Optimal Transport and Deep Learning: Learning from one another

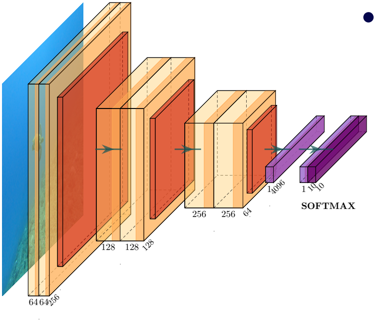
PhD defense

Kilian Fatras

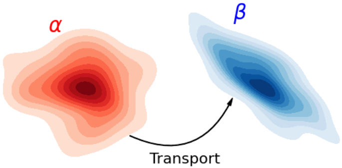
25th November 2021

Obelix and Panama Teams

Supervised by Nicolas Courty and Rémi Flamary



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

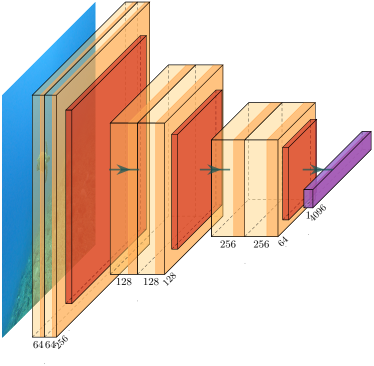


- **Introduction on optimal transport**
  - Definitions
  - Properties
  - Entropic variant

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 neurons
- Parameters of neural networks: connections between layers
- Different layers: convolutional layers, fully connected layers, ...



Motivating example: Classification

Classification problem: predicting the class of a given image

Hawskbill turtle

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 (

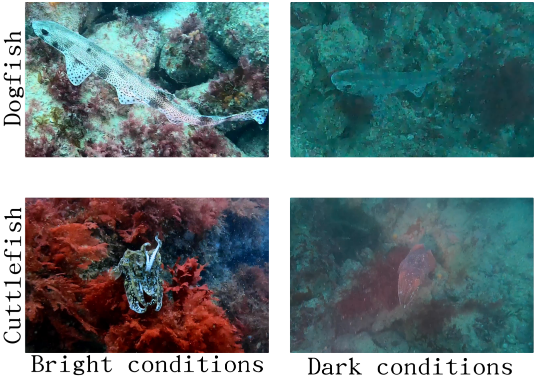
)

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

- 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

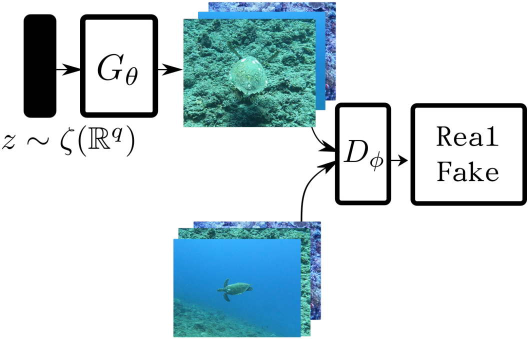


Domain adaptation (DA) setting

- 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

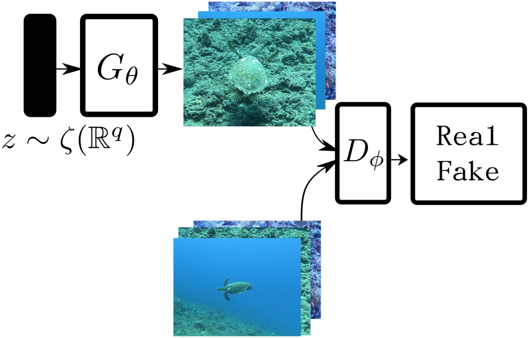
Goal: generating new images



- 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

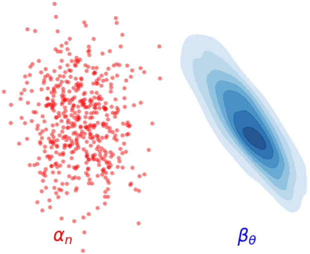
Training samples as distributions paradigm

Applications use probability distributions to train neural networks

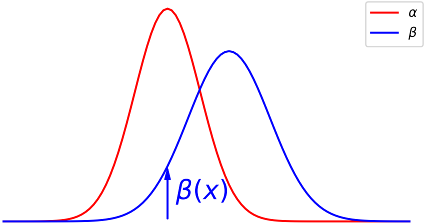
- Classification: function L takes probability vectors as inputs
- Domain adaptation: align embedding probability distributions from domains
- GANs: distance between generated and training distributions

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

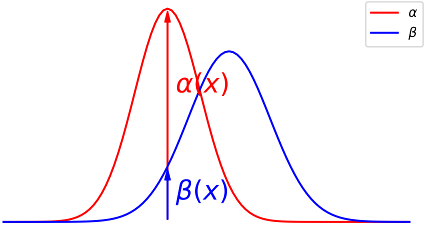
Goal : Find a suitable function L between probability distributions



Comparing probability distributions



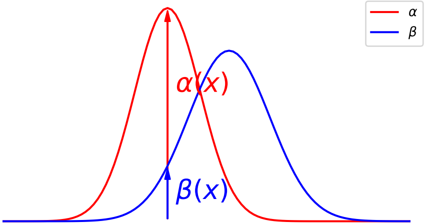
Comparing probability distributions



φ-divergences compare mass ratio point-wise $\alpha(\boldsymbol{x})/\beta(\boldsymbol{x})$ ($\beta(\boldsymbol{x}) > 0$)

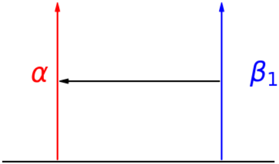
$$L_\varphi(\alpha|\beta) = \int_{\mathcal{X}} \varphi\left(\frac{d\alpha}{d\beta}\right) d\beta$$

Comparing probability distributions



φ-divergences compare mass ratio point-wise $\alpha(\boldsymbol{x})/\beta(\boldsymbol{x})$ $(\beta(\boldsymbol{x}) > 0)$

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

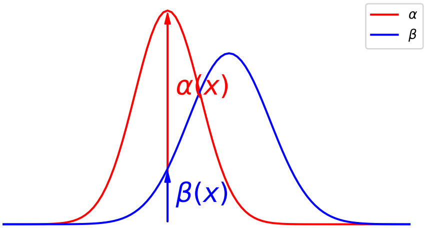


- φ-divergences cannot compare Diracs

$\to$ fail to capture the geometry

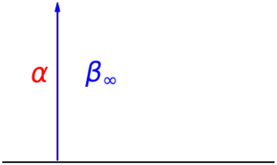
- $\mathrm{KL}(\alpha|\beta_t) = +\infty$

Comparing probability distributions



φ-divergences compare mass ratio point-wise $\alpha(\boldsymbol{x})/\beta(\boldsymbol{x})$ $(\beta(\boldsymbol{x}) > 0)$

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

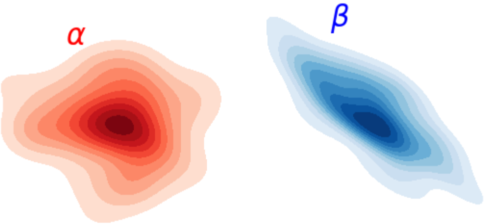


- φ-divergences cannot compare Diracs

$\rightarrow$ fail to capture the geometry

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

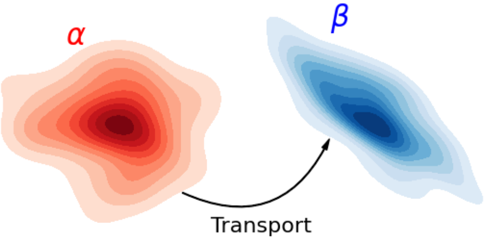
Optimal Transport definition



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

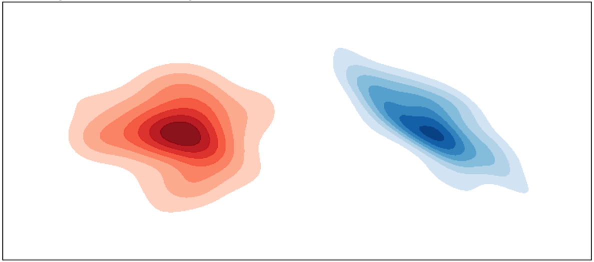
Optimal Transport definition



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

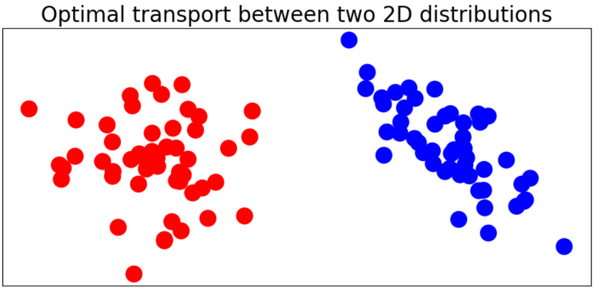


Definition (Kantorovich problem [Kantorovich, 1942])

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

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

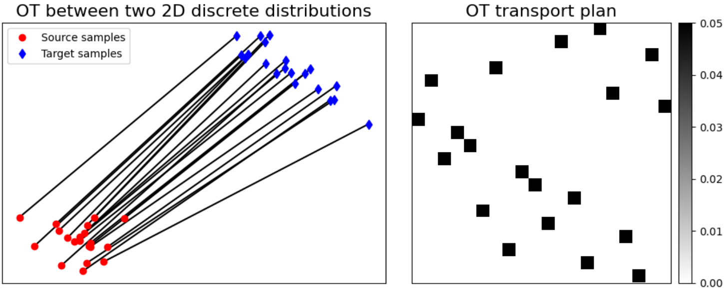
Discrete Optimal Transport



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)

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

$$U(\alpha, \beta) = \left\{ \Pi \in \left(\mathbb{R}^+\right)^{n \times n} | \Pi \mathbf{1}_n = \mathbf{a}, \Pi^{\mathbf{T}} \mathbf{1}_n = \mathbf{b} \right\}$$

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
- Computational complexity of discrete OT is $\mathcal{O}(n^3 log(n))$

Definition (Wasserstein distance)

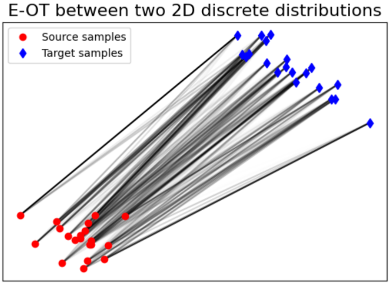
C is a ground metric, then OT cost W_p is a metric for $p \geq 1$ and where

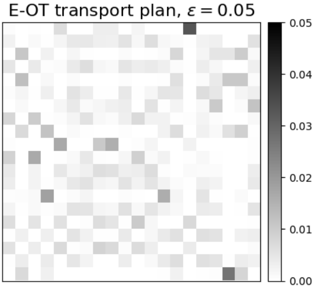
$$W_p(\alpha, \beta, C^p) = \left(\min_{\Pi \in U(\alpha,\beta)} \langle \Pi, C^p \rangle_F \right)^{1/p}$$

Proposition (Kantorovich–Rubinstein duality)

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

Entropic Optimal Transport



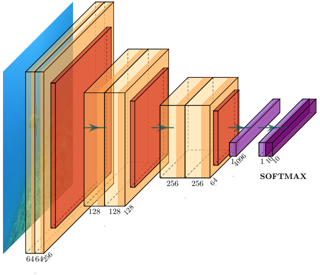


Definition (Entropic Optimal Transport [Cuturi, 2013])

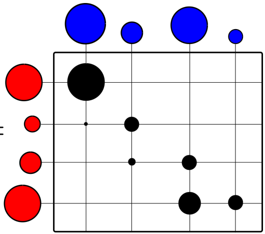
$$\mathrm{OT}^{\varepsilon}(\alpha, \beta, C) = \min_{\Pi \in U(\alpha,\beta)} \sum_{i,j} \Pi_{i,j} C_{i,j} + \varepsilon\, \mathrm{KL}(\Pi | \alpha \otimes \beta)$$

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

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



$\Pi =$



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

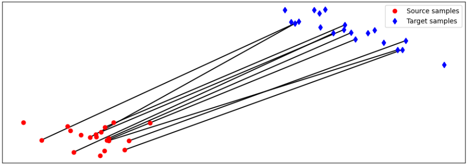
- **Summary on optimal transport**
  - \+ Loss function/distance between distributions of samples
  - \+ Leverages geometry of sample spaces through C
  - \- Cubical computational complexity of discrete OT
  - +/- Faster and easy computable entropic variant

Table of contents

- **Optimal transport as a loss function in deep learning**

Geometry on label distributions ← Optimal transport adversarial regularization for noisy labels

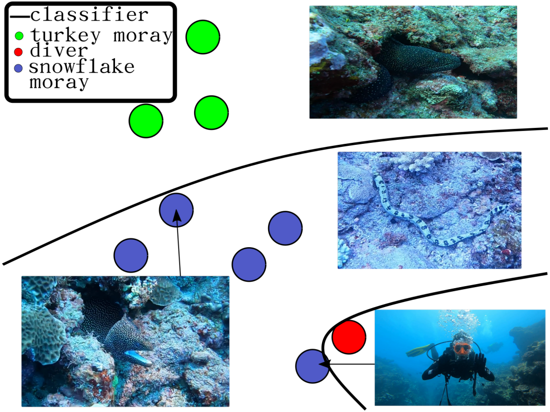
Geometry on sample distributions ← Optimal transport loss function to generate misclassified data



- **Minibatch optimal transport**
  - Minibatch optimal transport formalism
  - Unbalanced minibatch optimal transport

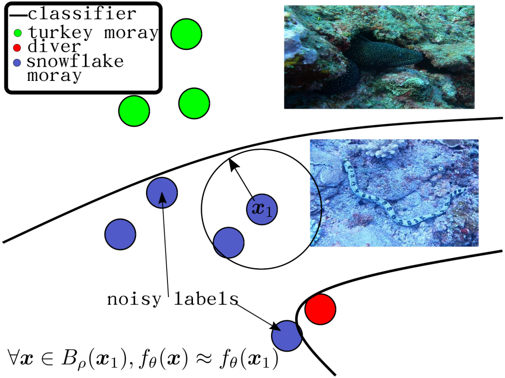
Wasserstein Adversarial Regularization for noisy labels

- Label noise: label does not correspond to the image class
- Occurs in real-world dataset → neural networks overfitting [Zhang et al., 2017].



- Enforce uniformity prediction around the vicinity of samples
- Use optimal transport in the regularization

Robust optimization illustration

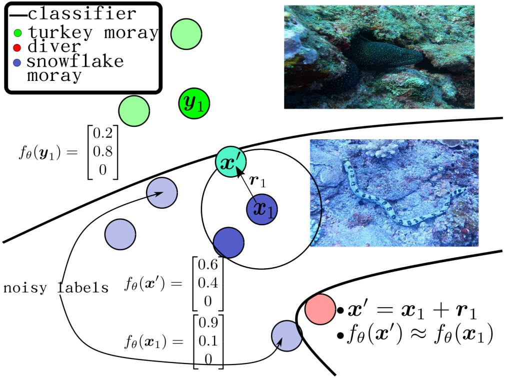


Robust optimization:

$$\operatorname{argmin}_\theta \sum_{i=1}^{n} \max_{\boldsymbol{x}_i^u \in B_\rho(\boldsymbol{x}_i)} L_{\mathrm{CE}}(f_\theta(\boldsymbol{x}_i^u), \boldsymbol{y}_i) \tag{1}$$

- Intuition: noisy labels mitigated by uniform prediction
- Cannot rely on labels $\rightarrow$ use prediction

Virtual adversarial training

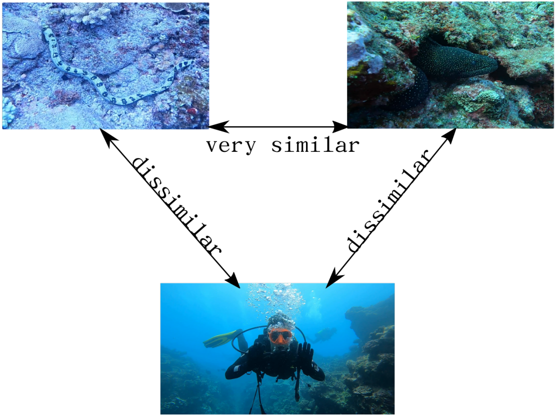


VAT's loss function [Miyato et al., 2018]:

$$\mathcal{L}_{\text{VAT}}((X,Y), f_\theta) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\mathrm{L}(\boldsymbol{y}_i, f_\theta(\boldsymbol{x}_i))}_{\text{Supervised loss}} + \beta \underbrace{\mathrm{KL}(\hat{f}_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_i + \boldsymbol{r}_i))}_{\text{regularization term}}$$

$$\text{where } \boldsymbol{r}_i = \underset{\boldsymbol{r}, \|\boldsymbol{r}\| \leq \rho}{\operatorname{argmax}} \ \mathrm{KL}(\hat{f}_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_i + \boldsymbol{r}))$$

Encoding class similarities



- KL divergence penalizes errors between classes in the same manner
- Replace KL divergence by optimal transport

Wasserstein adversarial regularization

The WAR loss function is:

$$\mathcal{L}_{\text{WAR}}((X,Y),f_\theta) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{L(\boldsymbol{y}_i, f_\theta(\boldsymbol{x}_i))}_{\text{Supervised loss}} + \beta \underbrace{\mathrm{OT}^{\varepsilon}_{\boldsymbol{C}}(\hat{f}_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_i+\boldsymbol{r}_i))}_{\text{regularization term}}$$

$$\text{where } \boldsymbol{r}_i = \underset{\boldsymbol{r}, \|\boldsymbol{r}\| \le \rho}{\text{argmax}}\ \mathrm{OT}^{\varepsilon}_{\boldsymbol{C}}(\hat{f}_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_i+\boldsymbol{r}))$$

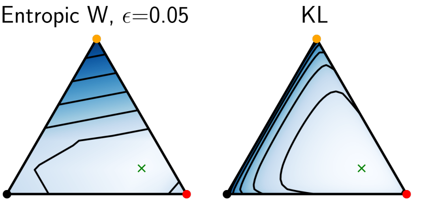
For the ground cost $\boldsymbol{C}$, we want:

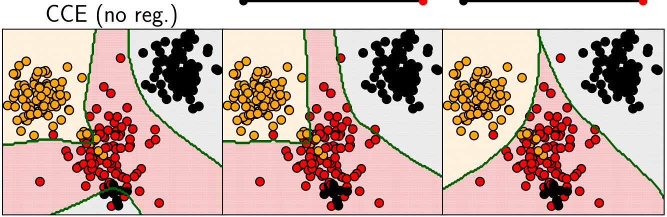
- High cost between close classes to get complex boundaries
- Small cost between non-similar classes to get smooth boundaries

The OT cost between labels was also studied in [Frogner et al., 2015]

Learning with WAR

$$\boldsymbol{C} = \begin{array}{cc} & \begin{array}{ccc} {\color{black}\bullet} & {\color{red}\bullet} & {\color{orange}\bullet} \end{array} \\ \begin{array}{c} {\color{black}\bullet} \\ {\color{red}\bullet} \\ {\color{orange}\bullet} \end{array} & \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 5 \\ 1 & 5 & 0 \end{pmatrix} \end{array}$$



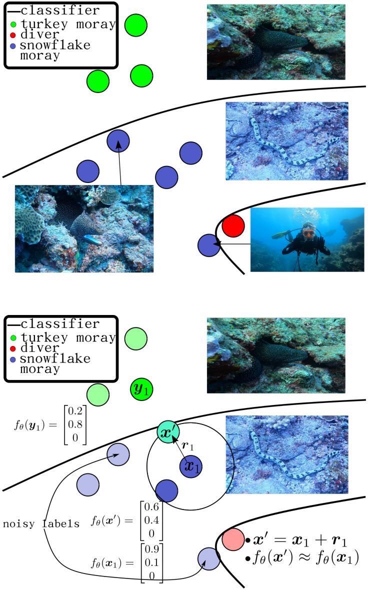


- Regularization geometry for different adversarial regularizations
- (Top) Regularization values on the simplex of class probabilities
- (Down) Classification boundaries for different methods

- Test accuracy (%) on Fashion-MNIST, CIFAR-10, and CIFAR-100 datasets
- Varying noise rates (20% and 40%)

Dataset /	noise	CCE	Bootsoft	CoTeaching	VAT	$\mathrm{WAR}_{\boldsymbol{C}}$
F-MNIST	20%	89.02±0.47	88.17±0.11	91.24±0.06	93.10±0.14	93.37±0.08
	40%	78.85±0.56	73.84±0.28	86.83±0.10	89.74±0.10	90.41±0.02
CIFAR-10	20%	85.26±0.09	85.35 ± 0.8	86.19 ±0.07	88.91±0.09	89.12±0.48
	40%	76.23±0.15	74.32 ± 0.2	80.87±0.09	81.98±0.25	84.55±0.78
CIFAR-100	20%	58.81±0.10	58.97±0.08	60.90±0.03	65.44±0.11	62.72±0.16
	40%	42.45±0.12	41.73±0.08	42.73±0.08	55.75±0.14	58.86±0.21

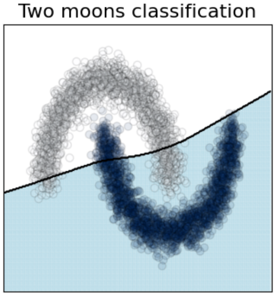
Summary on Wasserstein Adversarial Regularization

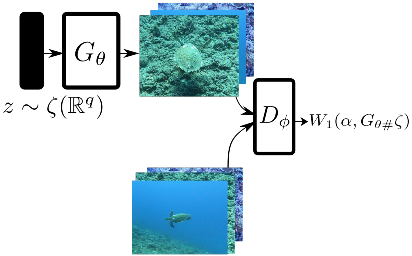


- Noisy labels are corrupted labels and hurt the performances of neural networks
- Promoting uniform classification around inputs mitigate their influence
- Integrate optimal transport to control the uniform classification

Published in [Fatras et al., 2021a]

Generating misclassified data

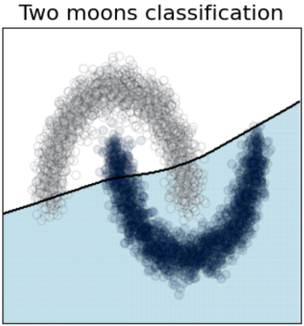




- Attack the classifier
- Introduction on misclassified examples
- How OT can be used to generate data
- Generating misclassified examples with just the output of the classifier

Generating misclassified data

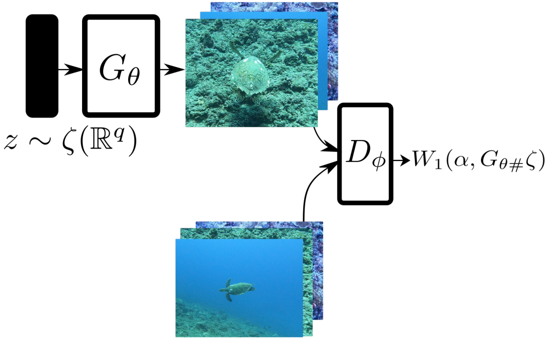
Our objective is to generate samples from the gray class which are classified as blue data



- Generate adversarial examples
- Most adversarial examples generation uses classifier architecture
- Some methods hurt quality of adversarial images

Generating data with Wasserstein GAN

To generate data, we use the *Wasserstein Generative Adversarial Networks* (WGAN) [Arjovsky et al., 2017]



We use the Kantorovich-Rubinstein duality theorem and we minimize:

$$\min_\theta W_1(\alpha, G_{\theta\#}\zeta) = \min_\theta \max_\phi \mathbb{E}_{\boldsymbol{x}\sim\alpha}[\mathcal{D}_\phi(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim\zeta}[\mathcal{D}_\phi(G_\theta(\boldsymbol{z}))]$$

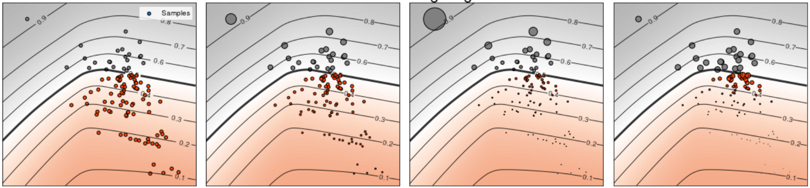
$\rightarrow \mathcal{D}_\phi$ needs to be 1-Lipschitz

Defining a new distribution

Create a new distribution which gives bigger weights to misclassified data, $\frac{1}{n}\sum_{i=1}^{n}\delta_{\boldsymbol{x}_i} \to \sum_{i=1}^{n} a_i\delta_{\boldsymbol{x}_i}, \sum_{i=1}^{n} a_i = 1$

- Hard weighting (only consider misclassified samples)
- Soft weighting (weight depends on how much the sample is misclassified)

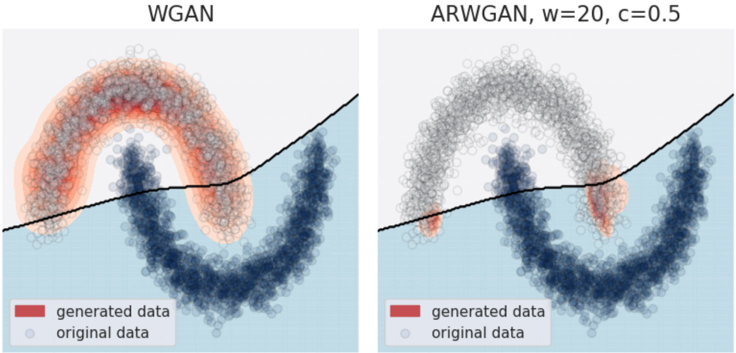
Illustration of the reweighting scheme



(a) Initial Classifier (b) softmax, $w = 5$, $\kappa = 1.0$ (c) softmax, $w = 10$, $\kappa = 1.0$ (d) softmax, $w = 20$, $\kappa = 0.5$

Generating misclassified data

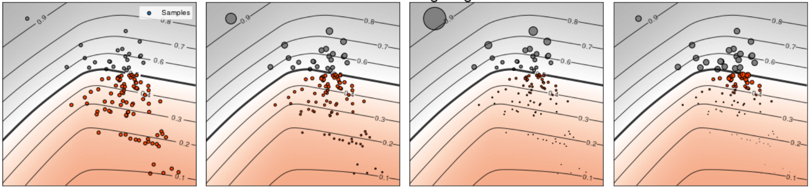
- Fool classifiers on hyperspectral images
- Transfer misclassified examples to unseen classifiers
- Can modify images to fool classifier
- Can fool state of the art detector Yolo V3



- Attacked the classifier
- Used WGAN to generate samples
- Created new empirical distributions
- Applied on remote sensing data

Published in [Burnel et al., 2021, Burnel et al., 2020]

Illustration of the reweighting scheme



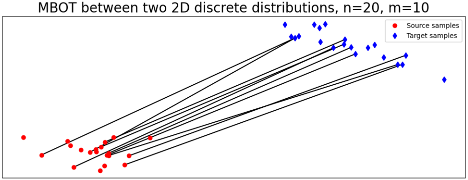
(a) Initial Classifier (b) softmax, $w = 5$, $\kappa = 1.0$ (c) softmax, $w = 10$, $\kappa = 1.0$ (d) softmax, $w = 20$, $\kappa = 0.5$

Table of contents

- **Optimal transport as a loss function in deep learning**

Geometry on label distributions ← Optimal transport adversarial regularization for noisy labels

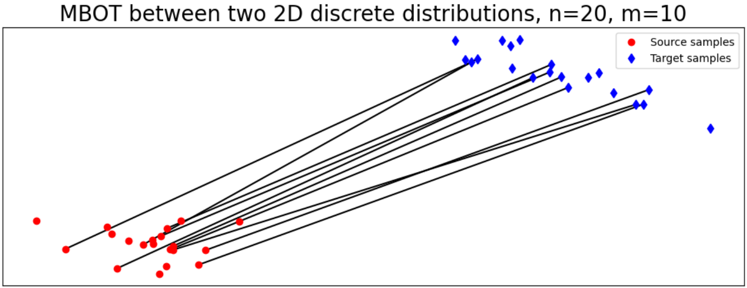
Geometry on sample distributions ← Optimal transport loss function to generate misclassified data



- **Minibatch optimal transport**
  - Minibatch optimal transport formalism
  - Unbalanced minibatch optimal transport transport

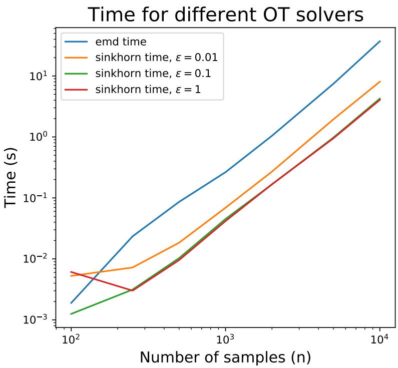
Minibatch optimal transport

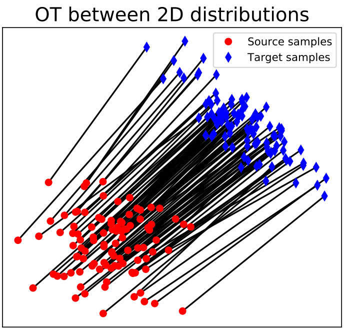
- Minibatch optimal transport formalism
- Loss properties
- Statistical and optimization properties



Time experiment

Optimal transport can be computed between a lot of samples depending on the application

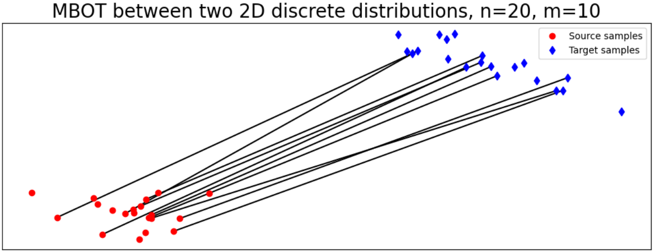




Limits

Can not be used in Big Data scenario !

Minibatch Optimal Transport definition

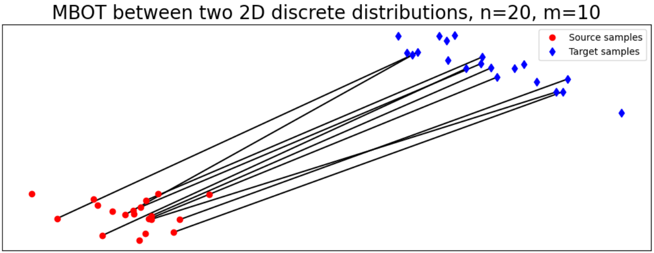


Let $m \leq n$, [Damodaran et al., 2018, Genevay et al., 2018] compute optimal transport between minibatch (MBOT) of distributions

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 any OT variants h (Gromov-Wasserstein distance, Sliced Wasserstein distance, ...)

Estimate minibatch OT distance

Definition (Complete minibatch estimator)

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

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

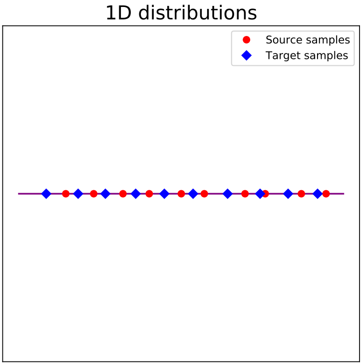
- where $\mathcal{P}_m$ is the set of all m-tuples without replacement
- $\Pi^m(X,Y)$ is an admissible transport plan between the input probability distributions $\Pi \in U(\mu_n, \mu_n)$

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

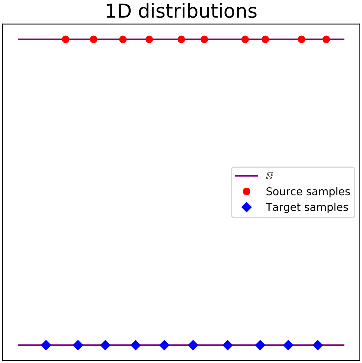
1D OT closed-form



1D closed-form

- Sorted 1D data with uniform weights
- Optimal Transport plan is the identity scaled by a uniform weight

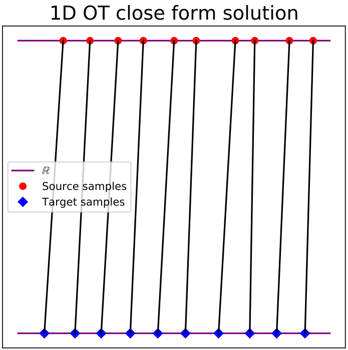
1D OT closed-form

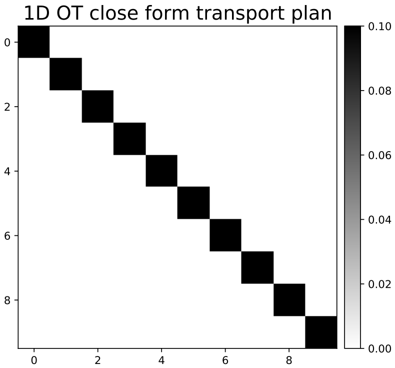


1D closed-form

- Sorted 1D data with uniform weights
- Optimal Transport plan is the identity scaled by a uniform weight

1D OT closed-form

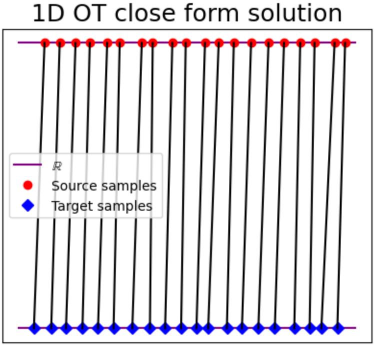


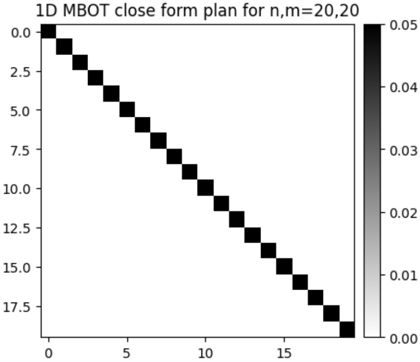


1D closed-form

- Sorted 1D data with uniform weights
- Optimal Transport plan is the identity scaled by a uniform weight

1D Minibatch Optimal Transport closed-form

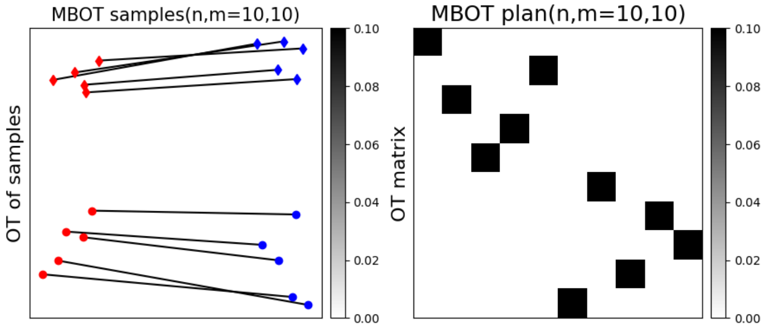




$$\pi_{j,k} = \frac{1}{m}\binom{n}{m}^{-2}\sum_{i=i_{\min}}^{i_{\max}}\binom{j-1}{i-1}\binom{k-1}{i-1}\binom{n-j}{m-i}\binom{n-k}{m-i}$$

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

2D Minibatch Optimal Transport



MBOT on a toy example

- 2 balanced classes in the source and target domains
- Classes are in different clusters

Estimator properties

Proposition

We have the following properties:

- $\overline{h}^m$, $\widetilde{h}_k^m$ *are unbiased estimators of* E_h
- *Strictly positive loss:* $\overline{h}(\alpha, \alpha) > 0$

Difference with OT

- ⚠ Minibatch OT is not a metric!
- OT empirical estimator is a biased estimator of OT between continuous measures ($\mathbb{E}_{\alpha_n, \beta_n} W(\alpha_n, \beta_n) \neq W(\alpha, \beta)$) [Bellemare et al., 2017]

From now on, we suppose that α and β compactly supported and the cost c is at least continuous on $\mathcal{X}$ and $\mathcal{Y}$

Deviation bounds

How far is our incomplete estimator $\widetilde{h}_k^m$ to the expectation over minibatches E_h?

Theorem (Maximal deviation bound)

Let $\delta \in (0,1)$, three integers $k \geq 1$ and $m \leq n$ be fixed. Consider two n-tuples $X \sim \alpha^{\otimes n}$ and $Y \sim \beta^{\otimes n}$. With probability at least $1-\delta$ on the draw of X, Y and D_k we have:

$$|\widetilde{h}_k^m(X,Y) - E_h| \leq M \left(\sqrt{\frac{\log(\frac{2}{\delta})}{2\lfloor \frac{n}{m} \rfloor}} + \sqrt{\frac{2\log(\frac{2}{\delta})}{k}} \right), \tag{2}$$

where M is an OT upper bound

Data fitting problem

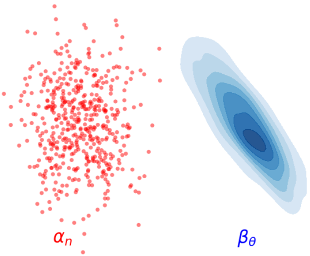
Let discrete samples $(\boldsymbol{x}_i)_{i=1}^n \in \mathbb{R}^d$ and their empirical distribution α_n

Goal: to fit a parametric model $\theta \mapsto \beta_\theta \in \mathcal{M}_1^+(\mathbb{R}^d)$ to α_n using OT

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

It is known as Minimum Wasserstein estimator

Parameters are updated as $\theta_{t+1} = \theta_t + \eta_{\mathrm{lr}} \nabla_\theta \, \mathrm{OT}_c(\alpha_n, \beta_\theta)$



Data fitting problem

Let discrete samples $(\boldsymbol{x}_i)_{i=1}^n \in \mathbb{R}^d$ and their empirical distribution α_n

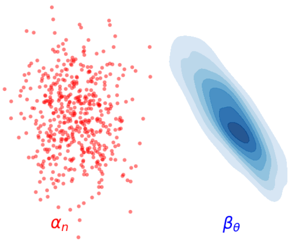
Goal: to fit a parametric model $\theta \mapsto \beta_\theta \in \mathcal{M}_1^+(\mathbb{R}^d)$ to α_n using OT

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \quad \widetilde{h}_k^m(\alpha_n, \beta_\theta)$$

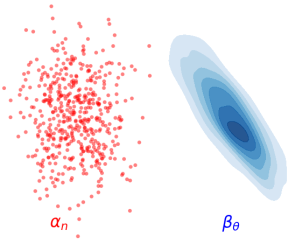
Replace Wasserstein distance by minibatch OT to use SGD

Parameters are updated as $\theta_{t+1} = \theta_t + \eta_{\mathrm{lr}} \nabla_\theta \widetilde{h}_k^m(\alpha_n, \beta_\theta)$

Minimum Wasserstein estimator



Minimum MB Wasserstein estimator



Data fitting problem

Let discrete samples $(\boldsymbol{x}_i)_{i=1}^n \in \mathbb{R}^d$ and their empirical distribution α_n

Goal: to fit a parametric model $\theta \mapsto \beta_\theta \in \mathcal{M}_1^+(\mathbb{R}^d)$ to α_n using OT

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \quad \widetilde{h}_k^m(\alpha_n, \beta_\theta)$$

We replace the Wasserstein distance by minibatch OT to use SGD

Parameters are updated as $\theta_{t+1} = \theta_t + \eta_{\mathrm{lr}} \nabla_\theta \widetilde{h}_k^m(\alpha_n, \beta_\theta)$

Does minimizing this expectation with SGD converge towards the right minimum ?

- $\overline{h}^m, \widetilde{h}_k^m$ are unbiased estimators of E_h
- Can we exchange gradients and expectations?

Exchange gradient and expectation

Consider Clarke generalized gradients (defined for locally Lipschitz functions [Clarke, 1990])

Theorem

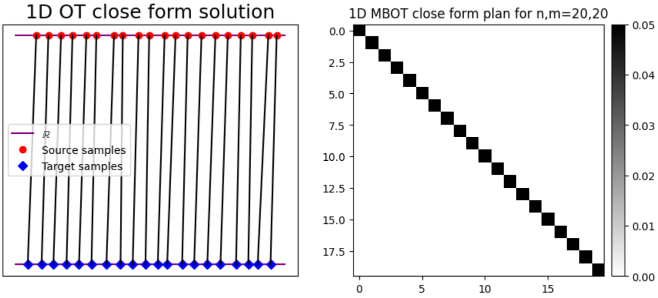
Let $\hat{X}, \{\hat{Y}_\theta\}_{\theta \in \Theta}$ be two m-tuples of random vectors compactly supported and C^m a $\mathbf{C}^1$ cost. Under an additional integrability assumption, we have:

$$\partial_\theta \, \mathbb{E}[h(\mu_m, \mu_m, C^m(\hat{X}, \hat{Y}_\theta))] = \mathbb{E}[\partial_\theta h(\mu_m, \mu_m, C^m(\hat{X}, \hat{Y}_\theta))],$$

with both expectation being finite. Furthermore the function $\theta \mapsto -\mathbb{E}[h(\mu_m, \mu_m, C^m(\hat{X}, \hat{Y}_\theta))]$ is also Clarke regular.

$\rightarrow$ SGD converges almost surely [Davis et al., 2020]

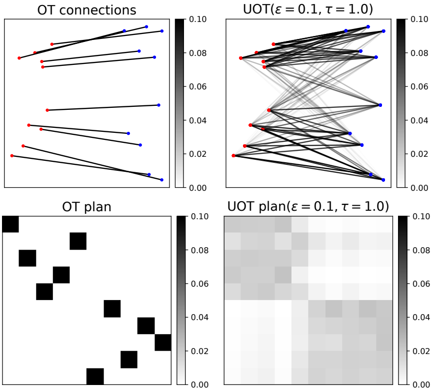
Summary on minibatch optimal transport



- Minibatch optimal transport formalism
- MBOT is a transport problem but not a distance
- Good statistical and optimization properties

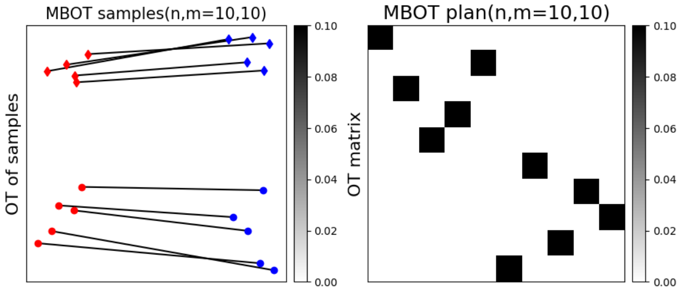
Published in [Fatras et al., 2020b, Fatras et al., 2020a] and submitted in [Fatras et al., 2021c]

Unbalanced minibatch optimal transport



- Limits of MBOT
- Unbalanced minibatch OT
- Domain adaptation experiments

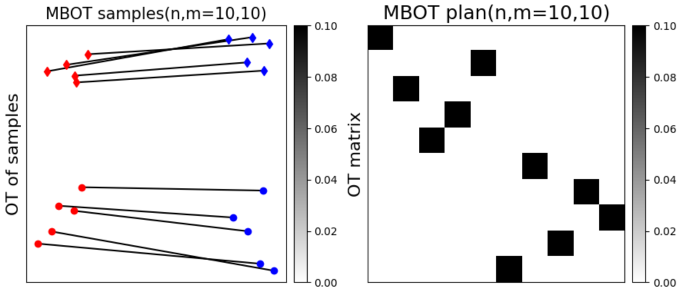
Limits of Minibatch Optimal Transport



Limits of minibatch OT

⚠ Minibatches and marginal constraints create connections between classes !

Limits of Minibatch Optimal Transport



This is due to

- the sampling effect
- the marginal constraints

These force users to use large minibatch size [Damodaran et al., 2018]

Unbalanced Optimal Transport

Definition

Unbalanced optimal transport (UOT) measures the OT cost between probability distributions with relaxed marginals

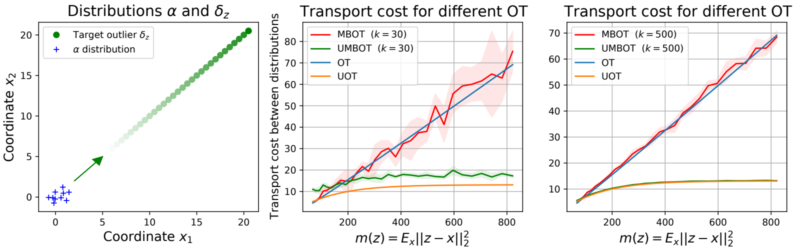
$$\mathrm{UOT}^{\tau,\varepsilon}(\alpha,\beta,c) = \min_{\pi \in \mathcal{M}_+(\mathcal{X}\times\mathcal{Y})} \int c d\pi + \varepsilon \mathtt{KL}(\pi|\alpha \otimes \beta)$$
$$+\tau(\mathtt{KL}(\pi_1\|\alpha) + \mathtt{KL}(\pi_2\|\beta)),$$

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 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

Robust OT

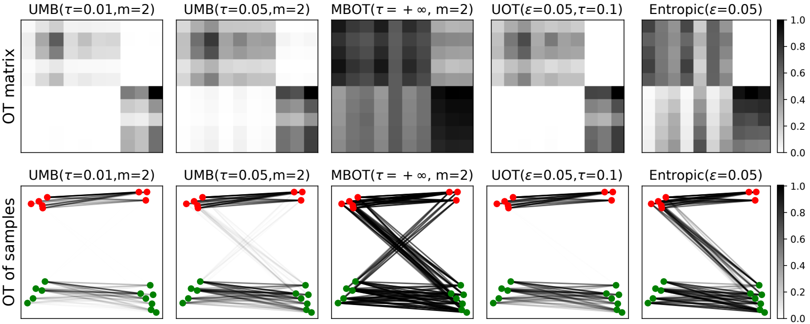


Lemma

Let (α, β) be two probability distributions. For $\zeta \in [0,1]$, write $\tilde{\alpha} = \zeta\alpha + (1-\zeta)\delta_{\boldsymbol{z}}$. Write $m(\boldsymbol{z}) = \int C(\boldsymbol{z}, \boldsymbol{y}) d\beta(\boldsymbol{y})$.

$$\mathrm{UOT}^{\tau,0}(\tilde{\alpha}, \beta, C) \lesssim \zeta\, \mathrm{UOT}^{\tau,0}(\alpha, \beta, C) + 2\tau(1-\zeta)(1 - e^{-m(\boldsymbol{z})/2\tau})$$

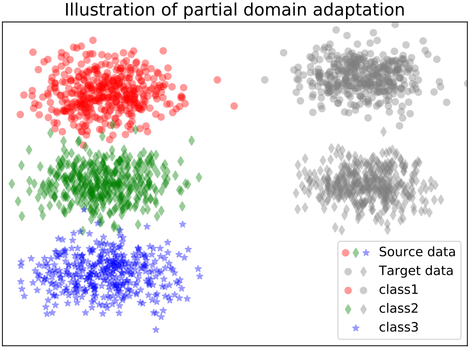
Unbalanced minibatch OT plan



Unbalanced MBOT keeps the same properties as MBOT

- Same loss properties (not a distance but symmetric)
- Same deviation bounds rates
- Same unbiased gradients

Domain adaptation problem



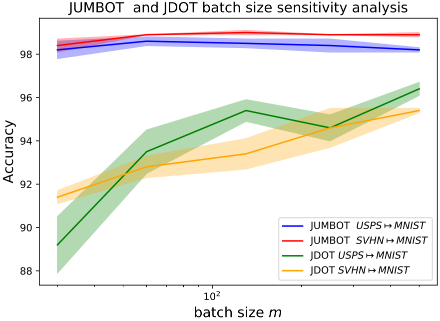
Method

- JUMBOT aligns a joint law between embedded samples and labels like DEEPJDOT
- Use unbalanced minibatch OT instead of minibatch OT
- Can be applied to partial DA unlike DEEPJDOT

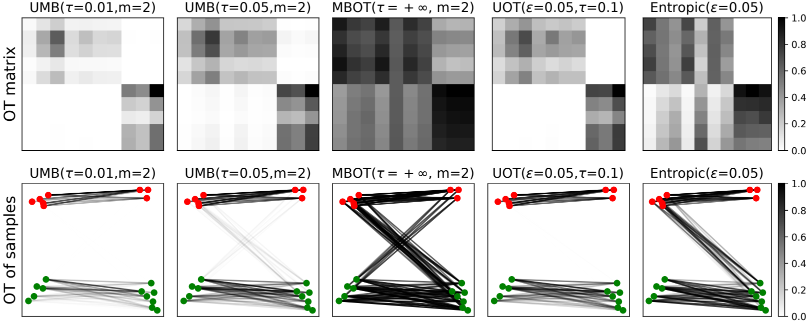
Results, ablation and sensitivity

JUMBOT outperforms state of the art methods on digits (U-M-S), Office Home and VisDA datasets including Partial Office-Home

Methods	U $\to$ M	S $\to$ M
DEEPJDOT	96.4 ± 0.3	95.4 ± 0.1
ENTROPIC DEEPJDOT	97.1 ± 0.3	97.6 ± 0.1
JUMBOT	$\mathbf{98.2 \pm 0.1}$	$\mathbf{98.9 \pm 0.1}$



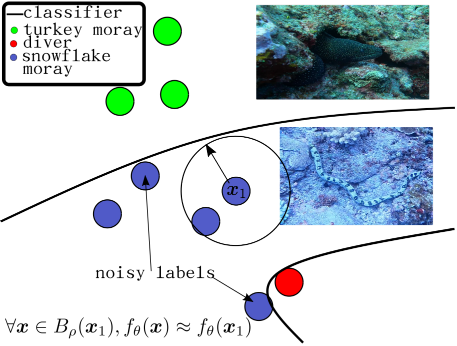
Summary on unbalanced minibatch optimal transport

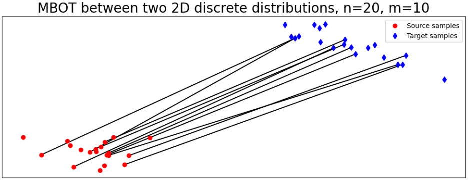


- Minibatch optimal transport creates non-optimal connections
- Replace OT by unbalanced OT
- Same statistical and optimization properties
- JUMBOT outperforms MBOT methods on DA experiments

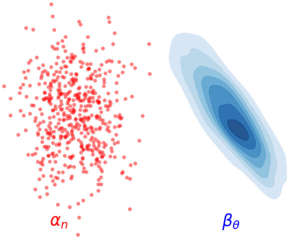
Published in [Fatras et al., 2021b]

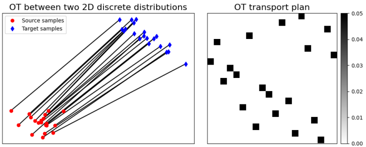
Conclusion





- **Optimal transport in deep learning**
  - as an adversarial regularization for noisy labels
  - to generate misclassified data
- **Minibatch optimal transport**
  - transport problem but not a distance
  - good statistical and optimization properties
  - unbalanced optimal transport to mitigate bad connections





Future work on

- **Optimal transport in deep learning**
  - Ground cost for WAR
  - Optimal transport for out-of-distribution samples
  - Normalizing flow
  - To learn weights to make OT robust to outliers
- **Optimal transport**
  - Sliced unbalanced OT
  - Unbalanced OT when $\tau \to 0$
  - New sampling schemes of MBOT

Publications

- Wasserstein adversarial regularization [Fatras et al., 2021a]
- ARWGAN [Burnel et al., 2020, Burnel et al., 2021]
- Minibatch optimal transport formalism [Fatras et al., 2020b, Fatras et al., 2020a, Fatras et al., 2021c]
- Unbalanced minibatch optimal transport [Fatras et al., 2021b]
- Stochastic optimization [Pedregosa et al., 2019]
- Open source OT library [Flamary et al., 2021]

Thank you for your attention !

References i

[Arjovsky et al., 2017] Arjovsky, M., Chintala, S., and Bottou, L. (2017).
Wasserstein generative adversarial networks.
In *Proceedings of the 34th International Conference on Machine Learning.*

[Bellemare et al., 2017] Bellemare, M. G., Danihelka, I., Dabney, W., Mohamed, S., Lakshminarayanan, B., Hoyer, S., and Munos, R. (2017).
The cramer distance as a solution to biased wasserstein gradients.
CoRR, abs/1705.10743.

[Burnel et al., 2020] Burnel, J.-C., Fatras, K., and Courty, N. (2020).
Generating natural adversarial hyperspectral examples with a modified wasserstein gan.
In *C&ESAR 2020.*

[Burnel et al., 2021] Burnel, J.-C., Fatras, K., Flamary, R., and Courty, N. (2021).
Generating natural adversarial remote sensing images.
IEEE Transactions on Geoscience and Remote Sensing.

[Clarke, 1990] Clarke, F. H. (1990).
Optimization and nonsmooth analysis.
SIAM.

[Clémençon et al., 2016] Clémençon, S., Colin, I., and Bellet, A. (2016).
Scaling-up empirical risk minimization: Optimization of incomplete u-statistics.
Journal of Machine Learning Research.

[Courty et al., 2017] Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2017).
Optimal transport for domain adaptation.
IEEE Transactions on Pattern Analysis and Machine Intelligence.

[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.

[Davis et al., 2020] Davis, D., Drusvyatskiy, D., Kakade, S., and Lee, J. D. (2020).
Stochastic subgradient method converges on tame functions.
Foundations of computational mathematics, 20(1):119–154.

[Fatras et al., 2021a] Fatras, K., Bushan, B., Lobry, S., Flamary, R., Tuia, D., and Courty, N. (2021a).
Wasserstein adversarial regularization for learning with label noise.
IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1–1.

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

[Fatras et al., 2020a] Fatras, K., Zine, Y., Flamary, R., Gribonval, R., and Courty, N. (2020a).
Divergence wasserstein par lots.
In *52 èmes Journées de Statistiques de la Société Française de Statistique*.

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

[Fatras et al., 2021c] Fatras, K., Zine, Y., Majewski, S., Flamary, R., Gribonval, R., and Courty, N. (2021c).
Minibatch optimal transport distances; analysis and applications.
CoRR.

[Ferradans et al., 2013] Ferradans, S., Papadakis, N., Rabin, J., Peyré, G., and Aujol, J.-F. (2013).
Regularized discrete optimal transport.
In *Scale Space and Variational Methods in Computer Vision*. Springer Berlin Heidelberg.

[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.

[Frogner et al., 2015] Frogner, C., Zhang, C., Mobahi, H., Araya, M., and Poggio, T. A. (2015).
Learning with a wasserstein loss.
In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 2053–2061. Curran Associates, Inc.

[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.

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

[Majewski et al., 2018] Majewski, S., Miasojedow, B., and Moulines, E. (2018).
Analysis of nonsmooth stochastic approximation: the differential inclusion approach.
arXiv preprint arXiv:1805.01916.

[Miyato et al., 2018] Miyato, T., Maeda, S., Ishii, S., and Koyama, M. (2018).
Virtual Adversarial Training: A regularization method for supervised and semi-supervised learning.
IEEE Transactions on Pattern Analysis and Machine Intelligence.

[Pedregosa et al., 2019] Pedregosa, F., Fatras, K., and Casotto, M. (2019).
Proximal splitting meets variance reduction.
In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1–10. PMLR.

[Peng et al., 2017] Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., and Saenko, K. (2017).
Visda: The visual domain adaptation challenge.

[Peyré and Cuturi, 2019] Peyré, G. and Cuturi, M. (2019).
Computational optimal transport.
Foundations and Trends® in Machine Learning, 11(5-6):355–607.

References vii

[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.

[Zhang et al., 2017] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017).
Understanding deep learning requires rethinking generalization.
In *The International Conference on Learning Representations*.

Appendix

Hyperparameters i

Datasets	η_1	η_2	η_3	τ	ε
Digits	0.1	0.1	1	1	0.1
VisDA	0.005	1	1	0.3	0.01
Office-Home	0.01	0.5	1	0.5	0.01
Partial Office-Home	0.003	0.75	10	0.06	0.01

Clarke regularity

Definition

A function f is said to be Clarke regular at $\boldsymbol{x}$ provided:

- For all $\boldsymbol{v}$, the usual one-sided directional derivative $f'(\boldsymbol{x}, \boldsymbol{v})$ exists
- For all $\boldsymbol{v}$, $f'(\boldsymbol{x}; \boldsymbol{v}) = f^{\circ}(\boldsymbol{x}; \boldsymbol{v})$

Full echange gradients and expectations theorem

Theorem

Let $\boldsymbol{u}$ be uniform probability vectors and let $\boldsymbol{X}$ be a $\mathbb{R}^{dm}$-valued random variable, and $\{\boldsymbol{Y}_\theta\}$ a family of $\mathbb{R}^{dm}$-valued random variables defined on the same probability space, indexed by $\theta \in \Theta$, where $\Theta \subset \mathbb{R}^q$ is open. Assume that $\theta \mapsto \boldsymbol{Y}_\theta$ is $\mathbf{C}^1$. Consider a $\mathbf{C}^1$ cost C, $h \in \{\mathrm{UOT}^{\tau,\varepsilon}\}$, and assume in addition that the random variables $\boldsymbol{X}, \{Y_\theta\}_{\theta\in\Theta}$ are compactly supported. If for all $\theta \in \Theta$ there exists an open neighbourhood U, $\theta \in U \subset \Theta$, and a random variable $K_U : \Omega \to \mathbb{R}$ with finite expected value, such that

$$\|C(\boldsymbol{X}(\omega), \boldsymbol{Y}_{\theta_1}(\omega)) - C(\boldsymbol{X}(\omega), \boldsymbol{Y}_{\theta_2}(\omega))\| \leq K_U(\omega)\|\theta_1 - \theta_2\| \qquad (3)$$

then we have

$$\partial_\theta \, \mathbb{E}\left[h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right] = \mathbb{E}\left[\partial_\theta h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right]. \qquad (4)$$

with both expectation being finite. Furthermore the function $\theta \mapsto -\mathbb{E}\left[h(\boldsymbol{u}, \boldsymbol{u}, C(\boldsymbol{X}, \boldsymbol{Y}_\theta))\right]$ is also Clarke regular.