Data-driven transfer operator approximation, model reduction, and system identification

Stefan Klus

joint work with: Andreas Bittracher, Patrick Gelß, Brooke Husic, Mattes Mollenhauer, Krikamol Muandet, Jan-Hendrik Niemann, Feliks Nüske, Sebastian Peitz, Ingmar Schuster, Cecilia Clementi, Jens Eisert, Frank Noé, Christof Schütte

Freie Universität Berlin

Scaling Cascades in Complex Systems

MATH+

Transfer operators

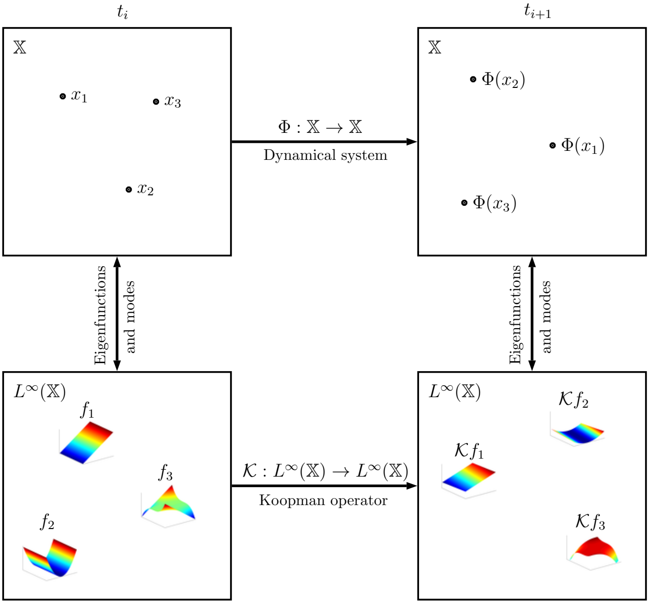
- Stochastic process $\{X_t\}_{t\geq 0}$ defined on $\mathbb{X} \subset \mathbb{R}^d$.
- Probability density: $\mathbb{P}[X_{t+\tau} \in \mathbb{A} \mid X_t = x] = \int_{\mathbb{A}} p_\tau(y \mid x) \, \mathrm{d}y$.

Definition (Transfer operators)

1. $\mathcal{P}^\tau p(y) = \int p_\tau(y \mid x) p(x) \, \mathrm{d}x.$
2. $\mathcal{T}^\tau u(y) = \frac{1}{\pi(y)} \int p_\tau(y \mid x) \, \pi(x) \, u(x) \, \mathrm{d}x.$
3. $\mathcal{K}^\tau f(x) = \int p_\tau(y \mid x) f(y) \, \mathrm{d}y = \mathbb{E}[f(X_{t+\tau}) \mid X_t = x].$

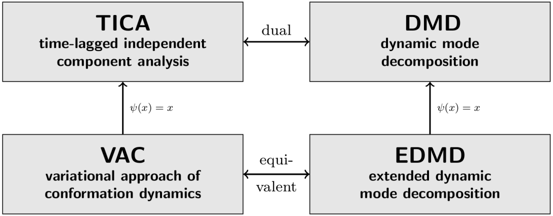
- π is the invariant density, i.e., $\mathcal{P}^\tau \pi = \pi$.
- $\mathcal{P}^\tau$ describes the evolution of densities and $\mathcal{K}^\tau$ the evolution of observables.

Pictorial representation



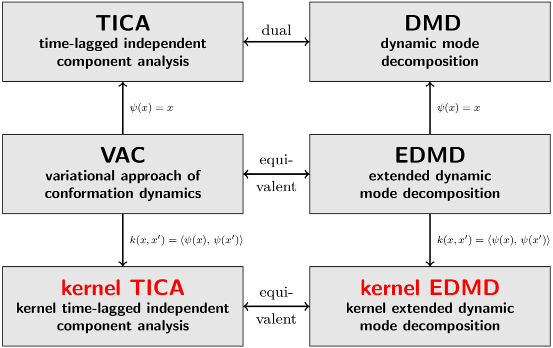
Williams et al. (2015).

Data-driven methods



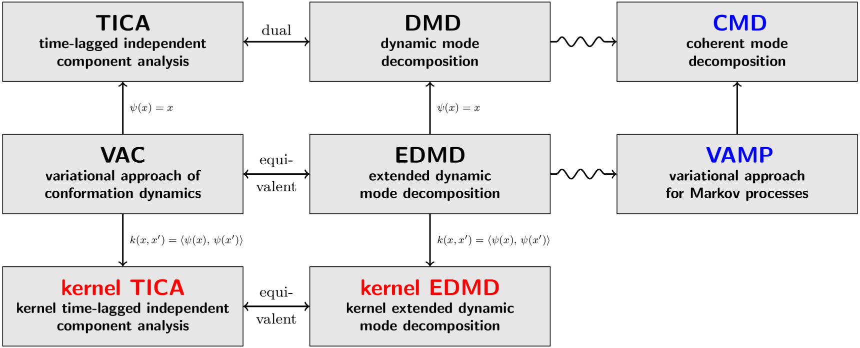
Molgedey et al. (1994). Bach et al. (2003). Schmid (2010). Noé et al. (2013). Nüske et al. (2014). Williams et al. (2015). Schwantes et al. (2015). Mardt et al. (2018).

Data-driven methods



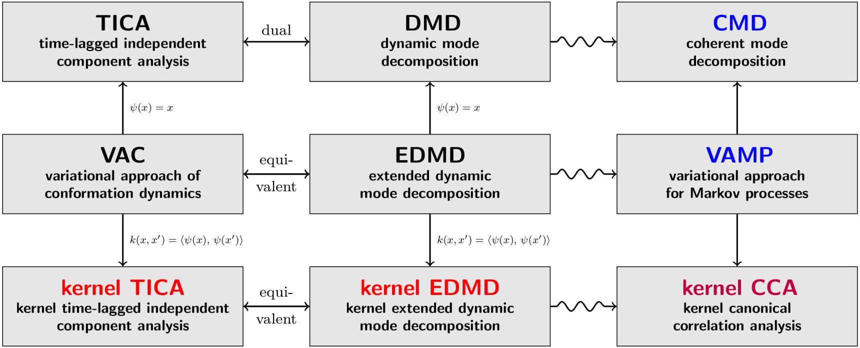
Molgedey et al. (1994). Bach et al. (2003). Schmid (2010). Noé et al. (2013). Nüske et al. (2014). Williams et al. (2015). Schwantes et al. (2015). Mardt et al. (2018).

Data-driven methods



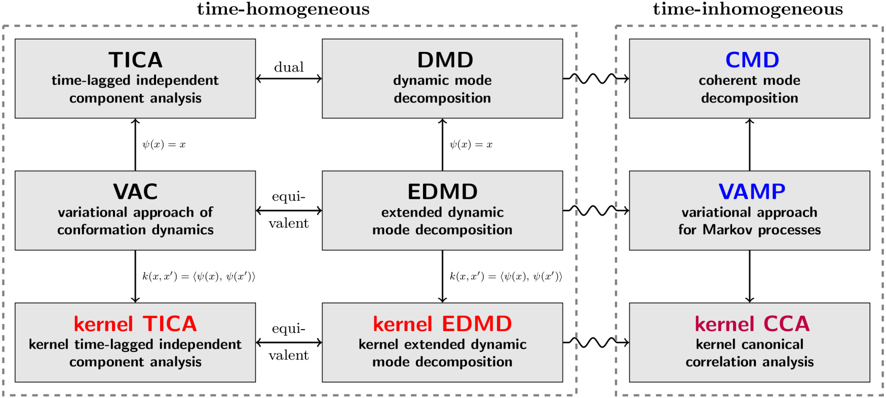
Molgedey et al. (1994). Bach et al. (2003). Schmid (2010). Noé et al. (2013). Nüske et al. (2014). Williams et al. (2015). Schwantes et al. (2015). Mardt et al. (2018).

Data-driven methods



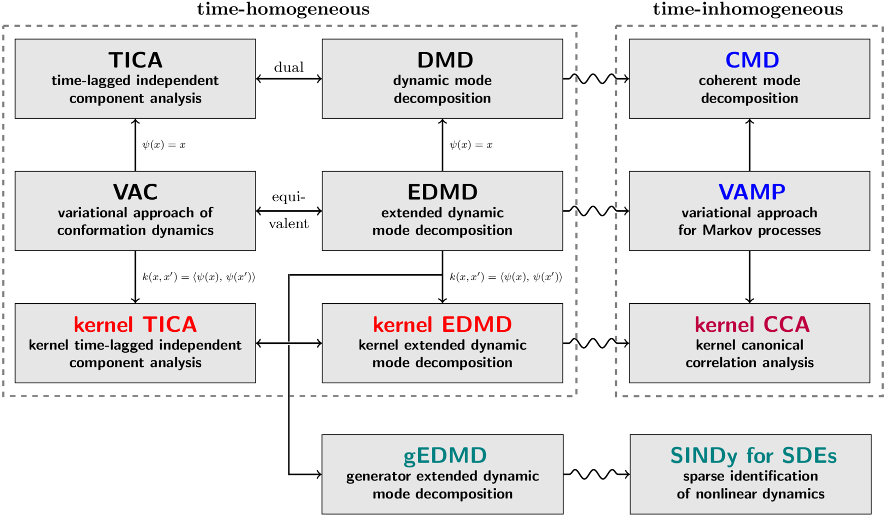
Molgedey et al. (1994). Bach et al. (2003). Schmid (2010). Noé et al. (2013). Nüske et al. (2014). Williams et al. (2015). Schwantes et al. (2015). Mardt et al. (2018).

Data-driven methods



Molgedey et al. (1994). Bach et al. (2003). Schmid (2010). Noé et al. (2013). Nüske et al. (2014). Williams et al. (2015). Schwantes et al. (2015). Mardt et al. (2018).

Data-driven methods

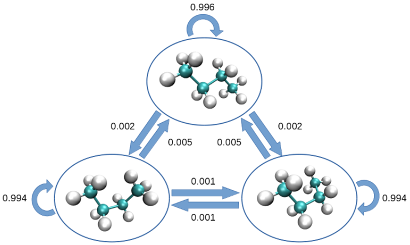


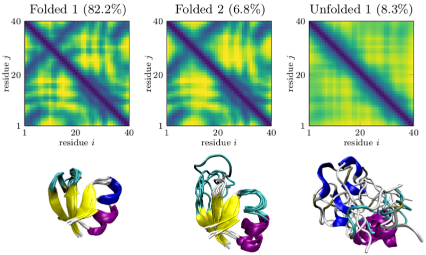
Molgedey et al. (1994). Bach et al. (2003). Schmid (2010). Noé et al. (2013). Nüske et al. (2014). Williams et al. (2015). Schwantes et al. (2015). Mardt et al. (2018).

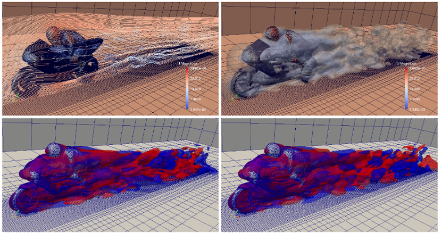
Outline

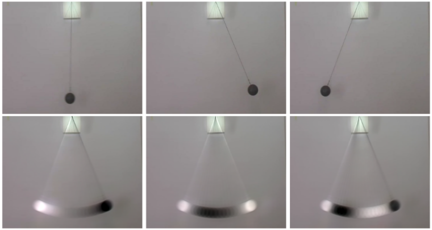
1. Kernel-based methods: Metastable sets
2. Kernel-based methods: Coherent sets
3. Generator EDMD

Metastable sets









Reproducing kernel Hilbert spaces

Definition (Reproducing kernel Hilbert space)

Let $\mathbb{X}$ be a set and $f\colon \mathbb{X} \to \mathbb{R}$. Then $\mathbb{H}$ is called an RKHS if there is a kernel $k\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ such that:

1. $\langle f, k(x,\cdot)\rangle_{\mathbb{H}} = f(x)$ for all $f \in \mathbb{H}$ and
2. $\mathbb{H} = \overline{\text{span}\{k(x,\cdot) \mid x \in \mathbb{X}\}}$.

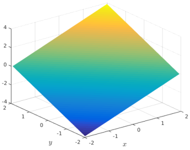
- In particular $\langle k(x,\cdot), k(x',\cdot)\rangle_{\mathbb{H}} = k(x,x')$.
- Function evaluation $f(x)$ equivalent to inner product evaluation in $\mathbb{H}$.
- Define $\phi(x) = k(x,\cdot)$ to be the *canonical feature map* such that $k(x,x') = \langle \phi(x), \phi(x')\rangle_{\mathbb{H}}$.

Schölkopf and Smola (2001). Shawe-Taylor and Cristianini (2004). Steinwart and Christmann (2008).

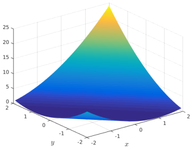
Example (Positive definite kernels on $\mathbb{R}^d$)

1. Linear kernel: $k(x, x') = x^\top x'$.
2. Polynomial kernel of degree p: $k(x, x') = (c + x^\top x')^p$.
3. Gaussian kernel: $k(x, x') = \exp\left(-\frac{1}{2\sigma^2}\|x - x'\|_2^2\right)$.
4. Laplacian kernel: $k(x, x') = \exp\left(-\frac{1}{\sigma}\|x - x'\|_2\right)$.

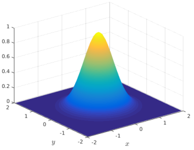
1



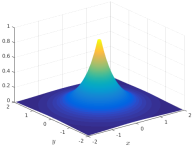
2



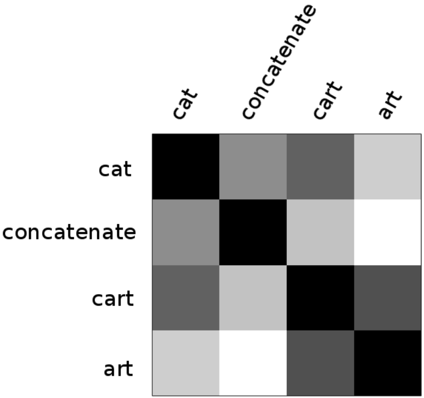
3

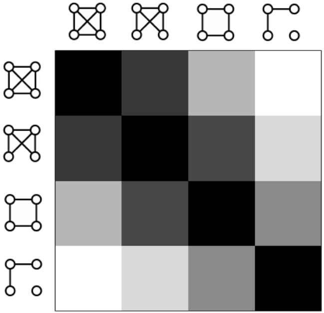


4



More kernels





Covariance operators

- (X, Y): random variable on $\mathbb{X} \times \mathbb{Y}$.
- ϕ and ψ: feature maps associated with the kernels k and l.

Definition (Covariance operators)

Define $\mathcal{C}_{XX} : \mathbb{H}_X \to \mathbb{H}_X$ and $\mathcal{C}_{YX} : \mathbb{H}_X \to \mathbb{H}_Y$ as

$$\mathcal{C}_{XX} = \int \phi(X) \otimes \phi(X) \, \mathrm{d}\mathbb{P}(X) = \mathbb{E}_X[\phi(X) \otimes \phi(X)],$$

$$\mathcal{C}_{YX} = \int \psi(Y) \otimes \phi(X) \, \mathrm{d}\mathbb{P}(Y, X) = \mathbb{E}_{YX}[\psi(Y) \otimes \phi(X)].$$

- Thus, $\mathbb{E}_{XY}[f(X)g(Y)] = \langle f, \mathcal{C}_{XY} g \rangle_{\mathbb{H}_X} = \langle \mathcal{C}_{YX} f, g \rangle_{\mathbb{H}_Y}$.

Proposition (Fukumizu et al. (2004))

$$\mathbb{E}_{Y|X}[g(Y) \mid X = \cdot\,] = \mathcal{C}_{XX}^{-1} \mathcal{C}_{XY} g.$$

Baker (1970/1973). Fukumizu et al. (2004). Muandet et al. (2017).

Empirical estimates of covariance operators

- $\mathbb{P}(X, Y)$ typically not known explicitly.
- Training data: $\mathbb{D}_{XY} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn i.i.d.
- Feature matrices:

$$\Phi = \left[\phi(x_1), \ldots, \phi(x_n)\right] \text{ and } \Psi = \left[\psi(y_1), \ldots, \psi(y_n)\right].$$

- Gram matrices: $G_{XX} = \Phi^\top \Phi$ and $G_{YY} = \Psi^\top \Psi$.

Definition (Covariance operator estimates)

$$\widehat{\mathcal{C}}_{XX} = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i) \otimes \phi(x_i) = \frac{1}{n} \Phi \Phi^\top,$$

$$\widehat{\mathcal{C}}_{YX} = \frac{1}{n} \sum_{i=1}^{n} \psi(y_i) \otimes \phi(x_i) = \frac{1}{n} \Psi \Phi^\top.$$

Transfer operators in RKHS

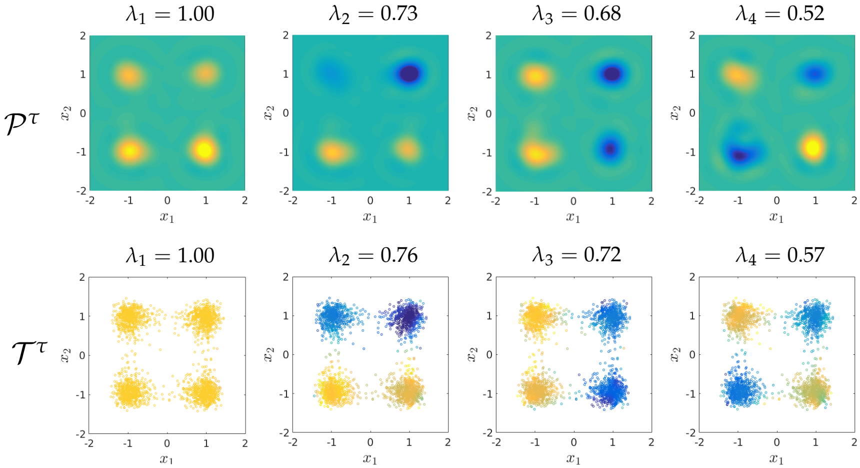
Theorem

1. $\mathcal{P}_k^\tau = (\mathcal{C}_{XX} + \varepsilon\mathcal{I})^{-1}\mathcal{C}_{YX} \approx \Psi\left(G_{XY}^{-1}(G_{XX} + n\varepsilon I)^{-1}G_{XY}\right)\Phi^\top.$
2. $\mathcal{K}_k^\tau = (\mathcal{C}_{XX} + \varepsilon\mathcal{I})^{-1}\mathcal{C}_{XY} \approx \Phi\,(G_{XX} + n\varepsilon I)^{-1}\,\Psi^\top.$

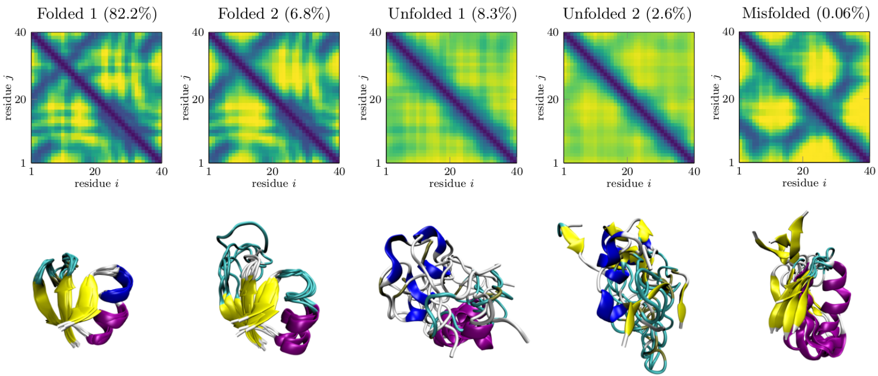
	Perron–Frobenius	Koopman
Kernel operator	$(G_{XX} + n\varepsilon I)^{-1}G_{XY}\,\mathbf{v} = \lambda\mathbf{v}$	$(G_{XX} + n\varepsilon I)^{-1}G_{YX}\,\mathbf{v} = \lambda\mathbf{v}$
Embedded operator	$G_{XY}(G_{XX} + n\varepsilon I)^{-1}\,\mathbf{v} = \lambda\mathbf{v}$	$G_{YX}(G_{XX} + n\varepsilon I)^{-1}\,\mathbf{v} = \lambda\mathbf{v}$
Eigenfunction	$\varphi = \Phi\,G_{XX}^{-1}\,\mathbf{v}$	$\varphi = \Phi\,\mathbf{v}$

Need to solve only an auxiliary matrix eigenvalue problem!

Quadruple-well problem



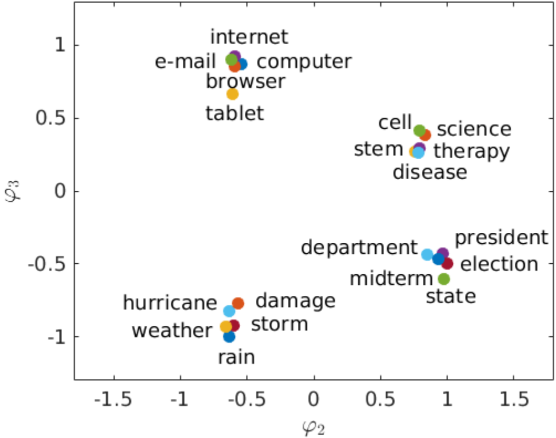
NTL9

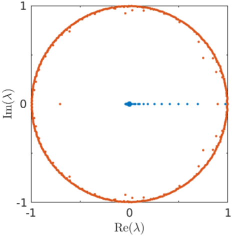


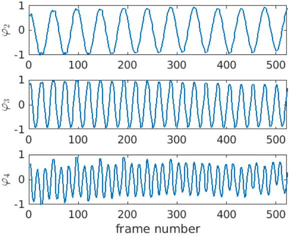
Folded 1 (82.2%)
Folded 2 (6.8%)
Unfolded 1 (8.3%)
Unfolded 2 (2.6%)
Misfolded (0.06%)
residue j
residue i
1
20
40

Text data

Reuters: *Macron's announcement Wednesday was the latest attempt by a government to find ways to handle the worldwide spread of disinformation on social media – "fake news", as U.S. President Donald Trump calls it. His plan would allow judges to block a website or a user account, in particular during an election, and oblige internet platforms to publish the names of those behind sponsored contents.*







$\lambda_2 \approx 0.69$

$\lambda_3 \approx 0.58$

$\lambda_4 \approx 0.49$

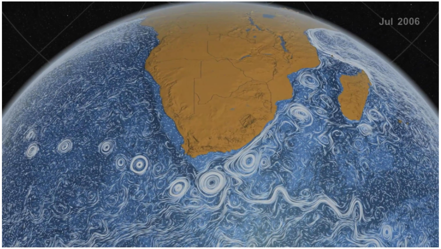
Outline

1 Kernel-based methods: Metastable sets

2 Kernel-based methods: Coherent sets

3 Generator EDMD

Coherent sets



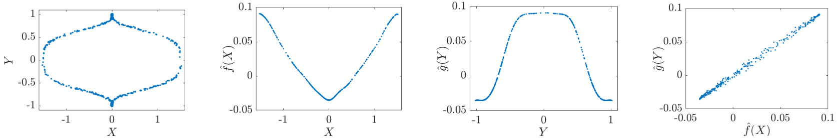
Perpetual ocean, NASA


Wingtip vortices, NASA Langley Research Center


Jupiter's great red spot, NASA/JPL/Space Science Institute


Math predicts oil spill's path through ocean, Futurity

Kernel CCA



- Kernel CCA is defined as

$$\sup_{\substack{f \in \mathbb{H}_X \\ g \in \mathbb{H}_Y}} \langle g, \mathcal{C}_{YX} f \rangle_{\mathbb{H}_Y} \quad \text{s.t.} \quad \begin{cases} \langle f, \mathcal{C}_{XX} f \rangle_{\mathbb{H}_X} = 1, \\ \langle g, \mathcal{C}_{YY} g \rangle_{\mathbb{H}_Y} = 1. \end{cases}$$

- The solution is given by

$$\begin{cases} \mathcal{C}_{YX} f = \rho \, \mathcal{C}_{YY} g, \\ \mathcal{C}_{XY} g = \rho \, \mathcal{C}_{XX} f. \end{cases}$$

- Can be rewritten as

$$\underbrace{\left(\mathcal{C}_{XX} + \varepsilon \mathcal{I}\right)^{-1} \mathcal{C}_{XY}}_{\mathcal{K}_k^\tau} \underbrace{\left(\mathcal{C}_{YY} + \varepsilon \mathcal{I}\right)^{-1} \mathcal{C}_{YX}}_{\mathcal{P}_k^\tau ?} f = \rho^2 f.$$

Melzer et al. (2001). Bach et al. (2003). Fukumizu et al. (2007).

Coherent sets

Lemma

The eigenvalue problem can be written as

$$\Phi B \Phi^\top \hat{f} = \rho^2 \hat{f},$$

with $B = (G_{XX} + n\varepsilon I)^{-1}(G_{YY} + n\varepsilon I)^{-1} G_{YY}$.

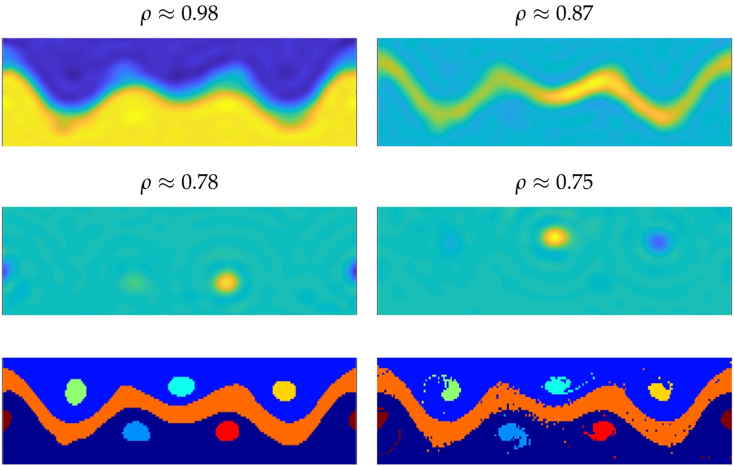
Algorithm (Kernel CCA)

1. *Choose a kernel* k *and regularization* ε.
2. *Compute the centered Gram matrices* G_{XX} *and* G_{YY}.
3. *Solve* $G_{XX}(G_{XX} + n\varepsilon I)^{-1}(G_{YY} + n\varepsilon I)^{-1} G_{YY} v = \rho^2 v$.

Kernel CCA applied to Lagrangian data results in coherent sets!

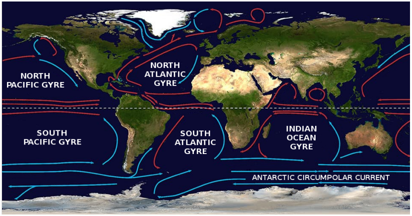
Froyland et al. (2010/2013/2015). Williams et al. (2015). Hadjighasem et al. (2016). Banisch et al. (2017).

Bickley jet

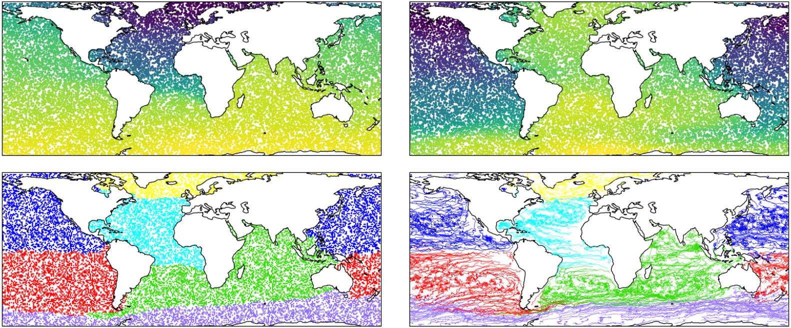


Rypina et al. (2007). Banisch et al. (2017).

Ocean data

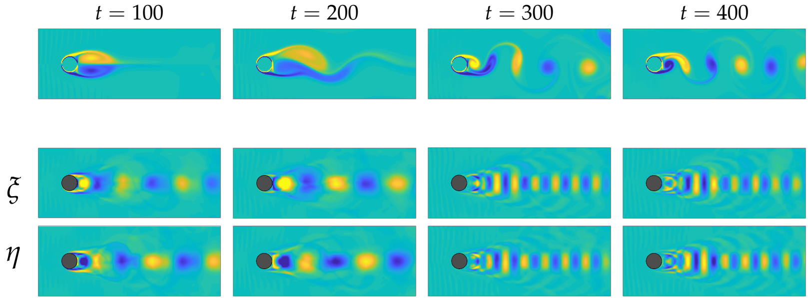


National Ocean Service: What is a gyre?



Coherent mode decomposition

- CMD is a hybrid of CCA and DMD.
- Uses the linear kernel given by $k(x, x') = x^\top x'$.
- We obtain two modes for each eigenvalue.



Outline

1 Kernel-based methods: Metastable sets

2 Kernel-based methods: Coherent sets

3 Generator EDMD

The Koopman generator

$$\mathrm{d}X_t = b(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}W_t, \quad a = \sigma\sigma^\top$$

Definition (Infinitesimal generator)

The generator $\mathcal{L}$ is defined by $\mathcal{L}f = \lim_{\tau \to 0} \frac{1}{\tau}\left(\mathcal{K}^\tau f - f\right).$

$$\mathcal{L}f = \sum_{i=1}^{d} b_i \frac{\partial f}{\partial x_i} + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} a_{ij}\frac{\partial^2 f}{\partial x_i \partial x_j}$$

$$\mathcal{L}^* f = -\sum_{i=1}^{d} \frac{\partial (b_i f)}{\partial x_i} + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2 (a_{ij} f)}{\partial x_i \partial x_j}$$

$$f(x) = x_i \quad \Rightarrow \quad (\mathcal{L}f)(x) = b_i(x)$$

$$f(x) = x_i x_j \quad \Rightarrow \quad (\mathcal{L}f)(x) = b_i(x)\,x_j + b_j(x)\,x_i + a_{ij}(x)$$

gEDMD

- Given basis functions $\{\psi_i\}_{i=1}^n$, define

$$\mathrm{d}\psi_k(x) = \sum_{i=1}^d b_i(x)\,\frac{\partial \psi_k}{\partial x_i}(x) + \frac{1}{2}\sum_{i=1}^d \sum_{j=1}^d a_{ij}(x)\,\frac{\partial^2 \psi_k}{\partial x_i \partial x_j}(x).$$

- Compute transformed data matrices

$$\Psi_X = \begin{bmatrix} \psi_1(x_1) & \dots & \psi_1(x_m) \\ \vdots & \ddots & \vdots \\ \psi_n(x_1) & \dots & \psi_n(x_m) \end{bmatrix}, \quad \mathrm{d}\Psi_X = \begin{bmatrix} \mathrm{d}\psi_1(x_1) & \dots & \mathrm{d}\psi_1(x_m) \\ \vdots & \ddots & \vdots \\ \mathrm{d}\psi_k(x_1) & \dots & \mathrm{d}\psi_k(x_m) \end{bmatrix}.$$

- Obtain matrix representation of the generator

$$\widehat{L}^\top = \mathrm{d}\Psi_X \Psi_X^+ = \left(\mathrm{d}\Psi_X \Psi_X^\top\right)\left(\Psi_X \Psi_X^\top\right)^+ = \widehat{A}\,\widehat{G}^+.$$

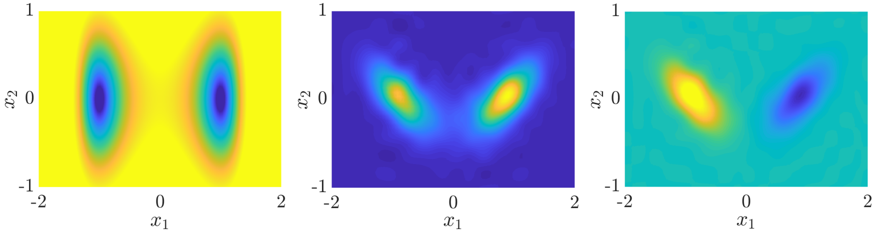
- Related to SINDy, KRONIC, KLT.

Brunton et al. (2016). Kaiser et al. (2017). Mauroy et al. (2016).

Double-well potential

$$\mathrm{d}X_t = b(X_t)\,\mathrm{d}t + \sigma(X_t)\,\mathrm{d}W_t$$

$$b(x) = \begin{bmatrix} -4x_1^3 + 4x_1 \\ -2x_2 \end{bmatrix}, \ \sigma(x) = \begin{bmatrix} 0.7 & x_1 \\ 0 & 0.5 \end{bmatrix}$$



Double-well potential

	1	x_1	x_2	x_1^2	$x_1 x_2$	x_2^2
1	0	0	0	0.49	0	0.25
x_1	0	4	0	0	0.5	0
x_2	0	0	-2	0	0	0
x_1^2	0	0	0	9	0	0
$x_1 x_2$	0	0	0	0	2	0
x_2^2	0	0	0	0	0	-4
x_1^3	0	-4	0	0	0	0
$x_1^2 x_2$	0	0	0	0	0	0
$x_1 x_2^2$	0	0	0	0	0	0
x_2^3	0	0	0	0	0	0
x_1^4	0	0	0	-8	0	0
$x_1^3 x_2$	0	0	0	0	-4	0
$x_1^2 x_2^2$	0	0	0	0	0	0
$x_1 x_2^3$	0	0	0	0	0	0
x_2^4	0	0	0	0	0	0

$$b(x) = \begin{bmatrix} -4x_1^3 + 4x_1 \\ -2x_2 \end{bmatrix}$$

$$\sigma(x) = \begin{bmatrix} 0.7 & x_1 \\ 0 & 0.5 \end{bmatrix}$$

$$a(x) = \begin{bmatrix} 0.49 + x_1^2 & 0.5x_1 \\ 0.5x_1 & 0.25 \end{bmatrix}$$

Double-well potential

	1	x_1	x_2	x_1^2	$x_1 x_2$	x_2^2
1	0	0	0	0.49	0	0.25
x_1	0	4	0	0	0.5	0
x_2	0	0	-2	0	0	0
x_1^2	0	0	0	9	0	0
$x_1 x_2$	0	0	0	0	2	0
x_2^2	0	0	0	0	0	-4
x_1^3	0	-4	0	0	0	0
$x_1^2 x_2$	0	0	0	0	0	0
$x_1 x_2^2$	0	0	0	0	0	0
x_2^3	0	0	0	0	0	0
x_1^4	0	0	0	-8	0	0
$x_1^3 x_2$	0	0	0	0	-4	0
$x_1^2 x_2^2$	0	0	0	0	0	0
$x_1 x_2^3$	0	0	0	0	0	0
x_2^4	0	0	0	0	0	0

$$b(x) = \begin{bmatrix} -4x_1^3 + 4x_1 \\ -2x_2 \end{bmatrix}$$

$$\sigma(x) = \begin{bmatrix} 0.7 & x_1 \\ 0 & 0.5 \end{bmatrix}$$

$$a(x) = \begin{bmatrix} 0.49 + x_1^2 & 0.5x_1 \\ 0.5x_1 & 0.25 \end{bmatrix}$$

Double-well potential

	1	x_1	x_2	x_1^2	$x_1 x_2$	x_2^2
1	0	0	0	0.49	0	0.25
x_1	0	4	0	0	0.5	0
x_2	0	0	-2	0	0	0
x_1^2	0	0	0	9	0	0
$x_1 x_2$	0	0	0	0	2	0
x_2^2	0	0	0	0	0	-4
x_1^3	0	-4	0	0	0	0
$x_1^2 x_2$	0	0	0	0	0	0
$x_1 x_2^2$	0	0	0	0	0	0
x_2^3	0	0	0	0	0	0
x_1^4	0	0	0	-8	0	0
$x_1^3 x_2$	0	0	0	0	-4	0
$x_1^2 x_2^2$	0	0	0	0	0	0
$x_1 x_2^3$	0	0	0	0	0	0
x_2^4	0	0	0	0	0	0

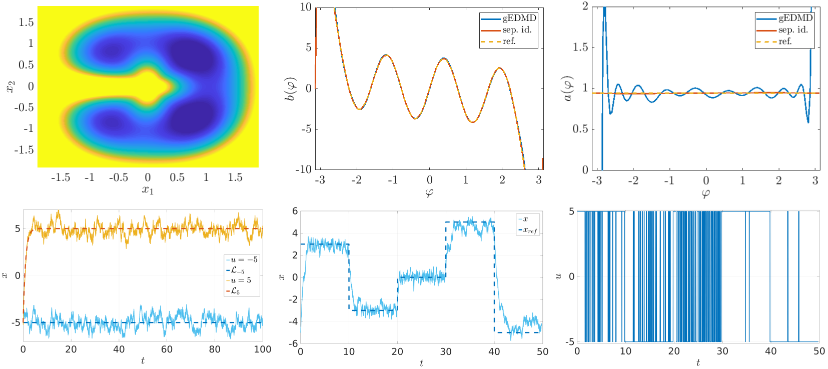
$$b(x) = \begin{bmatrix} -4x_1^3 + 4x_1 \\ -2x_2 \end{bmatrix}$$

$$\sigma(x) = \begin{bmatrix} 0.7 & x_1 \\ 0 & 0.5 \end{bmatrix}$$

$$a(x) = \begin{bmatrix} 0.49 + x_1^2 & 0.5x_1 \\ 0.5x_1 & 0.25 \end{bmatrix}$$

Applications

- Computation of eigenvalues, eigenfunctions, and modes.
- System identification and discovery of conservation laws.
- Learning coarse-grained models.
- Control.



MNIST and fashion MNIST

0 T-shirt/top
1 Trousers
2 Pullover
3 Dress
4 Coat
5 Sandal
6 Shirt
7 Sneaker
8 Bag
9 Ankle boot

Lecun et al. (1998). Xiao et al. (2017).

SINDy/MANDy for classification

- Basis functions are given by tensor products of the form

$$\mathbf{\Psi}(x) = \begin{bmatrix} \cos(\alpha\, x_1) \\ \sin(\alpha\, x_1) \end{bmatrix} \otimes \begin{bmatrix} \cos(\alpha\, x_2) \\ \sin(\alpha\, x_2) \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} \cos(\alpha\, x_d) \\ \sin(\alpha\, x_d) \end{bmatrix}.$$

- Solve minimization problem

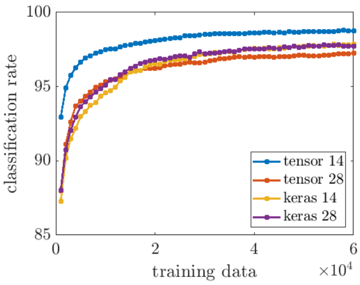
$$\min_{\mathbf{\Xi}} \sum_{i=1}^{m} \left\| y^{(i)} - \mathbf{\Xi}^\top \mathbf{\Psi}(x^{(i)}) \right\|_2^2.$$

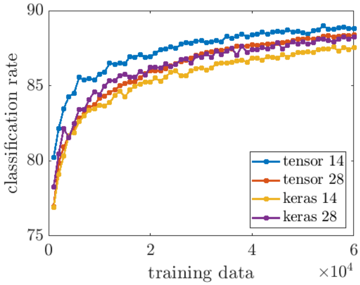
- Obtain classifier

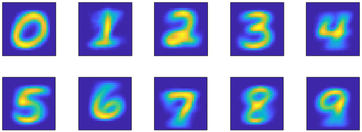
$$f(x) = \mathbf{\Xi}^\top \mathbf{\Psi}(x).$$

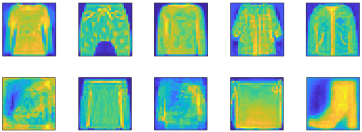
Stoudenmire et al. (2016). Gelß et al. (2019).

SINDy/MANDy for classification









Python implementation

https://github.com/sklus/d3s/

https://github.com/PGelss/scikit_tt

Thank you!