Dynamic mode decomposition: an alternative algorithm for full rank datasets

G.H. Nedzhibov

Abstract

Dynamic mode decomposition (DMD) is a modal decomposition technique that describes high-dimensional dynamic data using coupled spatial-temporal modes. It combines the main features of performing principal components analysis (PCA) in space and power spectral analysis in time. The method is an equation-free in the sense that it does not require knowledge of the underlying governing equations and it is entirely data-driven. The purpose of this paper is to introduce a new algorithm for computing the dynamic mode decomposition in the case of full rank data. The new approach is more economic from a computational point of view, which is an advantage when working with large data.

MSC: 65P02, 37M02, 37L65

keywords: Dynamic mode decomposition, Koopman operator, Singular value decomposition, Equation-free.

1 Introduction

Dynamic mode decomposition algorithm (*DMD method*) has been established as a leading technique for identifying spatiotemporal coherent structures from high-dimensional data. It can be considered to be a numerical approximation to Koopman spectral analysis, and in this sense it is applicable to nonlinear dynamical systems (see [2, 3]). Last years the popularity of DMD method has grown and it has been applied for a variety of dynamical systems in many different fields such as video processing [12], epidemiology [13], neuroscience [15], financial trading [16, 17, 18], robotics [14], cavity flows [4, 6] and various jets [2, 5]. For a review of the DMD literature, we refer the reader to [8, 9, 19].

DMD Method

We briefly consider the DMD algorithm, which was introduced for the first time by Schmid [1] in the fluid mechanics community.

The standard definition of DMD consider a sequential set of data $Z = \{z_0, \ldots, z_m\}$, where each $z_k \in R^n$. The data z_i could be from measurements, experiments or simulations collected at time t_i from a given nonlinear system, assume that the data are equispaced in time, with a time step $\triangle t$ and the collection time starts from t_0 to t_m.

The method uses arrangement of the data set into two large data matrices:

$$X = [z_0, \ldots, z_{m-1}] \quad \text{and} \quad Y = [z_1, \ldots, z_m]. \tag{1}$$

The main assumption of the method is that there exists a linear (unknown) operator A relating z_k to the subsequent z_{k+1}:

$$z_{k+1} = Az_k. \tag{2}$$

Expression (2) is equivalent to

$$Y = AX. \tag{3}$$

Then the dynamic mode decomposition of the data matrix Z is given by the eigendecomposition of A. The *DMD modes* and *eigenvalues* are intended to approximate the eigenvectors and eigenvalues of A.

One possible approach for approximating the operator A is to use the singular value decomposition (SVD) of the data matrix $X = U\Sigma V^*$ and the expression

$$A \approx YX^\dagger = YV\Sigma^{-1}U^*\,, \tag{4}$$

where $X^\dagger$ is the pseudoinverse of X.

It should be noted that if n is large, i.e. $n \gg m$, calculating the eigendecomposition of the $n \times n$ matrix A can be prohibitively expensive.

The following algorithm allows to calculate DMD modes and eigenvalues without directly calculating A.

Algorithm 1: Exact DMD Algorithm [8]

1. Compute the SVD of $X = U\Sigma V^*$ and substitute into (4) :
   $A = YV\Sigma^{-1}U^*$.
2. Define $\tilde{A} = U^*AU = U^*YV\Sigma^{-1}$.
3. Compute the eigendecomposition of $\tilde{A}$
   $\tilde{A}W = W\Lambda$,
   where W is the eigenvector matrix and Λ is the diagonal matrix of eigenvalues $\Lambda = diag\{\lambda_i\}$. Each λ_i is a DMD eigenvalue.
4. Compute the DMD modes
   $\Phi = YV\Sigma^{-1}W$.
   Each column ϕ_i of Φ is a DMD mode corresponding to eigenvalue λ_i.

Finally, we can reconstruct the approximate dynamics of data set Z as

$$z_{DMD}(k) = \Phi \Lambda^k b \,, \tag{5}$$

where $b = \Phi^{\dagger} z_0$, and $\Phi^{\dagger}$ is the pseudo-inverse of Φ.

An alternative DMD algorithm

In our recent publication [20] we introduced a new algorithm for computing DMD decomposition of A. The new algorithm follows the same steps as in Algorithm 1. The distinctive feature is the following. We use eigendecomposition of the following matrix

$$\hat{A} = \Sigma^{-1} U^* Y V \,, \tag{6}$$

instead eigendecomposition of matrix $\tilde{A}$, in Step 2 of Algorithm 1.

The two matrices $\tilde{A}$ and $\hat{A}$ are similar, with transformation matrix Σ, therefore have the same eigenvalues. For the DMD modes we use the formula

$$\Phi = YV\hat{W}\,, \tag{7}$$

where $\hat{W}$ is the eigenvector matrix of $\hat{A}$, i.e.

$$\hat{A}\hat{W} = \hat{W}\Lambda\,. \tag{8}$$

It is seen that although the matrices $\tilde{A}$ and $\hat{A}$ have a similar representation, the expression (7) for calculating the DMD modes is simpler than the corresponding formula in Algorithm 1, Step 4. Therefore the algorithm introduced in [20] is more efficient in terms of computational cost.

2 A new DMD algorithm for full rank datasets

As mentioned in the previous section, the DMD method involves approximating the eigendecomposition of best fit linear operator A, which refers to

$$Y = AX\,.$$

In fact, the algorithms of the DMD method allow the calculation of DMD modes and eigenvalues without direct calculation of A.

For this purpose a reduced-order approximation of the matrix A is used, such as

$$\tilde{A} = U^* AU = U^* YV\Sigma^{-1} \tag{9}$$

defined in Algorithm 1, or

$$\hat{A} = \Sigma^{-1} U^* AU, \tag{10}$$

defined by (6).

In case the matrix X has full rank, i.e. $rankX = m$, then both $\tilde{A}$ and $\hat{A}$ will be $m \times m$ matrices. Further, we will assume that matrix X is a full rank matrix.

Our goal is to introduce a more computationally efficient algorithm for calculating DMD modes and eigenvalues in this particular case.

Let us consider the following matrix

$$\bar{A} = V\hat{A}V^*\,, \tag{11}$$

where V is the $m \times m$ unitary matrix from the SVD of $X = U\Sigma V^*$. Obviously, matrices $\hat{A}$ and $\bar{A}$ are unitarily similar, whith V^* the similarity transformation matrix.

From (6) and (11), we get

$$\bar{A} = V\Sigma^{-1}U^{*}Y\,, \tag{12}$$

which yields

$$\bar{A} = X^{\dagger}Y\,, \tag{13}$$

where $X^{\dagger}$ is the Moore-Penrose pseudoinverse of X.

Denoting the eigen-decomposition of $\bar{A}$ by

$$\bar{A}\bar{W} = \bar{W}\Lambda\,, \tag{14}$$

where columns of $\bar{W}$ are eigenvectors and Λ is a diagonal matrix containing the corresponding eigenvalues.

Relations (10), (11) and (14) yield

$$A(X\bar{W}) = (X\bar{W})\Lambda, \tag{15}$$

which means that

$$\Phi = X\bar{W} \tag{16}$$

is the matrix of DMD modes.

Theorem 1 *Let (λ, w), with $\lambda \neq 0$, be an eigenpair of $\bar{A}$ defined by (13). Then the corresponding eigenpair of A is (λ, φ), where*

$$\varphi = Yw.$$

Next, we resume the results from above in a form of an algorithm.

Algorithm 2: DMD Algorithm for full rank dataset

1. Define $\bar{A} = X^{\dagger}Y$, where $X^{\dagger}$ is the pseudoinverse of X.
2. Compute the eigendecomposition of $\bar{A}$
   $\bar{A}\bar{W} = \bar{W}\Lambda$,
   where $\bar{W}$ is the eigenvector matrix and Λ is the diagonal matrix of eigenvalues $\Lambda = diag\{\lambda_i\}$. Each λ_i is a DMD eigenvalue.
3. Compute the DMD modes
   $\Phi = Y\bar{W}$.
   Each column ϕ_i of Φ is a DMD mode corresponding to eigenvalue λ_i.

---

3 Application of the new algorithm

Originally DMD theory focuses on full-rank, sequential time series, where typically $n \gg m$. Since the method was introduced it found applications in many different fields. This presume, in practice, that in some cases the inverse relationship is also valid, namely the number of measurements m taken in time may be greater than dimension of spatial measurements per time n, i.e. $m < n$.

The following options are available.

- When the high-dimensional dynamics of the data has some underlying low-dimensional structure, it may be possible to capture the key dynamics of the data with relatively few DMD modes. In this case, the rank of the data-set is equal to the number of DMD modes.
- In some data sets, there are linear dependencies among the measurements (snapshots), i.e. the rank of data-set is too low and the DMD fails to fully capture the dynamics of the system. The solution of this rank mismatch is by rearranging the data set in modified (augmented) data matrices inspired by the Hankel matrix constructed in the eigenvalue realization algorithm (ERA), see [8].

Illustrative Examples

The new algorithm introduced in Section 2 has the advantage of being more cost-effective than standard DMD algorithms when used on a full-range data set. With the following examples, we will illustrate algorithm introduced in Section 2.

Example 1. *Standing waves*

It is known that the standard DMD algorithm is not able to represent a standing wave in the data [8]. For example, if only measurements of a single sine or cosine wave are collected, DMD fails to capture periodic oscillations in the data.

To demonstrate this we perform DMD method on a single measurement

$$x(t) = cos(t).$$

In this case, the X matrix only contains a single row

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_{n-1} \end{bmatrix}$$

and the DMD algorithm only returns a single eigenvalue, which is unable to capture the oscillation in the data, see Figure 1. In fact, it is necessary to have two complex conjugate eigenvalues corresponding to the *sine* and *cosine* pair in order for DMD to capture standing wave.

To solve this issue we construct augmented data matrices (shift-stacked data matrices):

$$X_{aug} = \begin{bmatrix} x_1 & x_2 & \dots & x_{n-2} \\ x_2 & x_3 & \dots & x_{n-1} \end{bmatrix} \text{ and } Y_{aug} = \begin{bmatrix} x_2 & x_3 & \dots & x_{n-1} \\ x_3 & x_4 & \dots & x_n \end{bmatrix}. \quad (17)$$

Since the X_{aug} matrix contains two rows that are linearly independent, i.e. it is a full-rank matrix, then it has two (conjugate pair complex) DMD eigenvalues. We performed both mentioned algorithms to represent $x(t)$, see Figure 1.

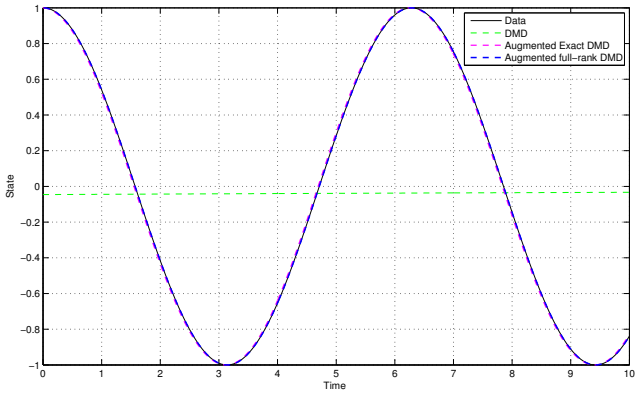

Figure 1: Example of DMD, Augmented exact DMD and Augmented full-rank DMD on a standing wave example $x(t) = cos(t)$.

We can say that new algorithm (*Algorithm 2*) produces the same result as the exact DMD procedure (*Algorithm 1*).

Example 2. *Example case from finance*

We can demonstrate the rank deficiency problem with an example from finance. We can use DMD method to discover the evolutionary patterns in commodities market. In particular, if we consider the evolution in the price of only one type of commodity, we will get the rank related issue. In fact, this problem is quite similar to the standing wave problem.

Let us consider price evolution of the Brent Crude Oil for the period 01.02.2022 – 28.02.2022, containing 21 trading days, see Fig 2. Similarly to Example 1, the data matrix X contains a single row

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_{21} \end{bmatrix},$$

where each x_i is the closing price on the respective day. In order to overcome the rank mismatch issue we construct augmented data matrices

$$X_{aug} = \begin{bmatrix} x_1 & x_2 & \dots & \dots \\ x_2 & x_3 & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots \\ x_s & x_{s+1} & \dots & x_{20} \end{bmatrix}; \; Y_{aug} = \begin{bmatrix} x_2 & x_3 & \dots & \dots \\ x_3 & x_4 & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots \\ x_{s+1} & x_{s+2} & \dots & x_{21} \end{bmatrix}, \quad (18)$$

where we can choose s such that $11 \leq s \leq 19$, which ensures that the matrices X_{aug} and Y_{aug} will be such that the number of rows will be greater than the number of columns. For each s matrix X_{aug} is of full rank.

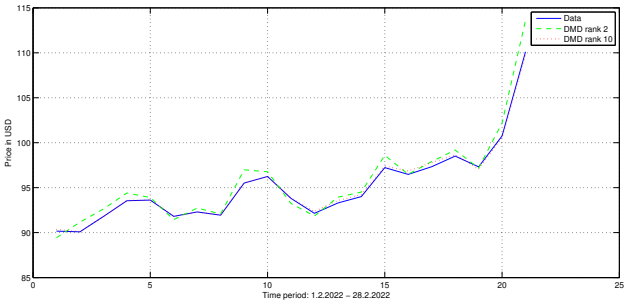

Figure 2: Brent Crude Oil price for the period 1.2.2022 - 28.2.2022.

We performed the DMD method by using Algorithm 2 on augmented data matrices X_{aug} and Y_{aug} for each s. The results show that the best approximation of the real data is obtained at the maximum rank of X_{aug}, which is $rank(X_{aug}) = 10$ obtained for $s = 11$. Figure 2 shows the two approximations for $rank(X_{aug}) = 2$ and $rank(X_{aug}) = 10$, respectively.

4 Conclusion

The purpose of this study was to present a new approach for computing approximate DMD modes and eigenvalues. As a result, we have introduced a new algorithm, alternative procedure for executing the DMD decomposition in the case of full rank data set. We demonstrate the performance of the presented algorithms with numerical examples. From the obtained results we can conclude that the introduced approach gives identical results with those of the *exact DMD method*.

References

[1] P. J. Schmid and J. Sesterhenn. Dynamic mode decomposition of numerical and experimental data. In 61st Annual Meeting of the APS Division of Fluid Dynamics. American Physical Society, November 2008.

[2] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. J. Fluid Mech., 641:115-127, December 2009.

[3] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. Nonlin. Dynam., 41(1-3):309-325, August 2005.

[4] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. J. Fluid Mech., 656:5-28, August 2010.

[5] P. J. Schmid. Application of the dynamic mode decomposition to experimental data. Exp. Fluids, 50(4):1123-1130, April 2011.

[6] A. Seena and H. J. Sung. Dynamic mode decomposition of turbulent cavity ows for selfsustained oscillations. Int. J. Heat Fluid Fl., 32(6):1098-1110, December 2011.

[7] I. Mezić. Analysis of fluid flows via spectral properties of the Koopman operator. Annu. Rev. Fluid Mech., 45:357-378, 2013.

[8] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, J. Nathan Kutz. On dynamic mode decomposition: Theory and applications. Journal of Computational Dynamics, 2014, 1 (2) : 391-421. doi: 10.3934/jcd.2014.1.391

[9] J. Nathan Kutz, Steven L. Brunton, Bingni W. Brunton, and Joshua Proctor (2016). Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems, SIAM 2016, ISBN 978-1-611-97449-2, pp. 1-234.

[10] K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses. J. Nonlinear Sci., 22:887-915, 2012.

[11] S. Bagheri, Koopman-mode decomposition of the cylinder wake, J. Fluid Mech., vol. 726, pp. 596-623, 2013.

[12] J. Grosek, J. Nathan Kutz, Dynamic Mode Decomposition for Real-Time Background/Foreground Separation in Video, arXiv: 1404.7592.

[13] J. L. Proctor and P. A. Eckhoff, Discovering dynamic patterns from infectious disease data using dynamic mode decomposition, International health, 7 (2015), pp. 139-145.

[14] E. Berger, M. Sastuba, D. Vogt, B. Jung, and H. B. Amor, Estimation of perturbations in robotic behavior using dynamic mode decomposition, Journal of Advanced Robotics, 29 (2015), pp. 331-343.

[15] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz, Extracting spatialtemporal coherent patterns in large-scale neural recordings using dynamic mode decomposition, Journal of Neuroscience Methods, 258 (2016), pp. 1-15.

[16] J. Mann and J. N. Kutz, Dynamic mode decomposition for financial trading strategies, Quantitative Finance, (2016), pp. 1-13.

[17] Ling-xiao Cui, Wen Long, 2016. "Trading strategy based on dynamic mode decomposition: Tested in Chinese stock market," Physica A: Statistical Mechanics and its Applications, Elsevier, vol. 461(C), pages 498-508.

[18] D. P. Kuttichira, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using dynamic mode decomposition," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017, pp. 55-60, doi: 10.1109/ICACCI.2017.8125816.

[19] Zhe Bai, Eurika Kaiser, Joshua L. Proctor, J. Nathan Kutz, Steven L. Brunton, Dynamic Mode Decomposition for CompressiveSystem Identification, AIAA Journal, Vol. 58, No. 2, 2020, pp. 561-574.

[20] G. Nedzhibov, Dynamic Mode Decomposition: a new approach for computing the DMD modes and eigenvalues, Ann. Acad. Rom. Sci. Ser. Math. Appl. Vol. 14, No. 1-2, 2022, pp. 5-16.