Domain Adaptation: old and new with Optimal Transport

Nicolas Courty (IRISA, University Bretagne Sud), France

Joint work with R. Flamary, A. Rakotomamonjy, D. Tuia, A. Habrard, B. Bushan, I. Redko, K. Fatras and many more...

Monday, 5th of July 2021

Outline

Domain adaptation

Amazon

Traditional supervised learning

- We want to learn predictor such that $y \approx f(\mathbf{x})$.
- Actual $\mathcal{P}(X, Y)$ unknown.
- We have access to training dataset $(\mathbf{x}_i, y_i)_{i=1,\dots,n}$ $(\widehat{\mathcal{P}}(X, Y))$.
- We choose a loss function $\mathcal{L}(y, f(\mathbf{x}))$ that measure the discrepancy.

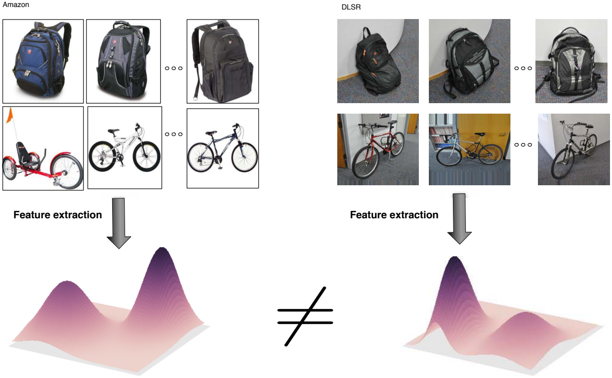
Empirical risk minimization

We week for a predictor f minimizing

$$\min_f \left\{ \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\widehat{\mathcal{P}}} \mathcal{L}(y, f(\mathbf{x})) = \sum_j \mathcal{L}(y_j, f(\mathbf{x}_j)) \right\} \tag{1}$$

- Well known generalization results for predicting on new data.
- Loss is usually $\mathcal{L}(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$ for least square regression or $\mathcal{L}(y, f(\mathbf{x})) = \max(0, 1 - yf(\mathbf{x}))^2$ for squared Hinge loss SVM.
- Cross-entropy for neural networks (among others)

Domain Adaptation problem

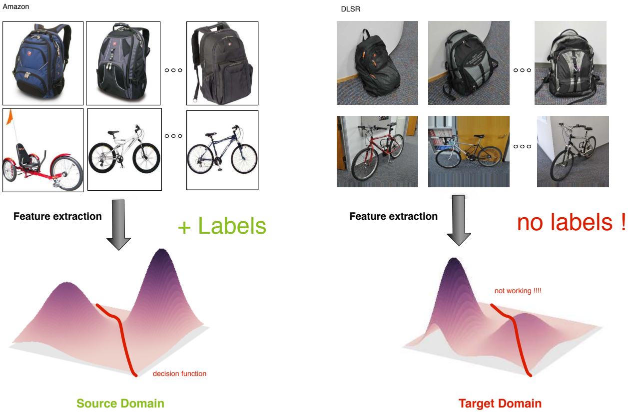


Probability Distribution Functions over the domains

Our context

- Classification problem with data coming from different sources (domains).
- Distributions are different but related.

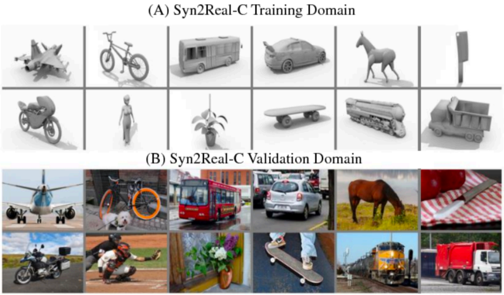
Unsupervised domain adaptation problem



Problems

- Labels only available in the **source domain**, and classification is conducted in the **target domain**.
- Classifier trained on the source domain data performs badly in the target domain

Is Domain Adaptation a real problem ?



- Ubiquitous problem in Deep Learning ! People can not afford to label billions of data for every single problems
- Novel interesting challenges if one considers learning from synthetic data

What about Remote Sensing ?

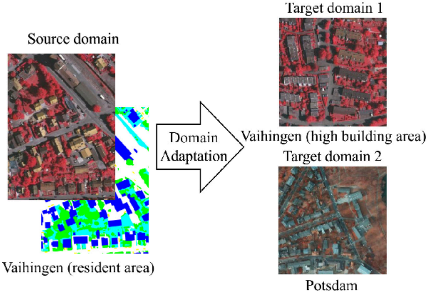


Image from [Benjdira et al., 2019]

Remote Sensing context

The sources of shift between a labelled source and the target are numerous

- different atmospheric conditions, time of acquisition
- different geographic zones, different spectral responses/shapes for objects of the same class
- different captors with varying spatial/spectral resolution, or even nature of the data (LiDAR, RADAR, etc.)

Short state-of-the-art

Problem: how to learn a classifier that can be good on several domains with only labels in one of the domain ?

- Theory [Urner et al., 2011, Ben-David et al., 2012] measures the difficulty of this task in terms of discrepancy of the representations of the data.
- Possible solutions include:
  - find domain invariant representation of the data (subspace projection, feature learning)
  - transform data from one domain into 'similar' versions in the other domain (adversarial methods)
  - Most of the time a notion of divergence between the distributions is involved:
    - Second order statistical moments
    - Maximum Mean Discrepancy (MMD)
    - Optimal Transport !

The tree that hides the forest

Several variants of this problem can be considered:

- **Unsupervised** vs **semi-supervised Domain Adaptation**: depending on the available knowledge from the source
- **Heterogeneous Domain Adaptation**: data do not lie in the same space
- **Multi** vs **Single source Domain Adaptation**: when the number of available source domains is more than one.
- **Covariate** vs **Target shift**: are the class-conditional distributions $\mathcal{P}(X|\text{label})$ different, or is it the class proportions $\mathcal{P}(\text{label})$?
- **Domain Generalization**: several source domains are available, but the target is not; One wants to achieve the best generalization performance.
- **Source free Domain Adaptation**: only a classifier on the source domain is available (not the samples)
- many more (Few|one|zero shot domain adaptation|generalization, federated DA, etc.)

Change in the class space

When the label space is changing:

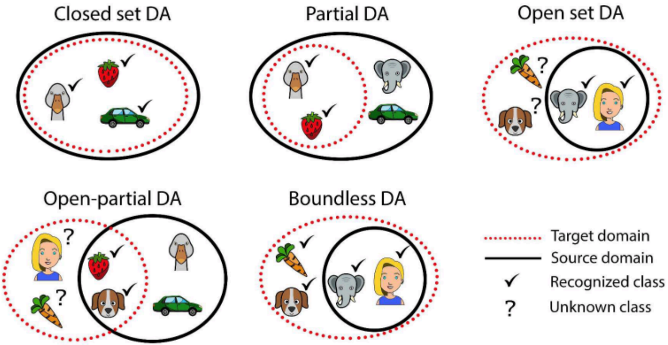
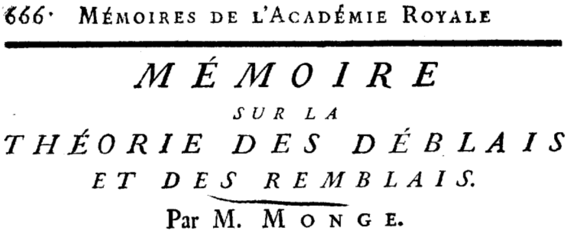


Image adapted from [You et al., 2019]

Outline



666. MÉMOIRES DE L'ACADÉMIE ROYALE

MÉMOIRE

SUR LA

THÉORIE DES DÉBLAIS

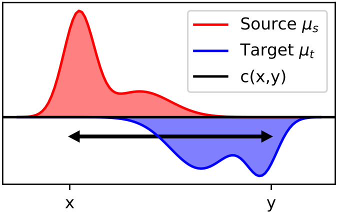
ET DES REMBLAIS.

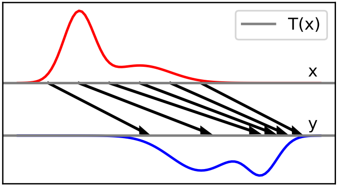
Par M. MONGE.

Problem [Monge, 1781]

- How to move dirt from one place (déblais) to another (remblais) while minimizing the effort ?
- Find a mapping T between the two distributions of mass (transport).
- Optimize with respect to a displacement cost $c(x, y)$ (optimal).

The origins of optimal transport

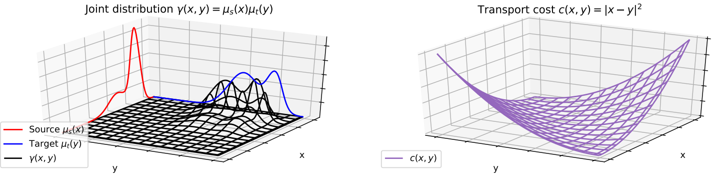




Problem [Monge, 1781]

- How to move dirt from one place (déblais) to another (remblais) while minimizing the effort ?
- Find a mapping T between the two distributions of mass (transport).
- Optimize with respect to a displacement cost $c(x, y)$ (optimal).

Optimal transport (Kantorovich formulation)



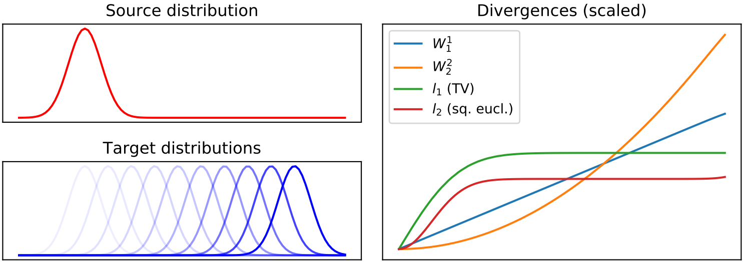
- The Kantorovich formulation [Kantorovich, 1942] seeks for a probabilistic $\pi \in \mathcal{P}(\Omega_s \times \Omega_t)$ between Ω_s and Ω_t:

$$\pi_0 = \underset{\pi}{\operatorname{argmin}} \int_{\Omega_s \times \Omega_t} c(\mathbf{x},\mathbf{y})\pi(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y}, \tag{2}$$

$$\text{s.t.} \quad \pi \in \Pi = \left\{ \pi \geq 0, \int_{\Omega_t} \pi(\mathbf{x},\mathbf{y})d\mathbf{y} = \mu_s, \int_{\Omega_s} \pi(\mathbf{x},\mathbf{y})d\mathbf{x} = \mu_t \right\}$$

- π is a joint probability measure with marginals μ_s and μ_t.
- Linear Program that always have a solution.

Wasserstein distance



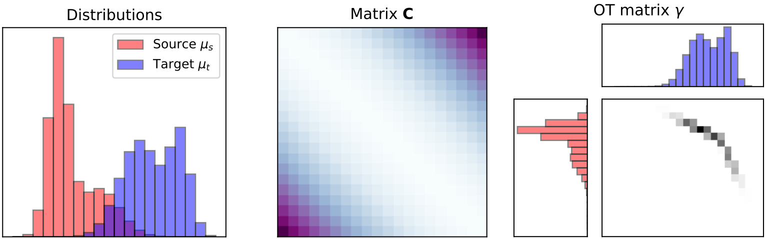
Wasserstein distance

$$W_p^p(\mu_s, \mu_t) = \min_{\pi \in \Pi} \int_{\Omega_s \times \Omega_t} c(\mathbf{x}, \mathbf{y}) \pi(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y}) \sim \pi} [c(\mathbf{x}, \mathbf{y})] \tag{3}$$

where $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^p$

- Do not need the distribution to have overlapping support.
- Subgradients can be computed with the dual variables of the LP.
- Works for continuous and discrete distributions (histograms, empirical).

Discrete Optimal transport



OT Linear Program

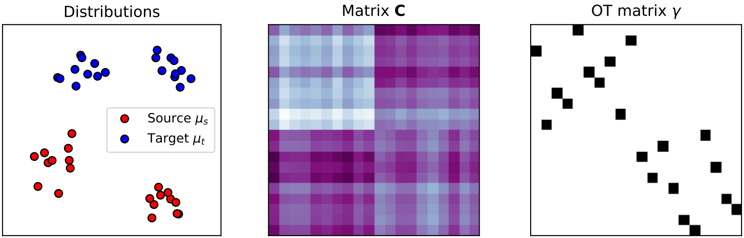
$$\pi_0 = \underset{\pi \in \Pi}{\text{argmin}} \quad \left\{ \langle \pi, \mathbf{C} \rangle_F = \sum_{i,j} \gamma_{i,j} c_{i,j} \right\}$$

where **C** is a cost matrix with $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ and the marginals constraints are

$$\Pi = \left\{ \pi \in (\mathbb{R}^+)^{n_s \times n_t} | \, \pi \mathbf{1}_{n_t} = \mu_s, \pi^T \mathbf{1}_{n_s} = \mu_t \right\}$$

Solved with Network Flow solver of complexity $O(n^3 \log(n))$.

Discrete Optimal transport



OT Linear Program

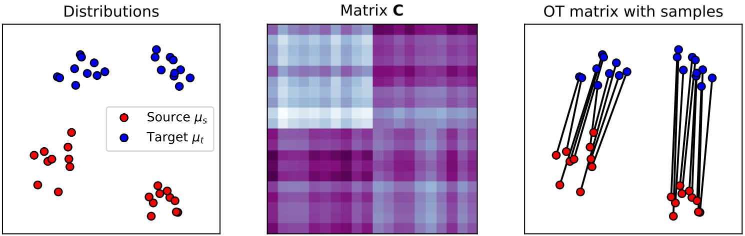
$$\boldsymbol{\pi}_0 = \underset{\boldsymbol{\pi}\in\Pi}{\text{argmin}} \quad \left\{ \langle \boldsymbol{\pi}, \mathbf{C} \rangle_F = \sum_{i,j} \gamma_{i,j} c_{i,j} \right\}$$

where **C** is a cost matrix with $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ and the marginals constraints are

$$\Pi = \left\{ \boldsymbol{\pi} \in (\mathbb{R}^+)^{n_s \times n_t} | \, \boldsymbol{\pi}\mathbf{1}_{n_t} = \mu_s, \boldsymbol{\pi}^T\mathbf{1}_{n_s} = \mu_t \right\}$$

Solved with Network Flow solver of complexity $O(n^3 \log(n))$.

Discrete Optimal transport



OT Linear Program

$$\boldsymbol{\pi}_0 = \underset{\boldsymbol{\pi} \in \Pi}{\text{argmin}} \quad \left\{ \langle \boldsymbol{\pi}, \mathbf{C} \rangle_F = \sum_{i,j} \gamma_{i,j} c_{i,j} \right\}$$

where **C** is a cost matrix with $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ and the marginals constraints are

$$\Pi = \left\{ \boldsymbol{\pi} \in (\mathbb{R}^+)^{n_s \times n_t} | \, \boldsymbol{\pi} \mathbf{1}_{n_t} = \mu_s, \boldsymbol{\pi}^T \mathbf{1}_{n_s} = \mu_t \right\}$$

Solved with Network Flow solver of complexity $O(n^3 \log(n))$.

Regularized optimal transport

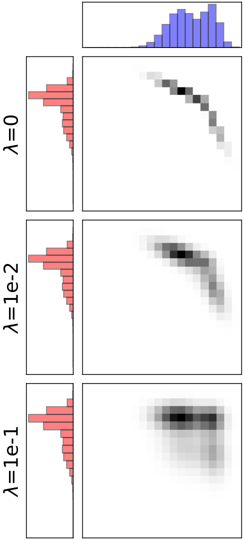
$$\gamma_0^\lambda = \underset{\pi \in \Pi}{\operatorname{argmin}} \quad \langle \pi, \mathbf{C} \rangle_F + \lambda \Omega(\pi), \qquad (4)$$

Regularization term $\Omega(\pi)$

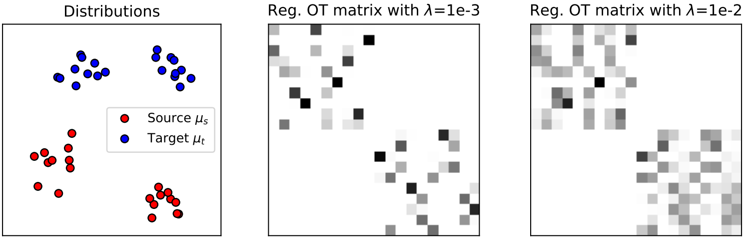
- Entropic regularization [Cuturi, 2013].
- Group Lasso [Courty et al., 2016a].
- KL, Itakura Saito, β-divergences, L2, etc.

Why regularize?

- Smooth the “distance” estimation:
  $W_\lambda(\mu_s, \mu_t) = \left\langle \gamma_0^\lambda, \mathbf{C} \right\rangle_F$
- Encode prior knowledge on the data.
- Better posed problem (convex, stability).
- Fast algorithms to solve the OT problem.



Entropic regularized optimal transport

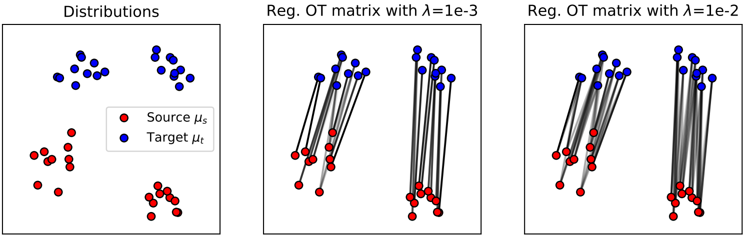


Entropic regularization [Cuturi, 2013]

$$\Omega(\pi) = \sum_{i,j} \pi(i,j) \log \pi(i,j)$$

- Regularization with the negative entropy of π.
- Solution of the form $\pi_0^\lambda = \text{diag}(\mathbf{u}) \exp(-\mathbf{C}/\lambda) \text{diag}(\mathbf{v})$.
- **Sinkhorn-Knopp** algorithm (implementation in parallel, GPU).

Entropic regularized optimal transport

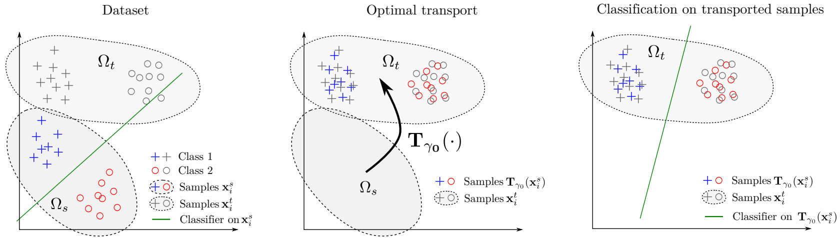


Entropic regularization [Cuturi, 2013]

$$\Omega(\pi) = \sum_{i,j} \pi(i,j) \log \pi(i,j)$$

- Regularization with the negative entropy of π.
- Solution of the form $\pi_0^\lambda = \text{diag}(\mathbf{u}) \exp(-\mathbf{C}/\lambda) \text{diag}(\mathbf{v})$.
- **Sinkhorn-Knopp** algorithm (implementation in parallel, GPU).

Optimal transport for domain adaptation



Assumptions

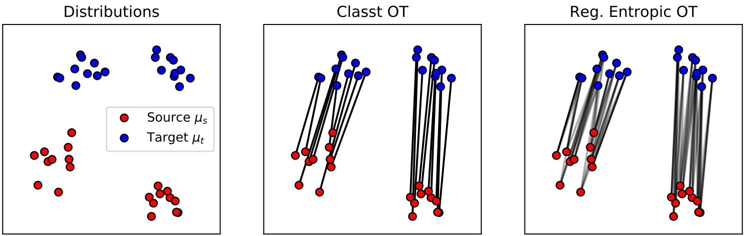
- There exist a transport in the feature space $\mathbf{T}$ between the two domains.
- The transport preserves the conditional distributions:

$$P_s(y|\mathbf{x}_s) = P_t(y|\mathbf{T}(\mathbf{x}_s)).$$

3-step strategy [Courty et al., 2016b, PAMI]

1. Estimate optimal transport between distributions.
2. Transport the training samples with barycentric mapping .
3. Learn a classifier on the transported training samples.

Transporting the discrete samples

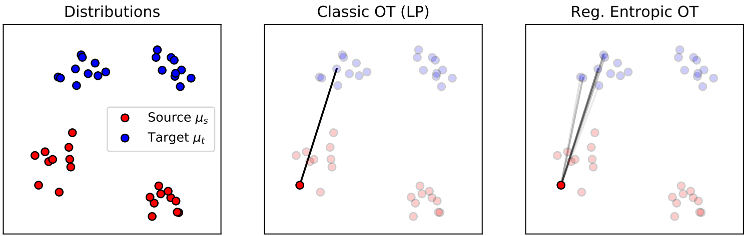


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\pi_0}(\mathbf{x}_i^s) = \operatorname*{argmin}_{\mathbf{x}} \sum_j \boldsymbol{\pi}_0(i,j) c(\mathbf{x}, \mathbf{x}_j^t). \tag{5}$$

- The mass of each source sample is spread onto the target samples (line of $\boldsymbol{\pi}_0$).
- The mapping is the barycenter of the target samples weighted by $\boldsymbol{\pi}_0$
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).

Transporting the discrete samples

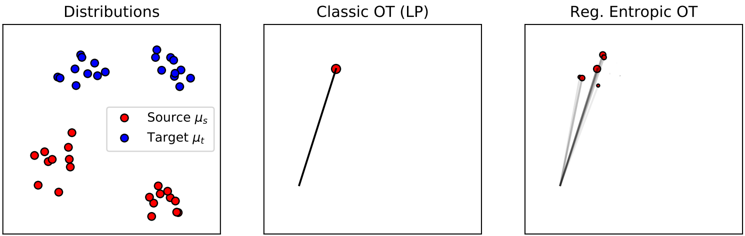


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\pi_0}(\mathbf{x}_i^s) = \underset{\mathbf{x}}{\text{argmin}} \sum_j \pi_0(i,j) c(\mathbf{x}, \mathbf{x}_j^t). \tag{5}$$

- The mass of each source sample is spread onto the target samples (line of π_0).
- The mapping is the barycenter of the target samples weighted by π_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).

Transporting the discrete samples

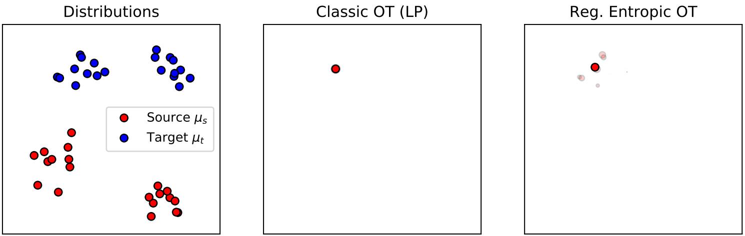


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\pi_0}(\mathbf{x}_i^s) = \underset{\mathbf{x}}{\operatorname{argmin}} \quad \sum_j \pi_0(i,j) c(\mathbf{x}, \mathbf{x}_j^t). \tag{5}$$

- The mass of each source sample is spread onto the target samples (line of π_0).
- The mapping is the barycenter of the target samples weighted by π_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).

Transporting the discrete samples

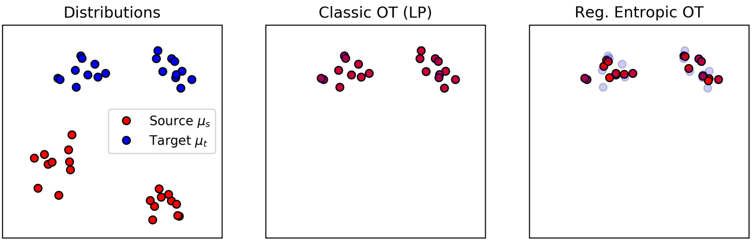


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\pi_0}(\mathbf{x}_i^s) = \underset{\mathbf{x}}{\operatorname{argmin}} \quad \sum_j \pi_0(i,j) c(\mathbf{x}, \mathbf{x}_j^t). \tag{5}$$

- The mass of each source sample is spread onto the target samples (line of π_0).
- The mapping is the barycenter of the target samples weighted by π_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).

Transporting the discrete samples

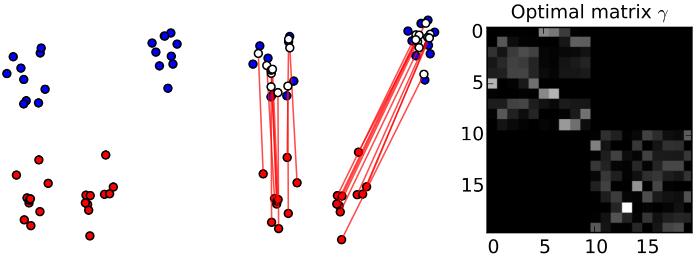


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\pi_0}(\mathbf{x}_i^s) = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_j \pi_0(i,j)c(\mathbf{x}, \mathbf{x}_j^t). \tag{5}$$

- The mass of each source sample is spread onto the target samples (line of π_0).
- The mapping is the barycenter of the target samples weighted by π_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).

Class-based regularization



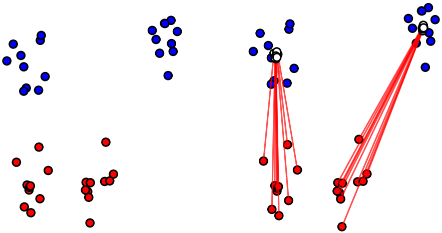
Group lasso regularization

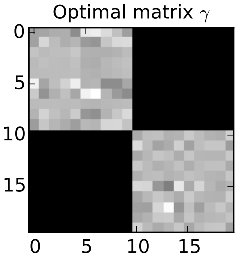
- We group components of π using classes from the source domain:

$$\Omega_c(\pi) = \sum_j \sum_c ||\pi(\mathcal{I}_c, j)||_q^p, \tag{6}$$

- $\mathcal{I}_c$ contains the indices of the lines related to samples of the class c in the source domain.
- $|| \cdot ||_q^p$ denotes the ℓ_q norm to the power of p.
- For $p \leq 1$, we encourage a target domain sample j to receive masses only from "same class" source samples.

Class-based regularization





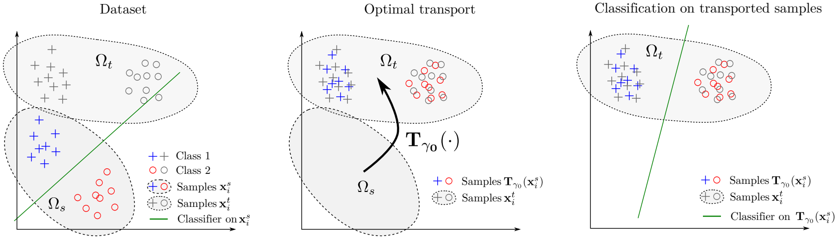
Group lasso regularization

- We group components of π using classes from the source domain:

$$\Omega_c(\pi) = \sum_j \sum_c ||\pi(\mathcal{I}_c, j)||_q^p, \tag{6}$$

- $\mathcal{I}_c$ contains the indices of the lines related to samples of the class c in the source domain.
- $||\cdot||_q^p$ denotes the ℓ_q norm to the power of p.
- For $p \leq 1$, we encourage a target domain sample j to receive masses only from "same class" source samples.

Optimal transport for domain adaptation



Discussion

- Works very well in practice for large class of transformation [Courty et al., 2016b].
- Can use other type of estimated mapping [Perrot et al., 2016, Seguy et al., 2017].

But

- Model transformation only in the feature space.
- Requires the same class proportion between domains [Tuia et al., 2015].
- We estimate a $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ mapping for training a classifier $f : \mathbb{R}^d \rightarrow \mathbb{R}$.

Adapting directly Joint Distributions

Joint distribution and classifier estimation

Joint distribution OT (JDOT, [Courty et al., 2017, NIPS])

- Model the transformation of labels (allow change of proportion/value).
- Learn an optimal target predictor with no labels on target samples.
- Approach theoretically justified (learning bound, cf. paper)

Joint distributions and dataset

- We work with the joint feature/label distributions.
- Let $\mathcal{P}_s(X,Y) \in \mathcal{P}(\Omega \times \mathcal{C})$ and $\mathcal{P}_t(X,Y) \in \mathcal{P}(\Omega \times \mathcal{C})$ the source and target joint distribution.
- We have access to an empirical sampling $\widehat{\mathcal{P}}_s = \frac{1}{N_s}\sum_{i=1}^{N_s} \delta_{\mathbf{x}_i^s,\mathbf{y}_i^s}$ of the source distribution defined by $\mathbf{X}_s = \{\mathbf{x}_i^s\}_{i=1}^{N_s}$ and label information $\mathbf{Y}_s = \{\mathbf{y}_i^s\}_{i=1}^{N_s}$.
- but the target domain is defined only by an empirical distribution in the feature space with samples $\mathbf{X}_t = \{\mathbf{x}_i^t\}_{i=1}^{N_t}$.

Joint distribution OT

Proxy joint distribution

- Let f be a $\Omega \rightarrow \mathcal{C}$ function from a given class of hypothesis $\mathcal{H}$.
- We define the following joint distribution that use f as a proxy of y

$$\mathcal{P}_t^f = (\mathbf{x}, f(\mathbf{x}))_{\mathbf{x}\sim\mu_t} \tag{7}$$

and its empirical counterpart $\widehat{\mathcal{P}}_t^{\,f} = \frac{1}{N_t}\sum_{i=1}^{N_t} \delta_{\mathbf{x}_i^t, f(\mathbf{x}_i^t)}$.

Learning with JDOT

We propose to learn the predictor f that minimize :

$$\min_f \quad \left\{ W_1(\widehat{\mathcal{P}}_s, \widehat{\mathcal{P}}_t^{\,f}) = \inf_{\boldsymbol{\pi}\in\Delta} \sum_{ij} \mathcal{D}(\mathbf{x}_i^s, \mathbf{y}_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t))\boldsymbol{\pi}_{ij} \right\} \tag{8}$$

- Δ is the transport polytope.
- distance in joint space: $\mathcal{D}(\mathbf{x}_i^s, \mathbf{y}_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t)) = \alpha\|\mathbf{x}_i^s - \mathbf{x}_j^t\|^2 + \mathcal{L}(\mathbf{y}_i^s, f(\mathbf{x}_j^t))$ with $\alpha > 0$.
- We search for the predictor f that better align the joint distributions.

Optimization problem

$$\min_{f\in\mathcal{H},\boldsymbol{\pi}\in\Delta} \quad \sum_{i,j} \boldsymbol{\pi}_{i,j}\left(\alpha d(\mathbf{x}_i^s,\mathbf{x}_j^t)+\mathcal{L}(y_i^s,f(\mathbf{x}_j^t))\right)+\lambda\Omega(f) \tag{9}$$

Optimization procedure

- $\Omega(f)$ is a regularization for the predictor f
- We propose to use block coordinate descent (BCD)/Gauss Seidel.
- Provably converges to a stationary point of the problem.

π update for a fixed f

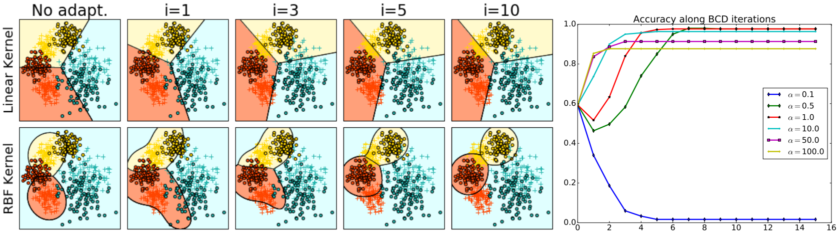
- Classical OT problem.
- Solved by network simplex.
- Regularized OT can be used (add a term to problem (9))

f update for a fixed π

$$\min_{f\in\mathcal{H}} \quad \sum_{i,j} \boldsymbol{\pi}_{i,j}\mathcal{L}(y_i^s,f(\mathbf{x}_j^t))+\lambda\Omega(f) \tag{10}$$

- Weighted loss from all source labels.
- π performs label propagation.

Classification with JDOT



Least square regression with quadratic regularization

For a fixed π the optimization problem is equivalent to

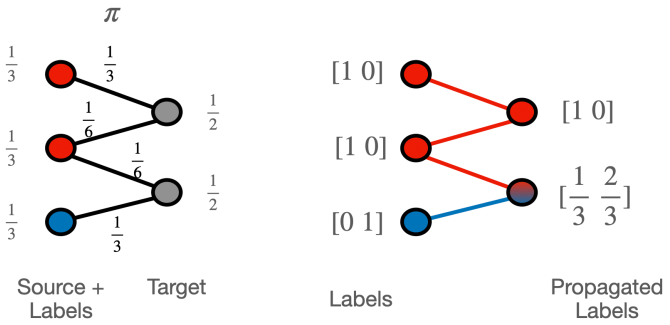
$$\min_{f \in \mathcal{H}} \quad \sum_j \frac{1}{n_t} \|\widehat{y}_j - f(\mathbf{x}_j^t)\|^2 + \lambda \|f\|^2 \qquad (11)$$

- $\widehat{y}_j = n_t \sum_j \pi_{i,j} y_i^s$ is a weighted average of the source target values (a.k.a label propagation).
- Note that this problem is linear instead of quadratic.
- Can use any solver (linear, kernel ridge, neural network).

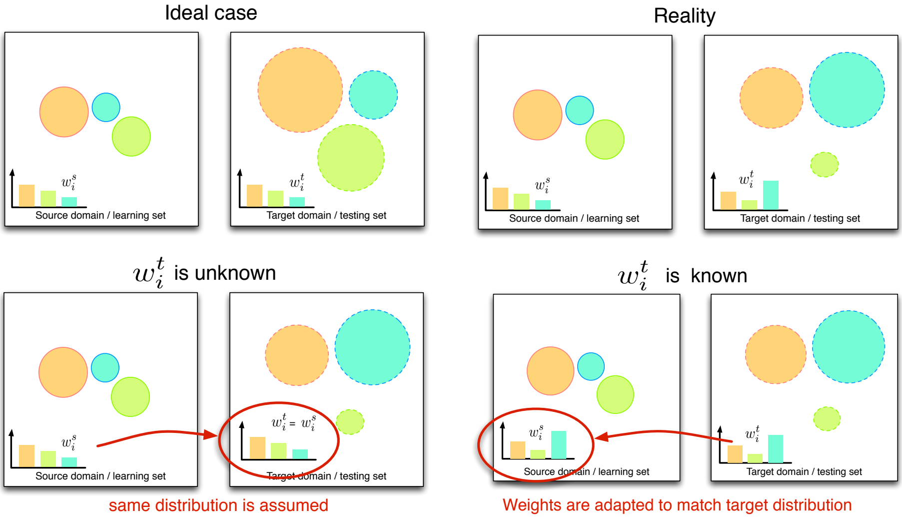
What is Label propagation ?

The operation $\widehat{y}_j = n_t \sum_j \boldsymbol{\pi}_{i,j} y_i^s$ can be understood intuitively as a way to estimate (one-hot encoded) labels onto the samples of the target domain

- thanks to the coupling matrix $\boldsymbol{\pi}$



Impact of the changes in class distributions ?



Solving for Target Shift

Target Shift in DA

Target Shift in DA

- Class conditional distributions are the same between domains
- Only proportions of samples from each class is changing
- The Generalized Target Shift problem is harder and consider that both are changing (not covered in this presentation)

JCPOT [Redko et al., 2019, AISTATS 2019]

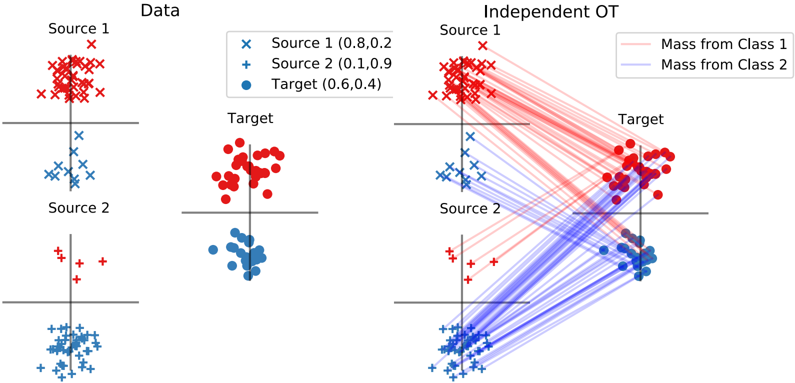
The idea is to simultaneously estimate the proportions of classes Δ in the target domain.

- Possible in the multi-source domain adaptation context (several source domains are available)
- we cast the problem as a Wasserstein barycenter problem:

$$\min_{\Delta} \sum_{k=1}^{K} W_p^p(\mu_s^{(k)}, \mu_t^{\Delta})$$

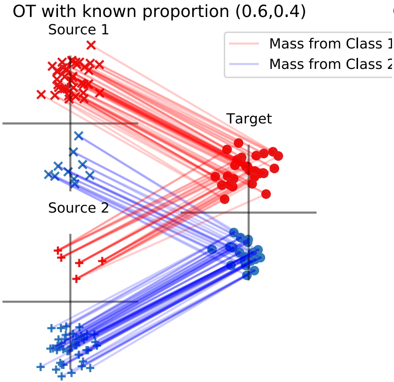
where K is the number of available source domains.

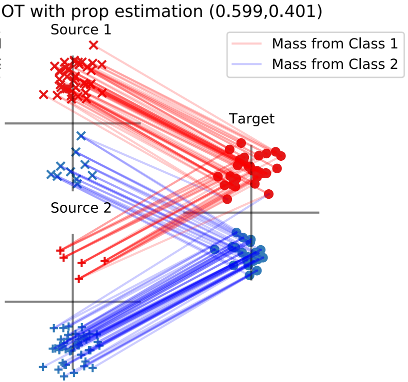
Illustrations of JCPOT (1/2)



Covariate shift DA mixes instances from different classes!

Illustrations of JCPOT (2/2)



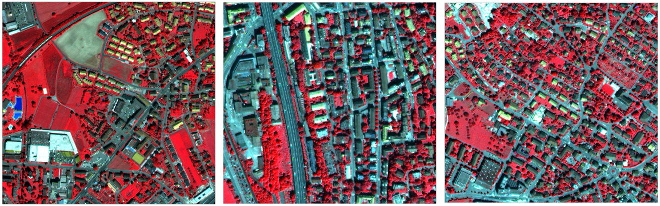


Our method handles target shift efficiently!

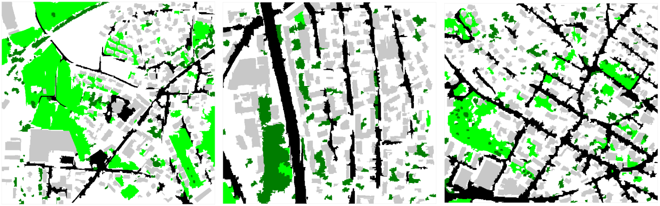
Remote Sensing data

- Zurich Summer' data set composed of 20 **satellite images**
- **4 classes**: Roads, Buildings, Trees and Grass
- **17 source** and **1 target** domain
- Average **class proportions** $[0.25 \pm 0.07,\ 0.4 \pm 0.13,\ 0.22 \pm 0.11,\ 0.13 \pm 0.11]$

Input satellite images



Satellite images with 4 classes

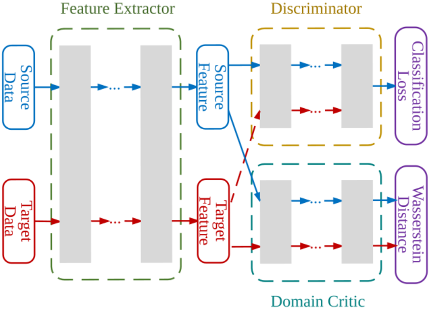


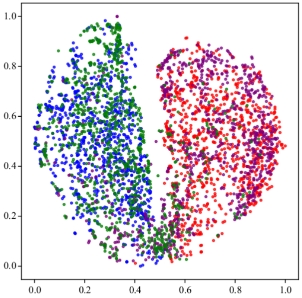
Classification results

# of source domains	Average class proportions	# of source instances	No adaptation	`OTDA` `PT`	`OTDA` `LP`	`MDA` `Causal`	`JCPOT` `LP`	Target only
2	[0.17 0.4 0.16 0.27]	2’936	0.61	0.52	0.57	0.65	**0.66**	0.65
5	[0.22 0.39 0.18 0.21]	6’716	0.62	0.55	0.6	0.66	**0.68**	0.64
8	[0.25 0.46 0.17 0.12]	16’448	0.63	0.54	0.59	0.67	**0.71**	0.65
11	[0.26 0.48 0.16 0.1]	21’223	0.63	0.54	0.58	0.67	**0.72**	0.673
14	[0.26 0.45 0.19 0.1]	27’875	0.63	0.52	0.58	0.67	**0.72**	0.65
17	[0.25 0.42 0.20 0.13]	32’660	0.63	0.5	0.59	0.67	**0.73**	0.61

Going deep !

Domain adaptation with Wasserstein distance





(d) t-SNE of WDGRL features

Domain adaptation for deep learning [Shen et al., 2018]

- Modern DA aim at aligning source and target in the deep representation : DANN [Ganin et al., 2016], MMD [Tzeng et al., 2014], CORAL [Sun and Saenko, 2016].
- Wasserstein distance used as objective for the adaptation [Shen et al., 2018].

Large scale JDOT [Damodaran et al., 2018, ECCV]

Large scale JDOT

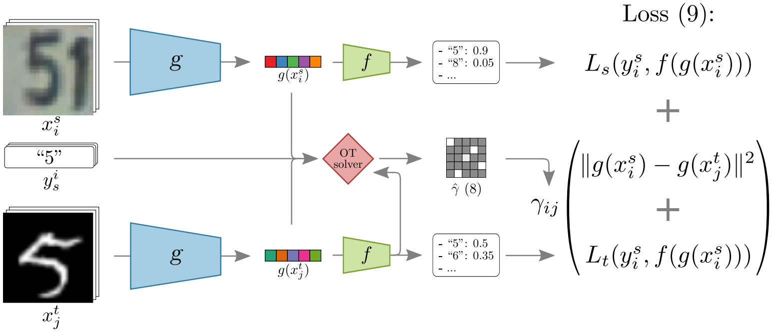
- How to scale JDOT to tackle large datasets/ deep learning architectures ?
- Use minibatches instead for computing the transport in the primal [Genevay et al., 2017]
- Learn simultaneously the best embedding !
- Evaluate batch-local couplings on (sufficiently large) couples of random (without replacement) batches in source and target domain
- update f from these couplings

Algorithm : Deep JDOT

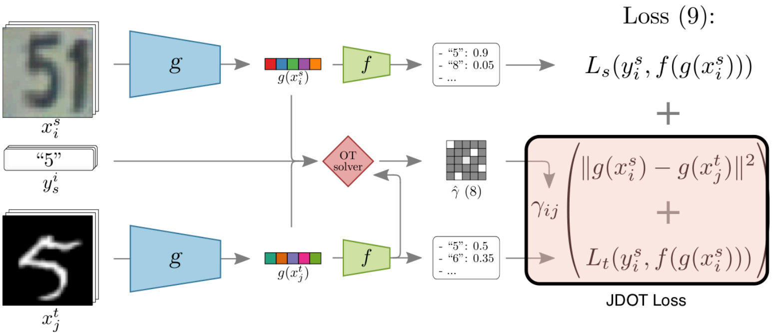
Inputs: Source data X^s, y^s, Target data X^t

for BCD Iterations **do**
  for each Source/Target minibatch **do**
    Solve OT with JDOT loss
    Perform label propagation on minibatch
  end for
  Update model f on one epoch
end for

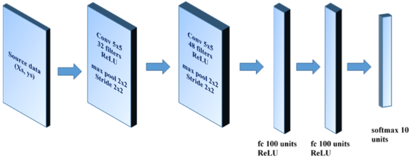
DeepJDOT in a glance



DeepJDOT in a glance



Large scale datasets



Description	MNIST→ USPS	USPS→MNIST	SVHN→MNIST	MNIST→ MNIST-M
Source samples	60000	9298	73257	60000
Target samples	9298	60000	60000	60000
height/width	16×16	16×16	32×32×3	28×28×3

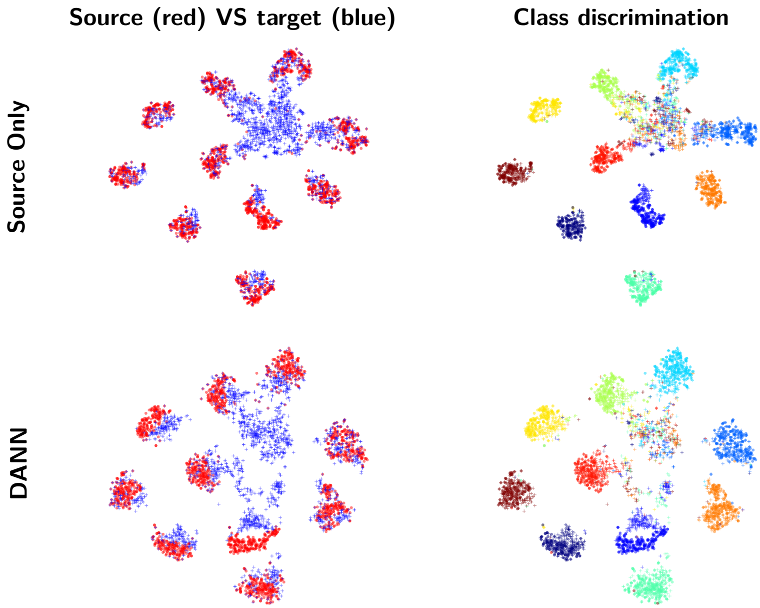
- Four cross domain digits datasets: MNIST, USPS, SVHN, MNIST-M .
- We consider a deep convolutional architecture.
- Dropout is used on the dense layers when training.

Experimental Results for large scale JDOT

Method	Adaptation:source→target			
	MNIST → USPS	USPS → MNIST	SVHN → MNIST	MNIST → MNIST-M
Source only	94.8	59.6	60.7	60.8
DeepCORAL [6]	89.33	91.5	59.6	66.5
MMD [14]	88.5	73.5	64.8	72.5
DANN [8]	*95.7*	90.0	70.8	75.4
ADDA [21]	92.4	93.8	76.0[5]	78.8
AssocDA [16]	-	-	*95.7*	*89.5*
Self-ensemble[4] [42]	88.14	92.35	93.33	-
DRCN [40]	91.8	73.6	81.9	-
DSN [41]	91.3	-	82.7	83.2
CoGAN [9]	91.2	89.1	-	-
UNIT [18]	**95.9**	*93.5*	90.5	-
GenToAdapt [19]	95.3	90.8	92.4	-
I2I Adapt [20]	92.1	87.2	80.3	-
StochJDOT	93.6	90.5	67.6	66.7
DeepJDOT (ours)	*95.7*	**96.4**	**96.7**	**92.4**
target only	95.8	98.7	98.7	96.8

- StochJDOT = ablation study, no learning of the embedding (cost is Euclidean distance in original feature space)

Emebddings


Source (red) VS target (blue)
Class discrimination
Source Only
DANN

Embeddings

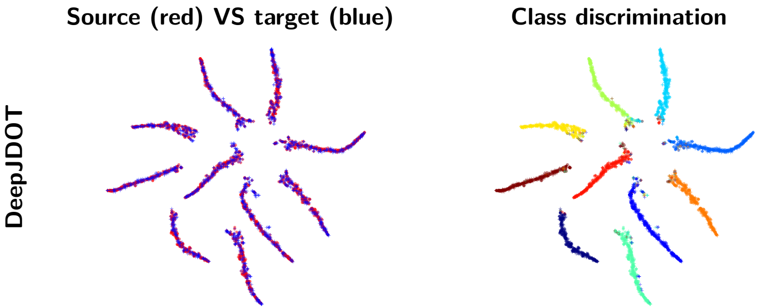
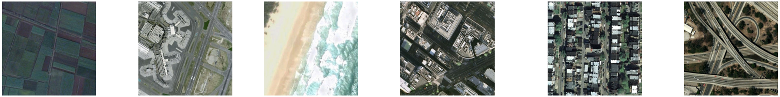


Figure: t-SNE embeddings of 2‘000 test samples for MNIST (source) and MNIST-M (target) for `Source only` classifier, `DANN` and `DeepJDOT`. The left column shows domain comparisons, where colors represent the domain. The right column shows the ability of the methods to discriminate classes (samples are colored w.r.t. their classes).

Remote Sensing data

Image classification problem between UC Merced (top) and WHU-RS19 (bottom) datasets (backbone network is ResNet-50).

agricultural airplane beach buildings med. resid. overpass



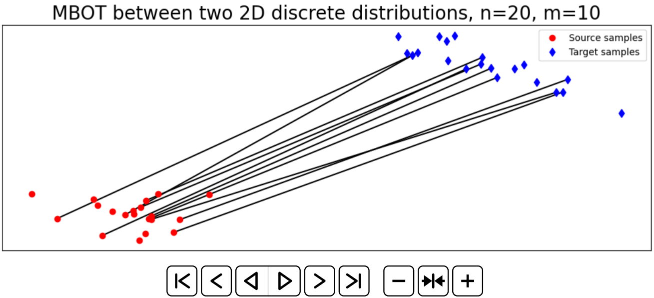
farmland airport beach commercial residential viaduct

Table: Overall accuracies for the discussed datasets and domain adaptation methods.

Method	Adaptation: source → target	
	UC Merced → WHU-RS19	WHU-RS19 → UC Merced
Source only	0.66	0.59
MMD	0.68	0.68
DeepCORAL	0.68	0.67
DeepJDOT	**0.75**	**0.73**
Target only	1.00	1.00
Source & target	1.00	1.00

Wait ! You said Mini-batch ????

Minibatch Optimal Transport



Idea : Compute OT between the minibatches from domains

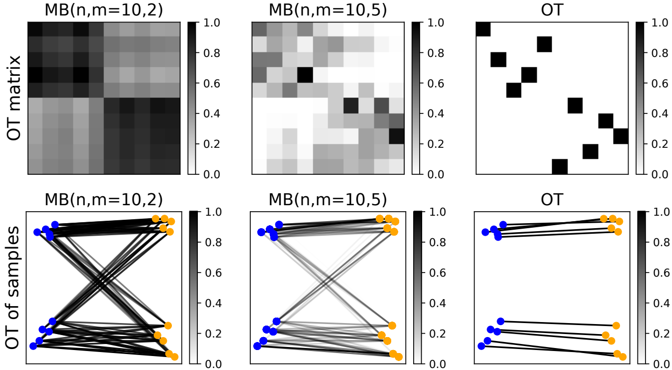
Minibatch Optimal Transport

$$\mathrm{MBOT}_p^p(\mu_s, \mu_t) := \mathbb{E}_{(X,Y)\sim\mu_s^{\otimes m}\otimes\mu_t^{\otimes m}}[W_p^p(b_s, b_t)]$$

where $b_s = \frac{1}{m}\sum_i \delta_{X_i}$ and $b_t = \frac{1}{m}\sum_i \delta_{Y_i}$.

- Interesting asymptotic properties [Fatras et al., 2020]
- Exchange sum and gradients ! [Fatras et al., 2021b]
- Can be defined for other OT variants applied on the batches

MBOT behavior



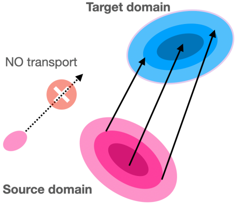
Properties of MBOT

- Preserve marginal constraints
- Globally, acts as a regularization of the π matrix

We need a way to mitigate effects of sampling !

Unbalanced Optimal Transport

Solution: change the original OT by Unbalanced Optimal Transport.



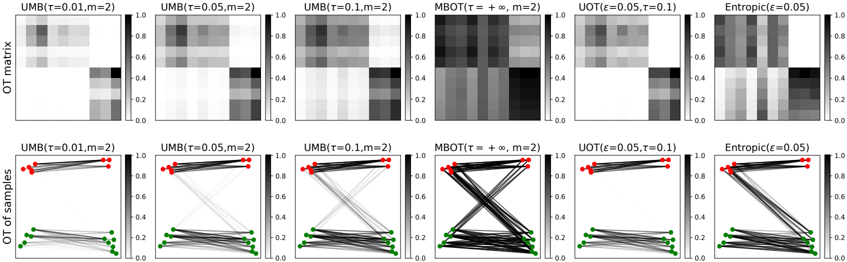
Definition

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

$$\mathrm{UOT}^{\tau,\varepsilon}(\alpha,\beta) = \min_{\boldsymbol{\pi}\in\mathcal{M}_+(\mathcal{X}\times\mathcal{Y})} \int c d\boldsymbol{\pi} + \varepsilon \mathtt{KL}(\pi|\alpha\otimes\beta) + \tau(\mathtt{KL}(\boldsymbol{\pi}_1\|\alpha) + \mathtt{KL}(\boldsymbol{\pi}_2\|\beta)),$$

where $\boldsymbol{\pi}$ is the transport plan, $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ the plan's marginals, $\tau \geq 0$ is the marginal penalization and $\varepsilon \geq 0$ is the regularization coefficient.

Minibatch Unbalanced OT



- Reduce the errors associated to bad matchings due to sample effects
- Target shift impact is reduced
- Allows to do partial domain adaptation !

JUMBOT [Fatras et al., 2021a, ICML 2021]

Simply replace the computation of OT in deep JDOT by Unbalanced OT

Office Home dataset

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

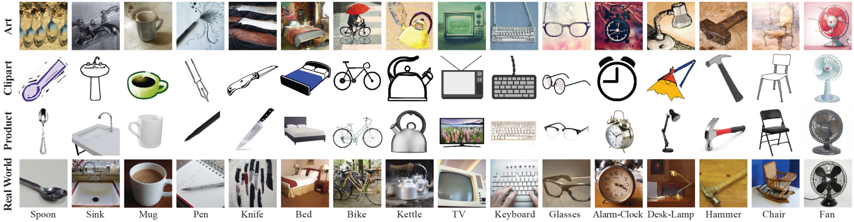


Figure taken from [Venkateswara et al., 2017]. 65 classes in the source and target domains for balanced DA and 25 classes in the target domains for partial DA.

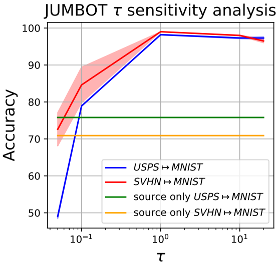
Office Home experiments

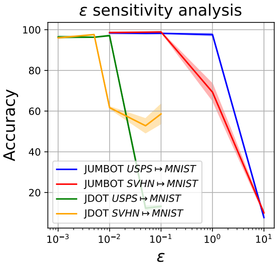
	Method	A-C	A-P	A-R	C-A	C-P	C-R	P-A	P-C	P-R	R-A	R-C	R-P	avg
DA	RESNET-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
	DANN (*)	44.3	59.8	69.8	48.0	58.3	63.0	49.7	42.7	70.6	64.0	51.7	78.3	58.3
	CDAN-E(*)	52.5	71.4	76.1	59.7	69.9	71.5	58.7	50.3	77.5	70.5	57.9	**83.5**	66.6
	DEEPJDOT (*)	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**

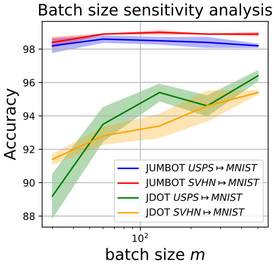
No experiments yet on remote sensing data !

Qualitative analysis: Ablation and sensitivity

Methods	U → M	S → M
DEEPJDOT	96.4 ± 0.3	95.4 ± 0.1
ENTROPIC DEEPJDOT	97.1 ± 0.3	97.6 ± 0.1
JUMBOT	**98.2** ± **0.1**	**98.9** ± **0.1**

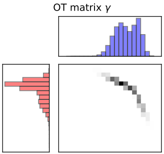


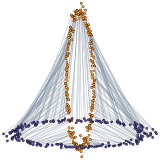


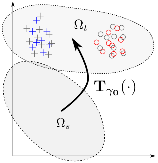


Outline

Optimal transport for deep Domain Adaptation







Learning with optimal transport

- A natural and powerful divergence for domain adaptation.
- Tunable cost functions encode complex relations in feature space.
- Recent optimization procedures opened it to medium/large scale datasets.
- Sensible loss between non overlapping distributions (not the case for MMD).

On-going works

- Domain adaptation on different tasks (detection, semantic segmentation, etc.) and on heterogeneous data (see our recent COOT NeurIPS paper)
- Domain adaptation for time series with OT
- Toward a unified approach for domain generalization with OT

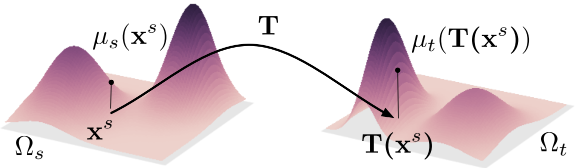
Thank you

Python code available on GitHub: `https://github.com/PythonOT/POT`

- OT LP solver, Sinkhorn (stabilized, $\varepsilon-$scaling, GPU)
- Domain adaptation with OT.
- Barycenters, Wasserstein unmixing.
- Wasserstein Discriminant Analysis.
- Gromov-Wasserstein and variants for graphs
- New ! Different backend support (Numpy, JAX, Pytorch)

Codes for DeepJDOT and JUMBOT also available from dedicated githubs (Tensorflow, PyTorch)

Papers available on my website:
`http://people.irisa.fr/Nicolas.Courty/`



References I

Ben-David, S., Shalev-Shwartz, S., and Urner, R. (2012).
Domain adaptation–can quantity compensate for quality?
In *Proc of ISAIM*.

Benjdira, B., Bazi, Y., Koubaa, A., and Ouni, K. (2019).
Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images.
Remote Sensing, 11(11).

Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. (2017).
Joint distribution optimal transportation for domain adaptation.
In *Neural Information Processing Systems (NIPS)*.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2016a).
Optimal transport for domain adaptation.
IEEE Transactions on Pattern Analysis and Machine Intelligence.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2016b).
Optimal transport for domain adaptation.
Pattern Analysis and Machine Intelligence, IEEE Transactions on.

Cuturi, M. (2013).
Sinkhorn distances: Lightspeed computation of optimal transportation.
In *Neural Information Processing Systems (NIPS)*, pages 2292–2300.

References II

Damodaran, B. B., Kellenberger, B., Flamary, R., Tuia, D., and Courty, N. (2018).
Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation.
CoRR, abs/1803.10081.

Fatras, K., Séjourné, T., Courty, N., and Flamary, R. (2021a).
Unbalanced minibatch optimal transport; applications to domain adaptation.
In *ICML*.

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

Fatras, K., Zine, Y., Majewski, S., Flamary, R., Gribonval, R., and Courty, N. (2021b).
Minibatch optimal transport distances; analysis and applications.

Ferradans, S., Papadakis, N., Peyré, G., and Aujol, J.-F. (2014).
Regularized discrete optimal transport.
SIAM Journal on Imaging Sciences, 7(3).

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016).
Domain-adversarial training of neural networks.
Journal of Machine Learning Research, 17(59):1–35.

References III

Genevay, A., Peyré, G., and Cuturi, M. (2017).
Sinkhorn-autodiff: Tractable wasserstein learning of generative models.
arXiv preprint arXiv:1706.00292.

Kantorovich, L. (1942).
On the translocation of masses.
C.R. (Doklady) Acad. Sci. URSS (N.S.), 37:199–201.

Monge, G. (1781).
Mémoire sur la théorie des déblais et des remblais.
De l'Imprimerie Royale.

Perrot, M., Courty, N., Flamary, R., and Habrard, A. (2016).
Mapping estimation for discrete optimal transport.
In *Neural Information Processing Systems (NIPS)*.

Redko, I., Courty, N., Flamary, R., and Tuia, D. (2019).
Optimal transport for multi-source domain adaptation under target shift.
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 849–858. PMLR.

Seguy, V., Bhushan Damodaran, B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. (2017).
Large-scale optimal transport and mapping estimation.

References IV

Shen, J., Qu, Y., Zhang, W., and Yu, Y. (2018).
Wasserstein distance guided representation learning for domain adaptation.
In *AAAI Conference on Artificial Intelligence*.

Sun, B. and Saenko, K. (2016).
Deep CORAL: Correlation Alignment for Deep Domain Adaptation, pages 443–450.
Springer International Publishing, Cham.

Tuia, D., Flamary, R., Rakotomamonjy, A., and Courty, N. (2015).
Multitemporal classification without new labels: a solution with optimal transport.
In *8th International Workshop on the Analysis of Multitemporal Remote Sensing Images*.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014).
Deep domain confusion: Maximizing for domain invariance.
arXiv preprint arXiv:1412.3474.

Urner, R., Shalev-Shwartz, S., and Ben-David, S. (2011).
Access to unlabeled data can speed up prediction time.
In *Proceedings of ICML*, pages 641–648.

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.

You, K., Long, M., Cao, Z., Wang, J., and Jordan, M. I. (2019).

Universal domain adaptation.

2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 2715–2724.