Gromov-Wasserstein Optimal Transport for Heterogeneous Domain Adaptation

Julien Malka
École Normale Supérieure
julien.malka@ens.fr

Supervised by

Rémi Flamary
Université Côte d'Azur
remi.flamary@unice.fr

Nicolas Courty
Université de Bretagne Sud
courty@univ-ubs.fr

Abstract

Optimal Transport distances have shown great potential these last year in tackling the homogeneous domain adaptation problem. This works present some adaptations of the state of the art homogeneous domain adaptations methods to work on heterogeneous domains.

1 Introduction

The classical supervised learning approach consist in estimating, from a set of samples alongside with their labels, their empirical joint distribution in order to predict real life unlabelled samples. More formally, let $\Omega_s \subset R^{d_s}$ be a measurable input set of dimension d_s, $P(\Omega_s)$ the set of probability distributions on Ω_s, and C the set of the possible labels. From a training set, $\{\mathbf{x}_i^s, y_i^s\}_{i=1}^N$ where $\{\mathbf{x}_i^s, y_i^s\} \in (\Omega_s \times C)$, one usually computes an empirical estimate of $P_s(\mathbf{x}^s, y)$, hoping to be then able to estimate labels of unknown examples.

Of course, this method only works if the real life samples are distributed the same way as the training samples. Unfortunately, this assumption is often unrealistic in practical applications. One field where this is particularly the case is computer vision: differences in the acquisition hardware (quality of image, internal post processing), the environmental factors (brightness, weather, background) can drastically alter the images. The domain adaptation task consist in learning in settings where this assumption is not true.

In this work, we will assume that additionally to the source domain Ω_s, we have a target domain $\Omega_t \subset R^{d_t}$ containing the samples we wish to predict. We also assume the existence of two distinct

joint probability distributions $P_s(\mathbf{x}^s, y)$ and $P_t(\mathbf{x}^t, y)$ on these domains. When the source and target domains have the same dimension, we call the task homogeneous domain adaptation whereas when it is not the case, we call it heterogeneous domain adaptation. The work focuses on the latter. In the last few years, the optimal transport theory has been used to solve the homogeneous domain adaptation, under the influence of several publications: see Courty et al. [2015], Courty et al. [2017], Damodaran et al. [2018].

This work largely focuses on adapting these methods to work in a heterogeneous context. We'll notably present some new methods using the Gromov-Wasserstein optimal transport problems.

Figure 1: Example of domain adaptation problem: on the left the training images, on the right the real life images.

2 Optimal Transport

2.1 Kantorovic formulation

The optimal transport problem is the one finding a transformation $\mathbf{T}$ moving the mass of a probability distribution μ_s onto an other μ_t, minimizing in the way a certain cost. If both the probability distributions are discrete, then we can write

$$\mu_s = \sum_{i=1}^{n} p_i^s \delta_{x_i^s} \quad \text{and} \quad \mu_t = \sum_{j=1}^{m} p_j^t \delta_{x_j^t} \tag{1}$$

where δx_i is the Dirac function at location $x_i \in R^d$ and $\sum_{i=1}^{n} p_i^s = \sum_{j=1}^{m} p_j^t = 1$
Let C be the matrix whose element C_{ij} is the cost to move a probability mass from x_i^s to x_j^t, we get the optimal transport problem formulation, called the Kantorovic formulation:

$$\mathbf{T} = \underset{\boldsymbol{\gamma} \in \mathcal{B}}{\operatorname{argmin}} \quad \langle \boldsymbol{\gamma}, \mathbf{C} \rangle_F \tag{2}$$

Where $\mathcal{B} = \left\{ \boldsymbol{\gamma} \in (\mathbb{R}^+)^{\mathbf{n} \times \mathbf{m}} | \, \boldsymbol{\gamma} \mathbf{1_n} = \mu_\mathbf{s}, \boldsymbol{\gamma}^\mathbf{T} \mathbf{1_m} = \mu_\mathbf{t} \right\}$ is the set of all couplings between μ_s and μ_t preserving mass and $\langle \cdot, \cdot \rangle_F$ is the Frobenius product.

If the cost matrix $\mathbf{C} = D^p$ for some $p \geq 1$, where D is a distance matrix, then it is proved in Villani [2008] that this defines a distance (the Wasserstein distance) between μ and ν.

$$W_p(\mu, \nu) = (\min_{\boldsymbol{\gamma} \in \mathcal{B}} \quad \langle \boldsymbol{\gamma}, \mathbf{D}^p \rangle_F)^{1/p} \tag{3}$$

In this case, we say that optimal transport is a canonical way to lift a distance on a metric space to a distance on the space of measures.
The Kantorovic problem in the discrete setting as described in this section is an optimisation problem that falls into the category of the linear programs and can be solved with well known combinatorial algorithms such as the network-simplex, that can find an optimal transformation in $O((n+m)nmlog(n+m))$. Since the work of Cuturi [2013], we also know an algorithm to compute and approximate solution of the Kantorovic problem (to be precise, a solution of the regularized Kantorovic problem) very efficiently, opening the door to applications in settings with large inputs, like in artificial intelligence. Were the reader wishing to know more about the Kantorovic optimal transport problem and the ways to solve it numerically, some good resources would be Villani [2008], Peyré and Cuturi [2018], Mondon and Malka [2020].

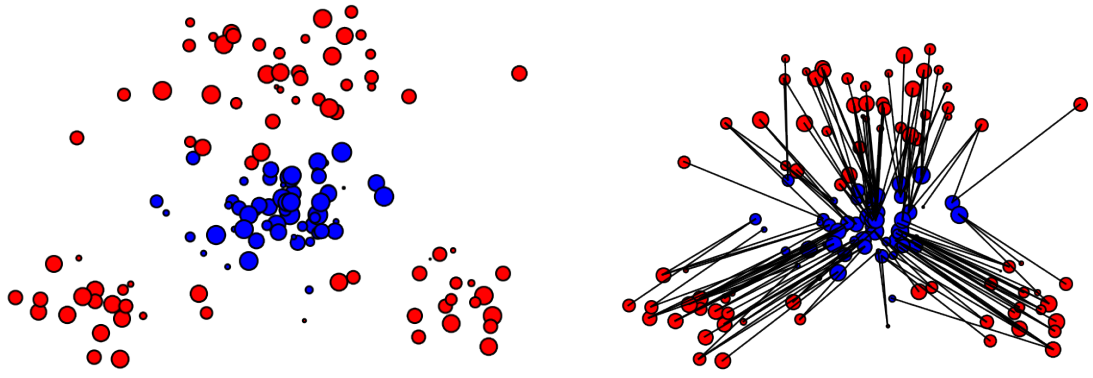


Figure 2: Example of an optimal coupling between two discrete distributions.

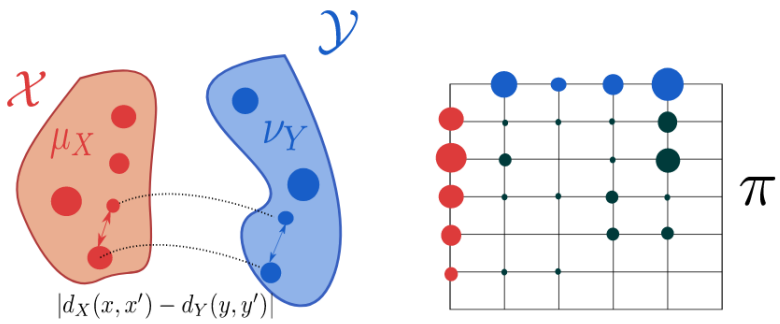


Figure 3: Gromov-Wasserstein coupling between two distributions μ_s and μ_t living in two different spaces. Figure from Titouan et al. [2019] page 9.

2.2 Gromov-Wasserstein

To have a correct formulation of the Kantorovic problem, we need the existence of the matrix C described in the section 2.1. This matrix may not exist, if for example μ_s and μ_t are not defined on the same space.

In that case, we assume the existence of $C \in R^{n \times n}$ and $\bar{C} \in R^{m \times m}$ two similarity matrix respectively in the two spaces. Formally, $C_{i,k}$ (respectively $\bar{C}_{i,k}$) is a measure of similarity between samples i and k. Intuitively, we will now map pairs of points together, and pairs of points that are similar (that is the distance within each pairs are close) are going to be matched. The Gromov-Wasserstein formulation of the optimal transport is then written as:

$$\begin{aligned} \mathbf{T} &= \underset{\boldsymbol{\gamma} \in \mathcal{B}}{\operatorname{argmin}} \quad \mathcal{E}_{C,\bar{C}}(\boldsymbol{\gamma}) \\ \mathcal{E}_{C,\bar{C}}(\boldsymbol{\gamma}) &= \sum_{i,j,k,l} |C_{i,k} - \bar{C}_{j,l}| \boldsymbol{\gamma}_{i,j} \boldsymbol{\gamma}_{k,l} \end{aligned} \tag{4}$$

As for the Kantorovic formulation, if C and $\bar{C}$ are (powers of) distances matrices, then this defines a distance on the set of metric-measure spaces quotiented by measure preserving isometries:

$$GW(C, \bar{C}, \mu, \nu) = \min_{\boldsymbol{\gamma} \in \mathcal{B}} \quad \mathcal{E}_{C,\bar{C}}(\boldsymbol{\gamma}) \tag{5}$$

This problem is a quadratic assignment problem but in the same fashion as for the Kantorovic problem, an approximate solution can be found in a reasonable time thanks to an appropriate relaxation (Solomon et al. [2016]).

2.3 Fused Gromov-Wasserstein

The optimal transport formulations mentioned in the previous sections fail at leveraging structural information of the data. In data science however, the structural information of the data is crucial

and not using it is a big waste. Formally we say that a structured object over a metric space (Ω, d) is the triplet $(X \times A, d_X, \mu)$ where (X, d_X) is a compact metric space, A is a compact of Ω, and μ is a fully supported probability measure over $X \times A$ (Vayer et al. [2018]). Simply put a dataset of samples alongside with their labels is a structured object as soon as we have a distance between the samples and a discrepancy measure between the labels.
Originally presented in Titouan et al. [2019] and studied in depth in Vayer et al. [2018], the Fused Gromov-Wasserstein is a convex combination of the Kantorovic and the Gromov-Wasserstein formulations of the optimal transport and consist in this optimisation problem:

$$\mathbf{T} = \underset{\boldsymbol{\gamma} \in \mathcal{B}}{\operatorname{argmin}} \sum_{i,j,k,l} ((1-\alpha)L(y_i^s, y_j^t) + \alpha |C_{i,k} - \bar{C}_{j,l}|) \boldsymbol{\gamma}_{i,j} \boldsymbol{\gamma}_{k,l} \tag{6}$$

Where L is a discrepancy function between the labels, and $\alpha \in [0, 1]$ is a tradeoff parameter between the structure term and the feature term.
Like in the precedent sections, the FGW problem induces a distance on the space of structured objects quotiented by the measure preserving maps that are feature and structure preserving.

3 Contribution

In all these situations, we will assume that the domain drift is caused by a (possibly nonlinear) transformation $\mathbf{T} : \Omega_s \mapsto \Omega_t$ that we wish to determine.

3.1 GW-OTDA

We propose a variation of Courty et al. [2015] suited for heterogeneous domains. We add one more hypothesis to the setup described in the introduction, that is the domain transformation $\mathbf{T} : \Omega_s \to \Omega_t$ we are looking for preserves the conditional distribution, formally:

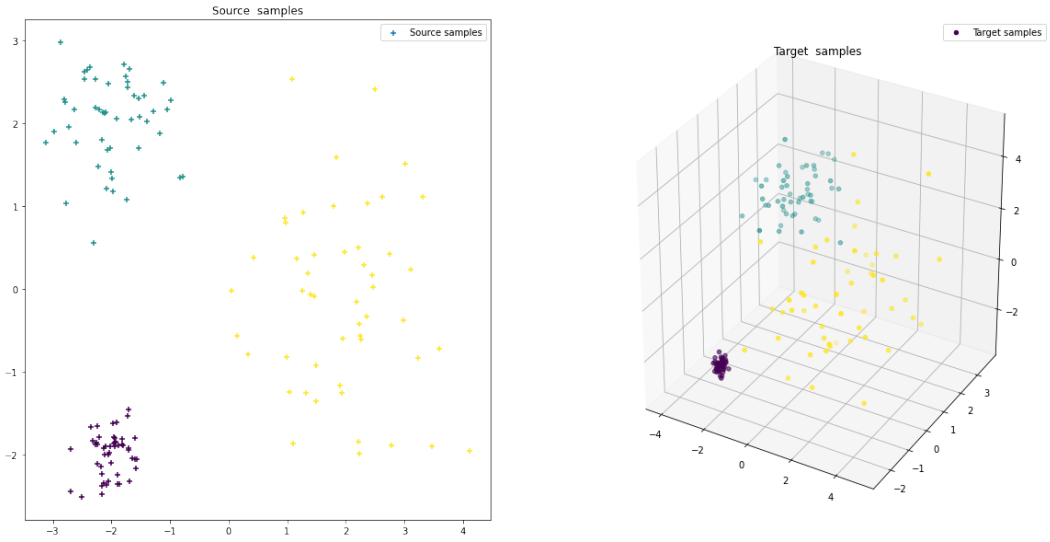
$$P_s(y|\mathbf{x}^s) = P_t(y|\mathbf{T}(\mathbf{x}^t))$$

From a probabilistic perspective we can say that we are looking for a transformation $\mathbf{T}$ such that $\mathbf{T}\#\mu_s = \mu_t$. Such transformation exists in large number, so we are going to add one more condition. This transformation $\mathbf{T}$ has to minimize a certain transportation cost $\sum_{i,j,k,l} |C_{i,k} - \bar{C}_{j,l}| \mathbf{T}_{i,j} \mathbf{T}_{k,l}$, hence transforming the problem to an optimal transportation problem. In all the methods presented in this work, we will take $C_{i,k}$ (respectively $\bar{C}_{i,k}$) to be the euclidean distance between the samples i and k. Using the optimal transport theory, we can then solve the domain adaptation task by:

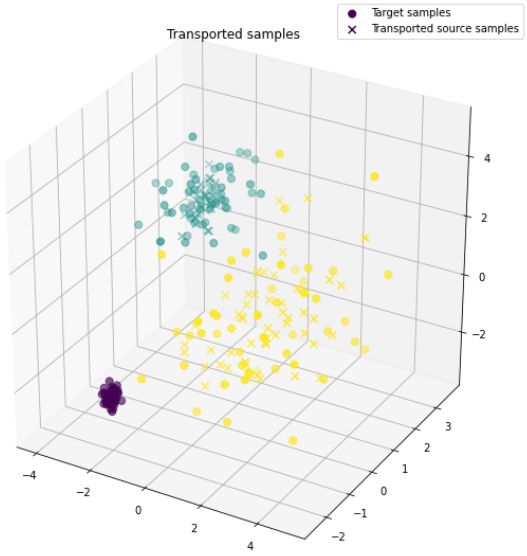
1. Estimating the transport map $\mathbf{T}$ using equation (4). In this work, we always set $\mu_s = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i^s}$ and $\mu_t = \frac{1}{m} \sum_{j=1}^{m} \delta_{x_j^t}$
2. Applying the mapping $\mathbf{T}$ to the source samples in order to move them to the target space while keeping their label. To map a source sample in the target space we use barycentric mapping. Given a sample x_i^t, it's corresponding sample in the target space is:

$$\widehat{\mathbf{x}_i^s} = \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \quad \sum_i \mathbf{T}(i,j)c(\mathbf{x}, \mathbf{x}_i^t) \tag{7}$$

3. Train a classifier on the target domain, using the transported samples.



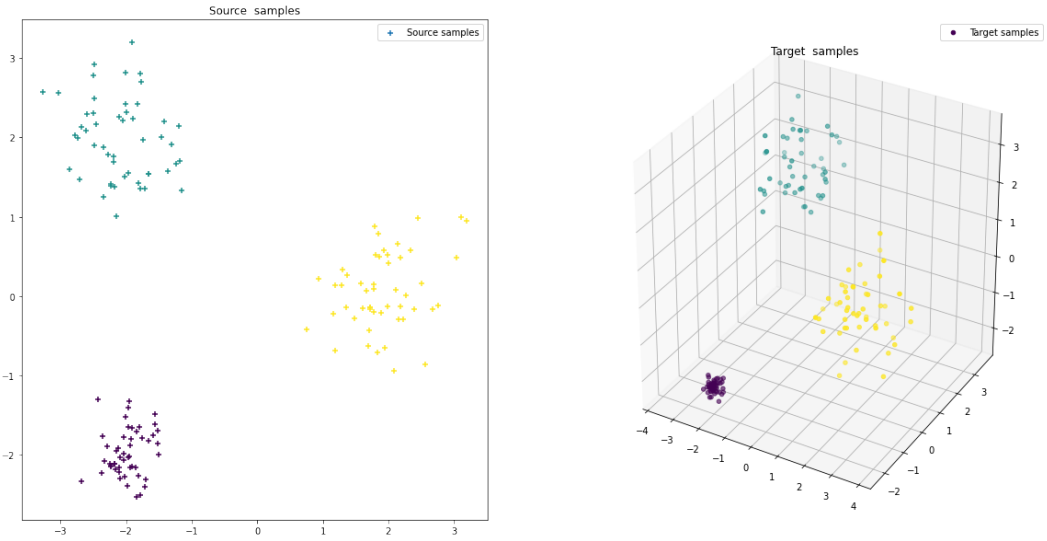
(a) Source (left) and target (right) datasets.



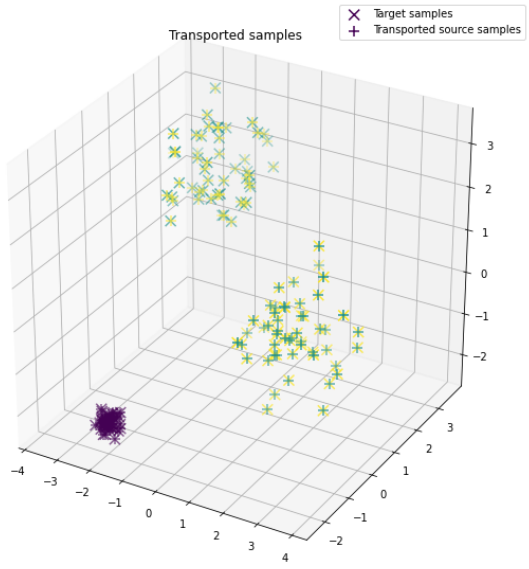
(b) Transported samples in the target space.

Figure 4: GW-OTDA on toy datasets.

While this method is pretty accurate, one has to be careful about the invariants of the Gromov-Wasserstein optimal transport, as you can see in the next example.



(a) Source (left) and target (right) datasets.



(b) Transported samples in the target space.

Figure 5: GW-OTDA on toy datasets, clusturing the samples well but with class inversion.

Note that a similar idea as presented here as been described in Yan et al. [2018], but we chose to include it in this work as it fits naturally into the way to introduce the next sections.

3.2 FGW-JDOT

In the last part, we made the additional assumption that $\mathbf{T}$ preserves the conditional distributions. While very helpful, this assumption is not very general or natural. In Courty et al. [2017], this assumption was lifted in the case of homogeneous domain adaptation. We will use the same idea

in the heterogeneous setting. We now want to directly align the joint distributions $P_s(\mathbf{x}^s, y)$ and $P_t(\mathbf{x}^t, y)$ using optimal transport. Of course, as we don't know the labels of the target samples, we cannot use an empirical estimate of $P_t(\mathbf{x}^t, y)$ and so we can't apply the same method as we described in section 3.1. Instead, we are going to use a surrogate version $f(x_i^t)$ depending on a classifier f and giving us a joint distribution on the target domain $P_t^f(x, f(x))$. Our goal is then going to jointly learn $\mathbf{T}$ and f.

Just like in the precedent section, we wish to find a transport plan T mapping P_s and P_t^f. As we want to leverage the structural information of our data (the labels), the optimal transport formulation of choice here will be the Fused Gromov-Wasserstein transport:

$$\mathbf{T} = \underset{\boldsymbol{\gamma}, f}{\operatorname{argmin}} \, \mathcal{E}_{f,\boldsymbol{\gamma}}$$

$$\mathbf{T} = \underset{\boldsymbol{\gamma}, f}{\operatorname{argmin}} \sum_{i,j,k,l} ((1-\alpha)L(y_i^s, f(x_j^t)) + \alpha |C_{i,k} - \bar{C}_{j,l}|)\boldsymbol{\gamma}_{i,j}\boldsymbol{\gamma}_{k,l} \tag{8}$$

This optimization problem has two variables to optimize: f and $\boldsymbol{\gamma}$. To solve it we use, as proposed in Courty et al. [2017] the Block Coordinate Descent algorithm.

Algorithm 1: Block Coordinate Descent

for *i from 1 to numberiter* **do**
- Minimize $\mathcal{E}_{f,\boldsymbol{\gamma}}$ with f fixed.
- Minimize $\mathcal{E}_{f,\boldsymbol{\gamma}}$ with $\boldsymbol{\gamma}$ fixed.

end

Minimizing $\mathcal{E}_{f,\boldsymbol{\gamma}}$ with f fixed boils down to a Fused-Gromov-Wasserstein optimal transport problem, whereas minimizing $\mathcal{E}_{f,\boldsymbol{\gamma}}$ with $\boldsymbol{\gamma}$ fixed is a simple optimization problem.

Below is an example of the FGW-JDOT method on a toy dataset, first taking domain A to be the source domain, and then domain B.

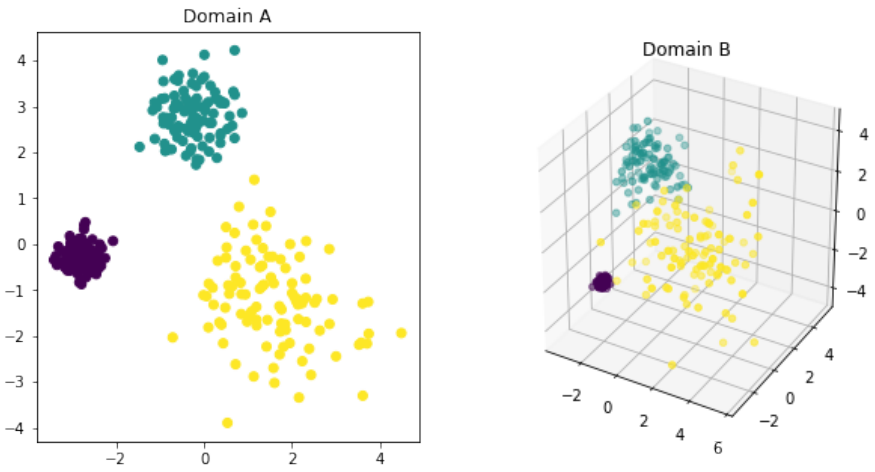
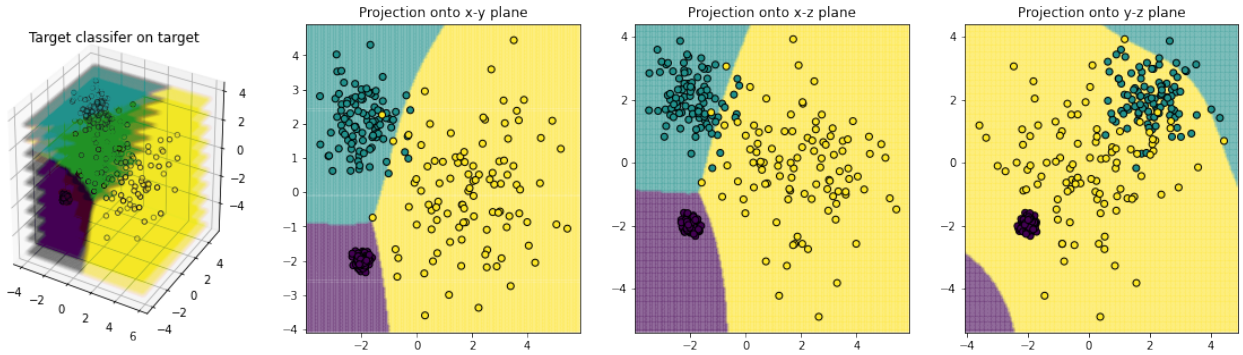
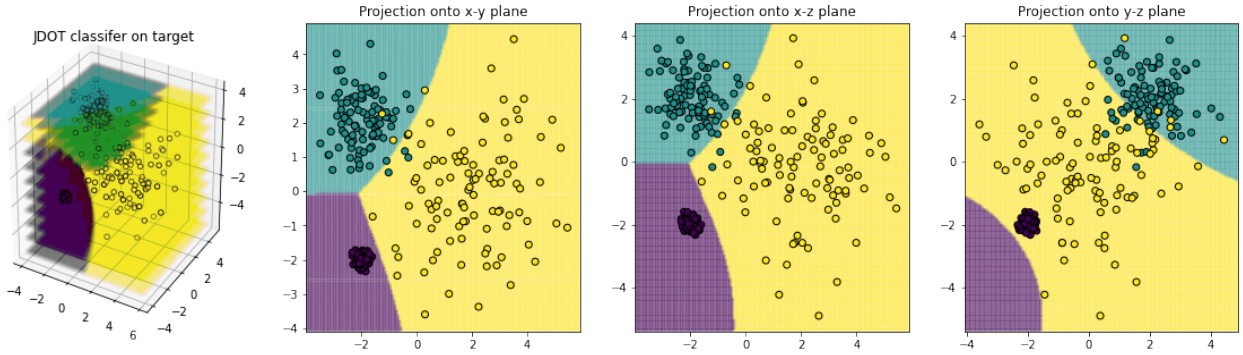


Figure 6: Toy dataset



(a) Target classifier trained with the target labels.



(b) Target classifier trained with the FGW-JDOT method.

Figure 7: FGW-JDOT results taking domain A to be the source domain.

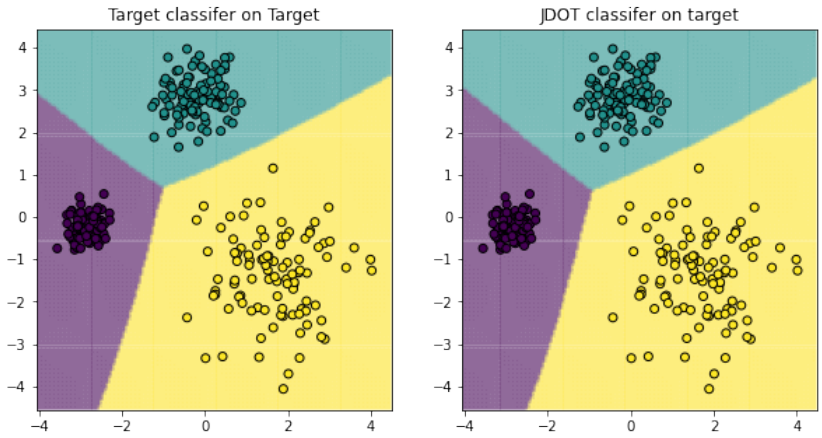


Figure 8: FGW-JDOT results taking domain B to be the source domain.

3.3 Deep FGW-JDOT

Following the same motivations and general idea as Damodaran et al. [2018], we propose a deep version of FGW-JDOT, to improve two main aspects of the method proposed in 3.2:

1. Solving for $\mathbf{T}$ becomes intractable for large datasets as the complexity of the Conditional Gradient algorithm scales quadratically with n and m.

2. The distances contained in the matrices C and $\bar{C}$ are the euclidean distances between the points in the source and target spaces. This distances matrices are not very informative about the semantic distance of our samples. Distances in a semantic spaces would be way more informative and would make the learning process more effective.

Our solution to both these problem is to adapt the general idea presented in 3.2 to work in a deep learning setting. We will present a stochastic method to solve the optimisation problem, and suggest the addition of two embedding functions g_s and g_t in order to have ground costs C and $\bar{C}$ that convey a more semantic meaning.
With these modifications, we now wish to solve the following optimisation problem:

$$\min_{f,\boldsymbol{\gamma}} \sum_{i,j,k,l} (\alpha |C^{g_s}_{i,k} - \bar{C}^{g_t}_{j,l}| + (1-\alpha) L(y^s_i, f(g(x^t_j))))\boldsymbol{\gamma}_{i,k}\boldsymbol{\gamma}_{j,l} \tag{9}$$

The stochastic learning procedure consist in randomly selecting a mini-batch of m samples, applying the following mini-batch update, and repeat until satisfactory performance is reached.

3.3.1 Mini-batch update procedure

The mini-batch update procedure is as follow:

1. Solve the optimisation problem for fixed f, g^s, g^t. This is equivalent to solving a Fused Gromov-Wassertein optimisation problem.

2. With fixed $\boldsymbol{\gamma}$, update f and g^t with a stochastic gradient descent update with the following loss function:

$$\sum_{i,j,k,l} (\alpha |C^{g_s}_{i,k} - \bar{C}^{g_t}_{j,l}| + (1-\alpha) L(y^s_i, f(g(x^t_j))))\boldsymbol{\gamma}_{i,k}\boldsymbol{\gamma}_{j,l} \tag{10}$$

4 Numerical experiments

For our numerical experiments, we've decided to use the MNIST-USPS dataset. Both MNIST and USPS datasets are handwritten digits recognition datasets but MNIST's samples are 28x28 pixels images whereas USPS's samples are 16x16 pixels images, so these datasets are a good fit for heterogeneous domain adaptation.

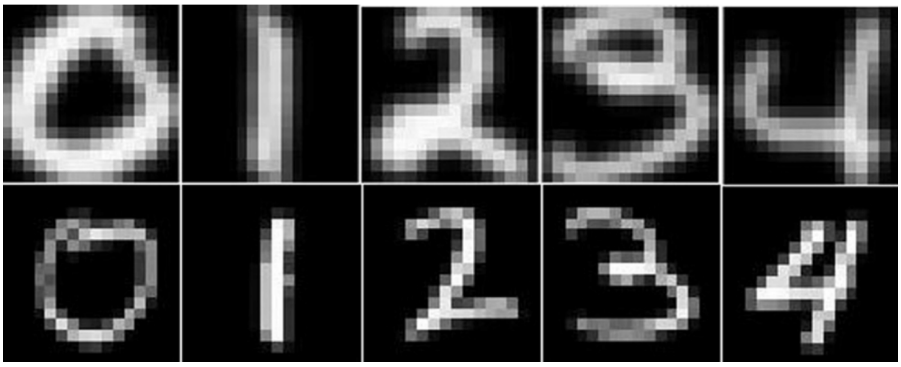


Figure 9: MNIST samples (on the top) and USPS samples (on the bottom)

4.1 Semi-supervised task

As we've seen during the examples, the methods explored in this work have serious geometrical invariants, preventing us to achieve good results on real life datasets. For this reason, our experiments are made in a slightly more convenient context: we have access to a small number (less than 10% of the of total number of samples) labelled examples in the target domain, allowing us to break the invariants. In practice, this allows us to set infinite cost for samples that we wish not to be matched. For example if x^s_5 has label 0 and x^t_3 has a known label and has label 1, then we artificially set $L(x^s_5, x^t_3) = \infty$, preventing the optimisation procedure to match them.

4.2 Results

For our experiment, we took the MNIST to be the source domain and the USPS the target domain. The USPS dataset contains about 9000 samples and we've randomly chosen 50 samples to be the known samples of our semi-supervised task (5.5%). We've used the Deep FGW-JDOT described in

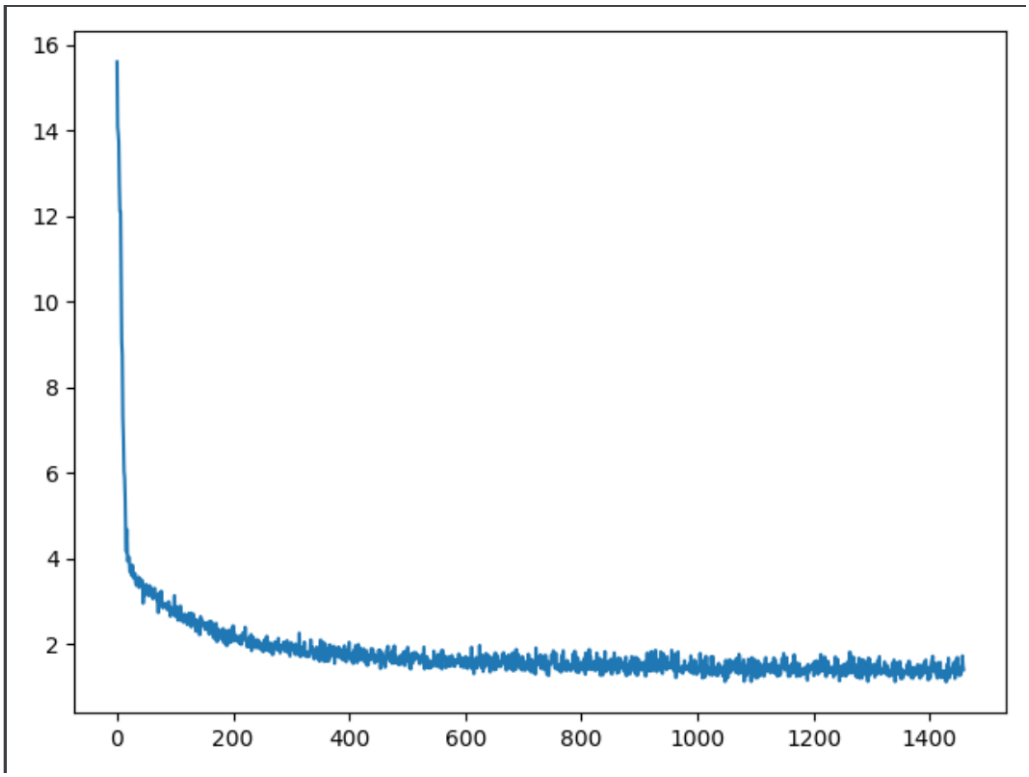


Figure 10: Training loss in function of the number of mini-batch seen during the experiment.

section 3.3. The two embeddings functions are two-layers convolutionnal neural networks whereas the classifier f is a fully connected deep neural network. Our experimental results show an accuracy of 78% on the USPS dataset, while the accuracy reached by training the same network using only the 50 labelled samples barely reaches 15%, hence proving that we've actually been able to transfert some knowledge!

5 Acknowledgments

I would like to thank Rémi Flamary and Nicolas Courty for their mentoring all along this internship. They've always be available to answer my many questions and have always given me enough insight to help me solving my problems while letting me find the solutions alone. This research experience has been great for me and I've learned a lot about how reasearch is actually made.

References

N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation, 2015.

N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy. Joint distribution optimal transportation for domain adaptation, 2017.

M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transportation distances, 2013.

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

C. Mondon and J. Malka. Du transport optimal et de son application en apprentissage statistique, 2020.

G. Peyré and M. Cuturi. Computational optimal transport, 2018.

J. Solomon, G. Peyré, V. G. Kim, and S. Sra. Entropic Metric Alignment for Correspondence Problems. *ACM Transactions on Graphics*, 35(4):72:1–72:13, June 2016.

V. Titouan, N. Courty, R. Tavenard, C. Laetitia, and R. Flamary. Optimal transport for structured data with application on graphs. volume 97 of *Proceedings of Machine Learning Research*, pages 6275–6284, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

T. Vayer, L. Chapel, R. Flamary, R. Tavenard, and N. Courty. Fused gromov-wasserstein distance for structured objects: theoretical foundations and mathematical properties, 2018.

C. Villani. *Optimal transport – Old and new*, volume 338, pages xxii+973. 01 2008.

Y. Yan, W. Li, H. Wu, H. Min, M. Tan, and Q. Wu. Semi-supervised optimal transport for heterogeneous domain adaptation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2969–2975. International Joint Conferences on Artificial Intelligence Organization, 7 2018.