

# AMATH 563 Homework 2: Dynamics and Model Discovery

Gloria(Tiange) Tang

5/6/2020

## Abstract

This study compares Regular Dynamic Mode Decomposition(RDMD), Time-Delay Dynamic Mode Decomposition(TDDMD), Lotka-Volterra(LV) model and Sparse Regression(SR) based on the classic Canadian lynx and snowshoe hare population dataset from 1845 to 1903 and the Belousov-Zhabotinsky chemical oscillator dataset.

## 1 Introduction and Overview

This study aims to compare the Dynamic Mode Decomposition methods and Least Square Solutions methods and find the best fit model for the considered data. It consists of three major parts: the first part *Algorithm Implementation on Lynx&Hare Dataset* uses Regular Dynamic Mode Decomposition(RDMD), Time-Delay Dynamic Mode Decomposition(TDDMD), Lotka-Volterra(LV) model and Sparse Regression(SR) on Canadian lynx and snowshoe dataset; the second part *Algorithm Implementation on BZ Chemical Oscillator Dataset* uses only DMD-based methods on the Belousov-Zhabotinsky chemical oscillator dataset; Finally, the third part *Model Evaluation* computes KL divergence, AIC score and BIC score for each model.

## 2 Theoretical Background

### 2.1 Singular Value Decomposition(SVD)

In linear algebra, SVD states that any matrix  $A$  can be factorized as

$$A_{m \times n} = U_{m \times m} S_{m \times n} V_{n \times n}^T \quad (2.1.1)$$

where  $U$  and  $V$  are orthogonal matrices with orthonormal eigenvectors chosen from  $AA^T$  and  $A^T A$  respectively.  $S$  is a diagonal matrix with  $r$  elements equal to the root of the positive eigenvalues of  $AA^T$  or  $A^T A$ . An important property of the SVD is that it provides an optimal low-rank approximation to a matrix  $A$ . Since the columns of the data used in this study are spatial measurements in

time, then  $U$  encodes spatial patterns, and  $V$  encodes temporal patterns. SVD plays a crucial role in the methods introduced in this study.

## 2.2 Regular Dynamic Mode Decomposition(RDMD)

Given a dataset of  $n$  spatial points and  $k$  snapshots of a given system in time, the DMD method allows us to do future state prediction by approximating the eigendecomposition of the high dimensional matrix  $A$  in

$$X' \approx AX \quad (2.2.1)$$

where  $X$  is of the shape  $n$  rows(spatial points) and  $m$  columns(number of snapshots sampled from the original dataset), and  $X'$  is a matrix of  $X$ 's next unit temporal step. The algorithm can be generally described as follows:

1. Take the SVD of  $X$  with  $r$  rank truncation
2. Compute low-rank approximation  $\tilde{A}$  of  $A$  by using the pseudoinverse of  $X$  obtained via the SVD
3. Compute the eigendecomposition of  $\tilde{A}$
4. Reconstruct the eigendecomposition of  $A$  from the eigenvectors and eigenvalues computed from the eigendecomposition of  $\tilde{A}$ . In particular, the reconstructed  $A$ 's eigenvectors are also called DMD modes. Each mode is associated with a damped (or driven) sinusoidal behavior in time.
5. Reconstruct  $X$  with DMD modes, eigenvalues of  $\tilde{A}$ , and the initial value

## 2.3 Time-Delay Dynamic Mode Decomposition(TDDMD)

According to Koopman Theory, it is possible to represent a nonlinear dynamical system in terms of an infinite-dimensional linear operator acting on a Hilbert space of measurement functions of the state of the system. One way is to obtain intrinsic measurement coordinates based on time-delayed measurements of the system. The algorithm can be generally described as follows:

1. Compute the Hankel Matrix( $H$ ), which stacks time-delayed embeddings by row
2. Take the SVD of  $H$  and determine the rank  $r$  of the system
3. Take  $H$  as the  $X$  in DMD, and follow the algorithm in 2.2

## 2.4 Model Discovery for Dynamical System

Given a model

$$X_t = N(1, X, X^2 \dots X^d \dots \sin(X), \cos(X) \dots) \quad (2.4.1)$$

with potentially many non-linear terms ( $X^d$  denotes a matrix with column vectors given by all possible time-series of  $d$ -th degree polynomials in the state  $x$ ), we are seeking to approximate  $N$  by a generalized linear model. The algorithm can be generally described as follows:

1. Construct a library of candidate nonlinear functions from the data in  $X$
2. Get the vector of coefficients using least-squares solutions
3. Promote sparsity by defining a threshold and shrinking the coefficients below the threshold to 0
4. Regress dynamics onto the remaining terms to find the sparse vector of coefficients

The first two steps are a general method of Least Square solutions, while considering all steps is called Sparse Regression to model discovery.

### 3 Algorithm Implementation on Lynx&Hare

Consider population dataset of Canadian Lynx and Snowshoe Hare from 1845 to 1903, yielding in an input matrix of size 2 by 30 (2 species and 30-year snapshots). The data is interpolated to 580 snapshots using B spline interpolation.

#### 3.1 Dynamic Mode Decomposition Based Methods

##### 3.1.1 Regular Dynamic Mode Decomposition(RDMD)

With the computing outcome of RDMD algorithm following the steps in 2.2, we reconstruct the interpolated data, and the result can be visualized as follows:

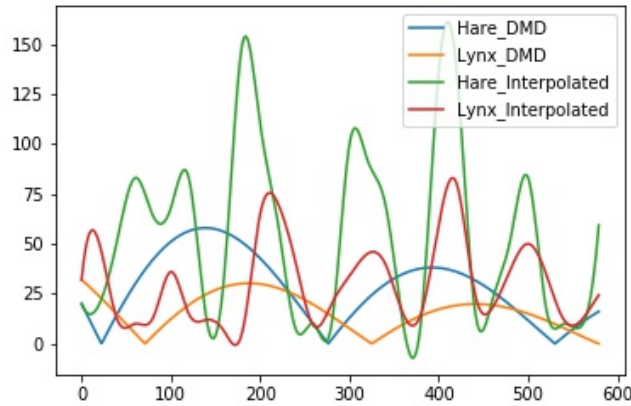


Figure 1: Comparison between Real and Reconstructed Data of RDMD

### 3.1.2 Time-Delay Dynamic Mode Decomposition(TDDMD)

The SVD of the Hankel matrix indicates that this is a rank-2 system.

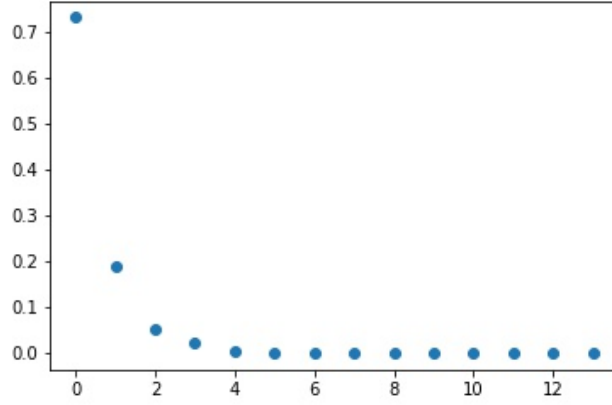


Figure 2: The Eigenvalues of H on Hare&Lynx Dataset

With the computing outcome of TDDMD algorithm following the steps in 2.3, we reconstruct the interpolated data, and the result can be visualized as follows:

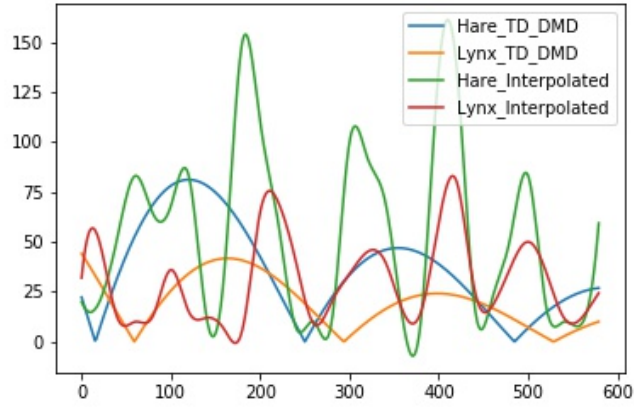


Figure 3: Comparison between Real and Reconstructed Data of TDDMD

## 3.2 $Ax = b$ Least Square Based Methods

### 3.2.1 Lotka-Volterra(LV)

The Lotka–Volterra equations, also known as the predator–prey equations, are frequently used to describe the dynamics of biological systems in which two species interact, one as a predator and the other as prey. The populations change through time according to the pair of equations:

$$\begin{aligned}\frac{dx}{dt} &= (b - py)x \\ \frac{dy}{dt} &= (rx - d)y\end{aligned}\tag{3.2.1}$$

where  $x$  is the population of hare and  $y$  is the population of lynx. We are aiming to use least square solutions to find  $b, p, r, d$  by constructing the model library  $\{x, y, xy\}$  and following the first and second steps of the algorithm described in 2.4.

The result is  $b = 0.45$ ,  $p = 0.10$ ,  $d = 0.44$ ,  $r = 0.05$ .

### 3.2.2 Sparse Regression(SR)

Following the algorithm described in 2.4, a library of terms which consists of up to third-degree polynomials and sines and cosines is constructed and the threshold is set as 0.5. The discovered model is

$$\frac{dx}{dt} = 1.29x - 1.02y + 2.16\sin(x) + 4.46\sin(y) + 0.73\cos(y)\tag{1}$$

$$\frac{dy}{dt} = 1.52\sin(x) - 0.78\cos(x) - 0.96\sin(y)\tag{2}$$

## 4 Algorithm Implementation on BZ

Consider BZ Chemical Oscillator Dataset which has 1200 snapshots of size  $451 * 351$  pixels. We convert the 3D input into 2D by reshaping the data to  $158301 * 1200$ .

### 4.1 Dynamic Mode Decomposition Based Methods

#### 4.1.1 Regular Dynamic Mode Decomposition(RDMD)

With the computing outcome of RDMD algorithm following the steps in 2.2, we reconstruct the 2D matrix, and this randomly selected snapshot shows how well the reconstructed data aligns with the original:

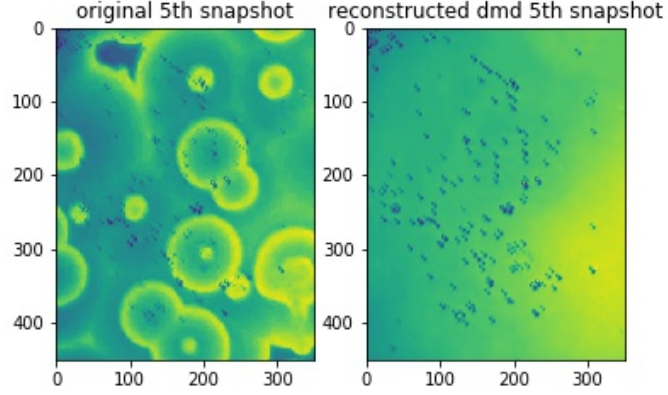


Figure 4: The 5<sup>th</sup> Snapshot Comparison

#### 4.1.2 Time-Delay Dynamic Mode Decomposition(TDDMD)

With the computing outcome of TDDMD algorithm following the steps in 2.3, it is found that the system is of rank 1 with an eigenvalue with around 80% Ddominance.

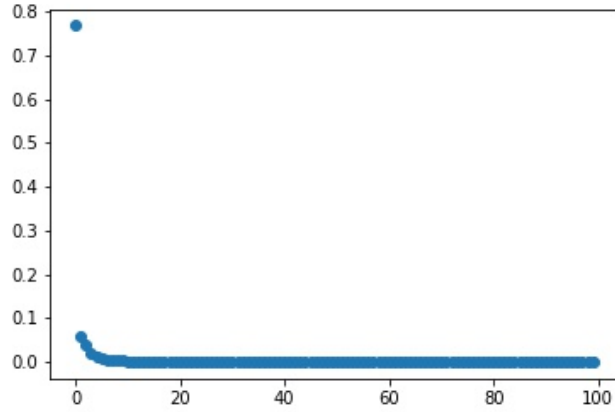


Figure 5: The Eigenvalues of H on BZ Dataset

## 5 Model Evaluation

For the Lynx&Hare Dataset, KL Divergence, AIC and BIC scores are used to evaluate the models separately for hare and lynx. Based on the KL divergence, the three best fit models are retained to compute its AIC and BIC scores respectively. For BZ Chemical Oscillator Dataset, only AICs and BICs are computed. KL Divergence is given as

$$D_{KL} = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \quad (4.1)$$

where  $P(x)$  is the true data density distribution and  $Q(x)$  is the reconstructed data density distribution. All inputs are normalized, and a very small number is added to the reconstructed data density distribution to avoid the outcome reaching infinity for computing KL Divergence. AIC is given as

$$AIC = 2K + n \ln\left(\frac{RSS}{n}\right) \quad (4.2)$$

BIC is given as

$$BIC = K \ln(n) + n \ln\left(\frac{RSS}{n}\right) \quad (4.3)$$

where  $RSS$  stands for Residual Sum of Squares between true data and reconstructed data,  $K$  is the number of terms and  $n$  is the number of data points used in the model.

Criteria	RDMD		TDMD		LV		SR	
	Hare	Lynx	Hare	Lynx	Hare	Lynx	Hare	Lynx
KL	0.69	0.67	0.71	0.65	0.29	0.26	0.08	0.09
AIC	4472.79	3727.18			4939.51	4218.36	3724.47	4174.87
BIC	4481.52	3735.90			4948.23	4227.09	3746.28	4187.96

Table 1: Hare&Lynx Implementation Result of KL Divergence, AIC and BIC

Method	AIC	BIC
RDMD	109510.15	109630.09
TDDMD	-650027.29	-650017.31

Table 2: BZ Implementation Result of KL Divergence, AIC and BIC

## 6 Summary and Conclusions

For Hare and Lynx dataset, as shown in Figure 1 and Figure 3, TDDMD captures the magnitude of the original data better than RDMD, which is also reflected in the KL divergence of Table 1, where DMD models weakly capture

the dynamics than the least square methods. However, if we look at AIC and BIC scores, RDMD performs better than LV where we implement least square fit on the data. An intuitive understanding might be that when calculating KL divergence, we only consider the frequency of values at bins for the density distribution hence neglect the power of how model is performing in terms of fitting the data one by one, which is reflected in AIC and BIC with the calculation of RSS. Overall, SR has the best performance since we are solving an under-determined system and it is believed that only a few of the library terms are associated with the model.

For the BZ dataset, we can tell from TDDMD significantly outperforms RDMD as we see a weak representation of original random snapshot in Figure 4 and negative AIC and BIC scores of TDDMD in Table 2. Further investigation needs to be conducted to conclude that TDDMD generally works better than RDMD in over-determined system and the other way around in under-determined system.

## A Main Python Functions Used in this Study

1.scipy.interpolate.splrep

Find the B-spline representation of 1-D curve.

2.numpy.linalg.svd()

Find singular value decomposition of a matrix.

3.numpy.linalg.lstsq()

Return the least-squares solution to a linear matrix equation.

4.sklearn.preprocessing.normalize()

Scale input vectors individually to unit norm (vector length) with default L2 norm.

5.numpy.histogram(density=TRUE)

The value of the probability density function at the bin, normalized such that the integral over the range is 1.

6.scipy.stats.entropy()

Calculate the KL divergence.

7.matplotlib.pyplot.imshow()

Display data as an image.

## B Original Jupyter Notebook

Please see <https://github.com/gloriatang0325/AMATH-563>