Transport methods for Bayesian computation

Youssef Marzouk
joint work with Daniele Bigoni, Matthew Parno,
Alessio Spantini, & Olivier Zahm

Department of Aeronautics and Astronautics
Center for Computational Engineering
Statistics and Data Science Center

Massachusetts Institute of Technology
`http://uqgroup.mit.edu`

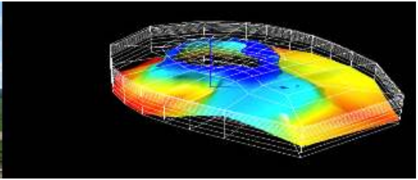
Support from AFOSR, DARPA, DOE

25–26 September 2019

Motivation: Bayesian inference in large-scale models

Observations y

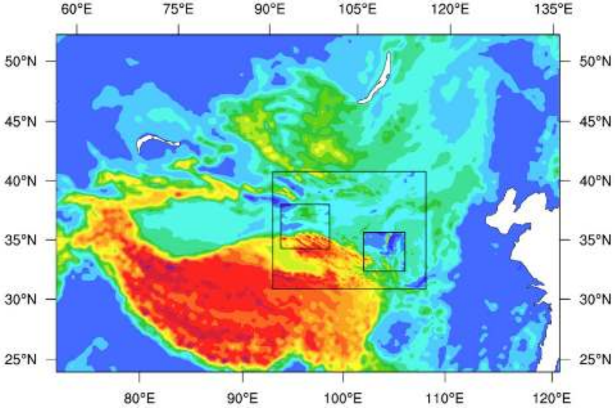
Parameters x



$$\pi_{\text{pos}}(x) := \underbrace{\pi(x|y) \propto \pi(y|x)\pi_{\text{pr}}(x)}_{\textbf{Bayes' rule}}$$

- Characterize the posterior distribution (density π_{pos})
- This is a challenging task since:
  - $x \in \mathbb{R}^n$ is typically **high-dimensional** (e.g., a discretized function)
  - π_{pos} is **non-Gaussian**
  - evaluations of the likelihood (hence π_{pos}) may be **expensive**
- π_{pos} can be evaluated up to a normalizing constant

Motivation: Sequential Bayesian inference



[image: NCAR]

- From batch to **sequential** approaches:
- State estimation (e.g., *filtering* and *smoothing*) in a Bayesian setting
  - Need **recursive** algorithms for characterizing the posterior

Plan for the lectures

Part 1 (Wednesday)

- Introduction to transport methods for inference and stochastic modeling
- Sparsity and decomposability of transport maps
- Bayesian inference in state-space models
- Dimension reduction in Bayesian inverse problems
- Low-rank structure in transport maps; greedy approximations

Part 2 (Thursday)

- Preconditioning MCMC using transport
- Nonlinear ensemble filtering methods
- Structure learning in non-Gaussian graphical models

Computational challenges of Bayesian inference

- Extract information from the posterior *(means, covariances, event probabilities, predictions)* by evaluating **posterior expectations:**

$$\mathbb{E}_{\pi_{\text{pos}}}[h(x)] = \int h(x)\pi_{\text{pos}}(x)dx$$

- Key strategy for making this computationally tractable:
  - Surrogates or approximations of the {forward model, likelihood function, posterior density}
  - Efficient and structure-exploiting **sampling schemes**

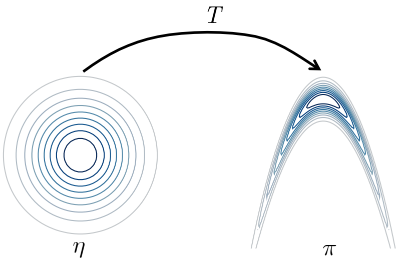
Computational challenges of Bayesian inference

- Extract information from the posterior *(means, covariances, event probabilities, predictions)* by evaluating **posterior expectations:**

$$\mathbb{E}_{\pi_{\text{pos}}}[h(x)] = \int h(x)\pi_{\text{pos}}(x)dx$$

- Key strategy for making this computationally tractable:
  - Surrogates or approximations of the {forward model, likelihood function, posterior density}
  - Efficient and structure-exploiting **sampling schemes**
- **These lectures: relate to notions of coupling and transport...**

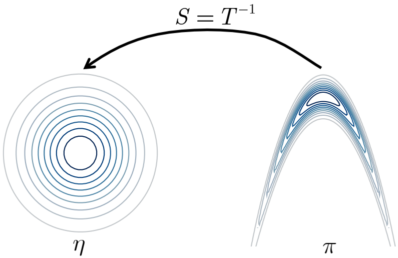
Deterministic couplings of probability measures



Core idea

- Choose a *reference distribution* η (e.g., standard Gaussian)
- Seek a transport map $T : \mathbb{R}^n \to \mathbb{R}^n$ such that $T_\sharp \eta = \pi$

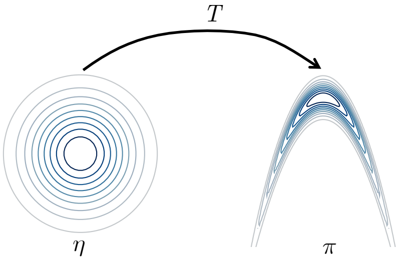
Deterministic couplings of probability measures



Core idea

- Choose a *reference distribution* η (e.g., standard Gaussian)
- Seek a transport map $T : \mathbb{R}^n \to \mathbb{R}^n$ such that $T_\sharp \eta = \pi$
- Equivalently, find $S = T^{-1}$ such that $S_\sharp \pi = \eta$

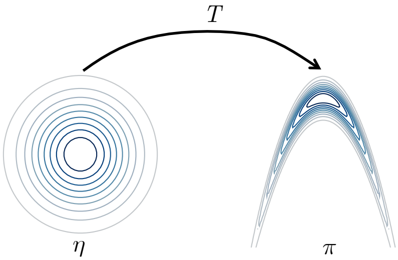
Deterministic couplings of probability measures



Core idea

- Choose a *reference distribution* η (e.g., standard Gaussian)
- Seek a transport map $T : \mathbb{R}^n \to \mathbb{R}^n$ such that $T_\sharp \eta = \pi$
- Equivalently, find $S = T^{-1}$ such that $S_\sharp \pi = \eta$
- In principle, enables *exact* (independent, unweighted) sampling!

Deterministic couplings of probability measures



Core idea

- Choose a *reference distribution* η (e.g., standard Gaussian)
- Seek a transport map $T : \mathbb{R}^n \to \mathbb{R}^n$ such that $T_\sharp \eta = \pi$
- Equivalently, find $S = T^{-1}$ such that $S_\sharp \pi = \eta$
- Satisfying these conditions only **approximately** can still be useful!

Choice of transport map

A useful building block is the **Knothe–Rosenblatt rearrangement:**

$$T(x) = \begin{bmatrix} T^1(x_1) \\ T^2(x_1, x_2) \\ \vdots \\ T^n(x_1, x_2, \ldots, x_n) \end{bmatrix}$$

- Unique triangular and monotone map satisfying $T_\sharp \eta = \pi$ for absolutely continuous η, π on $\mathbb{R}^n$
- Jacobian determinant easy to evaluate
- Monotonicity is essentially one-dimensional: $\partial_{x_k} T^k > 0$
- "Exposes" marginals, enables conditional sampling...

Choice of transport map

A useful building block is the **Knothe–Rosenblatt rearrangement:**

$$T(x) = \begin{bmatrix} T^1(x_1) \\ T^2(x_1, x_2) \\ \vdots \\ T^n(x_1, x_2, \dots, x_n) \end{bmatrix}$$

- Unique triangular and monotone map satisfying $T_\sharp \eta = \pi$ for absolutely continuous η, π on $\mathbb{R}^n$
- Jacobian determinant easy to evaluate
- Monotonicity is essentially one-dimensional: $\partial_{x_k} T^k > 0$
- "Exposes" marginals, enables conditional sampling...

- Numerical approximations can employ a *monotone parameterization* guaranteeing $\partial_{x_k} T^k > 0$. For example:

$$T^k(x_1, \dots, x_k) = a_k(x_1, \dots, x_{k-1}) + \int_0^{x_k} \exp(b_k(x_1, \dots, x_{k-1}, w))\, dw$$

How to construct triangular maps?

Construction #1: "maps from densities," i.e., *variational characterization* of the direct map T [Moselhy & M 2012]

How to construct triangular maps?

Construction #1: "maps from densities," i.e., *variational characterization* of the direct map T [Moselhy & M 2012]

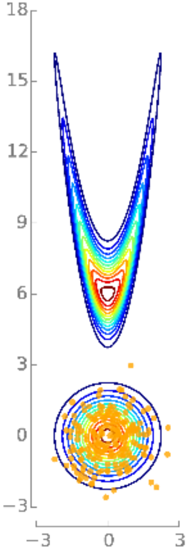
$$\min_{T \in \mathcal{T}_\triangle^h} \mathcal{D}_{KL}(\, T_\sharp \, \eta \,||\, \pi \,) = \min_{T \in \mathcal{T}_\triangle^h} \mathcal{D}_{KL}(\, \eta \,||\, T_\sharp^{-1} \, \pi \,)$$

- π is the "target" density on $\mathbb{R}^n$; η is, e.g., $\mathcal{N}(0, \mathbf{I}_n)$
- $\mathcal{T}_\triangle^h$ is a set of monotone lower triangular maps
  - $\mathcal{T}_\triangle^{h \to \infty}$ contains the *Knothe–Rosenblatt* rearrangement
- Expectation is with respect to the *reference* measure η
  - Compute via, e.g., Monte Carlo, sparse quadrature
- Use unnormalized evaluations of π and its gradients
- No MCMC or importance sampling
- In general non-convex, unless π is log-concave

Illustrative example

$$\min_T \mathbb{E}_\eta[-\log \pi \circ T - \sum_k \log \partial_{x_k} T^k]$$

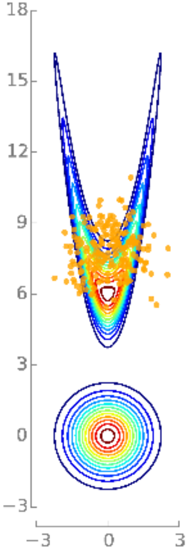
- Parameterized map $T \in \mathcal{T}_\triangle^h \subset \mathcal{T}_\triangle$
- Optimize over coefficients of parameterization
- Use gradient-based optimization
- The posterior is in the tail of the reference



Illustrative example

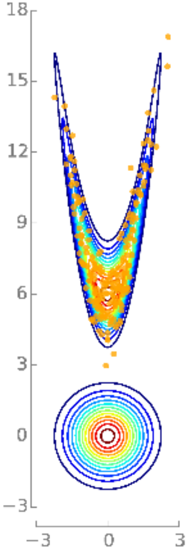
$$\min_T \mathbb{E}_\eta[-\log \pi \circ T - \sum_k \log \partial_{x_k} T^k]$$

- Parameterized map $T \in \mathcal{T}^h_\triangle \subset \mathcal{T}_\triangle$
- Optimize over coefficients of parameterization
- Use gradient-based optimization
- The posterior is in the tail of the reference



$$\min_{T} \mathbb{E}_{\eta}[-\log \pi \circ T - \sum_{k} \log \partial_{x_k} T^k]$$

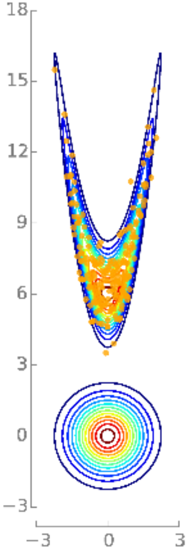
- Parameterized map $T \in \mathcal{T}_{\triangle}^{h} \subset \mathcal{T}_{\triangle}$
- Optimize over coefficients of parameterization
- Use gradient-based optimization
- The posterior is in the tail of the reference



Illustrative example

$$\min_T \mathbb{E}_\eta[-\log \pi \circ T - \sum_k \log \partial_{x_k} T^k]$$

- Parameterized map $T \in \mathcal{T}_\triangle^h \subset \mathcal{T}_\triangle$
- Optimize over coefficients of parameterization
- Use gradient-based optimization
- The posterior is in the tail of the reference



Useful features

- **Move** samples; don't just reweigh them
- *Independent* and *cheap* samples: $x_i \sim \eta \Rightarrow T(x_i)$
- Clear convergence criterion, even with unnormalized target density:

$$\mathcal{D}_{KL}(\, T_\sharp \, \eta \,||\, \pi \,) \approx \frac{1}{2} \, \mathbb{V}\mathrm{ar}_\eta \left[\log \frac{\eta}{T_\sharp^{-1} \bar{\pi}} \right]$$

Useful features

- **Move** samples; don't just reweigh them
- *Independent* and *cheap* samples: $x_i \sim \eta \Rightarrow T(x_i)$
- Clear convergence criterion, even with unnormalized target density:

$$\mathcal{D}_{KL}(\, T_\sharp \, \eta \,||\, \pi \,) \approx \frac{1}{2} \, \mathbb{V}\mathrm{ar}_\eta \left[\log \frac{\eta}{T_\sharp^{-1} \bar{\pi}} \right]$$

- Can either accept bias or reduce it by:
  - Increasing the complexity of the map $T \in \mathcal{T}_\triangle^h$
  - Sampling the pullback $T_\sharp^{-1} \pi$ using MCMC or importance sampling *(more on this later)*

Useful features

- **Move** samples; don't just reweigh them
- *Independent* and *cheap* samples: $x_i \sim \eta \Rightarrow T(x_i)$
- Clear convergence criterion, even with unnormalized target density:

$$\mathcal{D}_{KL}(\, T_\sharp\, \eta \,||\, \pi\,) \approx \frac{1}{2}\, \mathbb{V}\mathrm{ar}_\eta \left[\log \frac{\eta}{T_\sharp^{-1} \bar{\pi}} \right]$$

- Can either accept bias or reduce it by:
  - Increasing the complexity of the map $T \in \mathcal{T}_\triangle^h$
  - Sampling the pullback $T_\sharp^{-1}\pi$ using MCMC or importance sampling *(more on this later)*

- Related transport constructions for inference and sampling: Stein variational gradient descent [Liu & Wang 2016, DeTommaso 2018], normalizing flows [Rezende & Mohamed 2015], SOS polynomial flow [Jaini *et al.* 2019], Gibbs flow [Heng *et al.* 2015], particle flow filter [Reich 2011], implicit sampling [Chorin *et al.* 2009–2015], etc.

Ubiquity of triangular maps

Many "flows" recently proposed in machine learning are special cases of triangular maps:

- NICE: Nonlinear independent component estimation [Dinh et al. 2015]
$$T^k(x_1, \ldots, x_k) = \mu_k(x_{1:k-1}) + x_k$$
- Inverse autoregressive flow [Dinh et al. 2017]
$$T^k(x_1, \ldots, x_k) = (1 - \sigma_k(x_{1:k-1}))\mu_k(x_{1:k-1}) + x_k \sigma_k(x_{1:k-1})$$
- Masked autogressive flow [Papamakarios et al. 2017]
$$T^k(x_1, \ldots, x_k) = \mu_k(x_{1:k-1}) + x_k \exp(\alpha_k(x_{1:k-1}))$$
- Neural autoregressive flow [Huang et al. 2018
$$T^k(x_1, \ldots, x_k) = \mathrm{DNN}(x_k; w_k(x_{1:k-1}))$$
- Sum-of-squares polynomial flow [Jaini et al. 2019]

How to construct triangular maps?

Construction #2: "maps from samples"

$$\min_{S \in \mathcal{S}^h_\triangle} \mathcal{D}_{KL}(\, S_\sharp \pi \,||\, \eta \,) = \min_{S \in \mathcal{S}^h_\triangle} \mathcal{D}_{KL}(\, \pi \,||\, S_\sharp^{-1} \eta \,)$$

- Suppose we have Monte Carlo samples $\{x_i\}_{i=1}^M \sim \pi$
- For standard Gaussian η, this problem is **convex** and **separable**
- This is *density estimation via transport!* (cf. Tabak & Turner 2013)

How to construct triangular maps?

Construction #2: "maps from samples"

$$\min_{S \in \mathcal{S}_\triangle^h} \mathcal{D}_{KL}(\, S_\sharp \pi \,||\, \eta \,) = \min_{S \in \mathcal{S}_\triangle^h} \mathcal{D}_{KL}(\, \pi \,||\, S_\sharp^{-1} \eta \,)$$

- Suppose we have Monte Carlo samples $\{x_i\}_{i=1}^M \sim \pi$
- For standard Gaussian η, this problem is **convex** and **separable**
- This is *density estimation via transport!* (cf. Tabak & Turner 2013)
- Equivalent to maximum likelihood estimation of S

$$\widehat{S} \in \arg \max_{S \in \mathcal{S}_\triangle^h} \frac{1}{M} \sum_{i=1}^M \log \underbrace{S_\sharp^{-1} \eta}_{\text{pullback}}(x_i), \qquad \eta = \mathcal{N}(0, \mathbf{I}_n),$$

- Each component $\widehat{S}^k$ of $\widehat{S}$ can be computed *separately*, via smooth *convex optimization*

$$\widehat{S}^k \in \arg \min_{S^k \in \mathcal{S}_{\triangle,k}^h} \frac{1}{M} \sum_{i=1}^M \left(\frac{1}{2} S^k(x_i)^2 - \log \partial_k S^k(x_i) \right)$$

Underlying challenge: maps in high dimensions

- Major bottleneck: representation of the map, e.g., cardinality of the map basis
- How to make the construction/representation of high-dimensional transports tractable?

Low-dimensional structure of transport maps

Underlying challenge: maps in high dimensions

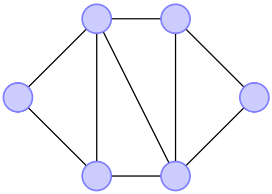
- Major bottleneck: representation of the map, e.g., cardinality of the map basis
- How to make the construction/representation of high-dimensional transports tractable?

Main ideas:

1. Exploit **Markov structure** of the target distribution
   - Leads to sparsity and/or decomposability of transport maps [Spantini, Bigoni, & M JMLR 2018]
2. Exploit certain **low rank** structure
   - Near-identity or "lazy" maps [Bigoni et al. arXiv:1906.00031]

Markov random fields

- Let $Z_1, \ldots, Z_n$ be random variables with joint density $\pi > 0$



$$(i,j) \notin \mathcal{E} \quad \text{iff} \quad Z_i \perp\!\!\!\perp Z_j \,|\, \mathbf{Z}_{\mathcal{V}\setminus\{i,j\}}$$

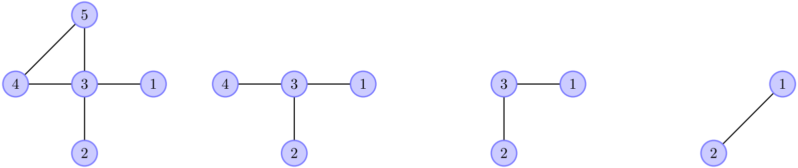
- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ encodes **conditional independence** (an I-map for π)
- **Theorem** [SBM 2018]: Define $\mathcal{G}$ s.t. $(i,j) \notin \mathcal{E}$ iff $\partial_{x_i,x_j} \log \pi = 0$. Then the resulting $\mathcal{G}$ is the unique minimal I-map for π.

Sparsity of transport maps

- Focus on the *inverse* triangular map S, where $S_\sharp \pi = \eta$
- **Theorem** [SBM 2018]: S (a *nonlinear function*) inherits the same sparsity pattern as the Cholesky factor of the incidence matrix (properly scaled) of a graphical model for π, provided that $\eta(\mathbf{x}) = \prod_i \eta(x_i)$

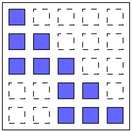
$$S(\mathbf{x}) = \left[\begin{array}{l} S^1(x_1) \\ S^2(x_1, x_2) \\ S^3(x_1, x_2, x_3) \\ \vdots \\ S^n(x_1, x_2, \ldots, x_n) \end{array} \right] \Longrightarrow \left[\begin{array}{l} S^1(x_1) \\ S^2(x_1, x_2) \\ S^3(\quad x_2, x_3) \\ \vdots \\ S^n(\qquad\qquad x_{n-1}, x_n) \end{array} \right]$$

How to compute the sparsity pattern



$\mathcal{G}^5$ $\mathcal{G}^4$ $\mathcal{G}^3$ $\mathcal{G}^2$

- **Compute marginal graphs:** $\mathcal{G}^{i-1}$ is obtained from $\mathcal{G}^i$ by removing node i and by turning its neighborhood into a clique (like *variable elimination*)
- **Sparsity of inverse transport:** the i-th component of S can depend, at most, on the variables in a neighborhood of node i in $\mathcal{G}^i$
- Sparsity depends on the ordering of the variables (similar heuristics as *sparse Cholesky*)



$\mathbf{P}_{kj} = \partial_{x_j} S^k$

Decomposable transport maps

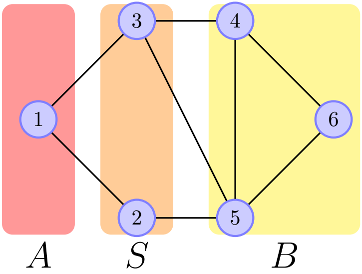
- **Definition:** a decomposable transport is a map $T = T_1 \circ \cdots \circ T_k$ that factorizes as the composition of finitely many maps of low effective dimension that are triangular (up to a permutation), e.g.,

$$T(\mathbf{x}) = \underbrace{\begin{bmatrix} A_1(x_1, x_2, x_3) \\ B_1(x_2, x_3) \\ C_1(x_3) \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}}_{T_1} \circ \underbrace{\begin{bmatrix} x_1 \\ A_2(x_2, x_3, x_4, x_5) \\ B_2(x_3, x_4, x_5) \\ C_2(x_4, x_5) \\ D_2(x_5) \\ x_6 \end{bmatrix}}_{T_2} \circ \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ A_3(x_4) \\ B_3(x_4, x_5) \\ C_3(x_4, x_5, x_6) \end{bmatrix}}_{T_3}$$

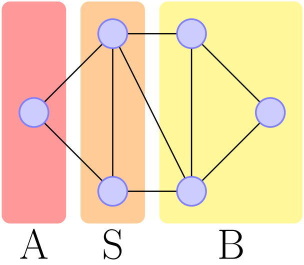
- **Theorem** [SBM 2018]: Decomposable graphical models for π lead to decomposable direct maps T, provided that $\eta(\mathbf{x}) = \prod_i \eta(x_i)$

Decomposable transport maps

- Example graph decomposition $\mathcal{V} = (\mathcal{A}, \mathcal{S}, \mathcal{B})$
- Effective dimension of each component map is $|\mathcal{A} \cup \mathcal{S}|$



Graph decomposition

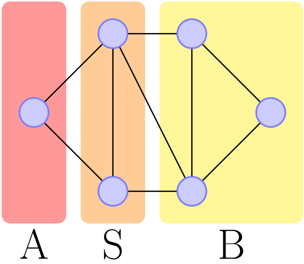


Definition

A triple (A, S, B) of disjoint nonempty subsets of the vertex set $\mathcal{V}$ forms a **decomposition** of $\mathcal{G}$ if the following hold

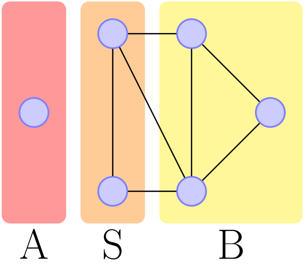
1. $\mathcal{V} = A \cup S \cup B$
2. S separates A from B in $\mathcal{G}$

Step 1: build a local map



- For a given decomposition (A, S, B), consider $\mathfrak{M}_1 : \mathbb{R}^3 \to \mathbb{R}^3$ s.t.
  1. $\mathfrak{M}_1(\mathbf{x}_A, \mathbf{x}_S) = \begin{bmatrix} A_1(\mathbf{x}_S, \mathbf{x}_A) \\ B_1(\mathbf{x}_S) \end{bmatrix}$ pushes forward η_3 to marginal $\pi_{\mathbf{x}_{S \cup A}}$
  2. Embed $\mathfrak{M}_1$ in $T_1(\mathbf{x}_A, \mathbf{x}_S, \mathbf{x}_B) = \begin{bmatrix} A_1(\mathbf{x}_S, \mathbf{x}_A) \\ B_1(\mathbf{x}_S) \\ \mathbf{x}_B \end{bmatrix}$, $T_1 : \mathbb{R}^6 \to \mathbb{R}^6$
- What can we say about the pullback density $T_1^\sharp \pi$?

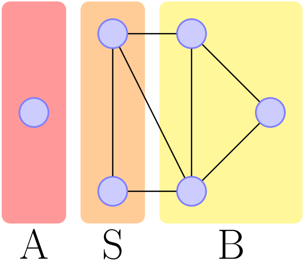
Local graph sparsification



$$\boxed{T = T_1}$$

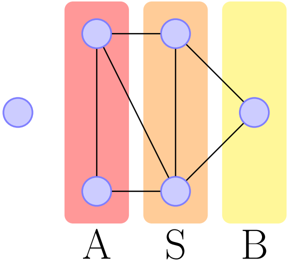
- **Figure:** Markov structure of the pullback of π through T
- Just remove any edge incident to any node in A
- T_1 is essentially a 3-D map
- Pulling back π through T_1 makes $\mathbf{Z}_A$ independent of $\mathbf{Z}_{S \cup B}$!

Do it recursively!



$$\boxed{T = T_1}$$

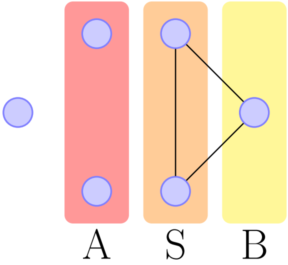
- **Figure:** Markov structure of the pullback of π through T
- **Recursion** at step k
  1. Consider a new decomposition (A, S, B)
  2. Compute transport T_k
  3. Pull back through T_k



$$\boxed{T = T_1}$$

- **Figure:** Markov structure of the pullback of π through T
- **Recursion** at step k
  1. Consider a new decomposition (A, S, B)
  2. Compute transport T_k
  3. Pull back through T_k

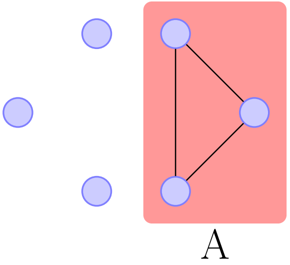
Step k: local graph sparsification



$$\boxed{T = T_1 \circ T_2}$$

- **Figure:** Markov structure of the pullback of π through T
- T_2 is essentially a 4-D map
- Each time we pull back by a new map we remove edges
- **Intuition:** Continue the recursion until no edges are left...

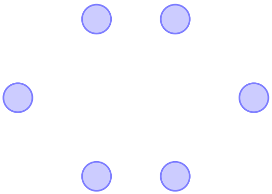
And so on...



$$\boxed{T = T_1 \circ T_2}$$

- **Figure:** Markov structure of the pullback of π through T
- T_2 is essentially a 4-D map
- Each time we pullback by a new map we remove edges
- **Intuition.** Continue the recursion until no edges are left...

Decomposable maps



$$\boxed{T = T_1 \circ T_2 \circ T_3}$$

- **Figure:** Markov structure of the pullback of π through T
- Decomposability of $\mathcal{G} \Rightarrow$ existence of **decomposable** couplings
- Anisotropic triangular structure of (T_i) is essential
- Idea: inference decomposed into smaller steps (no need for marginals!)
- In fact, we can make this more general. . .

Decomposition theorem

Theorem [Decomposition of transports]

Let $\mathcal{G}$ be an I-map for π and let $\eta = \prod_j \eta_{X_j}$ be a reference density.
If (A, S, B) is a decomposition of $\mathcal{G}$, then

1. $\exists$ a transport map:

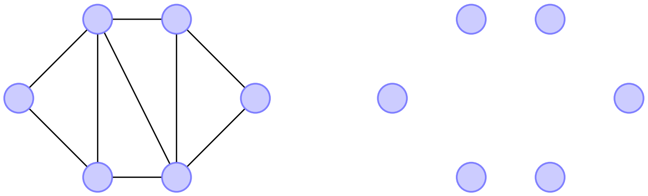
$$\boxed{T = T_1 \circ T_2}$$

   - T_1 is a monotone triangular transport s.t. $\eta \xrightarrow{T_1} \pi_{X_{A \cup S}} \cdot (\prod_{j \in B} \eta_{X_j})$
   - T_1 is the identity map along components in B: $T_1^k(\mathbf{x}) = x_k$ for $k \in B$
   - T_2 is **any** transport s.t. $\eta \xrightarrow{T_2} T_1^\sharp \pi$

2. $\mathbf{X}_A$ is independent of $\mathbf{X}_{S \cup B}$ w.r.t. the pullback density $T_1^\sharp \pi$
   - T_2 is the identity along components in A: $T_2^k(\mathbf{x}) = x_k$ for $k \in A$

- **Strategy**: recursively apply theorem to further decompose T_2

Graph decomposition (end result)



- *(right)* I-map for the pullback of π through T

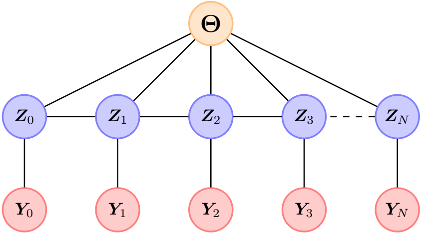
$$T(\mathbf{x}) = \underbrace{\begin{bmatrix} A_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \\ B_1(\mathbf{x}_2, \mathbf{x}_3) \\ C_1(\mathbf{x}_3) \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \mathbf{x}_6 \end{bmatrix}}_{T_1} \circ \underbrace{\begin{bmatrix} \mathbf{x}_1 \\ A_2(\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5) \\ B_2(\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5) \\ C_2(\mathbf{x}_4, \mathbf{x}_5) \\ D_2(\mathbf{x}_5) \\ \mathbf{x}_6 \end{bmatrix}}_{T_2} \circ \underbrace{\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ A_3(\mathbf{x}_4) \\ B_3(\mathbf{x}_4, \mathbf{x}_5) \\ C_3(\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6) \end{bmatrix}}_{T_3}$$

Transport maps and graphical models

Key message

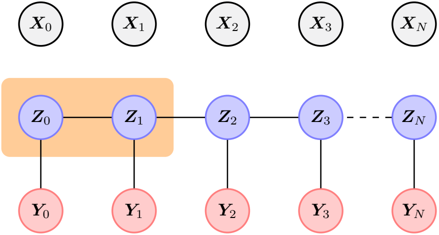
- Direct maps: **enforce decomposable structure** in the approximation space $\mathcal{T}_\triangle$, i.e., when solving $\min_{T \in \mathcal{T}_\triangle} \mathcal{D}_{KL}(\, T_\sharp \eta \,||\, \pi \,)$
- Inverse maps: **enforce sparsity** in the approximation space $\mathcal{S}_\triangle$, i.e., in solving $\min_{S \in \mathcal{S}_\triangle} \mathcal{D}_{KL}(\, \pi \,||\, S_\sharp^{-1} \, \eta \,)$
  - Can also use for *structure learning* in non-Gaussian graphical models
- A general tool for modeling and computation with *non-Gaussian Markov random fields*

Transport maps and state-space models



- In many situations, elements of the composition $T = T_1 \circ T_2 \circ \cdots \circ T_k$ can be constructed **sequentially**
- Yields new algorithms for smoothing and and joint state-parameter inference in state-space models [SBM 2018; Houssineau, Jasra, Singh 2018]

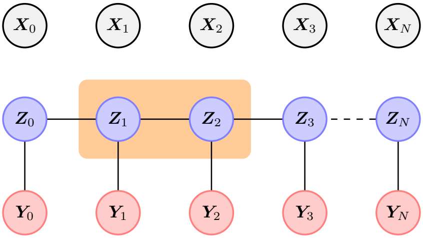
Application to state-space models (chain graph)



- Compute $\mathfrak{M}_0 : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ s.t.
$$\mathfrak{M}_0(\mathbf{x}_0, \mathbf{x}_1) = \begin{bmatrix} A_0(\mathbf{x}_0, \mathbf{x}_1) \\ B_0(\mathbf{x}_1) \end{bmatrix}$$
- Reference: $\eta_{\mathbf{X}_0} \eta_{\mathbf{X}_1}$
- Target: $\pi_{\mathbf{Z}_0} \, \pi_{\mathbf{Z}_1|\mathbf{Z}_0} \, \pi_{\mathbf{Y}_0|\mathbf{Z}_0} \, \pi_{\mathbf{Y}_1|\mathbf{Z}_1}$
- $\dim(\mathfrak{M}_0) \simeq 2 \times \dim(\mathbf{Z}_0)$

$$T_0(\mathbf{x}) = \begin{bmatrix} A_0(\mathbf{x}_0, \mathbf{x}_1) \\ B_0(\mathbf{x}_1) \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$

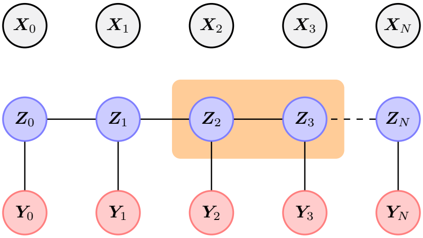
Second step: compute another 2-D map



- Compute $\mathfrak{M}_1 : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ s.t.
$$\mathfrak{M}_1(\mathbf{x}_1, \mathbf{x}_2) = \begin{bmatrix} A_1(\mathbf{x}_1, \mathbf{x}_2) \\ B_1(\mathbf{x}_2) \end{bmatrix}$$
- Reference: $\eta_{\mathbf{X}_1} \eta_{\mathbf{X}_2}$
- Target: $\eta_{\mathbf{X}_1}\, \pi_{\mathbf{Y}_2|\mathbf{Z}_2}\, \pi_{\mathbf{Z}_2|\mathbf{Z}_1}(\cdot |\, B_0\,(\cdot)\,)$
- Uses only one component of $\mathfrak{M}_0$

$$T_1(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_0 \\ A_1(\mathbf{x}_1, \mathbf{x}_2) \\ B_1(\mathbf{x}_2) \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$

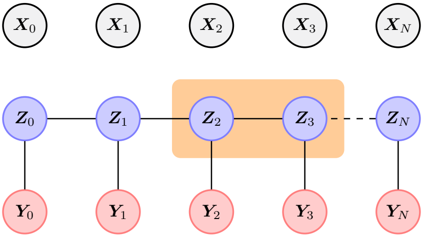
Proceed recursively forward in time



- Compute $\mathfrak{M}_2 : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ s.t.
$$\mathfrak{M}_2(\mathbf{x}_2, \mathbf{x}_3) = \begin{bmatrix} A_2(\mathbf{x}_2, \mathbf{x}_3) \\ B_2(\mathbf{x}_3) \end{bmatrix}$$
- Reference: $\eta_{\mathbf{X}_2}\eta_{\mathbf{X}_3}$
- Target: $\eta_{\mathbf{X}_2}\, \pi_{\mathbf{Y}_3|\mathbf{Z}_3}\, \pi_{\mathbf{Z}_3|\mathbf{Z}_2}(\cdot|\, B_1\,(\cdot)\,)$
- Uses only one component of $\mathfrak{M}_1$

$$T_2(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ A_2(\mathbf{x}_2, \mathbf{x}_3) \\ B_2(\mathbf{x}_3) \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$

A decomposition theorem for chains



Theorem.

1. $(B_k)_\sharp \, \eta_{\mathsf{X}_{k+1}} \;=\; \pi_{\mathsf{Z}_{k+1}\,|\,\mathsf{Y}_{0:k+1}}$ (*filtering*)
2. $(\mathfrak{M}_k)_\sharp \, \eta_{\mathsf{X}_{k:k+1}} \;\simeq\; \pi_{\mathsf{Z}_k,\mathsf{Z}_{k+1}\,|\,\mathsf{Y}_{0:k+1}}$ (*lag-1 smoothing*)
3. $(T_1 \circ \cdots \circ T_k)_\sharp \, \eta_{\mathsf{X}_{0:k+1}} \;=\; \pi_{\mathsf{Z}_{0:k+1}|\,\mathsf{Y}_{0:k+1}}$ (*full Bayesian solution)*

A nested decomposable map

- $\mathfrak{T}_k = T_0 \circ T_1 \circ \cdots \circ T_k$ characterizes the joint dist $\pi_{\mathbf{Z}_{0:k+1} | \mathbf{Y}_{0:k+1}}$

$$\mathfrak{T}_k \quad (\mathbf{x}) = \underbrace{\begin{bmatrix} A_0(\mathbf{x}_0, \mathbf{x}_1) \\ B_0(\mathbf{x}_1) \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_0} \circ \underbrace{\begin{bmatrix} \mathbf{x}_0 \\ A_1(\mathbf{x}_1, \mathbf{x}_2) \\ B_1(\mathbf{x}_2) \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_1} \circ$$

- Trivial to go from $\mathfrak{T}_k$ to $\mathfrak{T}_{k+1}$: just append a new map T_{k+1}
- No need to recompute $T_0, \ldots, T_k$ (nested transports)
- $\mathfrak{T}_k$ is dense and high-dimensional but **decomposable**

A nested decomposable map

- $\mathfrak{T}_k = T_0 \circ T_1 \circ \cdots \circ T_k$ characterizes the joint dist $\pi_{\mathbf{Z}_{0:k+1} | \mathbf{Y}_{0:k+1}}$

$$\mathfrak{T}_{k+1}(\mathbf{x}) = \underbrace{\begin{bmatrix} A_0(\mathbf{x}_0, \mathbf{x}_1) \\ B_0(\mathbf{x}_1) \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_0} \circ \underbrace{\begin{bmatrix} \mathbf{x}_0 \\ A_1(\mathbf{x}_1, \mathbf{x}_2) \\ B_1(\mathbf{x}_2) \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_1} \circ \underbrace{\begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ A_2(\mathbf{x}_2, \mathbf{x}_3) \\ B_2(\mathbf{x}_3) \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_2} \circ \cdots$$

- Trivial to go from $\mathfrak{T}_k$ to $\mathfrak{T}_{k+1}$: just append a new map T_{k+1}
- No need to recompute $T_0, \dots, T_k$ (nested transports)
- $\mathfrak{T}_k$ is dense and high-dimensional but **decomposable**

Some intuition for smoothing

- Each lag-one smoothing map implements a factorization:

$$\pi_{\mathbf{Z}_k,\mathbf{Z}_{k-1}|\mathbf{y}_{0:k}} = \pi_{\mathbf{Z}_k|\mathbf{y}_{0:k}}\,\pi_{\mathbf{Z}_{k-1}|\mathbf{Z}_k,\mathbf{y}_{0:k}} = \pi_{\mathbf{Z}_k|\mathbf{y}_{0:k}}\,\pi_{\mathbf{Z}_{k-1}|\mathbf{Z}_k,\mathbf{y}_{0:k-1}}$$

- The composition of maps then implements the following factorization:

$$\begin{aligned}\pi_{\mathbf{Z}_{0:N}|\mathbf{y}_{0:N}} &= \pi_{\mathbf{Z}_N|\mathbf{y}_{0:N}}\pi_{\mathbf{Z}_{N-1}|\mathbf{Z}_N,\mathbf{y}_{0:N-1}}\pi_{\mathbf{Z}_{N-2}|\mathbf{Z}_{N-1},\mathbf{y}_{0:N-2}} \\ &\quad \cdots \pi_{\mathbf{Z}_1|\mathbf{Z}_2,\mathbf{y}_{0:1}}\pi_{\mathbf{Z}_0|\mathbf{Z}_1,\mathbf{y}_0}\end{aligned}$$

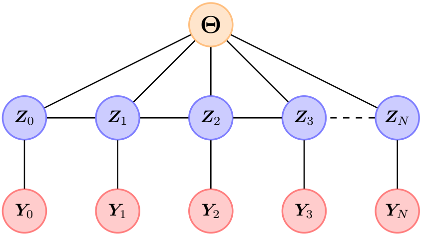
A single-pass algorithm on the model

- **Meta-algorithm:**
  1. Compute the maps $\mathfrak{M}_0, \mathfrak{M}_1, \ldots$, each of dimension $2 \times \dim(\mathbf{Z}_0)$
  2. Embed each $\mathfrak{M}_j$ into an identity map to form T_j
  3. Evaluate $T_0 \circ \cdots \circ T_k$ for the full Bayesian solution
- **Remarks**:
  - A single pass on the state-space model
  - **Non-Gaussian** generalization of the Rauch-Tung-Striebel smoother
  - Bias is *only* due to the numerical approximation of each map $\mathfrak{M}_i$
  - Can either accept the bias or reduce it by:
    - Increasing the complexity of each map $\mathfrak{M}_i$, or
    - Computing **weights** given by the proposal density

$$(T_0 \circ T_1 \circ \cdots \circ T_k)_\sharp \, \eta_{\mathsf{X}_{0:k+1}}$$

Joint parameter/state estimation

- Generalize to sequential **joint parameter/state estimation**

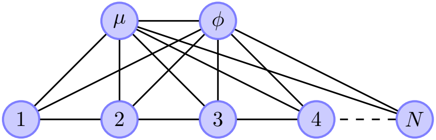


- $(T_0 \circ \cdots \circ T_k)_\sharp \, \eta_\Theta \, \eta_{\mathbf{X}_{0:k+1}} = \pi_{\Theta, \mathbf{Z}_{0:k+1} \,|\, \mathbf{Y}_{0:k+1}}$ *(full Bayesian solution)*
- Now $\dim(\mathfrak{M}_j) = 2 \times \dim(\mathbf{Z}_j) + \dim(\Theta)$
- **Remarks**:
  - No artificial dynamic for the static parameters
  - No a priori fixed-lag smoothing approximation

Example: stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = \textbf{Id}$$

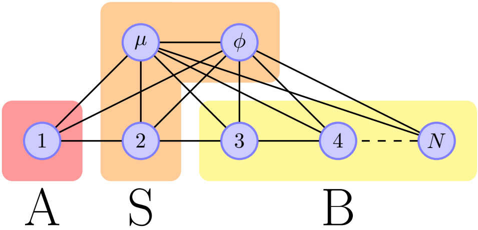


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Start with the identity map

Stochastic volatility model

- Build the decomposition recursively

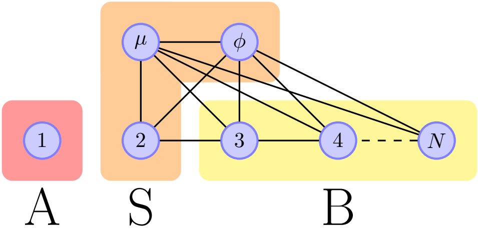
$$\mathfrak{T} = \textbf{Id}$$



- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Find a good first decomposition of $\mathcal{G}$

- Build the decomposition recursively

$$\mathfrak{T} = T_0$$

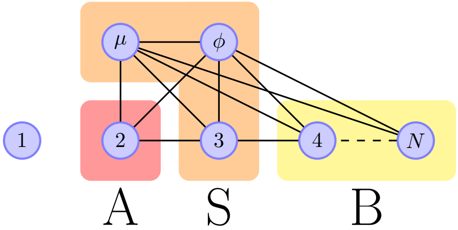


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Compute an (essentially) 4-D T_0 and pull back π
- Underlying approximation of $\mu, \phi, \mathbf{Z}_1 | \mathbf{Y}_1$

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = T_0$$

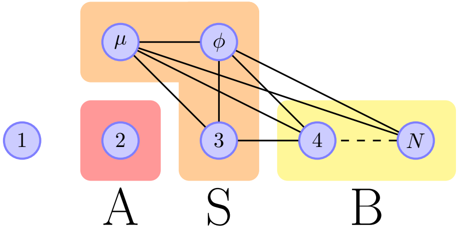


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Find a new decomposition
- Underlying approximation of $\mu, \phi, \mathbf{Z}_1 | \mathbf{Y}_1$

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = T_0 \circ T_1$$

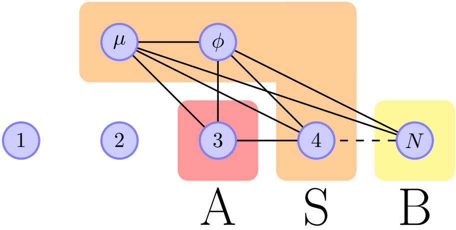


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Compute an (essentially) 4-D T_1 and pull back π
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:2} | \mathbf{Y}_{1:2}$

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = T_0 \circ T_1$$

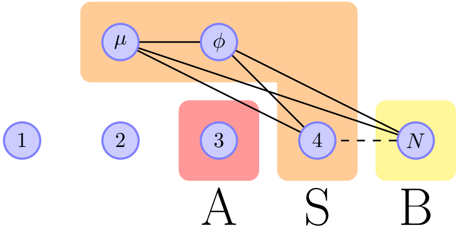


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Continue the recursion until no edges are left...
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:2} | \mathbf{Y}_{1:2}$

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = T_0 \circ T_1 \circ T_2$$

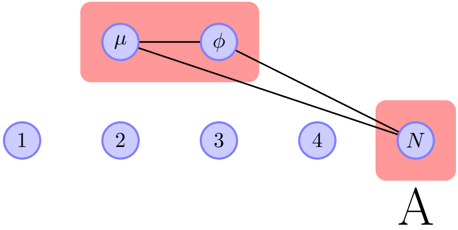


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Continue the recursion until no edges are left...
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:3}|\mathbf{Y}_{1:3}$

Stochastic volatility model

- Build the decomposition recursively

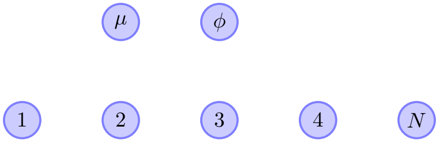
$$\mathfrak{T} = T_0 \circ T_1 \circ T_2 \circ \cdots \circ T_{N-3}$$



- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Continue the recursion until no edges are left. . .
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:N-1} | \mathbf{Y}_{1:N-1}$

Stochastic volatility model

- Build the decomposition recursively
$$\mathfrak{T} = T_0 \circ T_1 \circ T_2 \circ \cdots \circ T_{N-3} \circ T_{N-2}$$



- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Each map T_k is essentially 4-D regardless of N
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:N} | \mathbf{Y}_{1:N}$

Another decomposable map

$$\mathfrak{T}_{k+1}(\mathbf{x}) = \underbrace{\begin{bmatrix} P_0(x_\theta) \\ A_0(\mathbf{x}_\theta, \mathbf{x}_0, \mathbf{x}_1) \\ B_0(\mathbf{x}_\theta, \mathbf{x}_1) \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_0} \circ \underbrace{\begin{bmatrix} P_1(x_\theta) \\ \mathbf{x}_0 \\ A_1(\mathbf{x}_\theta, \mathbf{x}_1, \mathbf{x}_2) \\ B_1(\mathbf{x}_\theta, \mathbf{x}_2) \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_1} \circ \underbrace{\begin{bmatrix} P_2(x_\theta) \\ \mathbf{x}_0 \\ \mathbf{x}_1 \\ A_2(\mathbf{x}_\theta, \mathbf{x}_2, \mathbf{x}_3) \\ B_2(\mathbf{x}_\theta, \mathbf{x}_3) \\ \mathbf{x}_4 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_2} \circ \cdots$$

- $(P_0 \circ \cdots \circ P_k)_\sharp \, \eta_\Theta = \pi_{\Theta \,|\, \mathbf{Y}_{0:k+1}}$ *(parameter inference)*
- If $\mathfrak{P}_k = P_0 \circ \cdots \circ P_k$, then $\mathfrak{P}_k$ can be computed recursively as

$$\mathfrak{P}_k = \mathfrak{P}_{k-1} \circ P_k$$

$\Longrightarrow$ cost of evaluating $\mathfrak{P}_k$ does not grow with k

Example: stochastic volatility model

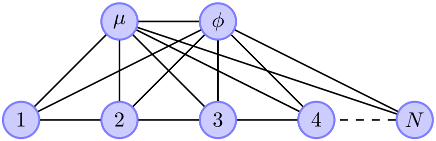
- **Stochastic volatility model:** Latent log-volatilities take the form of an AR(1) process for $t = 1, \ldots, N$:

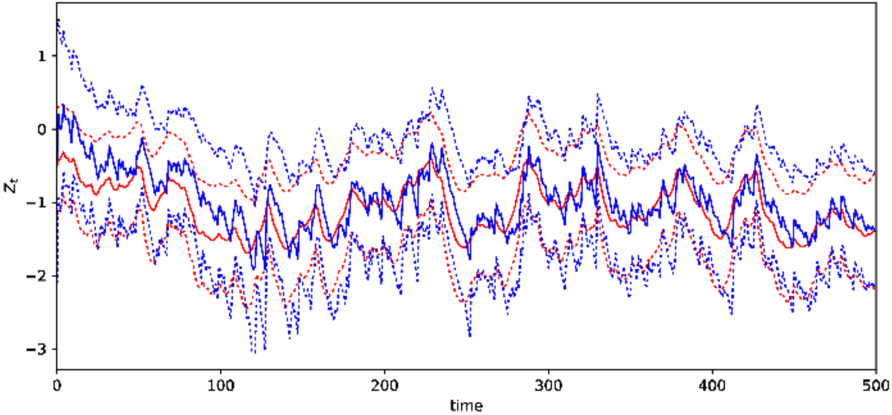
$$Z_{t+1} = \mu + \phi(Z_t - \mu) + \eta_t, \quad \eta_t \sim \mathcal{N}(0,1), \quad Z_1 \sim \mathcal{N}(0, 1/1-\phi^2)$$

  - Observe the mean return for holding an asset at time t

$$Y_t = \varepsilon_t \exp(0.5\, Z_t), \quad \varepsilon_t \sim \mathcal{N}(0,1), \quad t = 1, \ldots, N$$

  - Markov structure for $\pi \sim \mu, \phi, \mathbf{Z}_{1:N} | \mathbf{Y}_{1:N}$ is given by:

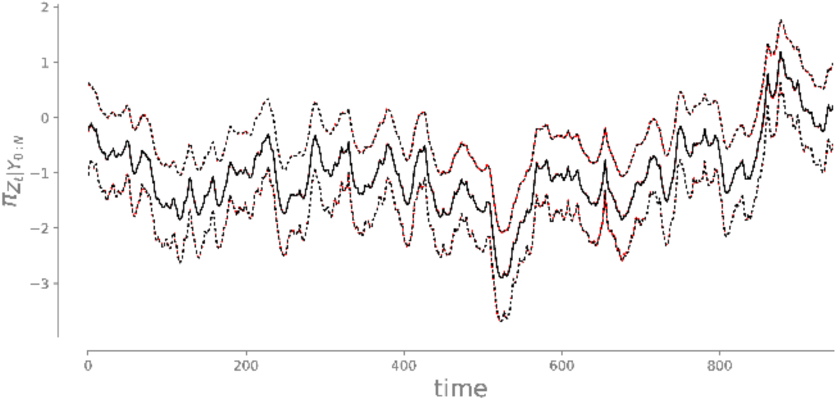




- Infer log-volatility of the pound/dollar exchange rate, starting on 1 October 1981
- Filtering (blue) versus smoothing (red) marginals

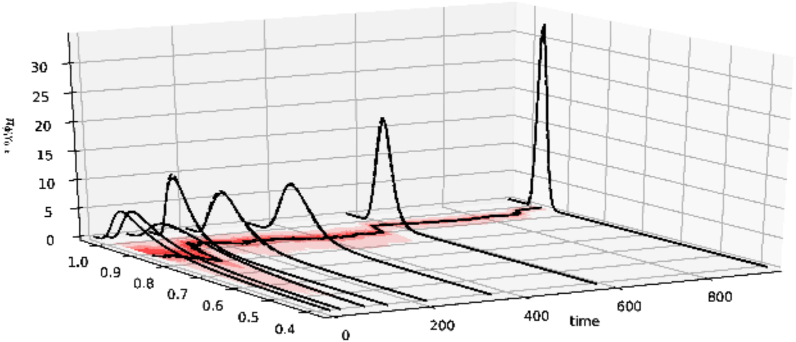
Smoothing marginals

- Just **re-evaluate** the 4-D maps backwards in time
- Comparison with a "reference" MCMC solution with 10^5 ESS (in red)

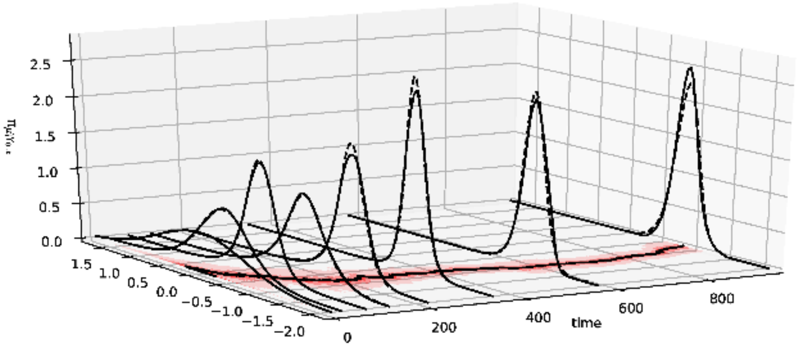


Static parameter ϕ

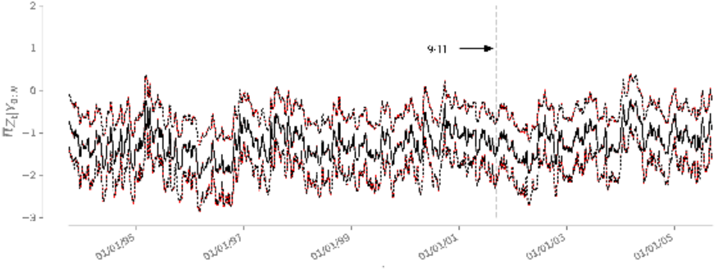
- **Sequential** parameter inference
- Comparison with a "reference" MCMC solution (**batch** algorithm)

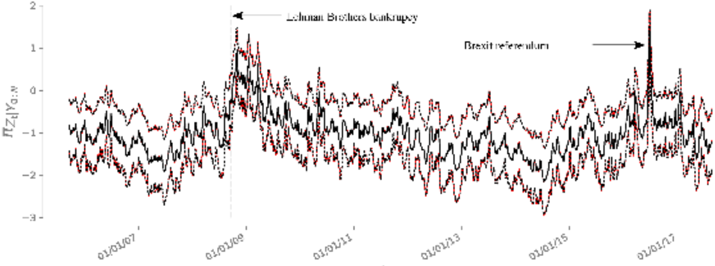


- Slow accumulation of error over time (**sequential** algorithm)
- Acceptance rate 75% for Metropolis independence sampler with transport proposal



Long-time smoothing (25 years)



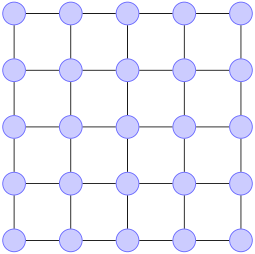


Stochastic volatility example

- Variance diagnostic $\mathbb{V}\mathrm{ar}_\eta[\log(\eta / T_\sharp^{-1}\bar{\pi})]$ values, for a 947-dimensional target π (smoothing and parameter estimation for 945 days) :
  - Laplace map = 5.68; linear maps = 1.49; *degree $\leq$ 7 maps = 0.11*

- **Important open question:** how does error in the approximation of the parameter posterior evolve over time?

Too many cycles...



- For certain graphs, sparsity/decomposability **do not imply decoupling** between the nominal dimension of the problem and the dimension of each transport T_i (or the sparsity of S)
  - Here, $\mathcal{G}$ is an $n \times n$ grid graph
  - $T^{S \cup A}$ acts on $2n$ dimensions at each stage

Beyond the Markov properties of π

- **Key idea:** seek **low-rank** structure and *near-identity* maps
- Example: fix target π to be the posterior density of a Bayesian inference problem,

$$\pi(\mathbf{z}) := \pi_{\text{pos}}(\mathbf{z}) \propto \pi_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y} \,|\, \mathbf{z})\, \pi_{\mathbf{Z}}(\mathbf{z})$$

- Let T_{pr} push forward the reference η to the prior $\pi_{\mathbf{Z}}$ (prior map)

$$\widehat{\pi}_{\text{pos}}(\mathbf{z}) := T_{\text{pr}}^{\sharp}\, \pi_{\text{pos}}(\mathbf{z}) \propto \pi_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y} \,|\, T_{\text{pr}}(\mathbf{z})\,)\, \eta(\mathbf{z})$$

Theorem [Graph decoupling]

If $\eta = \prod_i \eta_{X_i}$ and

$$\text{rank}\, \mathbb{E}_{\eta}\, [\, \nabla \log R \otimes \nabla \log R\,] = k, \qquad R = \widehat{\pi}_{\text{pos}}/\eta = \pi_{\mathbf{Y}|\mathbf{Z}} \circ T_{\text{pr}}$$

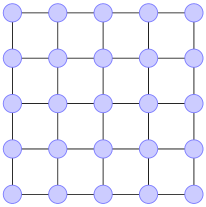
then there exists a rotation Q such that:

$$Q^{\sharp}\, \widehat{\pi}_{\text{pos}}(\mathbf{z}) = g(\, z_1, \ldots, z_k\,) \prod_{i>k}^{n} \eta_{X_i}(z_i)$$

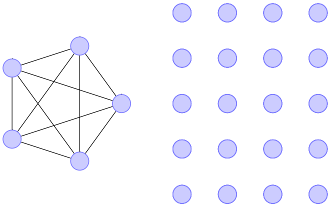
Changing the Markov structure...

- The pullback has a different Markov structure:

$$Q^\sharp\,\widehat{\pi}_{\text{pos}}(\mathbf{z}) = g(\,z_1, \ldots, z_k\,) \prod_{i>k}^{n} \eta_{X_i}(z_i)$$



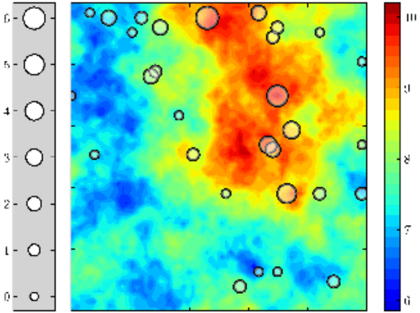
$\mathcal{G}$



$\mathcal{G}$ **Pullback**

- **Corollary:** There exists a transport $T_\sharp\,\eta = Q^\sharp\,\widehat{\pi}_{\text{pos}}$ of the form $T(\mathbf{x}) = [\,g(\mathbf{x}_{1:k}),\, x_{k+1}, \ldots,\, x_n\,]$, where $g : \mathbb{R}^k \to \mathbb{R}^k$.
- The composition $T_{\text{pr}} \circ Q \circ T$ pushes forward η to π_{pos}
- Why low rank structure? For example, **few data-informed directions**.

Log-Gaussian Cox process

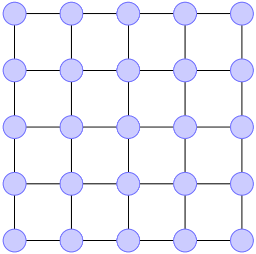


- 4096-D **GMRF prior**, $\mathbf{Z} \sim \mathcal{N}(\mu, \Gamma)$, Γ^{-1} specified through $\triangle + \kappa^2 \, \mathrm{Id}$
- 30 **sparse observations** at locations $i \in \mathcal{I}$, $\mathbf{Y}_i | \mathbf{Z}_i \sim \mathrm{Pois}(\exp \mathbf{Z}_i)$
- Posterior density $\mathbf{Z}|\mathbf{Y} \sim \pi_{\mathrm{pos}}$ is:

$$\pi_{\mathrm{pos}}(\mathbf{z}) \propto \prod_{i \in \mathcal{I}} \exp[-\exp(z_i) + z_i \cdot y_i] \exp\left[-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu})^{\top} \Gamma^{-1} (\mathbf{z} - \boldsymbol{\mu})\right]$$

- What is an independence map $\mathcal{G}$ for π_{pos}?

Log-Gaussian Cox process



$\mathcal{G}$

- 4096-D **GMRF prior**, $\mathbf{Z} \sim \mathcal{N}(\mu, \Gamma)$, Γ^{-1} specified through $\triangle + \kappa^2\,\mathrm{Id}$
- 30 **sparse observations** at locations $i \in \mathcal{I}$, $\mathbf{Y}_i | \mathbf{Z}_i \sim \mathrm{Pois}(\exp \mathbf{Z}_i)$
- Posterior density $\mathbf{Z}|\mathbf{Y} \sim \pi_{\mathrm{pos}}$ is:

$$\pi_{\mathrm{pos}}(\mathbf{z}) \propto \prod_{i\in\mathcal{I}} \exp[-\exp(z_i) + z_i \cdot y_i] \exp\left[-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top \Gamma^{-1}(\mathbf{z}-\boldsymbol{\mu})\right]$$

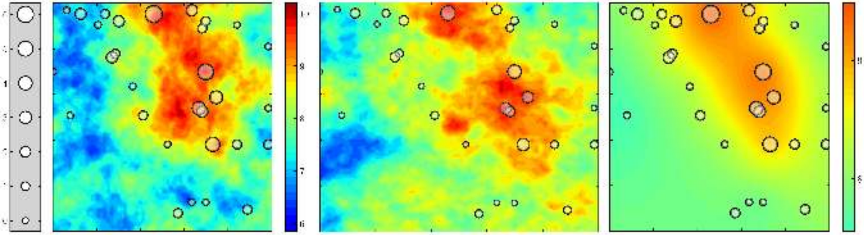
- What is an independence map $\mathcal{G}$ for π_{pos}? A 64×64 grid.

Log-Gaussian Cox process

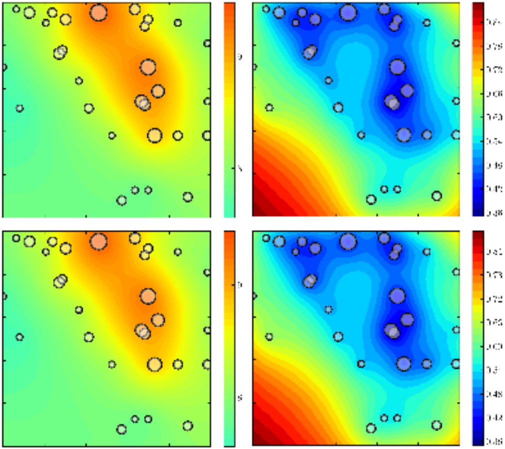
- Fix $\pi_{\text{ref}} \sim \mathcal{N}(0, \mathbf{I})$ and let T_{pr} push forward π_{ref} to π_{pr} (**prior map**)
- Consider the pullback $\widehat{\pi}_{\text{pos}} = T_{\text{pr}}^{\sharp}\, \pi_{\text{pos}}$ and find that

$$\operatorname{rank} \mathbb{E}_{\pi_{\text{ref}}} [\nabla \log R \otimes \nabla \log R] = 30 \ll n, \qquad R = \widehat{\pi}_{\text{pos}}/\pi_{\text{ref}}$$

- *Deflate* the problem and compute a transport map in **30** dimensions
  - Change from prior to posterior concentrated in a **low-dimensional subspace**



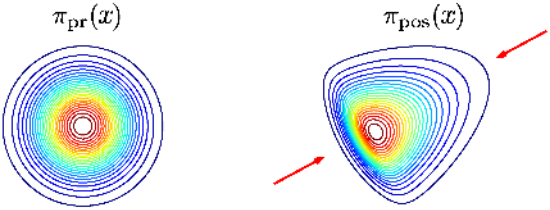
truth posterior sample posterior mean



- *(left)* $\mathbb{E}[\mathbf{Z}|\mathbf{y}]$, *(right)* $\mathbb{V}\mathrm{ar}[\mathbf{Z}|\mathbf{y}]$. *(top)* transport; *(bottom)* MCMC
- Excellent match with reference MCMC solution
- **Can we understand this structure more generally?**

A conjecture

In many situations, the data are informative only on a low-dimensional subspace



$$\boxed{\mathbb{R}^d = \underbrace{X_r}_{\pi_{\mathrm{pos}} \neq \pi_{\mathrm{pr}}} + \underbrace{X_\perp}_{\pi_{\mathrm{pos}} \approx \pi_{\mathrm{pr}}}}$$

Low effective dimensionality of Bayesian inverse problems

Underlying idea: the posterior distribution can be well approximated by

$$\widetilde{\pi}_{\mathrm{pos}}(x) \propto \widetilde{\mathcal{L}}(P_r x)\,\pi_{\mathrm{pr}}(x)$$

for some **positive function** $\widetilde{\mathcal{L}}$ and rank r **linear projector** $P_r \in \mathbb{R}^{d \times d}$

Low effective dimensionality of Bayesian inverse problems

Underlying idea: the posterior distribution can be well approximated by

$$\widetilde{\pi}_{\text{pos}}(x) \propto \widetilde{\mathcal{L}}(P_r x)\, \pi_{\text{pr}}(x)$$

for some **positive function** $\widetilde{\mathcal{L}}$ and rank r **linear projector** $P_r \in \mathbb{R}^{d \times d}$

P_r induces a decomposition of the space

$$x = x_r + x_\perp \qquad \begin{cases} x_r & \in \operatorname{Im}(P_r) \\ x_\perp & \in \operatorname{Ker}(P_r) \end{cases}$$

By construction, $x \mapsto \widetilde{\mathcal{L}}(P_r x) = \widetilde{\mathcal{L}}(x_r)$ is only a function of $x_r \in \operatorname{Im}(P_r) \equiv \mathbb{R}^r$.

Low effective dimensionality of Bayesian inverse problems

Underlying idea: the posterior distribution can be well approximated by

$$\widetilde{\pi}_{\text{pos}}(x) \propto \widetilde{\mathcal{L}}(P_r x)\, \pi_{\text{pr}}(x)$$

for some **positive function** $\widetilde{\mathcal{L}}$ and rank r **linear projector** $P_r \in \mathbb{R}^{d \times d}$

P_r induces a decomposition of the space

$$x = x_r + x_\perp \qquad \begin{cases} x_r & \in \operatorname{Im}(P_r) \\ x_\perp & \in \operatorname{Ker}(P_r) \end{cases}$$

By construction, $x \mapsto \widetilde{\mathcal{L}}(P_r x) = \widetilde{\mathcal{L}}(x_r)$ is only a function of $x_r \in \operatorname{Im}(P_r) \equiv \mathbb{R}^r$.

If $r \ll d$:

- Design **dimension-independent** MCMC algorithms to sample from π_{pos}. [Cui, Law, M 2016]
- Build surrogates for the **low-dimensional** function $x_r \mapsto \widetilde{\mathcal{L}}(x_r)$ with a reasonable complexity

Many methods for constructing P_r and $\widetilde{\mathcal{L}}$

- P_r can be defined as a projector onto the **dominant eigenspace** of a matrix $\mathbf{H} \in \mathbb{R}^{d \times d}$ which contains "relevant information"

Many methods for constructing P_r and $\widetilde{\mathcal{L}}$

- P_r can be defined as a projector onto the **dominant eigenspace** of a matrix $\mathbf{H} \in \mathbb{R}^{d \times d}$ which contains "relevant information"
  - **Likelihood-informed subspace (LIS)** [Cui et al 2014]

$$\mathbf{H}_{\mathrm{LIS}} = \int (\nabla G)^T \Gamma_{\mathrm{obs}}^{-1} (\nabla G) \, \mathrm{d}\pi_{\mathrm{pos}}$$

where $\mathcal{L}_y$ follows from $y \sim \mathcal{N}(G(x), \Gamma_{\mathrm{obs}})$

  - **Active subspace (AS)** [Constantine, Kent, Bui-Thanh 2015]

$$\mathbf{H}_{\mathrm{AS}} = \int \nabla \log \mathcal{L}_y \otimes \nabla \log \mathcal{L}_y \, \mathrm{d}\pi_{\mathrm{pr}}$$

Many methods for constructing P_r and $\widetilde{\mathcal{L}}$

- P_r can be defined as a projector onto the **dominant eigenspace** of a matrix $\mathbf{H} \in \mathbb{R}^{d \times d}$ which contains "relevant information"
  - **Likelihood-informed subspace (LIS)** [Cui et al 2014]

$$\mathbf{H}_{\mathrm{LIS}} = \int (\nabla G)^T \Gamma_{\mathrm{obs}}^{-1} (\nabla G) \, \mathrm{d}\pi_{\mathrm{pos}}$$

where $\mathcal{L}_y$ follows from $y \sim \mathcal{N}(G(x), \Gamma_{\mathrm{obs}})$

  - **Active subspace (AS)** [Constantine, Kent, Bui-Thanh 2015]

$$\mathbf{H}_{\mathrm{AS}} = \int \nabla \log \mathcal{L}_y \otimes \nabla \log \mathcal{L}_y \, \mathrm{d}\pi_{\mathrm{pr}}$$

- Different definitions of $\widetilde{\mathcal{L}}$:
  - Fix complementary parameters (LIS): $\widetilde{\mathcal{L}}(P_r x) = \mathcal{L}_y(P_r x + (I - P_r) m_0)$
  - Via the conditional expectation of the log-likelihood (AS)

$$\widetilde{\mathcal{L}}(P_r x) = \exp \mathbb{E}_{\pi_{\mathrm{pr}}}(\log \mathcal{L}_y | P_r x)$$

Objective

Build an approximation of π_{pos} of the form

$$\widetilde{\pi}_{\text{pos}}(x) \propto \widetilde{\mathcal{L}}(P_r x)\pi_{\text{pr}}(x) \qquad \text{with} \begin{cases} \widetilde{\mathcal{L}} : \mathbb{R}^d \to \mathbb{R}^+ \\ P_r \in \mathbb{R}^{d\times d} \text{ rank-}r \text{ projector} \end{cases}$$

such that

$$\boxed{D_{\text{KL}}(\pi_{\text{pos}}||\widetilde{\pi}_{\text{pos}}) \leq \varepsilon}$$

with $r = r(\varepsilon)$ much smaller than d.

Decomposition of the error

A "Pythagorean" theorem

For any P_r and $\widetilde{\mathcal{L}}$ we have

$$D_{\mathrm{KL}}\left(\pi_{\mathrm{pos}} \middle|\middle| \widetilde{\pi}_{\mathrm{pos}}\right) = \underbrace{D_{\mathrm{KL}}\left(\pi_{\mathrm{pos}} \middle|\middle| \pi^*_{\mathrm{pos}}\right)}_{=\text{function}(P_r)} + \underbrace{D_{\mathrm{KL}}\left(\pi^*_{\mathrm{pos}} \middle|\middle| \widetilde{\pi}_{\mathrm{pos}}\right)}_{=\text{function}(P_r,\widetilde{\mathcal{L}})}$$

where

$$\pi^*_{\mathrm{pos}}(x) \propto \mathbb{E}_{\pi_{\mathrm{pr}}}\left(\mathcal{L}_y \middle| P_r x\right)\pi_{\mathrm{pr}}(x)$$

Decomposition of the error

A "Pythagorean" theorem

For any P_r and $\widetilde{\mathcal{L}}$ we have

$$D_{\mathrm{KL}}\left(\pi_{\mathrm{pos}} \middle|\middle| \widetilde{\pi}_{\mathrm{pos}}\right) = \underbrace{D_{\mathrm{KL}}\left(\pi_{\mathrm{pos}} \middle|\middle| \pi^*_{\mathrm{pos}}\right)}_{=\mathrm{function}(P_r)} + \underbrace{D_{\mathrm{KL}}\left(\pi^*_{\mathrm{pos}} \middle|\middle| \widetilde{\pi}_{\mathrm{pos}}\right)}_{=\mathrm{function}(P_r, \widetilde{\mathcal{L}})}$$

where

$$\pi^*_{\mathrm{pos}}(x) \propto \mathbb{E}_{\pi_{\mathrm{pr}}}\left(\mathcal{L}_y \middle| P_r x\right) \pi_{\mathrm{pr}}(x)$$

This allows decoupling the construction of $\widetilde{\mathcal{L}}$ and P_r.

- Given P_r, the function $\widetilde{\mathcal{L}}$ such that $\widetilde{\mathcal{L}}(P_r x) = \mathbb{E}_{\pi_{\mathrm{pr}}}\left(\mathcal{L}_y \middle| P_r x\right)$ yields
$$D_{\mathrm{KL}}\left(\pi^*_{\mathrm{pos}} \middle|\middle| \widetilde{\pi}_{\mathrm{pos}}\right) = 0$$
- How to construct P_r such that
$$D_{\mathrm{KL}}\left(\pi_{\mathrm{pos}} \middle|\middle| \pi^*_{\mathrm{pos}}\right) \leq \varepsilon$$
with a rank $r \ll d$?

Constructing the projector P_r

Assumption (on the prior distribution)

There exist functions V and Ψ such that

$$\pi_{\mathrm{pr}}(x) \propto \exp\big(-V(x) - \Psi(x)\big) \quad \text{with} \quad \begin{cases} \nabla^2 V \succeq \Gamma \\ \exp(\sup \Psi - \inf \Psi) \leq \kappa \end{cases}$$

for some SPD matrix $\Gamma \in \mathbb{R}^{d \times d}$ and some $\kappa \geq 1$.

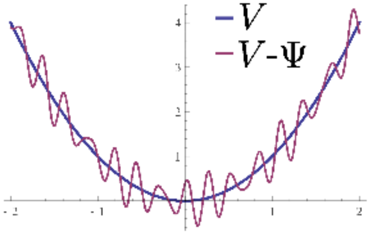
Constructing the projector P_r

Assumption (on the prior distribution)

There exist functions V and Ψ such that

$$\pi_{\mathrm{pr}}(x) \propto \exp\big(-V(x) - \Psi(x)\big) \quad \text{with} \quad \begin{cases} \nabla^2 V \succeq \Gamma \\ \exp(\sup \Psi - \inf \Psi) \leq \kappa \end{cases}$$

for some SPD matrix $\Gamma \in \mathbb{R}^{d\times d}$ and some $\kappa \geq 1$.



- Gaussian prior $\pi_{\mathrm{pr}} = \mathcal{N}(\mu_{\mathrm{pr}}, \Sigma_{\mathrm{pr}})$ satisfies this assumption with $\Gamma = \Sigma_{\mathrm{pr}}^{-1}$ and $\kappa = 1$
- Gaussian mixture $\pi_{\mathrm{pr}} \propto \sum_i \mathcal{N}(\mu_i, \Sigma_i)$ also satisfies this assumption

Constructing the projector P_r

Based on this assumption, π_{pr} satisfies the **logarithmic Sobolev inequality** [Ledoux 1997]

$$\int h^2 \log \frac{h^2}{\int h^2 \, \mathrm{d}\pi_{\mathrm{pr}}} \, \mathrm{d}\pi_{\mathrm{pr}} \leq 2\kappa \int \|\nabla h\|^2_{\Gamma^{-1}} \mathrm{d}\pi_{\mathrm{pr}}$$

for any function h with sufficient regularity.

Constructing the projector P_r

Based on this assumption, π_{pr} satisfies the **logarithmic Sobolev inequality** [Ledoux 1997]

$$\int h^2 \log \frac{h^2}{\int h^2 \, \mathrm{d}\pi_{\text{pr}}} \, \mathrm{d}\pi_{\text{pr}} \leq 2\kappa \int \|\nabla h\|^2_{\Gamma^{-1}} \mathrm{d}\pi_{\text{pr}}$$

for any function h with sufficient regularity.

Proposition (subspace logarithmic Sobolev inequality)

π_{pr} also satisfies

$$\int h^2 \log \frac{h^2}{\mathbb{E}(h^2 | P_r x)} \, \mathrm{d}\pi_{\text{pr}} \leq 2\kappa \int \|(I_d - P_r^T)\nabla h\|^2_{\Gamma^{-1}} \, \mathrm{d}\pi_{\text{pr}}$$

for any function h with sufficient regularity and any projector P_r.

Constructing the projector P_r

Corollary

For any projector P_r we have

$$D_{\mathrm{KL}}\left(\pi_{\mathrm{pos}} \middle\| \pi_{\mathrm{pos}}^*\right) \leq \frac{\kappa}{2} \mathcal{R}_{\pi_{\mathrm{pos}}}(P_r)$$

where

$$\mathcal{R}_{\pi_{\mathrm{pos}}}(P_r) = \int \|(I_d - P_r^T)\nabla \log \mathcal{L}_y\|_{\Gamma^{-1}}^2 \, \mathrm{d}\pi_{\mathrm{pos}}$$

Constructing the projector P_r

Corollary

For any projector P_r we have

$$D_{\mathrm{KL}}\left(\pi_{\mathrm{pos}} \middle|\middle| \pi^*_{\mathrm{pos}}\right) \leq \frac{\kappa}{2} \mathcal{R}_{\pi_{\mathrm{pos}}}(P_r)$$

where

$$\mathcal{R}_{\pi_{\mathrm{pos}}}(P_r) = \int \|(I_d - P_r^T) \nabla \log \mathcal{L}_y\|^2_{\Gamma^{-1}} \, \mathrm{d}\pi_{\mathrm{pos}}$$

Finding P_r that minimizes this bound corresponds to **PCA** of $\nabla \log \mathcal{L}_y(X)$.

- For a fixed r, the minimizer P_r^* of the **reconstruction error** $\mathcal{R}_{\pi_{\mathrm{pos}}}(P_r)$ is the Γ-orthogonal projector onto the dominant generalized eigenspace of

$$\mathbf{H} = \int \nabla \log \mathcal{L}_y \otimes \nabla \log \mathcal{L}_y \, \mathrm{d}\pi_{\mathrm{pos}}$$

- Furthermore we have $\mathcal{R}_{\pi_{\mathrm{pos}}}(P_r^*) = \sum_{i>r} \lambda_i$, where λ_i is the i-th generalized eigenvalue of $(\mathbf{H}, \Gamma)$

An idealized algorithm

1. Compute

$$\mathbf{H} = \int \nabla \log \mathcal{L}_y \otimes \nabla \log \mathcal{L}_y \; d\pi_{\text{pos}}$$

2. Define P_r as the projector on the dominant eigenspace of $(\mathbf{H}, \Gamma)$
3. Compute the conditional expectation

$$\widetilde{\mathcal{L}}(P_r x) = \mathbb{E}_{\text{pr}}(\mathcal{L}_y | P_r x)$$

Then $\pi^*_{\text{pos}}(x) \propto \widetilde{\mathcal{L}}(P_r x)\pi_{\text{Spr}}(x)$ satisfies

$$\boxed{D_{\text{KL}}\left(\pi_{\text{pos}} \middle|\middle| \pi^*_{\text{pos}}\right) \leq \frac{\kappa}{2} \sum_{i>r} \lambda_i}$$

- At step 2, we can choose the rank $r = r(\varepsilon)$ of P_r such that

$$D_{\text{KL}}(\pi_{\text{pos}} || \pi^*_{\text{pos}}) \leq \varepsilon$$

- A strong decay in λ_i implies $r(\varepsilon) \ll d$

An idealized algorithm

1. Compute
$$\mathbf{H} = \int \nabla \log \mathcal{L}_y \otimes \nabla \log \mathcal{L}_y \; \mathrm{d}\pi_{\mathrm{pos}}$$
2. Define P_r as the projector on the dominant eigenspace of $(\mathbf{H}, \Gamma)$
3. Compute the conditional expectation
$$\widetilde{\mathcal{L}}(P_r x) = \mathbb{E}_{\mathrm{pr}}(\mathcal{L}_y | P_r x)$$

Practical issues

- Evaluating $\mathbf{H}$ requires computing an integral over the posterior
- Computing the conditional expectation requires some effort

- Monte Carlo approximation of **H**:

$$\mathbf{H} \approx \widehat{\mathbf{H}}_K := \frac{1}{K}\sum_{i=1}^{K} \nabla \log \mathcal{L}_y(X_i) \otimes \nabla \log \mathcal{L}_y(X_i) \quad \text{with} \quad X_i \overset{\text{iid}}{\sim} \pi_{\text{pos}}$$

Proposition

Under some assumptions, **quasi-optimal projectors** are obtained with high probability $1-\delta$ if

$$K \geq \mathcal{O}\Big(\sqrt{\operatorname{rank}(H)} + \sqrt{\log(2\delta^{-1})}\Big)^2$$

- Key assumption: $\nabla \log \mathcal{L}_y(X)$ is *sub-Gaussian*, for $X \sim \pi_{\text{pos}}$

Approximation of $\pi^*_{\mathbf{pos}}(x) \propto \mathbb{E}_{\mathbf{pr}}(\mathcal{L}_y|P_r x)\pi_{\mathbf{pr}}(x)$

- The conditional expectation $\mathbb{E}_{\text{pr}}(\mathcal{L}_y|P_r x)$ can be expressed as
$$x \mapsto \int \mathcal{L}_y(P_r x + (I_d - P_r)z)\ \pi_{\text{pr}}(z|P_r x)\mathrm{d}z$$
where $\pi_{\text{pr}}(\cdot|P_r x)$ denotes the conditional prior, which depends on x.

Approximation of $\pi^*_{\text{pos}}(x) \propto \mathbb{E}_{\text{pr}}(\mathcal{L}_y | P_r x)\pi_{\text{pr}}(x)$

- The conditional expectation $\mathbb{E}_{\text{pr}}(\mathcal{L}_y | P_r x)$ can be expressed as
$$x \mapsto \int \mathcal{L}_y(P_r x + (I_d - P_r)z)\ \pi_{\text{pr}}(z | P_r x) \mathrm{d}z$$
where $\pi_{\text{pr}}(\cdot | P_r x)$ denotes the conditional prior, which depends on x.

- Consider the following Monte Carlo estimate
$$\widetilde{\mathcal{L}} : x \mapsto \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_y(P_r x + (I_d - P_r)Z_i) \quad , \quad Z_i \overset{\text{iid}}{\sim} \pi_{\text{pr}}$$
In general, $\widetilde{\mathcal{L}}(P_r x)$ is a biased estimator for $\mathbb{E}_{\text{pr}}(\mathcal{L}_y | P_r x)$.

Approximation of $\pi^*_{\mathrm{pos}}(x) \propto \mathbb{E}_{\mathrm{pr}}(\mathcal{L}_y|P_r x)\pi_{\mathrm{pr}}(x)$

- The conditional expectation $\mathbb{E}_{\mathrm{pr}}(\mathcal{L}_y|P_r x)$ can be expressed as
$$x \mapsto \int \mathcal{L}_y(P_r x + (I_d - P_r)z)\ \pi_{\mathrm{pr}}(z|P_r x)\mathrm{d}z$$
where $\pi_{\mathrm{pr}}(\cdot|P_r x)$ denotes the conditional prior, which depends on x.

- Consider the following Monte Carlo estimate
$$\widetilde{\mathcal{L}} : x \mapsto \frac{1}{M}\sum_{i=1}^{M} \mathcal{L}_y(P_r x + (I_d - P_r)Z_i) \quad , \quad Z_i \overset{\text{iid}}{\sim} \pi_{\mathrm{pr}}$$
In general, $\widetilde{\mathcal{L}}(P_r x)$ is a biased estimator for $\mathbb{E}_{\mathrm{pr}}(\mathcal{L}_y|P_r x)$.

Proposition

The random distribution $\widetilde{\pi}_{\mathrm{pos}}(x) \propto \widetilde{\mathcal{L}}(P_r x)\pi_{\mathrm{pr}}(x)$ is such that
$$\mathbb{E}\Big(D_{\mathrm{KL}}(\pi^*_{\mathrm{pos}}||\widetilde{\pi}_{\mathrm{pos}})\Big) \lesssim \left(C_1 + \frac{C_2}{M}\right) \mathcal{R}_{\pi_{\mathrm{pos}}}(P_r)$$

Approximating H using other distributions

- Recall that

$$\mathcal{R}_{\pi_{\text{pos}}}(P_r) = \int \|(I_d - P_r^T)\nabla \log \mathcal{L}_y\|^2_{\Gamma^{-1}} \, \mathrm{d}\pi_{\text{pos}}$$

- Let ρ be a tractable density and consider

$$\mathcal{R}_{\rho}(P_r) = \int \|(I_d - P_r^T)\nabla \log \mathcal{L}_y\|^2_{\Gamma^{-1}} \, \mathrm{d}\rho$$

Approximating H using other distributions

- Recall that

$$\mathcal{R}_{\pi_{\text{pos}}}(P_r) = \int \|(I_d - P_r^T)\nabla \log \mathcal{L}_y\|^2_{\Gamma^{-1}} \, \mathrm{d}\pi_{\text{pos}}$$

- Let ρ be a tractable density and consider

$$\mathcal{R}_{\rho}(P_r) = \int \|(I_d - P_r^T)\nabla \log \mathcal{L}_y\|^2_{\Gamma^{-1}} \, \mathrm{d}\rho$$

- The minimizer P_r^* of $P_r \mapsto \mathcal{R}_\rho(P_r)$ is such that

$$\boxed{\mathcal{R}_{\pi_{\text{pos}}}(P_r^*) \leq \left(\sup \frac{\pi_{\text{pos}}}{\rho}\right) \sum_{i>r} \lambda_i^{(\rho)}}$$

where $\lambda_i^{(\rho)}$ is the i-th generalized eigenvalue of

$$\mathbf{H}^{(\rho)} = \int \nabla \log \mathcal{L}_y \otimes \nabla \log \mathcal{L}_y \, \mathrm{d}\rho$$

A practical algorithm

1. Compute (e.g., with Monte Carlo)
$$\mathbf{H}^{(\rho)} = \int \nabla \log \mathcal{L}_y \otimes \nabla \log \mathcal{L}_y \, \mathrm{d}\rho.$$
2. Compute the projector P_r based on $\mathbf{H}^{(\rho)}$
3. Draw one sample $Z \sim \pi_{\mathrm{pr}}$ and let
$$\widetilde{\mathcal{L}} : x \mapsto \mathcal{L}_y(P_r x + (I_d - P_r) Z)$$

Then $\widetilde{\pi}_{\mathrm{pos}}(x) \propto \widetilde{\mathcal{L}}(P_r x)\pi_{\mathrm{pr}}(x)$ is such that

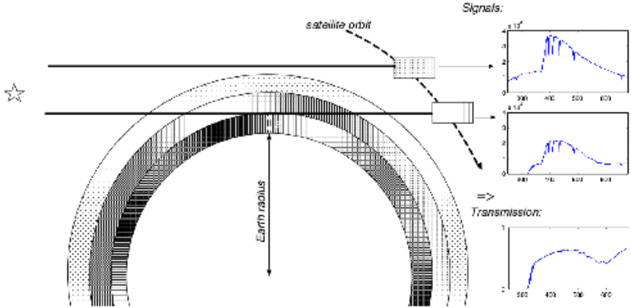
$$\boxed{\mathbb{E}\Big(D_{\mathrm{KL}}\big(\pi_{\mathrm{pos}}\big|\big|\widetilde{\pi}_{\mathrm{pos}}\big)\Big) \leq (cst)\Big(\sup \frac{\pi_{\mathrm{pos}}}{\rho}\Big) \sum_{i>r} \lambda_i^{(\rho)}}$$

- Ideally, ρ should be close to π_{pos}
- The spectrum of $(\mathbf{H}^{(\rho)}, \Gamma)$ is still an indicator for the low effective dimensionality of the problem!

GOMOS: atmospheric remote sensing (e.g., [Tamminen et al. 2004])

- Estimate gas densities $x = \varrho^{\text{gas}}(z)$ from transmission spectra $y_\omega(z)$
- Beer's law:

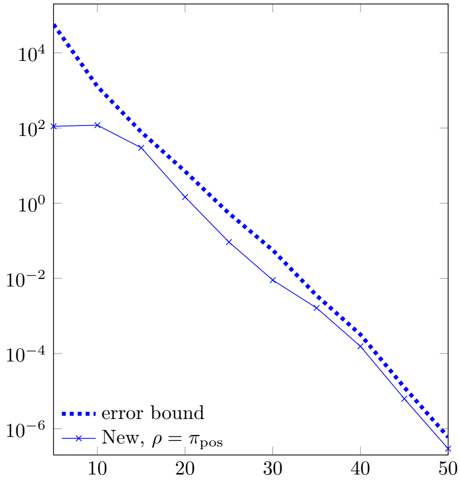
$$y_\omega(z) = \exp\left(- \int_{\text{light path}} \sum_{\text{gas}} \alpha_\omega^{\text{gas}}(z(\zeta))\, \varrho^{\text{gas}}(z(\zeta))\, \mathrm{d}\zeta \right) + \xi$$



- Gaussian prior $\mathcal{N}(\mu_{\text{pr}}, \Sigma_{\text{pr}})$ (hence $\Gamma = \Sigma_{\text{pr}}^{-1}$ and $\kappa = 1$)
- After discretization of the atmosphere, $\dim(x) = 200$

Results

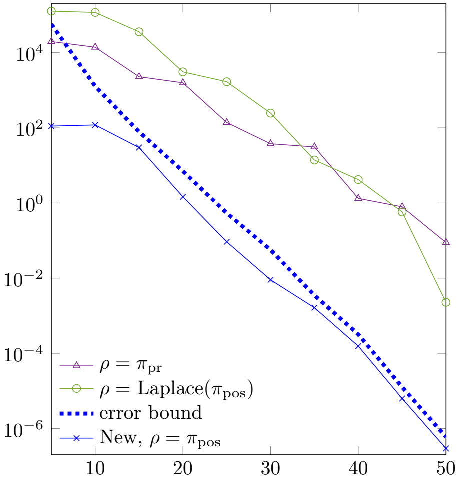
$$D_{\mathrm{KL}}(\pi_{\mathrm{pos}}||\widetilde{\pi}_{\mathrm{pos}}) = \mathit{function}(r)$$



$$\mathbf{H} = \int \nabla \log \mathcal{L}_y \otimes \nabla \log \mathcal{L}_y \; \mathrm{d}\pi_{\mathrm{pos}}$$

Results

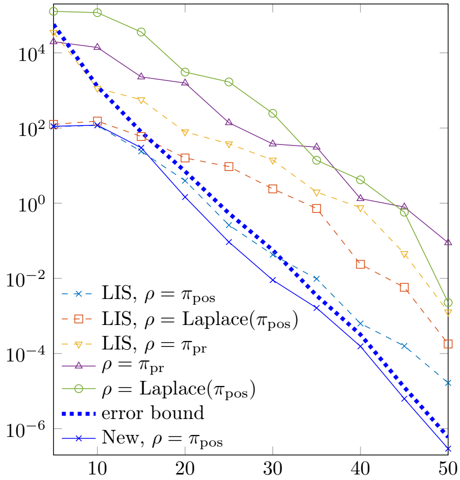
$$D_{\mathrm{KL}}(\pi_{\mathrm{pos}}||\widetilde{\pi}_{\mathrm{pos}}) = \mathit{function}(r)$$



$$\mathbf{H}^{(\rho)} = \int \nabla \log \mathcal{L}_y \otimes \nabla \log \mathcal{L}_y \, \mathrm{d}\rho$$

Results

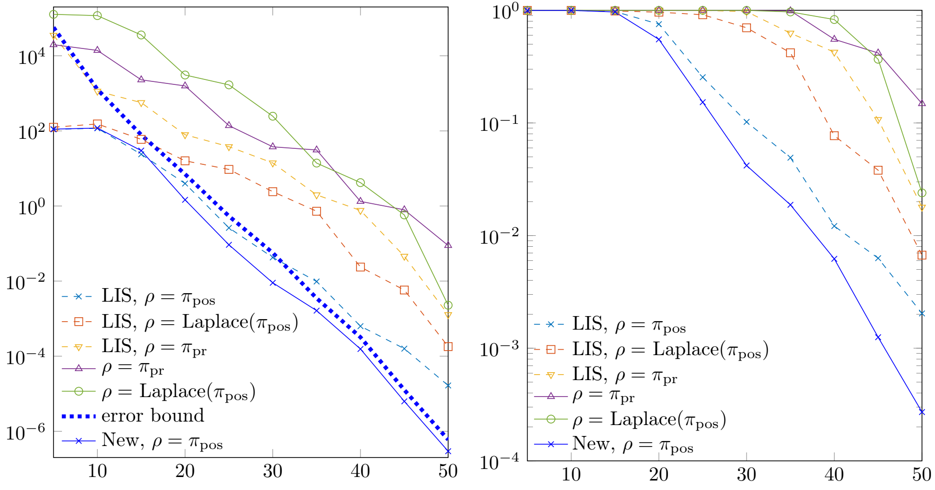
$$D_{\mathrm{KL}}(\pi_{\mathrm{pos}}||\widetilde{\pi}_{\mathrm{pos}}) = \mathit{function}(r)$$



$$\mathbf{H}^{(\rho)} = \int \nabla \log \mathcal{L}_y \otimes \nabla \log \mathcal{L}_y \; d\rho \qquad \mathbf{H}^{(\rho)}_{\mathrm{LIS}} = \int (\nabla G)^T \Gamma_{\mathrm{obs}}^{-1} (\nabla G) \; d\rho$$

Results

$D_{\mathrm{KL}}(\pi_{\mathrm{pos}} || \widetilde{\pi}_{\mathrm{pos}}) = \mathit{function}(r)$ $\quad$ $d_{\mathrm{Hell}}(\pi_{\mathrm{pos}}, \widetilde{\pi}_{\mathrm{pos}}) = \mathit{function}(r)$



$$\mathbf{H}^{(\rho)} = \int \nabla \log \mathcal{L}_y \otimes \nabla \log \mathcal{L}_y \, \mathrm{d}\rho \qquad \mathbf{H}^{(\rho)}_{\mathrm{LIS}} = \int (\nabla G)^T \Gamma_{\mathrm{obs}}^{-1} (\nabla G) \, \mathrm{d}\rho$$

An iterative algorithm

Draw M samples $Y_1, \dots, Y_M$ from π_{pr}

for $\ell = 0, \dots, L$ **do**

  if $\ell = 0$ **then**

    Draw K samples $X_1^{(\ell)}, \dots, X_K^{(\ell)}$ from π_{pr}

    Compute $\nabla \log \mathcal{L}_y(X_k^{(\ell)})$ and set the weights $w_k^{(\ell)} = 1$

  else

    Draw K samples $X_1, \dots, X_K$ from $\hat{\nu}_r^{(\ell)}$ (e.g, using MCMC)

    Compute $\nabla \log \mathcal{L}_y(X_k^{(\ell)})$ and $w_k^{(\ell)} = \frac{\mathcal{L}_y(X_k^{(\ell)})}{\hat{F}_r^{(\ell)}(X_k^{(\ell)})}$

  Assemble the matrix

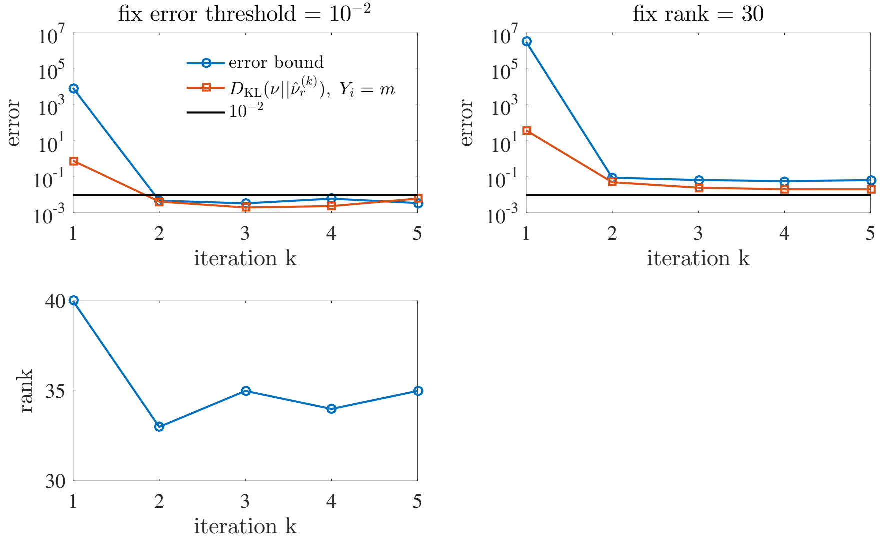
$$\hat{H}^{(\ell)} = \frac{1}{\sum_{k=1}^{K} w_k^{(\ell)}} \sum_{k=1}^{K} w_k^{(\ell)} \left(\nabla \log \mathcal{L}_y(X_k^{(\ell)})\right)\left(\nabla \log \mathcal{L}_y(X_k^{(\ell)})\right)^\top$$

  Compute a projector $P_r^{(\ell+1)}$ such that $\mathcal{R}_\Gamma(P_r^{(\ell+1)}, \hat{H}^{(\ell)}) \leq \varepsilon$

  Define the approximate distribution $\hat{\nu}_r^{(\ell+1)}$ as

$$\frac{\mathrm{d}\hat{\nu}_r^{(\ell+1)}}{\mathrm{d}\pi_{\mathrm{pr}}} \propto \hat{F}_r^{(\ell+1)}, \quad \text{where} \quad \hat{F}_r^{(\ell+1)} = \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_y\left(P_r^{(\ell+1)} x + (I_d - P_r^{(\ell+1)}) Y_i\right)$$

Iterative algorithm: results



(left) fixed threshold; *(right)* fixed rank

Summary on dimension reduction

Conclusions:

- Exploit the low effective dimensionality of Bayesian inverse problems
- Methodology:
  - Derive an upper bound on the error (KL-divergence)
  - Compute a minimizer of the upper bound using PCA on $\nabla \log \mathcal{L}_y$
- Better performance than existing gradient-based methods (e.g., likelihood-informed subspace or active subspace)

Summary on dimension reduction

Conclusions:

- Exploit the low effective dimensionality of Bayesian inverse problems
- Methodology:
  - Derive an upper bound on the error (KL-divergence)
  - Compute a minimizer of the upper bound using PCA on $\nabla \log \mathcal{L}_y$
- Better performance than existing gradient-based methods (e.g., likelihood-informed subspace or active subspace)

Open questions:

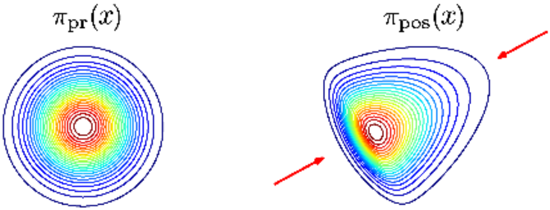
- Does there exist an optimal projector, i.e., a minimizer of the KL divergence?
- What is the best computational strategy to approximate $\mathbf{H}$?

Back to transport: low rank structure

- Let $U = [U_r \;\; U_\perp] \in \mathbb{R}^{n\times n}$ be a unitary matrix, with $U_r \in \mathbb{R}^{n\times r}$. A **lazy map** $T : \mathbb{R}^n \to \mathbb{R}^n$ takes the form:
$$T(z) = U_r \tau(z_1, \ldots, z_r) + U_\perp z_\perp$$
for some diffeomorphism $\tau : \mathbb{R}^r \to \mathbb{R}^r$.
  - Map $T \in \mathcal{T}_r(U)$ departs from the identity only on an r-dimensional subspace
- **Proposition:** For any lazy map $T \in \mathcal{T}_r(U)$, there exists a strictly positive function $f : \mathbb{R}^r \to \mathbb{R}_+$ such that
$$T_\sharp \eta(x) = f(U_r^\top x)\,\eta(x),$$
for all $x \in \mathbb{R}^n$ where $\eta = \mathcal{N}(0, \mathbf{I}_n)$. Conversely, any density of the form $f(U_r^\top x)\,\eta(x)$ for some $f : \mathbb{R}^r \to \mathbb{R}_+$ admits a lazy map representation.

Why would such structure (approximately) appear?

- *Bayesian inverse problems:* data only partially informative; posterior departs from the prior primarily on a low-dimensional subspace.
- Formalized by *likelihood-informed subspace* [Cui et al. 2014]; also, active subspace [Constantine et al. 2015], and recent refinements/connections [Zahm et al. 2018].



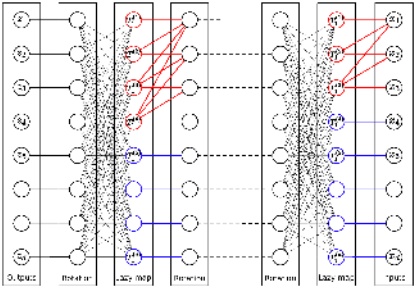
Error bound and subspace

How to find a good U_r?

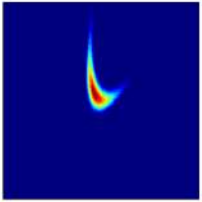
- Define
$$H_\pi := \int \left(\nabla \log \frac{\pi}{\eta}\right)\left(\nabla \log \frac{\pi}{\eta}\right)^\top d\pi$$
.
- Let (λ_i, u_i) be the ith eigenpair of H_π and put $U_r = [u_1\, u_2 \cdots u_r]$.
- **Theorem** [Zahm et al. 2018]:
$$\mathcal{D}_{KL}(\pi || T_\sharp^\star \eta) \leq \frac{1}{2}(\lambda_{r+1} + \ldots + \lambda_d).$$
where $T_\sharp^\star \eta = f^\star(U_r^\top x)\eta(x)$ and $f^\star(z_r) = \mathbb{E}_{X\sim\eta}\left[\frac{\pi(X)}{\eta(X)} \middle| U_r^\top X = z_r\right]$.
- Good approximation when the spectrum of H_π decays quickly
- Uses a *ridge approximation* of $d\pi/d\eta$ (e.g., the likelihood), with optimal profile function $f^\star$

Layers of lazy maps

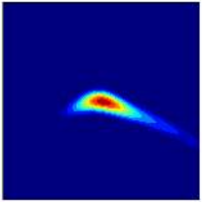
- What if (λ_i) do not decay quickly? What if we are limited to small r?
- Answer: **layers** of lazy maps, via a greedy construction
  - Given (π, η, r_1): compute H_π and construct a first lazy map T_1
  - Pull back π by T_1: $\pi_2 := (T_1^{-1})_\sharp \pi$
  - Given (π_2, η, r_2): compute H_{π_2} and construct a next lazy map $T_2 \ldots$
  - **Generic iteration**: at stage ℓ, build a lazy map to the pullback $\pi_\ell := (T_1 \circ T_2 \circ \cdots \circ T_{\ell-1})_\sharp^{-1} \pi$
  - **Stop** when $\frac{1}{2}\operatorname{Tr}(H_{\pi_\ell}) < \epsilon$



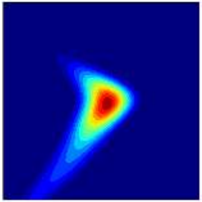
Example: rotated "banana" target distribution, $r = 1$ maps



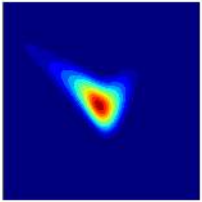
Target π



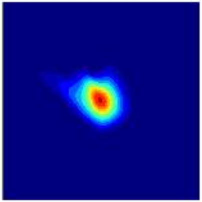
$\mathfrak{T}_1^\sharp \pi$



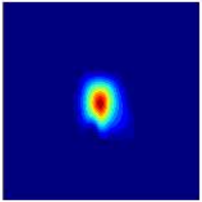
$\mathfrak{T}_2^\sharp \pi$



$\mathfrak{T}_3^\sharp \pi$

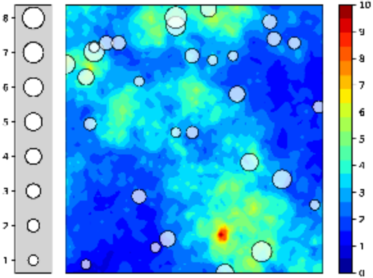


$\mathfrak{T}_5^\sharp \pi$

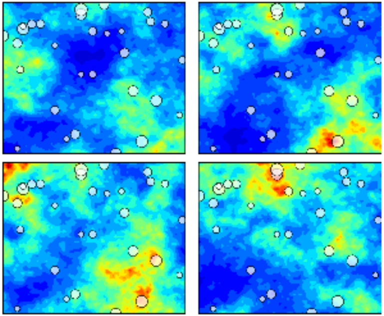


$\mathfrak{T}_8^\sharp \pi$

Example: log-Gaussian Cox process

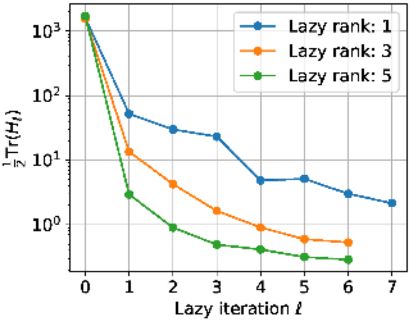


Field $\mathbf{\Lambda}^*$ and observations $\mathbf{y}^*$

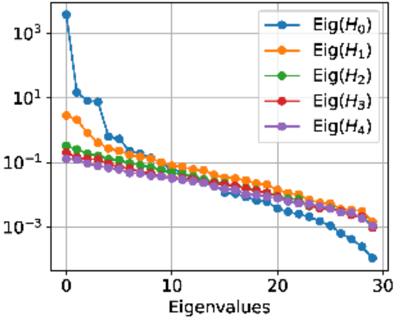


Realizations of $\mathbf{\Lambda} \sim \pi_{\mathbf{\Lambda}|\mathbf{y}^*}$

- Parameter dimension $n = 4096$, 30 observations; fixed ranks r



Convergence

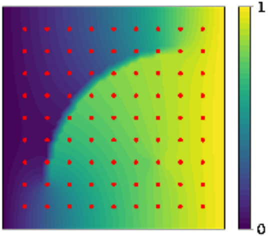


Spectrum of H_{π_ℓ}

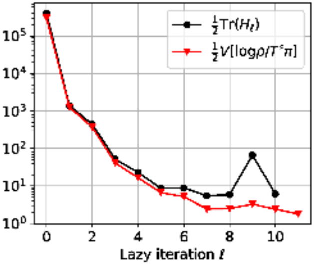
Example: elliptic PDE Bayesian inverse problem

$$
\begin{cases}
\nabla \cdot \left(e^{\kappa(\mathrm{x})} \nabla u(\mathbf{x})\right) = 0, \quad \text{for } \mathbf{x} \in \mathcal{D} := [0,1]^2 , \\
u(\mathbf{x}) = 0 \text{ for } x_1 = 0, \;\; u(\mathbf{x}) = 1 \text{ for } x_1 = 1, \;\; \frac{\partial u(\mathrm{x})}{\partial \mathbf{n}} = 0 \text{ for } x_2 \in \{0,1\}
\end{cases}
$$

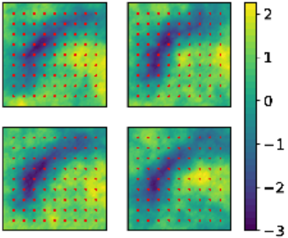
- Infer $\kappa(\mathbf{x})$, discretized with $n = 2601$ parameters; 81 observations; lazy maps of $r \leq 4$ and polynomial degree up to 2



$u(\mathbf{x})$ and observations



Convergence



Posterior realizations of $\kappa(\mathbf{x})$

Summary of Part 1

- **Central idea:** characterize complex/intractable distributions by constructing deterministic *couplings*
- Many kinds of low-dimensional structure (non-exhaustive):
  - Sparse maps, decomposable maps
  - Low rank structure (lazy maps)
- Exploiting the **pullback** distribution
  - Compositions of approximate maps, constructed greedily
  - *(Part 2)* Use approximate maps to *precondition* other sampling or cubature schemes

Extensions and open questions:

- Using sparse grids or QMC for **map construction**
- Zoo of map parameterizations and their approximation properties
- Tail behavior of maps
- Additional varieties of low-dimensional structure: hierarchical, multiscale, tensor, . . .
- Maps from samples:
  - *We will explore this in Part 2*

Summary of Part 1

Extensions and open questions:

- Using sparse grids or QMC for **map construction**
- Zoo of map parameterizations and their approximation properties
- Tail behavior of maps
- Additional varieties of low-dimensional structure: hierarchical, multiscale, tensor, . . .
- Maps from samples:
  - *We will explore this in Part 2*

Thanks for your attention!

References

- A. Spantini, R. Baptista, Y. Marzouk. "Coupling techniques for nonlinear ensemble filtering." arXiv:1907.00389.
- D. Bigoni, O. Zahm, A. Spantini, Y. Marzouk. "Greedy inference with layers of lazy maps." arXiv:1906.00031.
- O. Zahm, T. Cui, K. Law, A. Spantini, Y. Marzouk. "Certified dimension reduction in nonlinear Bayesian inverse problems." arXiv:1807.03712.
- A. Spantini, D. Bigoni, Y. Marzouk. "Inference via low-dimensional couplings." *JMLR* 19(66): 1–71, 2018.
- M. Parno, Y. Marzouk, "Transport map accelerated Markov chain Monte Carlo." *SIAM JUQ* 6: 645–682, 2018.
- G. Detomasso, T. Cui, A. Spantini, Y. Marzouk, R. Scheichl, "A Stein variational Newton method." NeurIPS 2018.
- R. Morrison, R. Baptista, Y. Marzouk. "Beyond normality: learning sparse probabilistic graphical models in the non-Gaussian setting." NeurIPS 2017.
- Y. Marzouk, T. Moselhy, M. Parno, A. Spantini, "An introduction to sampling via measure transport." *Handbook of Uncertainty Quantification*, R. Ghanem, D. Higdon, H. Owhadi, eds. Springer (2016). arXiv:1602.05023.

- **General python code at** `http://transportmaps.mit.edu`