project the linear dynamical system $A_{n\times n}$ into low-rank subspace, the similarity transform is performed:

$$\tilde{A}_{r\times r} = U_r^* A U_r = U_r^*(\bar{X}' V_r \Sigma_r^{-1} U_r^*) U_r = U_r^* \bar{X}' V_r \Sigma_r^{-1},$$

where U_r is low-rank embedding space and $U_r^* U_r = I$. Now, the dimension of the low-rank embedded linear dynamical system $\tilde{A}$ is defined as:

$$\tilde{A} \in \mathbb{R}^{r\times r}, \qquad r \ll n.$$

C. *Eigen value decomposition*

$\tilde{A}$ is the low-rank embedded linear dynamical system. Therefore, eigen value problem of $\tilde{A}$ is cheaply solved:

$$\tilde{A}W = W\Lambda,$$

where $W =$ [eigen vectors] and $\Lambda =$ [eigen values].

D. *Look back up high-dimensional space from low-dimensional space*

In the previous step, the eigen vectors W are calculated in the low-dimensional subspace, but not an original high-dimensional space. The eigen vectors W can be returned to the original space by calculating below:

$$\Phi = \bar{X}' V_r \Sigma_r^{-1} W,$$

where, Φ is **DMD modes** in the original space. The eigen values Λ do not change.

E. *Exact solution* x

We have performed from defining the linear dynamical system A to calculating the eigen vectors Φ and the eigen values Λ. Using the eigen vectors Φ and the eigen values Λ, the solution x can be calculated as:

$$\mathrm{x}(t) = \Phi e^{\Omega t}\mathrm{b} = \sum_{k=1}^{r} \phi_k e^{\omega_k t} b_k,$$

where $\Omega = \log\Lambda$ and b is arbitrary constants. The arbitrary constants b can be decide to solve using initial condition problem:

$$\mathrm{x}(0) = \Phi\mathrm{b},$$

then,

$$\mathrm{b} = \Phi^{\dagger}\mathrm{x}(0),$$

where $\Phi^{\dagger}$ is pseudo-inverse of Φ.