Domain adaptation with optimal transport

from mapping to learning with joint distribution

R. Flamary - Lagrange, OCA, CNRS, Université Côte d'Azur
Joint work with N. Courty, A. Habrard, A. Rakotomamonjy and B. Bushan Damodoran

Data Science Meetup
January 15, Nice

Table of content

Introduction

Supervised learning

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.

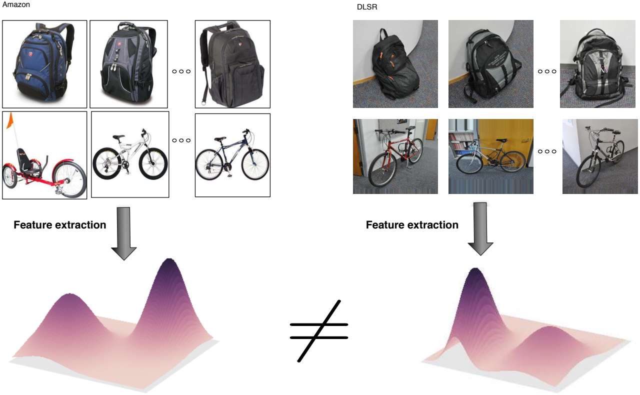
Empirical risk minimization

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 and is $\mathcal{L}(y, f(\mathbf{x})) = \max(0, 1 - yf(\mathbf{x}))^2$ for squared Hinge loss SVM.

Domain Adaptation problem

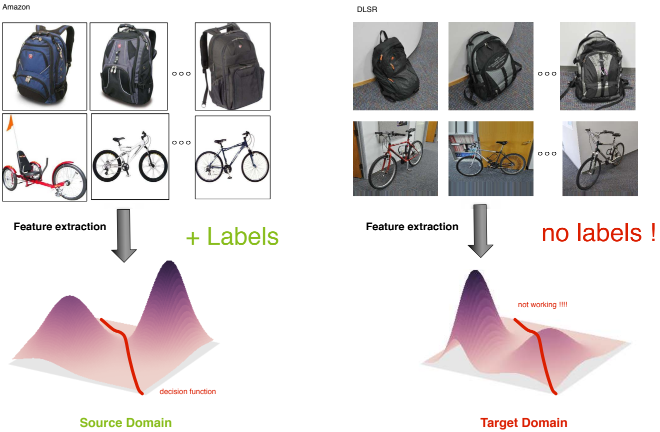


Probability Distribution Functions over the domains

Our context

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

Unsupervised domain adaptation problem

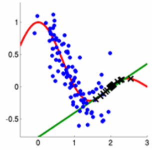


Problems

- 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

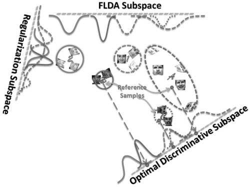
Reweighting schemes [Sugiyama et al., 2008]

- Distribution change between domains.
- Reweigh samples to compensate this change.



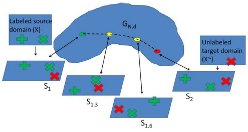
Subspace methods

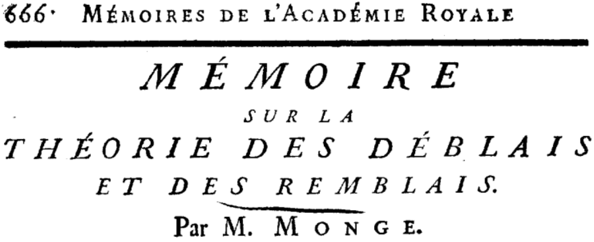
- Data is invariant in a common latent subspace.
- Minimization of a divergence between the projected domains [Si et al., 2010].
- Use additional label information [Long et al., 2014].



Gradual alignment

- Alignment along the geodesic between source and target subspace [R. Gopalan and Chellappa, 2014].
- Geodesic flow kernel [Gong et al., 2012].




666· MÉMOIRES DE L'ACADÉMIE ROYALE

MÉMOIRE

SUR LA

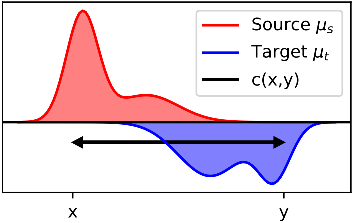
THÉORIE DES DÉBLAIS

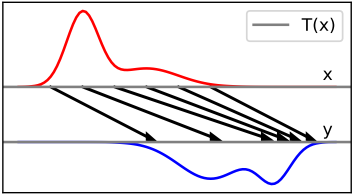
ET DES REMBLAIS.

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).

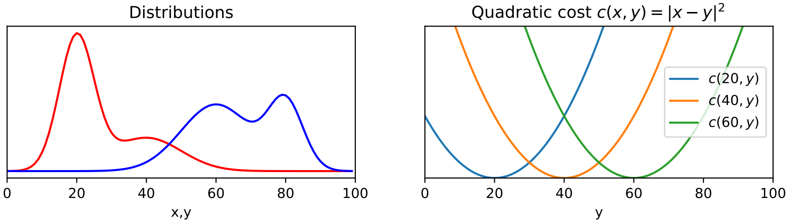




Problem [Monge, 1781]

- 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 (Monge formulation)

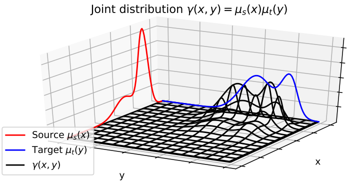


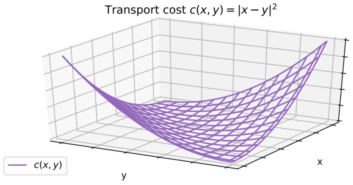
- Probability measures μ_s and μ_t on and a cost function $c : \Omega_s \times \Omega_t \to \mathbb{R}^+$.
- The Monge formulation [Monge, 1781] aim at finding a mapping $T : \Omega_s \to \Omega_t$

$$\inf_{T\#\mu_s=\mu_t} \quad \int_{\Omega_s} c(\mathbf{x}, T(\mathbf{x}))\mu_s(\mathbf{x})d\mathbf{x} \tag{2}$$

- Non-convex optimization problem, mapping does not exist in the general case.
- [Brenier, 1991] proved existence and unicity of the Monge map for $c(x, y) = \|x - y\|^2$ and distributions with densities.

Optimal transport (Kantorovich formulation)





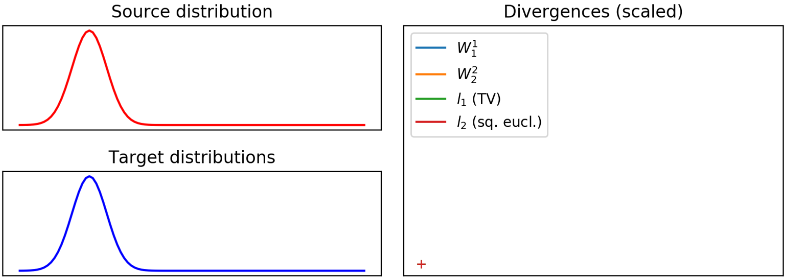
- The Kantorovich formulation [Kantorovich, 1942] seeks for a probabilistic coupling $\gamma \in \mathcal{P}(\Omega_s \times \Omega_t)$ between Ω_s and Ω_t:

$$\boldsymbol{\gamma}_0 = \underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \int_{\Omega_s \times \Omega_t} c(\mathbf{x}, \mathbf{y}) \boldsymbol{\gamma}(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}, \tag{3}$$

$$\text{s.t.} \quad \boldsymbol{\gamma} \in \mathcal{P} = \left\{ \boldsymbol{\gamma} \geq \mathbf{0}, \int_{\boldsymbol{\Omega_t}} \boldsymbol{\gamma}(\mathbf{x}, \mathbf{y}) \mathbf{dy} = \boldsymbol{\mu_s}, \int_{\boldsymbol{\Omega_s}} \boldsymbol{\gamma}(\mathbf{x}, \mathbf{y}) \mathbf{dx} = \boldsymbol{\mu_t} \right\}$$

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

Wasserstein distance



Wasserstein distance

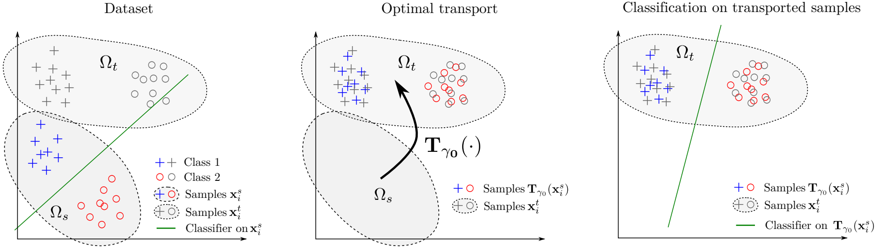
$$W_p^p(\mu_s, \mu_t) = \min_{\gamma \in \mathcal{P}} \int_{\Omega_s \times \Omega_t} c(\mathbf{x}, \mathbf{y}) \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} = E_{(\mathbf{x},\mathbf{y}) \sim \gamma}[c(\mathbf{x}, \mathbf{y})] \tag{4}$$

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

- A.K.A. Earth Mover's Distance (W_1^1) [Rubner et al., 2000].
- 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).

Optimal transport for domain adaptation

Optimal transport for domain adaptation



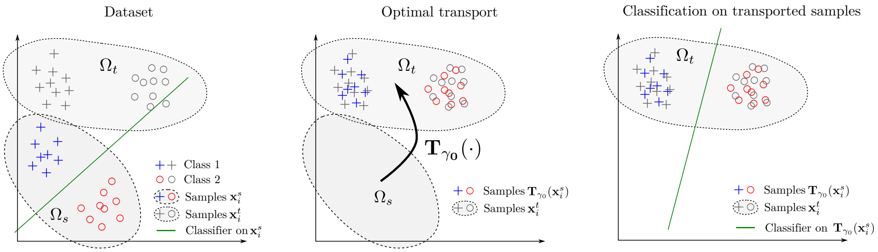
Assumptions

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

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

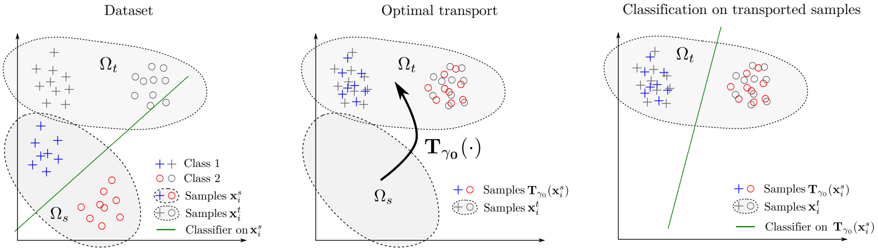
3-step strategy [Courty et al., 2016a]

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



Step 1 : Estimate optimal transport between distributions.

- Choose the ground metric (squared euclidean in our experiments).
- Using regularization allows
  - Large scale and regular OT with entropic regularization [Cuturi, 2013].
  - Class labels in the transport with group lasso [Courty et al., 2016a].
- Efficient optimization based on Bregman projections [Benamou et al., 2015] and
  - Majoration minimization for non-convex group lasso.
  - Generalized Conditionnal gradient for general regularization (cvx. lasso, Laplacian).



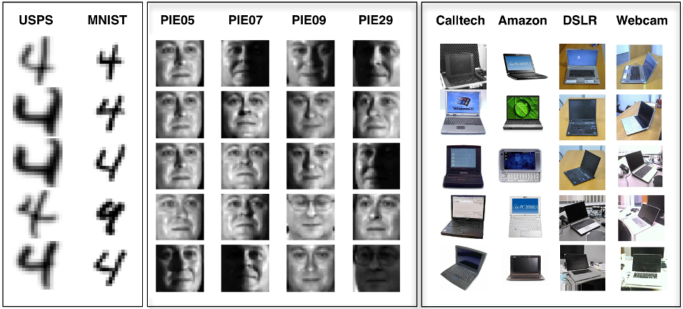
Step 2 : Transport the training samples onto the target distribution.

- The mass of each source sample is spread onto the target samples (line of γ_0).
- Transport using barycentric mapping [Ferradans et al., 2014].
- The mapping can be estimated for out of sample prediction [Perrot et al., 2016, Seguy et al., 2017].

Step 3 : Learn a classifier on the transported training samples

- Transported sample keep their labels.
- Classic ML problem when samples are well transported.

Visual adaptation datasets



Datasets

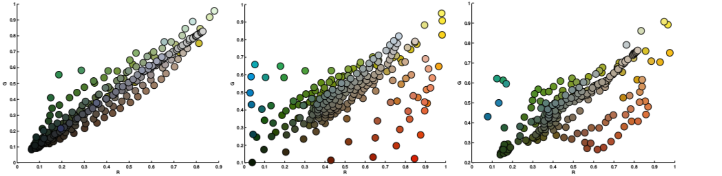
- **Digit recognition**, MNIST VS USPS (10 classes, d=256, 2 dom.).
- **Face recognition**, PIE Dataset (68 classes, d=1024, 4 dom.).
- **Object recognition**, Caltech-Office dataset (10 classes, d=800/4096, 4 dom.).

Numerical experiments

- Comparison with state of the art on the 3 datasets.
- OT works very well on digits and object recognition.
- Works well on deep features adaptation and extension to semi-supervised DA.

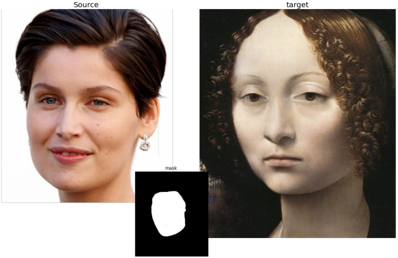
Pixels as empirical distribution [Ferradans et al., 2014]

X^0 Y^0 $\tilde{X}^0$



μ_{X^0} μ_{Y^0} $\mu_{\tilde{X}^0}$

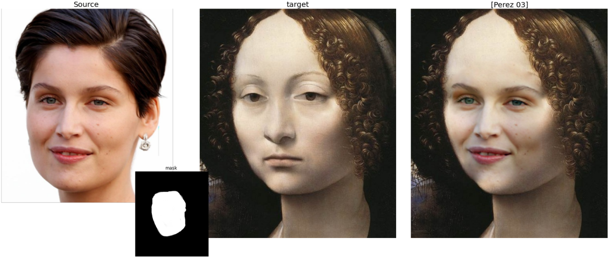
Image colorization [Ferradans et al., 2014]



Poisson image editing [Pérez et al., 2003]

- Use the color gradient from the source image.
- Use color border conditions on the target image.
- Solve Poisson equation to reconstruct the new image.

Seamless copy in images



Poisson image editing [Pérez et al., 2003]

- Use the color gradient from the source image.
- Use color border conditions on the target image.
- Solve Poisson equation to reconstruct the new image.

Seamless copy with gradient adaptation [Perrot et al., 2016]

- Transport the gradient from the source to target color gradient distribution.
- Solve the Poisson equation with the mapped source gradients.
- Better respect of the color dynamic and limits false colors.

Seamless copy in images

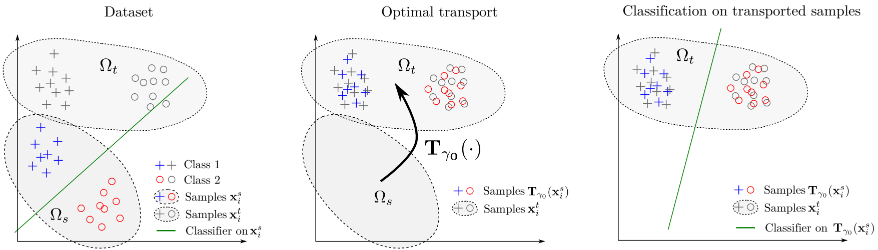
Poisson image editing [Pérez et al., 2003]

- Use the color gradient from the source image.
- Use color border conditions on the target image.
- Solve Poisson equation to reconstruct the new image.

Seamless copy with gradient adaptation [Perrot et al., 2016]

- Transport the gradient from the source to target color gradient distribution.
- Solve the Poisson equation with the mapped source gradients.
- Better respect of the color dynamic and limits false colors.

Seamless copy with gradient adaptation



Discussion

- Works very well in practice for large class of transformation [Courty et al., 2016a].
- Can use 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 $f : \mathbb{R}^d \rightarrow \mathbb{R}$.

Joint distribution OT for domain adaptation (JDOT)

Objectives of JDOT

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

Joint distributions and dataset

- We work with the joint feature/label distributions.
- Let $\Omega \in \mathbb{R}^d$ be a compact input measurable space of dimension d and $\mathcal{C}$ the set of labels.
- 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 $\hat{\mathcal{P}}_s = \frac{1}{N_s} \sum_{i=1}^{N_s} \delta_{\mathbf{x}_i^s, \mathbf{y}_i^s}$ 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 $\mathbf{X}_t = \{\mathbf{x}_i^t\}_{i=1}^{N_t}$.

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

$$\mathcal{P}_t^f = (\mathbf{x}, f(\mathbf{x}))_{\mathbf{x}\sim\mu_t} \tag{5}$$

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

Learning with JDOT

We propose to learn the predictor f that minimize :

$$\min_f \quad \left\{ W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}_t}^f) = \inf_{\boldsymbol{\gamma}\in\Delta} \sum_{ij} \mathcal{D}(\mathbf{x}_i^s, \mathbf{y}_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t))\boldsymbol{\gamma}_{ij} \right\} \tag{6}$$

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

Theorem 1

Let f be any labeling function of $\in \mathcal{H}$. Let $\Pi^* = \operatorname{argmin}_{\Pi \in \Pi(\mathcal{P}_s, \mathcal{P}_t^f)} \int_{(\Omega \times \mathcal{C})^2} \alpha d(\mathbf{x}_s, \mathbf{x}_t) + \mathcal{L}(y_s, y_t) d\Pi(\mathbf{x}_s, y_s; \mathbf{x}_t, y_t)$ and $W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}}_t^f)$ the associated 1-Wasserstein distance. Let $f^* \in \mathcal{H}$ be a Lipschitz labeling function that verifies the ϕ-probabilistic transfer Lipschitzness (PTL) assumption w.r.t. Π^* and that minimizes the joint error $err_S(f^*) + err_T(f^*)$ w.r.t all PTL functions compatible with Π^*. We assume the input instances are bounded s.t. $|f^*(\mathbf{x}_1) - f^*(\mathbf{x}_2)| \leq M$ for all $\mathbf{x}_1, \mathbf{x}_2$. Let $\mathcal{L}$ be any symmetric loss function, k-Lipschitz and satisfying the triangle inequality. Consider a sample of N_s labeled source instances drawn from $\mathcal{P}_s$ and N_t unlabeled instances drawn from μ_t, and then for all $\lambda > 0$, with $\alpha = k\lambda$, we have with probability at least $1 - \delta$ that:

$$\begin{aligned} \mathsf{err}_T(f) \ \leq \ & W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}}_t^f) + \sqrt{\frac{2}{c'} \log(\frac{2}{\delta})} \left(\frac{1}{\sqrt{N_S}} + \frac{1}{\sqrt{N_T}} \right) \\ & + err_S(f^*) + err_T(f^*) + kM\phi(\lambda). \end{aligned}$$

- First term is JDOT objective function.
- Second term is an empirical sampling bound.
- Last terms are usual in DA [Mansour et al., 2009, Ben-David et al., 2010].

Optimization problem

$$\min_{f\in\mathcal{H},\boldsymbol{\gamma}\in\Delta} \quad \sum_{i,j} \boldsymbol{\gamma}_{i,j} \left(\alpha d(\mathbf{x}_i^s, \mathbf{x}_j^t) + \mathcal{L}(y_i^s, f(\mathbf{x}_j^t))\right) + \lambda\Omega(f) \tag{7}$$

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.

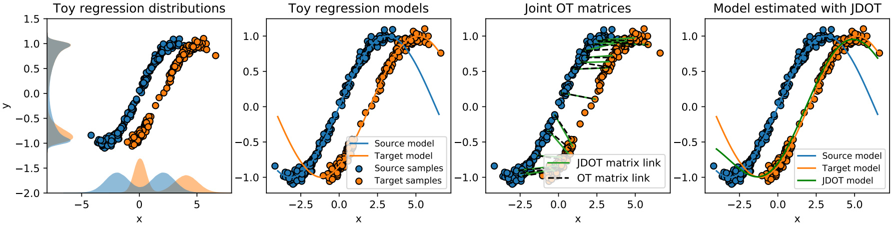
γ update for a fixed f

- Classical OT problem.
- Solved by network simplex.
- Regularized OT can be used (add a term to problem (7))

f update for a fixed γ

$$\min_{f\in\mathcal{H}} \quad \sum_{i,j} \boldsymbol{\gamma}_{i,j}\mathcal{L}(y_i^s, f(\mathbf{x}_j^t)) + \lambda\Omega(f) \tag{8}$$

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

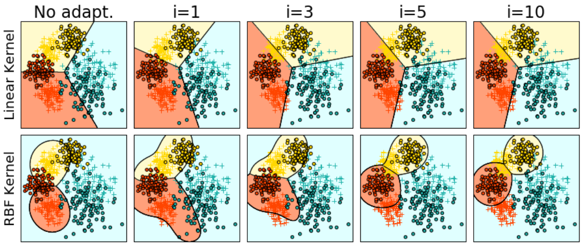


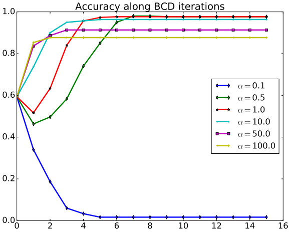
Least square regression with quadratic regularization

For a fixed γ the optimization problem is equivalent to

$$\min_{f \in \mathcal{H}} \quad \sum_j \frac{1}{n_t} \|\hat{y}_j - f(\mathbf{x}_j^t)\|^2 + \lambda \|f\|^2 \tag{9}$$

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





Multiclass classification with Hinge loss

For a fixed γ the optimization problem is equivalent to

$$\min_{f_k \in \mathcal{H}} \quad \sum_{j,k} \hat{P}_{j,k}\mathcal{L}(1, f_k(\mathbf{x}_j^t)) + (1 - \hat{P}_{j,k})\mathcal{L}(-1, f_k(\mathbf{x}_j^t)) + \lambda \sum_k \|f_k\|^2 \qquad (10)$$

- $\hat{\mathbf{P}}$ is the class proportion matrix $\hat{\mathbf{P}} = \frac{1}{N_t}\boldsymbol{\gamma}^\top \mathbf{P}^s$.
- $\mathbf{P}^s$ and $\mathbf{Y}^s$ are defined from the source data with One-vs-All strategy as

$$Y_{i,k}^s = \begin{cases} 1 & \text{if } y_i^s = k \\ -1 & \text{else} \end{cases}, \qquad P_{i,k}^s = \begin{cases} 1 & \text{if } y_i^s = k \\ 0 & \text{else} \end{cases}$$

with $k \in 1, \cdots, K$ and K being the number of classes.

Caltech-Office classification dataset

Domains	Base	SurK	SA	OT-IT	OT-MM	JDOT
caltech→amazon	92.07	91.65	90.50	89.98	**92.59**	91.54
caltech→webcam	76.27	77.97	81.02	80.34	78.98	**88.81**
caltech→dslr	84.08	82.80	85.99	78.34	76.43	**89.81**
amazon→caltech	84.77	84.95	85.13	85.93	**87.36**	85.22
amazon→webcam	79.32	81.36	**85.42**	74.24	85.08	84.75
amazon→dslr	86.62	87.26	**89.17**	77.71	79.62	87.90
webcam→caltech	71.77	71.86	75.78	**84.06**	82.99	82.64
webcam→amazon	79.44	78.18	81.42	89.56	90.50	**90.71**
webcam→dslr	96.18	95.54	94.90	**99.36**	**99.36**	98.09
dslr→caltech	77.03	76.94	81.75	**85.57**	83.35	84.33
dslr→amazon	83.19	82.15	83.19	**90.50**	**90.50**	88.10
dslr→webcam	96.27	92.88	88.47	**96.61**	**96.61**	**96.61**
Mean	83.92	83.63	85.23	86.02	86.95	**89.04**
Avg. rank	4.50	4.75	3.58	3.00	2.42	**2.25**

- Classical dataset [Saenko et al., 2010] dedicated to visual adaptation.
- Feature extraction by convolutional neural network [Donahue et al., 2014].
- Comparison with Surrogate Kernel [Zhang et al., 2013], Subspace Alignment [Fernando et al., 2013] and OT Domain Adaptation [Courty et al., 2016b].
- Parameter selected via reverse cross-validation [Zhong et al., 2010].
- SVM (Hinge loss) classifiers with linear kernel.
- Best ranking method and 2% accuracy gain in average.

Amazon Review Classification dataset

Excellent product that I completely hate, Apr 1, 2013

By Thirsty - See all my reviews

This review is from: Strollmaster 3000 (Baby Product)

The Strollmaster 3000 is every parent's dream - roomy, durable, safe, and easy to fold, with a unique 17-point harness. Best yet, it weighs just 1.6 lbs. and sells for an unbelievable $17.99. Unfortunately, it has one fatal flaw - the cupholder can only handle beverages up to 64 oz. I was dumbstruck as well. Is this America? I was left holding my 128 oz. Big Gulp like some kind of sucker. So, if you're into amazing, durable products that are a steal and virtually idyllic, then, sure, buy it. If you want to down a bathtub of Dr. Pepper, though. I'd pass.

Let it go... in the trash

By VPI1977 on December 24, 2017

Had high expectation, too much snow, too many animals, wish it had more ninjas. Also it would be better if these people ate more, I mean how are we suppose to make society better if people don't sit down to eat and socialize.

2 people found this helpful

Domains	**NN**	**DANN**	**JDOT** (mse)	**JDOT** (Hinge)
books→dvd	0.805	**0.806**	0.794	0.795
books→kitchen	0.768	0.767	0.791	**0.794**
books→electronics	0.746	0.747	0.778	**0.781**
dvd→books	0.725	0.747	0.761	**0.763**
dvd→kitchen	0.760	0.765	0.811	**0.821**
dvd→electronics	0.732	0.738	0.778	**0.788**
kitchen→books	0.704	0.718	**0.732**	0.728
kitchen→dvd	0.723	0.730	0.764	**0.765**
kitchen→electronics	**0.847**	0.846	0.844	0.845
electronics→books	0.713	0.718	0.740	**0.749**
electronics→dvd	0.726	0.726	**0.738**	0.737
electronics→kitchen	0.855	0.850	0.868	**0.872**
Mean	0.759	0.763	0.783	**0.787**

- Dataset aim at predicting reviews across domains [Blitzer et al., 2006].
- Comparison with Domain adversarial neural network [Ganin et al., 2016a].
- Classifier f is a neural network with same architecture as DANN.
- JDOT has better accuracy, classification loss is better than mean square error.

Wifi localization regression dataset

Domains	KRR	SurK	DIP	DIP-CC	GeTarS	CTC	CTC-TIP	JDOT
t1 $\rightarrow$ t2	80.84±1.14	90.36±1.22	87.98±2.33	91.30±3.24	86.76 ± 1.91	89.36±1.78	89.22±1.66	**93.03 ± 1.24**
t1 $\rightarrow$ t3	76.44±2.66	**94.97±1.29**	84.20±4.29	84.32±4.57	90.62±2.25	94.80±0.87	92.60 ± 4.50	90.06 ± 2.01
t2 $\rightarrow$ t3	67.12±1.28	85.83 ± 1.31	80.58 ± 2.10	81.22 ± 4.31	82.68 ± 3.71	87.92 ± 1.87	**89.52 ± 1.14**	86.76 ± 1.72
hallway1	60.02 ±2.60	76.36 ± 2.44	77.48 ± 2.68	76.24± 5.14	84.38 ± 1.98	86.98 ± 2.02	86.78 ± 2.31	**98.83±0.58**
hallway2	49.38 ± 2.30	64.69 ±0.77	78.54 ± 1.66	77.8± 2.70	77.38 ± 2.09	87.74 ± 1.89	87.94 ± 2.07	**98.45±0.67**
hallway3	48.42 ±1.32	65.73 ± 1.57	75.10± 3.39	73.40± 4.06	80.64 ± 1.76	82.02± 2.34	81.72 ± 2.25	**99.27±0.41**

- Objective is to predict position of a device on a discretized grid [Zhang et al., 2013].
- Same experimental protocol as [Zhang et al., 2013, Gong et al., 2016].
- Comparison with domain-invariant projection and its cluster regularized version ([Baktashmotlagh et al., 2013], **DIP** and **DIP-CC**), generalized target shift ([Zhang et al., 2015], **GeTarS**), and conditional transferable components, with its target information preservation regularization ([Gong et al., 2016], **CTC** and **CTC-TIP**).
- JDOT solves the adaptation problem for transfer across device (10% accuracy gain on Hallway).

Large scale JDOT

- JDOT do not scale well to large datasets/ deep learning.
- Use minibach for computing the transport in the primal [Genevay et al., 2017].
- 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

input Source data X^s, y^s, Targte data X^t

  for BCD Iterations **do**

    for each Source/Target minibatch **do**

      Solve OT with JDOT loss

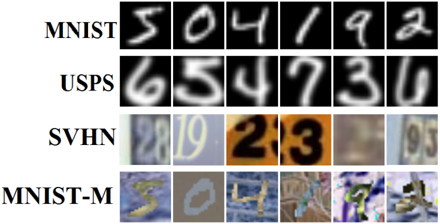
      Perform label propagation on minibatch

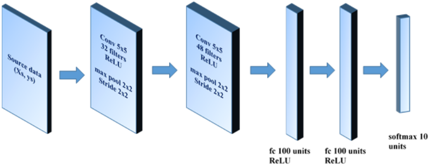
    end for

    Update model f on one epoch

  end for

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

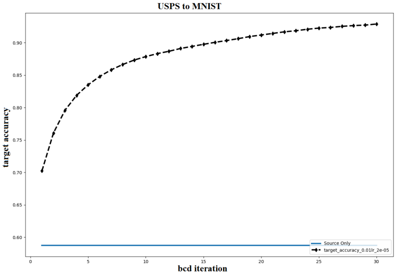
- Four cross domain digits datasets: MNIST, USPS, SVHN, MNIST-M .
- We consider a deep convolutional architecture.
- Dropout is used on the dens layers when training.
- Transport distance computed in the raw image space.

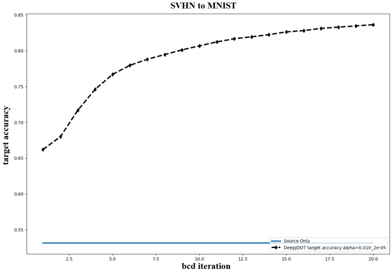
Experimental Results for large scale JDOT

Methods	MNIST→ USPS	USPS→MNIST	SVHN→MNIST	MNIST→ MNIST-M
Source only (SO)	86.18	58.73	53.15	59.52
DeepCoral [Sun and Saenko, 2016]	88.43 (22.0)	85.02 (64.6)	69.61 (35.6)	62.18 (0.07)
MMD [Long and Wang, 2015]	89.89 (36.3)	79.19 (50.3)	53.27 (0.01)	52.53 (-19.1)
DANN [Ganin et al., 2016b]	89.06 (28.2)	87.03 (70.0)	73.85* (44.7)	76.63 (46.6)
ADDA [Tzeng et al., 2017]	91.22 (49.3)	79.98 (52.2)	76.0* (49.4)	**79.16 (53.5)**
DeepJDOT	**91.50 (52.01)**	**91.21 (79.82)**	**83.62 (65.85)**	67.84 (22.67)
Train on Target (TO)	96.41	99.42	99.42	96.21

- Accuracy in % of the DA methods.
- The values in () represent the coverage gap between SO (source only) and TO (golden performance if the model is learnt on target labelled data), $\frac{DA-SO}{TO-SO}$.
- DeepJDOT is better in 3 out of 4 DA problems.
- Plots represent test performances along the BCD iterations.

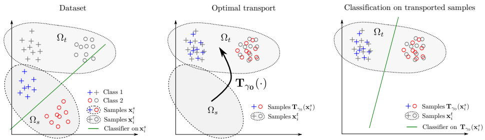
Experimental Results for large scale JDOT

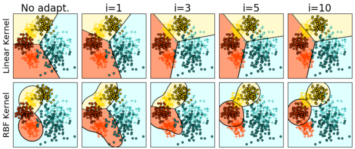




- Accuracy in % of the DA methods.
- The values in () represent the coverage gap between SO (source only) and TO (golden performance if the model is learnt on target labelled data), $\frac{DA-SO}{TO-SO}$.
- DeepJDOT is better in 3 out of 4 DA problems.
- Plots represent test performances along the BCD iterations.

Conclusion





Optimal transport for DA

- Model transformation of the features.
- Conditional distribution preserved.
- Mapping between distributions.
- Learn classifier on the transported samples.

Joint distribution OT for DA

- Model transformation of the joint distribution.
- General framework for DA.
- Theoretical justification with generalization bound.

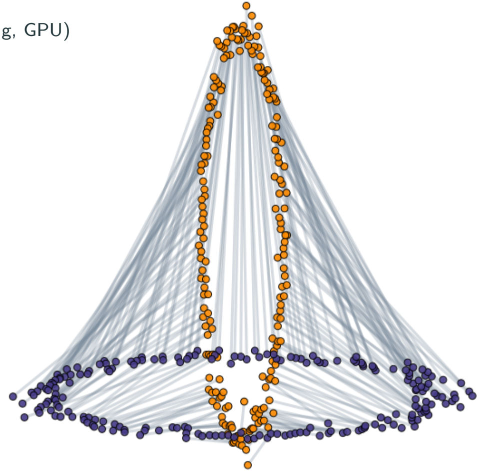
Next ?

- SGD OT on the semi-dual [Genevay et al., 2016] or dual [Seguy et al., 2017].
- Learn simultaneously the best feature representation [Shen et al., 2017].

Thank you

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

- OT LP solver, Sinkhorn (stabilized, $\epsilon-$scaling, GPU)
- Domain adaptation with OT.
- Barycenters, Wasserstein unmixing.
- Wasserstein Discriminant Analysis.



Python code for JDOT on GitHub:
`https://github.com/rflamary/JDOT`

Papers available on my website:
`https://remi.flamary.com/`

Post docs available in:
Nice, Rouen, Rennes (France)

References I

Baktashmotlagh, M., Harandi, M., Lovell, B., and Salzmann, M. (2013).
Unsupervised domain adaptation by domain invariant projection.
In *ICCV*, pages 769–776.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010).
A theory of learning from different domains.
Machine Learning, 79(1-2):151–175.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. (2015).
Iterative Bregman projections for regularized transportation problems.
SISC.

Blitzer, J., McDonald, R., and Pereira, F. (2006).
Domain adaptation with structural correspondence learning.
In *Proc. of the 2006 conference on empirical methods in natural language processing*, pages 120–128.

References II

Brenier, Y. (1991).
Polar factorization and monotone rearrangement of vector-valued functions.
Communications on pure and applied mathematics, 44(4):375–417.

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

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

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

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014).
Decaf: A deep convolutional activation feature for generic visual recognition.
In *ICML*.

Fernando, B., Habrard, A., Sebban, M., and Tuytelaars, T. (2013).
Unsupervised visual domain adaptation using subspace alignment.
In *ICCV*.

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. (2016a).
Domain-adversarial training of neural networks.
Journal of Machine Learning Research, 17(59):1–35.

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

Genevay, A., Cuturi, M., Peyré, G., and Bach, F. (2016).
Stochastic optimization for large-scale optimal transport.
In *NIPS*, pages 3432–3440.

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

Gong, B., Shi, Y., Sha, F., and Grauman, K. (2012).
Geodesic flow kernel for unsupervised domain adaptation.
In *CVPR*.

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., and Schölkopf, B. (2016).
Domain adaptation with conditional transferable components.
In *ICML*, volume 48, pages 2839–2848.

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

Long, M. and Wang, J. (2015).
Learning transferable features with deep adaptation networks.
CoRR, abs/1502.02791.

Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. (2014).
Transfer joint matching for unsupervised domain adaptation.
In *CVPR*, pages 1410–1417.

Mansour, Y., Mohri, M., and Rostamizadeh, A. (2009).
Domain adaptation: Learning bounds and algorithms.
In *Proc. of COLT*.

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

Pérez, P., Gangnet, M., and Blake, A. (2003).
Poisson image editing.
ACM Trans. on Graphics, 22(3).

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

R. Gopalan, R. L. and Chellappa, R. (2014).
Unsupervised adaptation across domain shifts by generating intermediate data representations.
IEEE Transactions on Pattern Analysis and Machine Intelligence, page To be published.

Rubner, Y., Tomasi, C., and Guibas, L. J. (2000).
The earth mover's distance as a metric for image retrieval.
International journal of computer vision, 40(2):99–121.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010).
Adapting visual category models to new domains.
In *ECCV*, LNCS, pages 213–226.

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

Shen, J., Qu, Y., Zhang, W., and Yu, Y. (2017).
Wasserstein distance guided representation learning for domain adaptation.
arXiv preprint arXiv:1707.01217.

Si, S., Tao, D., and Geng, B. (2010).
Bregman divergence-based regularization for transfer subspace learning.
IEEE Transactions on Knowledge and Data Engineering, 22(7):929–942.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. (2008).
Direct importance estimation for covariate shift adaptation.
Annals of the Institute of Statistical Mathematics, 60(4):699–746.

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., Saenko, K., and Darrell, T. (2017).
Adversarial discriminative domain adaptation.
CoRR, abs/1702.05464.

Zhang, K., Gong, M., and Schölkopf, B. (2015).
Multi-source domain adaptation: A causal view.
In *AAAI Conference on Artificial Intelligence*, pages 3150–3157.

Zhang, K., Zheng, V. W., Wang, Q., Kwok, J. T., Yang, Q., and Marsic, I. (2013).
Covariate shift in Hilbert space: A solution via surrogate kernels.
In *ICML*.

References X

Zhong, E., Fan, W., Yang, Q., Verscheure, O., and Ren, J. (2010).
Cross validation framework to choose amongst models and datasets for transfer learning.
In *ECML/PKDD*.