From E-DMD to D-DMD: A Leap in Koopman Operator Approximation

Contents

1 From E-DMD to D-DMD: A Leap in Koopman Operator Approximation

The shift from Extended Dynamic Mode Decomposition (E-DMD) to Deep Dynamic Mode Decomposition (D-DMD) marks a significant advance in data-driven analysis of nonlinear dynamical systems via the Koopman operator. As detailed in ”Learning Deep Neural Network Representations for Koopman Operators of Nonlinear Dynamical Systems,” this transition is driven by E-DMD’s limitations with complex systems and the need for a more automated, scalable, and accurate method. (Figures 1 and 2 of the referenced paper visually contrast E-DMD and D-DMD).

1.1 Extended Dynamic Mode Decomposition (E-DMD): Foundation and Hurdles

E-DMD approximates the Koopman operator, which linearly describes observable evolution in dynamical systems, even nonlinear ones. It lifts state variables to a higher-dimensional space using a predefined dictionary of basis functions, where the dynamics ideally become linear.

1.1.1 Mathematical Formulation of E-DMD

For a discrete-time nonlinear system $x_{n+1} = f(x_n)$ with observables $y_n = h(x_n)$, the Koopman operator $\mathcal{K}$ satisfies $\mathcal{K}\psi(x_n) = \psi(x_{n+1})$, where $\psi(x_n)$ is an observable function in a lifted space. Ideally, the span of the chosen dictionary functions ψ_i forms a *Koopman-invariant subspace*, meaning that applying $\mathcal{K}$ to any function in this subspace yields another function within the same subspace. This ensures the linear operator K can fully describe the evolution within this lifted space.

E-DMD uses a chosen dictionary $\mathcal{D} = [\psi_1(x) \quad \dots \quad \psi_m(x)]$. **Figure 1 of the paper illustrates this: a scientist manually curates $\mathcal{D}$ to lift observables before estimating K.** From time-series data $(x_1, \dots, x_N)$, matrices of lifted states are formed: $Y_p = [\psi(x_0) \quad \dots \quad \psi(x_{N-1})]$
$Y_f = [\psi(x_1) \quad \dots \quad \psi(x_N)]$
(Note: The paper’s indexing/matrix orientation differs slightly, e.g., its Eq. 4 for Y_f, Y_p').

The goal is to find K (Koopman operator approximation) minimizing reconstruction error:

$$\min_K ||Y_f - KY_p||_2$$

A regularized version can prevent overfitting:

$$\min_K ||Y_f - KY_p||_2 + \lambda||K||_{2,1}$$

1.1.2 Limitations of E-DMD

1. **Manual Dictionary Selection Invariance Challenge:** Choosing dictionary functions (ψ_i) that form a good approximation of a Koopman-invariant subspace is critical but manual, needing domain knowledge and extensive tuning (see Figure 1’s ”manual evaluation” loop). For complex nonlinear systems, finding a true finite-dimensional invariant subspace with predefined bases (like polynomials) is often infeasible. Poor choices yield inaccurate models.

2. **Computational Complexity Scalability of Fixed Bases:** Describing complex systems often requires an expansive set of basis functions (e.g., orthonormal polynomials). As noted in the reference paper, this approach ”does not scale well,” leading to a combinatorial explosion in dictionary size (e.g., mn functions, plus cross terms). This makes computing K expensive and memory-intensive.

3. **Overfitting:** Large, fixed dictionaries relative to data can lead to overfitting.

4. **Fixed Dictionary:** The dictionary is static; it cannot adapt to data during learning.

1.2 Deep Dynamic Mode Decomposition (D-DMD): Automation and Enhancement

D-DMD uses deep neural networks to simultaneously learn the basis function dictionary and the Koopman operator, aiming to overcome E-DMD's issues by learning representations that better approximate Koopman-invariant subspaces.

1.2.1 Mathematical Formulation of D-DMD

The dictionary is the output of a deep neural network, $N_\Psi(x_n, \theta)$, with parameters θ. The lifting is $\psi(x_n) = N_\Psi(x_n, \theta)$. **Figure 2 of the paper shows this: data x_t, x_{t+1} pass through N_Ψ (dictionary $\mathcal{D}_{NN}$) for lifted states. K predicts $\psi_{est}(x_{t+1})$ from $\psi(x_t)$. A loss function on the prediction error updates K and θ, enabling "Automated dictionary learning."** The network implicitly learns functions aiming to span an effective Koopman-invariant subspace.

The learning problem jointly optimizes K and θ:

$$\min_{K,\theta} ||N_\Psi(x_{n+1}, \theta) - KN_\Psi(x_n, \theta)||_2 + \lambda_1||K||_2 + \lambda_2||\theta||_1$$

This seeks network parameters θ and an operator K for accurate linear prediction in the learned lifted space.

1.2.2 Motivation for D-DMD

1. **Automated Dictionary Learning for Invariance:** DNNs automatically learn effective basis functions from data, aiming to discover representations that approximate Koopman-invariant subspaces without manual, expert-driven dictionary design (see Figure 2's automated loop).

2. **Improved Scalability Expressivity:** DNNs efficiently model complex nonlinearities, potentially finding more compact and effective dictionaries than combinatorially large fixed bases, thus handling higher-dimensional systems better.

3. **Enhanced Prediction Accuracy:** Optimized, learned dictionaries that better capture invariant properties yield higher-fidelity Koopman models, improving long-term forecasting.

4. **Simultaneous Learning:** The dictionary and K are learned together (Figure 2), allowing co-adaptation for a better linear representation.

1.3 Key Feature Comparison: E-DMD vs. D-DMD

A concise comparison highlights D-DMD's advantages:

The Core Idea: Lifting Dynamics Koopman Invariance

- **Shared Goal (Both E-DMD D-DMD):** Start with complex nonlinear system data (x_n or observations y_n). "Lift" this data using a set of functions (the dictionary ψ) into a new space. The fundamental aim is for this new space to be (or approximate) a Koopman-invariant subspace, where dynamics become linear: $\psi(x_{n+1}) \approx K\psi(x_n)$.

The Dictionary ψ: Who Chooses It Scalability Challenges

- **E-DMD:** The scientist manually chooses and fixes ψ upfront (e.g., polynomials, RBFs).
  - Achieving good Koopman invariance is challenging with fixed bases for complex systems.

  - Orthonormal polynomial bases (and similar) suffer from poor scalability: their size explodes combinatorially with system dimension and desired complexity, leading to computational bottlenecks and overfitting.

- **D-DMD:** The dictionary itself is learned by a Deep Neural Network (N_Ψ).
  - The network N_Ψ learns optimal functions composing ψ directly from data, aiming to find an effective, data-driven approximation of a Koopman-invariant subspace.
  - Aims to find more efficient/compact representations than manually specified, large polynomial bases.

Finding K Optimizing ψ (The Learning Process):

- **E-DMD:** A two-step process. 1. ψ is fixed. 2. K is found (typically via least squares / pseudoinverse $Y_f Y_p^\dagger$ on lifted data). This is a direct solution for K given ψ, but ψ is not optimized.
- **D-DMD:** An end-to-end, simultaneous process. The dictionary (network N_Ψ with parameters θ) and K are learned together using iterative gradient descent. Both θ and K are adjusted to best satisfy $\psi(x_{n+1}, \theta) \approx K\psi(x_n, \theta)$.

Adaptability Optimality of the Dictionary:

- **E-DMD:** K's quality depends heavily on the initial, fixed ψ. A suboptimal ψ (e.g., one that doesn't span an invariant subspace well) leads to a suboptimal K. Dictionary is static.
- **D-DMD:** The neural network actively seeks a ψ best suited for linearizing the data's specific dynamics and approximating an invariant subspace. Dictionary is adaptive.

Handling Partial Observations (y_n) Complex Data:

- **E-DMD:** Requires careful manual design of $\psi(y_n)$ or techniques like time-delay embedding (more manual tuning).
- **D-DMD:** N_Ψ can learn complex features from y_n, implicitly representing/reconstructing unobserved state information as part of finding the best ψ.

Optimization Method, Scalability, Complexity:

- **E-DMD:** Relies on direct solution for K (e.g., pseudoinverse) given a fixed ψ.
  - Computationally expensive and memory-intensive for large manual dictionaries (like high-order polynomials).
  - Requires full data matrices for pseudoinverse; can be numerically sensitive.
  - Scalability is limited by the size and manual construction of the explicit dictionary.
- **D-DMD:** Employs iterative gradient descent (e.g., Adam, AdaGrad) for joint optimization of N_Ψ (dictionary) and K.

  - Highly scalable using mini-batch processing, suitable for large datasets and complex (deep) learned dictionaries.

  - Gradient descent is well-suited for optimizing high-dimensional neural network parameters.

  - More robust to problem scale, though involves non-convex optimization.

The Big Payoff / Overall Advantage:

- **E-DMD:** Effective if a good dictionary (approximating an invariant subspace) is known or found via trial and error. Often limited by scalability of fixed bases.

- **D-DMD:** Learning a potentially superior and more efficient dictionary (better approximating an invariant subspace) often yields a more accurate K. This significantly improves long-term prediction, especially for highly nonlinear, partially observed systems where good manual dictionary design is difficult.