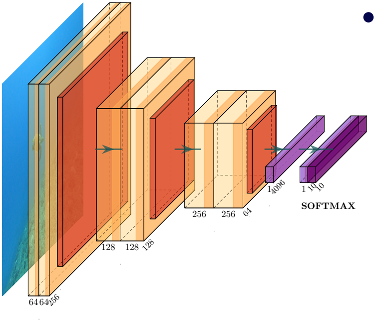
Introduction on Optimal Transport for Deep Learning

First definitions and properties

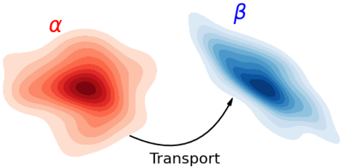
Kilian Fatras

March 24th, 2022

Mila, McGill



- **Introduction on deep learning**
    - Neural networks
    - Applications
    - Probability distributions



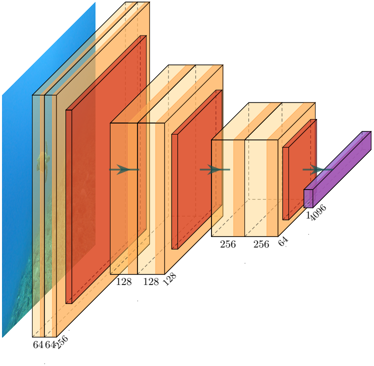
- **Introduction on optimal transport**
    - Definitions
    - Properties
    - Entropic variant

Introduction on Neural networks

Neural network illustration

Deep learning is a tool to estimate non-linear complex functions

- Neural networks: many stacked layers and each layer is made of neurones
- Parameters of neural networks: connections between layers
- Different layers: convolutional layers, fully connected layers, ...



Motivating example: Classification

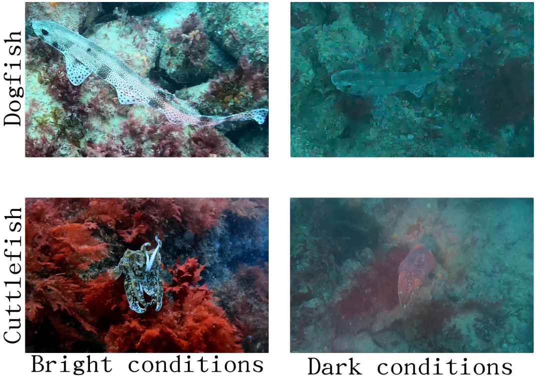
- Find a function f_θ which describes the relationship between the space of images and the space of classes
- f_θ is a **neural network** !

f_θ () $= \begin{array}{cc} \begin{pmatrix} 0.1 \\ 0.2 \\ 0.7 \end{pmatrix} & \begin{array}{c} \text{clown fish} \\ \text{grouper} \\ \text{turtle} \end{array} \end{array}$

- n training samples: $(\boldsymbol{x}_1, \boldsymbol{y}_1), \cdots, (\boldsymbol{x}_n, \boldsymbol{y}_n)$
- Goal: minimizing the *empirical risk* with respect to θ

$$\min_\theta R(f_\theta) = \min_\theta \frac{1}{N} \sum_{i=1}^{N} L(\boldsymbol{y}_i, f_\theta(\boldsymbol{x}_i))$$

Motivating example: Domain adaptation

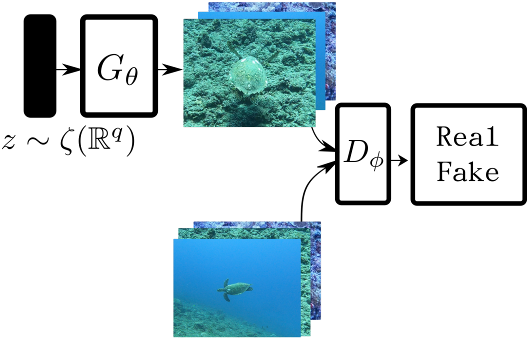


Domain adaptation (DA) setting

- Two domains with same classes, only one with labels
- Goal: classify unlabeled target data with source labeled data
- $\boldsymbol{x}_i^s, \boldsymbol{x}_j^t$ have same class $\rightarrow g_\phi(\boldsymbol{x}_i) \approx g_\phi(\boldsymbol{x}_j)$ and $\boldsymbol{y}_i = f_\theta(g_\phi(\boldsymbol{x}_j)))$

Motivating example: Generative adversarial networks

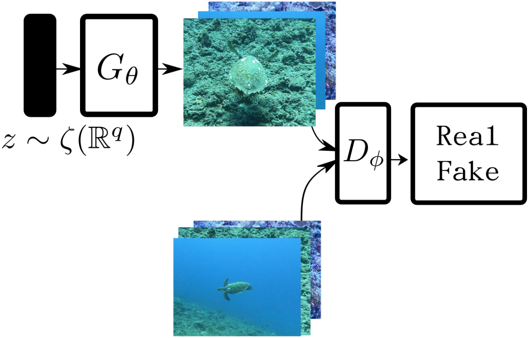
Goal: generating new images



- Generative adversarial networks (GANs) developed in [Goodfellow et al., 2014]
- G_θ tries to fool D_ϕ
- D_ϕ tries to predict if an image is real or not

Motivating example: Generative adversarial networks

Goal: generating new images



- $\alpha \in \mathcal{P}(\mathcal{X}), \zeta \in \mathcal{P}(\mathcal{Z})$ are probability distributions
- Loss: $\min_\theta \max_\phi \mathbb{E}_{\boldsymbol{x}\sim\alpha} \log\left(D_\phi(\boldsymbol{x})\right) - \mathbb{E}_{\boldsymbol{z}\sim\zeta} \log\left(1 - D_\phi(G_\theta(\boldsymbol{z}))\right)$

The loss can be reformulated with a Jensen-Shannon divergence between generated and training distributions

Training samples as distributions paradigm

Applications use probability distributions to train neural networks

- Classification: function L takes probability vectors as inputs
- Domain adaptation: align embedding probability distributions from domains
- GANs: distance between generated and training distributions

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \quad L(\alpha_n, \beta_\theta)$$

Goal : Find a suitable function L between probability distributions

Divergence and metric between probability distributions

Definition (Divergence)

Consider a set S. A divergence on S is a function $d : S \times S \mapsto [0, \infty]$ such that for all $\boldsymbol{x}, \boldsymbol{y}$:

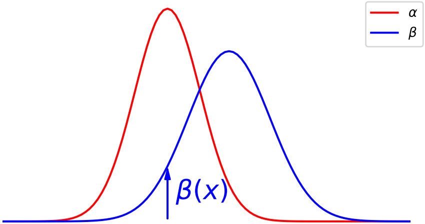
- $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non negativity)
- $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$ (separability)

Definition (Distance/Metric)

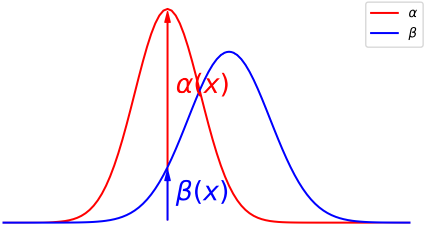
Consider a set S. A distance on S is a function $d : S \times S \mapsto [0, \infty]$ such that for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$:

- $d(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ (non negativity)
- $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$ (separability)
- $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ (symmetry)
- $d(\boldsymbol{x}, \boldsymbol{z}) \leq d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z})$ (triangle inequality)

Comparing probability distributions



Comparing probability distributions

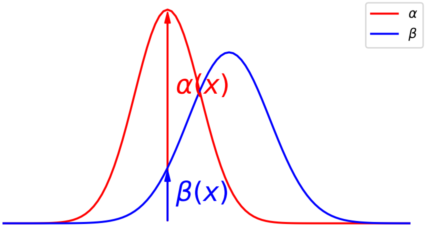


Suppose φ convex, $\varphi(1) = 0$ and α absolutely continuous *wrt* β.
φ-divergences compare mass ratio point-wise $\alpha(\boldsymbol{x})/\beta(\boldsymbol{x})$ ($\beta(\boldsymbol{x}) > 0$).

$$L_\varphi(\alpha|\beta) = \int_{\mathcal{X}} \varphi\left(\frac{d\alpha}{d\beta}\right) d\beta$$

We give several examples of φ-divergences.

Comparing probability distributions



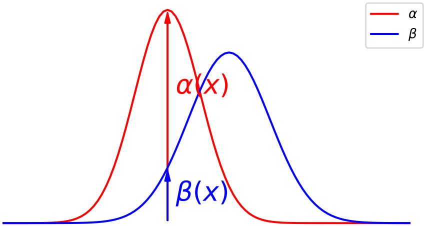
Suppose φ convex, $\varphi(1) = 0$ and α absolutely continuous *wrt* β.
φ-divergences compare mass ratio point-wise $\alpha(\boldsymbol{x})/\beta(\boldsymbol{x})$ ($\beta(\boldsymbol{x}) > 0$).

$$L_\varphi(\alpha|\beta) = \int_{\mathcal{X}} \varphi\left(\frac{d\alpha}{d\beta}\right) d\beta$$

We can get the Kullback-Leibler divergence for $\varphi(\boldsymbol{x}) = \boldsymbol{x}\log(\boldsymbol{x})$,

$$KL(\alpha|\beta) = \int_{\mathcal{X}} \log\left(\frac{d\alpha}{d\beta}\right) d\alpha$$

Comparing probability distributions



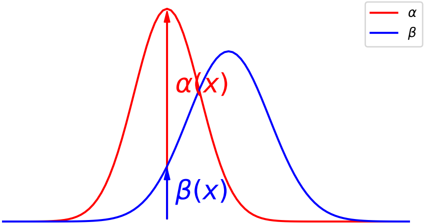
Suppose φ convex, $\varphi(1) = 0$ and α absolutely continuous *wrt* β.
φ-divergences compare mass ratio point-wise $\alpha(\boldsymbol{x})/\beta(\boldsymbol{x})$ $(\beta(\boldsymbol{x}) > 0)$.

$$L_\varphi(\alpha|\beta) = \int_{\mathcal{X}} \varphi\left(\frac{d\alpha}{d\beta}\right) d\beta$$

We can get the Total-Variation norm for $\varphi(\boldsymbol{x}) = \frac{1}{2}\,|\boldsymbol{x} - 1|$,

$$\mathrm{TV}(\alpha|\beta) = \int_{\mathcal{X}} \frac{1}{2}\left|\frac{d\alpha}{d\beta} - 1\right| d\alpha$$

Comparing probability distributions



Suppose φ convex, $\varphi(1) = 0$ and α absolutely continuous *wrt* β.
φ-divergences compare mass ratio point-wise $\alpha(\boldsymbol{x})/\beta(\boldsymbol{x})$ ($\beta(\boldsymbol{x}) > 0$).

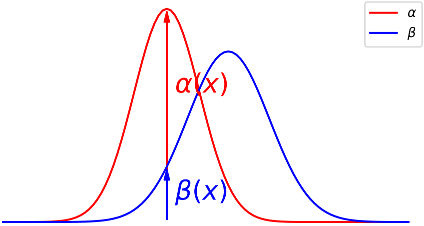
$$L_\varphi(\alpha|\beta) = \int_{\mathcal{X}} \varphi\left(\frac{d\alpha}{d\beta}\right) d\beta$$

- φ-divergences cannot compare Diracs

$\rightarrow$ fail to capture the geometry

- $\mathrm{KL}(\alpha|\beta_t) = +\infty$

Comparing probability distributions



Suppose φ convex, $\varphi(1) = 0$ and α absolutely continuous *wrt* β.
φ-divergences compare mass ratio point-wise $\alpha(\boldsymbol{x})/\beta(\boldsymbol{x})$ ($\beta(\boldsymbol{x}) > 0$).

$$L_\varphi(\alpha|\beta) = \int_{\mathcal{X}} \varphi\left(\frac{d\alpha}{d\beta}\right) d\beta$$

- φ-divergences cannot compare Diracs

$\rightarrow$ fail to capture the geometry

- $\mathrm{KL}(\alpha|\beta_t) = +\infty$ but $\mathrm{KL}(\alpha|\beta_\infty) = 0$

Weak convergence topology

Definition (Convergence in metric space)

A sequence $\{l_t\}_{t\in\mathbb{N}}$ of elements of a metric space (S,d) is said to converge to a limit $l \in S$ if $lim_{t\to\infty} d(l_t,l) = 0$.

For probability distributions, sequence β_t convergences to β with respect to a divergence d if $lim_{t\to\infty} d(\beta_t,\beta) = 0$. (Be carreful with the symmetry !)

φ-divergences do not metrize the weak convergence.

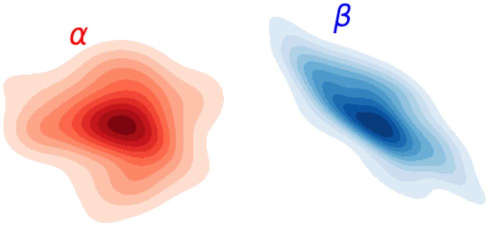
Example (TV-divergences)

For the probability sequence $\delta_{\frac{1}{n}}$, It is clear that $lim_{t\to\infty}\delta_{\frac{1}{n}} = \delta_0$ but we have $lim_{t\to\infty}\,\mathrm{TV}(\delta_{\frac{1}{n}},\delta_0) = lim_{t\to\infty} 1 = 1$.

So we are looking for a function d which ca compare probability distributions and which metrizes the weak convergence.

Introduction on Optimal Transport

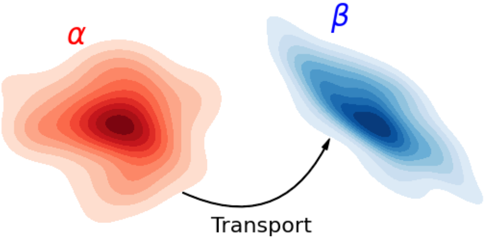
Optimal Transport definition



Ingredients

- Probability distributions $\alpha \in \mathcal{P}(\mathcal{X})$ and $\beta \in \mathcal{P}(\mathcal{Y})$
- A ground cost $c : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ with $\mathcal{X}$ and $\mathcal{Y}$ metric spaces

Optimal Transport definition



Ingredients

- Probability distributions $\alpha \in \mathcal{P}(\mathcal{X})$ and $\beta \in \mathcal{P}(\mathcal{Y})$
- A ground cost $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ with $\mathcal{X}$ and $\mathcal{Y}$ metric spaces

Optimal Transport definition

Definition (Kantorovich problem [Kantorovich, 1942])

$$\min_{\pi \in U(\alpha,\beta)} \int_{\mathcal{X}\times\mathcal{Y}} c(\mathbf{x},\mathbf{y}) d\pi(\mathbf{x},\mathbf{y})$$

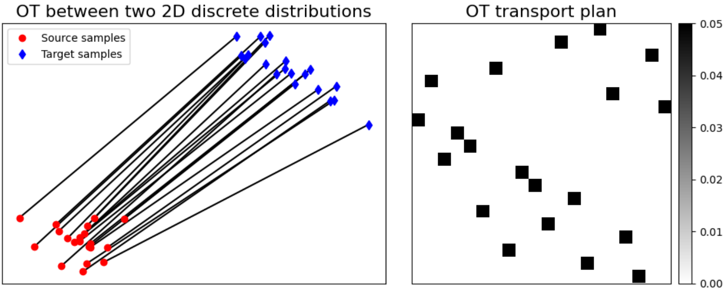
with : $U(\alpha,\beta) = \{\pi \in \mathcal{P}(\mathcal{X}\times\mathcal{Y}), \int_{\mathcal{Y}} \pi(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{y} = \alpha, \int_{\mathcal{X}} \pi(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{x} = \beta\}$

Discrete Optimal Transport

Discrete ingredients

- Discrete distributions $\alpha = \sum_{i=1}^{n} a_i \delta_{\boldsymbol{x}_i}$ and $\beta = \sum_{j=1}^{n} b_j \delta_{\boldsymbol{y}_j}$
- Cost matrix $C = C(X, Y)$, such that $C_{i,j} = c(\boldsymbol{x}_i, \boldsymbol{y}_j)$

Discrete Optimal Transport



For discrete distributions, OT becomes a linear program:

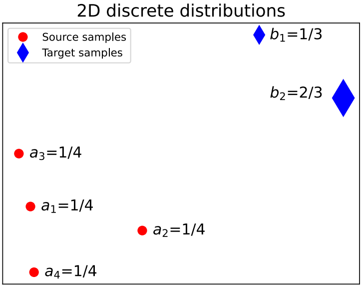
Definition (Discrete Optimal Transport)

$$\mathrm{OT}(\alpha, \beta, C) = \min_{\Pi \in U(\mathbf{a}, \mathbf{b})} \sum_{i,j} \Pi_{i,j} C_{i,j}$$

$$U(\mathbf{a}, \mathbf{b}) = \left\{ \Pi \in \left(\mathbb{R}^{+}\right)^{n_1 \times n_2} \middle| \Pi \mathbf{1}_{n_1} = \mathbf{a}, \Pi^{\mathbf{T}} \mathbf{1}_{n_2} = \mathbf{b} \right\}$$

Example of optimal plan

Consider the following 2D example:

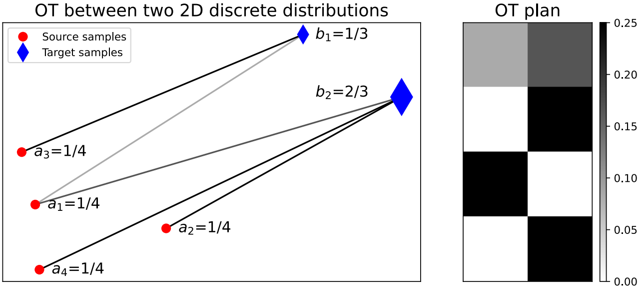


The probability distribution weights are:

$$\boldsymbol{a} = [1/4, 1/4, 1/4, 1/4]^\top$$
$$\boldsymbol{b} = [1/3, 2/3]^\top$$

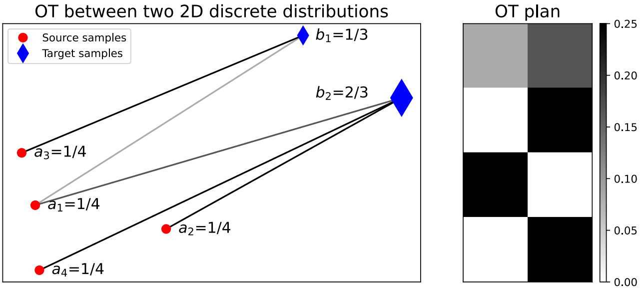
What is the optimal transport plan Π ?

Example of optimal plan



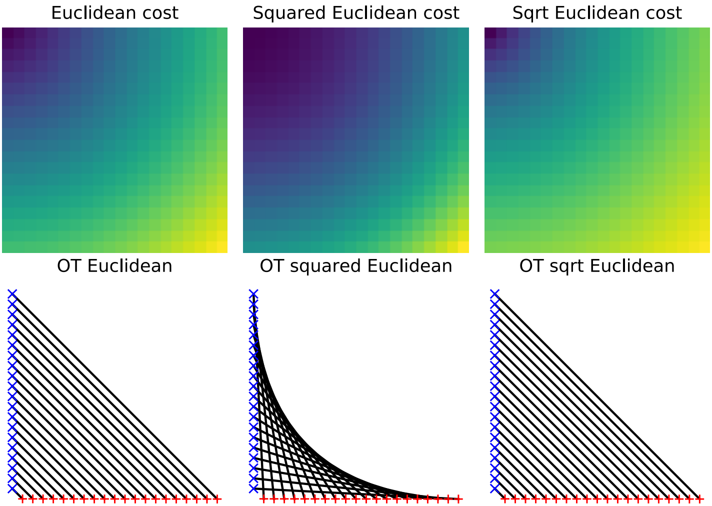
$$\Pi = \begin{bmatrix} 0.083 & 0.167 \\ 0 & 0.25 \\ 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \qquad \Pi \mathbf{1}_2 = \begin{bmatrix} 0.083 & 0.167 \\ 0 & 0.25 \\ 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \boldsymbol{a}$$

Example of optimal plan



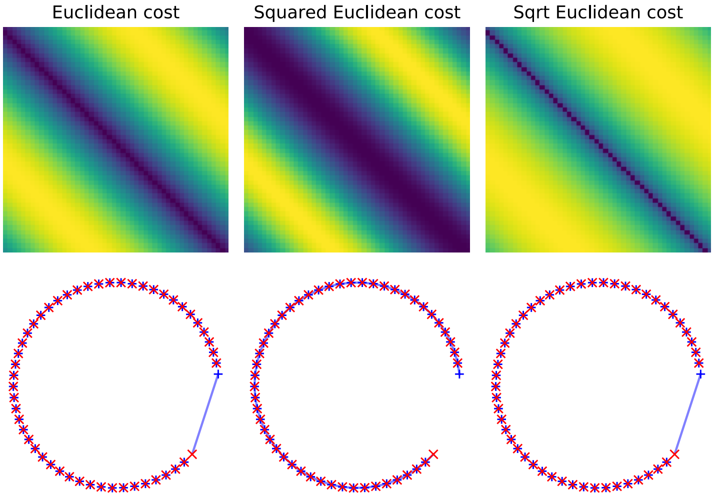
$$\Pi = \begin{bmatrix} 0.083 & 0.167 \\ 0 & 0.25 \\ 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \Pi^\top \mathbf{1}_2 = \begin{bmatrix} 0.083 & 0 & 0.25 & 0 \\ 0.167 & 0.25 & 0 & 0.25 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 2/3 \end{bmatrix} = \boldsymbol{b}$$

Optimal Transport connections



Computed with Python optimal Transport ! [Flamary et al., 2021]

Optimal Transport connections

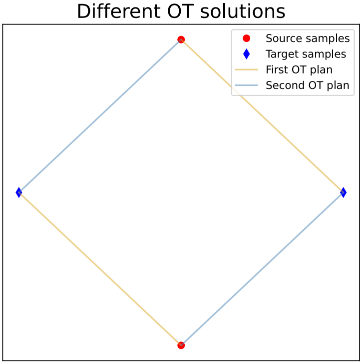


Computed with Python optimal Transport ! [Flamary et al., 2021]

Wasserstein distance

Some properties

- Leverages geometry of sample spaces through C
- A solution always exists (ex. $\pi = \alpha \otimes \beta$)
- $\langle \Pi, C \rangle_F$ is linear in the transport plan and in the cost
- Convex in the transport plan Π



Wasserstein distance

Some properties

- Leverages geometry of sample spaces through C
- A solution always exists (ex. $\pi = \alpha \otimes \beta$)
- $\langle \Pi, C \rangle_F$ is linear in the transport plan and in the cost
- Convex in the transport plan Π

Definition (Wasserstein distance)

C is a ground metric, then OT cost W_p is a metric for $p \geq 1$ and where

$$W_p(\alpha, \beta, C^p) = \left(\min_{\Pi \in U(\mathbf{a}, \mathbf{b})} \langle \Pi, C^p \rangle_F \right)^{1/p}$$

Proposition (Weak convergence)

The Wasserstein distance metrizes the weak convergence.

$$W_p(\delta_{\frac{1}{n}}, \delta_0, c) = c(\delta_{\frac{1}{n}}, \delta_0)$$

Dual of optimal transport

Optimal Transport has a dual program:

Proposition (Kantorovich duality)

$$\mathcal{L}(\alpha, \beta, c) = \sup_{(f,g) \in \mathcal{R}(c)} \int_{\mathcal{X}} f(\boldsymbol{x}) d\alpha(\boldsymbol{x}) + \int_{\mathcal{Y}} g(\boldsymbol{y}) d\beta(\boldsymbol{y}).$$

Where the set of admissible dual potentials is :

$$\mathcal{R}(c) = \{(f, g) \in \mathcal{C}(\mathcal{X}) \times \mathcal{C}(\mathcal{Y}) : \forall (\boldsymbol{x}, \boldsymbol{y}), f(\boldsymbol{x}) + g(\boldsymbol{y}) \leq c(\boldsymbol{x}, \boldsymbol{y})\}.$$

Proposition (Discrete Kantorovich duality)

$$\mathcal{L}(\alpha, \beta, C) = \max_{(f,g) \in \mathcal{R}(C)} \langle f, \mathbf{a} \rangle + \langle g, \mathbf{b} \rangle.$$

Where the set of admissible dual potentials is :

$$\mathcal{R}(C) = \{(f, g) \in \mathbb{R}^n \times \mathbb{R}^n : \forall (i, j) \in [\![n]\!]^2, f_i + g_j \leq C_{i,j}\}.$$

Can be solved with simplex algorithm with complexity of $\mathcal{O}(n^3 log(n))$.

Kantorovich-Rubinstein duality theorem

For the case of the Wasserstein-1 distance, we have:

Proposition (Kantorovich–Rubinstein duality)

$$W_1(\alpha, \beta, C) = \sup_{f \in Lip^1(\mathcal{X})} \mathbb{E}_{\boldsymbol{x} \sim \alpha}[f(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z} \sim \beta}[f(\boldsymbol{z})].$$

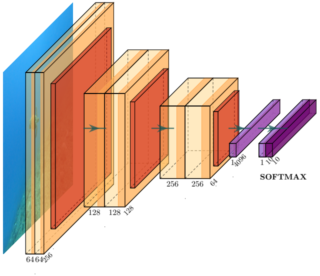
Supremum is intractable $\mapsto$ approximate it with a neural network.

Suppose α is the probability distributions of real images and β_θ is a parametric distribution we want to fit to α. We want to minimize

$$\begin{aligned}
\min_{\theta \in \Theta} W_1(\alpha, \beta_\theta, C) &= \min_{\theta \in \Theta} \sup_{f \in \mathtt{Lip}^1(\mathcal{X})} \mathbb{E}_{\boldsymbol{x} \sim \alpha}[f(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z} \sim \beta}[f(\boldsymbol{z})], \\
&\approx \min_{\theta \in \Theta} \max_{\phi \in \Phi} \mathbb{E}_{\boldsymbol{x} \sim \alpha}[f_\phi(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z} \sim \beta}[f_\phi(\boldsymbol{z})].
\end{aligned}$$

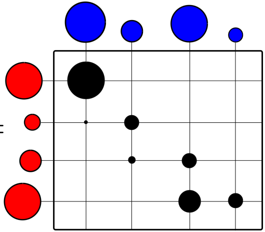
Where Φ is compact. To ensure Lipschitz constraint WGAN clips weights and WGAN-GP uses a gradient penalty.

Summary on neural networks and optimal transport



- **Summary on neural networks**
  - Neural networks are stacked layers of neurons
  - Competitive methods on classification, domain adaptation and GANs

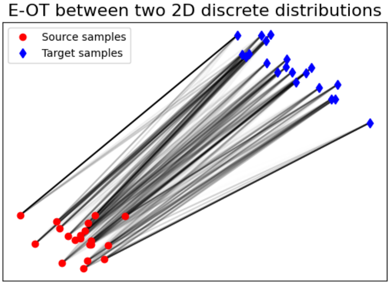
$\Pi =$

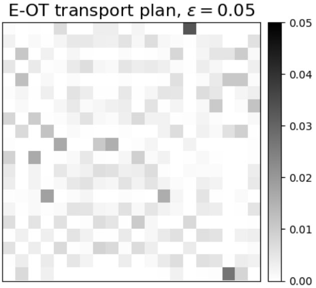


- **Summary on optimal transport**
  - \+ Loss function/distance between distributions of samples
  - \+ Leverages geometry of sample spaces through C
  - \- Cubical computational complexity of discrete OT
  - \+ Useful dual formulations

Entropic Optimal Transport

Entropic Optimal Transport





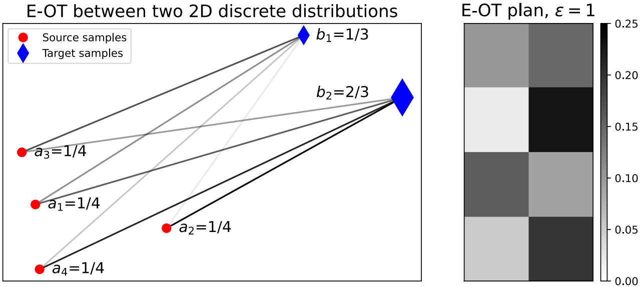
Definition (Entropic Optimal Transport [Cuturi, 2013])

$$\mathrm{OT}^{\varepsilon}(\alpha, \beta, C) = \min_{\Pi \in U(\mathbf{a}, \mathbf{b})} \sum_{i,j} \Pi_{i,j} C_{i,j} + \varepsilon \, \mathrm{KL}(\Pi | \mathbf{a} \otimes \mathbf{b})$$

$$\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}_{+}^{n}, \mathrm{KL}(\boldsymbol{x}|\boldsymbol{y}) = \sum_{i} \boldsymbol{x}_i \log\left(\frac{\boldsymbol{x}_i}{\boldsymbol{y}_i}\right) - \boldsymbol{x}_i + \boldsymbol{y}_i$$

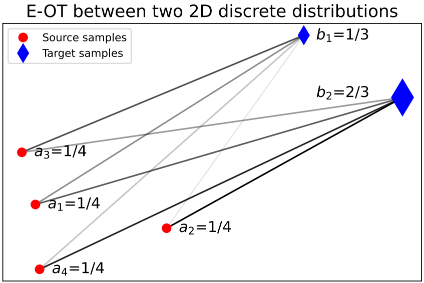
- Functional is strongly convex in the transport plan
- Computational complexity of entropic OT is $\mathcal{O}\left(\frac{n^2}{\varepsilon}\right)$

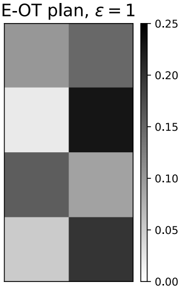
Example of optimal plan



$$\Pi = \begin{bmatrix} 0.10 & 0.15 \\ 0.02 & 0.23 \\ 0.16 & 0.09 \\ 0.05 & 0.20 \end{bmatrix} \qquad \Pi \mathbf{1}_2 = \begin{bmatrix} 0.10 & 0.15 \\ 0.02 & 0.23 \\ 0.16 & 0.09 \\ 0.05 & 0.20 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \boldsymbol{a}$$

Example of optimal plan





$$\Pi = \begin{bmatrix} 0.10 & 0.15 \\ 0.02 & 0.23 \\ 0.16 & 0.09 \\ 0.05 & 0.20 \end{bmatrix} \Pi^\top \mathbf{1}_2 = \begin{bmatrix} 0.10 & 0.02 & 0.16 & 0.05 \\ 0.15 & 0.23 & 0.09 & 0.20 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 2/3 \end{bmatrix} = \boldsymbol{b}$$

Form of the solution

Proposition (Convergence with ε)

We denote Π^ε the optimal transport plan of entropic OT. We have the following convergence property:

$$\mathrm{OT}^\varepsilon(\alpha,\beta,C) \underset{\varepsilon\to 0}{\to} \mathrm{OT}(\alpha,\beta,C)$$

$$\Pi^\varepsilon \underset{\varepsilon\to+\infty}{\to} \boldsymbol{a}\otimes\boldsymbol{b}$$

Proposition (Solution of the regularized Kantarovich problem)

The solution of the regularized (entropic) Kantarovich problem has the form:

$$\forall(i,j)\in[\![n]\!]\times[\![m]\!], P^\varepsilon_{i,j} = u_i \exp(-C/\varepsilon)_{i,j} v_j$$

for 2 unknown scaling variable $(u,v)\in\mathbb{R}^n_+\times\mathbb{R}^m_+$.

Sinkhorn algorithm

Algorithm 1 Pseudo-code Sinkhorn-Knopp algorithm

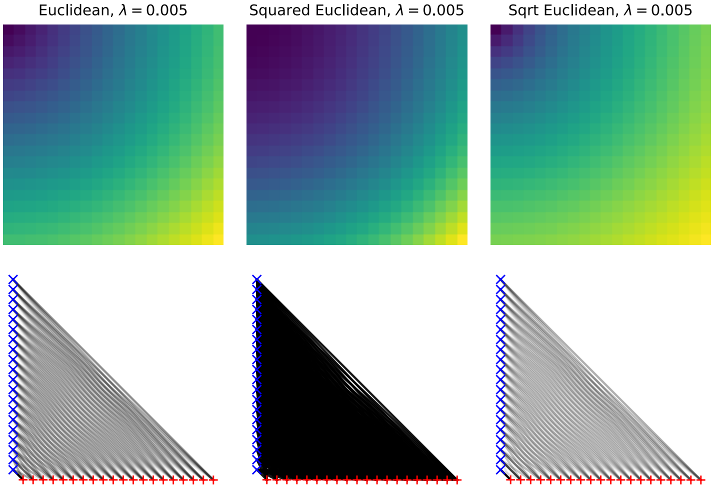
Require: Inputs : weights $(\mathbf{a}, \mathbf{b})$, cost matrix C, coefficient ε

```
u^(0) ← R^n_+
K ← exp(−C/ε)
for i in 1, ··· , κ do
    v^(i) ← b ⊘ K^T u^(i−1)
    u^(i) ← a ⊘ K v^(i)
end for
return Π = diag(u^(κ)) K diag(v^(κ))
```

$$u^{(0)} \leftarrow \mathbb{R}^n_+$$
$$\boldsymbol{K} \leftarrow \exp(-\boldsymbol{C}/\varepsilon)$$
for i in $1, \cdots, \kappa$ **do**
$$v^{(i)} \leftarrow \mathbf{b} \oslash \boldsymbol{K}^T u^{(i-1)}$$
$$u^{(i)} \leftarrow \mathbf{a} \oslash \boldsymbol{K} v^{(i)}$$
end for
return $\Pi = \mathrm{diag}(u^{(\kappa)})\boldsymbol{K}\,\mathrm{diag}(v^{(\kappa)})$

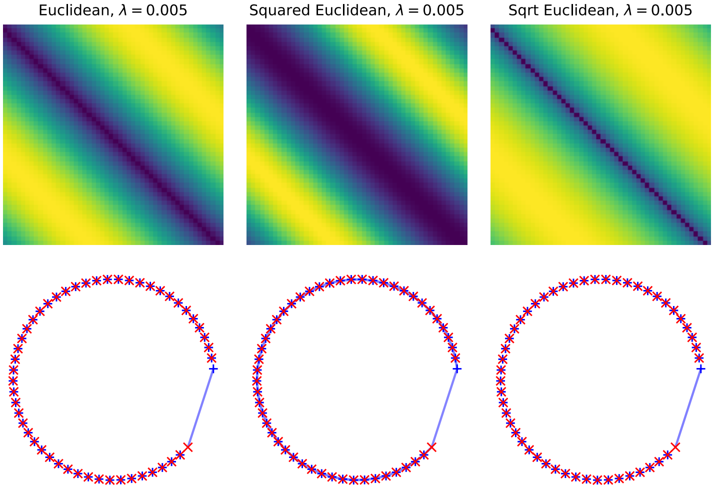
- The algorithm performs alternatively a scaling along the rows and columns of $\boldsymbol{K}$ to match the desired marginals
- Computational complexity $\mathcal{O}(\kappa n^2)$
- Fast implementation in parallel (GPU)

Optimal Transport connections



Computed with Python optimal Transport ! [Flamary et al., 2021]

Optimal Transport connections



Computed with Python optimal Transport ! [Flamary et al., 2021]

Dual of entropic optimal transport

Optimal Transport has a dual program:

Proposition (entropic OT duality)

$$\mathrm{OT}^{\varepsilon}(\alpha, \beta, C) = \max_{(f,g) \in (\mathbb{R}^n)^2} \langle f, \mathbf{a} \rangle + \langle g, \mathbf{b} \rangle - \varepsilon \langle e^{f/\varepsilon}, K e^{g/\varepsilon} \rangle.$$

Note the unconstrained dual contrary to exact OT.

The optimal (f, g) are linked to scalings (u, v) appearing in the Sinkhorn algorithm through

$$(u, v) = (e^{f/\varepsilon}, e^{g/\varepsilon}) \tag{1}$$

Derivative of entropic optimal transport

Proposition (Derivative with respect to weights)

For $\varepsilon > 0$, $(\boldsymbol{a}, \boldsymbol{b}) \mapsto \mathrm{OT}^\varepsilon((\boldsymbol{a}, \boldsymbol{X}), (\boldsymbol{b}, \boldsymbol{Y}), C)$ *is differentiable. Its gradient reads*

$$\nabla \mathrm{OT}^\varepsilon((\boldsymbol{a}, \boldsymbol{X}), (\boldsymbol{b}, \boldsymbol{Y}), C) = (f, g)$$

where (f, g) *is the unique solution, centered such that* $\sum_i f_i = \sum_j g_j = 0$*. For* $\varepsilon = 0$*, this formula defines the elements of the sub-differential.*

Proposition (Derivative with respect to the cost)

For fixed input histograms $(\boldsymbol{a}, \boldsymbol{b})$*, for* $\varepsilon > 0$*, the mapping* $C \mapsto \mathrm{OT}^\varepsilon((\boldsymbol{a}, \boldsymbol{X}), (\boldsymbol{b}, \boldsymbol{Y}), C)$ *is smooth, and*

$$\nabla_C \mathrm{OT}^\varepsilon((\boldsymbol{a}, \boldsymbol{X}), (\boldsymbol{b}, \boldsymbol{Y}), C) = \Pi^\varepsilon$$

For $\varepsilon = 0$*, this formula defines the set of upper gradients.*

Limits of entropic optimal transport

Unfortunately, entropic OT is not a distance.

Proposition (Entropic OT losses distance properties)

$$\mathrm{OT}^{\varepsilon}(\alpha, \alpha, C) > 0.$$

We can nonetheless define a new loss function called the Sinkhorn divergence as:

Proposition (Sinkhorn divergences)

$$\mathrm{S}^{\varepsilon}(\alpha, \beta, C) = \mathrm{OT}^{\varepsilon}(\alpha, \beta, C) - \frac{1}{2}(\mathrm{OT}^{\varepsilon}(\alpha, \alpha, C) + \mathrm{OT}^{\varepsilon}(\beta, \beta, C)).$$

The Sinkhorn divergence defines a divergence between probability measures [Feydy et al., 2019] and interpolate between OT and MMD [Gretton et al., 2012]. It has also better statistical properties than OT.

Unbalanced Optimal Transport

Unbalanced Optimal Transport

Definition

Unbalanced Optimal transport measures the distance between distributions, but with relaxed marginals.

$$\mathrm{UOT}^{\tau,\varepsilon}(\alpha,\beta,c) = \min_{\pi\in\mathcal{M}_+(\mathcal{X}\times\mathcal{Y})} \int c d\pi + \tau(\mathtt{KL}(\pi_1\|\alpha) + \mathtt{KL}(\pi_2\|\beta)),$$

where π is the transport plan, π_1 and π_2 the plan's marginals, $\tau \geq 0$ is the marginal penalization and $\varepsilon \geq 0$ is the regularization coefficient.

Difference with OT

- $\pi \in U(\alpha,\beta) \longrightarrow \pi \in \mathcal{M}_+(\mathcal{X}\times\mathcal{Y})$
- Fixed marginal constraints are replaced by $\mathtt{KL}(\pi_1\|\alpha)$ penalties
- Unique marginals π_1 and π_2
- `KL` can be replaced by `TV`

Entropic unbalanced Optimal Transport

Definition

Entropic unbalanced Optimal transport measures the distance between distributions, but with relaxed marginals.

$$\mathrm{UOT}^{\tau,\varepsilon}(\alpha,\beta,c) = \min_{\pi\in\mathcal{M}_+(\mathcal{X}\times\mathcal{Y})} \int c d\pi + \varepsilon \mathtt{KL}(\pi|\alpha\otimes\beta)$$
$$+\tau(\mathtt{KL}(\pi_1\|\alpha)+\mathtt{KL}(\pi_2\|\beta)),$$

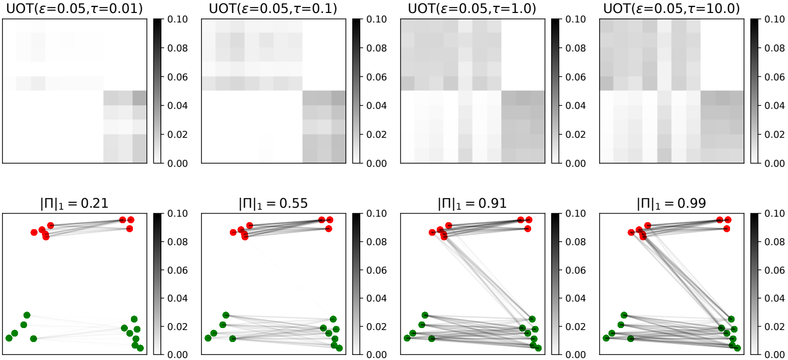
where π is the transport plan, π_1 and π_2 the plan's marginals, $\tau \geq 0$ is the marginal penalization and $\varepsilon \geq 0$ is the regularization coefficient.

Difference with UOT

- Unique solution Π
- Can be solved with a generalized Sinkhorn algorithm
- $\mathrm{UOT}^{\tau,\varepsilon}(\alpha,\alpha,c) > 0$ but can define a Sinkhorn UOT variant [Séjourné et al., 2019]

Influence of τ

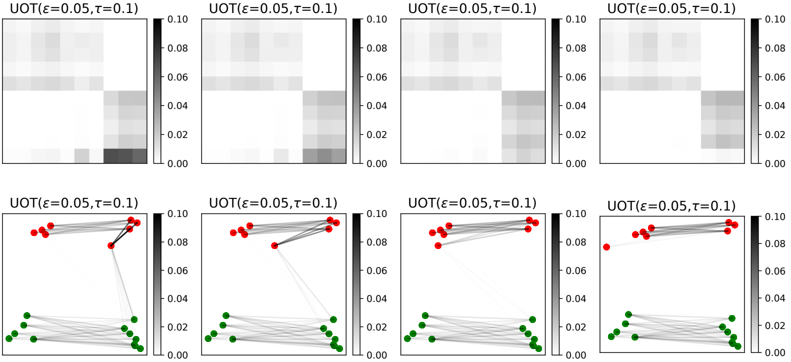
Let us study the optimal transport plan for a fixed problem and a various τ.



Key message: Smaller τ decreases the transported mass as it is less costly to be "lazy".

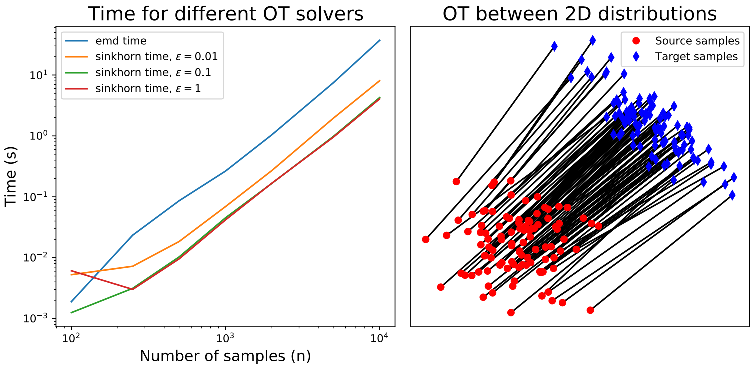
Influence of higher cost

Let us study the optimal transport plan for a dynamic problem and a fixed τ.



Key message: The more costly a sample is to transport, the less it is transported.

Time experiment



Limits

Can not be used in Big Data scenario !

Minibatch Optimal Transport

Minibatch Optimal Transport definition

Let $m \leq n$, [Damodaran et al., 2018, Genevay et al., 2018] compute optimal transport between minibatch of distributions.

Minibatch strategy

- Select m samples without replacement at random in domains
- Compute OT between the minibatches
- Average several MBOT terms $\rightarrow$ complexity $\mathcal{O}(m^3)$

Minibatch Optimal Transport definition

Expectation of minibatches

Computing OT kernel h between minibatches estimates:

$$E_h(\alpha, \beta, C) := \mathbb{E}_{(X,Y)\sim\alpha^{\otimes m}\otimes\beta^{\otimes m}}[h(\mu_m, \mu_m, C(X,Y))]$$

- Can be defined for OT variants h
- Justified in [Fatras et al., 2020]

Estimate minibatch OT distance

Definition (Complete minibatch estimator)

$$\overline{h}^m(X,Y) := \binom{n}{m}^{-2} \sum_{I,J \in \mathcal{P}_m} h(\mu_m, \mu_m, C_{I,J})$$

$$\Pi^m(X,Y) := \binom{n}{m}^{-2} \sum_{I,J \in \mathcal{P}_m} \Pi_{I,J}$$

- where $\mathcal{P}_m$ is the set of all m-tuples without replacement
- $\Pi^m(X,Y)$ is an admissible transport plan between the input probability distributions $\Pi \in U(\mu_n, \mu_n)$

Definition (Incomplete minibatch estimator)

$$\widetilde{h}_k^m(X,Y) := k^{-1} \sum_{(I,J) \in D_k} h(\mu_m, \mu_m, C_{I,J})$$

where $k > 0$ is an integer and D_k is a set of cardinality k whose elements are minibatches drawn at random

1D Minibatch Optimal Transport closed-form

From the 1D OT closed-form formula, we have:

$$\pi_{j,k} = \frac{1}{m}\binom{n}{m}^{-2}\sum_{i=i_{\min}}^{i_{\max}}\binom{j-1}{i-1}\binom{k-1}{i-1}\binom{n-j}{m-i}\binom{n-k}{m-i}$$

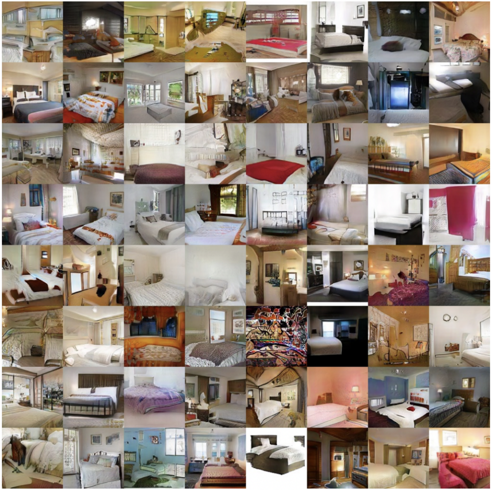
where $i_{\min} = \max(0, m-n+j, m-n+k)$ and $i_{\max} = \min(j, k)$

A few key home message on minibatch OT.

- Not a distance
- Can not define a divergence like Sinkhorn divergence
- Better statistical properties
- A new loss function based on OT but not OT

Applications

Generative models



Taken from [Gulrajani et al., 2017].

Office Home Domain Adaptation dataset

Network : pre-trained ResNet 50 with an additional classification layer.

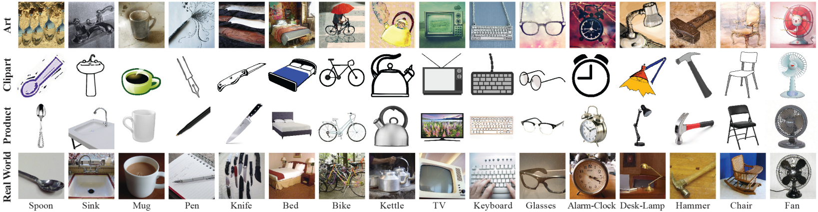


Figure taken from [Venkateswara et al., 2017]. 65 classes in the source and target domains for balanced DA and 25 classes in the target domains for partial DA.

Domain Adaptation experiments

	Method	A-C	A-P	A-R	C-A	C-P	C-R	P-A	P-C	P-R	R-A	R-C	R-P	avg
DA	RESNET-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
	DANN (*)	44.3	59.8	69.8	48.0	58.3	63.0	49.7	42.7	70.6	64.0	51.7	78.3	58.3
	CDAN-E(*)	52.5	71.4	76.1	59.7	69.9	71.5	58.7	50.3	77.5	70.5	57.9	**83.5**	66.6
	DEEPJDOT (*)	50.7	68.6	74.4	59.9	65.8	68.1	55.2	46.3	73.8	66.0	54.9	78.3	63.5
	ALDA (*)	52.2	69.3	76.4	58.7	68.2	71.1	57.4	49.6	76.8	70.6	57.3	82.5	65.8
	ROT (*)	47.2	71.8	76.4	58.6	68.1	70.2	56.5	45.0	75.8	69.4	52.1	80.6	64.3
	JUMBOT	**55.2**	**75.5**	**80.8**	**65.5**	**74.4**	**74.9**	**65.2**	**52.7**	**79.2**	**73.0**	**59.9**	83.4	**70.0**
PDA	RESNET-50	46.3	67.5	75.9	59.1	59.9	62.7	58.2	41.8	74.9	67.4	48.2	74.2	61.4
	DEEPJDOT(*)	48.2	66.2	76.6	56.1	57.8	64.5	58.3	42.7	73.5	65.7	48.2	73.7	60.9
	PADA	51.9	67.0	78.7	52.2	53.8	59.0	52.6	43.2	78.8	73.7	56.6	77.1	62.1
	ETN	59.2	77.0	79.5	62.9	65.7	75.0	68.3	55.4	84.4	75.7	57.7	**84.5**	70.4
	BA3US(*)	56.7	76.0	**84.8**	73.9	67.8	**83.7**	72.7	56.5	84.9	77.8	64.5	83.8	73.6
	JUMBOT	**62.7**	**77.5**	84.4	**76.0**	**73.3**	80.5	**74.7**	**60.8**	**85.1**	**80.2**	**66.5**	83.9	**75.5**

OT have state-of-the-art results [Fatras et al., 2021].

References i

[Cuturi, 2013] Cuturi, M. (2013).
Sinkhorn distances: Lightspeed computation of optimal transport.
In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 2292–2300. Curran Associates, Inc.

[Damodaran et al., 2018] Damodaran, B. B., Kellenberger, B., Flamary, R., Tuia, D., and Courty, N. (2018).
DeepJDOT: Deep Joint Distribution Optimal Transport for Unsupervised Domain Adaptation.
In *ECCV 2018 - 15th European Conference on Computer Vision*. Springer.

[Fatras et al., 2021] Fatras, K., Séjourné, T., Courty, N., and Flamary, R. (2021).
Unbalanced minibatch optimal transport; applications to domain adaptation.
In *Proceedings of the 38th International Conference on Machine Learning*.

[Fatras et al., 2020] Fatras, K., Zine, Y., Flamary, R., Gribonval, R., and Courty, N. (2020).
Learning with minibatch wasserstein: asymptotic and gradient properties.
In *AISTATS*.

[Feydy et al., 2019] Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-i., Trouve, A., and Peyré, G. (2019).
Interpolating between optimal transport and mmd using sinkhorn divergences.
In *Proceedings of Machine Learning Research*.

[Flamary et al., 2021] Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. (2021).
Pot: Python optimal transport.
Journal of Machine Learning Research, 22(78):1–8.

[Genevay et al., 2018] Genevay, A., Peyre, G., and Cuturi, M. (2018).
Learning generative models with sinkhorn divergences.
In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).
Generative adversarial nets.
In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

[Gretton et al., 2012] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012).
A kernel two-sample test.
J. Mach. Learn. Res., 13(null):723–773.

[Gulrajani et al., 2017] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017).
Improved training of wasserstein gans.
In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Kantorovich, 1942] Kantorovich, L. V. (1942).
On translation of mass (in russian).
Proceedings of the USSR Academy of Sciences.

[Séjourné et al., 2019] Séjourné, T., Feydy, J., Vialard, F.-X., Trouvé, A., and Peyré, G. (2019).
Sinkhorn divergences for unbalanced optimal transport.

[Venkateswara et al., 2017] Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. (2017).
Deep hashing network for unsupervised domain adaptation.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027.