RICE | CTBP
Center for Theoretical Biological Physics

Tensor-Based Koopman Analysis of Stochastic Dynamics

Feliks Nüske
Rice University
May 20, 2019

Koopman Operators

- We aim at computing eigenpairs of the Koopman operator for a stochastic system:

$$\mathcal{K}_\tau f(x) = \mathbb{E}^x \left[f(X_\tau) \right]$$

- **Galerkin Projection (EDMD)** onto $\mathbb{V} = \mathrm{span}\{f_i\}_{i=1}^N$

$$\mathbf{K}_\tau = (\mathbf{C}^0)^{-1}\mathbf{C}^\tau$$

$$\mathbf{C}^0 = \langle f_i, f_j \rangle_\mu$$

$$\mathbf{C}^\tau = \langle f_i, \mathcal{K}_\tau f_j \rangle_\mu$$

-

Noé and Nüske, *SIAM Multiscale Model. Simul.* (2013), Williams et al, *J. Nonlinear Sci.* (2015), Klus et al., *J. Nonlinear Sci.* (2018)

Data-Based Approximation

- Given snapshots of the process: $\{X_t\}_{t=1}^{m+\tau}$
- the **data-based** approximation of the Galerkin matrices is

$$\mathbf{C}^0(i,j) = \frac{1}{m}\sum_{t=1}^{m} f_i(X_t) f_j(X_t) = \frac{1}{m}\mathbf{X}\mathbf{X}^T$$

$$\mathbf{C}^\tau(i,j) = \frac{1}{m}\sum_{t=1}^{m} f_i(X_t) f_j(X_{t+\tau}) = \frac{1}{m}\mathbf{X}\mathbf{Y}^T$$

- where $\mathbf{X}(i,t) = f_i(X_t), \ \mathbf{Y}(i,t) = f_i(X_{t+\tau})$

Noé and Nüske, *SIAM Multiscale Model. Simul.* (2013), Williams et al, *J. Nonlinear Sci.* (2015), Klus et al., *J. Nonlinear Sci.* (2018)

AMUSE Algorithm

- To solve the eigenvalue problem for $\mathbf{K}_\tau$, we can proceed as follows:
- 1. SVD of **X**: $\mathbf{X} = \mathbf{V}\Sigma\mathbf{W}^T$
- 2. Diagonalize:

$$\mathbf{M} = \Sigma^{-1}\mathbf{V}^T\mathbf{X}\mathbf{Y}^T\mathbf{V}\Sigma^{-1} = \mathbf{W}^T\mathbf{Y}^T\mathbf{V}\Sigma^{-1}.$$

Tong et al, *IEEE International Symposium on Circuits and Systems*, (1990)
Klus et al., *J. Nonlinear Sci.* (2018)

Tensor-Product Basis

- For high-dimensional systems, the challenge is to use a sufficiently powerful basis set.
- For a d-dimensional system, we would like to use a **tensor-structured basis**:
- Univariate basis sets: $\mathbb{V}_p = \mathrm{span}\{f^p_{i_p}(x_p)\}_{i=1}^{n_p}$
- Tensor-Product Space: $\mathbb{V} = \mathbb{V}_1 \otimes \ldots \otimes \mathbb{V}_d$
- Elements of $\mathbb{V}$ are **tensors** in $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}$

Tensor Train Format (TT format)

- We choose to represent tensors in TT format:

$$\mathbf{A}(i_1, \ldots, i_d) = \mathbf{U}_1(i_1) \ldots \mathbf{U}_d(i_d)$$

- with cores $\mathbf{U}_p \in \mathbb{R}^{r_{p-1} \times n_p \times r_p} \quad (r_0 = r_d = 1),$

$$\mathbf{U}_p(i_p) = \mathbf{U}_p(:, i_p, :)$$

- TT-format is motivated by **Higher-Order SVD (HOSVD).**

Oseledets and Tyrtyshnikov, *SIAM J. Sci. Comput.* (2009)
Oseledets, *SIAM J. Sci. Comput.* (2011)

HOSVD

- Mode-k unfolding: $\mathbf{A}_{|k} \in \mathbb{R}^{\prod_{p=1}^{k} n_p \times \prod_{p=k+1}^{d} n_p}$
- HOSVD:

Setting $\mathbf{V}_0 = \mathbf{A}_{|1}$, and $r_0 = 1$, we repeat the following steps for $k = 1, \ldots, d-1$:

- Re-shape $\mathbf{V}_{k-1}$ to shape $r_{k-1}n_k \times \prod_{p=k+1}^{d} n_p$.
- Compute a compact SVD $\mathbf{V}_{k-1} = \mathbf{U}_k \mathbf{V}_k^T$ of rank r_k.
- Re-shape $\mathbf{U}_k$ to become the k-th core.

Oseledets and Tyrtyshnikov, *SIAM J. Sci. Comput.* (2009)
Oseledets, *SIAM J. Sci. Comput.* (2011)

HOSVD Truncation

- For a tensor in TT-format, the same procedure can be used to reduce the ranks:

Setting $\mathbf{V}_0 = \mathbf{U}_1$, we repeat the following steps for $k = 1, \ldots, d-1$:

- Re-shape $\mathbf{V}_{k-1}$ to shape $r_{k-1}n_k \times r_k$.
- Compute a reduced SVD $\mathbf{V}_{k-1} = \mathbf{U}'_k \mathbf{V}_k^T$ of (lower) rank, update r_k.
- Re-shape $\mathbf{U}'_k$ to become the k-th core.
- Update $\mathbf{V}_k$ by contraction of $\mathbf{V}_k^T$ and $\mathbf{U}_{k+1}$.

- Insight: the first k cores provide a TT-representation of the **left singular vectors** of the **k-th unfolding**.

Oseledets and Tyrtyshnikov, *SIAM J. Sci. Comput.* (2009), Oseledets, *SIAM J. Sci. Comput.* (2011)

TT-Decomposition of Time Series

- Consider time series in data-based Galerkin problem:

$$\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d \times m}$$

- **Exact TT-representation** of rank m:

$$\mathbf{X}(i_1, \ldots, i_d, t) = \mathbf{U}_1(i_1) \ldots \mathbf{U}_d(i_d)\mathbf{U}_{d+1}(t)$$
$$\mathbf{U}_1(i_1, t) = f^1_{i_1}(X_t),$$
$$\mathbf{U}_p(t, i_p, t') = \delta(t, t') f^p_{i_p}(X'_t),\ 2 \leq p \leq d$$
$$\mathbf{U}_{d+1}(t, t') = \delta(t, t').$$

- We can use **HOSVD-truncation** to reduce ranks.

Gelß et al, *J. Comput. Nonlinear Dyn.* (2019)
- Klus et al, *Nonlinearity,* (2018)

TEDMD

- Performing HOSVD truncation and retaining the last diagonal factor provides left singular vectors of

$$\mathbf{X} = \mathbf{X}_{|d} = \mathbf{V}\Sigma\mathbf{W}^T$$

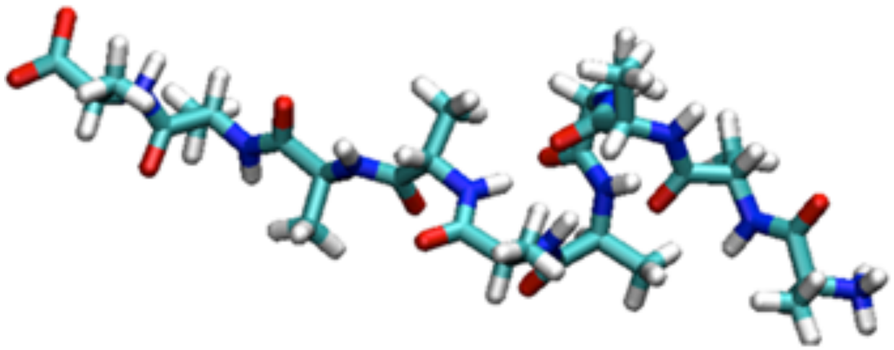
- with **V** in TT-format. With **Y** also in TT-format, we can solve the AMUSE problem efficiently.

$$\mathbf{M} = \mathbf{W}^T\mathbf{Y}_{|d}^T\mathbf{V}\Sigma^{-1}.$$

- It can be shown that **V** selects a linear subspace of full tensor space.

Example 1: Deca Alanine

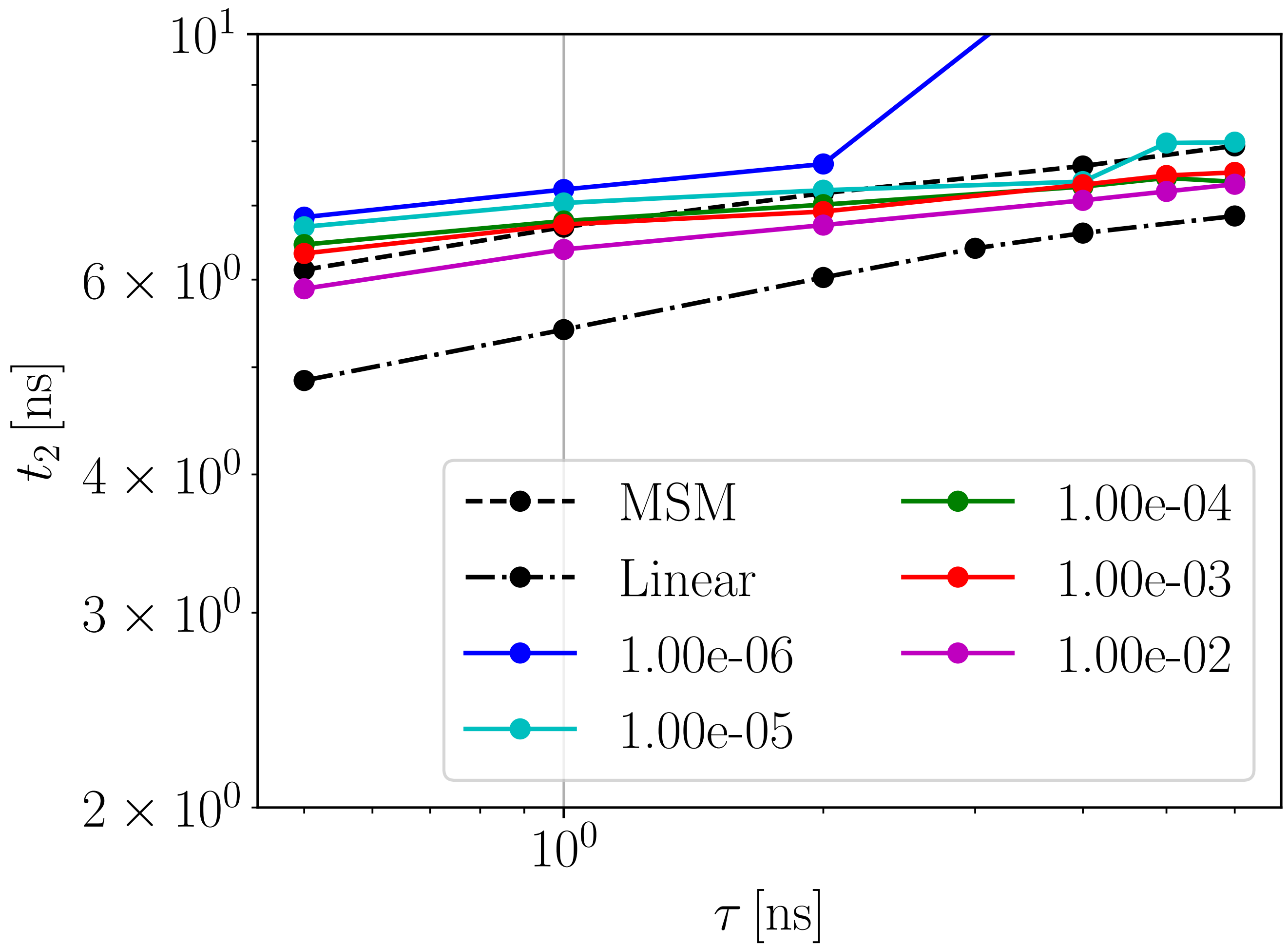
- Molecular Dynamics Simulations of Deca Alanine



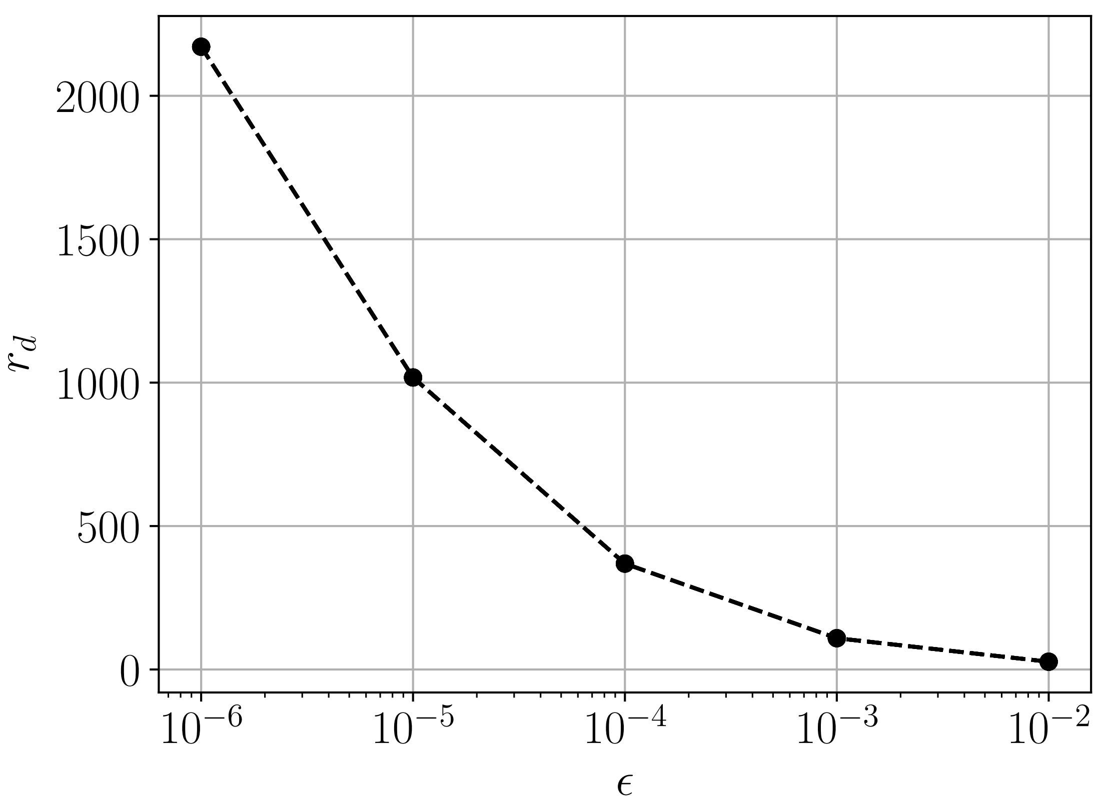
- Coordinates are d = 10 backbone torsion angles.
- Univariate bases comprised of the constant and 3 or 4 Gaussian functions.

Example 1 (Continued)

- Apply TEDMD for a sequence of truncation thresholds:
- Monitor **implied timescales** $t_i = \dfrac{\tau}{\log(\lambda_i(\tau))}$



Ranks



Example 1 (Continued)

- TEDMD helps us find accurate approximations of slow eigenpairs for Deca Alanine.
- Rank-truncation helps to find efficient representation and to avoid over-fitting.

Conclusions

- We have suggested to solve the Koopman eigenvalue problem using tensor-structured basis sets.
- TEDMD provides a tensor train approximation to this eigenvalue problem, which is also a projection onto a lower-dimensional subspace.
- Rank truncation helps to verify low-rank assumption and to avoid over-fitting.

Thank you for your attention!

Joint Work with

Stefan Klus (FU Berlin) and Patrick Gelß (FU Berlin)

RICE | CTBP Center for Theoretical Biological Physics

NSF

THE Welch FOUNDATION