Metric Learning in Optimal Transport for Domain Adaptation

Tanguy Kerdoncuff, **Rémi Emonet** and **Marc Sebban**
Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien
UMR 5516, F-42023, SAINT-ETIENNE, France
{tanguy.kerdoncuff, remi.emonet, marc.sebban}@univ-st-etienne.fr

Abstract

Domain Adaptation aims at benefiting from a labeled dataset drawn from a *source* distribution to learn a model from examples generated according to a different but related *target* distribution. Creating a domain-invariant representation between the two source and target domains is the most widely technique used. A simple and robust way to perform this task consists in (i) representing the two domains by subspaces described by their respective eigenvectors and (ii) seeking a mapping function which aligns them. In this paper, we propose to use Optimal Transport (OT) and its associated Wasserstein distance to perform this alignment. While the idea of using OT in domain adaptation is not new, the original contribution of this paper is two-fold: (i) we derive a generalization bound on the target error involving several Wasserstein distances. This prompts us to optimize the ground metric of OT to reduce the target risk. (ii) From this theoretical analysis, we design an algorithm (MLOT) which optimizes a Mahalanobis distance leading to a transportation plan that adapts better. Experiments demonstrate the effectiveness of this original approach.

1 Introduction

Domain adaptation (DA) has been shown to be very effective in many real world applications, *e.g.*, in computer vision, medical diagnosis, or recommender systems, to cite a few. The main idea is to use labeled data of a source domain to improve the performance of a classifier deployed on a related target domain which suffers from a lack of labeled examples. In this paper, we address a complex setting, called unsupervised DA, where there is only unlabeled data available from the target distribution.

Different approaches have been proposed to tackle this problem, some of them coming with theoretical guarantees (see, *e.g.* the survey [Redko *et al.*, 2019b]). One classical way is to learn a common latent space where the distribution shift is smaller. For instance, the Subspace Alignment algorithm (SA) [Fernando *et al.*, 2013] learns a classifier in a subspace obtained after a linear alignment of the source and target eigenvectors. In a similar manner, the Correlation Alignment (CORAL) [Sun *et al.*, 2016] uses the covariance of the source and target distributions to reduce the shift, while Transfer Component Analysis (TCA) [Pan *et al.*, 2011] looks for common features between the two domains. Some other works directly learn the target labels but do not gather the two distributions in a common feature space. This is the case of MEDA [Wang *et al.*, 2018] which learns a domain-invariant classifier in Grassman manifold. On the other hand, DA with deep learning has received much attention during the past decade from the computer vision community leading to a substantial amount of research to address visual tasks for which a large amount of training data or a pre-trained model is available (see, *e.g.* the survey [Wang and Deng, 2018]).

More recently, Optimal Transport (OT) has been shown to be a very promising tool to perform DA tasks. OT consists in mapping two source and target probability measures with a minimal cost of transportation associated to the so-called Wasserstein distance. Beyond its use in deep learning to solve visual DA tasks (see, *e.g.*, [Sun and Saenko, 2016; Bhushan Damodaran *et al.*, 2018]), this idea of reducing the shift by OT has been exploited in a more generic DA setting by the algorithm OTDA [Courty *et al.*, 2017b]. OTDA modifies the original Kantorovich optimization problem by resorting to a regularization preventing the transportation plan from moving two source points of different labels onto the same target example. Then, a classifier can be learned from the labeled source data and deployed over the target distribution. Based on this work, [Courty *et al.*, 2017a] ensures that the final classifier is coherent with the transportation plan.

Inspired from both SA and OTDA, our contribution aims at using OT for domain adaptation by aligning the source and target subspaces. The main conjecture we formulate in this paper is that the Euclidean distance usually used as the cost matrix in OT may not be the best metric to perform the adaptation. While learning a better metric (especially a Mahalanobis distance) in OT has been recently studied [Deshpande *et al.*, 2019; Genevay *et al.*, 2018; Paty and Cuturi, 2019; Cuturi and Avis, 2014], optimizing such a ground metric to address DA tasks has not receive attention yet. We fill this gap from both a theoretical and an algorithmic perspective. First, we formally establish a relation between the target error and the magnitude of different Wasserstein distances. This prompts us to see the Wasserstein distance as a parameterized

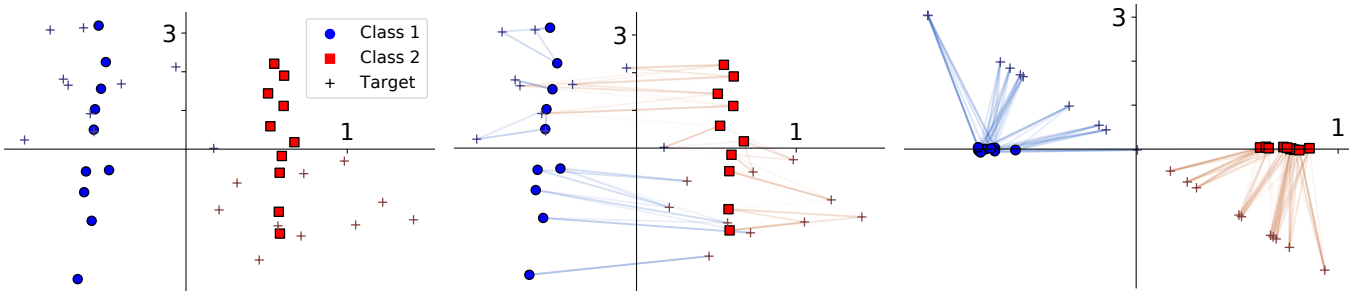

Figure 1: Behavior of MLOT on a toy dataset. On the left, the original source and target examples. In the middle, OTDA fails to transport correctly the blue and red classes. On the right, the proposed MLOT which combines a learned metric and some per-domain dimensionality reduction, leads to a perfect transportation plan. Notice the difference in scale between the two axes.

metric that might be minimized, leading to a better transportation plan for DA. We also formally make a link between a PCA and the minimization of the Wasserstein distance. Based on this theoretical analysis, we propose MLOT, an algorithm which optimizes a Mahalanobis distance that improves the Optimal Transport between the source and target subspaces generated by a PCA. Unlike OTDA which does not change the feature space, MLOT jointly optimizes (i) the dimensionality reduction of the source domain, (ii) the transportation plan between the source and the target and (iii) the underlying metric used in the transportation. The intuition behind MLOT is illustrated in Figure 1. The original source and target examples are represented on the left. The second figure shows the limitation of OTDA when the transportation is performed in the original feature space. The figure on the right gives evidence on the advantage of performing a PCA before learning jointly the ground metric and the transportation plan.

The rest of the paper is organized as follows: Section 2 reminds the main principles of OT, DA and Metric Learning. Section 3 is dedicated to the theoretical contribution of our paper which leads to the design of our MLOT algorithm in Section 4. An extensive experimental study is presented in Section 5 before our conclusion in Section 6.

2 Optimal Transport and Metric Learning

In this section, we briefly introduce Optimal Transport, its use in OTDA, and Metric Learning.

2.1 Optimal Transport and OTDA

Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a feature space. We consider here a source distribution μ_s and a target distribution μ_t both defined over $\mathcal{X}$. OT [Villani, 2008] aims at moving μ_s on the top of μ_t with a transportation plan of minimal cost. We will use in this paper the formulation proposed by Kantorovich [1942] which gives the Wasserstein distance between μ_s and μ_t. Let $\Pi(\mu_s, \mu_t)$ be the collection of all joint probability measures on $\mathcal{X} \times \mathcal{X}$ with marginals μ_s and μ_t respectively. Let suppose that μ_s and μ_t have a finite p-order moment with $p \geq 1$. One can define the p-Wasserstein distance at the power p as

$$\mathcal{W}_p^p(\mu_s, \mu_t) = \min_{\gamma \in \Pi(\mu_s, \mu_t)} \int_{\mathcal{X} \times \mathcal{X}} c(x_s, x_t)^p \, d\gamma(x_s, x_t), \quad (1)$$

where $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a cost function. The minimizer of the previous problem γ^* corresponds to the Optimal Transport plan. In practice, we deal with the empirical measures $\hat{\mu}_s$ and $\hat{\mu}_t$ supported on m_s and m_t examples respectively. In such a case, γ^* can be represented by a matrix of size $m_s \times m_t$ where $\gamma^*_{i,j}$ corresponds to the mass that should be sent from the point x_s^i to the point x_t^j. Let $\hat{\Pi}(\hat{\mu}_s, \hat{\mu}_t) = \{\gamma \in \mathbb{R}_+^{m_s \times m_t} | \gamma \mathbf{1}_{m_t} = \hat{\mu}_s, \gamma^T \mathbf{1}_{m_s} = \hat{\mu}_t\}$. Considering the Euclidean distance as the cost function, we denote by C^p the $m_s \times m_t$ matrix composed of the costs $C^p_{ij} = \|x_s^i - x_t^j\|_2^p$. The p-Wasserstein distance can be reformulated as follows:

$$\begin{aligned} \mathcal{W}_p^p(\hat{\mu}_s, \hat{\mu}_t) &= \min_{\gamma \in \hat{\Pi}(\hat{\mu}_s, \hat{\mu}_t)} \sum_{i=1}^{m_s} \sum_{j=1}^{m_t} \|x_s^i - x_t^j\|_2^p \gamma_{ij} \\ &= \min_{\gamma \in \hat{\Pi}(\hat{\mu}_s, \hat{\mu}_t)} \langle \gamma, C^p \rangle. \end{aligned} \quad (2)$$

It is worth noting that (i) Problem (2) is a linear program, (ii) the solution is not stable and (iii) the algorithmic complexity is supercubic. To change it into a strongly convex problem with a unique solution, we usually add a regularization term in the form of the classic Shannon entropy [Cuturi, 2013]. Let $\lambda_e \in \mathbb{R}_+$ be a small regularization parameter, we get

$$\mathcal{W}_p^p(\hat{\mu}_s, \hat{\mu}_t) \approx \left\langle \operatorname*{argmin}_{\gamma \in \hat{\Pi}(\hat{\mu}_s, \hat{\mu}_t)} \langle \gamma, C^p \rangle - \lambda_e \Omega_e(\gamma), C^p \right\rangle, \quad (3)$$

with $\Omega_e(\gamma) = -\sum_{i=1}^{m_s} \sum_{j=1}^{m_t} \gamma_{ij} \log(\gamma_{ij})$. Problem (3) can be efficiently solved by resorting to the Sinkhorn-Knopp algorithm [Cuturi, 2013]. The mass of a source point will now be spread across several target points preventing the algorithm from overfitting and leading to smoother solutions.

OTDA (Optimal Transport for Domain Adaptation) [Courty *et al.*, 2017b] was the first attempt to use Optimal Transport for Domain Adaptation. While Problem (3) is totally unsupervised, OTDA takes into account the labels of the transported points. Let us suppose that a discrete label $y_s \in \mathcal{Y} = [\![1, c]\!]$ is associated to each source example, with

c the number of classes. OTDA adds a penalty term to prevent two source points of different labels from being sent to the same target location. This takes the form of the following optimization problem:

$$\min_{\gamma \in \Pi(\hat{\mu}_s, \hat{\mu}_t)} \langle \gamma, C^p \rangle - \lambda_e \Omega_e(\gamma) + \lambda_c \Omega_c(\gamma), \quad (4)$$

where $\Omega_c(\gamma) = \sum_{j=1}^{m_t} \sum_{cl=1}^{c} \|\gamma(\mathcal{I}_{cl}, j)\|_2$ and $\gamma(\mathcal{I}_{cl}, j)$ is the column j of matrix γ with only the rows corresponding to samples of class cl.

2.2 Metric Learning

As said before, the cost matrix C used in OT is usually set to the Euclidean distance. Though this choice seems natural, it has a direct impact on the quality of the transportation. We suggest here to optimize this matrix by learning a metric that allows us to better match in a DA setting the source and target distributions, hopefully in a smaller feature space. Metric Learning (ML), and especially Mahalanobis distance learning, has been widely studied in the literature during the past decade (see, *e.g.*, the survey [Bellet *et al.*, 2015]). It typically boils down to optimizing the shape and the orientation of an ellipsoid rather than using the Euclidean ball. More formally, let $L \in \mathbb{R}^{k \times n}$ with $k \in [\![1, n]\!]$ and M be a PSD matrix such as $M = L^T L$. For all $(i, j) \in [\![1, m_s]\!]^2$, the squared Mahalanobis distance D^2_M parameterized by M is defined as

$$D^2_M(x^i_s, x^j_s) = (x^i_s - x^j_s)^T M (x^i_s - x^j_s) = \|L(x^i_s - x^j_s)\|^2_2. \quad (5)$$

Notice that L defines a unique M but there is more than one Cholesky decomposition of M. The goal of Metric Learning is to learn either the matrix L or M under semantic constraints, which typically aim to bring closer examples of the same class while pushing away data of different labels (see, *e.g.* LMNN [Weinberger and Saul, 2009] or ITML [Davis *et al.*, 2007]). The problem is often convex in M but the PSD constraint makes the optimization more complicated. The minimization in L is not convex but is simpler and gives good results in practice. In the rest of this paper, we will denote by $\Omega_l(L)$ the underlying objective function of the metric learning problem. Note that learning a Mahalanobis distance for OT has been recently studied [Deshpande *et al.*, 2019; Genevay *et al.*, 2018; Paty and Cuturi, 2019; Cuturi and Avis, 2014]. Our objective in the rest of this paper is to show how to optimize such a ground metric when OT is used to address domain adaptation tasks by the alignment of the source and target subspaces.

3 Theoretical Analysis of DA with OT

In this section, we derive two theoretical results. First, we establish a strong relation between a PCA and the minimization of the Wasserstein distance. Then, we derive a generalization bound on the target error whose terms depend on several Wasserstein distances. By changing the Euclidean costs by a Mahalanobis distance, we can see $\mathcal{W}^p_p(\hat{\mu}_s, \hat{\mu}_t)$ as a parameterized distance that might be optimized from training data. This leads to the design of a new Domain Adaptation algorithm, called MLOT (see Section 4), which learns the ground metric allowing us to optimize the transportation plan while performing the adaptation.

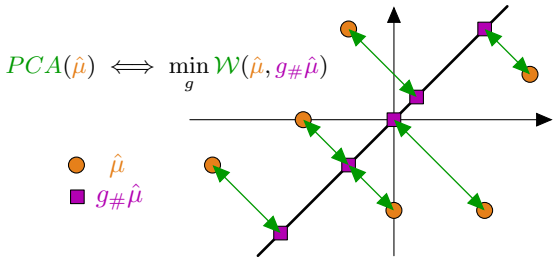

Figure 2: Standard PCA projecting a discrete distribution $\hat{\mu}$ (data in orange) onto a one-dimensional subspace (in purple). This projection is equivalent to finding the optimal mapping function g that minimizes the Wasserstein distance $\mathcal{W}(\hat{\mu}, g_{\#}\hat{\mu})$.

3.1 PCA and Wasserstein Distance

As usually done in OT, let us use the push forward notation. For any measurable function $g : \mathcal{X} \longrightarrow \mathcal{X}$ and distribution μ on $\mathcal{X}$, we define $g_{\#}\mu(B) = \mu(g^{-1}(B))$ for all Borel $B \in \mathcal{B}(\mathcal{X})$. In practice, this means that we draw a point from μ and then apply the transformation g to that point. If g is a linear function, it can be assimilated to its associated matrix G. Let $Dim(Im(g))$ be the dimension of the *affine* subspace formed by the image of g. This notation allows us to define the dimension of a non-centered vector space.

The following theorem aims at showing that PCA is the best way to reduce the dimension of a distribution in the sense of the Wasserstein distance. By dimensionality reduction, we mean that the transformed distribution lays in a subspace (not necessarily centered) of $\mathcal{X}$, but the points are still defined in $\mathcal{X}$ (see Figure 2). For the sake of clarity, we consider here the case of the 2-Wasserstein with the Euclidean distance as the underlying distance. We also focus on the case of a discrete centered distribution. The complete derivation for the more general case can be proved in a similar way.

Theorem 1. *Given a set of m examples $\{x^i\}_{i=1}^m$ lying in a n dimensional space and i.i.d. from a distribution μ. Let $\hat{\mu}$ the empirical counterpart of μ defined as $\hat{\mu} = \frac{1}{m}\sum_{i=1}^m \delta_{x^i}$ with $E(\hat{\mu}) = 0$. Let $d \in [\![1, n]\!]$ and V be the $d \times n$ matrix formed with the first d normalized eigenvectors of the covariance matrix of $\hat{\mu}$. Let $G_d = \{g : \mathbb{R}^n \longrightarrow \mathbb{R}^n | Dim(Im(g)) \leq d\}$. Then, we have*

$$\operatorname*{argmin}_{g \in G_d} \mathcal{W}_2^2(\hat{\mu}, g_{\#}\hat{\mu}) = V^T V. \quad (6)$$

Proof. Suppose that the minimizer $g^* \in G_d$ of Problem (6) exists. In such a case, let γ_{g^*} be the associated optimal coupling matrix. With these notations, we get:

$$\min_{g \in G_d} \mathcal{W}_2^2(\hat{\mu}, g_{\#}\hat{\mu}) = \sum\nolimits_{i,j} \|x^i - g^*(x^j)\|_2^2 (\gamma_{g^*})_{i,j}. \quad (7)$$

To get rid of the last term of Eq. (7), let us reorder the function g^* into $\tilde{g}^*(x^i) = g^*(x^j)$ when $(\gamma_{g^*})_{i,j} = \frac{1}{m}$. In this case, the corresponding matrix $\gamma_{\tilde{g}^*} = \frac{1}{m} I_m$ with I_m the identity matrix, and we get:

$$\min_{g \in G_d} \mathcal{W}_2^2(\hat{\mu}, g_{\#}\hat{\mu}) = \sum\nolimits_{i,j} \|x^i - \tilde{g}^*(x^j)\|_2^2 (\gamma_{\tilde{g}^*})_{i,j}$$

$$\min_{g\in G_d} \mathcal{W}_2^2(\hat{\mu}, g_{\#}\hat{\mu}) = \frac{1}{m}\sum_i \|x^i - \tilde{g}^*(x^i)\|_2^2 \geq \frac{1}{m}\sum_i \|x^i - V^T V x^i\|_2^2.$$

The last line comes from the PCA. It shows that $V^T V$ associated with $\frac{1}{m} I_m$ upper bounds the optimal solution g^*. But since $V^T V \in G_d$, it is actually the optimal solution. □

Theorem 1 tells us that PCA is the best way to reduce the dimension in the sense of the Wasserstein distance. The next result provides a generalization bound on the target error where Wasserstein distances are the main terms to minimize.

3.2 Upper Bound on the Target Risk

Let $\mathcal{H} = \{h | h : \mathcal{X} \to \mathcal{Y}\}$ be the hypothesis space. We assume the existence of a deterministic ground-truth function $f : \mathcal{X} \to \mathcal{Y}$ which gives the label associated to each point of $\mathcal{X}$. For all $(h, h') \in \mathcal{H}^2$, we define $\epsilon_t(h, h') = P_{x\sim\mu_t}(h(x) \neq h'(x))$ and $\epsilon_s(h, h') = P_{x\sim\mu_s}(h(x) \neq h'(x))$ which represent the disagreement between two classifiers. The goal of DA is to find the best $h \in \mathcal{H}$ which minimizes the target risk $\epsilon_t(h, f) = P_{x\sim\mu_t}(h(x) \neq f(x))$. In the same way, $\epsilon_s(h, f) = P_{x\sim\mu_s}(h(x) \neq f(x))$ is the source risk. To simplify the notations, let us use $\epsilon_t(h, f) = \epsilon_t(h)$ and $\epsilon_s(h, f) = \epsilon_s(h)$.

Lemma 1 (Generalization bound [Shen *et al.*, 2017])**.** *Let μ_s, μ_t be two probability measures on $\mathcal{X}$. Assume the hypotheses $h \in \mathcal{H}$ are all K-Lipschitz continuous for some $K \in \mathbb{R}_+^*$. Then, $\forall (h, h)' \in \mathcal{H}^2$, the following holds:*

$$\epsilon_t(h, h') \leq \epsilon_s(h, h') + 2K\mathcal{W}_2(\mu_s, \mu_t). \tag{8}$$

Building on the proof of Redko et al. [2017] based on the seminal works of Ben-David et al. [2007] and Mansour et al. [2009], we derive the following bound.

Theorem 2. *Let $g_s : \mathcal{X} \longrightarrow \mathcal{X}$ and $g_t : \mathcal{X} \longrightarrow \mathcal{X}$. Under the assumption of Lemma 1, $\forall h \in \mathcal{H}$, the following holds:*

$$\begin{aligned}\epsilon_t(h) \leq \epsilon_s(h) &+ 2K\left[\mathcal{W}_2(g_{s\#}\hat{\mu}_s, g_{t\#}\hat{\mu}_t)\right] \\ &+ 2K\left[\mathcal{W}_2(\hat{\mu}_s, g_{s\#}\hat{\mu}_s) + \mathcal{W}_2(g_{t\#}\hat{\mu}_t, \hat{\mu}_t)\right] \\ &+ 2K\left[\mathcal{W}_2(\mu_s, \hat{\mu}_s) + \mathcal{W}_2(\hat{\mu}_t, \mu_t)\right] + \lambda\end{aligned} \tag{9}$$

where λ is the combined error of the ideal hypothesis h^* that minimizes the combined error $\epsilon_s(h^*) + \epsilon_t(h^*)$.

Proof. We have :

$$\begin{aligned}\epsilon_t(h) &\leq \epsilon_t(h^*) + \epsilon_t(h^*, h) \\ &= \epsilon_t(h^*) + \epsilon_s(h, h^*) + \epsilon_t(h^*, h) - \epsilon_s(h, h^*) \\ &\leq \epsilon_t(h^*) + \epsilon_s(h, h^*) + 2K\mathcal{W}_2(\mu_s, \mu_t) \\ &\leq \epsilon_t(h^*) + \epsilon_s(h) + \epsilon_s(h^*) \\ &\quad + 2K\left[\mathcal{W}_2(\mu_s, \hat{\mu}_s) + \mathcal{W}_2(\hat{\mu}_s, \mu_t)\right] \\ &\leq \epsilon_s(h) + \lambda \\ &\quad + 2K\left[\mathcal{W}_2(\mu_s, \hat{\mu}_s) + \mathcal{W}_2(\hat{\mu}_s, \hat{\mu}_t) + \mathcal{W}_2(\hat{\mu}_t, \mu_t)\right] \\ &\leq \epsilon_s(h) + 2K\left[\mathcal{W}_2(g_{s\#}\hat{\mu}_s, g_{t\#}\hat{\mu}_t)\right] \\ &\quad + 2K\left[\mathcal{W}_2(\hat{\mu}_s, g_{s\#}\hat{\mu}_s) + \mathcal{W}_2(g_{t\#}\hat{\mu}_t, \hat{\mu}_t)\right] \\ &\quad + 2K\left[\mathcal{W}_2(\mu_s, \hat{\mu}_s) + \mathcal{W}_2(\hat{\mu}_t, \mu_t)\right] + \lambda.\end{aligned}$$

□

Algorithm 1 MLOT

Input: η (gradient step) X_s X_t Y_s

$V_s = PCA(X_s)$, $V_t = PCA(X_t)$
$L_s = V_s^T V_s$, $L_t = V_t^T V_t$
for $i = 1$ **to** N **do**
  $\gamma = \underset{\gamma\in\hat{\Pi}(\hat{\mu}_s,\hat{\mu}_t)}{\operatorname{argmin}} \langle\gamma, C^2(L_s, L_t)\rangle - \lambda_e\Omega_e(\gamma) + \lambda_c\Omega_{cl}(\gamma)$
  $L_s = L_s - \eta\nabla_{L_s}(\langle\gamma, C^2(L_s, L_t)\rangle + \lambda_l\Omega_l(L_s))$
end for
$\tilde{X}_s = \gamma L_t X_t$
classifier = classifier method($\tilde{X}_s, Y_s$)
$\hat{Y}_t$ = classifier(X_t)
return $\hat{Y}_t$

Note that if g_s and g_t are linear, $g_{s\#}\hat{\mu}_s$ and $g_{t\#}\hat{\mu}_t$ can be seen as linear projections of the source and target examples taking the form of two matrices that can be learned by any standard metric learning algorithm. Therefore, rather than discussing about its tightness, it is worth noticing that the previous bound is the first one which jointly relates (i) the target risk in domain adaptation, (ii) the minimization of the Wasserstein distance and (iii) the metrics that can be learned to get a better transportation plan. We can use Theorem 1 to minimize both $\mathcal{W}_2(\hat{\mu}_s, g_{s\#}\hat{\mu}_s)$ and $\mathcal{W}_2(g_{t\#}\hat{\mu}_t, \hat{\mu}_t)$. Note that $\mathcal{W}_2(\mu_s, \hat{\mu}_s)$ and $\mathcal{W}_2(\hat{\mu}_t, \mu_t)$ can be bounded under some assumptions using Theorem 2.1 in [Bolley *et al.*, 2007]. Moreover, λ is supposed to be small to allow the adaptation. A theoretical analysis about λ is available in [Redko *et al.*, 2019a].

The last term of interest that has to be minimized in the bound of Theorem 2 is $\mathcal{W}_2(g_{s\#}\hat{\mu}_s, g_{t\#}\hat{\mu}_t)$. We address this problem from an algorithmic perspective thanks to our algorithm MLOT presented in the next Section.

4 MLOT: Metric Learning in OT for DA

Inspired from OTDA [Courty *et al.*, 2017b], our algorithm MLOT leverages our previous theoretical analysis and resorts to an additional term $\Omega_l(L_s)$ dedicated to optimize a metric allowing us to get a better transportation plan. Let us consider the cost function $C^2(L_s, L_t)_{ij} = \|L_s x_s^i - L_t x_t^j\|_2^2$. MLOT takes the form of the following joint optimization problem:

$$\begin{aligned}\min_{L_s\in\mathbb{R}^{n\times n}, \gamma\in\hat{\Pi}(\hat{\mu}_s,\hat{\mu}_t)} &\langle\gamma, C^2(L_s, L_t)\rangle - \lambda_e\Omega_e(\gamma) \\ &+ \lambda_c\Omega_{cl}(\gamma) + \lambda_l\Omega_l(L_s).\end{aligned} \tag{10}$$

Note that MLOT only learns the matrix L_s associated to the source data. The reason is twofold. First, it prevents the algorithm from leading to a trivial minimal solution $\mathcal{W}_2(L_{s\#}\hat{\mu}_s, L_{t\#}\hat{\mu}_t)$ where both matrices L_s and L_t are null. By this way, MLOT tends to provide two different matrices L_s (which is learned) and L_t (set to $V_t^T V_t$ according to Theorem 1) which better capture the peculiarities of the two distributions. Second, labels - that are required to learn a metric - are available only in the source domain. To find the solution of Problem (10), we minimize the objective function w.r.t. γ and L_s alternately. From a practical point of view, we apply one step of gradient descent over L_s and then completely

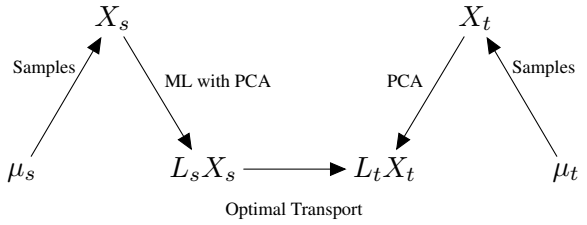

Figure 3: Workflow of MLOT.

compute the optimal γ. Since the problem is not convex, the initialization of L_s is key. According to Theorem 1, we use a PCA to set $L_s = V_s^T V_s$. At the initialization, L_s is a $n \times n$ matrix of rank d. But note that along the iterations, this rank can increase if it allows a better adaptation. The pseudo-code of MLOT is described in Algo. 1, where X_s and X_t are the source and target datasets, and Y_s is the set of source labels.

Note that any gradient descent-based metric learnign algorithm can be used to learn L_s via the term $\lambda_l \Omega_l(L_s)$. Note also that the computation of the barycenter of the transported points $\tilde{X}_s = \gamma L_t X_t$ makes sense only if the 2-Wasserstein is used. Figure 3 summarizes the workflow of MLOT.

5 Experiments

In this section, we perform experiments and demonstrate the effectiveness of MLOT compared to OTDA and other baselines on various datasets and types of features.

5.1 Datasets

We use the Office-Caltech dataset [Gong *et al.*, 2012] which is a classic benchmark on visual DA. We study the effect of using different features such as SURF features [Bay *et al.*, 2006] and DeCAF Deep Learning features [Donahue *et al.*, 2014]. The Office-Caltech dataset is composed of 4 different subsets (Amazon, Caltech, DSRL, Webcam) that are combined in a pairwise manner, to create 12 DA subproblems. The notation $A \rightarrow C$ means that Amazon is used as the source and Caltech as the target. There are the same 10 classes in each dataset containing from 157 to 1,123 images.

We also use the Office31 dataset [Saenko *et al.*, 2010] with features extracted from the 7^{th} layer of DeCAF Deep Learning network. The dataset is composed of 4,110 images in 3 subsets (Amazon, DSLR, Webcam) with 31 classes.

5.2 Setup and Cross-validation

We compare 10 different methods[1] that are able to handle arbirtary features. We exclude deep learning methods as they require having the original images and fine tuning a network. We compare: **NA**(No Adaptation). The classifier is learned on the source dataset and directly applied on the target. **LMNN**: Large Margin Nearest Neighbor [Weinberger and Saul, 2009]. **SA**: Subspace Alignment [Fernando *et al.*, 2013]. **CORAL**: CORrelation ALignment [Sun *et al.*, 2016]. **TCA**: Transfer Component Analysis [Pan *et al.*, 2011]. **OT**: Optimal Transport with entropy [Cuturi, 2013]. **OTDA**: Optimal Transport with entropy and class regularization [Courty *et al.*, 2017b]. **OTDA$_\mathbf{p}$**: OTDA after a PCA. **JDOT**: Joint Distribution Optimal Transportation [Courty *et al.*, 2017a]. **MLOT**: our method. Following [Courty *et al.*, 2017b] the final classification is done with a 1-Nearest Neighbor (1NN).

In unsupervised DA, there is no target label and it is impossible to use the classical cross-validation procedure to choose the best hyper-parameters. To fairly compare methods, we take inspiration from the work of [Zhong *et al.*, 2010] and apply the following strategy for all methods. We first assign pseudo-labels to the target points (using the considered method) and then use these target labels to re-assign labels to the source data, using a basis DA algorithm. Here, we choose SA [Fernando *et al.*, 2013] which has been shown to be one of the most robust DA method. We can then compare the actual source labels with the predicted source ones. We take the set of hyper-parameters that gives the best accuracy over 48 hours, limited to 1000 iterations. This back-and-forth adaptation is done independently for each pair of datasets. MLOT is parameterized by 5 hyper-parameters: the three regularization parameters (λ_e, λ_c, λ_l) which control the trade-off between each term in Eq. (10), the number of dimensions kept by the PCA (d) and the number of iterations (N). Note that in these experiments, we used arbitrarily LMNN [Weinberger and Saul, 2009] to learn L_s in the term $\Omega_l(L_s)$. Therefore, an additional parameter has to be tuned corresponding to the margin used in this metric learning algorithm. Note that SA and MLOT resort to a PCA. To speed-up the process, we used a "randomized"-PCA [Halko *et al.*, 2011] and run 10 iterations. This explains why the variance is indicated for these three methods in the reported results in Table 1.

The code of the 10 methods is available[2], together with the datasets, the code for the cross-validation that recreates Table 1, and the code that produces automatically Figures 1 and 4.

5.3 Analysis of the Results

The results are reported in Table 1. For the SURF features, MLOT outperforms, on average, all the other methods. MLOT outperforms OTDA 8 times over the 12 DA subproblems and yields impressive improvements for some cases (*e.g.* $W \rightarrow A$ and $C \rightarrow D$) and a clear gain on average (1.5 points). The results on Office-Caltech DeCAF6 features show the effectiveness of our method on deep learning features. MLOT outperforms by 1.8 the second best method (here OT). On the Office31 dataset, the best results are obtained by SA. MLOT is still very competitive and outperforms OTDA by 0.9 point on average.

As already mentioned, cross-validating the hyperparameters in unsupervised DA is key and is still an open problem, since we do not have access to labels from the target domain. We performed an experimental comparison to show how the cross-validation method used in this paper behaves when compared to a scenario where we would use the actual labels of the target examples. Table 2 reports the gap between the optimal hyperparameters and those obtained by our method inspired from [Zhong *et al.*, 2010]. We can see

[1] We could not run MEDA [Wang *et al.*, 2018] because the implementation requires the use of a proprietary software.

[2] https://github.com/Hv0nnus/MLOT

	Dataset	NA	LMNN	SA	CORAL	TCA	OT	OTDA	OTDA_p	JDOT	MLOT
SURF	A→C	26.0	40.3	40.2±0.2	25.4	40.0	33.9	40.2	39.4±0.5	39.9	**42.3**±0.6
	A→D	25.5	36.9	39.3±2.5	26.8	31.8	30.6	40.1	39.6±1.1	37.6	**40.8**±0.8
	A→W	29.8	38.0	39.9±1.3	26.8	**41.7**	32.5	37.3	39.8±0.9	38.0	41.3±1.1
	C→A	23.7	46.0	41.3±1.1	23.6	39.8	41.0	**52.7**	48.5±0.7	48.1	51.5±0.8
	C→D	25.5	45.9	45.4±1.2	26.1	44.6	36.9	47.8	51.4±1.4	49.7	**52.2**±1.3
	C→W	25.8	41.7	36.6±1.1	23.7	36.9	28.1	**46.4**	45.8±1.4	43.4	45.9±0.8
	D→A	28.5	31.1	35.4±1.0	28.8	32.9	29.3	32.4	37.8±1.0	32.8	**37.8**±0.7
	D→C	26.3	30.7	32.3±0.6	30.0	31.5	31.7	32.0	33.5±0.7	31.7	**34.4**±0.5
	D→W	63.4	77.3	88.5±1.1	84.4	84.7	**88.8**	**88.8**	87.5±1.2	82.7	87.8±0.7
	W→A	23.0	32.3	32.6±0.5	26.2	29.4	34.1	33.7	37.6±0.6	37.6	**38.0**±0.8
	W→C	19.9	30.4	29.0±0.6	22.6	29.2	30.1	**34.1**	33.3±0.5	33.1	33.2±0.6
	W→D	59.2	86.6	89.5±1.0	84.1	91.7	89.2	**92.4**	91.8±1.2	89.8	90.8±0.8
	AVG	31.4	44.8	45.8±1.0	35.7	44.5	42.2	48.2	48.8±0.9	47.0	**49.7**±0.8
DeCAF6	AVG	71.0	79.4	83.7±0.5	77.2	83.4	83.9	83.2	82.6±0.5	78.2	**84.7**±0.3
Office31	AVG	64.3	64.7	**66.5**±0.2	64.1	64.1	65.3	65.3	65.2±0.1	64.4	66.2±0.1
All datasets	AVG	53.8	62.6	65.1 ± 0.6	58.0	64.0	63.5	65.6	65.6 ± 0.6	63.0	**67.0** ± 0.5

Table 1: Accuracy of all the methods on 3 different types of features. The best method for each dataset is in bold.

OT	TCA	LMNN	SA	JDOT	OTDA	OTDA_p	MLOT
45.3	45.3	47.9	47.4	48.5	52.8	54	**55.1**
42.2	44.5	44.8	45.8	47.0	48.2	48.8	**49.7**

Table 2: Accuracy comparison on Office-Caltech (SURF features) between a cross-validation method that uses the true target labels (first line) and the cross-validation method used in this paper that exploits pseudo-labels in the unsupervised DA setting (second line). CORAL and NA are excluded as they do not have hyperparameter.

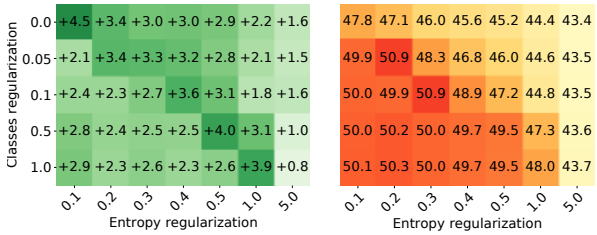

Figure 4: Absolute difference between the mean accuracy of MLOT and OTDA (on the left). The rows (resp. columns) correspond to the results with different values of the regularization parameter λ_c (resp. λ_e) on the entire Office-Caltech dataset with SURF features. On the right, accuracy of MLOT for each pair of parameters.

that the ranking of the methods is preserved, even though this experiment shows that there is still room for improving the way we may tune the parameters in unsupervised DA. Note that the other datasets show similar behaviors. To show the specific gain brought by MLOT compared to OTDA, we performed a last experiment where we set the hyper-parameters of MLOT as follows: $\lambda_l = 1$, $N = 10$, $margin_{LMNN} = 10$, $d = 70$; and we tune λ_e and λ_c. The results are reported in Figure 4. It is worth noticing that whatever the set of parameters, MLOT always yields better accuracy, which confirms the interest of learning a metric and using the PCA to initialize L_s and L_t. Note that when the entropy term has more importance (last column), the difference between the two methods is smaller because γ tends to be uniform. When the class regularization is set to 0 (first row), OTDA becomes similar to OT (Sinkhorn algorithm). This shows the effectiveness of MLOT even without the class regularization. However, the performances drop without this supervised information which tends to show that the metric learned and the class regularization are complementary. Notice that the results of MLOT are quite good for many values of the entropy and class regularizations (Figure 4 on the right). The best performance is 50.9 which is better than the result obtained by the cross-validation method. Once again, this is an evidence about the difficulty of tuning parameters in unsupervised DA.

6 Conclusion

We proposed in this paper a new Domain Adaptation (DA) method, called MLOT, benefiting from both Metric Learning (ML) and Optimal Transport (OT). Dedicated to address problems in the complex unsupervised DA setting, MLOT jointly learns a good metric and the optimal transportation plan. A theoretical study has driven the design of MLOT. We derived a bound on the target error which prompts us to learn the ground metric involved in the Wasserstein distance. The experimental study has shown very competitive results and a significant improvement compared to OTDA, the first method coupling OT and DA. Although deep learning is not at the core of this paper, note that we designed a differentiable version of MLOT using the PyTorch framework. While the first results, with pre-extracted features, did not bring any performance boost, this implementations opens the door to a full end-to-end model for Wasserstein-based domain adaptation.

Acknowledgements

This paper is part of the TADALoT Project funded by the region Auvergne-Rhône-Alpes (France) with the Pack Ambition Recherche (2017, 17 011047 01).

References

[Bay *et al.*, 2006] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[Bellet *et al.*, 2015] Aurélien Bellet, Amaury Habrard, and Marc Sebban. Metric learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2015.

[Ben-David *et al.*, 2007] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *NIPS*, 2007.

[Bhushan Damodaran *et al.*, 2018] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *ECCV*, 2018.

[Bolley *et al.*, 2007] François Bolley, Arnaud Guillin, and Cédric Villani. Quantitative concentration inequalities for empirical measures on non-compact spaces. *PTRF*, 2007.

[Courty *et al.*, 2017a] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation f or domain adaptation. In *NIPS*, 2017.

[Courty *et al.*, 2017b] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *PAMI*, 2017.

[Cuturi and Avis, 2014] Marco Cuturi and David Avis. Ground metric learning. *JMLR*, 2014.

[Cuturi, 2013] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013.

[Davis *et al.*, 2007] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007.

[Deshpande *et al.*, 2019] Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David A. Forsyth, and Alexander G. Schwing. Max-sliced wasserstein distance and its use for gans. *CoRR*, 2019.

[Donahue *et al.*, 2014] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[Fernando *et al.*, 2013] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.

[Genevay *et al.*, 2018] Aude Genevay, Gabriel Peyre, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *AISTATS*, 2018.

[Gong *et al.*, 2012] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[Halko *et al.*, 2011] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 2011.

[Kantorovich, 1942] L.V. Kantorovich. On the transfer of masses. *Dokl. Acad. Nauk. USSR37, 7–8*, 1942.

[Mansour *et al.*, 2009] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv:0902.3430*, 2009.

[Pan *et al.*, 2011] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 2011.

[Paty and Cuturi, 2019] François-Pierre Paty and Marco Cuturi. Subspace robust wasserstein distances. *arXiv preprint arXiv:1901.08949*, 2019.

[Redko *et al.*, 2017] Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical analysis of domain adaptation with optimal transport. In *ECML PKDD*, 2017.

[Redko *et al.*, 2019a] Ievgen Redko, Amaury Habrard, and Marc Sebban. On the analysis of adaptability in multi-source domain adaptation. *Machine Learning*, 2019.

[Redko *et al.*, 2019b] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. *Advances in Domain Adaptation Theory*. Elsevier, 2019.

[Saenko *et al.*, 2010] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[Shen *et al.*, 2017] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. *arXiv preprint arXiv:1707.01217*, 2017.

[Sun and Saenko, 2016] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.

[Sun *et al.*, 2016] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016.

[Villani, 2008] Cédric Villani. *Optimal transport: old and new*. Springer Science & Business Media, 2008.

[Wang and Deng, 2018] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 2018.

[Wang *et al.*, 2018] Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu. Visual domain adaptation with manifold embedded distribution alignment. In *ACM-MM*, 2018.

[Weinberger and Saul, 2009] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 2009.

[Zhong *et al.*, 2010] Erheng Zhong, Wei Fan, Qiang Yang, Olivier Verscheure, and Jiangtao Ren. Cross validation framework to choose amongst models and datasets for transfer learning. In *ECML PKDD*, 2010.