

Flow reconstruction using manifold learning and structure-preserving metrics

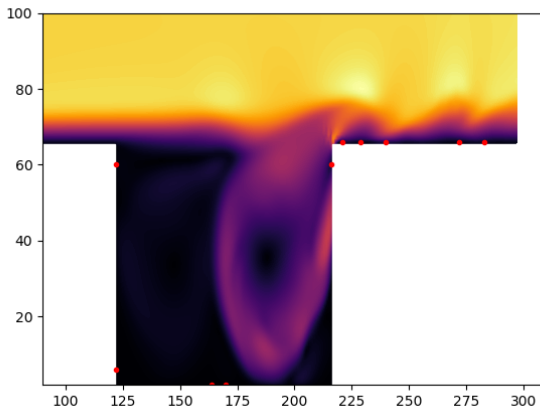
L. MATHELIN¹, A.A. GORODETSKY² & B. PODVIN¹

¹: U. Paris-Saclay, CNRS, LISN, France

²: Dpt. Aerospace Eng., U. Michigan, USA

SIAM CSE 2021

Motivation: Inverse problems



Goal: Estimate the flow field from limited wall-mounted sensors

Standard approach — ROM-based state estimation

Derive a reduced-order model: $\mathbf{y} \approx \hat{\mathbf{y}} = D \mathbf{x}$. D typically are PCA modes.

From a correlation kernel or a “training” (unsorted) sequence $Y := (\mathbf{y}^{(1)} \dots \mathbf{y}^{(n_{\text{snap}})})$:

$$Y \stackrel{\text{thin SVD}}{\approx} D \Sigma V^*.$$

For a given number n_D of retained modes, leads to the best approximation in the following sense:

$$\hat{Y} = D \hat{X} \in \arg \min_{\text{rank}[\tilde{Y}] \leq n_D} \|Y - \tilde{Y}\|_F \quad \text{with} \quad \hat{X} = \Sigma V^*.$$

- $\mathbf{y} \in \mathbb{R}^n$ field of interest,
- $D \in \mathbb{R}^{n \times n_D}$ the approximation basis [Dictionary],
- $\mathbf{x} \in \mathbb{R}^{n_D}$ the basis coefficients as estimated from the n_s sensors,
- $\mathbf{s} \in \mathbb{R}^{n_s}$, sensor information.

Standard approach — ROM-based state estimation

On site, what is measured is $\mathbf{s} = G\mathbf{y}$ only.

$G : \mathbf{y} \mapsto \mathbf{s}$: forward operator.

Observer such that

$$\hat{\mathbf{x}} \in \arg \min_{\tilde{\mathbf{x}} \in \mathbb{R}^{n_D}} \|\mathbf{s} - G D \tilde{\mathbf{x}}\|_2, \quad [\text{data misfit}]$$

or simply $\hat{\mathbf{x}} = (GD)^+ \mathbf{s}$. ← **requires** $n_s \geq n_D$ (if no additional hyp.)

The reconstructed field is finally:

$$\hat{\mathbf{y}} = D \hat{\mathbf{x}} = D (GD)^+ \mathbf{s}.$$

A dictionary learning algorithm

Without additional hypotheses, impossible to estimate $n_D > n_s$ modes

→ derive an over-complete dictionary D for sparse representation of the field $\mathbf{y} \in \mathcal{M}_y$ to be inferred

$$\mathbf{y} \underset{\epsilon}{\approx} D \mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^{n_D}$$

→ n_s -sparse approximation

$$\mathbf{y} \underset{\epsilon}{\approx} D \mathbf{x} \quad \text{with} \quad \|\mathbf{x}\|_0 \leq n_s, \quad \forall \mathbf{y} \in \mathcal{M}_y$$

$$\text{Find } \{D, X\} \in \arg \min_{\tilde{D}, \tilde{X}} \left\| Y - \tilde{D} \tilde{X} \right\|_F \quad \text{s.t.} \quad \left\| \tilde{\mathbf{x}}^{(i)} \right\|_0 \leq n_s, \quad \forall i, \quad X := \left(\mathbf{x}^{(1)} \dots \mathbf{x}^{(n_{\text{snap}})} \right).$$

A dictionary learning algorithm

→ use K-SVD algorithm

Repeat

- **Sparse Coding** : $X \in \arg \min_{\tilde{X}} \|Y - D\tilde{X}\|_F$ s.t. $\|\tilde{\mathbf{x}}^{(i)}\|_0 \leq n_s, \forall i.$

- **CodeBook Update** : Update D and X in order to lower $\|Y - DX\|_F$ while maintaining the support of $\{\mathbf{x}^{(i)}\}_i.$

A dictionary learning algorithm

→ use K-SVD algorithm

Repeat

- **Sparse Coding** : $X \in \arg \min_{\tilde{X}} \|Y - D\tilde{X}\|_F$ s.t. $\|\tilde{\mathbf{x}}^{(i)}\|_0 \leq n_s, \forall i.$

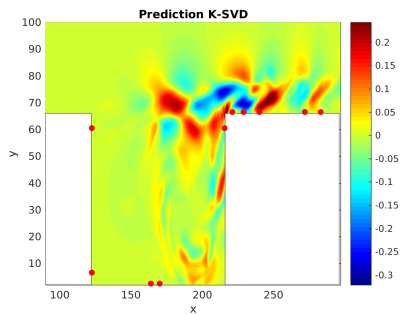
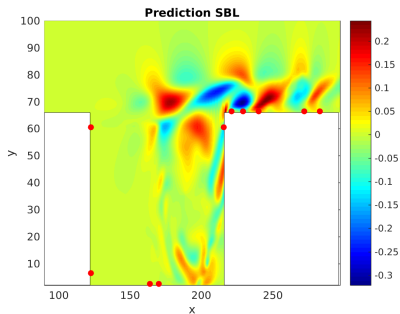
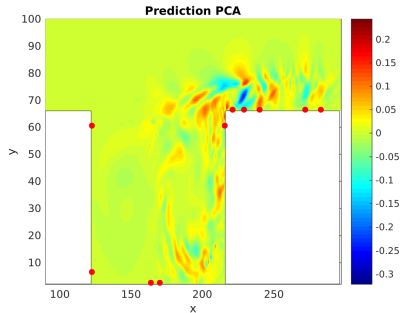
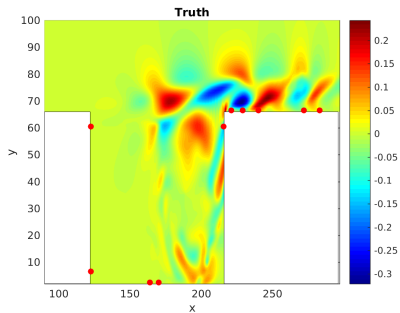
- **CodeBook Update** : Update D and X in order to lower $\|Y - DX\|_F$ while maintaining the support of $\{\mathbf{x}^{(i)}\}_i.$

But typically $\mathbf{x}^{(i)} \in \mathbb{R}^{n_D}$ **cannot** be estimated from measurements: **Observability issue**

→ determine D for estimating $\mathbf{x}^{(i)}$ from $\mathbf{s}^{(i)}$ instead of $\mathbf{y}^{(i)}$.

→ dictionary D both **accurate** and **observable**: $D(Y, S).$

Recovery performance



Beyond the linear realm. . .

- ⚠ Measurements may not be linearly related to the flow field,
- ⚠ The flow data may not live in a linear subspace.

Beyond the linear realm...

- ⚠ Measurements may not be linearly related to the flow field,
- ⚠ The flow data may not live in a linear subspace.

Followed approach

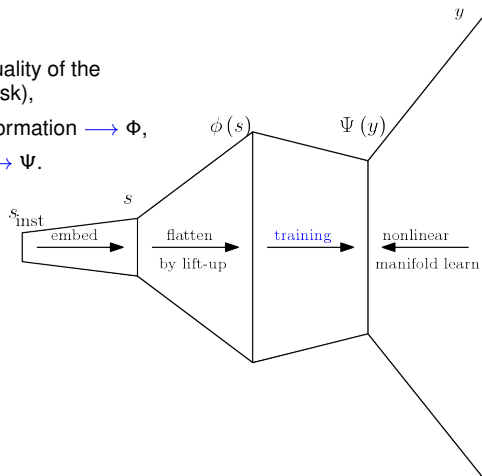
- Embed the measurements to improve the quality of the state vector proxy $\mathbf{s}_{\text{inst}} \rightarrow \mathbf{s}$ (lower Bayes risk),
- Unfold measurements to disentangle the information $\rightarrow \Phi$,
- Discover the manifold the flow data lie on $\rightarrow \Psi$.

Beyond the linear realm...

- ⚠ Measurements may not be linearly related to the flow field,
- ⚠ The flow data may not live in a linear subspace.

Followed approach

- Embed the measurements to improve the quality of the state vector proxy $\mathbf{s}_{\text{inst}} \rightarrow \mathbf{s}$ (lower Bayes risk),
- Unfold measurements to disentangle the information $\rightarrow \Phi$,
- Discover the manifold the flow data lie on $\rightarrow \Psi$.



Measurement features Φ

Multi Kernel Learning technique, $K_{i,j} := \langle \Phi_{\mathbf{s}_i}, \Phi_{\mathbf{s}_j} \rangle$,

$$K_{\boldsymbol{\mu}}(\cdot, \cdot) = \sum_l \mu_l K_l(\cdot, \cdot).$$

optimally aligned with flow features $\{\Psi_{\mathbf{y}_i}\}_i$:

$$\boldsymbol{\mu} \in \arg \max_{\tilde{\boldsymbol{\mu}}} \frac{\langle H K_{\tilde{\boldsymbol{\mu}}} H M \rangle_F}{\|H K_{\tilde{\boldsymbol{\mu}}}\|_F}, \quad H := I - n_{\text{snap}}^{-1} \mathbf{1} \mathbf{1}^T, \quad M := \Psi_Y^T \Psi_Y.$$

Measurement features Φ

Multi Kernel Learning technique, $K_{i,j} := \langle \Phi_{\mathbf{s}_i}, \Phi_{\mathbf{s}_j} \rangle$,

$$K_{\mu}(\cdot, \cdot) = \sum_l \mu_l K_l(\cdot, \cdot).$$

optimally aligned with flow features $\{\Psi_{\mathbf{y}_i}\}_i$:

$$\mu \in \arg \max_{\tilde{\mu}} \frac{\langle H K_{\tilde{\mu}} H M \rangle_F}{\|H K_{\tilde{\mu}}\|_F}, \quad H := I - n_{\text{snap}}^{-1} \mathbf{1} \mathbf{1}^T, \quad M := \Psi_Y^T \Psi_Y.$$

The Lagrangian writes (Bayes risk):

$$\begin{aligned} \mathcal{L} &:= \|\Psi_Y - \Psi_Y W \Phi(S)\|_F^2 + \lambda \|W\|_F^2, \\ \alpha &\propto \text{Tr} \left[\lambda W W^T + \Xi \Xi^T + \alpha^T (\Psi_Y - \Psi_Y W \Phi_S - \Xi) \right], \end{aligned} \quad \text{[Dual formulation]}$$

with $\Xi := \Psi_Y - \Psi_Y W \Phi_S$.

The adjoint variable matrix α satisfies a [Sylvester equation](#):

$$\lambda \left(\Psi_Y \Psi_Y^T \right)^{-1} \alpha + \alpha K_{\mu} = \Psi_Y.$$

Flow features Ψ

Nonlinear manifold learning: [Diffusion Maps](#)

Flow features Ψ

Nonlinear manifold learning: **Diffusion Maps**

Kernel N_{DM} to quantify the connectivity between snapshots $\{\mathbf{y}_i\}_i$:

$$N_{\text{DM}}(\mathbf{y}_i, \mathbf{y}_j) = \exp\left(-\text{dist}^2(\mathbf{y}_i, \mathbf{y}_j)\right).$$

With D_{eg} the **degree matrix**, the Markov matrix quantifies the transition probabilities:

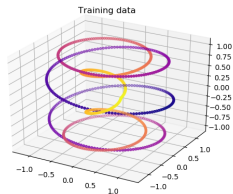
$$P = D_{\text{eg}}^{-1} N_{\text{DM}},$$

→ t -step (diffusion “time”) transition matrix:

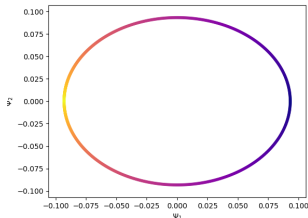
$$P^t = R \Gamma^t L^T.$$

Flow features Ψ_Y are coordinates on diffusion map:

$$\Psi_Y = \Gamma^t R^T.$$



Original space — High dimensional

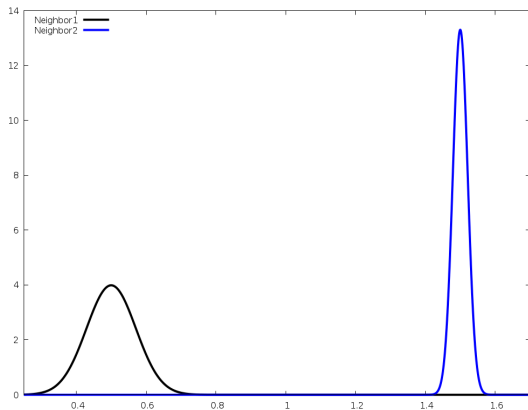


Feature space — Lower dimensional

Beyond L^2 ...

The Euclidean distance is a poor metric for reconstruction quality!

A motivation with 1-D point vortices:

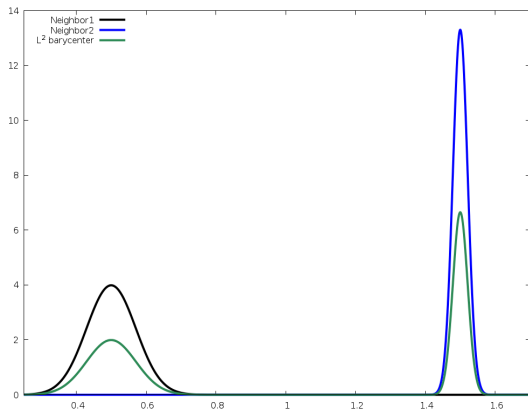


Large L^2 error, as soon as the two vortices are disjoint \rightarrow inconsistent.

Beyond L^2 ...

The Euclidean distance is a poor metric for reconstruction quality!

A motivation with 1-D point vortices:

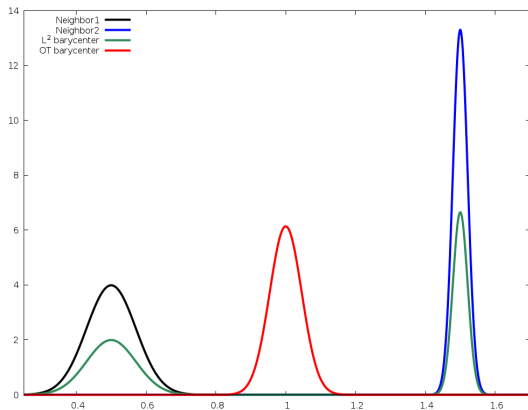


Large L^2 error, as soon as the two vortices are disjoint \rightarrow inconsistent.

Beyond L^2 ...

The Euclidean distance is a poor metric for reconstruction quality!

A motivation with 1-D point vortices:



Large L^2 error, as soon as the two vortices are disjoint \rightarrow inconsistent.

Euclidean distance is not suited for quantifying the quality of the reconstruction and interpolating.

\rightarrow **Optimal transport**

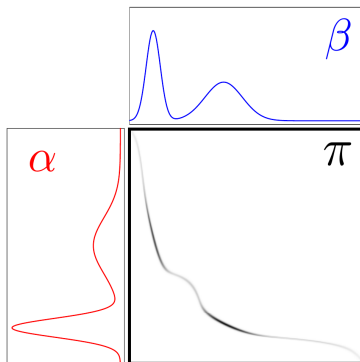
OT in a nutshell

Transport a measure α to another measure β minimizing some transport cost c .
One can define the cost c of moving an elementary amount of measure from \mathbf{x} to \mathbf{y} as a function of the Euclidean distance d : $c = d(\mathbf{x}, \mathbf{y})^p$.

The minimal total cost is a **distance**:

$$W_p^p(\alpha, \beta) := \min_{\pi} \int_{\Omega} d(\mathbf{x}, \mathbf{y})^p d\pi(\mathbf{x}, \mathbf{y}).$$

Very different geometry from standard information divergences (KL, Euclidean).



Peyré & Cuturi, 2020

Computationally involved but tractable thanks, *e.g.*, to entropic regularization
→ Sinkhorn algorithms.

Strategies for predicting

- Generative Adversarial Network → Not necessarily enough training examples, little control upon prediction manifold
- Wasserstein dictionary → Wrong subspace, blurry prediction
- Wasserstein barycenter → Computationally intensive, blurry prediction
- OT-based Diffusion Map → \in simplex of neighbors in the OT-sense, preserves divergence-free fields

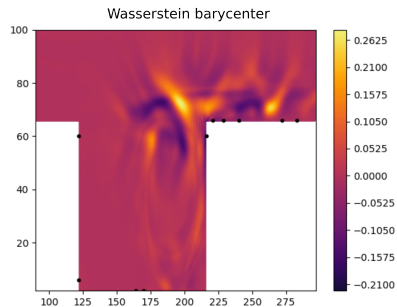
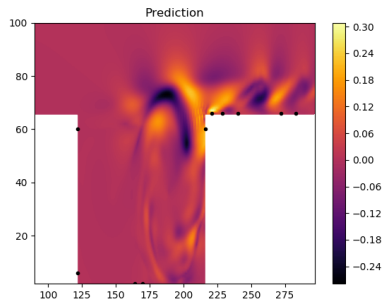
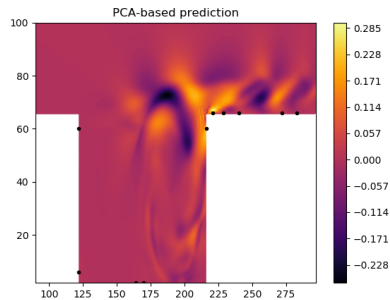
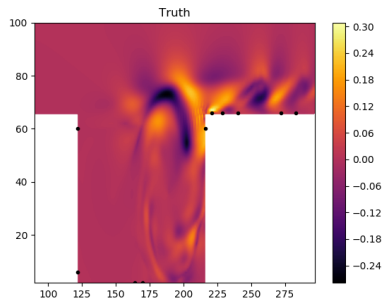
Strategies for predicting

- Generative Adversarial Network → Not necessarily enough training examples, little control upon prediction manifold
- Wasserstein dictionary → Wrong subspace, blurry prediction
- Wasserstein barycenter → Computationally intensive, blurry prediction
- OT-based Diffusion Map → \in simplex of neighbors in the OT-sense, preserves divergence-free fields

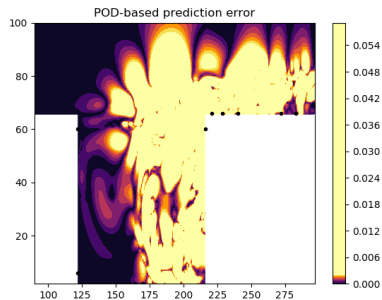
The feature kernel N_{DM} relies on (entropy-regularized) Wasserstein distances d_{Wass} estimated with Sinkhorn iterations:

$$N_{\text{DM}i,j} \sim \exp\left(-d_{\text{Wass}}^2(\mathbf{y}_i, \mathbf{y}_j)\right).$$

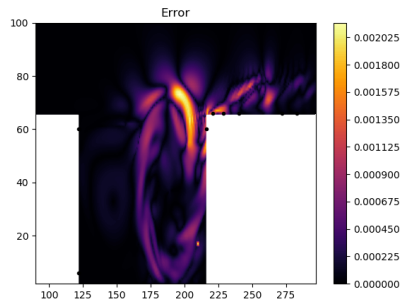
Reconstruction (fluctuating part)



Reconstruction error

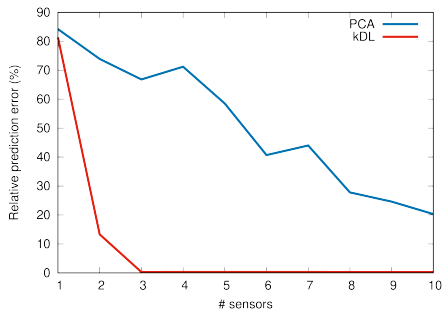
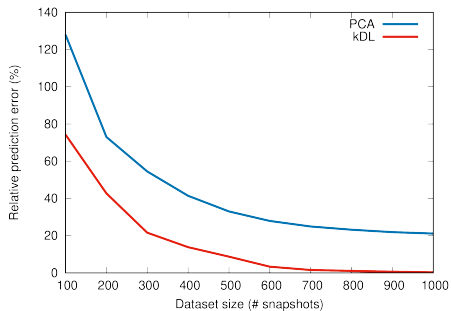


Regularized PCA: 22 % error



OT-DM: 0.1 % error

Required information



→ significantly outperforms PCA-based reconstruction.

Concluding remarks

- Offline/Online strategy for inference. First learn about the system at hand, then exploit,
- Need to enforce representation **accuracy** and **observability**,
- OT allows a consistent estimation strategy.

Future efforts:

- Assessment on more complex systems,
- Observability of the flow features to be enforced,
- Physically-consistent OT barycenters.

Wrapping-up...

Finally

$$\mathbf{s}_{\text{inst}} \longrightarrow \mathbf{s} \longrightarrow \Psi_{\mathbf{y}} \approx \tilde{\alpha} K_S(\mathbf{s}).$$

How to recover \mathbf{y} from $\Psi_{\mathbf{y}}$? \longrightarrow pre-image kernel problem

Wrapping-up...

Finally

$$\mathbf{s}_{\text{inst}} \longrightarrow \mathbf{s} \longrightarrow \Psi_{\mathbf{y}} \approx \tilde{\alpha} K_S(\mathbf{s}).$$

How to recover \mathbf{y} from $\Psi_{\mathbf{y}}$? \longrightarrow pre-image kernel problem

Approximation of the transition probabilities $p(\mathbf{y}, \mathbf{y}_i) \longrightarrow$ the distances $D(\Psi_{\mathbf{y}_i}, \Psi_{\mathbf{y}})$.

\longrightarrow easy to relate to distances $d(\mathbf{y}_i, \mathbf{y})$.

\longrightarrow can then estimate $\hat{\mathbf{y}} \approx \mathbf{y}$ from knowing its distance to each training samples $\{\mathbf{y}_i\}_i$.

At the end of the day:

$$\mathbf{s}_{\text{inst}} \longrightarrow \mathbf{s} \longrightarrow \Phi(\mathbf{s}) \longrightarrow \hat{\Psi}(\hat{\mathbf{y}}) \longrightarrow \hat{\mathbf{y}}$$

Embedding from 1 sensor

