Domain adaptation with optimal transport

Gilles GASSO - LITIS Lab, INSA Rouen Normandy

CIMPA Research School 2021

Lomé, Togo

With the courtesy of Rémi Flamary, CMAP, Ecole Polytechnique, France

Table of content

Introduction

Domain adaptation

"Supervising" domain adaptation

Transfer learning

Formulation

Unsupervised domain adaptation

Optimal transport

Optimal transport for domain adaptation

Learning strategy and mapping estimation

Linear Monge mapping and generalization

Experiments and discussions

Joint distribution OT for domain adaptation (JDOT)

Joint distribution and classifier estimation

Generalization bound

Learning with JDOT : regression and classification

Numerical experiments

Introduction

Supervised learning

Amazon

Traditional supervised learning

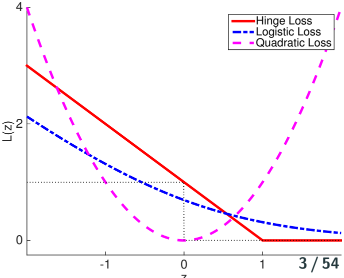
- We want to learn predictor such that $y \approx f(x)$, $f \in \mathcal{F}$.
- Actual $\mathcal{P}(X, Y)$ unknown.
- We have access to dataset $(x_i, y_i)_{i=1,\ldots,n}$ $(\widehat{\mathcal{P}}(X, Y))$.
- We choose a loss function $\mathcal{L}(y, f(x))$ that measure the discrepancy.

For binary classification

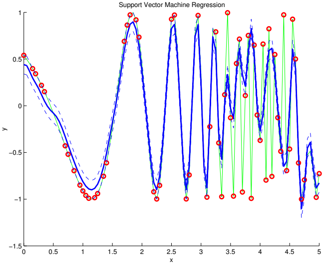
- We suppose $y \in \mathcal{Y} = \{-1, 1\}$
- 0 - 1 loss

$$\mathcal{L}(y, f(x)) = \mathbf{1}_{yf(x) \leq 0} = \begin{cases} 0 & \text{if} \quad yf(x) > 0 \\ 1 & \text{if} \quad yf(x) \leq 0 \end{cases}$$

measures the number of classification errors



Supervised learning



Traditional supervised learning

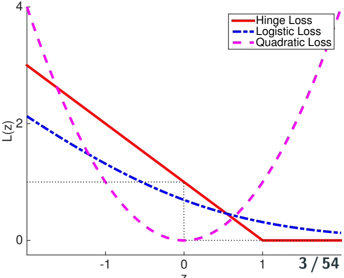
- We want to learn predictor such that $y \approx f(x)$, $f \in \mathcal{F}$.
- Actual $\mathcal{P}(X, Y)$ unknown.
- We have access to dataset $(x_i, y_i)_{i=1,\ldots,n}$ $(\widehat{\mathcal{P}}(X, Y))$.
- We choose a loss function $\mathcal{L}(y, f(x))$ that measure the discrepancy.

For regression

- We have $y \in \mathbb{R}$
- Least squares regression

$$\mathcal{L}y, f(x) = (y - f(x))^2$$

measures the square errors



Supervised learning

Amazon

Traditional supervised learning

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

Empirical risk minimization

- Empirical risk

$$\hat{R}(f) = \mathop{\mathbb{E}}_{(x,y)\sim\widehat{\mathcal{P}}} \mathcal{L}(y, f(x)) = \frac{1}{n} \sum_j \mathcal{L}(y_j, f(\mathbf{x}_j)) \tag{1}$$

- We seek for a model (predictor) minimizing the empirical risk

$$\hat{f} = \arg\min_f \left\{ \hat{R}(f) \right\} \tag{2}$$

Empirical risk minimization and over-fitting

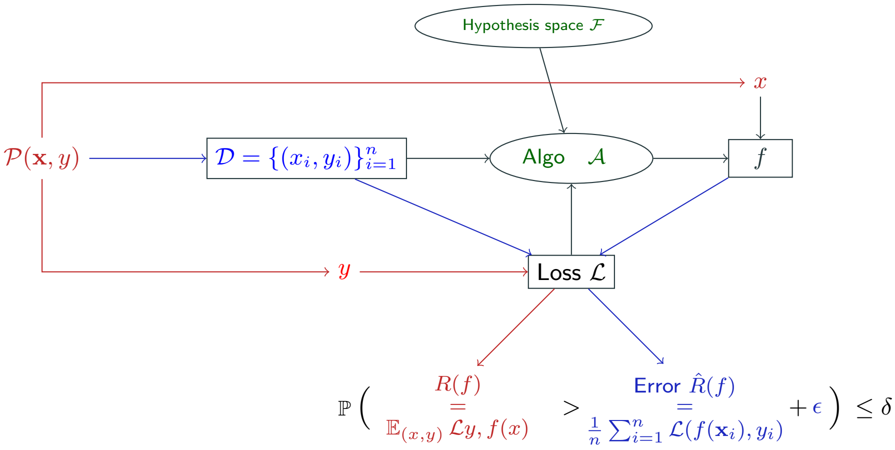
- Should we choose f based on $\hat{R}_(\hat{f})$? NO !
- as we can design a sufficiently complex function $\hat{f} \in f \in \mathcal{F}$ such that $\hat{R}(\hat{f}) \to 0$ but with high risk $R(\hat{f})$

  Recall the true expected risk is

$$R(f) = \underset{(\mathbf{x},y)\sim\mathcal{P}}{\mathbb{E}} \mathcal{L}(y, f(x)) = \int \mathcal{P}(x,y)\mathcal{L}(y, f(x))dxdy$$

$\Longrightarrow$ Control the complexity of the predictor f

The paradigm of statistical learning



With given $\mathcal{D}$, find a model f in a family $\mathcal{F}$ (linear, kernel SVM, Deep Network ...) with good generalization properties

Why learning occurs

Supremum on generalization error

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1\cdots n}$ be the dataset. Let $\mathcal{F}$ b a space of functions. For all $f \in \mathcal{F}$, with probability $1 - \delta$ we have

$$R(f) \leq \hat{R}(f) + \mathcal{O}\left(\sqrt{\frac{\zeta}{n}\log\frac{2en}{\zeta} + \frac{\log 2/\delta}{n}}\right)$$

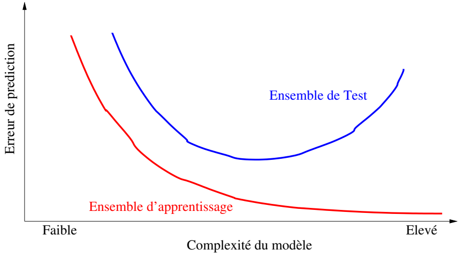
$\zeta > 0$ measures the "complexity" of the functions class $\mathcal{F}$

- Generalization occurs whenever $\zeta < \infty$
- Prefer $n >> \zeta$ (the number of available data increases with model complexity)

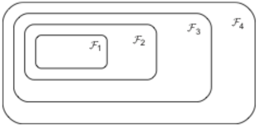
Generalization / over-fitting

$$R(f) \leq \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(f(x_i), y_i) + \text{term}(n, \zeta(\mathcal{F}))$$

- $\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(f(x_i), y_i)$ is not a good estimator of generalization ability
- Over-fitting appears with the increasing complexity of f



Complexity control: regularization



Let $k_1 < k_2 < k_3 < \cdots$

We define $\mathcal{F}_j = \{f : \Omega(f) \leq k_j\}$

$\Omega(f)$: regularisation function

Example : $\Omega(f) = \|f\|^2$

Minimization of the regularized empirical risk

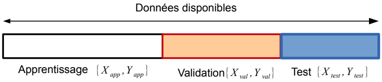
$$\min_f \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i) + \lambda \, \Omega(f)$$

- $\lambda > 0$: regularization parameter
- $\lambda >> 1 \rightarrow$ we encourage f to be of low complexity

Example : SVM $\min_f \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i) + \lambda \, \|f\|^2$ with loss $\mathcal{L}(y, f(x)) = \max(0, 1 - yf(x))$

Similar scheme is used to regularize the weights of a deep learning model (weight decay)

How to choose the "best" model?



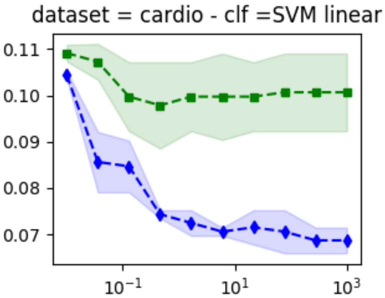
1. Randomly split available dataset $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{val} \cup \mathcal{D}_{test}$
2. Train several models with different levels of "complexity" on $\mathcal{D}_{train}$
3. Evaluate their performances (classification error, mean square error...) on $\mathcal{D}_{val}$
4. Select the model with the best performance on $\mathcal{D}_{val}$
5. Test the selected model on $\mathcal{D}_{test}$

Remark

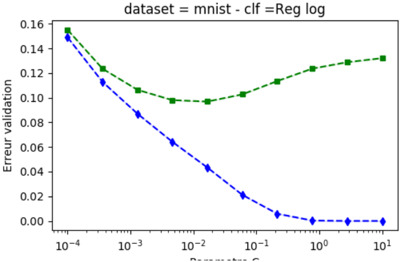
- $\mathcal{D}_{test}$ is used only once!

Cross-validation at work

K-Fold Cross-Validation



Cross-Validation

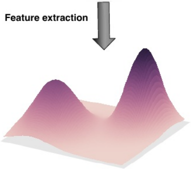


Which model to select?

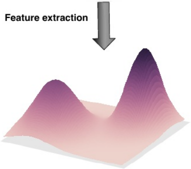
Implicit assumption

Training data (source)

Test data (target)



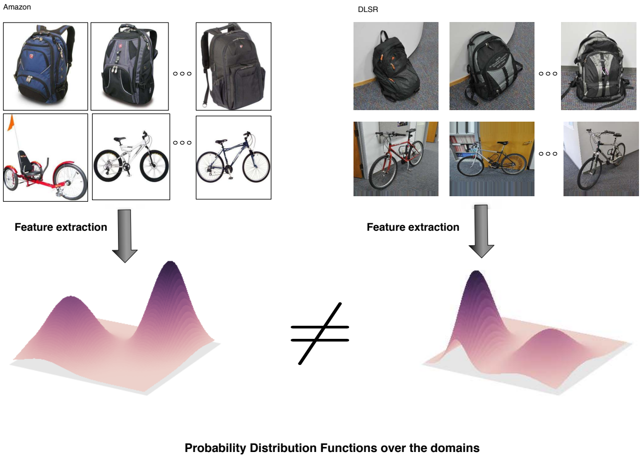
$$\approx$$



Remark

- Training and test data are expected to be drawn from the same (unknown) joint distribution $\mathcal{P}(X, Y)$

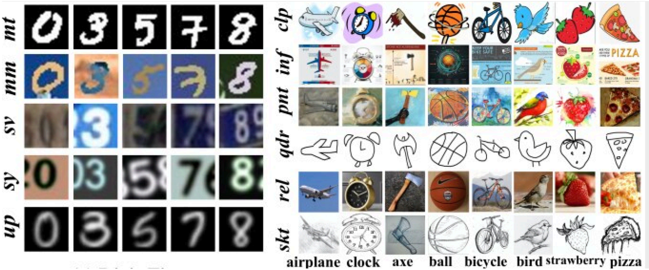
Domain Adaptation problem



Our context

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

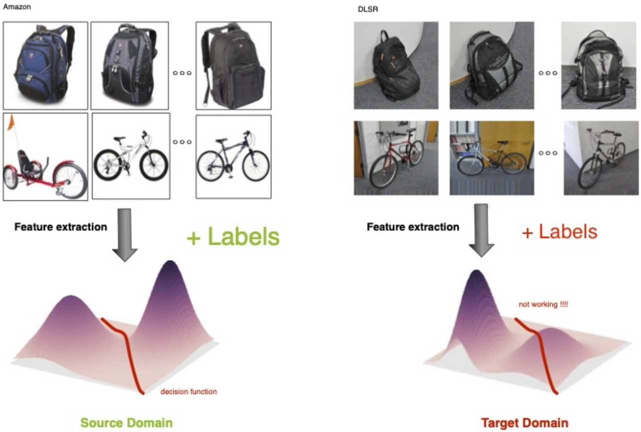
Domain adaptation



What is domain adaptation?

- Differences in instances difference $\not\Longrightarrow$ in the predictions
- Transfer knowledge from previous domain to a new domain to overcome the differences
- Domains are somehow related

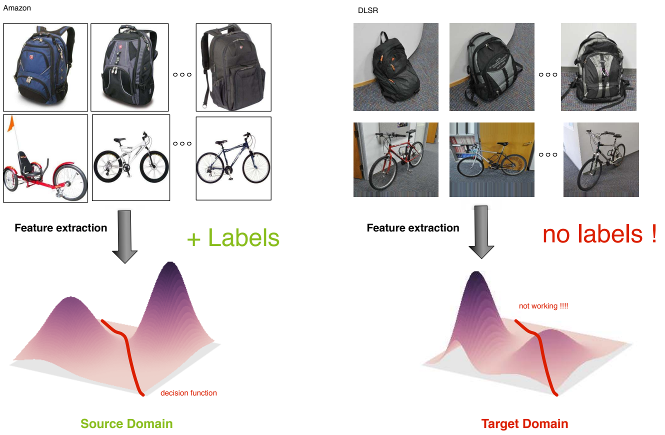
Supervised domain adaptation



Problems

- **Large labeled data** are available on source domain but only a **few labeled target data** are at disposal in the **source domain**,
- Classifier trained on the source domain data performs badly in the target domain

Unsupervised domain adaptation problem

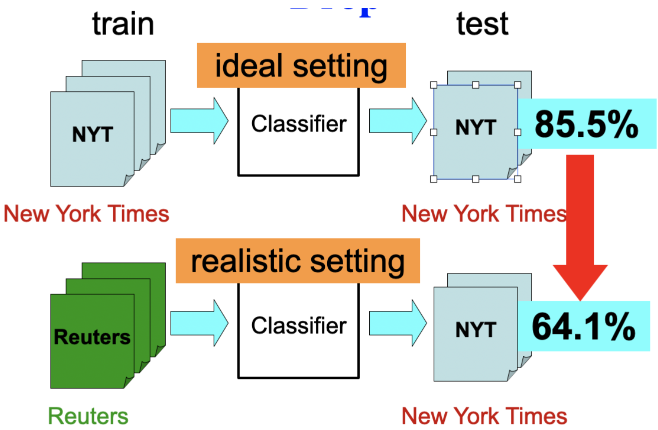


Problems

- 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

"Supervising" domain adaptation

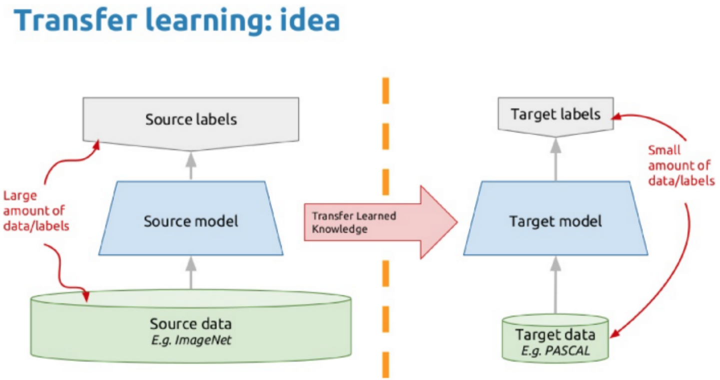
Transfer Learning



Distribution shift results in drop in performances!

Transfer Learning principle

Train on one (several) task(s), transfer on a new related one

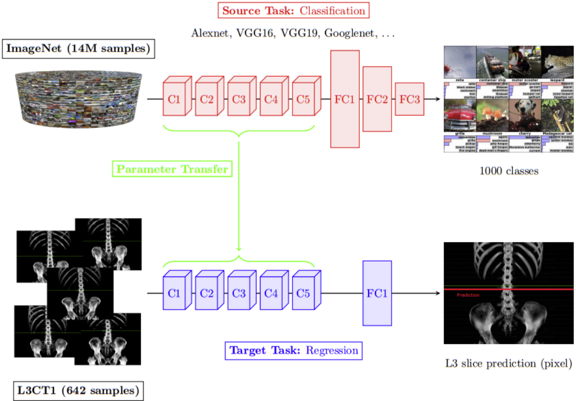


From `https://tinyurl.com/ykzxrkwe`

How to leverage large labeled source dataset to train the target model?

Example of image classification

- Train a base model (AlexNet, VGG16, etc.) using large scale source data (ImageNet) or upload pre-trained models
- Freeze part or full hidden layers parameters
- Fine-tune unfrozen layers of the base model using the few target labeled data



Formally

Notations

Source data are labeled $\mathcal{D}_s = \{(x_i^s, y_i^s) \in \mathcal{X}_s \times \mathcal{Y}_s\}_{i=1}^{n_s}$

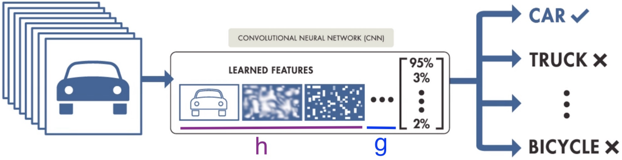
Target samples are only a few $\mathcal{D}_t = \{(x_j^t, y_j^t) \in \mathcal{X}_t\}_{j=1}^{n_t}$

	Joint dis.	Marginal dis.	Conditional dis.	Label dis.
Source	$\mathcal{P}_s(x, y)$	$\mathcal{P}_s(x)$	$\mathcal{P}_s(y/x)$	$\mathcal{P}_s(y)$
Target	$\mathcal{P}_t(x, y)$	$\mathcal{P}_t(x)$	$\mathcal{P}_t(y/x)$	$\mathcal{P}_t(y)$

Common assumptions

- Same instance and label spaces $\mathcal{X}_s = \mathcal{X}_t$ and $\mathcal{Y}_s = \mathcal{Y}_t$
- Joint distributions are drifted $\mathcal{P}_s(x, y) \neq \mathcal{P}_t(x, y)$
  - Covariate shift $\mathcal{P}_s(x) \neq \mathcal{P}_t(x)$ but $\mathcal{P}_s(y/x) \simeq \mathcal{P}_t(y/x)$
  - Label shift $\mathcal{P}_s(y) \neq \mathcal{P}_t(y)$ but $\mathcal{P}_s(x/y) \simeq \mathcal{P}_t(x/y)$

Formulation



- Let the source model be $f_s(x) = g_s \circ h(x)$ with h: the feature extraction map, and g_s: the classification function
- Train f_s on source data

$$\hat{R}(h, g_s) = \underset{(x^s, y^s) \sim \widehat{\mathcal{P}_s}}{\mathbb{E}} \mathcal{L}(y^s, g_s \circ h(x^s)) = \frac{1}{n_s} \sum_j \mathcal{L}(y_i^s, g_s \circ h(x_i^s)) \tag{3}$$

$$\hat{g}_s, \hat{h} = \arg\min_{g_s, h} \left\{ \hat{R}(h, g_s) \right\}$$

- Target model: $f_t(x) = g_t \circ h(x)$. h is shared with both f_s and f_t
- Keep h unchanged and tune g_t

$$\hat{g}_t = \arg\min_{g_t} \left\{ \underset{(x^t, y^t) \sim \widehat{\mathcal{P}_t}}{\mathbb{E}} \mathcal{L}(y^t, g_t \circ h(x^t)) = \frac{1}{n_t} \sum_j \mathcal{L}(y_j^t, g_s \circ h(x_j^t)) \right\}$$

Extension to multiple source domains

- Assume $S > 1$ source domains (tasks) with models being $f_s(x) = g_s \circ h(x)$, $s = 1, \ldots, S$
- Learn the shared representation function h

$$\hat{R}(h, g_1, \cdots, g_S) = \frac{1}{S} \sum_s \mathop{\mathbb{E}}_{(x^s, y^s) \sim \widehat{\mathcal{P}_s}} \mathcal{L}(y^s, g_s \circ h(x^s)) = \frac{1}{S n_s} \sum_s \sum_j \mathcal{L}(y_i^s, g_s \circ h(x_i^s))$$

$$\hat{h} = \arg\min_h \min_{g_1, \cdots, g_S} \left\{ \hat{R}(h, g_1, \cdots, g_S) \right\}$$

- Target model: $f_t(x) = g_t \circ h(x)$

$$\hat{g}_t = \arg\min_{g_t} \left\{ \mathop{\mathbb{E}}_{(x^t, y^t) \sim \widehat{\mathcal{P}_t}} \mathcal{L}(y^t, g_t \circ h(x^t)) = \frac{1}{n_t} \sum_j \mathcal{L}(y_j^t, g_s \circ h(x_j^t)) \right\}$$

Theoretical guarantees

- Target domain model: $f_t(x) = g_t \circ h(x)$ with h the shared representation function with the source domain model(s)

  With probability at least $1-\delta$, $\delta \in (0,1)$ [Tripuraneni et al., 2020]

$$R(\hat{g}_t, \hat{h}) \leq R(g_t^*, h^*) + \mathsf{dist}_{\mathcal{G}_t,\mathcal{G}_s}(\hat{h}, h^*) + \zeta(\mathcal{G}_s) + 8B\sqrt{\frac{\log 2/\delta}{n_t}}$$

- Worst-case representation distance

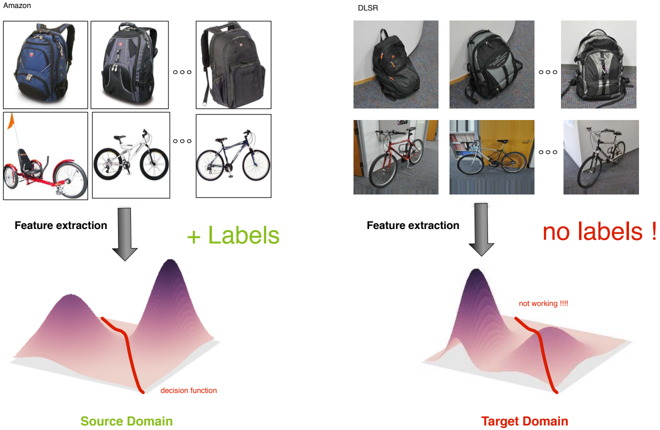
$$\mathsf{dist}_{\mathcal{G}_t,\mathcal{G}_s}(\hat{h}, h^*) = \sup_{g_t \in \mathcal{G}_t} \inf_{g_s \in \mathcal{G}_s} \mathop{\mathbb{E}}_{(x,y)} \left\{ \mathcal{L}(y, g_s \circ \hat{h}(x)) - \mathcal{L}(y, g_t \circ h^*(x)) \right\}$$

  measures the error due to using a biased feature representation $\hat{h} \neq h^*$

Generalization error bound depends on the complexity of the hypothesis, on the distance beetween source domain representation and the suitable target domain one

Unsupervised domain adaptation

Unsupervised domain adaptation problem



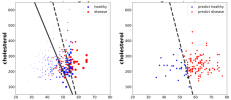
Problems

- 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

Domain adaptation short state of the art

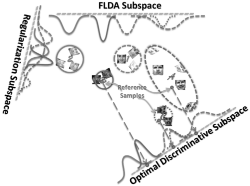
Reweighting schemes [Sugiyama et al., 2008]

- Distribution change across domains.
- Re-weight source samples by $\dfrac{\mathcal{P}_t(x^s)}{\mathcal{P}_s(x^s)}$ to compensate this change.



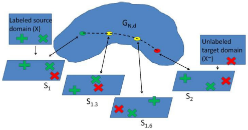
Subspace methods

- 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].



The underlying principle

Problem

We seek for a model f able to work either on source and target domains

Bounding the adaptation risk [Ben-David et al., 2010]

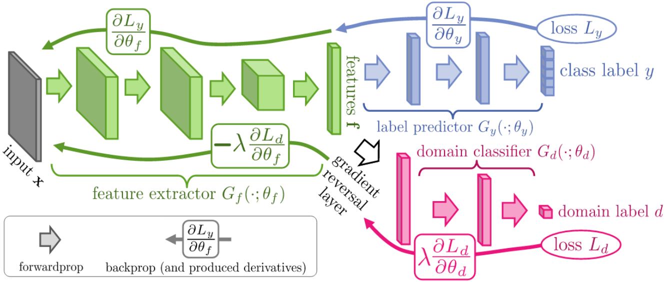
$$R_t(f) \leq R_s(f) + Div(\mathcal{P}_s(x), \mathcal{P}_t(x)) + \beta$$

- What we should care about: measure of distribution shift $Div(\mathcal{P}_s(x), \mathcal{P}_t(x))$
- What we expect: domain relatedness measured by $\beta = \inf_f R_s(f) + R_t(f)$

Most DA strategies

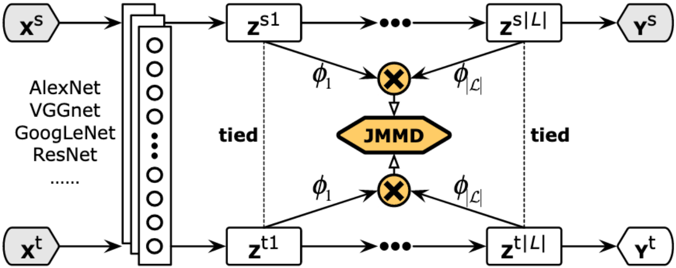
- Choose f with good properties (to get β minimal)
- Minimize distribution discrepancy

Domain adversarial network [Ganin et al., 2016]



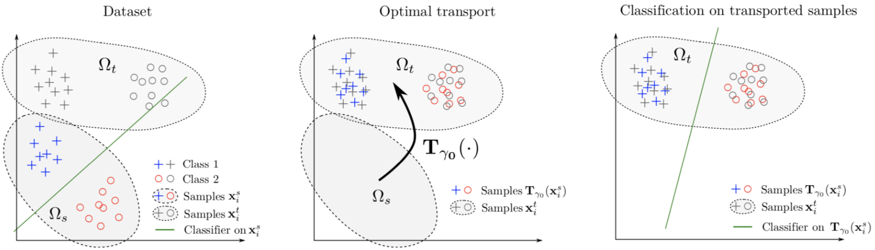
- Mapping source and target instances onto a domain-invariant latent subspace
- Ensure good prediction on source domain

Joint adaptation network [Long et al., 2017]

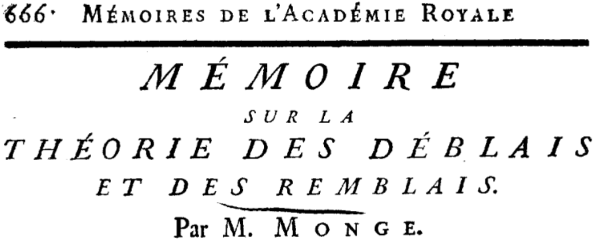


- Jointly align feature distributions across layers
- Based on kernel Maximum Mean Discrepancy between layer activation distributions $Div(\mathcal{P}_s(x), P_t(x)) \equiv \|m_z(\mathcal{P}_s) - m_z(\mathcal{P}_t)\|^2_{\mathcal{H}}$

Optimal transport domain adaptation [Courty et al., 2016]



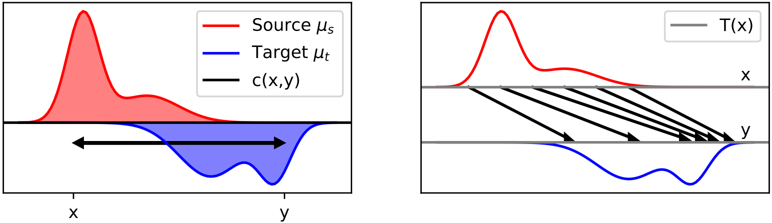
- Estimate a push-forward operator $\mathbf{T}$ between source and target distributions
- Map source samples onto target domain
- Learn a classification function


666. MÉMOIRES DE L'ACADÉMIE ROYALE

MÉMOIRE
SUR LA
THÉORIE DES DÉBLAIS
ET DES REMBLAIS.
Par M. MONGE.

Problem [Monge, 1781]

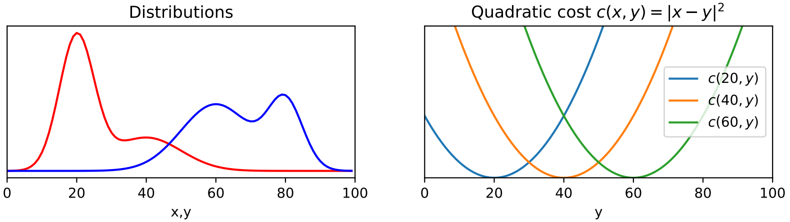
- How to move dirt from one place (déblais) to another (remblais) while minimizing the effort ?
- Find a mapping T between the two distributions of mass (transport).
- Optimize with respect to a displacement cost $c(x, y)$ (optimal).



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)

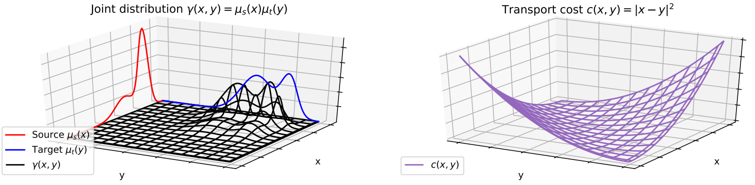


- Probability measures μ_s and μ_t on and a cost function $c : \Omega_s \times \Omega_t \rightarrow \mathbb{R}^+$.
- The Monge formulation [Monge, 1781] aim at finding a mapping $T : \Omega_s \rightarrow \Omega_t$

$$\inf_{T\#\mu_s=\mu_t} \quad \int_{\Omega_s} c(\mathbf{x}, T(\mathbf{x}))\mu_s(\mathbf{x})d\mathbf{x} \tag{4}$$

- 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)

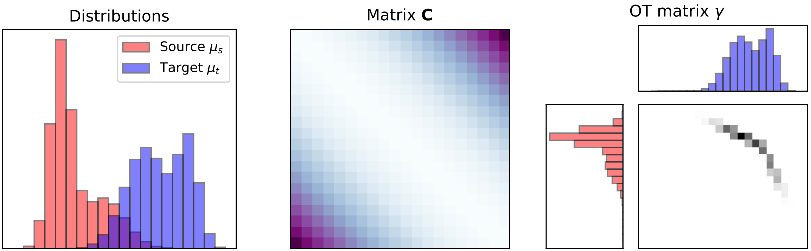


- The Kantorovich formulation [Kantorovich, 1942] seeks for a probabilistic coupling $\gamma \in \mathcal{P}(\Omega_s \times \Omega_t)$ between Ω_s and Ω_t:

$$\boldsymbol{\gamma}_0 = \underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \int_{\Omega_s \times \Omega_t} c(\mathbf{x}, \mathbf{y})\gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}, \tag{5}$$

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

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



OT Linear Program

When $\mu_s = \sum_{i=1}^{n} a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_{i=1}^{n} b_i \delta_{\mathbf{x}_i^t}$

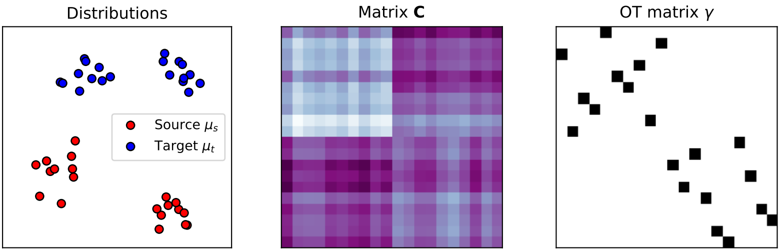
$$\boldsymbol{\gamma}_0 = \underset{\boldsymbol{\gamma} \in \mathcal{U}}{\operatorname{argmin}} \quad \left\{ \langle \boldsymbol{\gamma}, \mathbf{C} \rangle_F = \sum_{i,j} \gamma_{i,j} c_{i,j} \right\}$$

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

$$\mathcal{U} = \left\{ \boldsymbol{\gamma} \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} | \, \boldsymbol{\gamma} \mathbf{1}_{\mathbf{n_t}} = \mathbf{a}, \boldsymbol{\gamma}^{\mathbf{T}} \mathbf{1}_{\mathbf{n_s}} = \mathbf{b} \right\}$$

Linear program with $n_s n_t$ variables and $n_s + n_t$ constraints. Demo

Optimal transport with discrete distributions



OT Linear Program

When $\mu_s = \sum_{i=1}^{n} a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_{i=1}^{n} b_i \delta_{\mathbf{x}_i^t}$

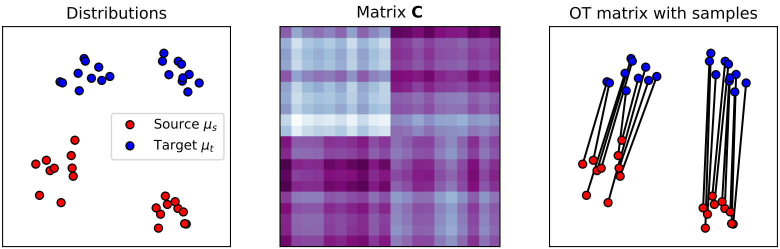
$$\boldsymbol{\gamma}_0 = \underset{\boldsymbol{\gamma} \in \mathcal{U}}{\operatorname{argmin}} \quad \left\{ \langle \boldsymbol{\gamma}, \mathbf{C} \rangle_F = \sum_{i,j} \gamma_{i,j} c_{i,j} \right\}$$

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

$$\mathcal{U} = \left\{ \boldsymbol{\gamma} \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} | \, \boldsymbol{\gamma} \mathbf{1}_{\mathbf{n_t}} = \mathbf{a}, \boldsymbol{\gamma}^{\mathbf{T}} \mathbf{1}_{\mathbf{n_s}} = \mathbf{b} \right\}$$

Linear program with $n_s n_t$ variables and $n_s + n_t$ constraints. Demo

Optimal transport with discrete distributions



OT Linear Program

When $\mu_s = \sum_{i=1}^{n} a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_{i=1}^{n} b_i \delta_{\mathbf{x}_i^t}$

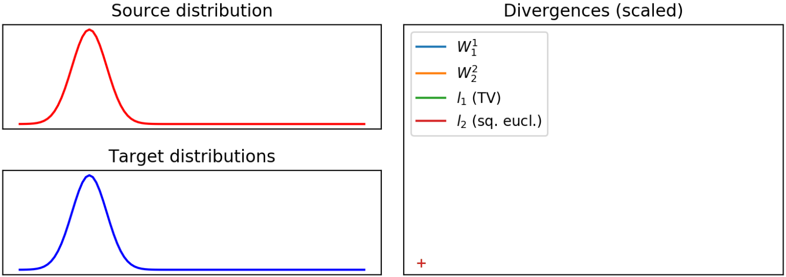
$$\boldsymbol{\gamma}_0 = \underset{\boldsymbol{\gamma} \in \mathcal{U}}{\text{argmin}} \quad \left\{ \langle \boldsymbol{\gamma}, \mathbf{C} \rangle_F = \sum_{i,j} \gamma_{i,j} c_{i,j} \right\}$$

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

$$\mathcal{U} = \left\{ \boldsymbol{\gamma} \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} | \, \boldsymbol{\gamma} \mathbf{1}_{\mathbf{n_t}} = \mathbf{a}, \boldsymbol{\gamma}^{\mathbf{T}} \mathbf{1}_{\mathbf{n_s}} = \mathbf{b} \right\}$$

Linear program with $n_s n_t$ variables and $n_s + n_t$ constraints. Demo

Wasserstein distance

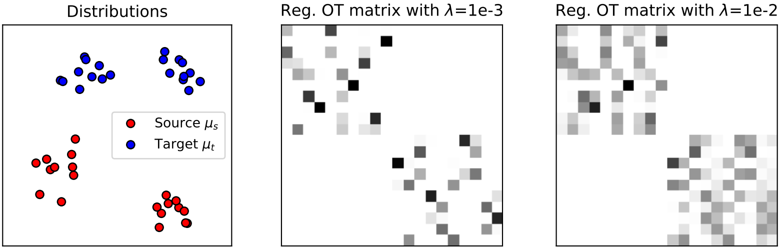


Wasserstein distance

$$W_p^p(\mu_s, \mu_t) = \min_{\gamma \in \mathcal{U}} \int_{\Omega_s \times \Omega_t} c(\mathbf{x}, \mathbf{y}) \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} = E_{(\mathbf{x}, \mathbf{y}) \sim \gamma}[c(\mathbf{x}, \mathbf{y})] \tag{6}$$

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).



Entropic regularization [Cuturi, 2013]

$$\gamma_0^\lambda = \underset{\gamma \in \mathcal{U}}{\operatorname{argmin}} \; \langle \gamma, \mathbf{C} \rangle_F - \lambda \Omega(\gamma), \tag{7}$$

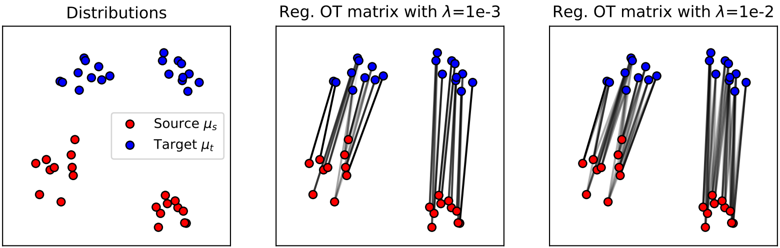
where $\Omega(\gamma) = -\sum_{i,j} \gamma(i,j) \log \gamma(i,j)$ computes the entropy of γ and

$$\mathcal{U} = \left\{ \gamma \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} \,|\, \gamma \mathbf{1}_{\mathbf{n_t}} = \mathbf{a}, \gamma^{\mathbf{T}} \mathbf{1}_{\mathbf{n_s}} = \mathbf{b} \right\}$$

.

- Entropy introduces smoothness.
- **Sinkhorn-Knopp** algorithm (efficient implementation in parallel, GPU).

Efficient regularized optimal transport



Entropic regularization [Cuturi, 2013]

$$\gamma_0^\lambda = \underset{\gamma \in \mathcal{U}}{\operatorname{argmin}} \langle \gamma, \mathbf{C} \rangle_F - \lambda \Omega(\gamma), \tag{7}$$

where $\Omega(\gamma) = -\sum_{i,j} \gamma(i,j) \log \gamma(i,j)$ computes the entropy of γ and

$$\mathcal{U} = \left\{ \gamma \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} \,|\, \gamma \mathbf{1_{n_t}} = \mathbf{a}, \gamma^{\mathbf{T}} \mathbf{1_{n_s}} = \mathbf{b} \right\}$$

.

- Entropy introduces smoothness.
- **Sinkhorn-Knopp** algorithm (efficient implementation in parallel, GPU).

Resolving the entropy regularized problem

Entropy-regularized transport

The solution of entropy regularized optimal transport problem is of the form
$$\boldsymbol{\gamma}_0^\lambda = \text{diag}(\mathbf{u}) \exp(-\mathbf{C}/\lambda) \text{diag}(\mathbf{v})$$

Why ? Consider the Lagrangian of the optimization problem:

$$L(\boldsymbol{\gamma}, \mathbf{u}, \mathbf{v}) = \sum_{ij} \boldsymbol{\gamma}_{ij} \mathbf{C}_{ij} + \lambda \boldsymbol{\gamma}_{ij} (\log \boldsymbol{\gamma}_{ij} - 1) + \mathbf{u}^{\mathbf{T}} (\boldsymbol{\gamma} \mathbf{1}_{n_t} - \mathbf{a}) + \mathbf{v}^{\mathbf{T}} (\boldsymbol{\gamma}^T \mathbf{1}_{n_s} - \mathbf{b})$$

$$\begin{aligned} \partial \mathcal{L} / \partial \boldsymbol{\gamma}_{ij} \quad &= \quad \mathbf{C}_{ij} + \lambda \log \boldsymbol{\gamma}_{ij} + u_i + v_j \\ \partial L / \partial \boldsymbol{\gamma}_{ij} = 0 \quad &\Longrightarrow \quad \boldsymbol{\gamma}_{ij} = \exp\left(\frac{u_i}{\lambda}\right) \exp\left(-\frac{\mathbf{C}_{ij}}{\lambda}\right) \exp\left(\frac{v_j}{\lambda}\right) \end{aligned}$$

- Through the **Sinkhorn theorem** diag$(\mathbf{u})$ and diag$(\mathbf{v})$ exist and are unique.
- Can be solved by the **Sinkhorn-Knopp** algorithm (implementation in parallel, GPU).

Sinkhorn-Knopp algorithm

The Sinkhorn-Knopp algorithm performs alternatively a scaling along the rows and columns of $\mathbf{K} = \exp(-\frac{\mathbf{C}}{\lambda})$ to match the desired marginals.

Algorithm 1 Sinkhorn-Knopp Algorithm (SK).

Require: $\mathbf{a}, \mathbf{b}, \mathbf{C}, \lambda$

  $\mathbf{u}^{(0)} = \mathbf{1}, \mathbf{K} = \exp(-\mathbf{C}/\lambda)$

  for i in $1, \ldots, n_{it}$ **do**

    $\mathbf{v}^{(i)} = \mathbf{b} \oslash \mathbf{K}^\top \mathbf{u}^{(i-1)}$ // Update right scaling

    $\mathbf{u}^{(i)} = \mathbf{a} \oslash \mathbf{K}\mathbf{v}^{(i)}$ // Update left scaling

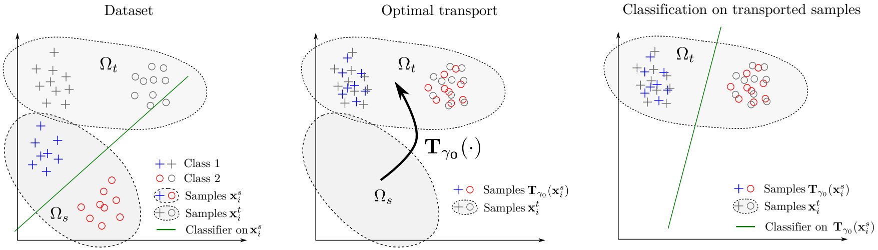
  end for

  return $\mathcal{T} = \text{diag}(\mathbf{u}^{(n_{it})})\mathbf{K}\text{diag}(\mathbf{v}^{(n_{it})})$

- Complexity $O(kn^2)$, where k iterations are required to reach convergence
- Fast implementation in parallel, GPU friendly
- Convolutive/Heat structure for K [Solomon et al., 2015]

Optimal transport for domain adaptation

Optimal transport for domain adaptation



Assumptions

- There exists an OT mapping T in the feature space between the two domains.
- The transport preserves the joint distributions:

$$\mathcal{P}_s(\mathbf{x}_s, y) = \mathcal{P}_t(T(\mathbf{x}_s), y).$$

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

1. Estimate optimal transport between distributions.
2. Transport the training samples on target domain.
3. Learn a classifier on the transported training samples.

Can be done the other way but needs a mapping for new samples.

Why does OTDA it work?

Expected risk

Let $R_s(f)$ be the expected risk of function f on the source domain.

$$R_s(f) := \mathbb{E}_{(x,y)\sim\mathcal{P}_s}\left[L(y, f(x))\right]. \tag{8}$$

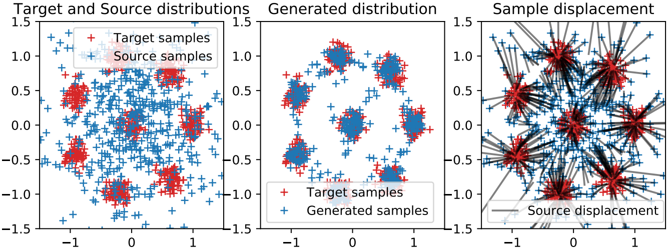
$R_t(f)$ is the expected risk in the target domain.

Generalization bound [Flamary et al., 2019]

Let f be a prediction rule in the source domain with a Lispschitz constant M_f and R_p the expected risk on domain p with a Lispschitz continuous loss L of constant M_L. Under the OTDA assumptions we have the following generalization bound

$$R_t(f \circ \hat{T}^{-1}) \leq R_s(f) + M_f M_L \mathbb{E}_{(x,y)\sim\mathcal{P}_s}\left[\|\hat{T}^{-1}(T(x)) - \hat{T}^{-1}(\hat{T}(x))\|\right] \tag{9}$$

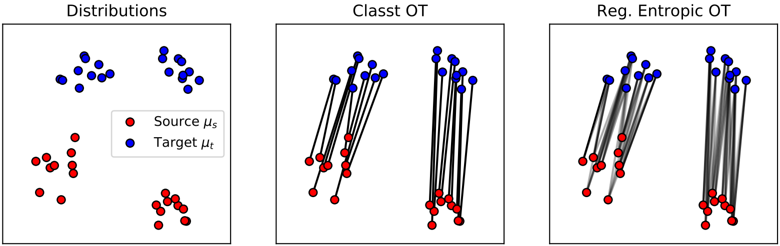
- Train a classifier f on source and estimate a mapping $\hat{T}^{-1}$ from target to source.
- True for any mapping T.
- Need out of sample mapping $\hat{T}^{-1}$ (to map new target samples).



Monge mapping estimation

- Mapping do not exist in general between empirical distributions.
- Barycentric mapping [Ferradans et al., 2014].
- Smooth mapping estimation [Perrot et al., 2016, Seguy et al., 2017].
- Closed form exist for transport between Gaussian distributions.
- Question of estimating the Monge Mapping: still an open problem theory suggests very hard ($O(n^{-1/d})$ [Hütter and Rigollet, 2019]) .

Transporting the discrete samples



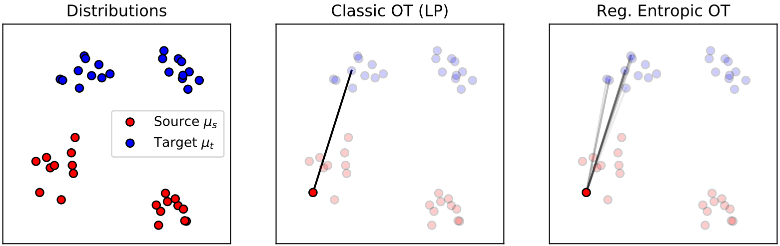
Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \underset{\mathbf{x}}{\operatorname{argmin}} \quad \sum_j \gamma_0(i,j) c(\mathbf{x}, \mathbf{x}_j^t). \tag{10}$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.

$$\widehat{\mathbf{x}_i^s} = \underset{\mathbf{x}}{\operatorname{argmin}} \quad \sum_j \gamma_0(i,j) c(\mathbf{x}, \mathbf{x}_j^t). \tag{11}$$

$$\hat{\mathbf{X}}_s = \operatorname{diag}(\gamma_0 \mathbf{1}_{n_t})^{-1} \gamma_0 \mathbf{X}_t \quad \text{and} \quad \hat{\mathbf{X}}_t = \operatorname{diag}(\gamma_0^\top \mathbf{1}_{n_s})^{-1} \gamma_0^\top \mathbf{X}_s. \tag{12}$$



Barycentric mapping [Ferradans et al., 2014]

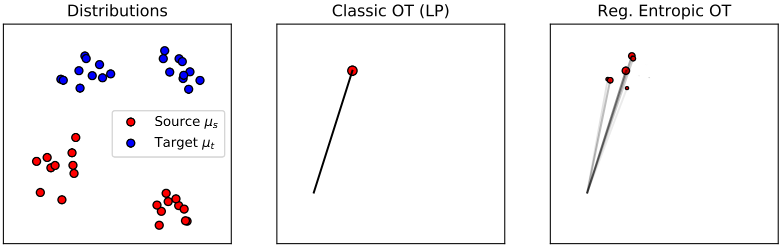
$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \operatorname*{argmin}_{\mathbf{x}} \quad \sum_j \gamma_0(i,j)\|\mathbf{x}-\mathbf{x}_j^t\|^2. \qquad (10)$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.

$$\widehat{\mathbf{x}_i^s} = \operatorname*{argmin}_{\mathbf{x}} \quad \sum_j \gamma_0(i,j)c(\mathbf{x},\mathbf{x}_j^t). \qquad (11)$$

$$\hat{\mathbf{X}}_s = \text{diag}(\gamma_0 \mathbf{1}_{n_t})^{-1}\gamma_0 \mathbf{X}_t \quad \text{and} \quad \hat{\mathbf{X}}_t = \text{diag}(\gamma_0^\top \mathbf{1}_{n_s})^{-1}\gamma_0^\top \mathbf{X}_s. \qquad (12)$$

Transporting the discrete samples



Barycentric mapping [Ferradans et al., 2014]

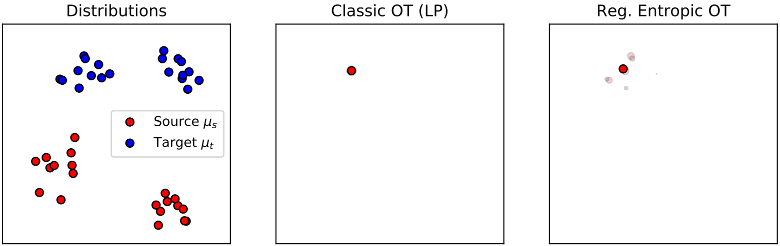
$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \gamma_0(i,j)} \sum_j \gamma_0(i,j)\mathbf{x}_j^t. \tag{10}$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.

$$\widehat{\mathbf{x}_i^s} = \operatorname*{argmin}_{\mathbf{x}} \sum_j \gamma_0(i,j) c(\mathbf{x}, \mathbf{x}_j^t). \tag{11}$$

$$\hat{\mathbf{X}}_s = \text{diag}(\gamma_0 \mathbf{1}_{n_t})^{-1}\gamma_0 \mathbf{X}_t \quad \text{and} \quad \hat{\mathbf{X}}_t = \text{diag}(\gamma_0^\top \mathbf{1}_{n_s})^{-1}\gamma_0^\top \mathbf{X}_s. \tag{12}$$

Transporting the discrete samples



Barycentric mapping [Ferradans et al., 2014]

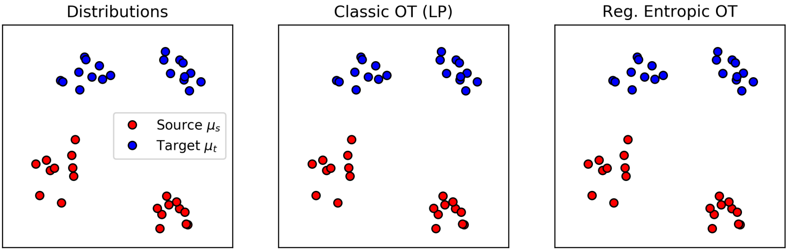
$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \gamma_0(i,j)} \sum_j \gamma_0(i,j)\mathbf{x}_j^t. \qquad (10)$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.

$$\widehat{\mathbf{x}_i^s} = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_j \gamma_0(i,j)c(\mathbf{x}, \mathbf{x}_j^t). \qquad (11)$$

$$\hat{\mathbf{X}}_s = \operatorname{diag}(\gamma_0 \mathbf{1}_{n_t})^{-1}\gamma_0 \mathbf{X}_t \quad \text{and} \quad \hat{\mathbf{X}}_t = \operatorname{diag}(\gamma_0^\top \mathbf{1}_{n_s})^{-1}\gamma_0^\top \mathbf{X}_s. \qquad (12)$$

Transporting the discrete samples



Barycentric mapping [Ferradans et al., 2014]

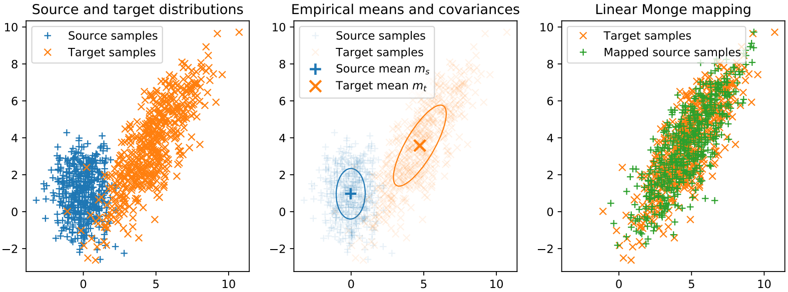
$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \gamma_0(i,j)} \sum_j \gamma_0(i,j)\mathbf{x}_j^t. \quad (10)$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.

$$\widehat{\mathbf{x}_i^s} = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_j \gamma_0(i,j)c(\mathbf{x}, \mathbf{x}_j^t). \quad (11)$$

$$\hat{\mathbf{X}}_s = \text{diag}(\gamma_0 \mathbf{1}_{n_t})^{-1}\gamma_0 \mathbf{X}_t \quad \text{and} \quad \hat{\mathbf{X}}_t = \text{diag}(\gamma_0^\top \mathbf{1}_{n_s})^{-1}\gamma_0^\top \mathbf{X}_s. \quad (12)$$

Special case: OT mapping between Gaussians



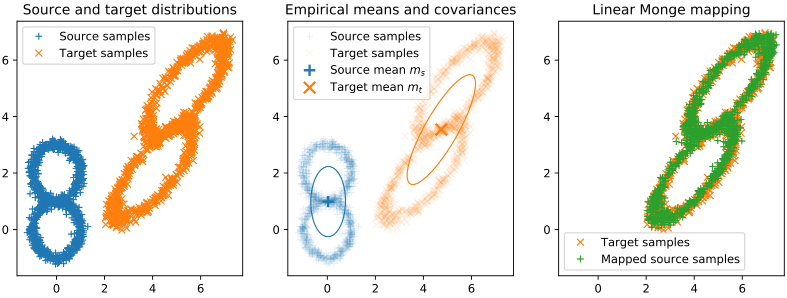
OT mapping between Gaussian distributions

- $\mu_s \sim \mathcal{N}(\mathbf{m}_1, \Sigma_1)$ and $\mu_t \sim \mathcal{N}(\mathbf{m}_2, \Sigma_2)$
- The optimal map T for $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$ is given by

$$T(\mathbf{x}) = \mathbf{m}_2 + A(\mathbf{x} - \mathbf{m}_1)$$

  with $A = \Sigma_1^{-1/2}(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}\Sigma_1^{-1/2}$.
- Can be estimated from empirical distributions.
- Linear mapping for any distributions with a density [Flamary et al., 2019].

Special case: OT mapping between Gaussians



OT mapping between Gaussian distributions

- $\mu_s \sim \mathcal{N}(\mathbf{m}_1, \Sigma_1)$ and $\mu_t \sim \mathcal{N}(\mathbf{m}_2, \Sigma_2)$
- The optimal map T for $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$ is given by

$$T(\mathbf{x}) = \mathbf{m}_2 + A(\mathbf{x} - \mathbf{m}_1)$$

  with $A = \Sigma_1^{-1/2}(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}\Sigma_1^{-1/2}$.
- Can be estimated from empirical distributions.
- Linear mapping for any distributions with a density [Flamary et al., 2019].

Empirical estimation of linear Monge mapping

- Empirical estimation of Gaussian parameters for μ_1 and μ_2.
- n_1 samples from μ_1, n_2 samples from μ_2.
- Estimate $\hat{T}$ with closed form solution.

Theorem ([Flamary et al., 2019])

Let μ_1 and μ_2 be sub-Gaussian distributions with expectations m_1, m_2 and positive-definite covariance operators Σ_1, Σ_2 respectively with eigenvalues in $[c, C]$ for some fixed absolute constants $0 < c \leq C < \infty$. We also assume that $n_j \geq C\mathbf{r}(\Sigma_j), \quad j = 1, 2,$ for some sufficiently large numerical constant $C > 0$.

Then, for any $t > 0$, we have with probability at least $1 - e^{-t} - \frac{1}{n_1}$,

$$\mathop{\mathbb{E}}_{s\sim\mu_1} \|T(x) - \hat{T}(x)\| \leq C' \left(\sqrt{\frac{\mathbf{r}(\Sigma_1)}{n_1}} \vee \sqrt{\frac{\mathbf{r}(\Sigma_2)}{n_2}} \vee \sqrt{\frac{t}{n_1 \wedge n_2}} \vee \frac{t}{n_1 \wedge n_2} \right) \sqrt{\mathbf{r}(\Sigma_1)},$$

where $C' > 0$ is a constant independent of $n_1, n_2, \mathbf{r}(\Sigma_1), \mathbf{r}(\Sigma_2)$ and $\mathbf{r}(B) = \frac{\mathrm{tr}(B)}{\lambda_{\max}(B)}$.

OTDA Generalization bound

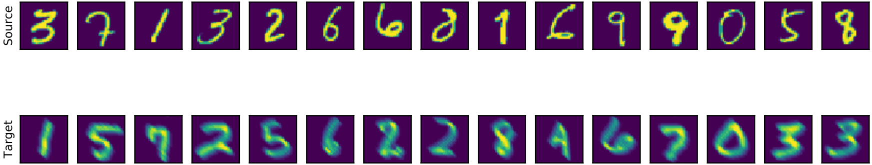
Estimator in source domain
Let $\mathcal{H}_K$ be a reproducing kernel Hilbert space (RKHS) associated with a symmetric nonnegatively definite kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ We consider the following empirical risk minimization estimator:

$$\hat{f}_{n_l} := \operatorname{argmin}_{\|f\|_{\mathcal{H}_K} \leq 1} \frac{1}{n_l} \sum_{i=1}^{n_l} l(Y_i^l, f(X_i^l)). \tag{13}$$

where we assume that the eigenvalues of the integral operator T_K of $\mathcal{H}_K$ decrease with $\lambda_k \asymp k^{-2\beta}$ for some $\beta > 1/2$ (see [Mendelson, 2002]).

OTDA generalization bound
If $R_s(f_*^s) = R_t(f_*^t)$ and $\hat{T}$ is the linear monge mapping estimator, under the assumptions of OTDA, we get with probability at least $1 - e^{-t} - \frac{1}{n_1}$,

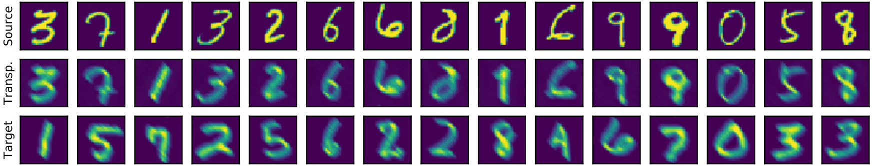
$$R_t(\hat{f}_{n_l} \circ \hat{T}^{-1}) - R_t(f_*^t) \lesssim n_l^{-2\beta/(1+2\beta)} + \frac{t}{n_l} + M_f M_L \left(\sqrt{\frac{\mathbf{r}(\Sigma_2)}{n_2}} \vee \sqrt{\frac{\mathbf{r}(\Sigma_1)}{n_1}} \vee \sqrt{\frac{t}{n_1 \wedge n_2}} \vee \frac{t}{n_1 \wedge n_2} \right) \sqrt{\mathbf{r}(\Sigma_1)}.$$



Numerical experiments

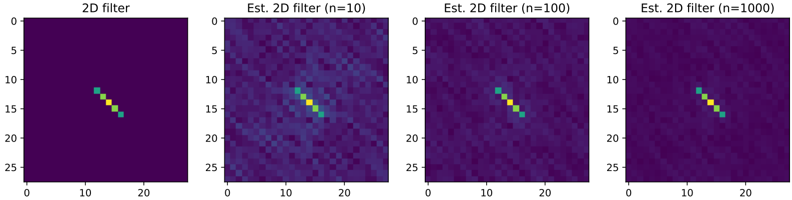
- Split MNIST dataset in two non-overlapping empirical distributions.
- Apply linear motion blur to the target distribution.
- Estimate mapping and transport source samples.
- Convolutional Monge Mapping for important speedup (FFT).

Linear Monge mapping on images



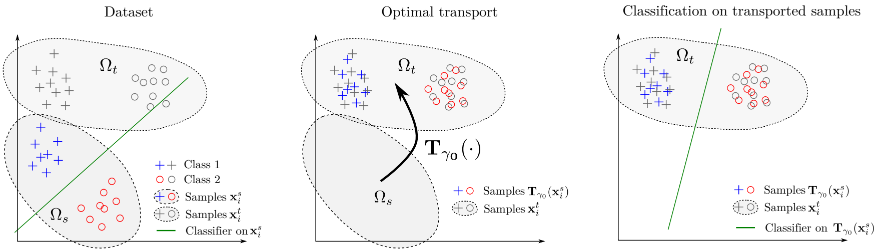
Numerical experiments

- Split MNIST dataset in two non-overlapping empirical distributions.
- Apply linear motion blur to the target distribution.
- Estimate mapping and transport source samples.
- Convolutional Monge Mapping for important speedup (FFT).



Numerical experiments

- Split MNIST dataset in two non-overlapping empirical distributions.
- Apply linear motion blur to the target distribution.
- Estimate mapping and transport source samples.
- Convolutional Monge Mapping for important speedup (FFT).



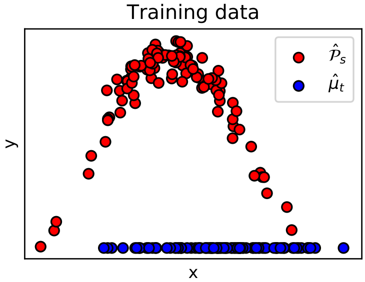
Discussion

- Works very well in practice for large class of transformation [Courty et al., 2016].
- Can use estimated mapping [Perrot et al., 2016, Seguy et al., 2017].
- Nice generalization bound for linear Monge mappings [Flamary et al., 2019].

But

- Model transformation only in the feature space.
- Requires the same class proportion between domains $\mathcal{P}_s(y) \approx \mathcal{P}_t(y)$ (no label shift) [Tuia et al., 2015].
- We estimate a $T : \mathbb{R}^d \to \mathbb{R}^d$ mapping for training a classifier $f : \mathbb{R}^d \to \mathbb{R}$.

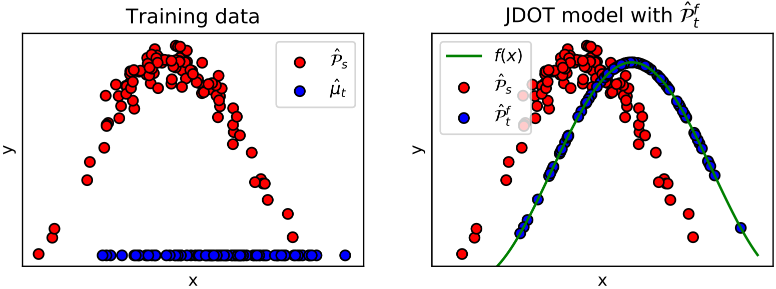
Joint distribution OT for domain adaptation (JDOT)



Main idea

- Objectives : allow changes in the label space i.e. $\mathcal{P}_s(y) \neq \mathcal{P}_t(y)$, learn directly a target predictor f.
- Joint feature/labels distribution $\hat{\mathcal{P}}_s(x^s, y^s)$ in source, only marginal feature distribution $\hat{\mu}_t = \hat{\mathcal{P}}_t(x^t)$ in target.
- Wasserstein needs the two distributions $\hat{\mathcal{P}}_s(x^s, y^s)$ and $\hat{\mathcal{P}}_t(x^t, y^t)$
- Use a proxy distribution: $\hat{\mathcal{P}_t}^f = \hat{\mathcal{P}_t}^f(x^t, f(x^t)) = \frac{1}{N_t}\sum_{i=1}^{N_t} \delta_{\mathbf{x}_i^t, f(\mathbf{x}_i^t)}$

Joint Distribution Optimal Transport for DA (JDOT)



Learning with JDOT [Courty et al., 2017]

$$\min_f \quad \left\{ W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}}_t{}^f) = \inf_{\gamma \in \Pi} \sum_{ij} \mathcal{D}(\mathbf{x}_i^s, \mathbf{y}_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t)) \gamma_{ij} \right\} \tag{14}$$

- $\hat{\mathcal{P}}_t{}^f = \frac{1}{N_t} \sum_{i=1}^{N_t} \delta_{\mathbf{x}_i^t, f(\mathbf{x}_i^t)}$ is the proxy joint feature/label distribution.
- $\mathcal{D}(\mathbf{x}_i^s, \mathbf{y}_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t)) = \alpha \|\mathbf{x}_i^s - \mathbf{x}_j^t\|^2 + \mathcal{L}(\mathbf{y}_i^s, f(\mathbf{x}_j^t))$ with $\alpha > 0$.
- We search for the predictor f that better align the joint distributions.
- OT matrix does the label propagation (no mapping).
- JDOT can be seen as minimizing a generalization bound.

Generalization bound (1)

We define a novel version of the Probabilistic Lipschitzness:

Probabilistic Lipschitzness [Urner et al., 2011, Ben-David et al., 2012]
Let $\phi : \mathbb{R} \to [0, 1]$. A labeling function $f : \Omega \to \mathbb{R}$ is ϕ-Lipschitz with respect to a distribution P over Ω if for all $\lambda > 0$

$$Pr_{x \sim P} \left[\exists y : [|f(x) - f(y)| > \lambda d(x, y)] \right] \leq \phi(\lambda).$$

Probabilistic Transfer Lipschitzness
Let μ_s and μ_t be respectively the source and target distributions. Let $\phi : \mathbb{R} \to [0, 1]$. A labeling function $f : \Omega \to \mathbb{R}$ and a joint distribution $\boldsymbol{\gamma}(\mu_s, \mu_t)$ over μ_s and μ_t are ϕ-Lipschitz transferable if for all $\lambda > 0$:

$$Pr_{(\mathbf{x}_1, \mathbf{x}_2) \sim \boldsymbol{\gamma}(\mu_s, \mu_t)} \left[|f(\mathbf{x}_1) - f(\mathbf{x}_2)| > \lambda d(\mathbf{x}_1, \mathbf{x}_2) \right] \leq \phi(\lambda).$$

Generalization bound (2)

Theorem 1

Let f be any labeling function of $\in \mathcal{H}$. Let $\boldsymbol{\gamma}^* = \operatorname{argmin}_{\boldsymbol{\gamma} \in \Pi(\mathcal{P}_s, \mathcal{P}_t^f)} \int_{(\Omega \times \mathcal{C})^2} \alpha d(\mathbf{x}_s, \mathbf{x}_t) + \mathcal{L}(y_s, y_t) d\boldsymbol{\gamma}(\mathbf{x}_s, y_s; \mathbf{x}_t, y_t)$ 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. $\boldsymbol{\gamma}^*$ and that minimizes the joint error $R_s(f^*) + R_t(f^*)$ w.r.t all PTL functions compatible with $\boldsymbol{\gamma}^*$. 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:

$$R_t(f) \leq W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}_t^f}) + \sqrt{\frac{2}{c'} \log(\frac{2}{\delta})} \left(\frac{1}{\sqrt{n_s}} + \frac{1}{\sqrt{n_t}} \right)$$
$$+ R_s(f^*) + R_t(f^*) + kM\phi(\lambda).$$

- 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

$$\min_{f\in\mathcal{H},\boldsymbol{\gamma}\in\mathcal{U}} \quad \sum_{i,j}\boldsymbol{\gamma}_{i,j}\left(\alpha d(\mathbf{x}_i^s,\mathbf{x}_j^t)+\mathcal{L}(y_i^s,f(\mathbf{x}_j^t))\right)+\lambda\Omega(f) \tag{15}$$

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

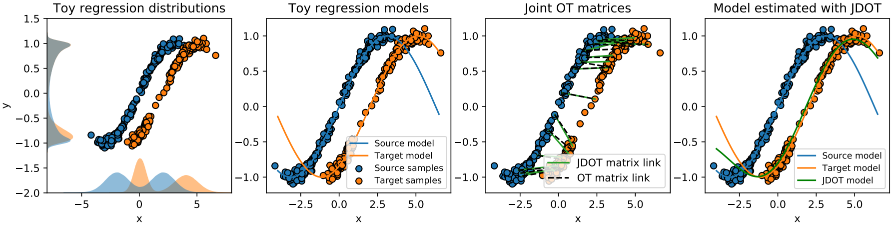
- Classical OT problem.
- Solved by network simplex.
- Regularized OT can be used (add a term to problem (15))

f update for a fixed γ

$$\min_{f\in\mathcal{H}} \quad \sum_{i,j}\boldsymbol{\gamma}_{i,j}\mathcal{L}(y_i^s,f(\mathbf{x}_j^t))+\lambda\Omega(f) \tag{16}$$

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

Regression with JDOT

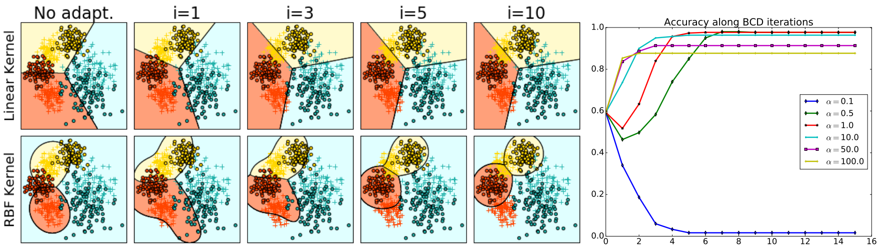


Least square regression with quadratic regularization

For a fixed γ the optimization problem is equivalent to

$$\min_{f \in \mathcal{H}} \quad \sum_j \frac{1}{n_t} \|\hat{y}_j - f(\mathbf{x}_j^t)\|^2 + \lambda \|f\|^2 \tag{17}$$

- $\hat{y}_j = n_t \sum_j \boldsymbol{\gamma}_{i,j} y_i^s$ is a weighted average of the source target values.
- Can use any solver (linear, kernel ridge, neural network).



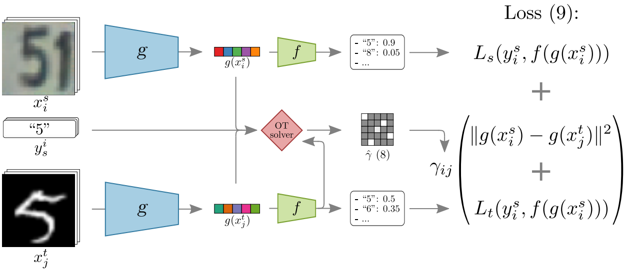
Multiclass classification with Hinge loss
For a fixed γ the optimization problem is equivalent to

$$\min_{f_k \in \mathcal{H}} \quad \sum_{j,k} \hat{P}_{j,k}\mathcal{L}(1, f_k(\mathbf{x}_j^t)) + (1 - \hat{P}_{j,k})\mathcal{L}(-1, f_k(\mathbf{x}_j^t)) + \lambda \sum_k \|f_k\|^2 \qquad (18)$$

- $\hat{\mathbf{P}}$ 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

$$Y^s_{i,k} = \begin{cases} 1 & \text{if } y_i^s = k \\ -1 & \text{else} \end{cases}, \quad P^s_{i,k} = \begin{cases} 1 & \text{if } y_i^s = k \\ 0 & \text{else} \end{cases}$$

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

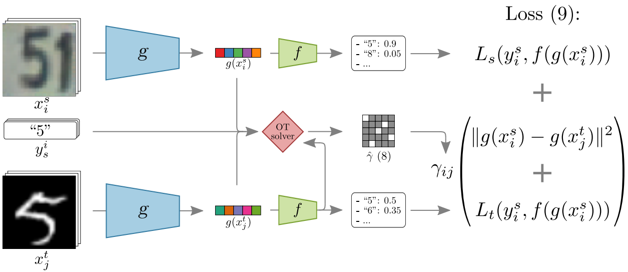


$$\min_{\gamma \in \Pi, f, g} \quad \frac{1}{n^s} \sum_i L_s\left(y_i^s, f(g(x_i^s))\right) + \sum_{i,j} \gamma_{ij} \left(\alpha \|g(x_i^s) - g(x_j^t)\|^2 + \lambda_t \mathcal{L}\left(y_i^s, f(g(x_j^t))\right)\right). \tag{19}$$

DeepJDOT [Damodaran et al., 2018]

- Learn simultaneously the embedding g and the classifier f.
- JDOT performed in the joint embedding/label space.

DeepJDOT



$$\min_{f,g} \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\mathcal{L}\left(y_i^s, f(g(x_i^s)\right) + \min_{\gamma\in\Pi}\sum_{i,j}^{m}\gamma_{ij}\left(\alpha\|g(x_i^s)-g(x_j^t)\|^2 + \lambda_t\mathcal{L}\left(y_i^s, f(g(x_j^t))\right)\right)\right] \tag{19}$$

DeepJDOT [Damodaran et al., 2018]

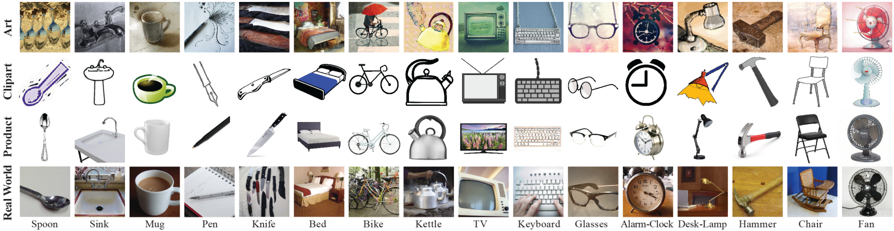
- Learn simultaneously the embedding g and the classifier f.
- JDOT performed in the joint embedding/label space.
- Use minibatch to estimate OT and update g, f at each iterations.
- Scales to large datasets and estimate a representation for both domains.

DeepJDOT in action

DeepJDOT [Damodaran et al., 2018]

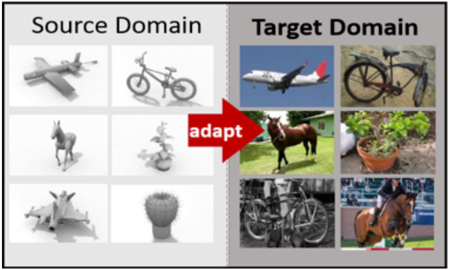
- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are important.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).

DeepJDOT in action



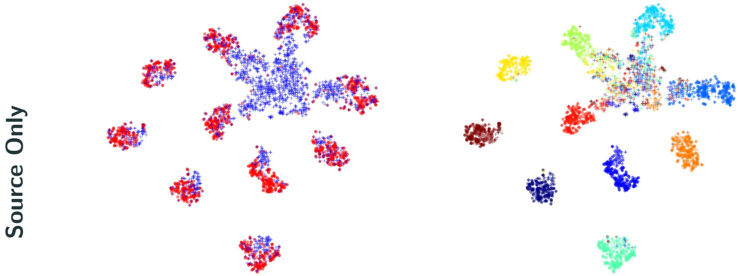
DeepJDOT [Damodaran et al., 2018]

- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are important.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).



DeepJDOT [Damodaran et al., 2018]

- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are important.
- TSNE projections of embeddings (MNIST→MNIST-M).



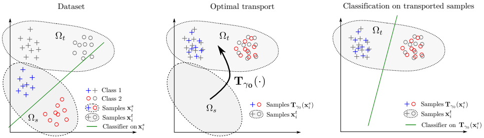
DeepJDOT [Damodaran et al., 2018]

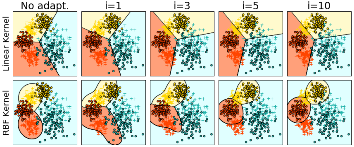
- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are important.
- TSNE projections of embeddings (MNIST→MNIST-M).

DeepJDOT [Damodaran et al., 2018]

- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are important.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).

Conclusion





Optimal transport for DA

- Model transformation of the features.
- Joint distribution preserved.
- Mapping between distributions.
- Learn classifier on the transported samples.
- Generalization bound when mapping estimation bounded.

Joint distribution OT for DA

- Model transformation of the joint distribution.
- General framework for DA.
- Estimate directly the predictor.
- Theoretical justification with generalization bound.
- Can also estimate feature extraction.

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.

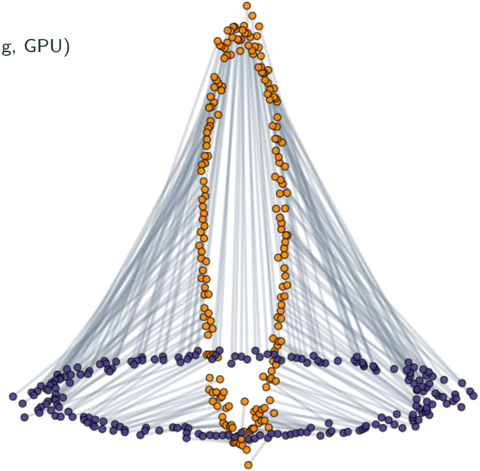
Python code for JDOT on GitHub:

`https://github.com/rflamary/JDOT`

Papers available on my website:

`https://remi.flamary.com/`

Post docs available in: Nice (France)



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.

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.

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

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., 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. (2016).
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.

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

References iii

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

Flamary, R., Lounici, K., and Ferrari, A. (2019).
Concentration bounds for linear monge mapping estimation and optimal transport domain adaptation.
arXiv preprint arXiv:1905.10155.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016).
Domain-adversarial training of neural networks.
The Journal of Machine Learning Research, 17(1):2096–2030.

Gayraud, N. T., Rakotomamonjy, A., and Clerc, M. (2017).
Optimal transport applied to transfer learning for p300 detection.
In *BCI 2017-7th Graz Brain-Computer Interface Conference*, page 6.

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

Hütter, J.-C. and Rigollet, P. (2019).
Minimax rates of estimation for smooth optimal transport maps.
arXiv preprint arXiv:1905.05828.

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

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

Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2017).
Deep transfer learning with joint adaptation networks.
In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org.

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

Mendelson, S. (2002).
Geometric parameters of kernel machines.
In Kivinen, J. and Sloan, R. H., editors, *Computational Learning Theory*, pages 29–43, Berlin, Heidelberg. Springer Berlin Heidelberg.

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

Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., and Saenko, K. (2017).
Visda: The visual domain adaptation challenge.
arXiv preprint arXiv:1710.06924.

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.

Rivet, B., Souloumiac, A., Attina, V., and Gibert, G. (2009).
xdawn algorithm to enhance evoked potentials: application to brain–computer interface.
IEEE Transactions on Biomedical Engineering, 56(8):2035–2043.

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.

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

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.

Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., and Guibas, L. (2015).
Convolutional wasserstein distances: Efficient optimal transportation on geometric domains.
ACM Transactions on Graphics (TOG), 34(4):66.

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.

Tripuraneni, N., Jordan, M. I., and Jin, C. (2020).
On the theory of transfer learning: The importance of task diversity.
arXiv preprint arXiv:2006.11650.

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*.

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 *(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR)*.