Optimal Transport Methods and Applications

Yunpeng Li

Major collaborators: Yongxin Yang, Xiongjie Chen

Department of Computer Science

University of Surrey, UK

16th January, 2023

Bellairs Workshop 2023

Outline

Introduction to optimal transport

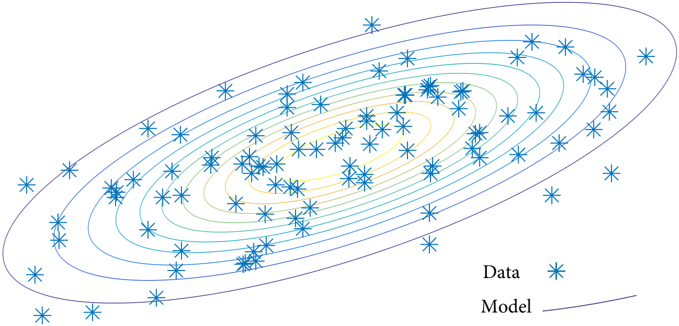
A computationally efficient variant

- Augmented Wasserstein distances

Applications

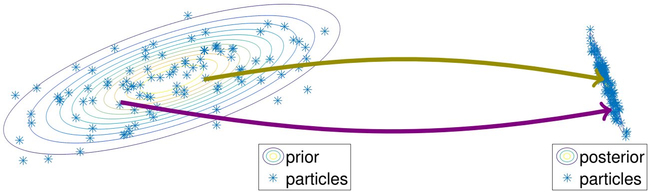
- Reinforcement learning
- Finance

Motivating Examples: density fitting



How to measure the distance between probability distributions?

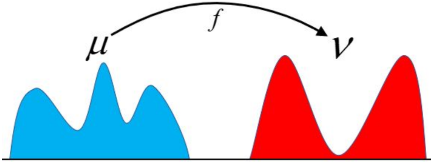
Motivating Examples: Bayesian inference



How to transform between probability distributions efficiently?

Optimal transport

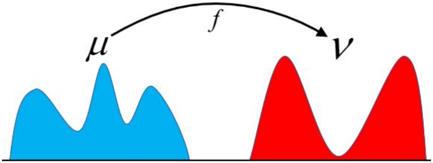
Knowing point-to-point transport costs,
transport a source distribution μ to a target distribution ν
with minimum overall costs.



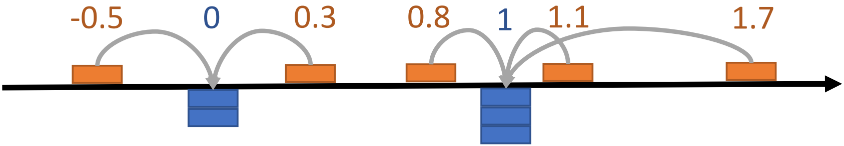
How to find the transport map?

Optimal transport

Knowing point-to-point transport costs,
transport a source distribution μ to a target distribution ν
with minimum overall costs.



How to find the transport map?



The optimal transport is illustrated by grey arrows.

Monge's formulation

Let μ and ν be probability measures defined on Ω, and $c : \Omega \times \Omega \to [0, +\infty)$ a distance metric.

Gaspard Monge
(1746-1818)

The Monge problem finds a transport map $T : \Omega \to \Omega$ minimising the expectation of cost function:

$$M(T) := \int_{\Omega} c(x, T(x))\mu(x)\,.$$

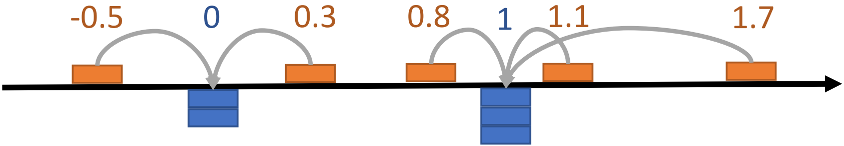
Monge's formulation

Let μ and ν be probability measures defined on Ω, and $c : \Omega \times \Omega \to [0, +\infty)$ a distance metric.

Gaspard Monge (1746-1818)

The Monge problem finds a transport map $T : \Omega \to \Omega$ minimising the expectation of cost function:

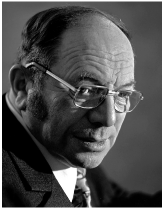
$$M(T) := \int_{\Omega} c(x, T(x))\mu(x)\,.$$



The Monge map is illustrated by grey arrows.

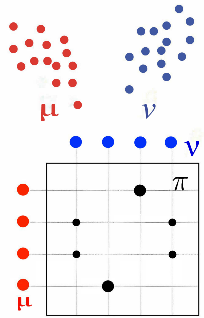
Kantorovich's formulation

Consider the distribution π defined on $\Omega \times \Omega$ that satisfies $\pi(A \times \Omega) = \mu(A)$, $\pi(\Omega \times B) = \nu(B)$, i.e. π is a joint distribution with marginals μ and ν.



Leonid Kantorvich (1912-1986)

An illustration[1] of the joint distribution $\pi(x, y)$.



[1] Peyre et al., "Computational Optimal Transport", Now Publishers, 2019.

Kantorovich's formulation

Kantorovich's formulation tries to find a joint distribution π that minimises:

$$\int_{\Omega\times\Omega} c(x,y)\mathrm{d}\pi(x,y)\,.$$

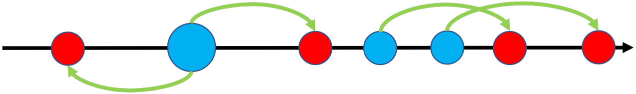
Kantorovich's formulation

Kantorovich's formulation tries to find a joint distribution π that minimises:

$$\int_{\Omega \times \Omega} c(x, y) \mathrm{d}\pi(x, y) \,.$$

π corresponds to a transport map:

	0.25	0.25	0.25	0.25	μ
0.5	0.25	0.25	0.0	0.0	
0.25	0.0	0.0	0.25	0.0	
0.25	0.0	0.0	0.0	0.25	
ν					

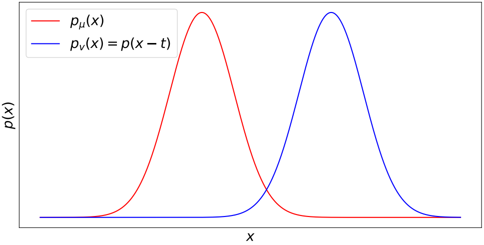


Distances between probability measures

Given two probability measures μ and ν, we want to measure the discrepancy between them by computing a distance metric $D(\cdot, \cdot)$:

$$D(\mu, \nu) : P(\Omega) \times P(\Omega) \to \mathbb{R},$$

where $P(\Omega)$ is the set of all Borel probability measures defined on Ω.

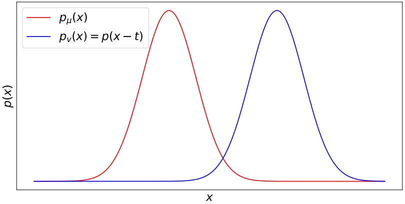


Examples of discrepancy measures

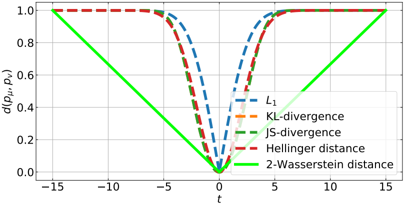
Denote by p_μ and p_ν the densities of μ and ν, we can evaluate the distance between μ and ν by computing the following discrepancy measures:

- L_k-metrics ($k \geq 1$): $L_k(\mu, \nu) = \left(\int_\Omega |p_\mu(x) - p_\nu(x)|^k dx \right)^{\frac{1}{k}}$
- KL-divergence: $D_{KL}(\mu||\nu) = \int_\Omega p_\mu(x) \log \left(\frac{p_\mu(x)}{p_\nu(x)} \right) dx$
- JS-divergence: $\text{JSD}(\mu||\nu) = \frac{1}{2} D_{KL}(\mu||\nu) + \frac{1}{2} D_{KL}(\nu||\mu)$
- Hellinger distance: $H^2(\mu, \nu) = \frac{1}{2} \int_\Omega \left(\sqrt{p_\mu(x)} - \sqrt{p_\nu(x)} \right)^2 dx$
- Wasserstein distance:
  $W_k(\mu, \nu) = \left(\inf_{\pi \in \Omega(\mu,\nu)} \int_{\Omega \times \Omega} c(x, y)^k d\pi(x, y) \right)^{\frac{1}{k}}$

Comparisons between different discrepancy measures



PDF of μ and ν



Only Wasserstein distance captures the geometry of the space

Properties of Wasserstein distance

Wasserstein distance is a valid metric.

- Symmetry
- Triangular inequality
- Identity of indiscernibles
- Non-negativity

Wasserstein distance can capture the underlying geometry of the space.

Limitations of Wasserstein distance

Computing the optimal transport plans is computationally intensive when the sample size is large.

More specifically, denote by n the number of samples, the computational complexity of computing the Wasserstein distance is:

$$\mathcal{O}\left(n^3 \log(n)\right).$$

Outline

Introduction to optimal transport

A computationally efficient variant

- Augmented Wasserstein distances

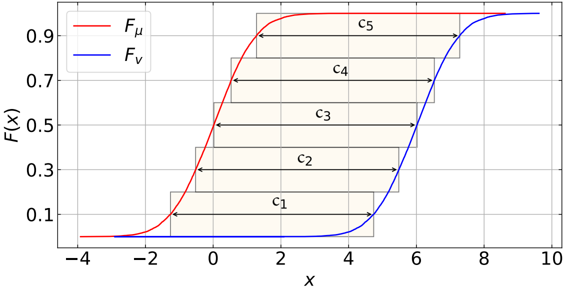
Applications

- Reinforcement learning
- Finance

Wasserstein distance in one-dimensional space

In one-dimensional space, the optimal transport plan has closed-form solution:

$$W_k(\mu,\nu) = \left(\int_0^1 c\big(F_\mu^{-1}(z), F_\nu^{-1}(z)\big)^k dz \right)^{\frac{1}{k}}.$$



The Wasserstein distance equals the area

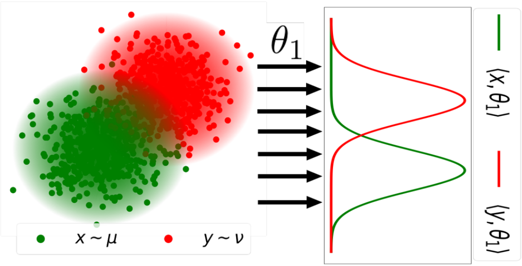
How to obtain 1-D distributions?

- Project high-dimensional distributions onto 1-dimensional spaces through Radon transform $\mathcal{R}_\mu(\,\cdot\,;\,\theta)$ (linear projections via dot product $\langle x, \theta \rangle$).

How to obtain 1-D distributions?

- Project high-dimensional distributions onto 1-dimensional spaces through Radon transform $\mathcal{R}_\mu(\,\cdot\,;\,\theta)$ (linear projections via dot product $\langle x, \theta \rangle$).

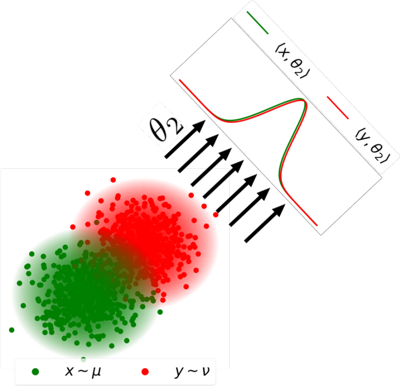
- Radon transform



How to obtain 1-D distributions?

- Project high-dimensional distributions onto 1-dimensional spaces through Radon transform $\mathcal{R}_\mu(\cdot \,;\, \theta)$ (linear projections via dot product $\langle x, \theta \rangle$).

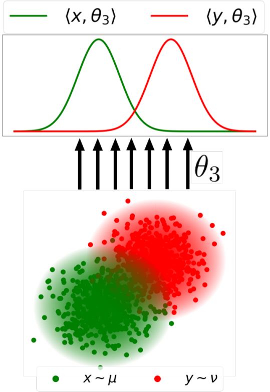
- Radon transform



How to obtain 1-D distributions?

- Project high-dimensional distributions onto 1-dimensional spaces through Radon transform $\mathcal{R}_\mu(\,\cdot\,;\,\theta)$ (linear projections via dot product $\langle x, \theta \rangle$).

- Radon transform



Sliced Wasserstein Distance[2] (SWD)

Definition:

$$\mathsf{SWD}_k(\mu, \nu) = \left(\int_{\mathbb{S}^{d-1}} W_k^k\big(\mathcal{R}_\mu(\cdot, \theta), \mathcal{R}_\nu(\cdot, \theta)\big) d\theta \right)^{\frac{1}{k}}.$$

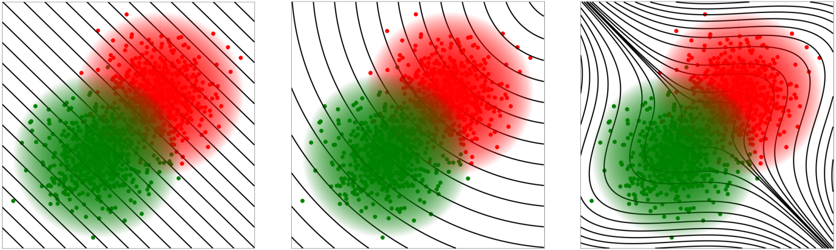
Intuitive interpretation:

- Obtain multiple one-dimensional distribution by using Radon transform $\mathcal{R}_\mu(\,\cdot\,;\,\theta)$.
- Average the Wasserstein distances between projected one-dimensional distributions.

[2] Bonnel et al., "Sliced and Radon Wasserstein barycenters of measures", JMIV, 2015

Generalized Sliced Wasserstein Distance[3] (GSWD)

- Obtain 1-dimensional distributions through generalized Radon transform $\mathcal{G}_\mu(\,\cdot\,,\theta)$ (nonlinear projections via defining function $\beta(x,\theta)$).

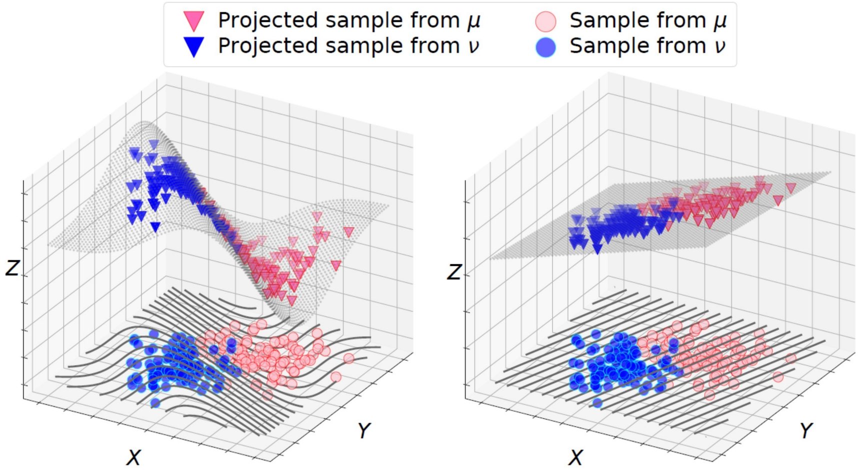


Inner product	Circular	Polynomial
$\langle x, \theta \rangle$	$\lvert\lvert x - r \cdot \theta \rvert\rvert_2$	$\sum_{\alpha=m} \theta_\alpha x^\alpha$

[3]Kolouri et al., "Generalized sliced Wasserstein distance", NeurIPS, 2019

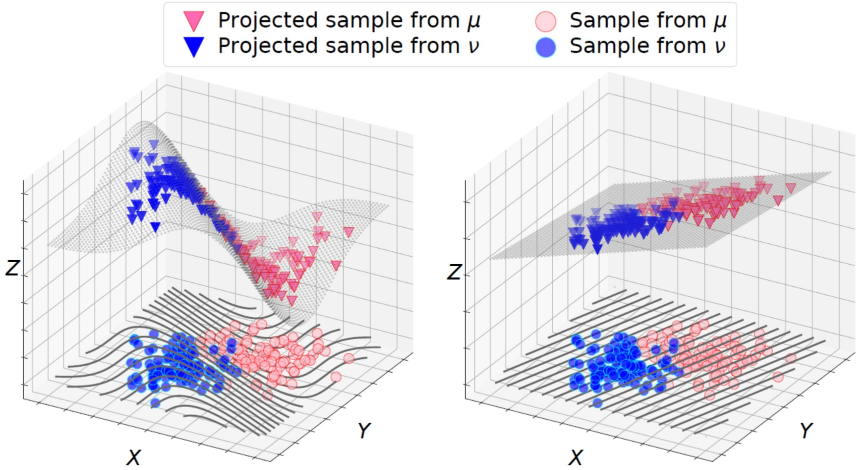
Importance of flexible nonlinear projections

Nonlinear projections can have higher projection efficiency than linear projections:



Importance of flexible nonlinear projections

Nonlinear projections can have higher projection efficiency than linear projections:



Generalized Sliced Wasserstein Distance[3] (GSWD)

× Limited choice of defining function $\beta(\cdot)$, must satisfy non-trivial conditions to guarantee a valid metric[5].

[3] Kolouri et al., "Generalized sliced Wasserstein distance", NeurIPS, 2019

Generalized Sliced Wasserstein Distance[3] (GSWD)

× Limited choice of defining function $\beta(\cdot)$, must satisfy non-trivial conditions to guarantee a valid metric[5].

× $\beta(\cdot)$ user-specified and not data-adaptive.

[3] Kolouri et al., "Generalized sliced Wasserstein distance", NeurIPS, 2019

Research questions

How to construct flexible hypersurfaces where the compared distributions are projected onto?

Research questions

How to construct flexible hypersurfaces where the compared distributions are projected onto?

Our methods

Spatial Radon transform;

Research questions

- How to construct flexible hypersurfaces where the compared distributions are projected onto?
- Can we optimise the hypersurface by learning from data to improve the projection efficiency?

Our methods

- Spatial Radon transform;

Research questions

How to construct flexible hypersurfaces where the compared distributions are projected onto?

Can we optimise the hypersurface by learning from data to improve the projection efficiency?

Our methods

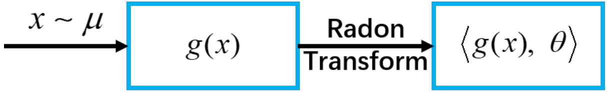
Spatial Radon transform;

Augmented sliced Wasserstein distance (ASWD).

Spatial Radon transform (SRT)

- How does the spatial Radon transform $\mathcal{H}_\mu(\cdot, \theta; g)$ construct nonlinear projections?

Radon transform	Generalized RT	Spatial RT
$\langle x, \theta \rangle$	$\beta(x, \theta)$	$\langle g(x), \theta \rangle$



Augmented sliced Wasserstein distance (ASWD)[4]

Definition:

$$\mathsf{ASWD}_k(\mu,\nu;g) = \left(\int_{\mathbb{S}^{d_\theta - 1}} W_k^k\big(\mathcal{H}_\mu(\cdot,\theta;g), \mathcal{H}_\nu(\cdot,\theta;g)\big) d\theta \right)^{\frac{1}{k}}$$

- Averages the Wasserstein distances between 1-D distributions obtained through spatial Radon transform.

[4] X. Chen, Y. Yang, and Y. Li. "Augmented Sliced Wasserstein Distances", ICLR 2022

Augmented sliced Wasserstein distance (ASWD)[4]

Definition:

$$\text{ASWD}_k(\mu,\nu;g) = \left(\int_{\mathbb{S}^{d_\theta-1}} W_k^k\big(\mathcal{H}_\mu(\cdot,\theta;g),\mathcal{H}_\nu(\cdot,\theta;g)\big)d\theta\right)^{\frac{1}{k}}$$

- Averages the Wasserstein distances between 1-D distributions obtained through spatial Radon transform.

Is ASWD a valid metric?

Theorem 1

The augmented sliced Wasserstein distance (ASWD) of order $k \in [1,+\infty)$ *with a fixed mapping* $g(\cdot):\mathbb{R}^d \to \mathbb{R}^{d_\theta}$ *is a metric on* $P_k(\mathbb{R}^d)$ *if and only if* $g(\cdot)$ *is injective.*

[4] X. Chen, Y. Yang, and Y. Li. "Augmented Sliced Wasserstein Distances", ICLR 2022

Injectivity of spatial Radon transform

Lemma 1

The spatial Radon transform is an injection on $P_k(\mathbb{R}^d)$ *if and only if the mapping* $g(\cdot)$ *is an injection.*

Is ASWD a valid metric when $g(\cdot)$ is optimised?

Optimisation objective:

$$g^*(\cdot) = \underset{g}{\operatorname{argmax}}\{\mathsf{ASWD}_k(\mu, \nu; g) - \lambda(\mathbb{E}_{x\sim\mu}^{\frac{1}{k}}\left[||g(x)||_2^k\right] + \mathbb{E}_{y\sim\nu}^{\frac{1}{k}}\left[||g(y)||_2^k\right])\}$$

Corollary 1.1

The ASWD is a valid metric when $\lambda \geq 1$.

Experiment Results[4]

A simple injective mapping $g_\omega(x) = [x, \phi_\omega(x)]$ adopted for all experiments.

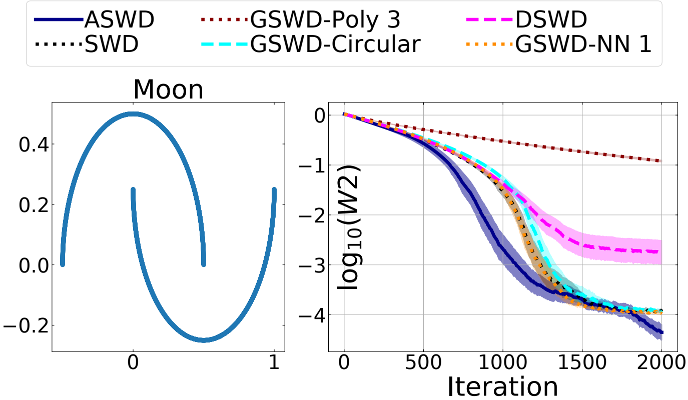
- Sliced Wasserstein flow (Section 5.1 and Appendix G);
- Generative modeling (Section 5.2 and Appendix H);
- Sliced Wasserstein autoencoders (Appendix I);
- Image color transferring (Appendix J);
- Sliced Wasserstein barycenter (Appendix K).

[4] X. Chen, Y. Yang, and Y. Li. "Augmented Sliced Wasserstein Distances", ICLR 2022

Sliced Wasserstein flow

Evolve a source distribution μ to a target distribution ν by minimizing different distance metrics between μ and ν:

$$\partial_t \mu_t = -\nabla \text{SWD}(\mu_t, \nu)\,,$$



Generative modelling

Train GAN models with different metrics on CELEBA and CIFAR10 datasets:

- The ASWD produced the lowest Fréchet Inception Distance (FID) score compared with other evaluated metrics:

L=1000	SWD (Bonneel et al., 2015)		GSWD (Kolouri et al., 2019a)		DSWD (Nguyen et al., 2021)		ASWD	
	FID	t (s/it)	FID	t (s/it)	FID	t (s/it)	FID	t (s/it)
CIFAR10	102.3±5.3	0.36	98.2±5.1	2.22	62.3 ± 5.7	1.30	**59.3±3.2**	1.38
CELEBA	86.5±4.1	0.38	85.2±6.3	2.19	71.3±4.7	1.28	**67.4±2.1**	1.38

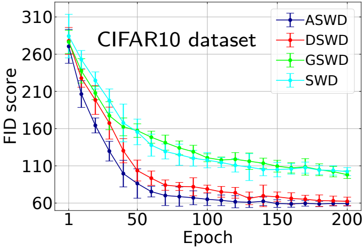
Generative modelling

Train GAN models with different metrics on CELEBA and CIFAR10 datasets:

- The ASWD produced the lowest Fréchet Inception Distance (FID) score compared with other evaluated metrics:

L=1000	SWD (Bonneel et al., 2015)		GSWD (Kolouri et al., 2019a)		DSWD (Nguyen et al., 2021)		ASWD	
	FID	t (s/it)	FID	t (s/it)	FID	t (s/it)	FID	t (s/it)
CIFAR10	102.3±5.3	0.36	98.2±5.1	2.22	62.3 ± 5.7	1.30	**59.3±3.2**	1.38
CELEBA	86.5±4.1	0.38	85.2±6.3	2.19	71.3±4.7	1.28	**67.4±2.1**	1.38

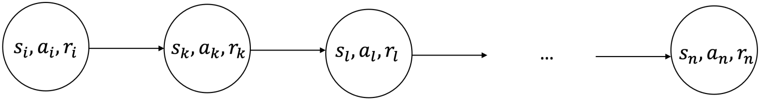
- The ASWD also has higher convergence rate in terms of the FID score:



Outline

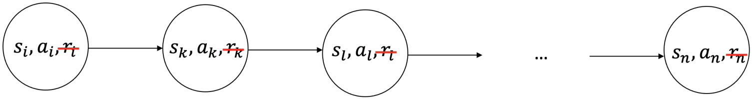
Reinforcement learning (RL)

Standard Markov Decision Process



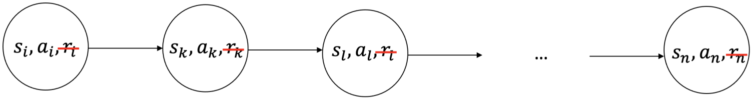
Reinforcement learning (RL) without rewards

Often reward is unavailable or hard to define



Reinforcement learning (RL) without rewards

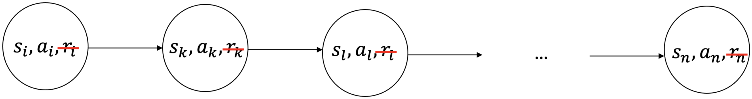
Often reward is unavailable or hard to define



- Instead, **learn** from demonstrations
- Inverse RL: Explicitly infer reward, optimise with RL (**ill-posed, computationally expensive)**

Reinforcement learning (RL) without rewards

Often reward is unavailable or hard to define

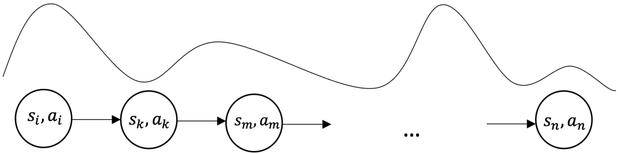


- Instead, **learn** from demonstrations
- Inverse RL: Explicitly infer reward, optimise with RL (**ill-posed, computationally expensive)**
- Imitation learning: Learn from demonstration directly, without explicit reward inference

Imitation learning

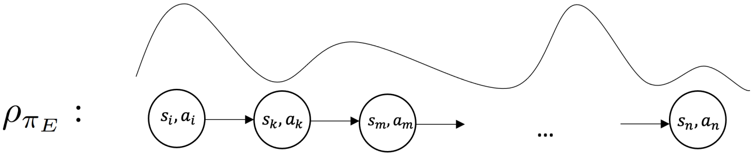
Demonstrator policy π_E with occupancy measure ρ_{π_E}:

ρ_{π_E} :

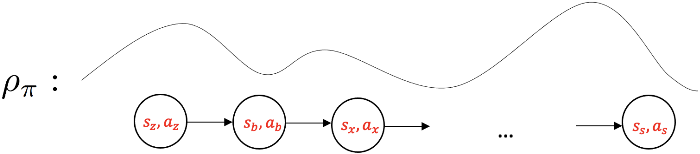


Imitation learning

Demonstrator policy π_E with occupancy measure ρ_{π_E}:

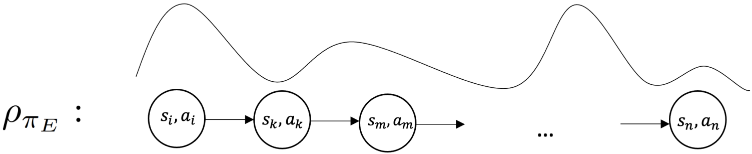


Learner policy π with occupancy measure ρ_π:

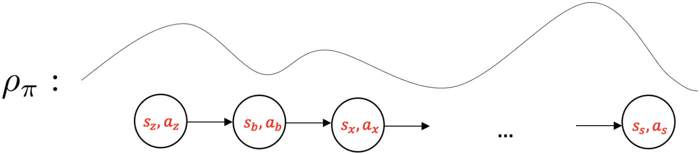


Imitation learning

Demonstrator policy π_E with occupancy measure ρ_{π_E}:



Learner policy π with occupancy measure ρ_π:



- Measure similarity with metric $\mathcal{D}(\rho_\pi, \rho_{\pi_E})$

Imitation learning

Objective: Find π such that
$\mathcal{D}(\rho_\pi, \rho_{\pi_E})$ is minimised.

Different similarity metrics $\mathcal{D}(\rho_\pi, \rho_{\pi_E})$

- Supervised learning: Behaviour Cloning (BC)
- Kullback-Leibler Divergence: Adversarial Inverse RL (AIRL)[5]
- Jensen-Shannon divergence: Generative Adversarial Imitation Learning (GAIL)[6]
- ... and any $f-$divergence[7]
- Dual Wasserstein: Wasserstein Adversarial Imitation Learning[8]
- Bounded Wasserstein: Primal Wasserstein Imitation Learning[9]

[5] Fu et al., "Learning Robust Rewards with Adversarial Inverse Reinforcement Learning", ICLR 2018

[6] Ho and Ermon ,"Generative adversarial imitation learning", NIPS 2016

[7] Ghasemipour et al., "A Divergence Minimization Perspective on Imitation Learning", CORL 2019

[8] Xiao et al., "Wasserstein Adversarial Imitation Learning", arXiv 2019

[9] Dadashi et al., "Primal Wasserstein Imitation Learning", ICLR 2021

Different similarity metrics $\mathcal{D}(\rho_\pi, \rho_{\pi_E})$

Limitations:

- Do not account for the distributions' metric space
- Not robust to disjoint measures
- Often solved with generative adversarial training, inheriting its disadvantages such as training instability
- Intractable
- Locally Optimal

Sinkhorn Distance[10]

$$\mathcal{W}_s^\beta(\rho_\pi, \rho_{\pi_E})_c = \inf_{\zeta_\beta \in \Omega_\beta(\rho_\pi, \rho_{\pi_E})} \mathbb{E}_{x,y\sim\zeta_\beta}\left[c(x,y)\right]$$

where $\Omega_\beta(p,q)$ denotes the set of all joint distributions in $\Omega(p,q)$ with entropy of at least $\mathcal{H}(p) + \mathcal{H}(q) - \beta$.

- This entropy regularised optimal transport problem can be solved by an algorithm called *Sinkhorn-Knopp's fixed point iteration*, and the solving process is differentiable.

[10] Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances", NIPS 2013

Sinkhorn Distance in Imitation Learning[11]

Sample transport cost:

$$v_c\left((s_z, a_z) \sim \rho_\pi\right) = \sum_{(s_i, a_i) \sim \rho_{\pi_E}} c\,\overbrace{\left((s_z, a_z), (s_i, a_i)\right)}^{\text{Distance cost}} \; \zeta_\beta \underbrace{\left((s_z, a_z), (s_i, a_i)\right)}_{\text{Optimal Transport Plan}}$$

[11] G. Papagiannis and Y. Li, "Imitation Learning with Sinkhorn Distances", ECML-PKDD 2022

Sinkhorn Distance in Imitation Learning[11]

Sample transport cost:

$$v_c\left((s_z, a_z) \sim \rho_\pi \right) = \sum_{(s_i, a_i) \sim \rho_{\pi_E}} c\,\overbrace{\left((s_z, a_z), (s_i, a_i) \right)}^{\text{Distance cost}} \; \zeta_\beta \underbrace{\left((s_z, a_z), (s_i, a_i) \right)}_{\text{Optimal Transport Plan}}$$

$$\mathcal{W}_s^\beta(\rho_\pi, \rho_{\pi_E})_c = \sum_{(s_z, a_z) \sim \rho_\pi} v_c\left((s_z, a_z) \right)$$

[11] G. Papagiannis and Y. Li, "Imitation Learning with Sinkhorn Distances", ECML-PKDD 2022

Sinkhorn Distance in Imitation Learning[11]

$$\mathcal{W}_s^{\beta}(\rho_{\pi}, \rho_{\pi_E})_{c_w} = \sum_{(s_z, a_z) \sim \rho_{\pi}} v_{c_w}((s_z, a_z))$$

$-v_{c_w}$ per sample reward proxy in reinforcement learning

- Cost learned using a neural network (NN) parameterised by w.
- Cosine distance between the output of the NN for each state-action pair.

[11] G. Papagiannis and Y. Li, "Imitation Learning with Sinkhorn Distances", ECML-PKDD 2022

Sinkhorn Distance in Imitation Learning[11]

SIL's Optimisation Objective:

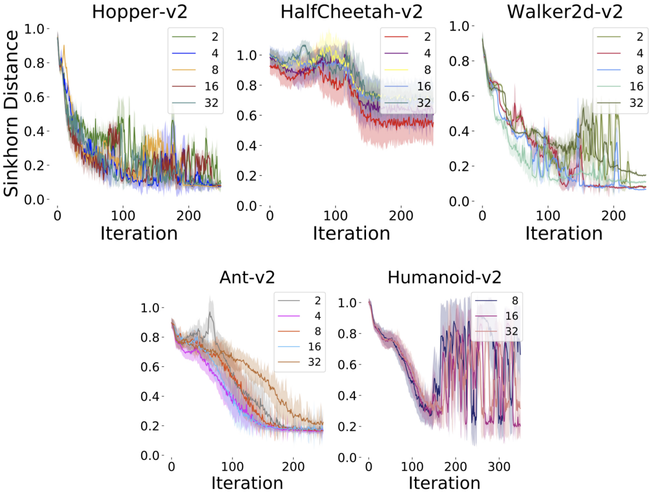
$$\min_{\pi} \max_{w} \mathcal{W}_s^{\beta}(\rho_\pi, \rho_{\pi_E})_{c_w}$$

Repeat to convergence:

Step 1: Optimise w parameterised as a NN to maximize $\mathcal{W}_s^{\beta}(\rho_\pi, \rho_{\pi_E})_{c_w}$
Step 2: Optimise π to minimise $\mathcal{W}_s^{\beta}(\rho_\pi, \rho_{\pi_E})_{c_w}$ using $-v_{c_w}$ as reward.

[11] G. Papagiannis and Y. Li, "Imitation Learning with Sinkhorn Distances", ECML-PKDD 2022

Results Overview



Successful imitation learning with various numbers of demonstrations.

Results Overview

Best performance on each experiment against benchmarks

Environments	Trajectories	BC	GAIL	AIRL	SIL
Hopper-v2	2	×	×	✓	×
	4	×	×	✓	×
	8	×	×	✓	×
	16	×	×	✓	×
	32	×	✓	×	×
HalfCheetah-v2	2	×	×	×	✓
	4	×	×	×	✓
	8	×	×	×	✓
	16	×	✓	×	×
	32	×	×	×	✓
Walker2d-v2	2	×	×	✓	×
	4	×	×	✓	×
	8	×	×	✓	×
	16	×	×	✓	×
	32	×	×	✓	×
	2	×	×	×	✓

	Trajectories	BC	GAIL	AIRL	SIL
Ant-v2	4	×	×	×	✓
	8	×	×	×	✓
	16	×	×	×	✓
	32	×	×	✓	×
Humanoid-v2	8	✓	×	×	×
	16	×	×	×	✓
	32	×	✓	×	×

SIL performs SOTA against benchmarks on some environments; on par on the rest.

Outline

Introduction to optimal transport

Computationally efficient variants

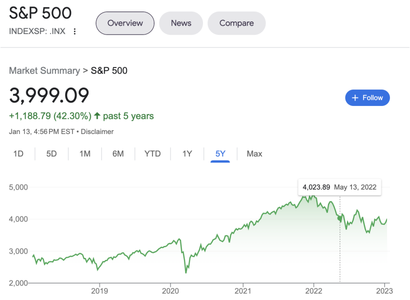
- Augmented Wasserstein distances

Applications

- Reinforcement learning
- Finance

Index Tracking

Index tracking is a popular form of **passive investing**, aiming to replicate the performance of a given index by constructing a portfolio which contains some constituents of the index.



Index Tracking

The objective is regression, minimising the tracking error $\min_{w} \|Xw - y\|_2^2$:

- $X \in \mathbb{R}^{N \times D}$ are the return of assets
- $y \in \mathbb{R}^{N}$ is the target index (benchmark)
- N is the number of timesteps (e.g., $N = 750$ trading days)
- D is the number of assets (e.g., $D = 500$ stocks)
- $w \in \mathbb{R}^{D}$ is the weight of each asset to hold in order to approximate the index y

Some Constraints

Beyond the simplest form, some constraints exist in this study

- **long-only**, i.e., $w_i \geq 0, \ \forall i$
- **the capital is fully utilised**, i.e., $\sum_i w_i = 1$

With the constraints, our objective becomes

- $\min_{w \geq \mathbf{0}, \sum_i w_i = 1} \|Xw - y\|_2^2$
- A non-negative regression problem with sum-to-one constraint

Cardinality Constraint

It becomes *much harder* if we want to control how many assets to buy

- Reduces transaction costs
- Makes the portfolio more manageable

$$\min_{w \geq \mathbf{0}, \sum_i w_i = 1, \|w\|_0 = K} \|Xw - y\|_2^2$$

- $\|w\|_0$ is the ℓ_0 norm, which is the number of non-zero elements in w
- This suggests that we will buy *exactly* K assets

Our contribution[12]

Why is it hard to find $\min_{w \geq \mathbf{0}, \sum_i w_i = 1, \|w\|_0 = K} \|Xw - y\|_2^2$?

- Asset selection (which elements in w are non-zero) is a discrete optimisation problem
- Capital allocation (what values of those non-zero elements) is a continuous optimisation problem
- If we want to optimise them jointly, gradient-based methods are not feasible because of asset selection part

We propose a **reparametrisation** for this problem, so it can *approximate* the gradient of asset selection, therefore we call it *differentiable asset selection*.

[12] Y. Zheng, Y. Li, Q. Xu, T. Hospedales, Y. Yang, "Index Tracking with Differentiable Asset Selection", ICAIF 2020

Reparameterisation

$$\min_{\tilde{w},s} \|Xw(\tilde{w},s) - y\|_2^2$$

- $w_i = \frac{1}{\sum_i \exp(\tilde{w}_i) z_i} \exp(\tilde{w}_i) z_i$
- $[z_1, z_2, \ldots, z_D] = \text{TopK}(s)$

$\text{TopK} : \mathbb{R}^D \rightarrow \{0,1\}^D$

- $s = [-0.5, 1.7, 0.3, 0.8, 1.1] \longrightarrow z = \text{Top3}(s) = [0, 1, 0, 1, 1]$

Reparameterisation

$$\min_{\tilde{w},s} \|Xw(\tilde{w}, s) - y\|_2^2$$

- $w_i = \frac{1}{\sum_i \exp(\tilde{w}_i) z_i} \exp(\tilde{w}_i) z_i$
- $[z_1, z_2, \ldots, z_D] = \mathrm{TopK}(s)$

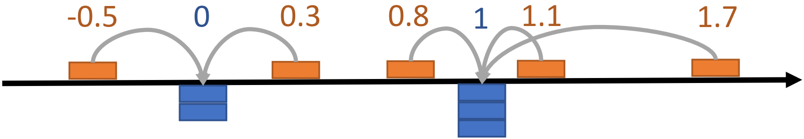
$\mathrm{TopK} : \mathbb{R}^D \to \{0, 1\}^D$

- $s = [-0.5, 1.7, 0.3, 0.8, 1.1] \longrightarrow z = \mathrm{Top3}(s) = [0, 1, 0, 1, 1]$

Note that $\tilde{w} \in \mathbb{R}^D$ and $s \in \mathbb{R}^D$, thus we just need to find a smoothed version of $\mathrm{TopK}(\cdot)$.

- This can be done by formulating the TopK operator as an optimal transportation problem.

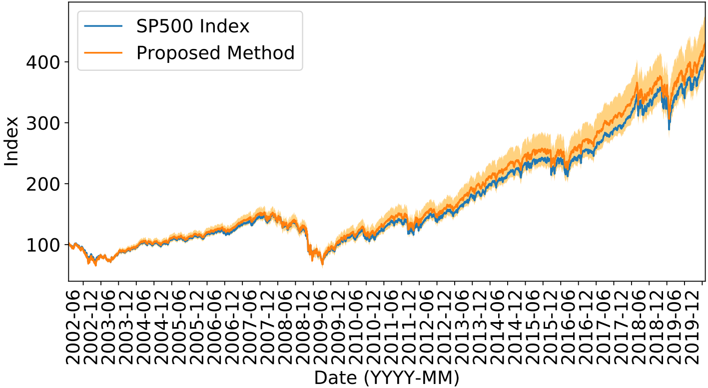
TopK via OT



Recall that this (entropy regularised) OT problem can be solved by an algorithm called *Sinkhorn-Knopp's fixed point iteration*, and the solving process is differentiable.

Stochasticity Analysis

- Out-of-sample performance of 100 runs when $K = 50$. Orange line: mean; shadow area: 1 standard deviation.
- The proposed method is consistently effective. Errors are accumulated so the shadow area becomes larger as time progresses.



Summary

- Augmented Sliced Wasserstein distances: a data-adaptive distance metric with high projection efficiency.
- Achieved through novel incorporation of injective neural networks to learn nonlinear projections.
- The Sinkhorn algorithm can be used to in distance minimisation and differentiable top-K/sorting functions with applications in RL, finance, image retrieval etc.

Link to code:

- Augmented sliced Wasserstein distances: https://github.com/xiongjiechen/Normalizing-Flows-DPFs.
- Imitation learning with Sinkhorn Distances: https://github.com/gpapagiannis/sinkhorn-imitation
- Index tracking with differentiable asset selection: available upon request.