Optimal Transportation Methods in Nonlinear Filtering

Presented at
Nonlinear Dynamics and Controls Lab (NDCL)

Amirhossein Taghvaei

Department of Aeronautics & Astronautics
University of Washington, Seattle

Feb 25, 2022

Outline

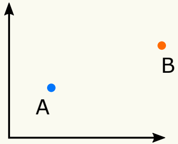
- Introduction to optimal transport
- Application to nonlinear filtering
- Possible collaboration directions

Outline

- Introduction to optimal transport
- Application to nonlinear filtering
- Possible collaboration directions

Brief introduction to optimal transportation (OT)
Geometry for probability distributions

Euclidean geometry



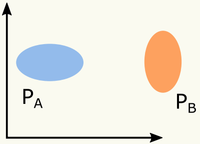
Optimal control problem:

$$\min_u \int_0^1 \|u_t\|^2 \mathrm{d}t$$
$$\text{s.t.} \quad \dot{x}_t = u_t,$$
$$x_0 = A, \; x_1 = B$$

Optimal trajectory gives us:

- how to interpolate
- a metric (Euclidean distance)

OT geometry



Optimal control problem:

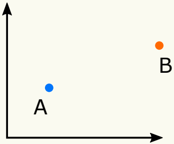
$$\min_U \mathbb{E}\left[\int_0^1 \|U_t\|^2 \mathrm{d}t\right]$$
$$\text{s.t.} \quad \dot{X}_t = U_t,$$
$$X_0 \sim P_A, \; X_1 \sim P_B$$

Optimal trajectory gives us:

- how to interpolate (OT map)
- a metric (Wasserstein distance)

C. Villani, Topics in Optimal Transportation, 2003

Euclidean geometry



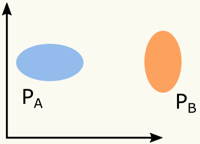
Optimal control problem:

$$\min_u \int_0^1 \|u_t\|^2 \mathrm{d}t$$
$$\text{s.t.} \quad \dot{x}_t = u_t,$$
$$x_0 = A,\ x_1 = B$$

Optimal trajectory gives us:

- how to interpolate
- a metric (Euclidean distance)

OT geometry



Optimal control problem:

$$\min_U \mathbb{E}\left[\int_0^1 \|U_t\|^2 \mathrm{d}t\right]$$
$$\text{s.t.} \quad \dot{X}_t = U_t,$$
$$X_0 \sim P_A,\ X_1 \sim P_B$$

Optimal trajectory gives us:

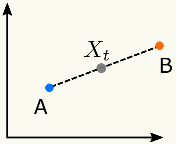
- how to interpolate (OT map)
- a metric (Wasserstein distance)

C. Villani, Topics in Optimal Transportation, 2003

Brief introduction to optimal transportation (OT)

Geometry for probability distributions

Euclidean geometry



Optimal control problem:

$$\min_u \int_0^1 \|u_t\|^2 \mathrm{d}t$$

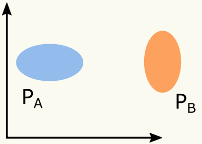
$$\text{s.t.} \quad \dot{x}_t = u_t,$$

$$x_0 = A,\ x_1 = B$$

Optimal trajectory gives us:

- how to interpolate
- a metric (Euclidean distance)

OT geometry



Optimal control problem:

$$\min_U \mathbb{E}\left[\int_0^1 \|U_t\|^2 \mathrm{d}t\right]$$

$$\text{s.t.} \quad \dot{X}_t = U_t,$$

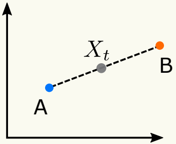
$$X_0 \sim P_A,\ X_1 \sim P_B$$

Optimal trajectory gives us:

- how to interpolate (OT map)
- a metric (Wasserstein distance)

C. Villani, Topics in Optimal Transportation, 2003

Euclidean geometry



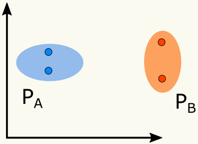
Optimal control problem:

$$\min_{u} \int_0^1 \|u_t\|^2 \mathrm{d}t$$
$$\text{s.t.} \quad \dot{x}_t = u_t,$$
$$x_0 = A,\ x_1 = B$$

Optimal trajectory gives us:

- how to interpolate
- a metric (Euclidean distance)

OT geometry



Optimal control problem:

$$\min_{U} \mathbb{E}\left[\int_0^1 \|U_t\|^2 \mathrm{d}t\right]$$
$$\text{s.t.} \quad \dot{X}_t = U_t,$$
$$X_0 \sim P_A,\ X_1 \sim P_B$$

Optimal trajectory gives us:

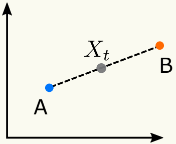
- how to interpolate (OT map)
- a metric (Wasserstein distance)

C. Villani, Topics in Optimal Transportation, 2003

Brief introduction to optimal transportation (OT)

Geometry for probability distributions

Euclidean geometry



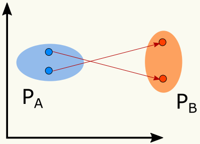
Optimal control problem:

$$\min_u \int_0^1 \|u_t\|^2 \mathrm{d}t$$
$$\text{s.t.} \quad \dot{x}_t = u_t,$$
$$x_0 = A,\ x_1 = B$$

Optimal trajectory gives us:

- how to interpolate
- a metric (Euclidean distance)

OT geometry



Optimal control problem:

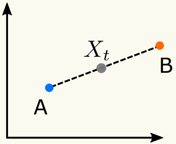
$$\min_U \mathbb{E}\left[\int_0^1 \|U_t\|^2 \mathrm{d}t\right]$$
$$\text{s.t.} \quad \dot{X}_t = U_t,$$
$$X_0 \sim P_A,\ X_1 \sim P_B$$

Optimal trajectory gives us:

- how to interpolate (OT map)
- a metric (Wasserstein distance)

C. Villani, Topics in Optimal Transportation, 2003

Euclidean geometry



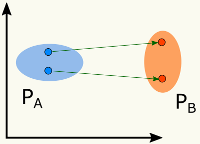
Optimal control problem:

$$\min_u \int_0^1 \|u_t\|^2 \mathrm{d}t$$
$$\text{s.t.} \quad \dot{x}_t = u_t,$$
$$x_0 = A,\ x_1 = B$$

Optimal trajectory gives us:

- how to interpolate
- a metric (Euclidean distance)

OT geometry



Optimal control problem:

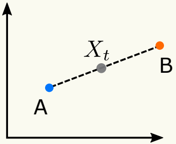
$$\min_U \mathbb{E}\left[\int_0^1 \|U_t\|^2 \mathrm{d}t\right]$$
$$\text{s.t.} \quad \dot{X}_t = U_t,$$
$$X_0 \sim P_A,\ X_1 \sim P_B$$

Optimal trajectory gives us:

- how to interpolate (OT map)
- a metric (Wasserstein distance)

C. Villani, Topics in Optimal Transportation, 2003

Euclidean geometry



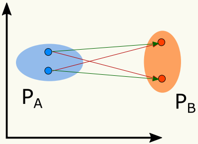
Optimal control problem:

$$\min_u \int_0^1 \|u_t\|^2 \mathrm{d}t$$
$$\text{s.t.} \quad \dot{x}_t = u_t,$$
$$x_0 = A,\ x_1 = B$$

Optimal trajectory gives us:

- how to interpolate
- a metric (Euclidean distance)

OT geometry



Optimal control problem:

$$\min_U \mathbb{E}\left[\int_0^1 \|U_t\|^2 \mathrm{d}t\right]$$
$$\text{s.t.} \quad \dot{X}_t = U_t,$$
$$X_0 \sim P_A,\ X_1 \sim P_B$$

Optimal trajectory gives us:

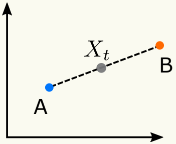
- how to interpolate (OT map)
- a metric (Wasserstein distance)

C. Villani, Topics in Optimal Transportation, 2003

Brief introduction to optimal transportation (OT)

Geometry for probability distributions

Euclidean geometry



Optimal control problem:

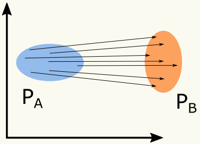
$$\min_{u} \int_0^1 \|u_t\|^2 \mathrm{d}t$$

$$\text{s.t.} \quad \dot{x}_t = u_t, \qquad x_0 = A,\ x_1 = B$$

Optimal trajectory gives us:

- how to interpolate
- a metric (Euclidean distance)

OT geometry



Optimal control problem:

$$\min_{U} \mathbb{E}\left[\int_0^1 \|U_t\|^2 \mathrm{d}t\right]$$

$$\text{s.t.} \quad \dot{X}_t = U_t, \qquad X_0 \sim P_A,\ X_1 \sim P_B$$

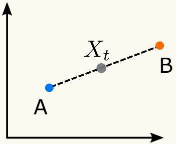
Optimal trajectory gives us:

- how to interpolate (OT map)
- a metric (Wasserstein distance)

C. Villani, Topics in Optimal Transportation, 2003

Brief introduction to optimal transportation (OT)
Geometry for probability distributions

Euclidean geometry



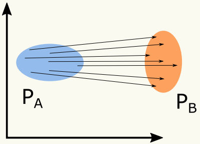
Optimal control problem:

$$\min_{u} \int_0^1 \|u_t\|^2 \mathrm{d}t$$
$$\text{s.t.} \quad \dot{x}_t = u_t, \qquad x_0 = A,\ x_1 = B$$

Optimal trajectory gives us:

- how to interpolate
- a metric (Euclidean distance)

OT geometry



Optimal control problem:

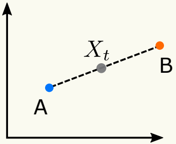
$$\min_{U} \mathbb{E}\left[\int_0^1 \|U_t\|^2 \mathrm{d}t\right]$$
$$\text{s.t.} \quad \dot{X}_t = U_t, \qquad X_0 \sim P_A,\ X_1 \sim P_B$$

Optimal trajectory gives us:

- how to interpolate (OT map)
- a metric (Wasserstein distance)

C. Villani, Topics in Optimal Transportation, 2003

Euclidean geometry



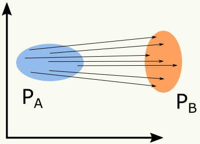
Optimal control problem:

$$\min_u \int_0^1 \|u_t\|^2 \mathrm{d}t$$
$$\text{s.t.} \quad \dot{x}_t = u_t,$$
$$x_0 = A,\ x_1 = B$$

Optimal trajectory gives us:

- how to interpolate
- a metric (Euclidean distance)

OT geometry



Optimal control problem:

$$\min_U \mathbb{E}\left[\int_0^1 \|U_t\|^2 \mathrm{d}t\right]$$
$$\text{s.t.} \quad \dot{X}_t = U_t,$$
$$X_0 \sim P_A,\ X_1 \sim P_B$$

Optimal trajectory gives us:

- how to interpolate (OT map)
- a metric (Wasserstein distance)

C. Villani, Topics in Optimal Transportation, 2003

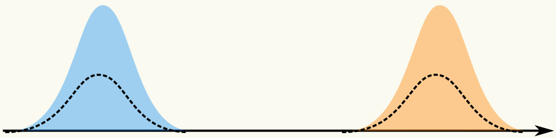
Why is OT geometry useful?

L_2 average vs OT average

- average by modelling as functions: $\frac{p_1(x)+p_2(x)}{2}$
- average by modelling as prob. dist.: OT average
- interpolating between images
- Shape interpolation (Solomon, et. al. 2015)

Why is OT geometry useful?

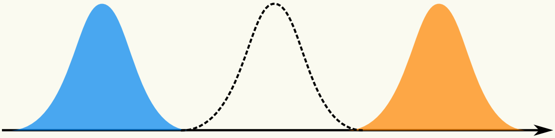
L_2 average vs OT average



- average by modelling as functions: $\dfrac{p_1(x) + p_2(x)}{2}$
- average by modelling as prob. dist.: OT average
- interpolating between images
- Shape interpolation (Solomon, et. al. 2015)

Why is OT geometry useful?

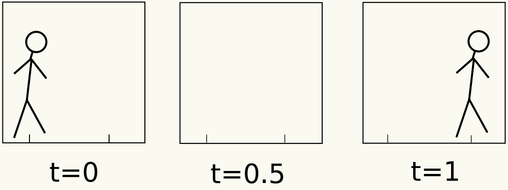
L_2 average vs OT average



- average by modelling as functions: $\frac{p_1(x)+p_2(x)}{2}$
- average by modelling as prob. dist.: OT average
- interpolating between images
- Shape interpolation (Solomon, et. al. 2015)

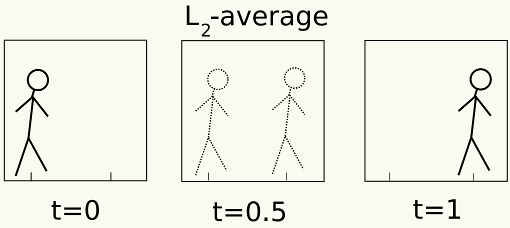
Why is OT geometry useful?

L_2 average vs OT average



- average by modelling as functions: $\frac{p_1(x)+p_2(x)}{2}$
- average by modelling as prob. dist.: OT average
- interpolating between images
- Shape interpolation (Solomon, et. al. 2015)

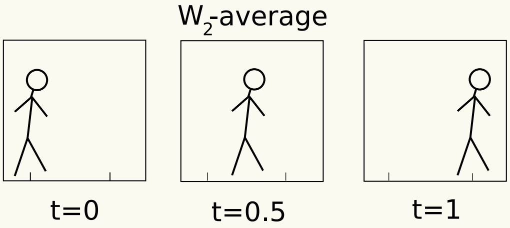
Why is OT geometry useful?
L_2 average vs OT average



- average by modelling as functions: $\frac{p_1(x) + p_2(x)}{2}$
- average by modelling as prob. dist.: OT average
- interpolating between images
- Shape interpolation (Solomon, et. al. 2015)

Why is OT geometry useful?

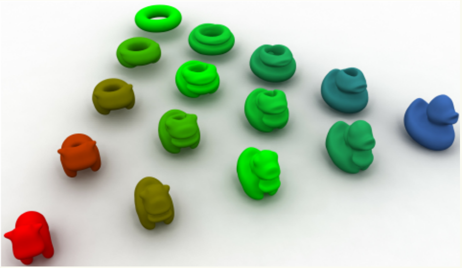
L_2 average vs OT average



- average by modelling as functions: $\dfrac{p_1(x) + p_2(x)}{2}$
- average by modelling as prob. dist.: OT average
- interpolating between images
- Shape interpolation (Solomon, et. al. 2015)

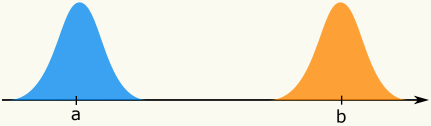
Why is OT geometry useful?

L_2 average vs OT average



- average by modelling as functions: $\dfrac{p_1(x)+p_2(x)}{2}$
- average by modelling as prob. dist.: OT average
- interpolating between images
- Shape interpolation (Solomon, et. al. 2015)

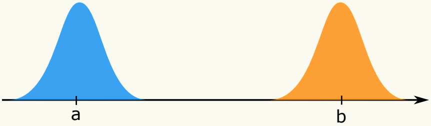
Why is OT geometry useful?
KL divergence vs Wasserstein distance



- Kullback-Leibler divergence: $D(p_1 \| p_2) = \dfrac{(a-b)^2}{\sigma^2}$ $\rightarrow$ diverges as $\sigma \rightarrow 0$
- Wasserstein distance : $W_2(p_1, p_2) = |a - b|$

M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, ICML, 2017

Why is OT geometry useful?
KL divergence vs Wasserstein distance



- Kullback-Leibler divergence: $D(p_1 \| p_2) = \frac{(a-b)^2}{\sigma^2}$ $\rightarrow$ diverges as $\sigma \rightarrow 0$
- Wasserstein distance : $W_2(p_1, p_2) = |a-b|$

M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, ICML, 2017

Why is OT geometry useful?

KL divergence vs Wasserstein distance

- Kullback-Leibler divergence: $D(p_1 \| p_2) = \dfrac{(a-b)^2}{\sigma^2}$ $\rightarrow$ diverges as $\sigma \rightarrow 0$
- Wasserstein distance : $W_2(p_1, p_2) = |a - b|$

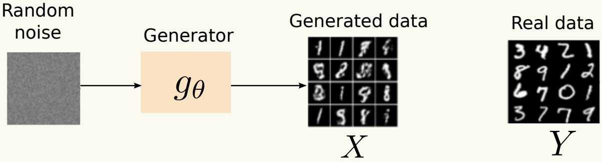
M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, ICML, 2017

Why is OT geometry useful?

KL divergence vs Wasserstein distance

- Kullback-Leibler divergence: $D(p_1 \| p_2) = \dfrac{(a-b)^2}{\sigma^2}$ $\rightarrow$ diverges as $\sigma \rightarrow 0$
- Wasserstein distance : $W_2(p_1, p_2) = |a - b|$

Application: Learning generative models (GAN)



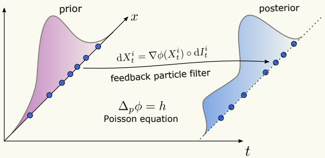
make the distribution of X close to Y: $\min_g \; W(P_X, P_Y)$

M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, ICML, 2017

Research overveiw

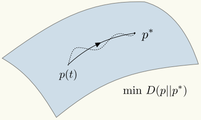
Control & Optimization for probability distributions

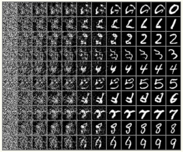
(I) Optimal filtering & control



- Optimal transportation methods in nonlinear filtering: The feedback particle filter, CSM, 2021
- An optimal transport formulation of the ensemble Kalman filter, TAC, 2021

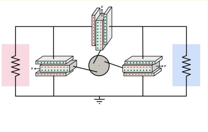
(II) Machine learning

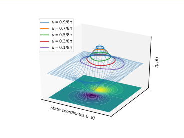




- OT mapping via input-convex neural networks, ICML, 2020
- Scalable computations of Wasserstein barycenter via input convex neural networks, ICML, 2021
- Variational Wasserstein gradient flow, Submitted to ICML, 2022

(III) Stochastic thermodynamics





- Energy harvesting from anisotropic fluctuations, PRE, 2021
- On the relation between information and power in stochastic thermodynamic engines, (L-CSS), 2021
- Maximal power output of a stochastic thermodynamic engine, Automatica, 2021

Common objectives:

- develop efficient and scalable algorithms
- understand fundamental limitations

Theoretical tools:

- optimal transportation
- mean-field optimal control
- statistical learning

Outline

- Introduction to optimal transport
- Application to nonlinear filtering
- Possible collaboration directions

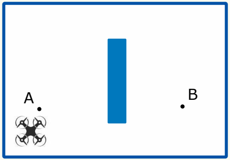
Outline

- Introduction to optimal transport
- Application to nonlinear filtering
- Possible collaboration directions

Motivation
Uncertainties are integral to control systems

Sources of uncertainty:

- incomplete model (e.g. disturbances)
- noisy sensors (e.g. LIDAR, IMU)
- augmenting stochastic ML modules
  (e.g. data driven models, object detection)



It is necessary to quantify uncertainty for:

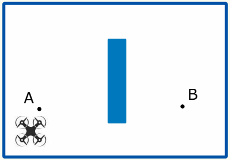
- reliable prediction (e.g. to avoid collision)
- decision making and planning under uncertainty

Nonlinear filtering is a principled approach to quantify uncertainty

Motivation
Uncertainties are integral to control systems

Sources of uncertainty:

- incomplete model (e.g. disturbances)
- noisy sensors (e.g. LIDAR, IMU)
- augmenting stochastic ML modules
  (e.g. data driven models, object detection)



It is necessary to quantify uncertainty for:

- reliable prediction (e.g. to avoid collision)
- decision making and planning under uncertainty

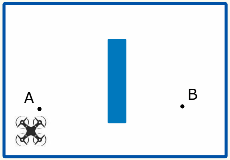
Nonlinear filtering is a principled approach to quantify uncertainty

Motivation

Uncertainties are integral to control systems

Sources of uncertainty:

- incomplete model (e.g. disturbances)
- noisy sensors (e.g. LIDAR, IMU)
- augmenting stochastic ML modules
  (e.g. data driven models, object detection)



It is necessary to quantify uncertainty for:

- reliable prediction (e.g. to avoid collision)
- decision making and planning under uncertainty

$\boxed{\text{Nonlinear filtering is a principled approach to quantify uncertainty}}$

Nonlinear filtering problem
Mathematical model

$$\begin{tikzcd} {} \arrow[r] & X_{k-1} \arrow[r] \arrow[d] & X_k \arrow[r] \arrow[d] & X_{k+1} \arrow[r] \arrow[d] & {} \\ & Y_{k-1} & Y_k & Y_{k+1} & \end{tikzcd}$$

- X_k is the state (unknown)
- Y_k is the observation
- dynamic and observation model are given

Questions: Given history of observation $Y_{1:k} := \{Y_1, \ldots, Y_k\}$,

- What is the most likely value of X_k?
- What is the probability of $X_k \in A$?
- What is the best m.s.e estimate for X_k?
- ...

Answer: given by the conditional distribution $\pi_k = \mathsf{P}(X_k | Y_{1:k})$ (posterior, belief)

J. Xiong, An introduction to stochastic filtering theory, 2008

Nonlinear filtering problem
Mathematical model

$$
\begin{tikzcd}
{} \arrow[r] & X_{k-1} \arrow[r] \arrow[d] & X_k \arrow[r] \arrow[d] & X_{k+1} \arrow[r] \arrow[d] & {} \\
& Y_{k-1} & Y_k & Y_{k+1} &
\end{tikzcd}
$$

- X_k is the state (unknown)
- Y_k is the observation
- dynamic and observation model are given

Questions: Given history of observation $Y_{1:k} := \{Y_1, \ldots, Y_k\}$,

- What is the most likely value of X_k?
- What is the probability of $X_k \in A$?
- What is the best m.s.e estimate for X_k?
- ...

Answer: given by the conditional distribution $\pi_k = \mathsf{P}(X_k | Y_{1:k})$ (posterior, belief)

J. Xiong, An introduction to stochastic filtering theory, 2008

Nonlinear filtering problem
Mathematical model

$$
\begin{tikzcd}
{} \arrow[r] & X_{k-1} \arrow[r] \arrow[d] & X_k \arrow[r] \arrow[d] & X_{k+1} \arrow[r] \arrow[d] & {} \\
& Y_{k-1} & Y_k & Y_{k+1} &
\end{tikzcd}
$$

- X_k is the state (unknown)
- Y_k is the observation
- dynamic and observation model are given

Questions: Given history of observation $Y_{1:k} := \{Y_1, \ldots, Y_k\}$,

- What is the most likely value of X_k? $\arg\max_x \mathsf{P}(X_k = x | Y_{1:k})$
- What is the probability of $X_k \in A$? $\int_A \mathsf{P}(X_k = x | Y_{1:k}) \mathrm{d}x$
- What is the best m.s.e estimate for X_k? $\int x P(X_k = x | Y_{1:k}) \mathrm{d}x$
- . . .

Answer: given by the conditional distribution $\pi_k = \mathsf{P}(X_k | Y_{1:k})$ (posterior, belief)

J. Xiong, An introduction to stochastic filtering theory, 2008

Recursive update for posterior

Given $\pi_k = \mathsf{P}(X_k|Y_{1:k})$, obtain $\pi_{k+1} = \mathsf{P}(X_{k+1}|Y_{1:k+1})$ according to

- Step 1: dynamics update

$$\mathsf{P}(X_k|Y_{1:k}) \xrightarrow{\text{dynamics}} \mathsf{P}(X_{k+1}|Y_{1:k})$$

- Step 2: Bayes law

$$\mathsf{P}(X_{k+1}|Y_{1:k+1}) = \frac{P(Y_{k+1}|X_{k+1})P(X_{k+1}|Y_{1:k}))}{P(Y_{k+1}|Y_{1:k})}$$

As a result:

- π_k has Markov property
- stores all information up to k

Nonlinear filtering algorithm:

- numerical representation of π_k
- numerical implementation of the update law

Recursive update for posterior

Given $\pi_k = \mathsf{P}(X_k|Y_{1:k})$, obtain $\pi_{k+1} = \mathsf{P}(X_{k+1}|Y_{1:k+1})$ according to

- Step 1: dynamics update

$$\mathsf{P}(X_k|Y_{1:k}) \xrightarrow{\text{dynamics}} \mathsf{P}(X_{k+1}|Y_{1:k})$$

- Step 2: Bayes law

$$\mathsf{P}(X_{k+1}|Y_{1:k+1}) = \frac{P(Y_{k+1}|X_{k+1})P(X_{k+1}|Y_{1:k}))}{P(Y_{k+1}|Y_{1:k})}$$

As a result:

- π_k has Markov property
- stores all information up to k

Nonlinear filtering algorithm:

- numerical representation of π_k
- numerical implementation of the update law

Recursive update for posterior

Given $\pi_k = \mathsf{P}(X_k|Y_{1:k})$, obtain $\pi_{k+1} = \mathsf{P}(X_{k+1}|Y_{1:k+1})$ according to

- Step 1: dynamics update

$$\mathsf{P}(X_k|Y_{1:k}) \xrightarrow{\text{dynamics}} \mathsf{P}(X_{k+1}|Y_{1:k})$$

- Step 2: Bayes law

$$\mathsf{P}(X_{k+1}|Y_{1:k+1}) = \frac{P(Y_{k+1}|X_{k+1})P(X_{k+1}|Y_{1:k}))}{P(Y_{k+1}|Y_{1:k})}$$

As a result:

- π_k has Markov property
- stores all information up to k

Nonlinear filtering algorithm:

- numerical representation of π_k
- numerical implementation of the update law

Recursive update for posterior

Given $\pi_k = \mathsf{P}(X_k|Y_{1:k})$, obtain $\pi_{k+1} = \mathsf{P}(X_{k+1}|Y_{1:k+1})$ according to

- Step 1: dynamics update

$$\mathsf{P}(X_k|Y_{1:k}) \xrightarrow{\text{dynamics}} \mathsf{P}(X_{k+1}|Y_{1:k})$$

- Step 2: Bayes law

$$\mathsf{P}(X_{k+1}|Y_{1:k+1}) = \frac{P(Y_{k+1}|X_{k+1})P(X_{k+1}|Y_{1:k}))}{P(Y_{k+1}|Y_{1:k})}$$

As a result:

$$\longrightarrow \pi_{k-1} \longrightarrow \pi_k \longrightarrow \pi_{k+1} \longrightarrow$$

- π_k has Markov property
- stores all information up to k

Nonlinear filtering algorithm:

- numerical representation of π_k
- numerical implementation of the update law

Recursive update for posterior

Given $\pi_k = \mathsf{P}(X_k|Y_{1:k})$, obtain $\pi_{k+1} = \mathsf{P}(X_{k+1}|Y_{1:k+1})$ according to

- Step 1: dynamics update

$$\mathsf{P}(X_k|Y_{1:k}) \xrightarrow{\text{dynamics}} \mathsf{P}(X_{k+1}|Y_{1:k})$$

- Step 2: Bayes law

$$\mathsf{P}(X_{k+1}|Y_{1:k+1}) = \frac{P(Y_{k+1}|X_{k+1})P(X_{k+1}|Y_{1:k}))}{P(Y_{k+1}|Y_{1:k})}$$

As a result:

$$\longrightarrow \pi_{k-1} \longrightarrow \pi_k \longrightarrow \pi_{k+1} \longrightarrow$$

- π_k has Markov property
- stores all information up to k

Nonlinear filtering algorithm:

- numerical representation of π_k
- numerical implementation of the update law

Kalman filter
Linear Gaussian setting

- linear dynamics: $X_{k+1} = AX_k + Be_k$
- linear observation: $Z_k = HX_K + W_k$

Kalman filter: posterior π_k is Gaussian $N(m_k, \Sigma_k)$

$$\text{Update for mean:} \quad m_{k+1} = \underbrace{Am_k}_{\text{dynamics}} + \underbrace{\mathsf{K}_k(Z_k - Hm_k)}_{\text{correction}}$$

$$\text{Update for variance:} \quad \Sigma_{k+1} = \text{(Ricatti equation)}$$

- application in navigation and guidance
- fails to represent multi-modal distributions $\rightarrow$ particle filters

R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, 1960
R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory, 1961

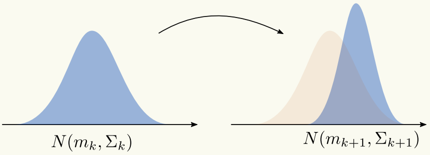
Kalman filter
Linear Gaussian setting

- linear dynamics: $X_{k+1} = AX_k + Be_k$
- linear observation: $Z_k = HX_K + W_k$

Kalman filter: posterior π_k is Gaussian $N(m_k, \Sigma_k)$

Update for mean: $m_{k+1} = \underbrace{Am_k}_{\text{dynamics}} + \underbrace{\mathsf{K}_k(Z_k - Hm_k)}_{\text{correction}}$

Update for variance: $\Sigma_{k+1} =$ (Ricatti equation)



- application in navigation and guidance
- fails to represent multi-modal distributions $\rightarrow$ particle filters

R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, 1960
R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory, 1961

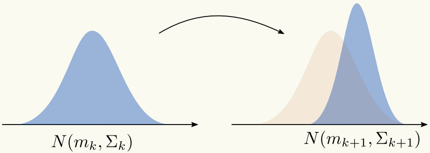
Kalman filter
Linear Gaussian setting

- linear dynamics: $X_{k+1} = AX_k + Be_k$
- linear observation: $Z_k = HX_K + W_k$

Kalman filter: posterior π_k is Gaussian $N(m_k, \Sigma_k)$

Update for mean: $m_{k+1} = \underbrace{Am_k}_{\text{dynamics}} + \underbrace{\mathsf{K}_k(Z_k - Hm_k)}_{\text{correction}}$

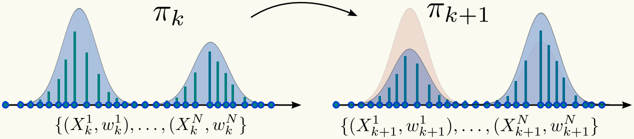
Update for variance: $\Sigma_{k+1} =$ (Ricatti equation)



- application in navigation and guidance
- fails to represent multi-modal distributions $\to$ particle filters

R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, 1960
R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory, 1961

- Approximate π_k with weighted empirical distribution of particles
- Apply the update rule to the particles and weights



- Step 1: update particles according to dynamics
- Step 2: update the weights according to Bayes rule

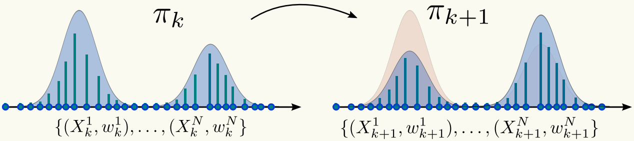
$$w_{k+1}^i \propto w_k^i P(Z_{k+1}|X_{k+1}^i)$$

Properties:

- exact in the limit as $N \to \infty$
- weight degeneracy $\to$ curse of dimensionality

N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation (1993).
P. Del Moral, A.Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. (2001)
A. Doucet and A. Johansen, A Tutorial on Particle Filtering and Smoothing: Fifteen years later (2008).
P. Bickel, B. Li, and T. Bengtsson, Sharp failure rates for the bootstrap particle filter in high dimensions (2008).

- Approximate π_k with weighted empirical distribution of particles
- Apply the update rule to the particles and weights



- Step 1: update particles according to dynamics
- Step 2: update the weights according to Bayes rule

$$w_{k+1}^i \propto w_k^i P(Z_{k+1}|X_{k+1}^i)$$

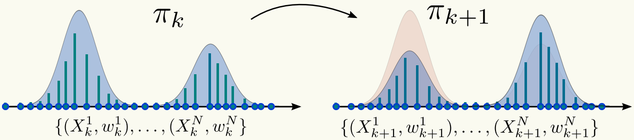
Properties:

- exact in the limit as $N \to \infty$
- weight degeneracy $\to$ curse of dimensionality

N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation (1993).
P. Del Moral, A.Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. (2001)
A. Doucet and A. Johansen, A Tutorial on Particle Filtering and Smoothing: Fifteen years later (2008).
P. Bickel, B. Li, and T. Bengtsson, Sharp failure rates for the bootstrap particle filter in high dimensions (2008).

Particle filters
Monte-Carlo approximation

- Approximate π_k with weighted empirical distribution of particles
- Apply the update rule to the particles and weights



- Step 1: update particles according to dynamics
- Step 2: update the weights according to Bayes rule

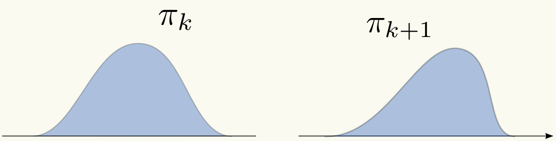
$$w_{k+1}^i \propto w_k^i P(Z_{k+1}|X_{k+1}^i)$$

Properties:

- exact in the limit as $N \to \infty$
- weight degeneracy $\to$ curse of dimensionality

N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation (1993).
P. Del Moral, A.Guionnet. On the stability of interacting processes with applications to filtering and genetic algorithms. (2001)
A. Doucet and A. Johansen, A Tutorial on Particle Filtering and Smoothing: Fifteen years later (2008).
P. Bickel, B. Li, and T. Bengtsson, Sharp failure rates for the bootstrap particle filter in high dimensions (2008).

Optimal transportation viewpoint



- approximate π_k with empirical distribution of particles (uniform weights)
- main task of a particle-based algorithm:

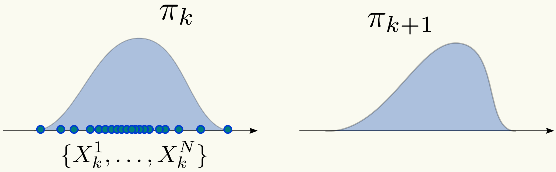
$$\text{given:} \quad \{X_k^1, \ldots, X_k^N\} \sim \pi_k$$
$$\text{generate:} \quad \{X_{k+1}^1, \ldots, X_{k+1}^N\} \sim \pi_{k+1}$$

OT approach: update particles with the optimal transport map from π_k to π_{k+1}

$$X_{k+1}^i = T_k(X_k^i)$$

Question: How to numerically approximate the optimal transport map T_k?

Optimal transportation viewpoint



- approximate π_k with empirical distribution of particles (uniform weights)
- main task of a particle-based algorithm:

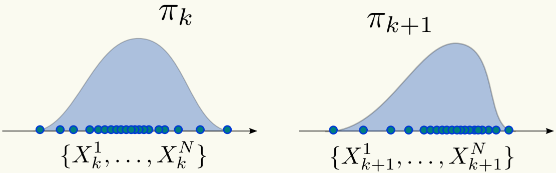
$$\text{given:} \quad \{X_k^1, \ldots, X_k^N\} \sim \pi_k$$
$$\text{generate:} \quad \{X_{k+1}^1, \ldots, X_{k+1}^N\} \sim \pi_{k+1}$$

OT approach: update particles with the optimal transport map from π_k to π_{k+1}

$$X_{k+1}^i = T_k(X_k^i)$$

Question: How to numerically approximate the optimal transport map T_k?

Optimal transportation viewpoint



- approximate π_k with empirical distribution of particles (uniform weights)
- main task of a particle-based algorithm:

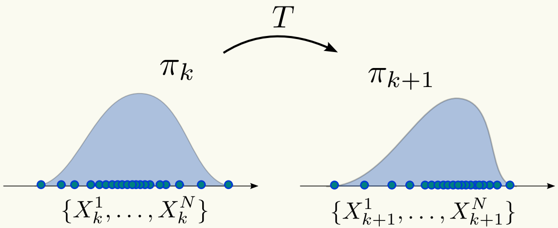
$$\text{given:} \quad \{X_k^1, \ldots, X_k^N\} \sim \pi_k$$
$$\text{generate:} \quad \{X_{k+1}^1, \ldots, X_{k+1}^N\} \sim \pi_{k+1}$$

OT approach: update particles with the optimal transport map from π_k to π_{k+1}

$$X_{k+1}^i = T_k(X_k^i)$$

Question: How to numerically approximate the optimal transport map T_k?

Optimal transportation viewpoint



- approximate π_k with empirical distribution of particles (uniform weights)
- main task of a particle-based algorithm:

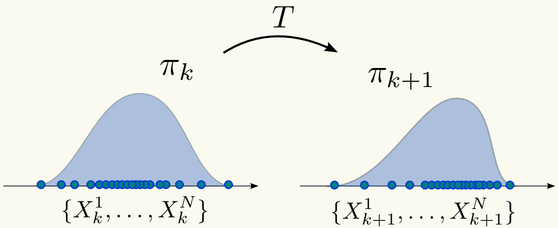
$$\text{given:} \quad \{X_k^1, \dots, X_k^N\} \sim \pi_k$$
$$\text{generate:} \quad \{X_{k+1}^1, \dots, X_{k+1}^N\} \sim \pi_{k+1}$$

OT approach: update particles with the optimal transport map from π_k to π_{k+1}

$$X_{k+1}^i = T_k(X_k^i)$$

Question: How to numerically approximate the optimal transport map T_k?

Optimal transportation viewpoint



- approximate π_k with empirical distribution of particles (uniform weights)
- main task of a particle-based algorithm:

$$\text{given:} \quad \{X_k^1, \ldots, X_k^N\} \sim \pi_k$$
$$\text{generate:} \quad \{X_{k+1}^1, \ldots, X_{k+1}^N\} \sim \pi_{k+1}$$

OT approach: update particles with the optimal transport map from π_k to π_{k+1}

$$X_{k+1}^i = T_k(X_k^i)$$

Question: How to numerically approximate the optimal transport map T_k?

Optimal transport particle filter

Variational approach: OT map is approximated by solving an optimization problem

$$T(x) = \nabla_x f(x, Y_k), \quad \text{where} \quad f = \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_k^i, Y_k^i))$$

- this is a stochastic optimization problem
- $\mathcal{F}$ is a subset of functions
- approximation becomes exact when $N = \infty$ and $\mathcal{F}$ is all functions

Computational properties:

Optimal transport particle filter

Variational approach: OT map is approximated by solving an optimization problem

$$T(x) = \nabla_x f(x, Y_k), \quad \text{where} \quad f = \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_k^i, Y_k^i))$$

- this is a stochastic optimization problem
- $\mathcal{F}$ is a subset of functions
- approximation becomes exact when $N = \infty$ and $\mathcal{F}$ is all functions

Computational properties:

Optimal transport particle filter

Variational approach: OT map is approximated by solving an optimization problem

$$T(x) = \nabla_x f(x, Y_k), \quad \text{where} \quad f = \arg\min_{f \in \mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} \ell(f(X_k^i, Y_k^i))$$

- this is a stochastic optimization problem
- $\mathcal{F}$ is a subset of functions
- approximation becomes exact when $N = \infty$ and $\mathcal{F}$ is all functions

Computational properties:

Optimal transport particle filter

Variational approach: OT map is approximated by solving an optimization problem

$$T(x) = \nabla_x f(x, Y_k), \quad \text{where} \quad f = \underset{f \in \mathcal{F}}{\arg\min} \ \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_k^i, Y_k^i))$$

- this is a stochastic optimization problem
- $\mathcal{F}$ is a subset of functions
- approximation becomes exact when $N = \infty$ and $\mathcal{F}$ is all functions

Computational properties:

Optimal transport particle filter

Variational approach: OT map is approximated by solving an optimization problem

$$T(x) = \nabla_x f(x, Y_k), \quad \text{where} \quad f = \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_k^i, Y_k^i))$$

- this is a stochastic optimization problem
- $\mathcal{F}$ is a subset of functions
- approximation becomes exact when $N = \infty$ and $\mathcal{F}$ is all functions

Computational properties:

- Only requires a simulator for dynamics and observation (explicit model not needed)
- Allows application of ML tools (stochastic optimization and neural nets)
- Challenge: solving a non-convex optimization problem on function space

Optimal transport particle filter

Variational approach: OT map is approximated by solving an optimization problem

$$T(x) = \nabla_x f(x, Y_k), \quad \text{where} \quad f = \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_k^i, Y_k^i))$$

- this is a stochastic optimization problem
- $\mathcal{F}$ is a subset of functions
- approximation becomes exact when $N = \infty$ and $\mathcal{F}$ is all functions

Computational properties:

- Only requires a simulator for dynamics and observation (explicit model not needed)
- Allows application of ML tools (stochastic optimization and neural nets)
- Challenge: solving a non-convex optimization problem on function space

Optimal transport particle filter

Variational approach: OT map is approximated by solving an optimization problem

$$T(x) = \nabla_x f(x, Y_k), \quad \text{where} \quad f = \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_k^i, Y_k^i))$$

- this is a stochastic optimization problem
- $\mathcal{F}$ is a subset of functions
- approximation becomes exact when $N = \infty$ and $\mathcal{F}$ is all functions

Computational properties:

- Only requires a simulator for dynamics and observation (explicit model not needed)
- Allows application of ML tools (stochastic optimization and neural nets)
- Challenge: solving a non-convex optimization problem on function space

Optimal transport particle filter

Variational approach: OT map is approximated by solving an optimization problem

$$T(x) = \nabla_x f(x, Y_k), \quad \text{where} \quad f = \arg\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_k^i, Y_k^i))$$

- this is a stochastic optimization problem
- $\mathcal{F}$ is a subset of functions
- approximation becomes exact when $N = \infty$ and $\mathcal{F}$ is all functions

Computational properties:

- Only requires a simulator for dynamics and observation (explicit model not needed)
- Allows application of ML tools (stochastic optimization and neural nets)
- Challenge: solving a non-convex optimization problem on function space

Question: Can this approach mitigate the curse of dimensionality?

Curse of dimensionality
Linear Gaussian setting

- Method 1: OT approach with $\mathcal{F} = \{$quadratic functions$\}$ →Ensemble Kalman filter
- Method 2: standard particle filter
- Compare number of particles to achieve error ϵ:

Question: Does this extend to nonlinear setting? (ongoing work)

S. C. Surace, A. Kutschireiter, J. Pfister, How to avoid the curse of dimensionality: scalability of particle filters . . . , SIAM review, 2019
A. Taghvaei, P. G. Mehta, An optimal transport formulation of ensemble Kalman filter, (TAC) 2020

Curse of dimensionality
Linear Gaussian setting

- Method 1: OT approach with $\mathcal{F} = \{\text{quadratic functions}\}$ $\rightarrow$Ensemble Kalman filter
- Method 2: standard particle filter
- Compare number of particles to achieve error ϵ:

Question: Does this extend to nonlinear setting? (ongoing work)

S. C. Surace, A. Kutschireiter, J. Pfister, How to avoid the curse of dimensionality: scalability of particle filters . . . , SIAM review, 2019
A. Taghvaei, P. G. Mehta, An optimal transport formulation of ensemble Kalman filter, (TAC) 2020

Curse of dimensionality
Linear Gaussian setting

- Method 1: OT approach with $\mathcal{F} = \{\text{quadratic functions}\}$ $\rightarrow$Ensemble Kalman filter
- Method 2: standard particle filter
- Compare number of particles to achieve error ϵ:

Question: Does this extend to nonlinear setting? (ongoing work)

S. C. Surace, A. Kutschireiter, J. Pfister, How to avoid the curse of dimensionality: scalability of particle filters . . . , SIAM review, 2019
A. Taghvaei, P. G. Mehta, An optimal transport formulation of ensemble Kalman filter, (TAC) 2020

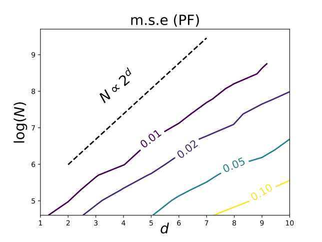
Curse of dimensionality
Linear Gaussian setting

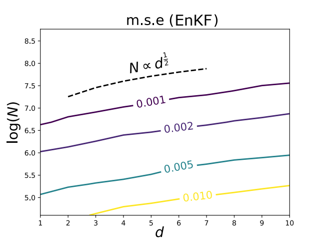
- Method 1: OT approach with $\mathcal{F} = \{\text{quadratic functions}\}$ $\rightarrow$Ensemble Kalman filter
- Method 2: standard particle filter
- Compare number of particles to achieve error ϵ:

Question: Does this extend to nonlinear setting? (ongoing work)

S. C. Surace, A. Kutschireiter, J. Pfister, How to avoid the curse of dimensionality: scalability of particle filters . . . , SIAM review, 2019
A. Taghvaei, P. G. Mehta, An optimal transport formulation of ensemble Kalman filter, (TAC) 2020

- Method 1: OT approach with $\mathcal{F} = \{\text{quadratic functions}\}$ →Ensemble Kalman filter
- Method 2: standard particle filter
- Compare number of particles to achieve error ϵ:





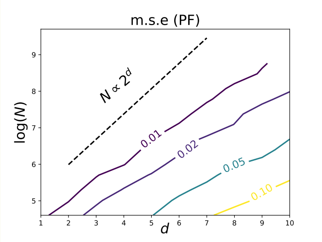
PF: $N \approx O(\frac{e^d}{\epsilon})$ EnKF: $N \approx O(\frac{d^2}{\epsilon})$

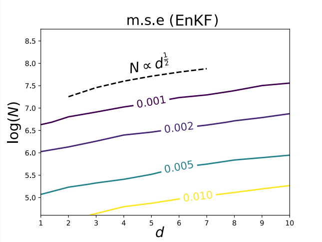
Question: Does this extend to nonlinear setting? (ongoing work)

S. C. Surace, A. Kutschireiter, J. Pfister, How to avoid the curse of dimensionality: scalability of particle filters . . . , SIAM review, 2019
A. Taghvaei, P. G. Mehta, An optimal transport formulation of ensemble Kalman filter, (TAC) 2020

Curse of dimensionality
Linear Gaussian setting

- Method 1: OT approach with $\mathcal{F} = \{\text{quadratic functions}\}$ $\rightarrow$Ensemble Kalman filter
- Method 2: standard particle filter
- Compare number of particles to achieve error ϵ:





PF: $N \approx O(\frac{e^d}{\epsilon})$

EnKF: $N \approx O(\frac{d^2}{\epsilon})$

Question: Does this extend to nonlinear setting? (ongoing work)

S. C. Surace, A. Kutschireiter, J. Pfister, How to avoid the curse of dimensionality: scalability of particle filters . . . , SIAM review, 2019
A. Taghvaei, P. G. Mehta, An optimal transport formulation of ensemble Kalman filter, (TAC) 2020

Main feature of OT approach
Flexibility in choosing $\mathcal{F}$

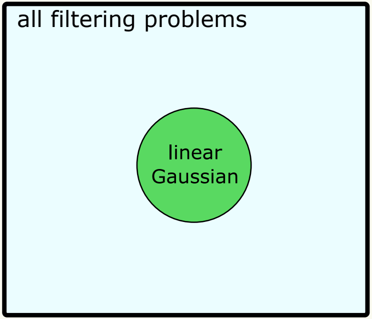
- particle filters treat all filtering problems similarly
- they can not take advantage of possibly existing simplicity or regularity
- OT approach allows flexibility (a knob) by changing optimization domain
- One can trade-off computational effort by the range of problems it can solve

Main feature of OT approach

Flexibility in choosing $\mathcal{F}$

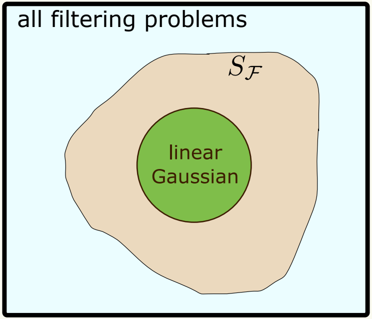
- particle filters treat all filtering problems similarly
- they can not take advantage of possibly existing simplicity or regularity
- OT approach allows flexibility (a knob) by changing optimization domain
- One can trade-off computational effort by the range of problems it can solve

Main feature of OT approach
Flexibility in choosing $\mathcal{F}$



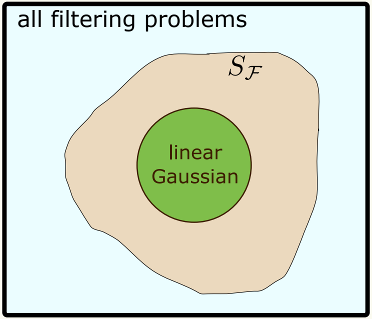
- particle filters treat all filtering problems similarly
- they can not take advantage of possibly existing simplicity or regularity
- OT approach allows flexibility (a knob) by changing optimization domain
- One can trade-off computational effort by the range of problems it can solve

Main feature of OT approach
Flexibility in choosing $\mathcal{F}$



- particle filters treat all filtering problems similarly
- they can not take advantage of possibly existing simplicity or regularity
- OT approach allows flexibility (a knob) by changing optimization domain
- One can trade-off computational effort by the range of problems it can solve

Main feature of OT approach
Flexibility in choosing $\mathcal{F}$



- particle filters treat all filtering problems similarly
- they can not take advantage of possibly existing simplicity or regularity
- OT approach allows flexibility (a knob) by changing optimization domain
- One can trade-off computational effort by the range of problems it can solve

Summary

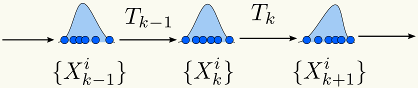
- **Mathematical model:**

$$
\begin{tikzcd}
{} \arrow[r] & X_{k-1} \arrow[r] \arrow[d] & X_k \arrow[r] \arrow[d] & X_{k+1} \arrow[r] \arrow[d] & {} \\
& Y_{k-1} & Y_k & Y_{k+1} &
\end{tikzcd}
$$

- **Nonlinear filtering:** compute the posterior $\pi_k = \mathsf{P}(X_k | Y_{1:k})$

$$
\longrightarrow \pi_{k-1} \longrightarrow \pi_k \longrightarrow \pi_{k+1} \longrightarrow
$$

- **OT approach:**



- **Feature:** variational approach enables flexibility (a knob) by changing optimization domain

Optimal transportation methods in nonlinear filtering: The feedback particle filter, IEEE CSM, 2021

Outline

- Introduction to optimal transport
- Application to nonlinear filtering
- Possible collaboration directions

Outline

- Introduction to optimal transport
- Application to nonlinear filtering
- Possible collaboration directions

Optimal sensing

Nonlinear filtering provides framework to study the effect of sensors on uncertainty:

- sparse sensing: select minimal number of sensors
- how to optimally place sensors
- study biological sensing mechanisms in terms of their effect on uncertainty

Optimal sensing

Nonlinear filtering provides framework to study the effect of sensors on uncertainty:

- sparse sensing: select minimal number of sensors
- how to optimally place sensors
- study biological sensing mechanisms in terms of their effect on uncertainty

Optimal sensing

Nonlinear filtering provides framework to study the effect of sensors on uncertainty:

- sparse sensing: select minimal number of sensors
- how to optimally place sensors
- study biological sensing mechanisms in terms of their effect on uncertainty

Optimal sensing

Nonlinear filtering provides framework to study the effect of sensors on uncertainty:

- sparse sensing: select minimal number of sensors
- how to optimally place sensors
- study biological sensing mechanisms in terms of their effect on uncertainty

Active perception

- What if we have control?

$$
\begin{tikzcd}
{} \arrow[r] & X_{k-1} \arrow[r, "u_{k-1}"] \arrow[d] & X_k \arrow[r, "u_k"] \arrow[d] & X_{k+1} \arrow[r] \arrow[d] & {} \\
& Y_{k-1} & Y_k & Y_{k+1} &
\end{tikzcd}
$$

- u_k can only depend on the history $\{Y_1, \dots, Y_k\}$
- How to design control policies to reduce uncertainty optimally?
- How to solve optimal control problems

$$\min_u \mathbb{E}[\sum_{k=1}^{T} c(X_k, u_k) + \phi(X_T)]$$

Active perception

- What if we have control?

$$
\begin{tikzcd}
{} \arrow[r] & X_{k-1} \arrow[r, "u_{k-1}"] \arrow[d] & X_k \arrow[r, "u_k"] \arrow[d] & X_{k+1} \arrow[r] \arrow[d] & {} \\
& Y_{k-1} & Y_k & Y_{k+1} &
\end{tikzcd}
$$

- u_k can only depend on the history $\{Y_1, \ldots, Y_k\}$
- How to design control policies to reduce uncertainty optimally?
- How to solve optimal control problems

$$\min_u \mathbb{E}[\sum_{k=1}^{T} c(X_k, u_k) + \phi(X_T)]$$

Active perception

- What if we have control?

$$
\begin{tikzcd}
{} \arrow[r] & X_{k-1} \arrow[r, "u_{k-1}"] \arrow[d] & X_k \arrow[r, "u_k"] \arrow[d] & X_{k+1} \arrow[r] \arrow[d] & {} \\
 & Y_{k-1} & Y_k & Y_{k+1} &
\end{tikzcd}
$$

- u_k can only depend on the history $\{Y_1, \ldots, Y_k\}$
- How to design control policies to reduce uncertainty optimally?
- How to solve optimal control problems

$$\min_u \mathbb{E}[\sum_{k=1}^{T} c(X_k, u_k) + \phi(X_T)]$$

Active perception

- What if we have control?

$$
\begin{tikzcd}
{} \arrow[r] & X_{k-1} \arrow[r, "u_{k-1}"] \arrow[d] & X_k \arrow[r, "u_k"] \arrow[d] & X_{k+1} \arrow[r] \arrow[d] & {} \\
 & Y_{k-1} & Y_k & Y_{k+1} &
\end{tikzcd}
$$

- u_k can only depend on the history $\{Y_1, \ldots, Y_k\}$
- How to design control policies to reduce uncertainty optimally?
- How to solve optimal control problems

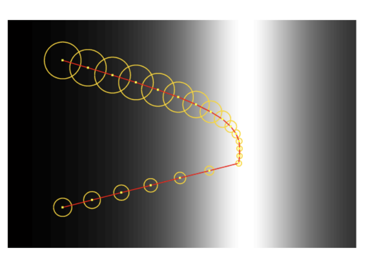
$$\min_u \mathbb{E}[\sum_{k=1}^{T} c(X_k, u_k) + \phi(X_T)]$$

Active perception

- Common approach: perform control and estimation independently (separation principle)
- Does not give solutions that might be expensive at first, but reduce uncertainty
- formulate as control problem for the posterior or belief
- use nonlinear filtering algorithms to represent and propagate posterior
- Application: perception aware motion planning

Active perception

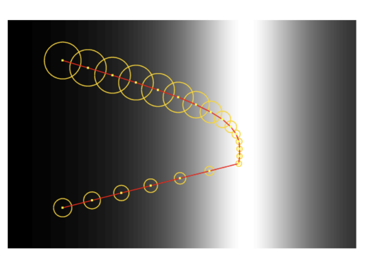
- Common approach: perform control and estimation independently (separation principle)
- Does not give solutions that might be expensive at first, but reduce uncertainty



- formulate as control problem for the posterior or belief
- use nonlinear filtering algorithms to represent and propagate posterior
- Application: perception aware motion planning

Active perception

- Common approach: perform control and estimation independently (separation principle)
- Does not give solutions that might be expensive at first, but reduce uncertainty



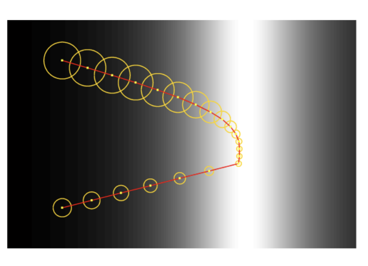
- formulate as control problem for the posterior or belief

$$\longrightarrow \pi_{k-1} \xrightarrow{u_{k-1}} \pi_k \xrightarrow{u_k} \pi_{k+1} \longrightarrow$$

- use nonlinear filtering algorithms to represent and propagate posterior
- Application: perception aware motion planning

Active perception

- Common approach: perform control and estimation independently (separation principle)
- Does not give solutions that might be expensive at first, but reduce uncertainty



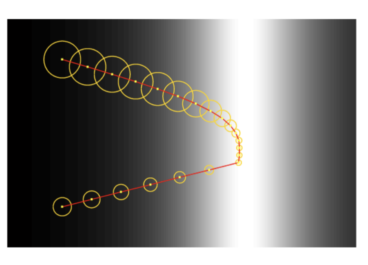
- formulate as control problem for the posterior or belief

$$\longrightarrow \pi_{k-1} \xrightarrow{u_{k-1}} \pi_k \xrightarrow{u_k} \pi_{k+1} \longrightarrow$$

- use nonlinear filtering algorithms to represent and propagate posterior
- Application: perception aware motion planning

Active perception

- Common approach: perform control and estimation independently (separation principle)
- Does not give solutions that might be expensive at first, but reduce uncertainty



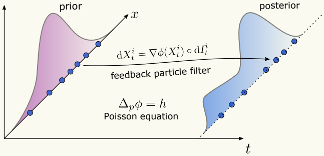
- formulate as control problem for the posterior or belief

$$\longrightarrow \pi_{k-1} \xrightarrow{u_{k-1}} \pi_k \xrightarrow{u_k} \pi_{k+1} \longrightarrow$$

- use nonlinear filtering algorithms to represent and propagate posterior
- Application: perception aware motion planning

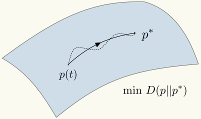
Final slide

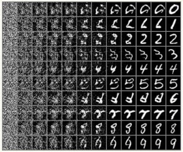
(I) Optimal filtering & control



- Optimal transportation methods in nonlinear filtering: The feedback particle filter, CSM, 2021
- An optimal transport formulation of the ensemble Kalman filter, TAC, 2021

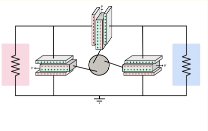
(II) Machine learning

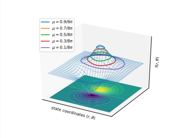




- OT mapping via input-convex neural networks, ICML, 2020
- Scalable computations of Wasserstein barycenter via input convex neural networks, ICML, 2021
- Variational Wasserstein gradient flow, Submitted to ICML, 2022

(III) Stochastic thermodynamics





- Energy harvesting from anisotropic fluctuations, PRE, 2021
- On the relation between information and power in stochastic thermodynamic engines, (L-CSS), 2021
- Maximal power output of a stochastic thermodynamic engine, Automatica, 2021

Possible collaboration:

- Optimal sensing
- Active perception

Thank you for your attention!