Learning a Reduced Order Dynamic Mode Decomposition by Random Observable Features

Sudam Surasinghe\textsuperscript{1}, Erik Bollt\textsuperscript{2}.

\textsuperscript{1} Department of Mathematics, \textsuperscript{2} Department of Electrical and Computer Engineering, Clarkson Center for Complex Systems Science (C\textsuperscript{3}S\textsuperscript{2}), Clarkson University, Potsdam, NY

Rework DMD through Johnson-Lindenstrauss theorem and Random Projection

- SVD based algorithms
  - need more memory
  - Computationally expensive
  - Optimal Projection

- Random Projection
  - Just matrix multiplication.
  - High quality

Given...

\[ \|x'_i - x'_j\| \simeq (1 \pm \epsilon)\|x_i - x_j\| \]

with \( L \ll N \)?

\[ L \geq \frac{C \ln M}{\epsilon^2} \]

Objective & Overview

- DMD separate the variables (space & time) and isolate dynamic structures by data.
- We will use Random projection for efficient calculations. \[
X(t) \approx \sum_{i=1}^{r} b_i \Phi_i(z)e^{\omega_i t}
\]

Example:

Analysis of data from Strait of Gibraltar by standard DMD.
Collaborate with: Erik Bollt, Marko Budišić, Kanaththa Priyankara, Sathsara Dias, Larry Pratt, Jose Sanchez-Garrido.
Background

**Dynamic Mode Decomposition (DMD)**
[Schmid, 2008]

Numerical procedure to exact dynamical features from flow based on Krylov sequence.

**DMD & Koopman operator**
[Rowley & Mezić, 2009]

Connect the DMD with Koopman operator.

**Extended DMD (EDMD)**
[Williams, 2015]

Better Approximation for Koopman operator.

**Kernel DMD**
[Williams, 2015]

Use kernel trick to reduce calculation in EDMD.

**Application of Koopman Operator**
[Mezić, 2002]

Dynamics of Physical systems that they model based on the spectral properties of the Koopman operator.

**Koopman Operator**
[Koopman, 1931]

Infinite dimensional linear operator which describe the evolution of observable functions.

**Randomized Dynamic Mode Decomposition (rDMD)**

- Random projection will projected data into reasonable space
- Simple and accurate calculation
- Low computational cost
- Reduce the storage cost

**We are looking at DMD in lens of Johnson-Lindenstrauss theorem and random projection.**
Koopman operator is a tool to analyze global dynamics of a dynamical system.

Koopman operator is defined by,

\[ \mathcal{K} : \mathcal{F} \to \mathcal{F}, \text{ which acts on function space } \mathcal{F} = \{ \text{set of functions } \psi : \mathcal{M} \to \mathbb{R} \} \]

\[ \psi'(x) = \mathcal{K}[\psi](x) = \int \delta(x' - f(x))\psi(x')dx' = \psi \circ f(x) \]

- Meaning measure \( f \) but downstream by \( \psi \).
- Adjoint operator of Frobenius-Perron operator.

Koopman operator is a

- Linear,
- infinite dimensional operator.

[Koopman, 1931]
Snapshot Matrix and Estimating Koopman operator

- $S_i$ contains measures of all states at step $i$.

$$X = \begin{bmatrix}
S_1 & \cdots & S_M \\
\vdots & \ddots & \vdots \\
S_M & \cdots & S_{M+1}
\end{bmatrix} \quad Y = \begin{bmatrix}
\vdots & \ddots & \vdots \\
S_2 & \cdots & S_{M+1}
\end{bmatrix}$$

- Koopman operator acts as a time shift on columns $X \xrightarrow{\mathcal{K}} Y$

$$K_D = \arg \min ||KX - Y||_F$$

$$K_D = YX^+$$

[Rowley & Mezić 2009]
Commonly Used DMD Algorithm

\[
X = \begin{bmatrix} \vdots & \cdots & \vdots \\ S_1 & \cdots & S_M \\ \vdots & \vdots & \vdots \end{bmatrix}_{N \times M} \quad Y = \begin{bmatrix} \vdots & \cdots & \vdots \\ S_2 & \cdots & S_{M+1} \\ \vdots & \vdots & \vdots \end{bmatrix}_{N \times M}
\]

\[
K_D (N \times N) = YX^\dagger
\]

Computing spectrum of \( K_D (N \times N) \) is expensive

Therefore, it estimated by

\[
K_D (L \times L) = U^*Y(U^*X)^\dagger
\]

where \( X = U_L \Sigma_L V_L^* \).

\[ L \ll N \]

- There are a few variations of the basic algorithm.
- DMD spectrum is a numerical estimator for Koopman spectrum.

Exact DMD

\[
\begin{align*}
\text{Input: } & X, Y \\
1 & [U_r, \Sigma_r, V_r] = \text{SVD}(X); \quad \text{// Truncated SVD} \\
2 & \hat{K}_D = U_r^*YV_r\Sigma_r^{-1}; \quad \text{// Compression } K \downarrow K_D \\
3 & [\Psi, \Lambda] = \text{eig}(\hat{K}_D); \quad \text{// diag(\Lambda) equals to eigenvalues of } K \\
4 & \Phi = YV_r\Sigma_r^{-1}\Psi; \quad \text{// } \Phi \text{ estimates eigenvectors of } K \\
5 & b = \Phi^\dagger X[:,1]; \quad \text{// Estimates the coefficients} \\
\text{Output: } & \Phi, \quad \Lambda, \quad b
\end{align*}
\]

\[
X(t) \approx \sum_{i=1}^{r} b_i \Phi_i(z) e^{\omega_i t} \quad \omega_i = \ln(\lambda_i)/\Delta t
\]

Why DMD?

- Isolate specific dynamic structures
- Equation free modeling
- Reduce the dimension of the data
- Can identify physically meaningful decomposition

[Tu, et. al, 2014]
DMD in Projected Space

\[
\mathbf{K}_D = \arg \min \| \mathbf{KX} - \mathbf{Y} \|_F \quad \Rightarrow \quad \text{DMD: } \mathbf{K}_D^{(N \times N)} = \mathbf{YX}^\dagger
\]

DMD in projected space with rank L projector \( P : \mathbb{R}^N \rightarrow \mathbb{R}^L \)

\[
\hat{\mathbf{K}}_D = \arg \min \| \mathbf{KPX} - \mathbf{PY} \|_F
\]

\[
\hat{\mathbf{K}}_D^{(L \times L)} = P_{L \times N} \mathbf{Y} (P_{L \times N} \mathbf{X})^\dagger
\]

\[
\hat{\mathbf{K}}_D = P\mathbf{K}_D P^\dagger
\]

Therefore \( \hat{\mathbf{K}}_D \) and \( \mathbf{K}_D \) have common eigenvalues.

If \( \hat{\phi}_L \) is eigenvector of \( \hat{\mathbf{K}}_D \), then \( \hat{\phi} = P^\dagger \hat{\phi}_L \) is a eigenvector for \( \mathbf{K}_D \).

\( \hat{\phi} = \mathbf{Y}(\mathbf{PX})^\dagger \hat{\phi}_L \) is also a eigenvector of \( \mathbf{K}_D \).

DMD with SVD projection

Projection matrix \( P = \mathbf{U}_L^* \) where \( \mathbf{X} = \mathbf{U}_L \Sigma_L \mathbf{V}_L^* \)

DMD random projection (rDMD)

Projection matrix \( P = \mathbf{R}_{L \times N} = (\frac{1}{\sqrt{L}} \mathbf{R}_{i,j}) \)

where elements \( \mathbf{R}_{i,j} \) distributed \( \mathbf{R}_{i,j} \overset{iid}{\sim} \mathcal{N}(0, 1) \)
The random projection method is based on the Johnson-Lindenstrauss lemma.

**Theorem [Johnson-Lindenstrauss Lemma]:**

For any $0 < \epsilon < 1$ and any integer $M > 1$ let $L$ be a positive integer such that $L \geq L_0$ with $L_0 = \frac{C \ln M}{\epsilon^2}$, where $C$ is a suitable constant ($C \approx 8$ in practice, $C = 2$ is good enough). Then for any set $X$ of $M$ data points in $\mathbb{R}^N$, there exists a map $f : \mathbb{R}^N \rightarrow \mathbb{R}^L$ such that for all $x_1, x_2 \in X$,

$$
(1 - \epsilon)\|x_1 - x_2\|^2 \leq \|f(x_1) - f(x_2)\|^2 \leq (1 + \epsilon)\|x_1 - x_2\|^2.
$$

[Johnson et. al, 1984]

**Theorem [Random Projection]**

For any $0 < \epsilon, \delta < \frac{1}{2}$ and positive integer $N$,

there exits a random matrix of $B$ of size $L \times N$ such that for $L \geq L_0$ with $L_0 = \frac{C \ln(1/\delta)}{\epsilon^2}$ and for any unit-length vector $x \in \mathbb{R}^N$, $Pr\{||Bx||^2 - 1| > \epsilon\} \leq \delta$

or $Pr\{||Bx||^2 - 1| > \epsilon\} \leq e^{-CL\epsilon^2}$

[Papadimitriou et. al., 1998]
**Randomized DMD**

**DMD in projected space with rank L random projection R**

Error bound of estimating $\tilde{z}'$ by using the rDMD

$$\|z' - \tilde{z}'\| \leq \frac{\|Rz' - \hat{\mathbf{K}}_D Rz\|}{1 - \epsilon}$$

with at least probability of $O(1/M^2)$ for any $0 < \epsilon < 1$ with $L > \frac{C \log(M)}{\epsilon^2}$.

**Why rDMD?**

- Simple calculation with high accuracy
- Reduce the computational cost
- Reduce the storage cost
- SVD based existing algorithms need to store high resolution data matrix and may lead to memory issues
- Our proposed algorithm can reduce the dimension of data just using matrix multiplication.

**Algorithm 1: Randomized DMD (rDMD)**

**Data:** $X, Y \in \mathbb{R}^{N \times M}$

**Input:** $\epsilon$

$L_0 = \frac{\epsilon \ln M}{\epsilon^2}$;

Choose $L$ such that $L \geq L_0$;

Construct a random matrix $R = \frac{1}{\sqrt{L}} (r_{ij}) \in \mathbb{R}^{L \times N}$ such that $r_{ij} \sim N(0,1)$;

Calculate $X_L := RX$, $Y_L := RY$;

Calculate $\hat{\mathbf{K}} = Y_L X_L^\dagger$;

$[\Lambda, \Phi_L] = \text{eigs}(\hat{\mathbf{K}})$;

**Result:** $\text{diag}(\Lambda), YX_L^\dagger \Phi_L$
Our proposed algorithm can reduce the dimension of data just using matrix multiplication.
Our proposed algorithm can reduce the dimension of data just using matrix multiplication.
Application: Oceanographic Data (Gulf of Mexico)

Standard DMD

rDMD

rDMD(ran Proj): Background Mode

N=208285, M=104, L=90

NASA's stunning Perpetual Ocean animation visualizes ocean currents (Image: NASA/Goddard Space Flight Center)
Error Analysis with Logistic map example

\[ ||z' - \hat{z}'|| \leq \frac{||Rz' - \hat{K}_D Rz||}{1 - \epsilon} \]

rDMD: Logistic Map: \( X_{n+1} = 3.56994x_n(1-x_n) \)

Error: \( E = ||Z(t) - \hat{Z}(\hat{t})|| \)

Distortion(\( \epsilon \))

Error

Dim.(L) of Projected Space

Error
Summary

- DMD on projected space approximate Koopman Operator
- JL-theory grantee a Random projection. $L << N. \quad L \geq \frac{C \ln M}{e^2}$
- rDMD
  - Computationally Efficient simple algorithm
  - Reduce the storage cost.
- We will extend our current randomized method to extended DMD and kernel DMD.
References

The End
Random Projection

**Random projection to EDMD**

Extended DMD: \( \mathbf{K}_E(N \times N) = \Psi(X') \Psi(X)^\dagger \)

Random projection matrix \( B \) can be used to reduce the dimensionality of input/output space and the new operator is given by:

\[
\hat{\mathbf{K}}_E(L \times L) = B_{L \times N} \Psi(X') \left( B_{L \times N} \Psi(X) \right)^\dagger \\
B^\dagger \hat{\mathbf{K}}_E = \mathbf{K}_E B^\dagger
\]

Therefore \( \hat{\mathbf{K}}_E \) and \( \mathbf{K}_E \) have common eigenvalues.

If \( \hat{\nu} \) is an eigenvector of \( \hat{\mathbf{K}}_E \), then \( \nu = B^\dagger \hat{\nu} \) is an eigenvector for \( \mathbf{K}_E \).

**Random Projection with kernelized DMD**

We can construct \( B = \Psi(b)^T = [\psi(b_1) \psi(b_2) \ldots \psi(b_L)]^T \) by randomly sampled \( b_i \in \mathcal{M} \).

Then entries \( \hat{\Psi}(X)_{ij} = k(b_i, x_j) \) and \( \hat{\Psi}(X)_{ij} = k(b_i, x'_j) \) can be calculated implicitly.
Proposed randomized kernelized DMD

Koopman Operator on projected space $\mathcal{F}_N$

$\mathcal{F}_N = \text{span}\{\psi_i: i = 1, \ldots, N\}$

$\mathcal{F}_N = \text{span}\{\psi_i: i = 1, \ldots, N\}$

$\mathcal{F}_N = \text{span}\{\psi_i: i = 1, \ldots, N\}$

$\psi_i: \mathcal{M} \to \mathbb{R}$

$\psi_i: \mathcal{M} \to \mathbb{R}$

$\Psi(X) = \begin{pmatrix}
\psi_1(x_1) & \cdots & \psi_1(x_M) \\
\vdots & \ddots & \vdots \\
\psi_N(x_1) & \cdots & \psi_N(x_M)
\end{pmatrix}_{N \times M}$

$\Psi(Y) = \begin{pmatrix}
\psi_1(y_1) & \cdots & \psi_1(y_M) \\
\vdots & \ddots & \vdots \\
\psi_N(y_1) & \cdots & \psi_N(y_M)
\end{pmatrix}_{N \times M}$

There are $M$ data points of $\psi = [\psi_1 \ldots \psi_N]^T \in \mathbb{R}^N$

Goal is to reduce this dimension to $L \ll N$.

Use random projection matrix $B \in \mathbb{R}^{L \times N}$

$L \geq \frac{C \ln M}{\epsilon^2}$ with $\epsilon$ - isometry.

$\Phi(X) = B\Psi(X)$

$\Phi(Y) = B\Psi(Y) \in \mathbb{R}^{L \times M}$

We choose $B = \Psi(b)^T = [\psi(b_1), \ldots, \psi(b_L)]^T$, where $b_i \in \mathcal{M}$ for $i = 1, \ldots, L$ and selected randomly.

$\Phi(X) = \Psi(b)^T\Psi(X) = G$

Now instead of calculate $G, Q$ by inner product of the feature space, we can reduce the calculation by using a kernel $k$.

This kernel should produce $L$ dimensional feature space.

$G_{ij} = k(b_i, x_j)$

$Q_{ij} = k(b_i, y_j)$

$\hat{A}$

$K_{rk} = \arg \min \|KG - Q\|_F$