Domain Adaptation from shallow to deep learning

Rémi Flamary - CMAP, École Polytechnique

June 26 2022

16ème École d'été de Peyresq en traitement du signal et des images

URL : `https://tinyurl.com/tuto-da`

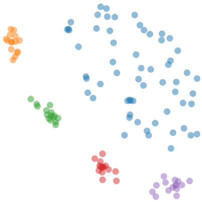
Table of content

Domain adaptation problem and generalization

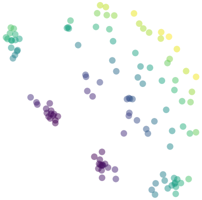
Supervised learning objective

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_i^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix}$$

Classification



Regression



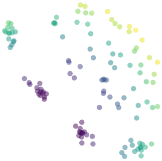
Objective

- Training dataset : $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with observations $\mathbf{x}_i \in \mathbb{R}^d$ and labels $y_i \in \mathcal{Y}$.
- Train a function $f(\cdot) : \mathbb{R}^d \to \mathcal{Y}$ on the dataset.

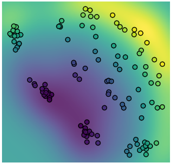
Data distribution

- $\mathcal{P}$ is the true joint feature/label distribution of the data.
- Data $\mathbf{x}_i, y_i \sim \mathcal{P}$ is supposed to be drawn I.I.D from $\mathcal{P}$
- $\widehat{\mathcal{P}} = \frac{1}{n}\sum_i \delta_{\mathbf{x}_i, y_i}$ is the training empirical distribution.
- $\mathcal{P}_\mathcal{X}$ and $\mathcal{P}_\mathcal{Y}$ are respectively the feature ($\mathbf{x}$) and labels (y) marginals of $\mathcal{P}$.

Regression

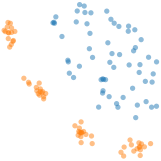


$$\Longrightarrow$$

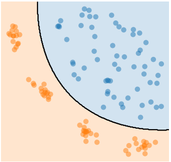


$$\{\mathbf{x}_i, y_i\}_{i=1}^n \quad \Rightarrow \quad f : \mathbb{R}^d \to \mathbb{R}$$

Binary classifiation

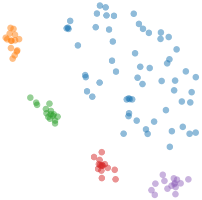


$$\Longrightarrow$$

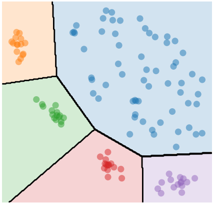


$$\{\mathbf{x}_i, y_i\}_{i=1}^n \quad \Rightarrow \quad f : \mathbb{R}^d \to \{-1, 1\}$$

Multiclass classification

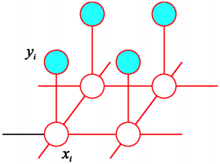


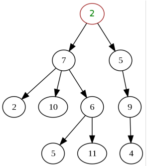
$\Longrightarrow$

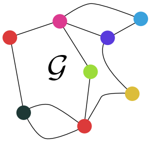


$$\{\mathbf{x}_i, y_i\}_{i=1}^n \quad \Rightarrow \quad f : \mathbb{R}^d \to \{1, \ldots, K\}, \quad \text{with} \quad f(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\, f_k(\mathbf{x})$$

Structured prediction

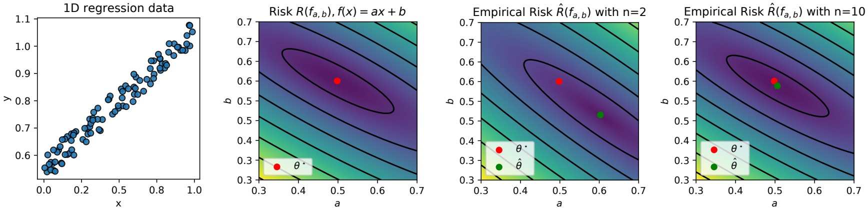






$$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n \quad \Rightarrow \quad f : \mathcal{X} \to \mathcal{Y}, \quad \text{with} \quad f(\mathbf{x}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}}\, \tilde{f}(\mathbf{x}, \mathbf{y})$$

Risk and Empirical Risk



- We define the **true risk** or expected loss $\mathcal{R}$ for a predictor f wrt distribution $\mathcal{P}$ as

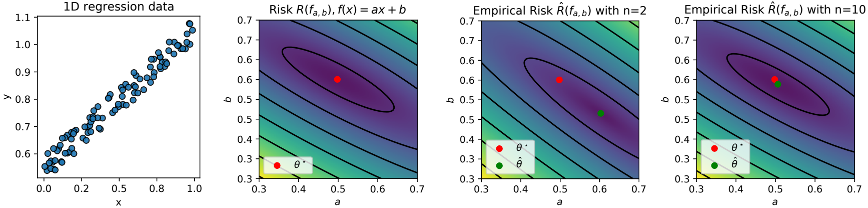
$$\mathcal{R}(f) = \mathcal{R}_{\mathcal{P}}(f) = E_{\mathbf{x},y\sim\mathcal{P}}[L(y, f(\mathbf{x}))], \tag{1}$$

where the loss $L(y, \hat{y})$ measures a discrepancy between the actual and the predicted label.

- The **Empirical risk** for predictor f is the risk using the empirical distribution $\widehat{\mathcal{P}}$:

$$\widehat{\mathcal{R}}(f) = \mathcal{R}_{\widehat{\mathcal{P}}}(f) = E_{\mathbf{x},y\sim\widehat{\mathcal{P}}}[L(y, f(\mathbf{x}))] = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)), \tag{2}$$

Empirical risk minimization and generalization



- **Empirical risk minimization** :

$$\min_{f \in \mathcal{H}} \quad \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)), \tag{3}$$

- Classical generalization bounds can be expressed for a given predictor $f \in \mathcal{H}$ as

$$\mathcal{R}(f) \leq \widehat{\mathcal{R}}(f) + \mathcal{O}\left(\frac{C(\mathcal{H})}{\sqrt{n}}\right) \tag{4}$$

  where $C(\mathcal{H})$ is a measure of complexity of the hypothesis space $\mathcal{H}$.

- Bound above have motivated the use of regularization or limited complexity (layer/parameters) on small datasets.

Divergences
Let $\mathcal{P}^s$ and $\mathcal{P}^t$ be probability distributions on $\mathcal{X}$ of density $P^s(x)$ and $P^t(x)$ respectively. A divergence D has the following properties:

- $D(\mathcal{P}^s, \mathcal{P}^t) \geq 0$, $\forall \mathcal{P}^s, \mathcal{P}^t$
- $D(\mathcal{P}^s, \mathcal{P}^t) = 0$ if and only if $\mathcal{P}^s = \mathcal{P}^t$

Classical divergences

- **Kullback-Leibler**

$$KL(\mathcal{P}^s|\mathcal{P}^t) = \int_{\mathcal{X}} P^s(\mathbf{x}) \log\left(\frac{P^s(\mathbf{x})}{P^t(\mathbf{x})}\right) d\mathbf{x} \tag{5}$$

- **Total Variation**

$$TV(\mathcal{P}^s, \mathcal{P}^t) = \int_{\mathcal{X}} |P^s(\mathbf{x}) - P^t(\mathbf{x})| d\mathbf{x} \tag{6}$$

Both divergences do not work well on discrete distributions with non overlapping support.

Maximum Mean Discrepancy (MMD)

Principle

- Project $\mathbf{x}$ in a Reproducing Kernel Hilbert Space $\mathcal{H}$ (RKHS) with ϕ.
- The MMD can be expressed as the distance between the means in the RKHS Hilbert space as

$$MMD^2(\mathcal{P}^s, \mathcal{P}^t) = \|E_{\mathbf{x}\sim\mathcal{P}^s}[\phi(\mathbf{x})] - E_{\mathbf{x}\sim\mathcal{P}^t}[\phi(\mathbf{x})]\|^2_{\mathcal{H}} \tag{7}$$

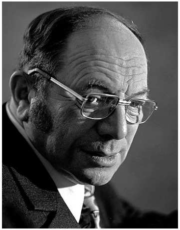
- In the RKHS the kernel can be expressed as $k(\mathbf{x}, \mathbf{x}') =< \phi(\mathbf{x}), \phi(\mathbf{x}') >$ and the MMD can be reformulated as:

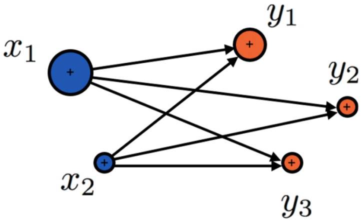
$$MMD^2(\mathcal{P}^s, \mathcal{P}^t) = E_{\mathbf{x},\mathbf{x}'\sim\mathcal{P}^s}[k(\mathbf{x}, \mathbf{x}')] + E_{\mathbf{x},\mathbf{x}'\sim\mathcal{P}^t}[k(\mathbf{x}, \mathbf{x}')] - 2E_{\mathbf{x}\sim\mathcal{P}^s,\mathbf{x}'\sim\mathcal{P}^t}[k(\mathbf{x}, \mathbf{x}')] \tag{8}$$

- The unbiased estimator of MMD between two empirical distributions is

$$\begin{aligned} MMD^2(\hat{\mathcal{P}}^s, \hat{\mathcal{P}}^t) =& \frac{1}{n_s(n_s-1)} \sum_{i=1,j=1}^{n^s,n^s} k(\mathbf{x}_i^s, \mathbf{x}_j^s) + \frac{1}{n_t(n_t-1)} \sum_{j=1}^{n^t} k(\mathbf{x}_j^t, \mathbf{x}_j^t) \\ & - \frac{2}{n_s n_t} \sum_{i=1,j=1}^{n^s,n^t} k(\mathbf{x}_i^s, \mathbf{x}_j^t) \end{aligned} \tag{9}$$

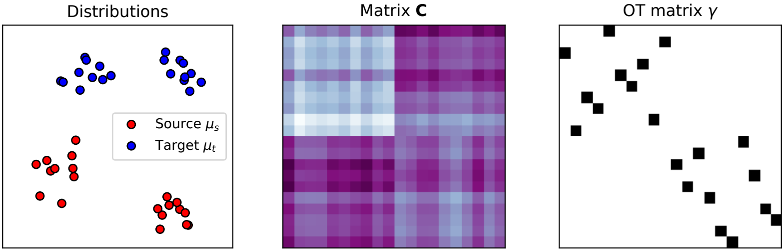
Optimal transport





- Problem introduced by Gaspard Monge in his memoire [Monge, 1781].
- How to move mass while minimizing a cost (mass + cost)
- Monge formulation seeks for a mapping between two mass distribution.
- Reformulated by Leonid Kantorovich (1912–1986), Economy nobelist in 1975
- Focus on where the mass goes, allow splitting [Kantorovich, 1942].
- Applications originally for resource allocation problems

Optimal transport between discrete distributions



Kantorovitch formulation : OT Linear Program

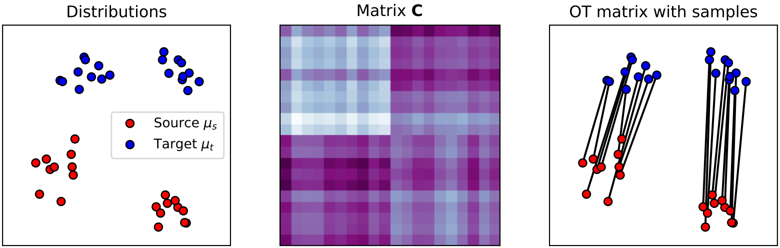
When $\color{red}{\mathcal{P}^s} = \sum_{i=1}^{n_s} \color{red}{a_i \delta_{\mathbf{x}_i^s}}$ and $\color{blue}{\mathcal{P}^t} = \sum_{i=1}^{n_t} \color{blue}{b_i \delta_{\mathbf{x}_i^t}}$

$$\min_{\mathbf{T}\in\Pi(\mathcal{P}^s,\mathcal{P}^t)} \quad \left\{ \langle \mathbf{T}, \mathbf{C} \rangle_F = \sum_{i,j} T_{i,j} c_{i,j} \right\}$$

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

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

- Linear program with $n_s n_t$ variables and $n_s + n_t$ constraints. Solving the OT problem with network simplex is $O(n^3 \log(n))$ for $n = n_s = n_t$.
- Entropic regularization solved efficiently with Sinkhorn [Cuturi, 2013].



Kantorovitch formulation : OT Linear Program

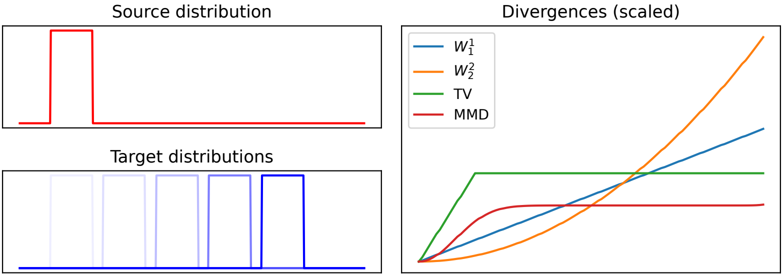
When $\mathcal{P}^s = \sum_{i=1}^{n_s} a_i \delta_{\mathbf{x}_i^s}$ and $\mathcal{P}^t = \sum_{i=1}^{n_t} b_i \delta_{\mathbf{x}_i^t}$

$$\min_{\mathbf{T} \in \Pi(\mathcal{P}^s, \mathcal{P}^t)} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle_F = \sum_{i,j} T_{i,j} c_{i,j} \right\}$$

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

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

- Linear program with $n_s n_t$ variables and $n_s + n_t$ constraints. Solving the OT problem with network simplex is $O(n^3 \log(n))$ for $n = n_s = n_t$.
- Entropic regularization solved efficiently with Sinkhorn [Cuturi, 2013].



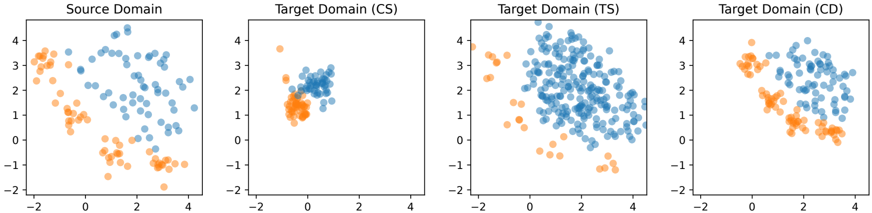
Wasserstein distance

$$W_p^p(\mathcal{P}^s, \mathcal{P}^t) = \min_{\mathbf{T}\in\mathcal{P}} \int_{\Omega_s\times\Omega_t} \|\mathbf{x}-\mathbf{y}\|^p \mathbf{T}(\mathbf{x},\mathbf{y}) d\mathbf{x}d\mathbf{y} = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim\mathbf{T}}[\|\mathbf{x}-\mathbf{y}\|^p] \tag{10}$$

In this case we have $c(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^p$

- A.K.A. Earth Mover's Distance when $p = 1$ (W_1^1) [Rubner et al., 2000].
- Do not need the distribution to have overlapping support.
- Works for continuous and discrete distributions (histograms, empirical).

Data shift



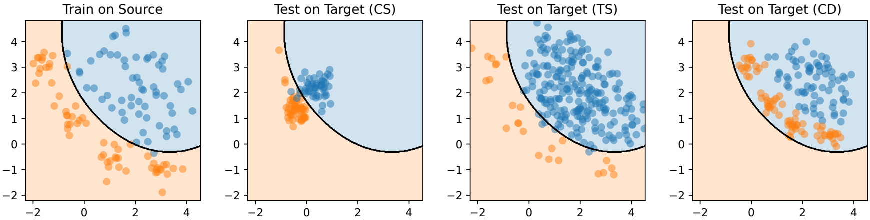
Shift happens...

- Data shift : $\mathcal{P}^s \neq \mathcal{P}^t$
- $\mathcal{P}^s$ is the training distribution (Source domain)
- $\mathcal{P}^t$ is the test distribution (Target domain)
- A classifier learned on $\mathcal{P}^s$ might fail on $\mathcal{P}^t$.

... but Domain Adaptation (DA) is here for you

- Aim at learning a function f that works on $\mathcal{P}^t$ using data samples from $\mathcal{P}^s$.
- Unsupervised DA suppose that we have samples $\mathbf{x}^t$ from $\mathcal{P}^t$ but no labels.

Data shift



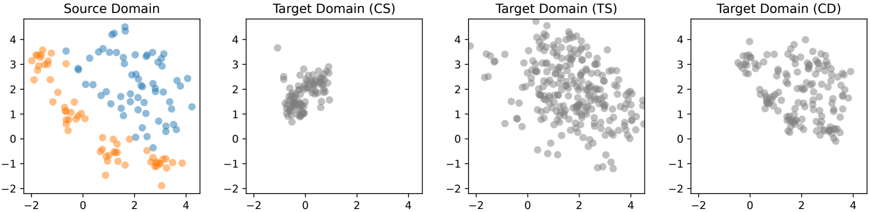
Shift happens...

- Data shift : $\mathcal{P}^s \neq \mathcal{P}^t$
- $\mathcal{P}^s$ is the training distribution (Source domain)
- $\mathcal{P}^t$ is the test distribution (Target domain)
- A classifier learned on $\mathcal{P}^s$ might fail on $\mathcal{P}^t$.

... but Domain Adaptation (DA) is here for you

- Aim at learning a function f that works on $\mathcal{P}^t$ using data samples from $\mathcal{P}^s$.
- Unsupervised DA suppose that we have samples $\mathbf{x}^t$ from $\mathcal{P}^t$ but no labels.

Data shift



Shift happens...

- Data shift : $\mathcal{P}^s \neq \mathcal{P}^t$
- $\mathcal{P}^s$ is the training distribution (Source domain)
- $\mathcal{P}^t$ is the test distribution (Target domain)
- A classifier learned on $\mathcal{P}^s$ might fail on $\mathcal{P}^t$.

... but Domain Adaptation (DA) is here for you

- Aim at learning a function f that works on $\mathcal{P}^t$ using data samples from $\mathcal{P}^s$.
- Unsupervised DA suppose that we have samples $\mathbf{x}^t$ from $\mathcal{P}^t$ but no labels.

Data shift

Shift happens...

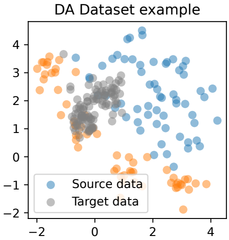
- Data shift : $\mathcal{P}^s \neq \mathcal{P}^t$
- $\mathcal{P}^s$ is the training distribution (Source domain)
- $\mathcal{P}^t$ is the test distribution (Target domain)
- A classifier learned on $\mathcal{P}^s$ might fail on $\mathcal{P}^t$.

... but Domain Adaptation (DA) is here for you

- Aim at learning a function f that works on $\mathcal{P}^t$ using data samples from $\mathcal{P}^s$.
- Unsupervised DA suppose that we have samples $\mathbf{x}^t$ from $\mathcal{P}^t$ but no labels.

Domain Adaptation Problem

$$\mathbf{X}^s = \begin{bmatrix} \mathbf{x}_1^{s\top} \\ \mathbf{x}_2^{s\top} \\ \vdots \\ \mathbf{x}_i^{s\top} \\ \vdots \\ \mathbf{x}_{n_s}^{s\top} \end{bmatrix}, \quad \mathbf{y}^s = \begin{bmatrix} y_1^s \\ y_2^s \\ \vdots \\ y_i^s \\ \vdots \\ y_{n_s}^s, \end{bmatrix}, \quad \mathbf{X}^t = \begin{bmatrix} \mathbf{x}_1^{t\top} \\ \vdots \\ \mathbf{x}_i^{t\top} \\ \vdots \\ \mathbf{x}_{n_t}^{t\top} \end{bmatrix}$$



Data and distributions

- Source dataset : $\{\mathbf{x}_i^s, y_i^s\}_{i=1}^{n_s}$ with $\mathbf{x}_i^s, y_i^s \sim \mathcal{P}^s$, and $\widehat{\mathcal{P}}^s = \frac{1}{n_s}\sum_{i=1}^{n_s}\delta_{\mathbf{x}_i^s, y_i^s}$.
- Target dataset : $\{\mathbf{x}_j^t\}_{j=1}^{n_t}$ with $\mathbf{x}_j^t \sim \mathcal{P}_{\mathcal{X}}^t$, and $\widehat{\mathcal{P}}_{\mathcal{X}}^t = \frac{1}{n_t}\sum_{j=1}^{n_t}\delta_{\mathbf{x}_j^t}$

Objective

- Train a function $f(\cdot) : \mathbb{R}^d \to \mathcal{Y}$ on the datasets that performs well on $\mathcal{P}^t$.
- The performance when training on source depends on how similar $\mathcal{P}^s$ and $\mathcal{P}^t$ are.
- The data shift can be compensated for some special cases of shifts.

How to compensate for shift ?

- Numerous DA approaches propose to model the shift and compensate for it.
- There exist several types of shifts that are more or less complex to handle.

Notations

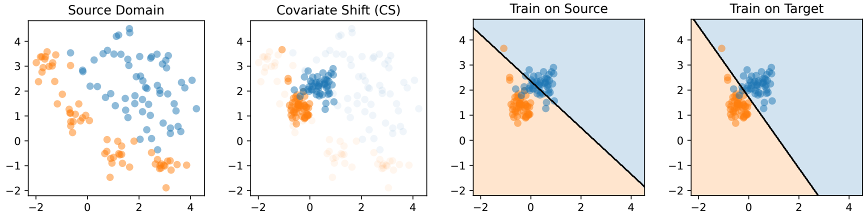
- We will us $P(\mathbf{x}, y)$ as the probability density of distribution $\mathcal{P}$ (P^s for $\mathcal{P}^s$, ...).
- The Bayes theorem gives us

$$P(\mathbf{x}, y) = P(\mathbf{x}|y)P_{\mathcal{Y}}(y) = P(y|\mathbf{x})P_{\mathcal{X}}(\mathbf{x}) \tag{11}$$

Types of data shift and their intuition [Moreno-Torres et al., 2012]

- **Covariate shift**, $P^s_{\mathcal{X}}(\mathbf{x}) \neq P^t_{\mathcal{X}}(\mathbf{x})$, $P^s(y|\mathbf{x}) = P^t(y|\mathbf{x})$
- **Target shift**, $P^s_{\mathcal{Y}}(y) \neq P^t_{\mathcal{Y}}(y)$, $P^s(\mathbf{x}|y) = P^t(\mathbf{x}|y)$
- **Concept drift**, $P^s(y|\mathbf{x}) \neq P^t(y|\mathbf{x})$ or $P^s(\mathbf{x}|y) \neq P^t(\mathbf{x}|y)$
- **Sample-selection bias**, $P^s(\mathbf{x}, y) \neq P^t(\mathbf{x}, y)P(s|\mathbf{x}, y)$

Covariate Shift (CS)



Principle

- Conditionals probabilities : $P^s(y|\mathbf{x}) = P^t(y|\mathbf{x})$
- Feature marginals are different : $P^s_{\mathcal{X}}(\mathbf{x}) \neq P^t_{\mathcal{X}}(\mathbf{x})$

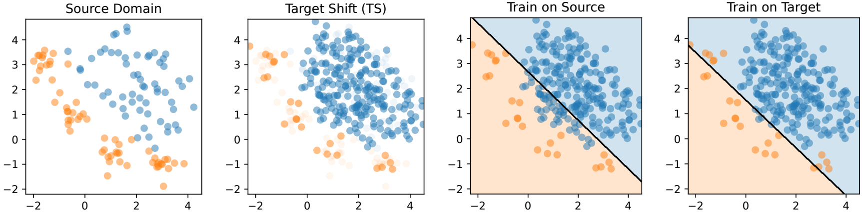
Compensating for the shift

- Covariate shift can be compensated using sample weighting [Shimodaira, 2000].
- The target risk can be expressed as an expectation on the source distribution

$$\mathcal{R}_{\mathcal{P}^t}(f) = E_{\mathbf{x},y\sim\mathcal{P}^s}\left[\frac{P^t_{\mathcal{X}}(\mathbf{x})}{P^s_{\mathcal{X}}(\mathbf{x})} L(y, f(\mathbf{x}))\right] \tag{12}$$

So if the ratio $w(\mathbf{x}) = \frac{P^t_{\mathcal{X}}(\mathbf{x})}{P^s_{\mathcal{X}}(\mathbf{x})}$ is estimated one can learn from an empirical source distribution (careful that supp$(\mathcal{P}^s_{\mathcal{X}}) \subseteq$ supp$(\mathcal{P}^t_{\mathcal{X}})$ or else division by 0).

Target Shift (TS)



Principle (a.k.a prior shift or label shift)

- Conditionals probabilities : $P^s(\mathbf{x}|y) = P^t(\mathbf{x}|y)$
- Label marginals are different : $P^s_{\mathcal{Y}}(y) \neq P^t_{\mathcal{Y}}(y)$

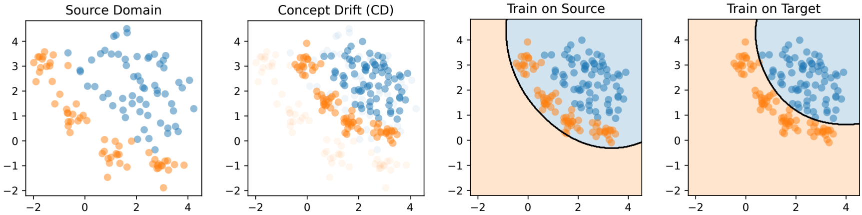
Compensating for the shift

- Target shift can be compensated using sample weighting [Shimodaira, 2000].
- The target risk can be expressed as ane expectation on the source distribution

$$\mathcal{R}_{\mathcal{P}^t}(f) = E_{\mathbf{x},y\sim\mathcal{P}^s}\left[\frac{P^t_{\mathcal{Y}}(y)}{P^s_{\mathcal{Y}}(y)} L(y, f(\mathbf{x}))\right] \quad (13)$$

So if the ratio $w(y) = \frac{P^t_{\mathcal{Y}}(y)}{P^s_{\mathcal{Y}}(y)}$ is known it can be used to reweight samples ($P^t_{\mathcal{Y}}(y)$ cannot be estimated from target data).

Concept Drift (CD)



Principle (a.k.a Conditional shift)

- Conditionals probabilities are different : $P^s(\mathbf{x}|y) \neq P^t(\mathbf{x}|y)$ or $P^s(y|\mathbf{x}) \neq P^t(y|\mathbf{x})$

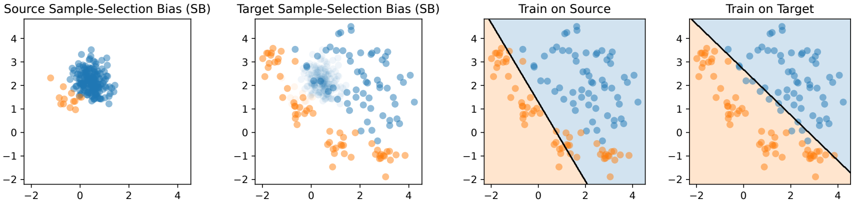
Compensating for the shift

- Hardest shift because requires a model for the transformation between the conditional probabilities (can model sensor drift).
- In the special case where there exists a mapping m in the feature space ($P^s(y|m(\mathbf{x})) = P^t(y|\mathbf{x})$) then

$$\mathcal{R}_{\mathcal{P}^t}(f) = E_{\mathbf{x},y\sim\mathcal{P}^s}\left[L(y, f(m(\mathbf{x})))\right] \tag{14}$$

- The marginals $\mathcal{P}_\mathcal{Y}$ or $\mathcal{P}_\mathcal{X}$ are usually the same but when they are not the problem is known as **generalized target shift**.

Sample-Selection Bias (SSB)



Principle

- The exists a multiplicative sampling bias : $P^s(\mathbf{x}, y) = S(\mathbf{x}, y)P^t(\mathbf{x}, y)$

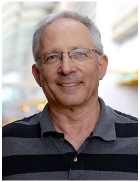
Compensating for the shift

- Requires a good estimation of $S(\mathbf{x}, y)$ to be able to compensate.
- When $S(\mathbf{x}, y)$ is known

$$\mathcal{R}_{\mathcal{P}^t}(f) = E_{\mathbf{x},y\sim\mathcal{P}^s}\left[\frac{1}{S(\mathbf{x}, y)}L(y, f(\mathbf{x}))\right] \tag{15}$$

- Same technique use for polls when estimation the votes in political elections.

Theory of generalization in DA



S. Ben-David

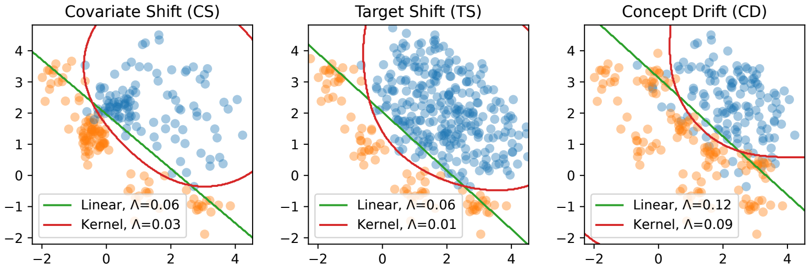
Y. Mansour

C. Cortes

A short and partial history of DA generalization

- Seminal results by [Ben-david et al., 2006] provided first bounds on $0-1$ classification losses using VC-dim.
- Generalization bounds for regression and classification by [Mansour et al., 2009].
- Bounds for regression using generalized discrepancy by [Cortes and Mohri, 2011, Cortes et al., 2015].
- Impossibility theorems [Ben-David et al., 2010, Ben-David and Urner, 2012].
- Bounds with MMD [Redko, 2015] and Wasserstein [Redko et al., 2017] discrepancies.
- PAC Bayes bounds for DA [Germain et al., 2013, Germain et al., 2016].

- Recent survey in [Redko et al., 2020a] and the book [Redko et al., 2019b], thesis of Sophiane Dhouib.



Definition [Ben-David et al., 2010, Def. 5]

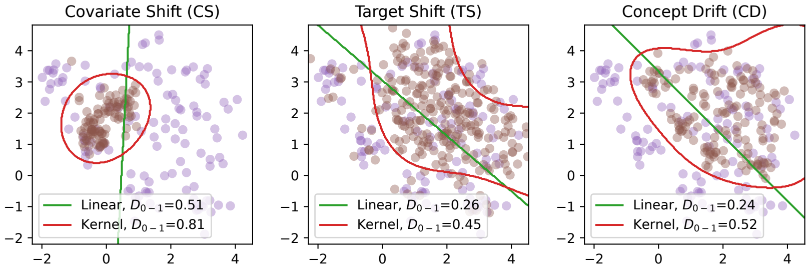
Let $\mathcal{P}^s$ and $\mathcal{P}^s$ be the distributions in the source and target domain respectively, the domain disagreement can be expressed for a given hypothesis space $\mathcal{H}$ as

$$\Lambda^{\mathcal{H}}(\mathcal{P}^s, \mathcal{P}^t) = \inf_{f \in \mathcal{H}} \quad \mathcal{R}_{\mathcal{P}^s}(f) + \mathcal{R}_{\mathcal{P}^t}(f) \tag{16}$$

- Measures if one can learn a unique predictor $\bar{f} \in \mathcal{H}$ that works on both domains.
- Originally proposed with loss L equal to the 0-1 loss in [Ben-david et al., 2006][1].

[1] Ben-david, S., Blitzer, J., Crammer, K., and Pereira, O. (2006). Analysis of representations for domain adaptation. **In Neural Information Processing Systems (NIPS). MIT Press**

Discrepancy distance between marginals



Definition [Mansour et al., 2009, Def. 4][2]

The discrepancy distance between two feature marginals $\mathcal{P}^s_{\mathcal{X}}$ and $\mathcal{P}^t_{\mathcal{X}}$ is defined as

$$D^{\mathcal{H}}_L(\mathcal{P}^s_{\mathcal{X}}, \mathcal{P}^t_{\mathcal{X}}) = \sup_{f, f' \in \mathcal{H}^2} \left| E_{\mathbf{x} \sim \mathcal{P}^s_{\mathcal{X}}}[L(f(\mathbf{x}), f'(\mathbf{x})) - E_{\mathbf{x} \sim \mathcal{P}^t_{\mathcal{X}}}[L(f(\mathbf{x}), f'(\mathbf{x}))] \right| \qquad (17)$$

- Measures the ability of two predictors to have different losses across domains (no labels needed). For classification to discriminate between source/target samples.
- Proposed in [Ben-David et al., 2010] for classification with L being the 0-1 loss illustrated above (and called $d_{\mathcal{H}\Delta\mathcal{H}}$).

[2] Mansour, Y., Mohri, M., and Rostamizadeh, A. (2009). Domain adaptation: Learning bounds and algorithms. **In Conference on Learning Theory (COLT), pages 19–30**

DA generalization bound [Ben-david et al., 2006, Thm 1][3]

The generalization of a predictor f on target can be bounded with probability $1-\delta$ as

$$\mathcal{R}_{\mathcal{P}^t}(f) \leq \mathcal{R}_{\widehat{\mathcal{P}}^s}(f) + D_{0-1}^{\mathcal{H}}(\widehat{\mathcal{P}}_{\mathcal{X}}^s, \widehat{\mathcal{P}}_{\mathcal{X}}^t) + \Lambda^{\mathcal{H}}(\mathcal{P}^s, \mathcal{P}^t) + \sqrt{\frac{4}{n}\left(C(\mathcal{H})\log\frac{2en}{C(\mathcal{H})} + \log\frac{4}{\delta}\right)} \tag{18}$$

- $C(\mathcal{H})$ is the VC (Vapnik-Chervonenkis) dimension that measures the complexity of the hypothesis space [Vapnik, 2006] and $n = n_s = n_t$.
- Bound on the classification error wih loss L equal to the $0-1$ loss.
- Similar result with general loss L in [Mansour et al., 2009] using Rademacher complexity instead of VC dimension.
- Generalization bounds for regression in [Cortes and Mohri, 2011].
- Similar bounds can replace the term $D_L^{\mathcal{H}}$ with MMD [Redko, 2015] and Wasserstein [Redko et al., 2017] discrepancies.

[3] Ben-david, S., Blitzer, J., Crammer, K., and Pereira, O. (2006). Analysis of representations for domain adaptation. **In Neural Information Processing Systems (NIPS). MIT Press**

$$\mathcal{R}_{\mathcal{P}^t}(f) \leq \underbrace{\mathcal{R}_{\widehat{\mathcal{P}}^s}(f)}_{\text{1. ERM}} + \underbrace{D^{\mathcal{H}}_{0-1}(\widehat{\mathcal{P}}^s_{\mathcal{X}}, \widehat{\mathcal{P}}^t_{\mathcal{X}})}_{\text{2. Emp. Marg. disc.}} + \underbrace{\Lambda^{\mathcal{H}}(\mathcal{P}^s, \mathcal{P}^t)}_{\text{3. Dom. disag.}} + \underbrace{\sqrt{\frac{4}{n}\left(C(\mathcal{H}) \log \frac{2en}{C(\mathcal{H})} + \log \frac{4}{\delta}\right)}}_{\text{4. Sampling term}}$$

1. Empirical risk on the samples of the source domain.
2. Empirical feature marginal discrepancy (how much $\widehat{\mathcal{P}}^s_{\mathcal{X}}$ and $\widehat{\mathcal{P}}^t_{\mathcal{X}}$ are different?).
3. Domain disagreement (can we train a predictor that work for both?)
4. Sampling term decreases with n but increases with complexity of $\mathcal{H}$ (overfiting).

Strategies (minimizing the bound)

- Train the predictor f on source while limiting the complexity (min 1+4).
- Change the empirical feature distributions to minimize the discrepancy (min 2, by re-weighting of feature learning).
- Hope that there exists and $\bar{f}$ that works on both domains or else you need to compensate for the shift (min 3).

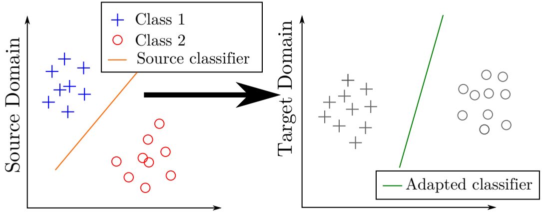
Supervised ML VS the real world

- DA comes from a practical problem : the test data does not follow the same distribution of the training data.
- Other practical constraints (or other sources of information) can lead to different problems :
  - Some labeled samples in target domains.
  - Multiple sources of information.
  - Data lying in different spaces ($\mathcal{X}^s \neq \mathcal{X}^t$), e.g. change of sensor.

Variants of DA problems

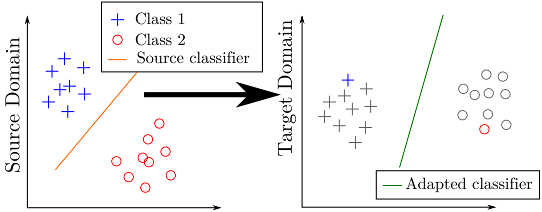
- Unsupervised DA and Semi-supervised DA.
- Multi-Source DA (MSDA) and Multi-target DA (MTDA).
- Heterogeneous DA (HDA)

Unsupervised DA



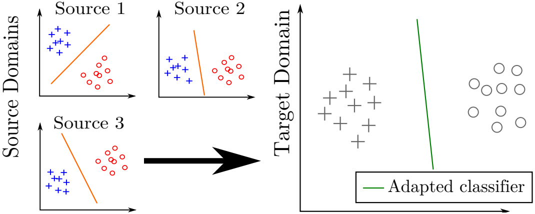
- Source : $\{\mathbf{x}_i^s, y_i^s\}_{i=1}^{n_s}$
- Target : $\{\mathbf{x}_j^t\}_{j=1}^{n_t}$
- Requires assumptions on the shift (CS, TS, CD, SSB).

Semi-Supervised DA



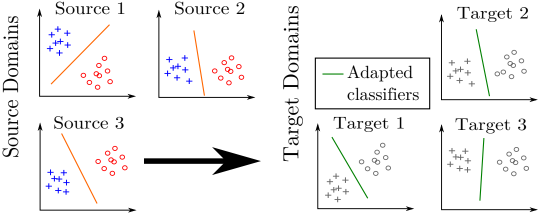
- Source : $\{\mathbf{x}_i^s, y_i^s\}_{i=1}^{n_s}$
- Target : $\{\mathbf{x}_j^t\}_{j=1}^{n_t}$, $\{y_j^t\}_{j=1}^{n_l}$
- The few $n_l \ll n_t$ labeled target samples can help guide the learning on target.

Multi-source DA

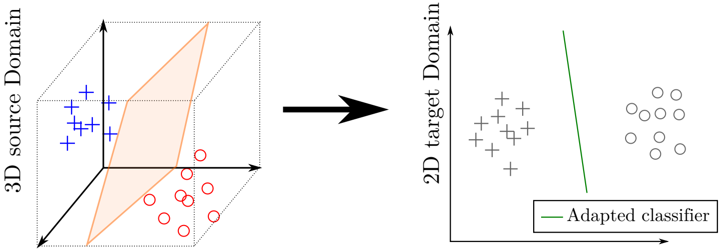


- Sources : $\{\mathbf{X}_k^s, \mathbf{y}_k^s\}_{k=1}^{D}$
- Target : $\{\mathbf{x}_j^t\}_{j=1}^{n_t}$
- D source domains available.
- Can use similarity between source and target domains.

Multi-DA (Multi-Source + Multi-Target)



- Source : $\{\mathbf{X}_k^s, \mathbf{y}_k^s\}_{k=1}^{D_s}$
- Target : $\{\mathbf{X}_k^t\}_{k=1}^{D_t}$
- $D_s = 1$ is Multi-Target DA and $D_t = 1$ is MSDA.
- Strong relation to Multi-Task Learning (MTL is $D_t = 0$)



Principle

- Feature samples lie in different spaces $\mathcal{X}^s \neq \mathcal{X}^t$.
- In the general case no relation is known a priori between the two spaces.
- Very hard problem so post approach perform semi-supervised HDA.
- Example: change in sensors or resolution and no knowledge about their correspondances.

DA VS Other ML techniques

DA VS Transfer Learning [Thrun and Pratt, 2012]

- Main difference : in TL the labels in the target domain can be different from the source domain ($\mathcal{Y}^s \neq \mathcal{Y}^t$) and usually labels are available in target.
- DA is a special case of transfer learning where the prediction task is the same.
- TL also often uses a pre-trained predictor (on source) instead of the raw datas.

DA VS Domain Generalization [Zhou et al., 2021]

- Main difference : DG searches for a unique predictor f that works on all possible domains and no samples are available from any of the target domains.
- One predictor to rule them all (a lot of research in computer vision).

DA VS semi-supervised learning [Chapelle et al., 2006]

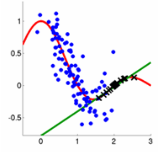
- Main difference : data assumptions are very different (often same distribution).
- Semi-supervised learning methods can be used on DA data (same datasets).
- Tools of semi-supervised (manifold, los density separation) also used in DA.

Always check what is solved in individual papers Tl, DA DG are not always used consistently.

Classical Domain Adaptation methods

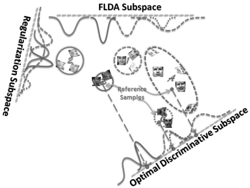
Reweighting schemes [Sugiyama et al., 2008]

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



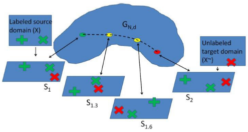
Subspace methods

- 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].



Alignment/mapping methods

- Alignment along the geodesic between source and target subspace [Gopalan et al., 2014].
- Geodesic flow kernel [Gong et al., 2012].
- Mapping alignment based on Optimal Transport [Courty et al., 2016].



Principle of sample reweighting

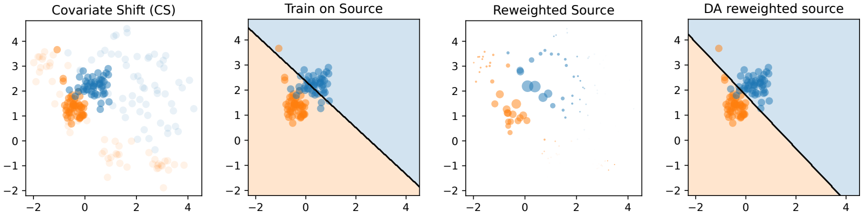
- The risk on target can be computed with

$$\mathcal{R}_{\mathcal{P}^t}(f) = E_{\mathbf{x},y\sim\mathcal{P}^s}\left[\frac{P^t(\mathbf{x},y)}{P^s(\mathbf{x},y)}L(y,f(\mathbf{x}))\right] \tag{19}$$

- If one can estimate a weighting function $w(\mathbf{x},y) = \frac{P^t(\mathbf{x},y)}{P^s(\mathbf{x},y)}$ then a good strategy is to minimize the reweighted source ERM

$$\min_{f\in\mathcal{H}} \quad \left\{\widehat{\mathcal{R}}^w(f) = \frac{1}{n_s}\sum_{i=1}^{n_s} w(\mathbf{x}_i,y_i)L(y_i,f(\mathbf{x}_i))\right\} \tag{20}$$

- Depending on the quality of the estimation of w the re-weighting can perfectly compensate the following data shifts
    - Covariate Shift (if supp($\mathcal{P}^s_{\mathcal{X}}$) $\subseteq$ supp($\mathcal{P}^t_{\mathcal{X}}$)) [Shimodaira, 2000].
    - Target Shift (if supp($\mathcal{P}^s_{\mathcal{Y}}$) $\subseteq$ supp($\mathcal{P}^t_{\mathcal{Y}}$)) .
    - Sample Selection Bias (if $S \neq 0$ on supp($\mathcal{P}^s$))
- Most methods propose ways to estimate w depending on the assumption and the data availability.



Principle

- Under Covariate Shift assumption, the optimal weight is $w(\mathbf{x}) = \frac{P^t_{\mathcal{X}}(\mathbf{x})}{P^s_{\mathcal{X}}(\mathbf{x})}$.
- The target risk can be bounded empirically [Cortes et al., 2010] for $\delta > 0$ with probability $1-\delta$

$$\mathcal{R}_{\mathcal{P}^t}(f) \leq \widehat{\mathcal{R}}^w(f) + 2^{5/4}\sqrt{D_R(\mathcal{P}^s_{\mathcal{X}}, \mathcal{P}^t_{\mathcal{X}})}\sqrt[3/8]{\frac{4}{n}\left(d\log\frac{2en}{d} + \log\frac{4}{\delta}\right)} \qquad (21)$$

where D_R is the 2-order Rényi divergence.

- Main difficulty is the estimation of the weights $w_i = w(\mathbf{x}_i)$ from empirical distributions.

Feature sample reweighting (2)

Estimation of the weights

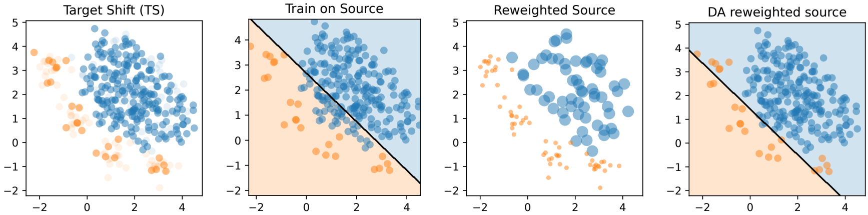
- Gaussian Approximation [Shimodaira, 2000] : $\hat{w}(\mathbf{x}) = \frac{\mathcal{N}(\mathbf{x}|\hat{\mu}^t, \hat{\Sigma}^t)}{\mathcal{N}(\mathbf{x}|\hat{\mu}^s, \hat{\Sigma}^s)}$
- Ratio of kernel density estimation [Sugiyama and Müller, 2005]

$$\hat{w}(\mathbf{x}) = \frac{\frac{1}{n_t}\sum_i k_{\sigma^t}(\mathbf{x}, \mathbf{x}_i^t)}{\frac{1}{n_s}\sum_j k_{\sigma^t}(\mathbf{x}, \mathbf{x}_j^s)} \tag{22}$$

- Nearest neighbor density estimation [Loog, 2012, Kremer et al., 2015]
- Divergence minimization methods

$$\min_w \quad D\left(\frac{1}{n_s}\sum_i w(\mathbf{x}_i^s)\delta_{\mathbf{x}_i^s}, \widehat{\mathcal{P}}_{\mathcal{X}}^t\right) \tag{23}$$

  where D is a divergence such as
  - MMD for Kernel Mean Matching (KMM) [Huang et al., 2006, Gretton et al., 2009].
  - Kullback-Leilbler divergence for KL Importance Estimation Procedure (KLIEP) [Sugiyama et al., 2007].
  - L2 norm between the weights and the ratio (with kernels)[Kanamori et al., 2009].
- Logistic regression classifying source VS target and use the conditional probability $\hat{w}(\mathbf{x}) \propto P(\text{domain} = \text{target}|\mathbf{x})$ as scaling [Sugiyama et al., 2012].

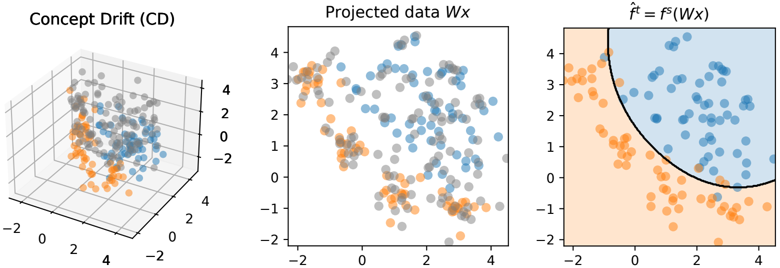


Principle and methods

- Under target Shift assumption, the optimal weight is $w(y) = \frac{P^t_{\mathcal{Y}}(y)}{P^s_{\mathcal{Y}}(y)}$.
- The target risk can be bounded empirically similarly to covariate shift [Cortes et al., 2010].
- Black Box Shift Estimation (BBSE) [Lipton et al., 2018] uses a pre-trained trained classifier h with confusion matrix $\hat{\mathbf{C}}_{h(\mathbf{x}),y}$ on source and estimates the ratios as

$$\hat{\mathbf{w}} = \hat{\mathbf{C}}^{-1}_{h(\mathbf{x}),y}\hat{\mathbf{p}} \quad \text{where} \quad \hat{p}_i = \hat{P}^t(h(\mathbf{x}) = i)$$

- $\widehat{\mathcal{P}}^t_{\mathcal{Y}}(y)$ can be estimated by divergence minimization such as Kernel Mean Matching [Zhang et al., 2013] or Wasserstein distance [Redko et al., 2019a].



General principle

- Assumption: there exists a subspace of the data where the domains are similar ($\mathbf{W}\#\mathcal{P}^s_{\mathcal{X}} \approx \mathbf{W}\#\mathcal{P}^t_{\mathcal{X}}$) and where the label information is preserved.
- Estimate a projection $\mathbf{W} \in \mathbb{R}^{d' \times d}$ where $d' \leq d$ (in direct or kernel space).
- Project the source samples with $\mathbf{W}$ as $\tilde{\mathbf{x}}^s_i = \mathbf{W}\mathbf{x}^s_i$ ($\mathbf{W}\#\widehat{\mathcal{P}}^s_{\mathcal{X}} = \frac{1}{n_s}\delta_{\mathbf{W}\mathbf{x}^s_i}$.)
- Train a predictor $\hat{f}$ on the projected source samples $\{\tilde{\mathbf{x}}^s_i, y^s_i\}_i$.
- Predictor on target is $\hat{f}^s(\mathbf{x}) = \hat{f}(\mathbf{W}\mathbf{x})$.
- Works better on data in high dimension where such a subspace can exist.
- Nonlinear invariant transformation with kernels or deep learning (next section).

Principle [Pan et al., 2010]

- Search for a kernel subspace mapping m that minimizes the MMD divergence between the domains while preserving the variance.
- TCA consists in finding a (kernel) projection matrix $\mathbf{W}$ solving

$$\min_{\mathbf{W}} \quad \text{Tr}(\mathbf{W}^\top\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{W}) + \lambda\text{Tr}(\mathbf{W}^\top\mathbf{W}) \tag{24}$$

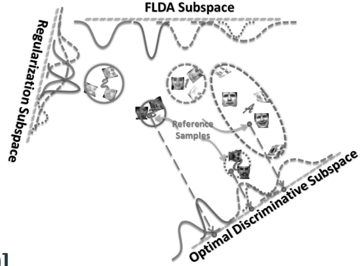
$$\text{s.t.} \quad \mathbf{W}^\top\mathbf{K}\mathbf{H}\mathbf{K}\mathbf{W} = \mathbf{I} \tag{25}$$

with $\mathbf{K} = \begin{bmatrix} \mathbf{K}^s & \mathbf{K}^{s,t} \\ \mathbf{K}^{t,s} & \mathbf{K}^t \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} \frac{1}{n_s^2}\mathbf{1} & -\frac{1}{n_s n_t}\mathbf{1} \\ -\frac{1}{n_s n_t}\mathbf{1} & \frac{1}{n_t^2}\mathbf{1} \end{bmatrix}, \quad \mathbf{H} = \mathbf{I} - \frac{1}{n_s + n_t}\mathbf{1}$

  $\mathbf{K}$ is the kernel matrix between all source and target samples, $\mathbf{L}$ is a scaling matrix used to compute the MMD between domains and $\mathbf{H}$ is a centering matrix used for computing the variance.

- The projection matrix $\mathbf{W}$ is obtained with an eigen-decomposition of $(\mathbf{K}\mathbf{L}\mathbf{K} + \lambda\mathbf{I})^{-1}K\mathbf{H}\mathbf{K}$.
- Can be seen as a kernel PCA between domains [Schölkopf et al., 1997].

Transfer Subspace Learning



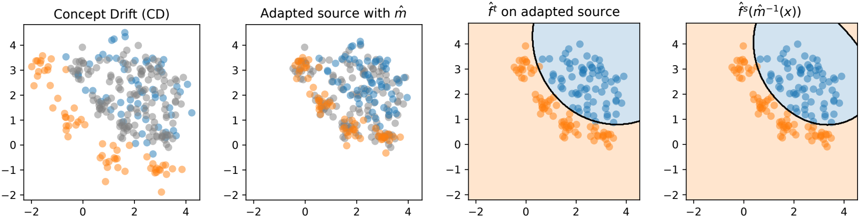
Principle [Si et al., 2010]

- Minimize the Bregman divergence between the projected domains and a learning loss as a function of the projection matrix $\mathbf{W} \in \mathbb{R}^{d' \times d}$

$$\min_{\mathbf{W}, \mathbf{W}^\top \mathbf{W} = I} \quad D\left(\mathbf{W} \# \widehat{\mathcal{P}}^s_{\mathcal{X}}, \mathbf{W} \# \widehat{\mathcal{P}}^t_{\mathcal{X}}\right) + F(\mathbf{W}) \tag{26}$$

  where $\#$ is the pushforward operator and the learning $F(\mathbf{W})$ loss can be :
  - Reconstruction loss (PCA)
  - Fisher Linear Discriminant loss (FDA)
  - Locality Preserving Projection loss (LPP) [He and Niyogi, 2003]
- Use MMD as divergence in [Baktashmotlagh et al., 2013].
- TSL with sample reweighting [Long et al., 2014]
- Use pseudo labels to promote discrimination (see self labeling) [Long et al., 2013]

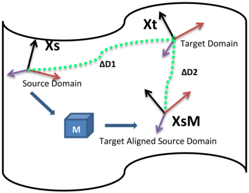
Alignment methods



General principle

- Assumption: there exists a mapping of the source data such that $P^s(m(\mathbf{x}), y) = P^t(\mathbf{x}, y)$ (concept drift).
- Estimate a projection the mapping $\hat{m}$ from the data (usually with some assumptions) and map the source samples $\tilde{\mathbf{x}}_i^s = \hat{m}(\mathbf{x}_i^s)$
- Several strategies:
  - Train a predictor on the projected source samples $\{\tilde{\mathbf{x}}_i^s, y_i^s\}_i$.
  - Train a predictor $\hat{f}^s$ on source and predict with $f^t(\mathbf{x}) = f^s(\hat{m}^{-1}(\mathbf{x}))$.
  - Train a prediction $\hat{f}$ invariant to the mapping $\hat{m}$ that is $\hat{f}(\mathbf{x}) = \hat{f}(\hat{m}(\mathbf{x}))$ (similar to subspace method but stronger assumption that such invariant classifier exists).

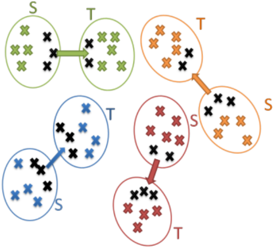
Subspace Alignment (SA)

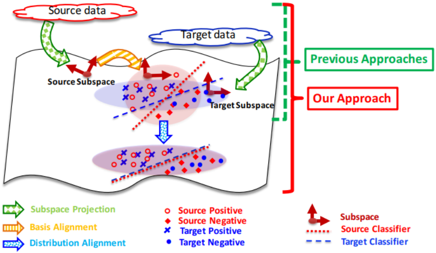


Principle [Fernando et al., 2013][4]

- The exists a mapping m between the source and target that aligns the covariances of source and target.
- The optimal mapping under their assumption is a correspondances between the sorted eigenvectors of the covariances.
- SA consists in the following steps :
  1. Estimate the $d' \leq d$ eigenvectors matrices with largest eigenvalues $\mathbf{V}^s$ and $\mathbf{V}^t$ on source and target.
  2. Apply the mapping $m(\mathbf{x}) = \mathbf{V}^t\mathbf{V}^{s\top}\mathbf{x}$ on the source samples to get $\tilde{\mathbf{x}}_i^s$.
  3. Train a target predictor $\hat{f}$ on adapted dataset $\{\tilde{\mathbf{x}}_i^s, y_i^s\}_i$

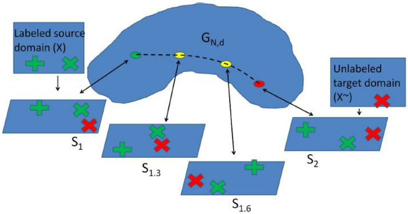
[4]Fernando, B., Habrard, A., Sebban, M., and Tuytelaars, T. (2013). Unsupervised visual domain adaptation using subspace alignment. **In Proceedings of the IEEE international conference on computer vision, pages 2960–2967**

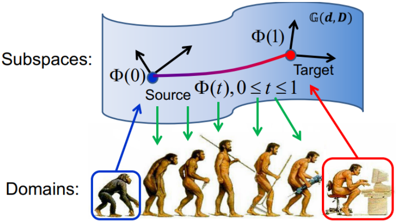




Extensions of Subspace Alignement

- Landmarks (selected in both domains) + kernel as pre-processing for subspace alignment [Aljundi et al., 2015].
- Joint estimation of subspace and classifier [Fernando et al., 2015].
- Subspace Distribution Alignment (SDA) perform SSA mapping plus a distribution alignment optimizing first and second order moments [Sun and Saenko, 2015].



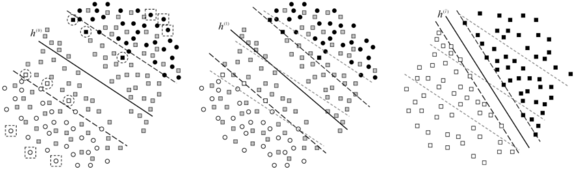


Geodesic on the Grassmann Manifold [Gopalan et al., 2011, Gopalan et al., 2013]

- Model evolution of the subspaces from $\mathbf{V}^s$ to $\mathbf{V}^t$ along the Grassmann Manifold.
- Update the data incrementally toward target and train classifier.
- Samples can be represented with domain invariant features (along the discretized geodesic).

Geodesic Flow Kernel (GFK) [Gong et al., 2012]

- Same modeling as above but complete integration instead of a discretization.
- Avoid the selection of the number of intermediate steps.
- Allow to compute features (and a kernel) invariant to the domain (integrated along the manifold) .



(a) (b) (c)

General principle

- Estimate labels for the target domain to learn a better classifier.
- Update the labels iteratively when updating the DA model.

Self-labeling DA methods

- SVM margin used to select target samples labeled that are used for updating predictor (DASVM) [Bruzzone and Marconcini, 2010].
- Iterative self labeling [Habrard et al., 2013] uses [Balcan et al., 2008].
- Label iteratively target samples with co-training [Chen et al., 2011] (inspired from semi-supervised co-training).
- Transfer Feature Learning aim at estimating a discriminant subspace and updates iteratively the target labels [Long et al., 2013].

Principle

- Minimax estimators are robust to changes in the target labels or training data.
- **Robust Bias-Aware classifier [Liu and Ziebart, 2014]** :

$$\min_{f\in\mathcal{H}} \max_{g\in\mathcal{H},\|g-\hat{f}^s\|\leq\varepsilon} \quad \frac{1}{n_t}\sum_{j=1}^{n^t} L(g(\mathbf{x}_j^t), f(\mathbf{x}_j^t))$$

- **Robust Covariate Shift Adjustment (RCSA) [Wen et al., 2014]**:

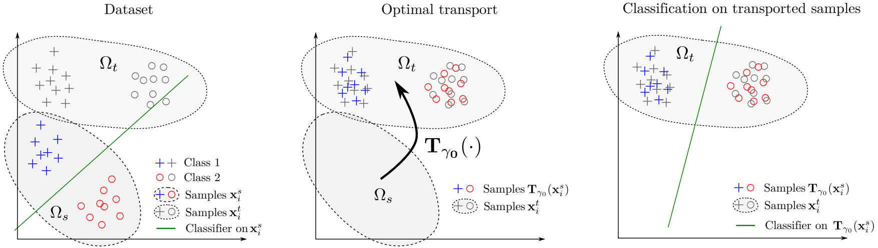
$$\min_{f\in\mathcal{H}} \max_{\mathbf{w}\in\Delta_n} \quad \frac{1}{n_t}\sum_{i=1}^{n^s} L(y_i^s, f(\mathbf{x}_i^s))w_i$$

Distributionaly Robust Optimization [Hu et al., 2018, Kuhn et al., 2019]

$$\min_{f} \max_{\mathcal{P}\in B_\varepsilon(\hat{\mathcal{P}}_s)} E_{\mathbf{x},y\sim\mathcal{P}}[L(y, f(\mathbf{x}))] \tag{27}$$

- $B_\varepsilon(\hat{\mathcal{P}}_s)$ is the ball around $\hat{\mathcal{P}}_s$ for a given divergence.
- This ensures a given performance when $\mathcal{P}^t$ is in the ball (close to $\hat{P}^s$).
- The ball can be the KL divergence [Hu et al., 2018] or Wasserstein distance [Kuhn et al., 2019].

Optimal transport for domain adaptation



Assumptions

1. There exist an OT mapping T in the feature space between the two domains.
2. The transport preserves the joint distributions:

$$P^s(\mathbf{x}, y) = P^t(T(\mathbf{x}), y).$$

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

1. Estimate optimal transport between distributions (use regularization).
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.

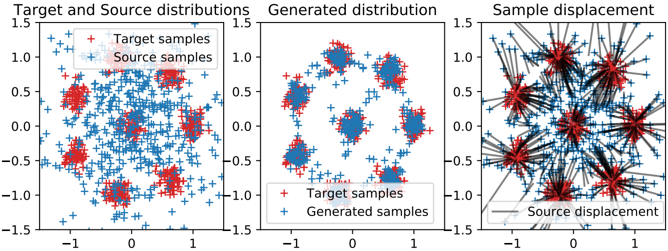
Generalization bound [Flamary et al., 2021]

Let f^s 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 assumption 2 we have the following generalization bound

$$R_t(f^s \circ \hat{T}^{-1}) \leq R_s(f^s) + M_f M_L \mathbb{E}_{(x,y)\sim\mathcal{P}_s}\left[\|\hat{T}^{-1}(T(x)) - \hat{T}^{-1}(\hat{T}(x))\|\right] \tag{28}$$

- Train a classifier f on source and estimate a mapping $\hat{T}^{-1}$ from target to source.
- True for any mapping T (not only OT mapping).
- Need out of sample mapping $\hat{T}^{-1}$ (to map new target samples).

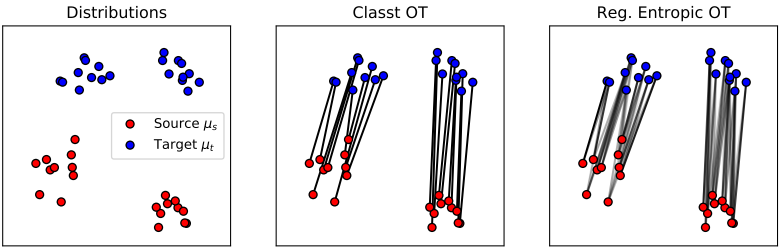
Mapping with optimal transport



Monge mapping estimation

- Mapping does not exist in general between empirical distributions.
- Barycentric mapping [Ferradans et al., 2014].
- Smooth mapping estimation [Perrot et al., 2016, Seguy et al., 2018].
- 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]) .

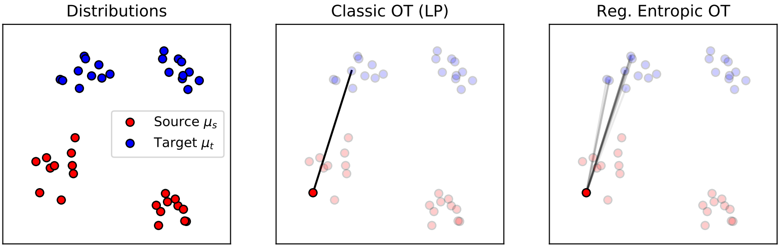
Transporting the discrete samples



Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\mathbf{T}_0}(\mathbf{x}_i^s) = \underset{\mathbf{x}}{\operatorname{argmin}} \quad \sum_j T_{i,j} c(\mathbf{x}, \mathbf{x}_j^t). \tag{29}$$

- The mass of each source sample is spread onto the target samples (line of $\mathbf{T}_0$).
- The mapping is the barycenter of the target samples weighted by $\mathbf{T}_0$
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

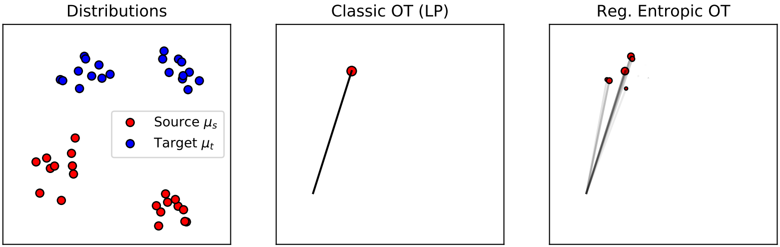


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\mathbf{T}_0}(\mathbf{x}_i^s) = \underset{\mathbf{x}}{\text{argmin}} \quad \sum_j \mathbf{T}_0(i,j) \|\mathbf{x} - \mathbf{x}_j^t\|^2. \tag{29}$$

- The mass of each source sample is spread onto the target samples (line of $\mathbf{T}_0$).
- The mapping is the barycenter of the target samples weighted by $\mathbf{T}_0$
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Transporting the discrete samples

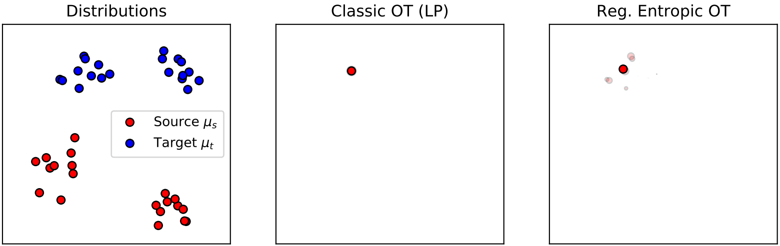


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\mathbf{T}_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \mathbf{T}_0(i,j)} \sum_j \mathbf{T}_0(i,j)\mathbf{x}_j^t. \tag{29}$$

- The mass of each source sample is spread onto the target samples (line of $\mathbf{T}_0$).
- The mapping is the barycenter of the target samples weighted by $\mathbf{T}_0$
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Transporting the discrete samples

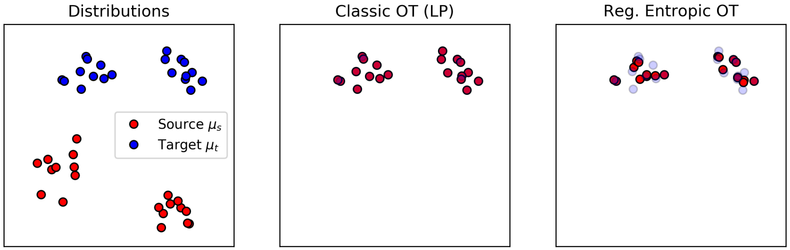


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\mathbf{T}_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \mathbf{T}_0(i,j)} \sum_j \mathbf{T}_0(i,j)\mathbf{x}_j^t. \tag{29}$$

- The mass of each source sample is spread onto the target samples (line of $\mathbf{T}_0$).
- The mapping is the barycenter of the target samples weighted by $\mathbf{T}_0$
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

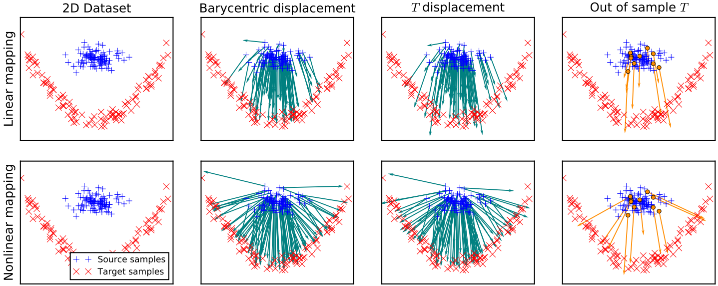
Transporting the discrete samples



Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\mathbf{T}_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \mathbf{T}_0(i,j)} \sum_j \mathbf{T}_0(i,j)\mathbf{x}_j^t. \quad (29)$$

- The mass of each source sample is spread onto the target samples (line of $\mathbf{T}_0$).
- The mapping is the barycenter of the target samples weighted by $\mathbf{T}_0$
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

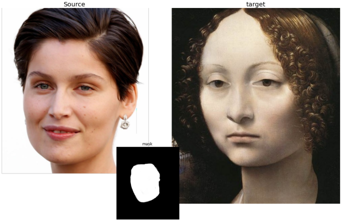


Simultaneous OT matrix and mapping [Perrot et al., 2016]

$$\min_{T,\mathbf{T}\in\mathcal{P}} \quad \langle \mathbf{T}, \mathbf{C}\rangle_F + \sum_i \|T(\mathbf{x}_i^s) - \hat{T}_{\mathbf{T}}(\mathbf{x}_i^s)\|^2 + \lambda \|T\|^2$$

- Estimate jointly the OT matrix and a smooth mapping approximating the barycentric mapping.
- The mapping is a regularization for OT.
- Controlled generalization error (statistical bound).
- Linear and kernel mappings T, limited to small scale datasets.

Seamless copy in images



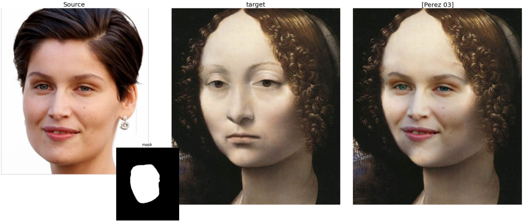
Poisson image editing [Pérez et al., 2003]

- 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.

Example and webcam demo: `https://github.com/ncourty/PoissonGradient`



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.

Example and webcam demo: `https://github.com/ncourty/PoissonGradient`

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.

Example and webcam demo: `https://github.com/ncourty/PoissonGradient`

Seamless copy in images

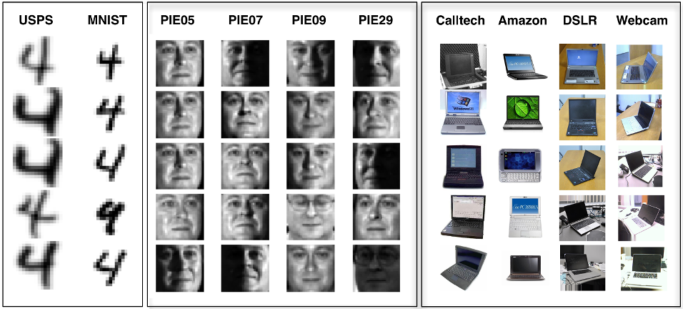
Poisson image editing [Pérez et al., 2003]

- 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.

Example and webcam demo: `https://github.com/ncourty/PoissonGradient`



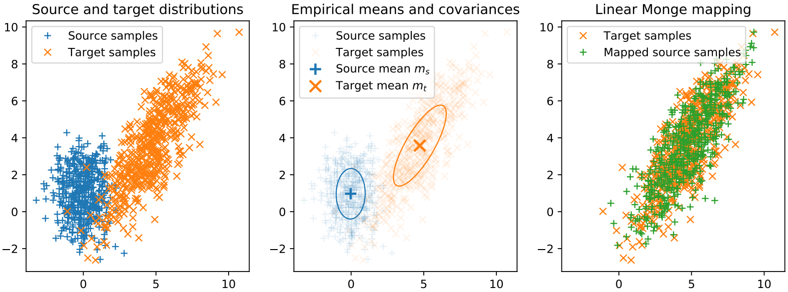
Datasets

- **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.

Special case: OT mapping between Gaussians



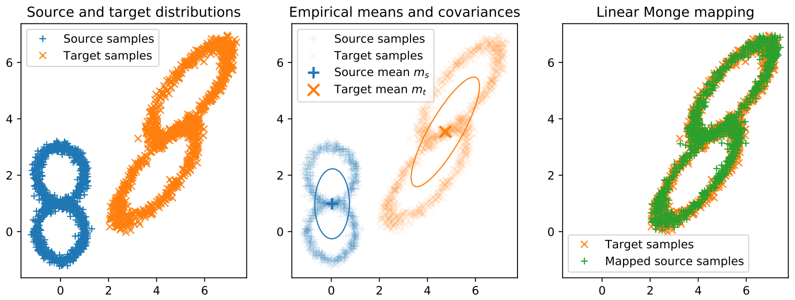
OT mapping between Gaussian distributions

- $\mathcal{P}^s{}_\mathcal{X} \sim \mathcal{N}(\mathbf{m}_1, \Sigma_1)$ and $\mathcal{P}^t{}_\mathcal{X} \sim \mathcal{N}(\mathbf{m}_2, \Sigma_2)$
- 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., 2021].



OT mapping between Gaussian distributions

- $\mathcal{P}^s{}_{\mathcal{X}} \sim \mathcal{N}(\mathbf{m}_1, \Sigma_1)$ and $\mathcal{P}^t{}_{\mathcal{X}} \sim \mathcal{N}(\mathbf{m}_2, \Sigma_2)$
- 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., 2021].

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., 2021])

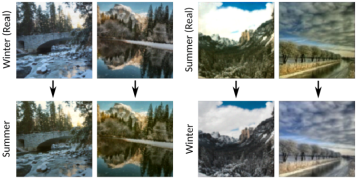
Let μ_1 and μ_2 be sub-Gaussian distributions with expectations m_1, m_2 and positive-definite covariance operators Σ_1, Σ_2 respectively with eigenvalues in $[c, C]$ for some fixed absolute constants $0 < c \leq C < \infty$. We also assume that $n_j \geq C\mathbf{r}(\Sigma_j), \quad j = 1, 2$, for some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

Monge mapping for Image-to-Image translation



Principle

- Encode image as a distribution in a DNN embedding.
- Transform between images using estimated Monge mapping.
- Linear Monge Mapping (Wasserstein Style Transfer [Mroueh, 2019]).
- Nonlinear Monge Mapping using input Convex Neural Networks [Korotin et al., 2019].
- Allows for transformation between two images but also style interpolation with Wasserstein barycenters.

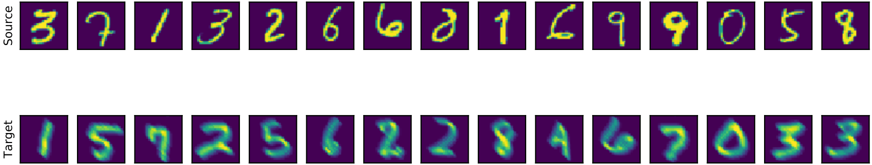
Estimator in source domain
Let $\mathcal{H}_K$ be a reproducing kernel Hilbert space (RKHS) associated with a symmetric nonnegatively definite kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ We consider the following empirical risk minimization estimator:

$$\hat{f}^s_{n_s} := \operatorname{argmin}_{\|f\|_{\mathcal{H}_K} \leq 1} \frac{1}{n_s} \sum_{i=1}^{n_s} L(y_i^s, f(\mathbf{x}_i^s)). \tag{30}$$

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]).

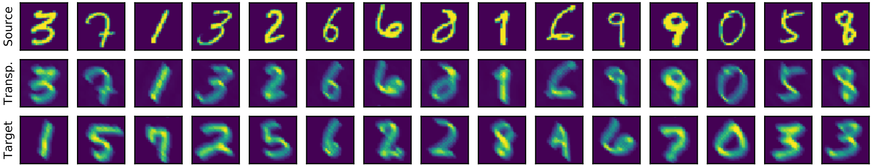
OTDA generalization bound
If $R_s(f_*^s) = R_t(f_*^t)$ and $\hat{T}$ is the linear monge mapping estimator, under the assumptions of OTDA, we get with probability at least $1 - e^{-t} - \frac{1}{n_1}$,

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



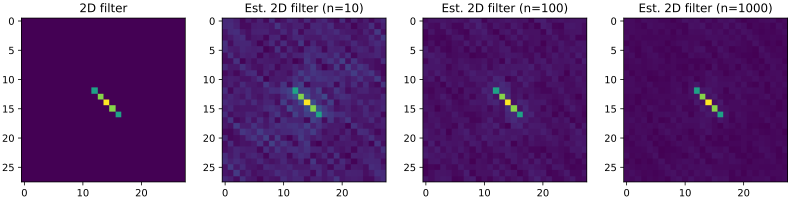
Numerical experiments

- 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).



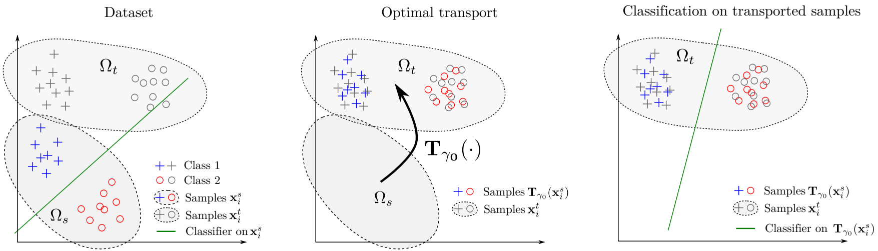
Numerical experiments

- 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., 2018].
- Nice generalization bound for linear Monge mappings [Flamary et al., 2021].

But

- Model transformation only in the feature space (requires $\mathcal{P}_{\mathcal{Y}}^s = \mathcal{P}_{\mathcal{Y}}^t$).
- Requires the same class proportion between domains [Tuia et al., 2015].
- Estimate a $T : \mathbb{R}^d \to \mathbb{R}^d$ mapping for training a classifier $f : \mathbb{R}^d \to \mathbb{R}$.

Deep Domain Adaptation

Generalization bound for shallow methods

$$\mathcal{R}_{\mathcal{P}^t}(f) \leq \underbrace{\mathcal{R}_{\widehat{\mathcal{P}}^s}(f)}_{\text{1. ERM}} + \underbrace{D^{\mathcal{H}}_{0-1}(\widehat{\mathcal{P}}^s_{\mathcal{X}}, \widehat{\mathcal{P}}^t_{\mathcal{X}})}_{\text{2. Emp. Marg. disc.}} + \underbrace{\Lambda^{\mathcal{H}}(\mathcal{P}^s, \mathcal{P}^t)}_{\text{3. Dom. disag.}} + \underbrace{\sqrt{\frac{4}{n}\left(C(\mathcal{H}) \log \frac{2en}{C(\mathcal{H})} + \log \frac{4}{\delta}\right)}}_{\text{4. Sampling term}}$$

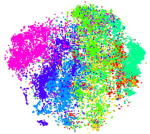
- Classical DA methods minimize part 1 and 2 by learning a classifier on source and limiting the discrepancy (e.g. with re-weighting).
- But they are limited by their original feature space of fixed kernel representations.

What deep learning can do?

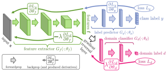
- Learn feature representation g that can both discriminate (part 1) and minimize the domain discrepancy (part 2).
- For concept drift with a feature mapping deep learning can be used to estimate this mapping between domain.

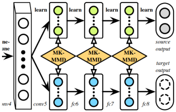
A short history of Deep DA

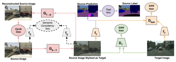
(b) Pairwise constraints (c) Invariant space



(d) DeCAF$_6$





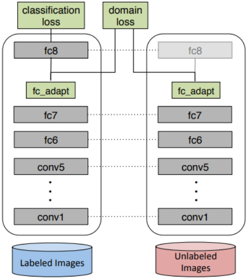


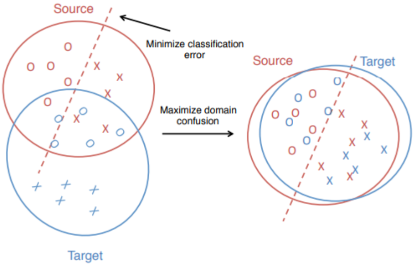
- Visual DA promoting similarity between pairs in the feature space (metric learning, partly supervised) [Saenko et al., 2010].
- A Deep Convolutional Activation Feature (DeCAF) one of the first open source visual features robust to domains and tasks [Donahue et al., 2014].
- Deep Domain Confusion [Tzeng et al., 2014] Deep Adaptation Network (DAN) uses MMD to minimize feature marginal domain discrepancy [Long et al., 2015].
- Domain Adversarial Neural Network (DANN) measure the discrepancy between domains using a classifier [Ganin et al., 2016].
- Joint Adaptation network (JAN) minimize the joint MMD across layers [Long et al., 2017].
- [Hoffman et al., 2018] Cycle-Consistent Domain Adaptation uses CycleGAN to lean mappings between domains.

Principle

$$\min_{f,g} \quad \underbrace{\frac{1}{n_s}\sum_{i=1}^{n_s} L(y_i^s, f(g(\mathbf{x}_i^s)))}_{\text{Loss on source}} + \lambda \underbrace{D(g\#\hat{\mathcal{P}}_{\mathcal{X}}^s, g\#\hat{\mathcal{P}}_{\mathcal{X}}^t)}_{\text{Disc. on feature marginals}} \tag{31}$$

- f is the predictor model in the embedding and g the embedding model, final predictor is $f \circ g$.
- D is a discrepancy measure between the empirical feature marginal distribution extracted with g.
- The main assumption is that one can learn an embedding that is both discriminant (for both domains) and invariant to the domains (the feature distributions are the same).
- Reasonable assumption in visual domain adaptation where a given class can be "disentangled" from the style or acquisition procedures of the domains.
- Several existing methods that differ mainly from their choice of D.

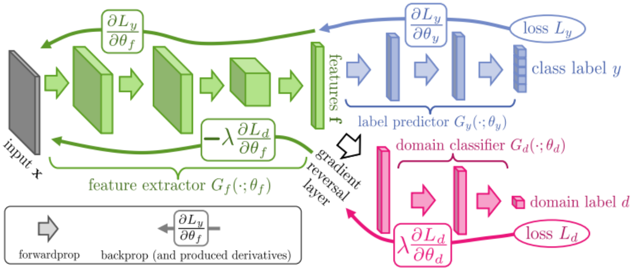




Principle [Tzeng et al., 2014][5]

- Choose the discrepancy D as MMD : $MMD(g\#\mathcal{P}^s_{\mathcal{X}}, g\#\mathcal{P}^t_{\mathcal{X}})^2$.
- The objective can be optimized efficiently with stochastic optimization.
- Extended to a joint MMD across layers called Deep Adaptation Networks (DAN) in [Long et al., 2015] : $MMD(\{g_l\}_l\#\mathcal{P}^s_{\mathcal{X}}, \{g_l\}_l\#\mathcal{P}^t_{\mathcal{X}})^2$ with g_l embedding function for layer l.

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

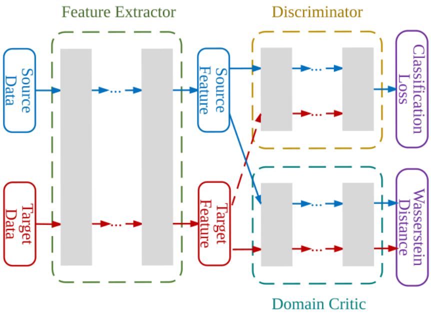


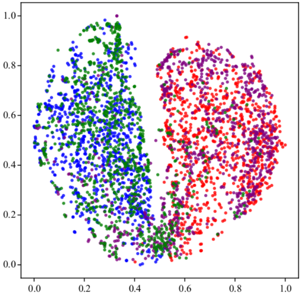
Principle [Ganin et al., 2016]

$$\min_{f,g}\max_{f^c}\frac{1}{n_s}\sum_{i=1}^{n_s}L(y_i^s,f(g(\mathbf{x}_i^s)))-\lambda\left(\frac{1}{n_s}\sum_{i=1}^{n_s}L^c(0,f^c(g(\mathbf{x}_i^s)))+\frac{1}{n_t}\sum_{j=1}^{n_t}L^c(1,f^c(g(\mathbf{x}_j^t)))\right) \tag{32}$$

- Choose the discrepancy D as minus the classification loss for an adversarial domain classifier (classical GAN objective).
- The backprop of g wrt the adversarial loss is negative : gradient reversal.
- Adversarial discriminant DA (ADDA) proposed to learn two independent embeddings g^s and g^t (no shared weights) [Tzeng et al., 2017].

Wasserstein Distance Guided Representation Learning (WDGRL)





(d) t-SNE of WDGRL features

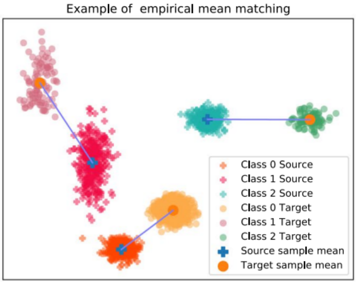
Principle [Shen et al., 2018]

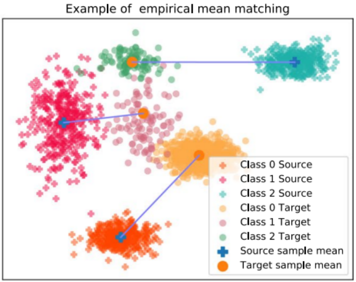
- Choose the discrepancy D as the Wasserstein distance (no vanishing gradients).
- Use the WGAN loss [Arjovsky et al., 2017] that relies on the dual formulation of the W_1 distance :

$$W_1(\mathcal{P}^s_{\mathcal{X}}, \mathcal{P}^t_{x}) = \max_{\phi \in \mathsf{Lip}_1} \quad E_{\mathbf{x}\sim\mathcal{P}^s_{\mathcal{X}}}[\phi(\mathbf{X})] - E_{\mathbf{x}\sim\mathcal{P}^t_{\mathcal{X}}}[\phi(\mathbf{X})] \tag{33}$$

- Approximating the Lipschitzness of ϕ with constraints or penalization [Gulrajani et al., 2017].

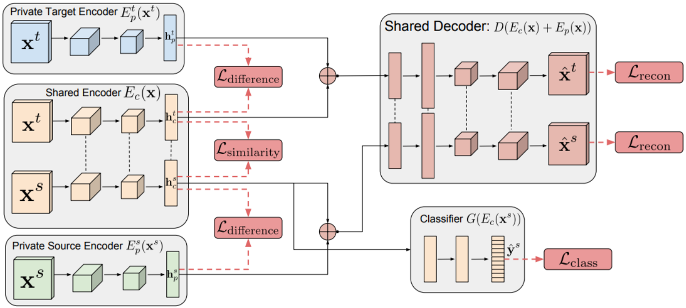
Match and reweight Domain Adaptation (MARS)





Principle [Rakotomamonjy et al., 2022]

- Proposed to handle both concept drift and target shift.
- Step 1 : estimation of target proportions $\hat{\mathbf{p}}^t$:
  - $\mathcal{P}_j^t, \tilde{\mathbf{p}} \leftarrow$ Estimate a mixture of K distribution on target (K-means/GMM)
  - $\mathbf{C} \leftarrow$ Compute the ground cost between $\mathcal{P}_\mathcal{X}(\mathbf{x}|y=i)$ and the mixture above.
  - $\mathbf{T}^* \leftarrow$ Solve OT between uniform weights on $\mathbf{C}$.
  - $\hat{\mathbf{p}}^t \leftarrow K\mathbf{T}^*\tilde{\mathbf{p}}$ compute target class proportion withy OT permutation.
- Step 2 : Perform domain invariant feature learning with Wasserstein distance [Shen et al., 2018] using the estimated class based reweighting on source (both on empirical risk and W_1).



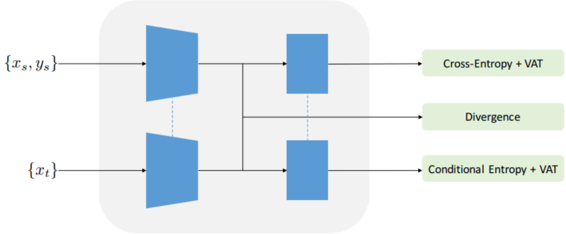
Domain Separation Networks [Bousmalis et al., 2016]

- Learn both an invariant embedding and domain specific (private) embeddings.
- Optimize classifier on labeled source using shared encoding and reconstruction losses from the private/shared encodings on both domains (disentanglement).

Deep Correlation Alignment (DeepCORAL) [Sun and Saenko, 2016]

$$D(g\#\mathcal{P}^s_{\mathcal{X}}, g\#\mathcal{P}^t_{\mathcal{X}}) = \|\hat{\boldsymbol{\Sigma}}^s - \hat{\boldsymbol{\Sigma}}^s\|_F^2 \tag{34}$$

where $\hat{\boldsymbol{\Sigma}} = E_{\mathbf{x}\sim g\#\widehat{\mathcal{P}}_{\mathcal{X}}}[(\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^\top]$, is with $\mathbf{m} = E_{\mathbf{x}\sim g\#\widehat{\mathcal{P}}_{\mathcal{X}}}[\mathbf{x}]$ is the empirical covariance in the feature space.



Principle [Shu et al., 2018]

- Adversarial loss between the embedding similar to DANN [Ganin et al., 2016].
- Conditional entropy minimization on target [Grandvalet and Bengio, 2004].

$$-E_{\mathbf{x}\sim\widehat{\mathcal{P}}^t_{\mathcal{X}}}[f(g(\mathbf{x}))^\top \log(f(g(\mathbf{x})))]$$

- Virtual Adversarial training (VAT) on target and source [Miyato et al., 2018]:

$$E_{\mathbf{x}\sim\widehat{\mathcal{P}}^t_{\mathcal{X}}}[\max_{\|\mathbf{v}\|\leq\varepsilon} KL(f(g(\mathbf{x}))|f(g(\mathbf{x}+\mathbf{v})))]$$

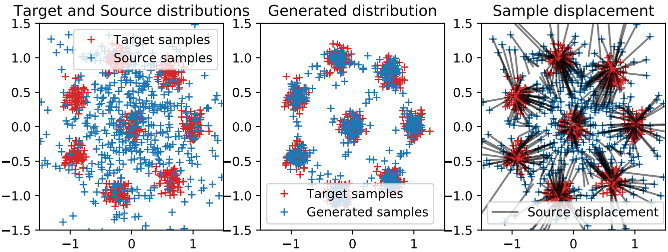
- Decision-boundary iterative refinement training promotes cluster assumptions on target (DIRT-T).

Principle [Benaim and Wolf, 2017]

- Conditional GAN can learn mappings between distributions.
- But there exists an infinity of mapping most of them do not preserve labels.
- Use regularization of the mapping so that it can preserve pairwise distance:

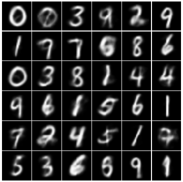
$$E_{\mathbf{x},\mathbf{x}'\sim(\widehat{\mathcal{P}}^s)^2}[|(\|\mathbf{x}-\mathbf{x}'\|-\mathbf{m}_s)/\sigma_s-(\|m(\mathbf{x})-m(\mathbf{x}')\|-\mathbf{m}_t)/\sigma_t|] \tag{35}$$

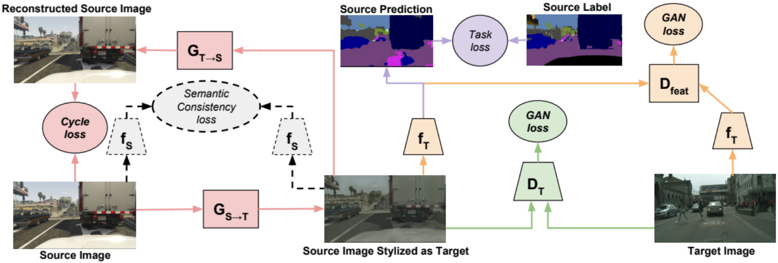
- Also promote consistant self distance between half of each images.



Large scale OT mapping estimation [Seguy et al., 2018]

- OTDA [Courty et al., 2016] has been shown to work on deep embedding but did not scale to large scale datasets.
- For a fixed feature representation one can estimate an OT mapping using entropic OT. 2-step procedure:
  1. Stochastic estimation of regularized $\hat{\mathbf{T}}$ in the dual with neural networks.
  2. Stochastic estimation of T with a neural network.
- Convergence to the true mapping for small regularization [Seguy et al., 2018] and to the entropic mapping for large n [Pooladian and Niles-Weed, 2021].

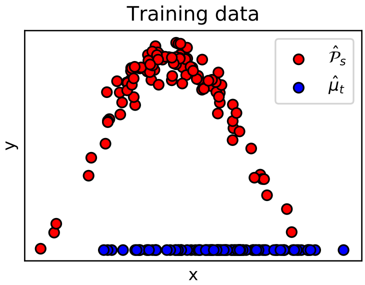




Principle [Hoffman et al., 2018][6]

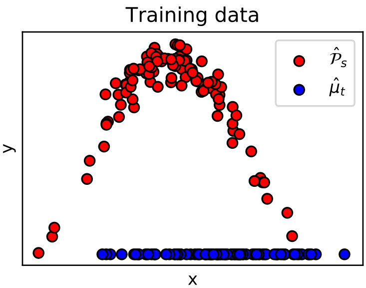
- Learn a mapping m from source to target and u from target to source such that $u(m(\mathbf{x})) \approx \mathbf{x}$ (both from reconstruction and semantic (class preservation)).
- Followed by an invariant DA between the mapped source and target data.
- Uses GAN losses to promote similarity between mapped source and target in the embedding.

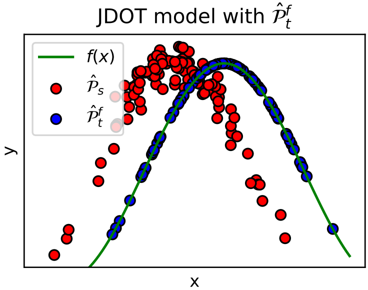
[6] Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. (2018). Cycada: Cycle-consistent adversarial domain adaptation. **In International conference on machine learning, pages 1989–1998. PMLR**



Main idea

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





Learning with JDOT [Courty et al., 2017]

$$\min_f \quad \left\{ W_1(\hat{\mathcal{P}}^s, \hat{\mathcal{P}^t}_f) = \inf_{\mathbf{T}\in\Pi} \sum_{ij} \mathcal{D}(\mathbf{x}_i^s, \mathbf{y}_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t)) T_{ij} \right\} \tag{36}$$

- $\hat{\mathcal{P}^t}_f = \frac{1}{n_t} \sum_{i=1}^{N_t} \delta_{\mathbf{x}_i^t, f(\mathbf{x}_i^t)}$ 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 corresponds to minimizing a generalization bound.

Optimization problem

$$\min_{f\in\mathcal{H},\mathbf{T}\in\Pi} \quad \sum_{i,j} T_{i,j}\left(\alpha d(\mathbf{x}_i^s,\mathbf{x}_j^t)+\mathcal{L}(y_i^s,f(\mathbf{x}_j^t))\right)+\lambda\Omega(f) \tag{37}$$

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.

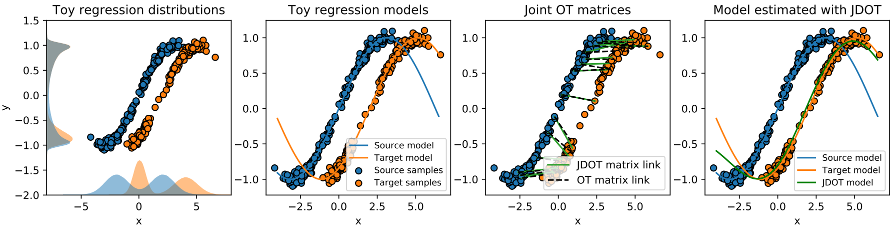
$\mathbf{T}$ update for a fixed f

- Classical OT problem.
- Solved by network simplex.
- Regularized OT can be used (add a term to problem (37))

f update for a fixed $\mathbf{T}$

$$\min_{f\in\mathcal{H}} \quad \sum_{i,j} T_{i,j}\mathcal{L}(y_i^s,f(\mathbf{x}_j^t))+\lambda\Omega(f) \tag{38}$$

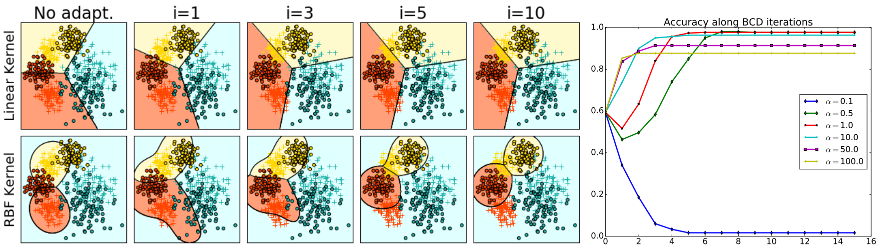
- Weighted loss from all source labels.
- $\mathbf{T}$ performs label propagation.



Least square regression with quadratic regularization
For a fixed $\mathbf{T}$ the optimization problem is equivalent to

$$\min_{f \in \mathcal{H}} \quad \sum_j \frac{1}{n_t} \|\hat{y}_j - f(\mathbf{x}_j^t)\|^2 + \lambda \|f\|^2 \tag{39}$$

- $\hat{y}_j = n_t \sum_j T_{i,j} y_i^s$ 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 $\mathbf{T}$ the optimization problem is equivalent to

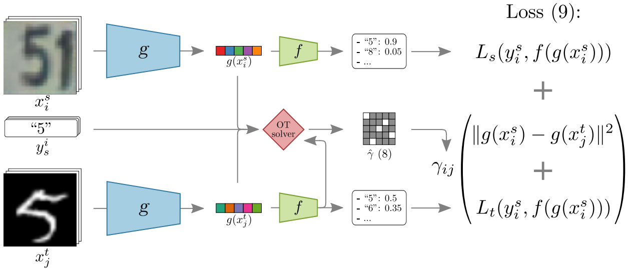
$$\min_{f_k \in \mathcal{H}} \quad \sum_{j,k} \hat{P}_{j,k}\mathcal{L}(1, f_k(\mathbf{x}_j^t)) + (1 - \hat{P}_{j,k})\mathcal{L}(-1, f_k(\mathbf{x}_j^t)) + \lambda \sum_k \|f_k\|^2 \qquad (40)$$

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

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

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

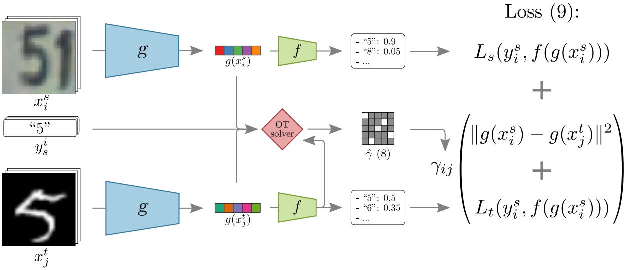
DeepJDOT



$$\min_{\mathbf{T}\in\Pi, f, g} \quad \frac{1}{n^s}\sum_i L_s\left(y_i^s, f(g(x_i^s))\right) + \sum_{i,j} T_{ij}\left(\alpha\|g(x_i^s) - g(x_j^t)\|^2 + \lambda_t \mathcal{L}\left(y_i^s, f(g(x_j^t))\right)\right). \tag{41}$$

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 [Fatras et al., 2020] .
- Scales to large datasets and estimates a representation for both domains.



$$\min_{f,g} \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\mathcal{L}\left(y_i^s, f(g(x_i^s)\right) + \min_{\mathbf{T}\in\Pi}\sum_{i,j}^{m} T_{ij}\left(\alpha\|g(x_i^s) - g(x_j^t)\|^2 + \lambda_t\mathcal{L}\left(y_i^s, f(g(x_j^t)\right)\right)\right] \tag{41}$$

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 [Fatras et al., 2020] .
- Scales to large datasets and estimates a representation for both domains.

DeepJDOT [Damodaran et al., 2018]

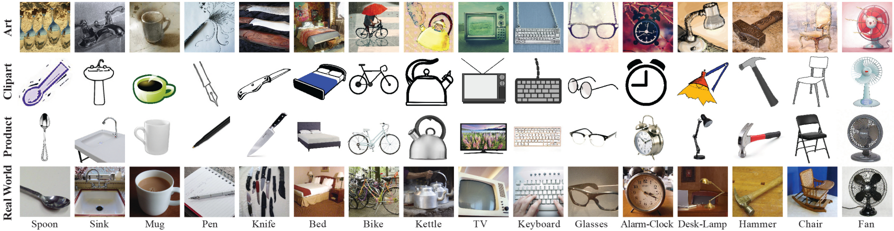
- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are important.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).

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

DeepJDOT [Damodaran et al., 2018]

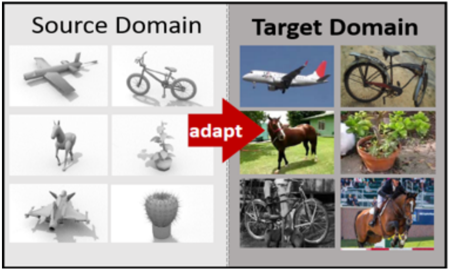
- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are important.
- TSNE projections of embeddings (MNIST→MNIST-M).

DeepJDOT in action



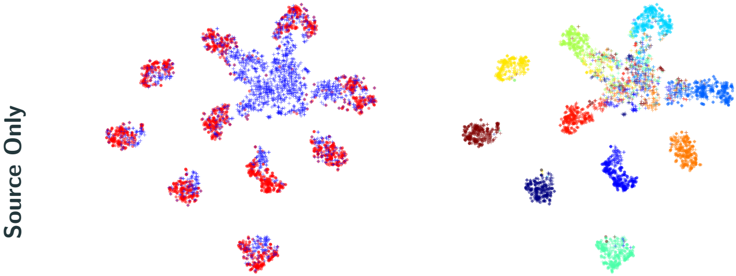
DeepJDOT [Damodaran et al., 2018]

- 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).



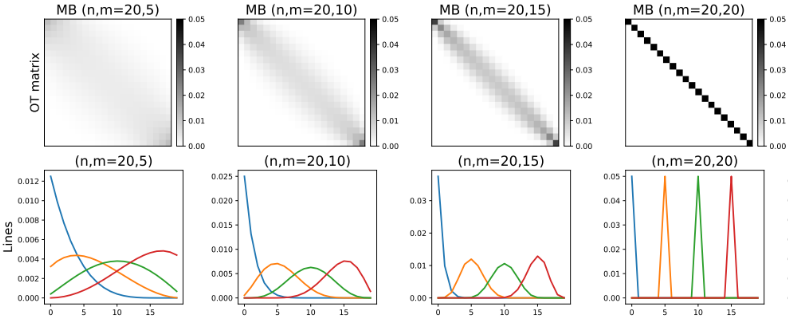
DeepJDOT [Damodaran et al., 2018]

- 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$\rightarrow$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$\rightarrow$MNIST-M).

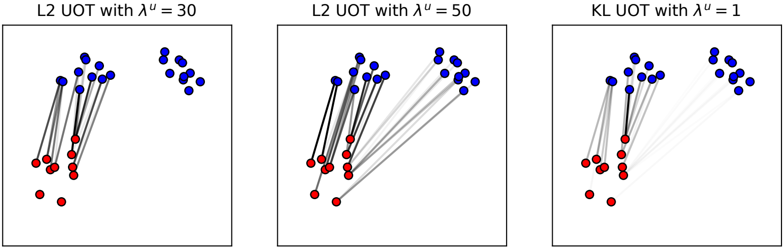
Minibatch Optimal Transport



Principle [Fatras et al., 2020]

$$MBOT_m(\mathcal{P}^s_{\mathcal{X}}, \mathcal{P}^t_{\mathcal{X}}) = E_{\hat{\mathcal{P}}^s_{\mathcal{X}} \sim \mathcal{P}^{s \otimes m}_{\mathcal{X}}, \hat{\mathcal{P}}^t_{\mathcal{X}} \sim \mathcal{P}^{t \otimes m}_{\mathcal{X}}}[W(\hat{\mathcal{P}}^s_{\mathcal{X}}, \hat{\mathcal{P}}^t_{\mathcal{X}})] \tag{42}$$

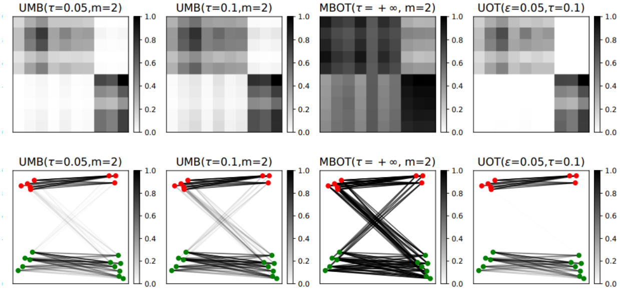
- Optimizing Wasserstein is numerically complex on large distributions.
- Numerous papers have been optimizing over minibatches [Genevay et al., 2017].
- $MBOT$ is biased ($MBOT_m(\mathcal{P}^s_{\mathcal{X}}, \mathcal{P}^s_{\mathcal{X}}) > 0$) but is actually a U-statistic and has nice convergence property (convergence in $O(n^{1/2})$).
- But the equivalent expected OT plan is dense and can be far from exct OT plan.



Unbalanced Optimal transport (UOT) [Benamou, 2003]

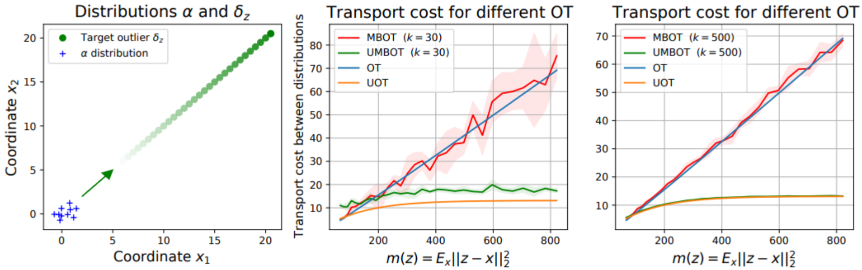
$$\min_{\mathbf{T}\geq 0} \quad \langle \mathbf{T}, \mathbf{C} \rangle_F + \lambda^u D_\varphi(\mathbf{T}\mathbf{1}_m, \mathbf{a}) + \lambda^u D_\varphi(\mathbf{T}^\top \mathbf{1}_n, \mathbf{b}) \tag{43}$$

- D_φ is a a Bregman divergence penalizing the violation of the marginal constraints.
- Only a portion of the total mass is transported, total mass can be unbalanced between source and target due to constraint relaxation.
- Closed form exists between Gaussians [Janati et al., 2020, Janati, 2021].
- Sinkhorn for regularized UOT [Chizat et al., 2018, Séjourné et al., 2019].
- UOT can be reformulated as a weighted Lasso regression (with data fitting D_φ) [Chapel et al., 2021].



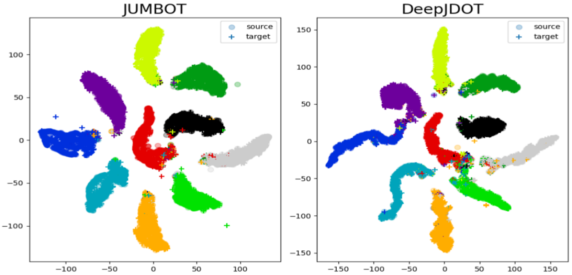
JUMBOT [Fatras et al., 2021]

- Main idea : DeepJDOT with minibatches and Unbalanced OT.
- Theoretical proof of robustness to outliers (UOT is upper bounded, not OT).
- Experiemnt on Partial DA (some classes are not in target) show robustness to different class proportions between domains.
- Better ability to handle samping noise on minibatch because good performance on small minibtach size.



JUMBOT [Fatras et al., 2021]

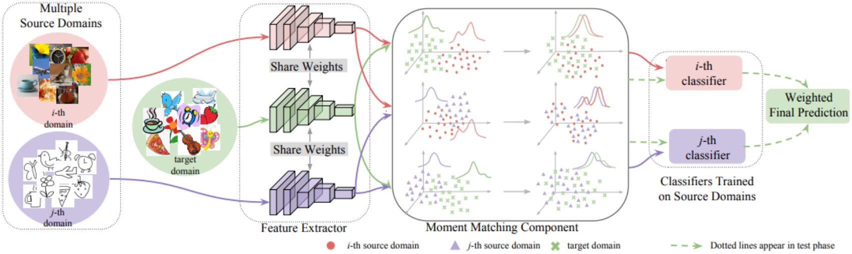
- Main idea : DeepJDOT with minibatches and Unbalanced OT.
- Theoretical proof of robustness to outliers (UOT is upper bounded, not OT).
- Experiemnt on Partial DA (some classes are not in target) show robustness to different class proportions between domains.
- Better ability to handle samping noise on minibatch because good performance on small minibtach size.



JUMBOT [Fatras et al., 2021]

- Main idea : DeepJDOT with minibatches and Unbalanced OT.
- Theoretical proof of robustness to outliers (UOT is upper bounded, not OT).
- Experiemnt on Partial DA (some classes are not in target) show robustness to different class proportions between domains.
- Better ability to handle samping noise on minibatch because good performance on small minibtach size.

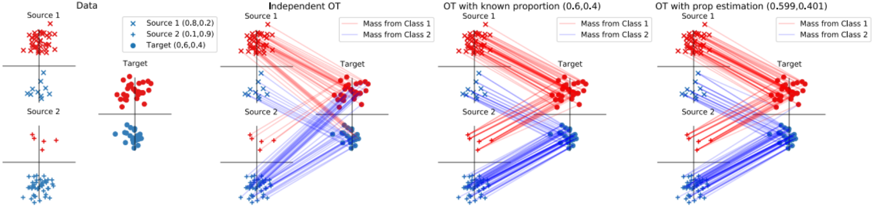
Domain Adaptation variants



Existing approaches

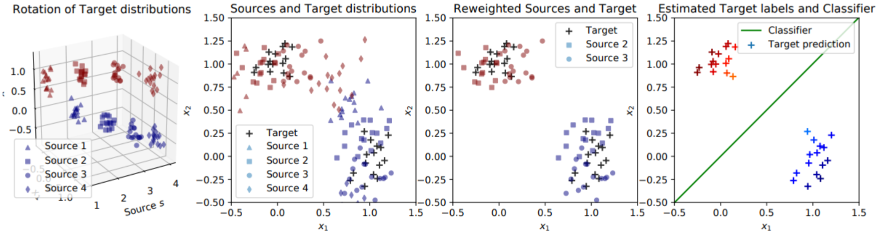
- Domain-Invariant Component Analysis (DICA) using kernel methods [Muandet et al., 2013].
- Moment Matching for Multi-source DA (M^3SDA) [Peng et al., 2019] estimates invariant representation and then perform weighting of source classifier.
- Wasserstein Barycenter Transport (WBT) [Montesuma and Mboula, 2021] computes Wasserstein barycenter of source domains and then performs OTDA.

Joint Class Proportion and OT estimation (JCPOT)



Principle [Redko et al., 2019a]

- Under target shift, source domains and target have different class proportions.
- JCPOT : Estimate the target class proportion by minimizing the sum of the Wasserstein distance of the class reweighted sources to the target.
- This estimation can be reformulated as a special case of Wasserstein barycenter.
- When target proportion are estimated perform OTDA using mapping or label propagation.

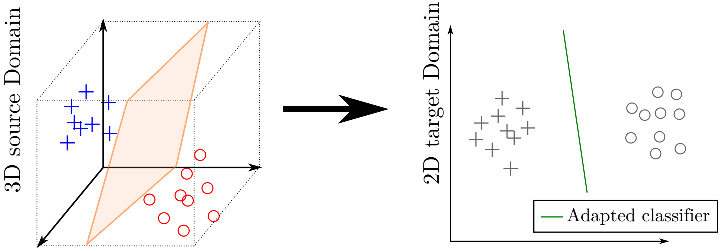


Principle [Turrisi et al., 2022]

$$\min_{\boldsymbol{\alpha}\in\Delta_D, f} \quad W_1\left(\sum_{k=1}^{D}\alpha_k\widehat{\mathcal{P}}_k^s, \widehat{\mathcal{P}}_f^t\right) \tag{44}$$

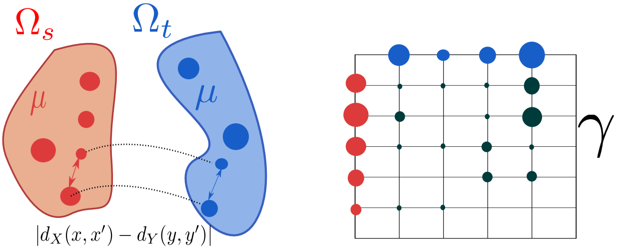
- Perform JDOT with a weighted sum of source domains.
- Optimize the weights $\boldsymbol{\alpha}$ on the simplex to minimize the JDOT loss.
- The weights will do automatically a selection of the source domains that are relevant to the task (as in close wrt the W_1).
- Generalization bound taking into account the number of samples per source domains and estimation of $\boldsymbol{\alpha}$.

Heterogeneous DA (HDA)



Existing methods

- Subspace projection then mapping estimation and SVM [Duan et al., 2012].
- Manifold alignment between domains [Wang and Mahadevan, 2011].
- Estimation of linear mapping between domains [Zhou et al., 2014].
- Mappoing using Optimal Transport across spaeces [Yan et al., 2018]



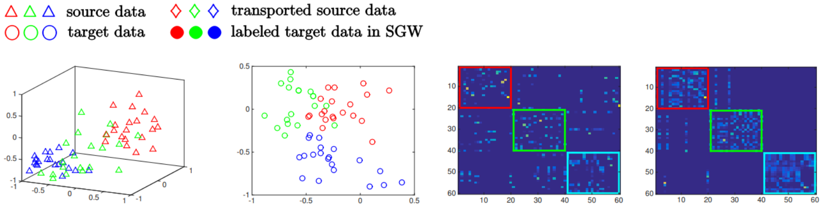
Inspired from Gabriel Peyré

GW for discrete distributions [Mémoli, 2011]

$$\mathcal{GW}_p(\mu_s, \mu_t) = \left(\min_{T \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} |D_{i,k} - D'_{j,l}|^p T_{i,j} T_{k,l} \right)^{\frac{1}{p}}$$

with $\mu_s = \sum_i a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_j b_j \delta_{x_j^t}$ and $D_{i,k} = \|\mathbf{x}_i^s - \mathbf{x}_k^s\|$, $D'_{j,l} = \|\mathbf{x}_j^t - \mathbf{x}_l^t\|$

- Distance between metric measured spaces : across different spaces.
- Search for an OT plan that preserve the pairwise relationships between samples.
- Invariant to isometry in either spaces (e.g. rotations and translation).



(a) source data (b) target data (c) **T** obtained by EGW (e) **T** obtained by SGW

Semi-supervised Heterogeneous Domain Adaptation [Yan et al., 2018]

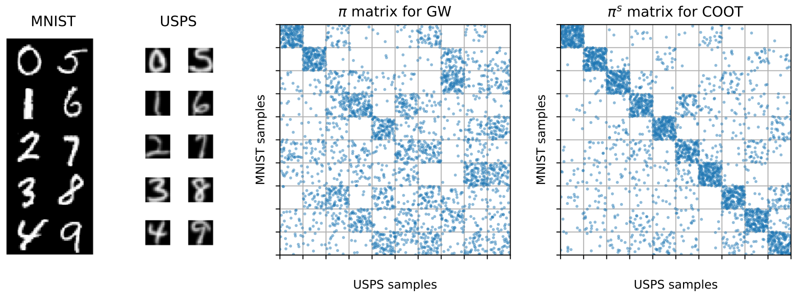
- Extension of OTDA [Courty et al., 2016] with GW.
- Use the OT matrix to transfer labels or samples between datasets.
- GW find correspondences across spaces but very noisy.
- Semi-supervised strategy allows very good performances.

Principle [Redko et al., 2020b]

$$\text{COOT}(\mathbf{X}, \mathbf{X}', \mathbf{w}, \mathbf{w}', \mathbf{v}, \mathbf{v}') = \min_{\substack{\boldsymbol{T}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{T}^v \in \Pi(\mathbf{v}, \mathbf{v}')}} \sum_{i,j,k,l} L(X_{i,k}, X'_{j,l}) \boldsymbol{T}^s_{i,j} \boldsymbol{T}^v_{k,l} \tag{45}$$

- $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$ and $\mathbf{X}' = [\mathbf{x}'_1, \ldots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$ contains the source and target data.
- $\mathbf{w} \in \Delta_n$ and $\mathbf{w}' \in \Delta_{n'}$ contain the weights of the samples in source and target datasets.
- $\mathbf{v} \in \Delta_d$ and $\mathbf{v}' \in \Delta_{d'}$ contain the weights of the features in source and target datasets.
- $L(\cdot, \cdot) : \mathbb{R}^2 \to \mathbb{R}^+$ is the similarity measure.
- $\mathbf{T}^s$ is the OT matrix between samples, $\mathbf{T}^v$ is the OT matrix between features/variables.
- COOT entropic regularized version adds some entropic terms to the objective value.

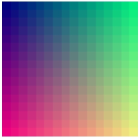
Illustration of COOT on real data



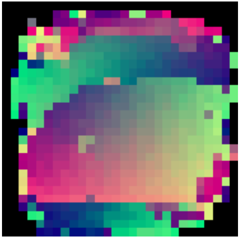
COOT between MNIST-USPS datasets

- Sample digits from MNIST 28×28 and USPS 16×16 ordered per classes.
- Uniform weights $\mathbf{w}, \mathbf{w}'$ on samples, weights $\mathbf{v}, \mathbf{v}'$ on feature is average value.
- Comparison between $\mathbf{T}$ from Gromov Wasserstein and COOT $\mathbf{T}^s$: better class correspondence.
- Visualization of $\mathbf{T}^s$ with colors across pixels: spatial structure preserved.
- Other application: finding correspondances between neurons in different architecture (adapt between embeddings: HDA).

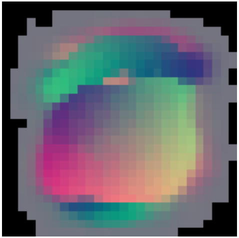
USPS colored pixels



MNIST pixels through π^v



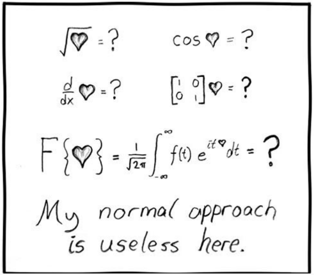
MNIST pixels through entropic π^v



COOT between MNIST-USPS datasets

- Sample digits from MNIST 28×28 and USPS 16×16 ordered per classes.
- Uniform weights $\mathbf{w}, \mathbf{w}'$ on samples, weights $\mathbf{v}, \mathbf{v}'$ on feature is average value.
- Comparison between $\mathbf{T}$ from Gromov Wasserstein and COOT $\mathbf{T}^s$: better class correspondence.
- Visualization of $\mathbf{T}^s$ with colors across pixels: spatial structure preserved.
- Other application: finding correspondances between neurons in different architecture (adapt between embeddings: HDA).

Domain Adaptation in Practice



Main practical problem

- No target labels are available.
- My usual validation procedure is useless here...
- And yet DA methods have parameters to choose.

What (some) people do?

- Maximize performance on target (very bad, more complex=more better)
- Validate on a few target labels (unrealistic).
- Use proxy on DA performance measure and validate (realistic, but rare).
- On datasets with multiple domains, validate params on one pair, and fix the params on all other pairs (unrealistic, ok for research, guilty).

Principle [Bruzzone and Marconcini, 2010]

1. Perform DA from source to target and learn $\hat{f}^t$.
2. Predict labels on target with f^t and perform DA from target to source.
3. Measure performance as the accuracy after the two DA steps.

Discussion

- Meaningful proxy for DA performance but be careful of some fails (e.g. OT).
- Better when using independent datasets for each DA so date needs to be split : validation done on smaller datasets.
- Works better on shallow methods (traditional CV).
- For deep learning, hard to use and does not help with early stopping.

Principle [Sugiyama et al., 2007]

$$\hat{\mathcal{R}}^{K}_{\mathcal{P}^t} = \sum_{k=1}^{K} \frac{1}{|\mathcal{T}_k|} \sum_{\mathbf{x},y\in\mathcal{T}_k} \hat{w}(\mathbf{x}) L(y, \hat{f}_k(\mathbf{x})) \tag{46}$$

where $\mathcal{T}_k$ defines a K partition of the source data and $\hat{f}_k$ is estimated on the complementary set.

- Can be used for any methods (especially shallow).
- Requires the estimation of the ratio $w(\mathbf{x}) = \frac{\hat{P}^t_{\mathcal{X}}(\mathbf{x})}{\hat{P}^s_{\mathcal{X}}(\mathbf{x})}$.
- Theoretically grounded for Covariate Shift.

Deep learning extension: Deep Embedded Validation (DEV) [You et al., 2019]

- IWCV where the reweighing is estimated with a source/target classifier in the embedding using approach from [Bickel et al., 2007].
- Variance reduction by control variate [Lemieux, 2014].

Unsupervised Domain Adaptation: A Reality Check

Kevin Musgrave
Cornell Tech

Serge Belongie
University of Copenhagen

Ser-Nam Lim
Meta AI

Paper : [Musgrave et al., 2021] [7]

- Meta Analysis from papers: Performance gain, Validation procedure.
- 30 out of 35 papers use target labels for validation.
- 10-20% performance gap between reported performance and performance with realistic validation.
- Comparison only on computer vision datasets.

---

[7] Musgrave, K., Belongie, S., and Lim, S.-N. (2021). Unsupervised domain adaptation: A reality check. **arXiv preprint arXiv:2111.15672**

Algorithm	Highlight
Adversarial	
DANN [7]	Gradient reversal layer
DC [42]	Uniform distribution loss
ADDA [43]	Frozen source model
CDAN [17]	Randomized dot product for combining multiple features
VADA [37]	Virtual adversarial training
Feature distance losses	
MMD [16]	Maximum mean discrepancy
CORAL [40]	Covariance matrix alignment
JMMD [19]	Joint MMD on multiple features
Maximum classifier discrepancy	
MCD [33]	Discrepancy = L1 distance
SWD [13]	Discrepancy = sliced wasserstein
STAR [20]	Stochastic classifier layer
Information maximization	
ITL [36]	Maximize info of class predictions, minimize info of domain predictions.
MCC [11]	Minimize class confusion via class correlations and entropy weighting
SENTRY [25]	Min or max entropy, based on pseudo label + augmentation consistency
SVD losses	
BSP [4]	Minimize singular values of features
BNM [5]	Max the sum of SVs of predictions
Image generation	
DRCN [8]	Reconstruct target images
GTA [35]	Generate source-like images from both source and target features
Pseudo labeling	
ATDA [32]	Two source classifiers that create pseudo labels for the target classifier
ATDOC [15]	Pseudo labels from soft k-NN labels
Mixup augmentations	
DM-ADA [47]	Soft domain labels derived from image and feature domain mixup
DMRL [46]	Mixup using domain and class labels
Other	
RTN [18]	Residual connection between source and target logits
AFN [48]	Increase the L2 norm of features
DSBN [3]	Separate batchnorm layers for source and target domains
SymNets [51]	Various operations on the concatenation of source and target predictions
GVB [6]	Minimize L1 norm of bridge layers

Paper : [Musgrave et al., 2021]

- Meta Analysis from papers: Performance gain, Validation procedure.
- 30 out of 35 papers use target labels for validation.

	Office31		OfficeHome	
Year	Source-only	DANN	Source-only	DANN
2016	-	2.2	-	-
2017	12.5	1.2	-	4.0
2018	23.4	8.5	28.1	11.5
2019	25.3	12.4	29.3	15.4
2020	23.9	14.1	31.5	17.2
2021	26.5	15.7	32.5	20.3

Table 2. The largest average SOTA-baseline performance gap per year. For example, the 2021 OfficeHome/DANN value of 20.3 is the gap on the Product→Art task, which is the task with the largest average SOTA-DANN gap for that year. Performance gap is measured as the absolute difference in accuracy.

Validator	# Papers	# Matches	# Repos
full oracle	0	-	30
subset oracle	3	2	2
src accuracy	0	-	1
src accuracy + loss	2	0	0
consistency + oracle	0	-	1
target entropy	0	-	1
reverse validation	2	0	0
IWCV [39]	2	0	0
DEV	2	0	0

Table 3. Validation methods in papers vs code. Out of 49 papers, 35 come with official repos. Of these 35 papers, 11 mention the validator that is used, and 2 use the same validator in both code and paper. 5 of the 6 papers that claim to use reverse validation, IWCV, or DEV, actually use oracle, and 1 uses target entropy.

Paper : [Musgrave et al., 2021]

- Meta Analysis from papers: Performance gain, Validation procedure.
- 30 out of 35 papers use target labels for validation.
- 10-20% performance gap between reported performance and performance with realistic validation.
- Comparison only on computer vision datasets.

Office 31

	AD	AW	DA	DW	WA	WD	Avg
Source-only	78.3	77.4	69.3	91.3	73.2	98.1	81.3
ADDA	71.0	73.7	64.5	89.1	65.5	93.2	76.2
AFN	88.6	85.8	69.6	96.8	69.6	99.4	85.0
AFN-DANN	87.7	93.4	70.7	96.5	72.8	99.6	86.8
ATDOC	85.8	84.0	73.3	95.0	72.0	99.1	84.9
ATDOC-DANN	85.9	91.5	**74.5**	96.6	73.8	98.7	86.8
BNM	86.7	91.2	73.3	97.1	75.6	98.9	87.1
BNM-DANN	88.7	91.4	72.7	96.6	75.5	99.6	87.4
BSP	81.3	78.2	70.0	96.2	69.7	**99.8**	82.5
BSP-DANN	85.6	90.4	71.8	96.3	73.0	99.6	86.1
CDAN	82.2	90.8	72.0	95.7	72.1	99.2	85.3
CORAL	84.3	84.2	69.9	91.7	70.6	98.4	83.2
DANN	87.5	91.7	71.8	96.3	73.5	99.4	86.7
DANN-FL8	85.1	91.1	72.5	96.7	74.0	99.6	86.5
DC	82.7	87.3	71.4	95.6	71.0	99.4	84.6
GVB	88.1	89.3	74.1	94.9	74.5	98.2	86.5
IM	90.4	87.1	72.1	96.7	72.2	99.4	86.3
IM-DANN	88.6	91.1	71.6	96.4	74.8	**99.8**	87.1
ITL	89.4	88.8	72.7	96.5	72.7	99.1	86.5
JMMD	86.2	87.8	70.8	96.9	71.7	**99.8**	85.5
MCC	91.2	91.5	72.8	97.1	75.5	99.4	87.9
MCC-DANN	**93.1**	**93.8**	73.2	96.7	**76.1**	99.4	**88.7**
MCD	86.6	86.5	68.2	96.8	69.1	98.7	84.3
MMD	85.8	86.0	71.1	96.1	71.7	99.6	85.1
MinEnt	85.2	88.5	72.5	96.8	72.9	98.7	85.8
RTN	85.7	87.0	72.0	**97.6**	72.1	98.8	85.5
STAR	78.4	77.4	60.6	95.9	63.6	98.5	79.1
SWD	80.9	79.0	68.9	96.4	68.3	97.9	81.9
SymNets	83.4	84.8	64.5	95.8	70.4	99.6	83.1
VADA	88.1	88.6	71.1	96.5	70.0	98.7	85.5

Paper : [Musgrave et al., 2021]

- Meta Analysis from papers: Performance gain, Validation procedure.
- 30 out of 35 papers use target labels for validation.
- 10-20% performance gap between reported performance and performance with realistic validation.
- Comparison only on computer vision datasets.

Unsupervised DA : a reality check

	Model	Office31	OfficeHome
Reported	Source-only	26.5	32.5
	DANN	15.7	20.3
Ours	Source-only	16.4	12.7
	DANN	5.6	7.9
	DANN-FL8	8.0	5.3

Table 8. Average reported performance gap in 2021 papers vs ours. Each number corresponds with the transfer task with the largest performance gap.

Paper : [Musgrave et al., 2021]

- Meta Analysis from papers: Performance gain, Validation procedure.
- 30 out of 35 papers use target labels for validation.
- 10-20% performance gap between reported performance and performance with realistic validation.
- Comparison only on computer vision datasets.

Domain Adaptation

- Dataset Shift in ML [Quionero-Candela et al., 2009].
- Types of data shift [Moreno-Torres et al., 2012].
- Recent very good review in [Kouw and Loog, 2019] (200 refs!).

Theory of DA

- Seminal works in [Ben-david et al., 2006].
- Recent survey of DA theory in [Redko et al., 2020a] and in the book [Redko et al., 2019b] from the same authors.

DA and deep learning

- Survey of DA with visual applications [Csurka, 2017].
- A survey of unsupervised domain adaptation [Wilson and Cook, 2020].

Practical DA

- IWCV [Sugiyama et al., 2007] and DEV [You et al., 2019] validation scores.
- DA reality check [Musgrave et al., 2021].

Time machine (Peyresq 2010)

[Aljundi et al., 2015] Aljundi, R., Emonet, R., Muselet, D., and Sebban, M. (2015). **Landmarks-based kernelized subspace alignment for unsupervised domain adaptation.** In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 56–63.

[Arjovsky et al., 2017] Arjovsky, M., Chintala, S., and Bottou, L. (2017). **Wasserstein gan.** *arXiv preprint arXiv:1701.07875*.

[Baktashmotlagh et al., 2013] Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. (2013). **Unsupervised domain adaptation by domain invariant projection.** In *Proceedings of the IEEE international conference on computer vision*, pages 769–776.

[Balcan et al., 2008] Balcan, M.-F., Blum, A., and Srebro, N. (2008). **A theory of learning with similarity functions.** *Machine Learning*, 72(1-2):89–112.

[Ben-david et al., 2006] Ben-david, S., Blitzer, J., Crammer, K., and Pereira, O. (2006). **Analysis of representations for domain adaptation.** In *Neural Information Processing Systems (NIPS)*. MIT Press.

[Ben-David et al., 2010] Ben-David, S., Luu, T., Lu, T., and Pál, D. (2010). **Impossibility theorems for domain adaptation.** In *Artificial Intelligence and Statistics Conference (AISTATS)*, pages 129–136.

[Ben-David and Urner, 2012] Ben-David, S. and Urner, R. (2012). **On the hardness of domain adaptation and the utility of unlabeled target samples.** In *International Conference on Algorithmic Learning Theory*, pages 139–153. Springer.

[Benaim and Wolf, 2017] Benaim, S. and Wolf, L. (2017). **One-sided unsupervised domain mapping.** *Advances in neural information processing systems*, 30.

[Benamou, 2003] Benamou, J.-D. (2003). **Numerical resolution of an "unbalanced" mass transport problem.** *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(5):851–868.

[Bickel et al., 2007] Bickel, S., Brückner, M., and Scheffer, T. (2007). **Discriminative learning for differing training and test distributions.** In *Proceedings of the 24th international conference on Machine learning*, pages 81–88.

[Bousmalis et al., 2016] Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. (2016). **Domain separation networks.** *Advances in neural information processing systems*, 29.

[Bruzzone and Marconcini, 2010] Bruzzone, L. and Marconcini, M. (2010). **Domain adaptation problems: A dasvm classification technique and a circular validation strategy.** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):770–787.

[Chapel et al., 2021] Chapel, L., Flamary, R., Wu, H., Févotte, C., and Gasso, G. (2021). **Unbalanced optimal transport through non-negative penalized linear regression.** In *Neural Information Processing Systems (NeurIPS)*.

[Chapelle et al., 2006] Chapelle, O., Scholkopf, B., and Zien, A. (2006). **Semi-supervised learning. 2006.** *Cambridge, Massachusettes: The MIT Press View Article*, 2.

[Chen et al., 2011] Chen, M., Weinberger, K. Q., and Blitzer, J. (2011). **Co-training for domain adaptation.** *Advances in neural information processing systems*, 24.

[Chizat et al., 2018] Chizat, L., Peyré, G., Schmitzer, B., and Vialard, F.-X. (2018). **Unbalanced optimal transport: Dynamic and kantorovich formulations.** *Journal of Functional Analysis*, 274(11):3090–3123.

[Cortes et al., 2010] Cortes, C., Mansour, Y., and Mohri, M. (2010). **Learning bounds for importance weighting.** *Advances in neural information processing systems*, 23.

[Cortes and Mohri, 2011] Cortes, C. and Mohri, M. (2011). **Domain adaptation in regression.** In *International Conference on Algorithmic Learning Theory*, pages 308–323. Springer.

[Cortes et al., 2015] Cortes, C., Mohri, M., and Muñoz Medina, A. (2015). **Adaptation algorithm and theory based on generalized discrepancy.** In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 169–178.

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

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

[Csurka, 2017] Csurka, G. (2017). **A comprehensive survey on domain adaptation for visual applications.** *Domain adaptation in computer vision applications*, pages 1–35.

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

[Damodaran et al., 2018] Damodaran, B., Kellenberger, B., Flamary, R., Tuia, D., and Courty, N. (2018). **Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation.** In *European Conference in Computer Vision (ECCV)*.

[Donahue et al., 2014] 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 *International Conference on Machine Learning (ICML)*, pages 647–655.

[Duan et al., 2012] Duan, L., Xu, D., and Tsang, I. (2012). **Learning with augmented features for heterogeneous domain adaptation.** *arXiv preprint arXiv:1206.4660*.

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

[Fatras et al., 2020] Fatras, K., Zine, Y., Flamary, R., Gribonval, R., and Courty, N. (2020). **Learning with minibatch wasserstein : asymptotic and gradient properties.** In *International Conference on Artificial Intelligence and Statistics (AISTAT)*.

[Fernando et al., 2013] Fernando, B., Habrard, A., Sebban, M., and Tuytelaars, T. (2013). **Unsupervised visual domain adaptation using subspace alignment.** In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967.

[Fernando et al., 2015] Fernando, B., Tommasi, T., and Tuytelaars, T. (2015). **Joint cross-domain classification and subspace learning for unsupervised adaptation.** *Pattern Recognition Letters*, 65:60–66.

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

[Flamary et al., 2021] Flamary, R., Lounici, K., and Ferrari, A. (2021). **Concentration bounds for linear monge mapping estimation and optimal transport domain adaptation.**

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

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

[Germain et al., 2013] Germain, P., Habrard, A., Laviolette, F., and Morvant, E. (2013). **A pac-bayesian approach for domain adaptation with specialization to linear classifiers.** In *International conference on machine learning*, pages 738–746. PMLR.

[Germain et al., 2016] Germain, P., Habrard, A., Laviolette, F., and Morvant, E. (2016). **A new pac-bayesian perspective on domain adaptation.** In *International conference on machine learning*, pages 859–868. PMLR.

[Gong et al., 2012] Gong, B., Shi, Y., Sha, F., and Grauman, K. (2012). **Geodesic flow kernel for unsupervised domain adaptation.** In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2066–2073.

[Gopalan et al., 2011] Gopalan, R., Li, R., and Chellappa, R. (2011). **Domain adaptation for object recognition: An unsupervised approach.** In *International Conference on Computer Vision (ICCV)*, pages 999–1006.

[Gopalan et al., 2013] Gopalan, R., Li, R., and Chellappa, R. (2013). **Unsupervised adaptation across domain shifts by generating intermediate data representations.** *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2288–2302.

[Gopalan et al., 2014] Gopalan, R., Li, R., 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.

[Grandvalet and Bengio, 2004] Grandvalet, Y. and Bengio, Y. (2004). **Semi-supervised learning by entropy minimization.** *Advances in neural information processing systems*, 17.

[Gretton et al., 2009] Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. (2009). **Covariate shift by kernel mean matching.** *Dataset shift in machine learning*, 3(4):5.

[Gulrajani et al., 2017] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. (2017). **Improved training of wasserstein gans.** *NIPS*.

[Habrard et al., 2013] Habrard, A., Peyrache, J.-P., and Sebban, M. (2013). **Iterative self-labeling domain adaptation for linear structured image classification.** *International Journal on Artificial Intelligence Tools*, 22(05):1360005.

[He and Niyogi, 2003] He, X. and Niyogi, P. (2003). **Locality preserving projections.** *Advances in neural information processing systems*, 16.

[Hoffman et al., 2018] Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. (2018). **Cycada: Cycle-consistent adversarial domain adaptation.** In *International conference on machine learning*, pages 1989–1998. PMLR.

[Hu et al., 2018] Hu, W., Niu, G., Sato, I., and Sugiyama, M. (2018). **Does distributionally robust supervised learning give robust classifiers?** In *International Conference on Machine Learning*, pages 2029–2037. PMLR.

[Huang et al., 2006] Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. (2006). **Correcting sample selection bias by unlabeled data.** *Advances in neural information processing systems*, 19.

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

[Janati, 2021] Janati, H. (2021). **Advances in Optimal transport and applications to neuroscience.** PhD thesis, Institut Polytechnique de Paris.

[Janati et al., 2020] Janati, H., Muzellec, B., Peyré, G., and Cuturi, M. (2020). **Entropic optimal transport between unbalanced gaussian measures has a closed form.** *Advances in Neural Information Processing Systems*, 33.

[Kanamori et al., 2009] Kanamori, T., Hido, S., and Sugiyama, M. (2009). **A least-squares approach to direct importance estimation.** *The Journal of Machine Learning Research*, 10:1391–1445.

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

[Korotin et al., 2019] Korotin, A., Egiazarian, V., Asadulaev, A., Safin, A., and Burnaev, E. (2019). **Wasserstein-2 generative networks.** *arXiv preprint arXiv:1909.13082*.

[Kouw and Loog, 2019] Kouw, W. M. and Loog, M. (2019). **A review of domain adaptation without target labels.** *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785.

[Kremer et al., 2015] Kremer, J., Gieseke, F., Pedersen, K. S., and Igel, C. (2015). **Nearest neighbor density ratio estimation for large-scale applications in astronomy.** *Astronomy and Computing*, 12:67–72.

[Kuhn et al., 2019] Kuhn, D., Esfahani, P. M., Nguyen, V. A., and Shafieezadeh-Abadeh, S. (2019). **Wasserstein distributionally robust optimization: Theory and applications in machine learning.** In *Operations Research & Management Science in the Age of Analytics*, pages 130–166. INFORMS.

[Lemieux, 2014] Lemieux, C. (2014). **Control variates.** *Wiley StatsRef: Statistics Reference Online*, pages 1–8.

[Lipton et al., 2018] Lipton, Z., Wang, Y.-X., and Smola, A. (2018). **Detecting and correcting for label shift with black box predictors.** In *International conference on machine learning*, pages 3122–3130. PMLR.

[Liu and Ziebart, 2014] Liu, A. and Ziebart, B. (2014). **Robust classification under sample selection bias.** *Advances in neural information processing systems*, 27.

[Long et al., 2015] Long, M., Cao, Y., Wang, J., and Jordan, M. (2015). **Learning transferable features with deep adaptation networks.** In *International conference on machine learning*, pages 97–105. PMLR.

[Long et al., 2013] Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. (2013). **Transfer feature learning with joint distribution adaptation.** In *International Conference on Computer Vision (ICCV)*, pages 2200–2207.

[Long et al., 2014] Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. (2014). **Transfer joint matching for unsupervised domain adaptation.** In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1410–1417.

[Long et al., 2017] Long, M., Zhu, H., Wang, J., and Jordan, M. I. (2017). **Deep transfer learning with joint adaptation networks.** In *International conference on machine learning*, pages 2208–2217. PMLR.

[Loog, 2012] Loog, M. (2012). **Nearest neighbor-based importance weighting.** In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE.

[Mansour et al., 2009] Mansour, Y., Mohri, M., and Rostamizadeh, A. (2009). **Domain adaptation: Learning bounds and algorithms.** In *Conference on Learning Theory (COLT)*, pages 19–30.

[Mémoli, 2011] Mémoli, F. (2011). **Gromov-Wasserstein distances and the metric approach to object matching.** *Foundations of Computational Mathematics*, pages 1–71.

[Mendelson, 2002] 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.

[Miyato et al., 2018] Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. (2018). **Virtual adversarial training: a regularization method for supervised and semi-supervised learning.** *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

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

[Montesuma and Mboula, 2021] Montesuma, E. F. and Mboula, F. M. N. (2021). **Wasserstein barycenter for multi-source domain adaptation.** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16785–16793.

[Moreno-Torres et al., 2012] Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., and Herrera, F. (2012). **A unifying view on dataset shift in classification.** *Pattern recognition*, 45(1):521–530.

[Mroueh, 2019] Mroueh, Y. (2019). **Wasserstein style transfer.** *arXiv preprint arXiv:1905.12828*.

[Muandet et al., 2013] Muandet, K., Balduzzi, D., and Schölkopf, B. (2013). **Domain generalization via invariant feature representation.** In *International Conference on Machine Learning*, pages 10–18. PMLR.

[Musgrave et al., 2021] Musgrave, K., Belongie, S., and Lim, S.-N. (2021). **Unsupervised domain adaptation: A reality check.** *arXiv preprint arXiv:2111.15672*.

[Pan et al., 2010] Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. (2010). **Domain adaptation via transfer component analysis.** *IEEE transactions on neural networks*, 22(2):199–210.

[Peng et al., 2019] Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. (2019). **Moment matching for multi-source domain adaptation.** In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415.

[Peng et al., 2017] 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*.

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

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

[Pooladian and Niles-Weed, 2021] Pooladian, A.-A. and Niles-Weed, J. (2021). **Entropic estimation of optimal transport maps.** *arXiv preprint arXiv:2109.12004*.

[Quionero-Candela et al., 2009] Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2009). **Dataset shift in machine learning.** The MIT Press.

[Rakotomamonjy et al., 2022] Rakotomamonjy, A., Flamary, R., Gasso, G., Alaya, M. E., Berar, M., and Courty, N. (2022). **Optimal transport for conditional domain matching and label shift.** *Machine Learning*, 111(5):1651–1670.

[Redko, 2015] Redko, I. (2015). **Nonnegative matrix factorization for unsupervised transfer learning.** PhD thesis, PhD thesis, Paris North University.

[Redko et al., 2019a] Redko, I., Courty, N., Flamary, R., and Tuia, D. (2019a). **Optimal transport for multi-source domain adaptation under target shift.** In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[Redko et al., 2017] Redko, I., Habrard, A., and Sebban, M. (2017). **Theoretical analysis of domain adaptation with optimal transport.** In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 737–753. Springer.

[Redko et al., 2019b] Redko, I., Morvant, E., Habrard, A., Sebban, M., and Bennani, Y. (2019b). **Advances in domain adaptation theory.** Elsevier.

[Redko et al., 2020a] Redko, I., Morvant, E., Habrard, A., Sebban, M., and Bennani, Y. (2020a). **A survey on domain adaptation theory.** *arXiv preprint arXiv:2004.11829*.

[Redko et al., 2020b] Redko, I., Vayer, T., Flamary, R., and Courty, N. (2020b). **Co-optimal transport.** In *Neural Information Processing Systems (NeurIPS)*.

[Rubner et al., 2000] 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 et al., 2010] Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010). **Adapting visual category models to new domains.** In *European Conference on Computer Vision (ECCV)*, LNCS, pages 213–226, Berlin, Heidelberg. Springer-Verlag.

[Schölkopf et al., 1997] Schölkopf, B., Smola, A., and Müller, K.-R. (1997). **Kernel principal component analysis.** In *International conference on artificial neural networks*, pages 583–588. Springer.

[Seguy et al., 2018] Seguy, V., Bhushan Damodaran, B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. (2018). **Large-scale optimal transport and mapping estimation.** In *International Conference on Leraning Representation (ICLR)*.

[Séjourné et al., 2019] Séjourné, T., Feydy, J., Vialard, F.-X., Trouvé, A., and Peyré, G. (2019). **Sinkhorn divergences for unbalanced optimal transport.** *arXiv preprint arXiv:1910.12958*.

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

[Shimodaira, 2000] Shimodaira, H. (2000). **Improving predictive inference under covariate shift by weighting the log-likelihood function.** *Journal of statistical planning and inference*, 90(2):227–244.

[Shu et al., 2018] Shu, R., Bui, H. H., Narui, H., and Ermon, S. (2018). **A dirt-t approach to unsupervised domain adaptation.** *arXiv preprint arXiv:1802.08735*.

[Si et al., 2010] 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 et al., 2007] Sugiyama, M., Krauledat, M., and Müller, K.-R. (2007). **Covariate shift adaptation by importance weighted cross validation.** *Journal of Machine Learning Research*, 8(5).

[Sugiyama and Müller, 2005] Sugiyama, M. and Müller, K.-R. (2005). **Input-dependent estimation of generalization error under covariate shift.** *Statistics & Decisions*, 23(4):249–279.

[Sugiyama et al., 2012] Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). **Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation.** *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044.

[Sugiyama et al., 2008] 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 and Saenko, 2015] Sun, B. and Saenko, K. (2015). **Subspace distribution alignment for unsupervised domain adaptation.** In *BMVC*, volume 4, pages 24–1.

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

[Thrun and Pratt, 2012] Thrun, S. and Pratt, L. (2012). **Learning to learn.** Springer Science & Business Media.

[Tuia et al., 2015] 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*.

[Turrisi et al., 2022] Turrisi, R., Flamary, R., Rakotomamonjy, A., and Pontil, M. (2022). **Multi-source domain adaptation via weighted joint distributions optimal transport.** In *Conference on Uncertainty in Artificial Intelligence (UAI)*.

[Tzeng et al., 2017] Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). **Adversarial discriminative domain adaptation.** In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176.

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

[Vapnik, 2006] Vapnik, V. (2006). **Estimation of dependences based on empirical data.** Springer Science & Business Media.

[Venkateswara et al., 2017] Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. (2017). **Deep hashing network for unsupervised domain adaptation.** In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027.

[Wang and Mahadevan, 2011] Wang, C. and Mahadevan, S. (2011). **Heterogeneous domain adaptation using manifold alignment.** In *IJCAI proceedings-international joint conference on artificial intelligence*, volume 22, page 1541.

[Wen et al., 2014] Wen, J., Yu, C.-N., and Greiner, R. (2014). **Robust learning under uncertain test distributions: Relating covariate shift to model misspecification.** In *International Conference on Machine Learning*, pages 631–639. PMLR.

[Wilson and Cook, 2020] Wilson, G. and Cook, D. J. (2020). **A survey of unsupervised deep domain adaptation.** *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46.

[Yan et al., 2018] Yan, Y., Li, W., Wu, H., Min, H., Tan, M., and Wu, Q. (2018). **Semi-supervised optimal transport for heterogeneous domain adaptation.** In *IJCAI*, volume 7, pages 2969–2975.

[You et al., 2019] You, K., Wang, X., Long, M., and Jordan, M. (2019). **Towards accurate model selection in deep unsupervised domain adaptation.** In *International Conference on Machine Learning*, pages 7124–7133. PMLR.

[Zhang et al., 2013] Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. (2013). **Domain adaptation under target and conditional shift.** In *International Conference on Machine Learning*, pages 819–827. PMLR.

[Zhou et al., 2014] Zhou, J. T., Tsang, I. W., Pan, S. J., and Tan, M. (2014). **Heterogeneous domain adaptation for multiple classes.** In *Artificial intelligence and statistics*, pages 1095–1103. PMLR.

[Zhou et al., 2021] Zhou, K., Liu, Z., Qiao, Y., Xiang, T., and Loy, C. C. (2021). **Domain generalization in vision: A survey.** *arXiv preprint arXiv:2103.02503*.