Inference via low-dimensional couplings

Youssef Marzouk
joint work with Alessio Spantini and Daniele Bigoni

Department of Aeronautics and Astronautics
Center for Computational Engineering
Statistics and Data Science Center
Massachusetts Institute of Technology
http://uqgroup.mit.edu

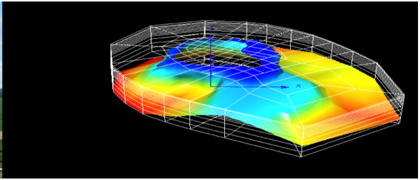
Support from DOE Office of Advanced Scientific Computing Research

1 May 2017

Bayesian inference in large-scale models

Observations y

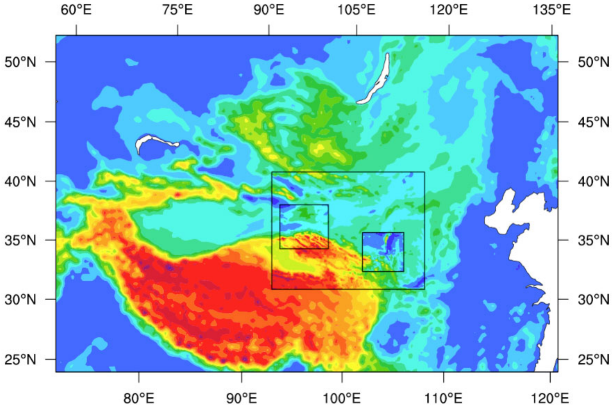
Parameters x



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

- Need to 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 π_{pos} may be **expensive**
- π_{pos} can be evaluated up to a normalizing constant

Sequential Bayesian inference



- State estimation (e.g., *filtering* and *smoothing*) or *joint state and parameter estimation*, in a Bayesian setting
  - Need **recursive, online** algorithms for characterizing the posterior

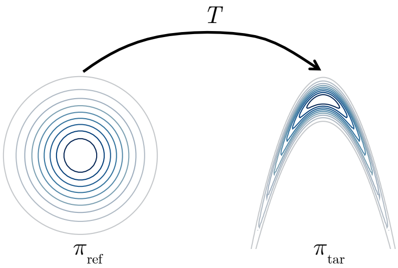
Computational challenges

- 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 strategies for making this computationally tractable
  - Approximations of the forward model, e.g., polynomial approximations, local interpolants, reduced order models, multi-fidelity approaches
  - Efficient and structure-exploiting **sampling (integration) schemes**

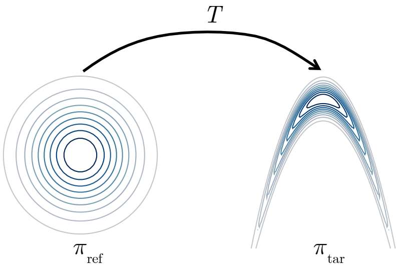
Deterministic coupling of probability measures



Core idea

- Choose π_{ref} (e.g., Gaussian). Set $\pi_{\text{tar}} := \pi_{\text{pos}}$.
- Seek a **transport map** $T : \mathbb{R}^n \to \mathbb{R}^n$ such that $T_\sharp \pi_{\text{ref}} = \pi_{\text{tar}}$

Deterministic coupling of probability measures



Core idea

- Choose π_{ref} (e.g., Gaussian). Set $\pi_{\text{tar}} := \pi_{\text{pos}}$.
- Seek a **transport map** $T : \mathbb{R}^n \to \mathbb{R}^n$ such that $T_\sharp \pi_{\text{ref}} = \pi_{\text{tar}}$
- **Useful outcomes...**
  - *Independent* and *unweighted* samples from the target
  - "Precondition" other sampling or quadrature schemes

- **Optimal transport:**

$$T^{\text{opt}} = \arg\min_T \int_{\mathbb{R}^n} c(x, T(x))\, d\pi_{\text{ref}}(x) \quad \text{s.t. } T_\sharp \pi_{\text{ref}} = \pi_{\text{tar}}$$

  - Monge (1781) problem; many nice properties, but numerically challenging in general continuous cases...

- **Knothe-Rosenblatt rearrangement:**

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

  - Exists and is unique (up to ordering) under mild conditions
  - Jacobian determinant easy to evaluate
  - Monotonicity is essentially one-dimensional: $\partial_{x_k} T^k > 0$

Computation of transports

Variational characterization of the direct map T [Moselhy & M 2012]:

$$\min_{T \in \mathcal{T}_\triangle} \mathcal{D}_{KL}(T_\sharp \pi_{\text{ref}} \,||\, \pi_{\text{tar}})$$

- $\mathcal{T}_\triangle$ is the set of monotone lower **triangular** maps
  - Contains the *Knothe-Rosenblatt* rearrangement
- Expectation is with respect to *reference* measure
  - Compute via, e.g., Monte Carlo, QMC, quadrature
- Use evaluations of π_{tar} (and its gradients) directly; **avoid** MCMC or importance sampling altogether!

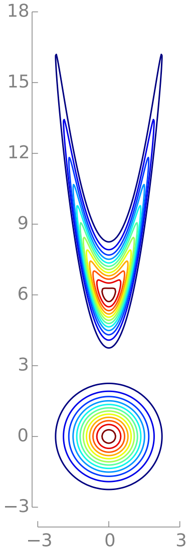
- Parameterize k-th component map $T^k(x)$ with coefficients $\mathbf{f}_k \in \mathbb{R}^{p_k}$
  - *Example:* monotone parameterization, $\partial_{x_k} T^k > 0$:

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

Simple example

$$\min_{\mathbf{f}_1,\ldots,\mathbf{f}_n} \mathbb{E}_{\pi_{\text{ref}}}\left[-\log \pi_{\text{tar}} \circ T - \sum_k \log \partial_{x_k} T^k \right]$$

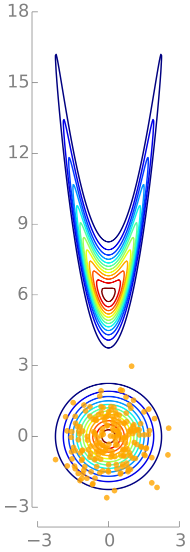
- Parameterized map $T(\mathbf{x}; \mathbf{f}_1, \ldots, \mathbf{f}_n)$
- Optimize over $\mathbf{f}_1, \ldots, \mathbf{f}_n$
- Use gradient-based optimization (here, BFGS)
- Approximate $\mathbb{E}_{\pi_{\text{ref}}}[g] \approx \sum_i w_i\, g(\mathbf{x}_i)$
- The posterior is in the tail of the reference!



Simple example

$$\min_{\mathbf{f}_1,\ldots,\mathbf{f}_n} \mathbb{E}_{\pi_{\mathrm{ref}}}[\, -\log \pi_{\mathrm{tar}} \circ T \; - \sum_k \log \partial_{x_k} T^k \,]$$

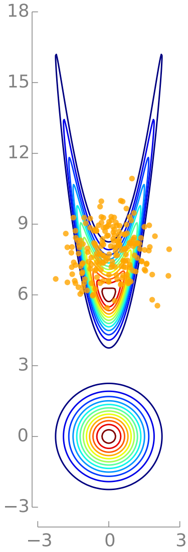
- Parameterized map $T(\,\mathbf{x}\,;\,\mathbf{f}_1,\ldots,\mathbf{f}_n\,)$
- Optimize over $\mathbf{f}_1,\ldots,\mathbf{f}_n$
- Use gradient-based optimization (here, BFGS)
- Approximate $\mathbb{E}_{\pi_{\mathrm{ref}}}[g] \approx \sum_i \, w_i \, g(\mathbf{x}_i)$
- The posterior is in the tail of the reference!



Simple example

$$\min_{\mathbf{f}_1,\ldots,\mathbf{f}_n} \mathbb{E}_{\pi_{\text{ref}}}[-\log \pi_{\text{tar}} \circ T - \sum_k \log \partial_{x_k} T^k]$$

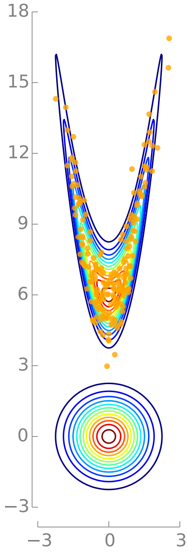
- Parameterized map $T(\mathbf{x}; \mathbf{f}_1, \ldots, \mathbf{f}_n)$
- Optimize over $\mathbf{f}_1, \ldots, \mathbf{f}_n$
- Use gradient-based optimization (here, BFGS)
- Approximate $\mathbb{E}_{\pi_{\text{ref}}}[g] \approx \sum_i w_i \, g(\mathbf{x}_i)$
- The posterior is in the tail of the reference!



Simple example

$$\min_{\mathbf{f}_1,\dots,\mathbf{f}_n} \mathbb{E}_{\pi_{\text{ref}}}[-\log \pi_{\text{tar}} \circ T - \sum_k \log \partial_{x_k} T^k]$$

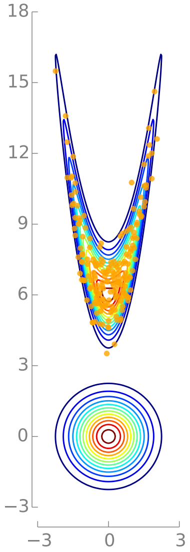
- Parameterized map $T(\mathbf{x}; \mathbf{f}_1, \dots, \mathbf{f}_n)$
- Optimize over $\mathbf{f}_1, \dots, \mathbf{f}_n$
- Use gradient-based optimization (here, BFGS)
- Approximate $\mathbb{E}_{\pi_{\text{ref}}}[g] \approx \sum_i w_i\, g(\mathbf{x}_i)$
- The posterior is in the tail of the reference!



Simple example

$$\min_{\mathbf{f}_1,\dots,\mathbf{f}_n} \mathbb{E}_{\pi_{\mathrm{ref}}}[-\log \pi_{\mathrm{tar}} \circ T - \sum_k \log \partial_{x_k} T^k]$$

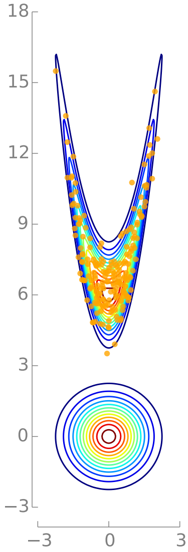
- Parameterized map $T(\mathbf{x}; \mathbf{f}_1, \dots, \mathbf{f}_n)$
- Optimize over $\mathbf{f}_1, \dots, \mathbf{f}_n$
- Use gradient-based optimization (here, BFGS)
- Approximate $\mathbb{E}_{\pi_{\mathrm{ref}}}[g] \approx \sum_i w_i \, g(\mathbf{x}_i)$
- The posterior is in the tail of the reference!



Simple example

Other possible transports:

- Stein variational gradient descent [Liu & Wang 2016]
- Normalizing flows [Rezende & Mohamed 2015]
- Particle flows [Heng *et al.* 2015; Doucet, Daum...]
- Approximations of the optimal transport [Tabak 2013–16]



Potential advantages

$$\min_{\mathbf{f}_1,\ldots,\mathbf{f}_n} \mathbb{E}_{\pi_{\text{ref}}}[-\log \pi_{\text{tar}} \circ T - \sum_{k}^{n} \log \partial_{x_k} T^k]$$

- **Move** samples; don't just reweigh them
- Use **optimization** to enhance **integration**

Potential advantages

$$\min_{\mathbf{f}_1,\ldots,\mathbf{f}_n} \mathbb{E}_{\pi_{\text{ref}}}[\,-\log \pi_{\text{tar}} \circ T \;-\sum_{k}^{n} \log \partial_{x_k} T^k\,]$$

- **Move** samples; don't just reweigh them
- Use **optimization** to enhance **integration**
- *Independent, unweighted,* and *cheap* samples from the target (or close to it): $x_i \sim \pi_{\text{ref}} \Rightarrow T(x_i) \sim \pi_{\text{tar}}$
- Clear convergence criterion, even with unnormalized target density:

$$\mathcal{D}_{KL}(\,T_\sharp\,\pi_{\text{ref}}\,||\,\pi_{\text{tar}}\,) \approx \frac{1}{2}\,\mathbb{V}\text{ar}_{\pi_{\text{ref}}}[\log \pi_{\text{ref}} - \log\, T_\sharp^{-1}\bar{\pi}_{\text{tar}}]$$

- Key steps are embarrassingly parallel

Potential advantages

$$\min_{\mathbf{f}_1,\ldots,\mathbf{f}_n} \mathbb{E}_{\pi_{\text{ref}}}[-\log \pi_{\text{tar}} \circ T - \sum_k^n \log \partial_{x_k} T^k]$$

- **Move** samples; don't just reweigh them
- Use **optimization** to enhance **integration**
- *Independent, unweighted,* and *cheap* samples from the target (or close to it): $x_i \sim \pi_{\text{ref}} \Rightarrow T(x_i) \sim \pi_{\text{tar}}$
- Clear convergence criterion, even with unnormalized target density:

$$\mathcal{D}_{KL}(T_\sharp \pi_{\text{ref}} \,||\, \pi_{\text{tar}}) \approx \frac{1}{2} \mathbb{V}\text{ar}_{\pi_{\text{ref}}}[\log \pi_{\text{ref}} - \log T_\sharp^{-1} \bar{\pi}_{\text{tar}}]$$

- Key steps are embarrassingly parallel
- Yet we exchange a high-dimensional sampling task for a **high-dimensional optimization problem**
  - Major bottleneck: **representation** of the map, e.g., cardinality of the map basis $\mathbf{f}_1, \ldots, \mathbf{f}_n$

Low-dimensional structure

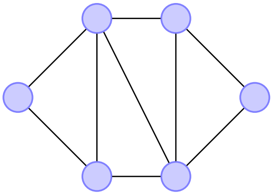
- How to make the construction/representation of **high-dimensional** transports tractable?

- **Key idea: exploit Markov structure of the posterior**
- Leads to various *low-dimensional* properties of transport maps:
  1. Decomposability
  2. Sparsity
  3. Low-rank/near-identity structure

- Property #1 above will yield new *online* algorithms for Bayesian **filtering, smoothing**, and **joint parameter/state estimation**

Markov networks

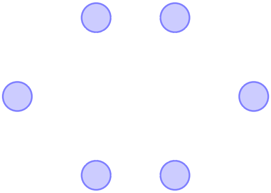
- Let $Z_1, \ldots, Z_n$ be random variables with joint density $\pi > 0$

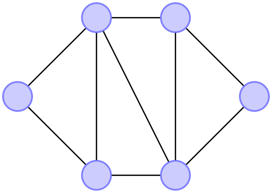


$$(i,j) \notin \mathcal{E} \quad \text{iff} \quad Z_i \perp\!\!\!\perp Z_j \,|\, \mathbf{Z}_{\mathcal{V}\setminus\{i,j\}}$$

- $\mathcal{G}$ encodes conditional independence (I-map for π)
- **Theorem:** Define $\mathcal{G}$ s.t. $(i,j) \notin \mathcal{E}$ if and only if $\partial_{x_i,x_j} \log \pi = 0$
  The resulting $\mathcal{G}$ is the unique minimal I-map for π
- Choice of the **probabilistic model** $\Longrightarrow$ graphical structure

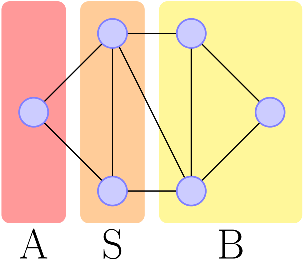
A motivating example





- Fix an **independent** reference density $\eta = \prod_j \eta_{X_j}$ (*left*)
- Seek a transport map $T : \mathbb{R}^6 \to \mathbb{R}^6$ from η to π (*right*)
- Is there a low-dimensional T?
- Yes, but we need two ingredients!
  1. Pullback density $T^\sharp \pi$: if $\mathbf{Z} \sim \pi$, then $T^{-1}(\mathbf{Z}) \sim T^\sharp \pi$
  2. Graph decomposition
- **Remark**: if T were the exact transport, we would have $T^\sharp \pi = \eta$

Graph decomposition

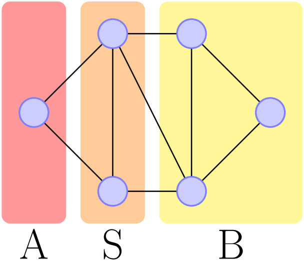


Definition

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

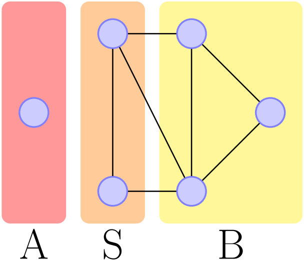
1. $\mathcal{V} = A \cup S \cup B$
2. S separates A from B in $\mathcal{G}$

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 marginal $\eta_{\mathbf{X}_{S \cup A}}$ to $\pi_{\mathbf{X}_{S \cup A}}$
  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 $T_1^\sharp \pi$?

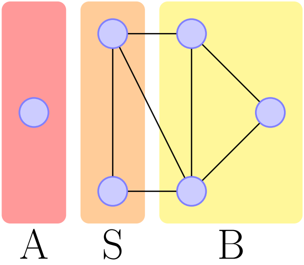
Local graph sparsification



$$\boxed{T = T_1}$$

- **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 $\mathbf{Z}_{S \cup B}$!

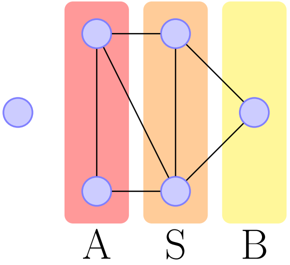
Do it recursively!



$$\boxed{T = T_1}$$

- **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

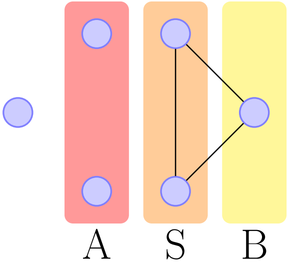
Step k: new decomposition and local map



$$\boxed{T = T_1}$$

- **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

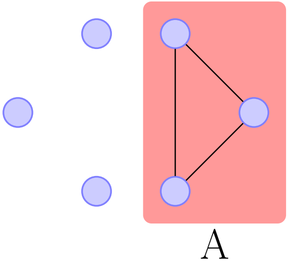
Step k: local graph sparsification



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

- **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...

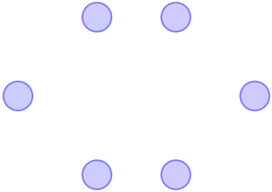
And so on...



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

- **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]

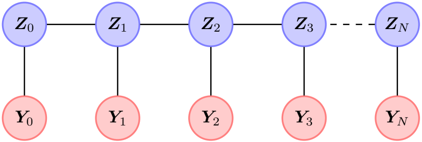
Let $\mathcal{G}$ be an I-map for π and let $\eta = \prod_j \eta_{X_j}$ be a reference density.
If (A, S, B) is a decomposition of $\mathcal{G}$, then

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

Applications to Bayesian filtering/smoothing

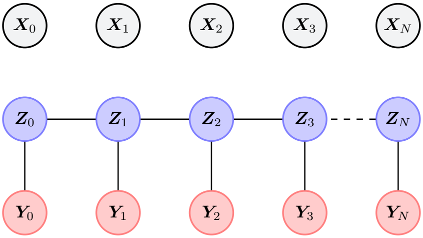
- **Nonlinear non-Gaussian** state-space model: $\pi_{\mathsf{Z}_k|\mathsf{Z}_{k-1}}$, $\pi_{\mathsf{Y}_k|\mathsf{Z}_k}$



- Ideally, interested in **recursively** updating the full Bayesian solution:
  $\pi_{\mathsf{Z}_{0:k}\,|\,\mathsf{Y}_{0:k}} \to \pi_{\mathsf{Z}_{0:k+1}\,|\,\mathsf{Y}_{0:k+1}}$ (more difficult)
- Or focus on approximating the filtering distribution:
  $\pi_{\mathsf{Z}_k\,|\,\mathsf{Y}_{0:k}} \to \pi_{\mathsf{Z}_{k+1}\,|\,\mathsf{Y}_{0:k+1}}$ (marginals of the full Bayesian solution)

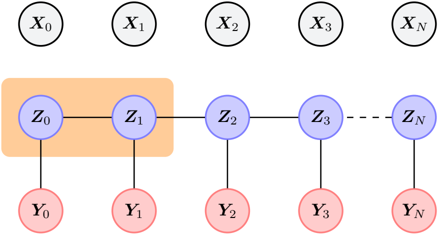
Apply the decomposition theorem to $\pi_{\mathsf{Z}_0,\ldots,\mathsf{Z}_k\,|\,\mathsf{Y}_0,\ldots,\mathsf{Y}_k}$ (just a tree!)

Coupling with an independent process



- Let $\mathbf{X}_0, \mathbf{X}_1, \ldots$ be an independent process with marginals $(\eta_{\mathbf{X}_k})_k$
- Seek a coupling between $\mathbf{X}_0, \ldots, \mathbf{X}_N$ and $\mathbf{Z}_0, \ldots, \mathbf{Z}_N | \mathbf{Y}_0, \ldots, \mathbf{Y}_N$
- Ideally, we would like a low-dimensional decomposable coupling!
- Let's see...

First step: compute a 2-D map



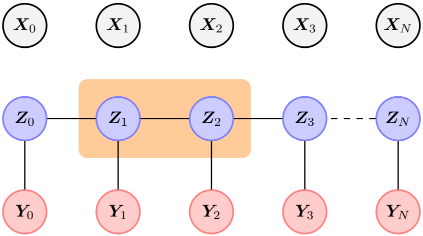
- Compute $\mathfrak{M}_0 : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ s.t.

$$\mathfrak{M}_0(\mathbf{x}_0, \mathbf{x}_1) = \begin{bmatrix} A_0(\mathbf{x}_0, \mathbf{x}_1) \\ B_0(\mathbf{x}_1) \end{bmatrix}$$

- Reference: $\eta_{\mathsf{X}_0} \eta_{\mathsf{X}_1}$
- Target: $\pi_{\mathsf{Z}_0} \, \pi_{\mathsf{Z}_1|\mathsf{Z}_0} \, \pi_{\mathsf{Y}_0|\mathsf{Z}_0} \, \pi_{\mathsf{Y}_1|\mathsf{Z}_1}$
- $\dim(\mathfrak{M}_0) \simeq 2 \times \dim(\mathbf{Z}_0)$

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

Second step: compute a 2-D map



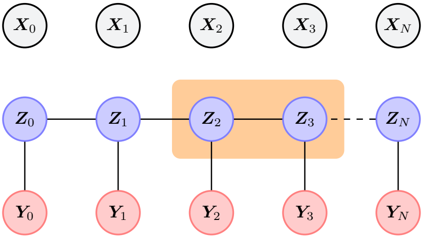
- Compute $\mathfrak{M}_1 : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ s.t.

$$\mathfrak{M}_1(\mathbf{x}_1, \mathbf{x}_2) = \begin{bmatrix} A_1(\mathbf{x}_1, \mathbf{x}_2) \\ B_1(\mathbf{x}_2) \end{bmatrix}$$

- 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



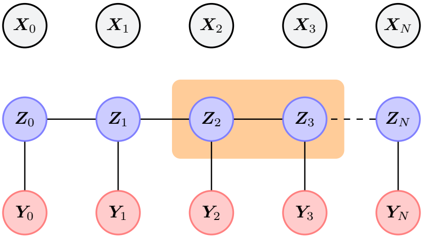
- Compute $\mathfrak{M}_2 : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ s.t.

$$\mathfrak{M}_2(\mathbf{x}_2, \mathbf{x}_3) = \begin{bmatrix} A_2(\mathbf{x}_2, \mathbf{x}_3) \\ B_2(\mathbf{x}_3) \end{bmatrix}$$

- Reference: $\eta_{\mathsf{X}_2} \eta_{\mathsf{X}_3}$
- Target: $\eta_{\mathsf{X}_2}\, \pi_{\mathsf{Y}_3|\mathsf{Z}_3}\, \pi_{\mathsf{Z}_3|\mathsf{Z}_2}(\cdot|\, B_1(\cdot)\,)$
- 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 coupling!

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

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

- $\mathfrak{T}_k$ is dense and high-dimensional but **decomposable**!
- 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)

A single-pass algorithm for online estimation

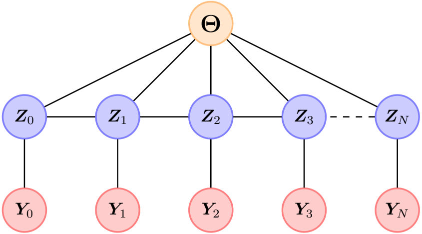
- **Algorithm:**
  1. Compute the maps $\mathfrak{M}_0, \mathfrak{M}_1, \ldots$, each of dimension $2 \times \dim(\mathbf{Z}_0)$
  2. Embed each $\mathfrak{M}_j$ into an identity map to form T_j
  3. Evaluate $T_0 \circ \cdots \circ T_k$ for the full Bayesian solution

- **Remarks**:
  - A single pass on the state-space model
  - Maps $\mathfrak{M}_0, \mathfrak{M}_1, \ldots$ need not be recomputed given new data
  - Constant effort per assimilated observation (online estimation)
  - Variational algorithm: no particles and no particle degeneracy!
  - Of course, we still need to compute each $\mathfrak{M}_j$ (many options)
  - In spirit, a non-Gaussian generalization of the RTS smoother

Full Bayesian solution $\simeq$ lag-1 smoothing (using couplings)

Joint parameter/state estimation

- Can be generalized 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)*
- But now $\dim(\mathfrak{M}_j) = 2 \times \dim(\mathbf{Z}_j) + \dim(\Theta)$
- **Remarks**:
  - Online algorithm (unlike, e.g., particle marginal Metropolis Hastings)
  - No artificial dynamic for the static parameters
  - No *a priori* fixed-lag smoothing approximation

Numerical example: stochastic volatility model

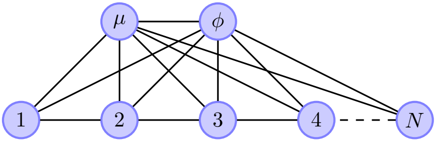
- **Stochastic volatility model:** Latent log-volatilities take the form of an AR(1) process for $t = 1, \ldots, N$:

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

  - 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:

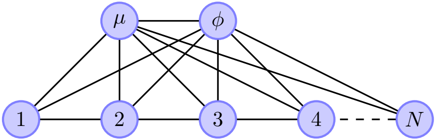


- **Joint state/parameter estimation problem**

Stochastic volatility model with hyperparameters

- Build the decomposition recursively

$$T = \mathbf{Id}$$

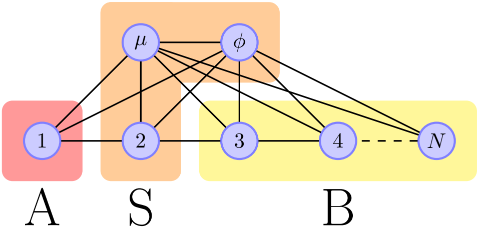


- **Figure:** Markov structure for the pullback of π through T
- Start with the identity map

Stochastic volatility model with hyperparameters

- Build the decomposition recursively

$$T = \textbf{Id}$$

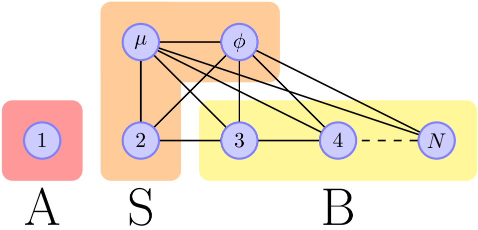


- **Figure:** Markov structure for the pullback of π through T
- Find a good first decomposition of $\mathcal{G}$

Stochastic volatility model with hyperparameters

- Build the decomposition recursively

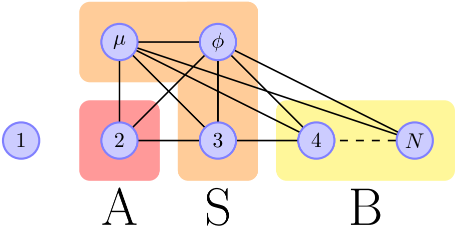
$$T = T_1$$



- **Figure:** Markov structure for the pullback of π through T
- Compute an (essentially) 4-D T_1 and pull back π
- Underlying approximation of $\mu, \phi, \mathbf{Z}_1 | \mathbf{Y}_1$

- Build the decomposition recursively

$$T = T_1$$

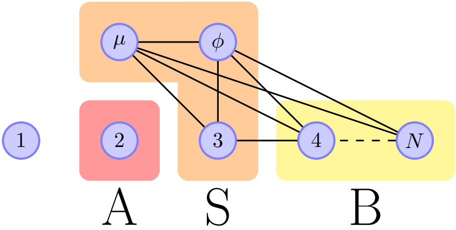


- **Figure:** Markov structure for the pullback of π through T
- Find a new decomposition
- Underlying approximation of $\mu, \phi, \mathbf{Z}_1 | \mathbf{Y}_1$

Stochastic volatility model with hyperparameters

- Build the decomposition recursively

$$T = T_1 \circ T_2$$

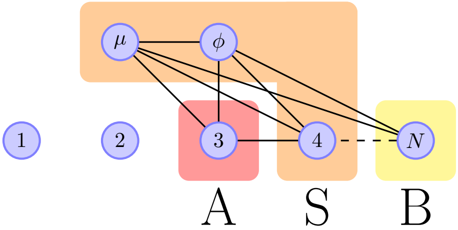


- **Figure:** Markov structure for the pullback of π through T
- Compute an (essentially) 4-D T_2 and pull back π
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:2} | \mathbf{Y}_{1:2}$

Stochastic volatility model with hyperparameters

- Build the decomposition recursively

$$T = T_1 \circ T_2$$

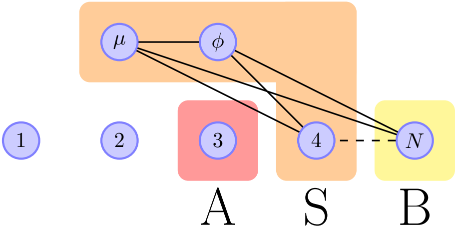


- **Figure:** Markov structure for the pullback of π through T
- Continue the recursion until no edges are left...
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:2} | \mathbf{Y}_{1:2}$

Stochastic volatility model with hyperparameters

- Build the decomposition recursively

$$T = T_1 \circ T_2 \circ T_3$$

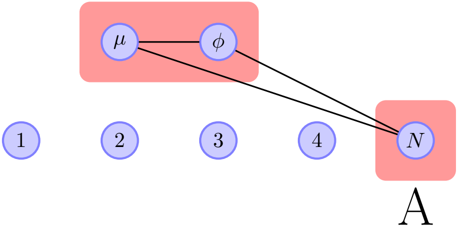


- **Figure:** Markov structure for the pullback of π through T
- Continue the recursion until no edges are left...
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:3}|\mathbf{Y}_{1:3}$

Stochastic volatility model with hyperparameters

- Build the decomposition recursively

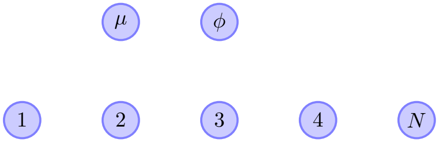
$$T = T_1 \circ T_2 \circ T_3 \circ \cdots \circ T_{N-2}$$



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

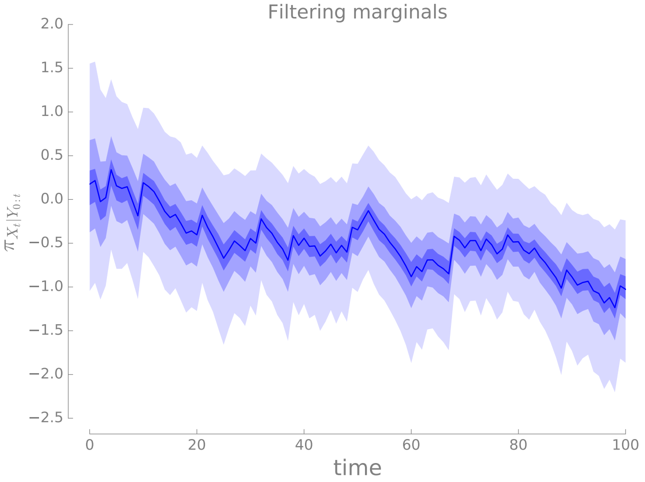
- Build the decomposition recursively

$$T = T_1 \circ T_2 \circ T_3 \circ \cdots \circ T_{N-2} \circ T_{N-1}$$



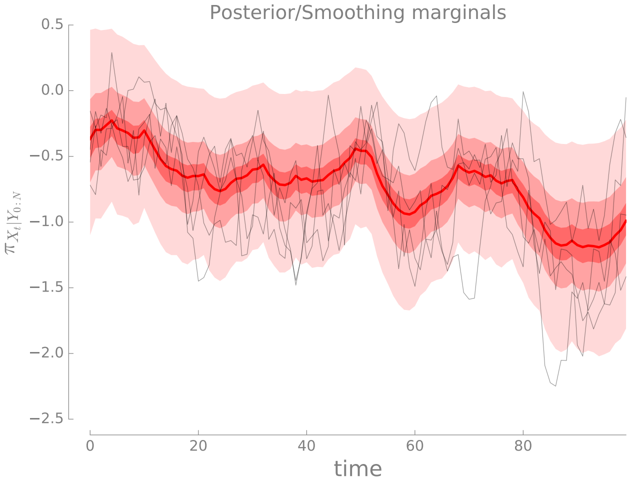
- **Figure:** Markov structure for the pullback of π through T
- Each map T_k is essentially 4-D regardless of N
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:N} | \mathbf{Y}_{1:N}$

Stochastic volatility example (102-dim)



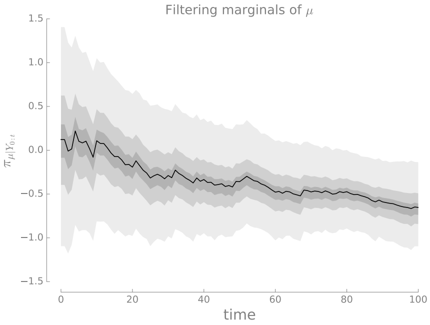
- Joint parameter/state inference problem solved with a single forward pass **[filtering]**

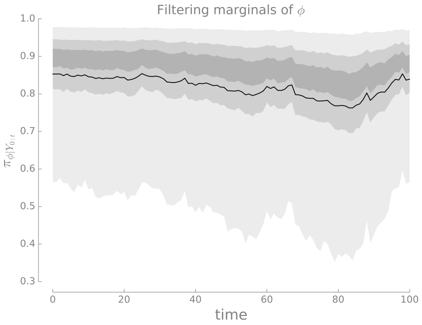
Stochastic volatility example (102-dim)



- Joint parameter/state inference problem solved with a single forward pass, by composing low-dimensional transports **[smoothing]**

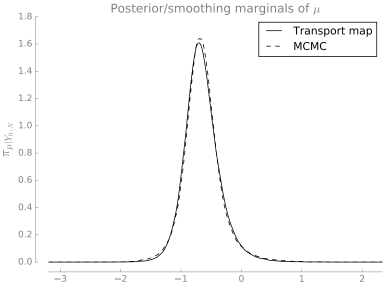
Stochastic volatility example

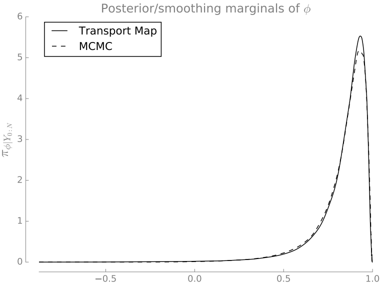




- **Online parameter estimation:** marginals of hyperparameters μ, ϕ, conditioning on successively more observations $\mathbf{y}_{0:k}$

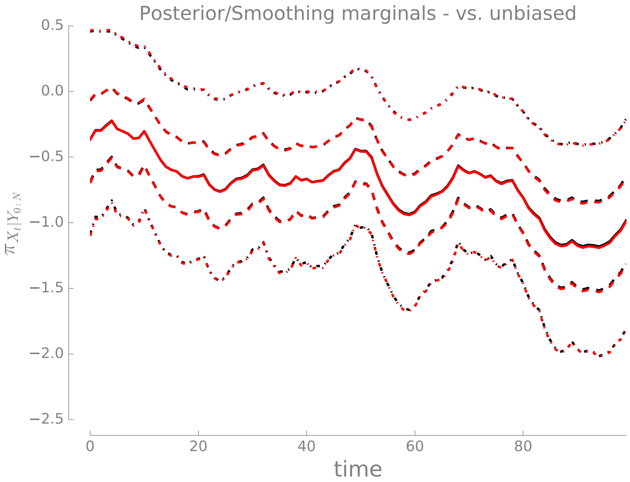
Stochastic volatility example





- Marginals of hyperparameters μ, ϕ: transport maps *(solid)*, MCMC with ESS $= 10^5$ *(dashed)*

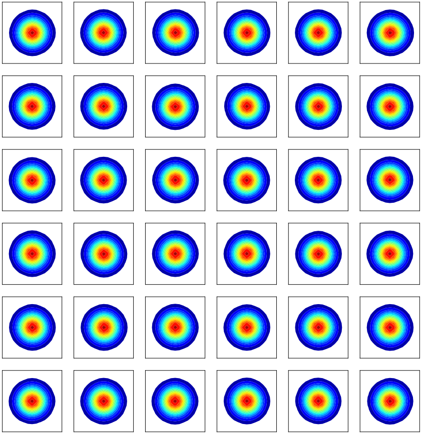
Stochastic volatility example



- Quantiles of smoothing marginals of the state $\mathbf{Z}_{0:N}$ (red) compared to MCMC (black)

Stochastic volatility example

- If $\eta \sim \mathcal{N}(0, \mathbf{I})$ and $T_\sharp \eta = \pi$, then $T^\sharp \pi$ should be **Gaussian**!



- **Figure:** 2-D random conditionals of the pullback density $T^\sharp \pi$
- Variance diagnostic $\approx 8.05 \times 10^{-2}$

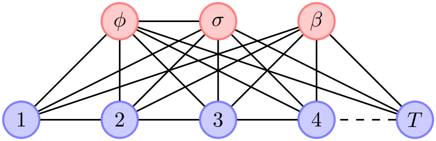
Theorem [Sparsity of triangular transports]

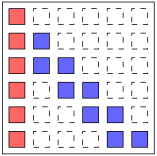
If $\mathcal{G}$ is an I-map for π_{pos}, then we can determine *tight* lower bounds on the sparsity patterns of:

- **Direct** transport $T_\sharp\, \pi_{\text{ref}} = \pi_{\text{pos}}$
- **Inverse** transport $S_\sharp\, \pi_{\text{pos}} = \pi_{\text{ref}}$

only by performing operations on the graph $\mathcal{G}$ (no need to evaluate π_{pos}).

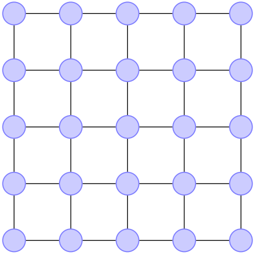
- **Example:** Sparsity of inverse transport $S_\sharp\, \pi_{\text{pos}} = \pi_{\text{ref}}$





- Result: **enforce** sparsity structure in the approximation space $\mathcal{S}_\triangle$, e.g., $\min_{S \in \mathcal{S}_\triangle} \mathcal{D}_{KL}(\, \pi_{\text{ref}} \,||\, S_\sharp\, \pi_{\text{pos}}\,)$

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
- Nonetheless, the notion of **composition of transports** has still potential...

- **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

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

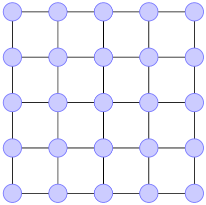
then there exists a rotation Q such that:

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

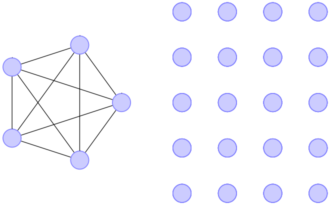
Changing the Markov structure...

- The pullback has a different Markov structure:

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



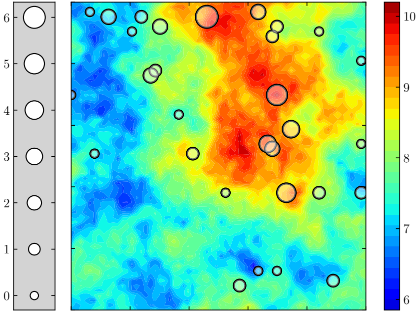
$\mathcal{G}$



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

- **Corollary:** There exists a transport $T_\sharp\, \eta = Q^\sharp\, \widehat{\pi}_{\text{pos}}$ of the form $T(\mathbf{x}) = [\, g(\mathbf{x}_{1:k}),\, x_{k+1}, \dots, x_n \,]$, 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

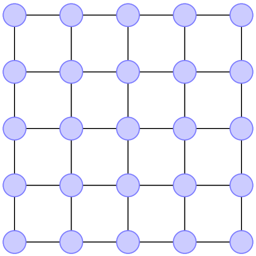


- 4096-D **GMRF prior**, $\mathbf{Z} \sim \mathcal{N}(\mu, \Gamma)$, Γ^{-1} specified through $\triangle + \kappa^2 \,\mathrm{Id}$
- 30 **sparse observations** at locations $i \in \mathcal{I}$, $\mathbf{Y}_i | \mathbf{Z}_i \sim \mathrm{Pois}(\exp \mathbf{Z}_i)$
- Posterior density $\mathbf{Z}|\mathbf{Y}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

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

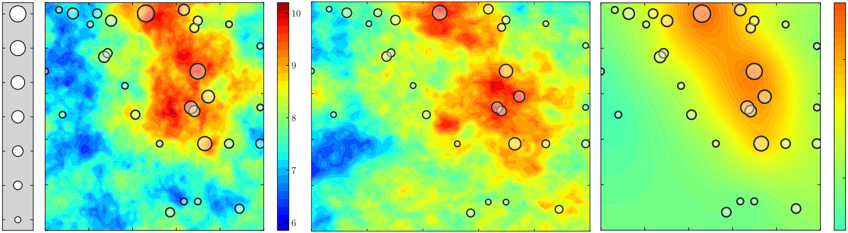
- What is an independence map $\mathcal{G}$ for π_{pos}? A 64×64 grid.

Log-Gaussian Cox process

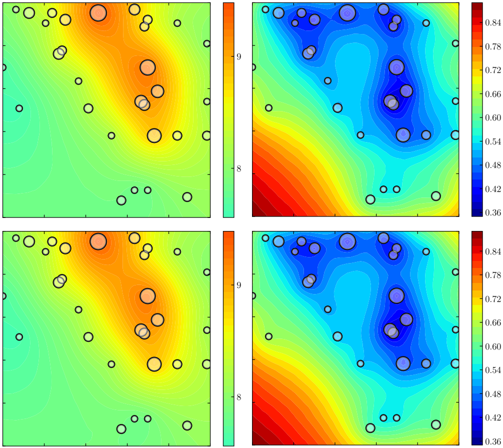
- Fix $\pi_{\text{ref}} \sim \mathcal{N}(0, \mathbf{I})$ and let T_{pr} push forward π_{ref} to π_{pr} (**prior map**)
- Consider the pullback $\widehat{\pi}_{\text{pos}} = T_{\text{pr}}^{\sharp}\, \pi_{\text{pos}}$ and find that

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

- *Deflate* the problem and compute a transport map in **30** dimensions
  - Change from prior to posterior concentrated in a **low-dimensional subspace** (*LIS* Cui, Law, M 2014; *AS* Constantine 2015)



truth　　posterior sample　　posterior mean



- *(left)* $\mathbb{E}[\mathbf{Z}|\mathbf{y}]$, *(right)* $\mathbb{V}\text{ar}[\mathbf{Z}|\mathbf{y}]$. *(top)* transport; *(bottom)* MCMC
- Excellent match with reference MCMC solution, on a problem of $n = 4096$ dimensions

Conclusions

- Bayesian inference through the variational construction of **deterministic couplings**
- Computation of transport maps in high dimensions, leveraging the **Markov structure** of the posterior:
  1. Decomposability of direct transports
     - New online algorithms for **Bayesian filtering, smoothing, and parameter estimation**
  2. Sparsity of triangular transports
  3. Near-identity transports
- Much **ongoing work...**
  - Adaptive parameterizations of monotone maps
  - Nonparametric transports and *gradient flows*
  - *Preconditioning* sparse quadrature and QMC schemes
  - *Approximately sparse* Markov structures

References

- A. Spantini, D. Bigoni, Y. Marzouk. "Inference via low-dimensional couplings." arXiv:1703.06131 **(main reference for this talk)**
- 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.
- A. Spantini, D. Bigoni, Y. Marzouk. "Variational inference via decomposable transports: algorithms for Bayesian filtering and smoothing" and "Adaptive construction of measure transports for Bayesian inference." NIPS workshop on Advances in Approximate Bayesian Inference (2016).
- M. Parno, T. Moselhy, Y. Marzouk, "A multiscale strategy for Bayesian inference using transport maps." *SIAM JUQ*, 4: 1160–1190 (2016).
- M. Parno, Y. Marzouk, "Transport map accelerated Markov chain Monte Carlo." arXiv:1412.5492.
- T. Moselhy, Y. Marzouk, "Bayesian inference with optimal maps." *J. Comp. Phys.*, 231: 7815–7850 (2012).

- Python code just released at `http://transportmaps.mit.edu`