Optimal Transport for Domain Adaptation

Nicolas Courty, Rémi Flamary, Devis Tuia, *Senior Member, IEEE*,
Alain Rakotomamonjy, *Member, IEEE*

Abstract—Domain adaptation is one of the most challenging tasks of modern data analytics. If the adaptation is done correctly, models built on a specific data representation become more robust when confronted to data depicting the same classes, but described by another observation system. Among the many strategies proposed, finding domain-invariant representations has shown excellent properties, in particular since it allows to train a unique classifier effective in all domains. In this paper, we propose a regularized unsupervised optimal transportation model to perform the alignment of the representations in the source and target domains. We learn a transportation plan matching both PDFs, which constrains labeled samples of the same class in the source domain to remain close during transport. This way, we exploit at the same time the labeled samples in the source and the distributions observed in both domains. Experiments on toy and challenging real visual adaptation examples show the interest of the method, that consistently outperforms state of the art approaches. In addition, numerical experiments show that our approach leads to better performances on domain invariant deep learning features and can be easily adapted to the semi-supervised case where few labeled samples are available in the target domain.

Index Terms—Unsupervised Domain Adaptation, Optimal Transport, Transfer Learning, Visual Adaptation, Classification.

1 Introduction

Modern data analytics are based on the availability of large volumes of data, sensed by a variety of acquisition devices and at high temporal frequency. But this large amounts of heterogeneous data also make the task of learning semantic concepts more difficult, since the data used for learning a decision function and those used for inference tend not to follow the same distribution. Discrepancies (also known as drift) in data distribution are due to several reasons and are application-dependent. In computer vision, this problem is known as the visual adaptation domain problem, where domain drifts occur when changing lighting conditions, acquisition devices, or by considering the presence or absence of backgrounds. In speech processing, learning from one speaker and trying to deploy an application targeted to a wide public may also be hindered by the differences in background noise, tone or gender of the speaker. In remote sensing image analysis, one would like to leverage from labels defined over one city image to classify the land occupation of another city. The drifts observed in the probability density function (PDF) of remote sensing images are caused by variety of factors: different corrections for atmospheric scattering, daylight conditions at the hour of acquisition or even slight changes in the chemical composition of the materials.

For those reasons, several works have coped with these drift problems by developing learning methods able to transfer knowledge from a source domain to a target domain for which data have different PDFs. Learning in this PDF discrepancy context is denoted as the domain adaptation problem [37]. In this work, we address the most difficult variant of this problem, denoted as **unsupervised domain adaptation**, where data labels are only available in the source domain. We tackle this problem by assuming that the effects of the drifts can be reduced if data undergo a phase of *adaptation* (typically, a non-linear mapping) where both domains look more alike.

Several theoretical works [2], [36], [22] have emphasized the role played by the divergence between the data probability distribution functions of the domains. These works have led to a principled way of solving the domain adaptation problem: transform data so as to make their distributions "closer", and use the label information available in the source domain to learn a classifier in the transformed domain, which can be applied to the target domain. Our work follows the same intuition and proposes a transformation of the source data that fits **a least effort principle**, *i.e.* an effect that is minimal with respect to a transformation cost or metric. In this sense, the adaptation problem boils down to: *i)* finding a transformation of the input data matching the source and target distributions and then *ii)* learning a new classifier from the transformed source samples. This process is depicted in Figure 1. In this paper, we advocate a solution for finding this transformation based on *optimal transport*.

Optimal Transport (OT) problems have recently raised interest in several fields, in particular because OT theory can be used for computing distances between probability distributions. Those distances, known under several names in the literature (Wasserstein, Monge-Kantorovich or Earth Mover distances) have important properties: *i)* They can be evaluated directly on empirical estimates of the distribu-

Manuscript received January 2015; revised September 2015.

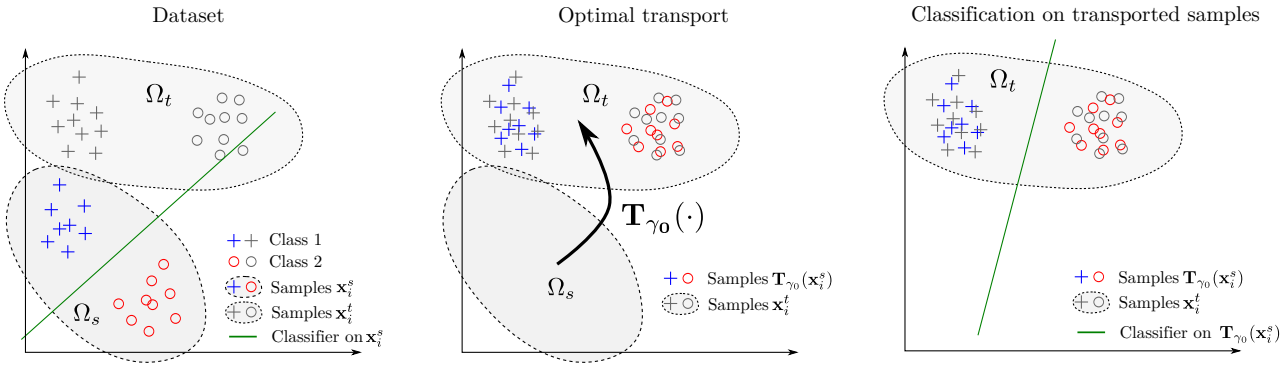


Fig. 1: Illustration of the proposed approach for domain adaptation. (left) dataset for training, *i.e.* source domain, and testing, *i.e.* target domain. Note that a classifier estimated on the training examples clearly does not fit the target data. (middle) a data dependent transportation map $\mathbf{T}_{\gamma 0}$ is estimated and used to transport the training samples onto the target domain. Note that this transformation is usually not linear. (right) the transported labeled samples are used for estimating a classifier in the target domain.

tions without having to smoothen them using non-parametric or semi-parametric approaches; *ii)* By exploiting the geometry of the underlying metric space, they provide meaningful distances even when the supports of the distributions do not overlap. Leveraging from these properties, we introduce a novel framework for unsupervised domain adaptation, which consists in learning an optimal transportation based on empirical observations. In addition, we propose several regularization terms that favor learning of better transformations *w.r.t.* the adaptation problem. They can either encode class information contained in the source domain or promote the preservation of neighborhood structures. An efficient algorithm is proposed for solving the resulting regularized optimal transport optimization problem. Finally, this framework can also easily be extended to the semi-supervised case, where few labels are available in the target domain, by a simple and elegant modification in the optimal transport optimization problem.

The remainder of this Section presents related works, while Section 2 formalizes the problem of unsupervised domain adaptation and discusses the use of optimal transport for its resolution. Section 3 introduces optimal transport and its regularized version. Section 4 presents the proposed regularization terms tailored to fit the domain adaptation constraints. Section 5 discusses algorithms for solving the regularized optimal transport problem efficiently. Section 6 evaluates the relevance of our domain adaptation framework through both synthetic and real-world examples.

1.1 Related works

Domain adaptation. Domain adaptation strategies can be roughly divided in two families, depending on whether they assume the presence of few labels in the target domain (semi-supervised DA) or not (unsupervised DA).

In the first family, methods which have been proposed include searching for projections that are discriminative in both domains by using inner products between source samples and transformed target samples [42], [32], [29]. Learning projections, for which labeled samples of the target domain fall on the correct side of a large margin classifier trained on the source data, have also been proposed [27]. Several works based on extraction of common features under pairwise constraints have also been introduced as domain adaptation strategies [26], [52], [47].

The second family tackles the domain adaptation problem assuming, as in this paper, that no labels are available in the target domain. Besides works dealing with sample reweighting [46], many works have considered finding a common feature representation for the two (or more) domains. Since the representation, or *latent space*, is common to all domains, projected labeled samples from the source domain can be used to train a classifier that is general [18], [38]. A common strategy is to propose methods that aim at finding representations in which domains match in some sense. For instance, adaptation can be performed by matching the means of the domains in the feature space [38], aligning the domains by their correlations [33] or by using pairwise constraints [51]. In most of these works, feature extraction is the key tool for finding a common latent space that embeds discriminative information shared by all domains.

Recently, the unsupervised domain adaptation problem has been revisited by considering strategies based on a gradual alignment of a feature representation. In [24], authors start from the hypothesis that domain adaptation can be better estimated when comparing gradual distortions. Therefore, they use intermediary projections of both domains along the Grassmannian geodesic connecting the source and target eigenvectors. In [23], [54], all sets of transformed intermediary domains are obtained by using

a geodesic-flow kernel. While these methods have the advantage of providing easily computable out-of-sample extensions (by projecting unseen samples onto the latent space eigenvectors), the transformation defined remains global and is applied in the same way to the whole target domain. An approach combining sample reweighting logic with representation transfer is found in [53], where authors extend the sample re-weighing to reproducing kernel Hilbert space through the use of surrogate kernels. The transformation achieved is again a global linear transformation that helps in aligning domains.

Our proposition strongly differs from those reviewed above, as it defines a local transformation for each sample in the source domain. In this sense, the domain adaptation problem can be seen as a graph matching problem [35], [10], [11] as each source sample has to be mapped on target samples under the constraint of marginal distribution preservation.

Optimal Transport and Machine Learning. The optimal transport problem has first been introduced by the French mathematician Gaspard Monge in the middle of the 19th century as a way to find a minimal effort solution to the transport of a given mass of dirt into a given hole. The problem reappeared in the middle of the 20th century in the work of Kantorovitch [30] and found recently surprising new developments as a polyvalent tool for several fundamental problems [49]. It was applied in a wide panel of fields, including computational fluid mechanics [3], color transfer between multiple images or morphing in the context of image processing [40], [20], [5], interpolation schemes in computer graphics [6], and economics, via matching and equilibriums problems [12].

Despite the appealing properties and application success stories, the machine learning community has considered optimal transport only recently (see, for instance, works considering the computation of distances between histograms [15] or label propagation in graphs [45]); the main reason being the high computational cost induced by the computation of the optimal transportation plan. However, new computing strategies have emerged [15], [17], [5] and made possible the application of OT distances in operational settings.

2 Optimal Transport and Application to Domain Adaptation

In this section, we present the general unsupervised domain adaptation problem and show how it can be addressed from an optimal transport perspective.

2.1 Problem and theoretical motivations

Let $\Omega \in \mathbb{R}^d$ be an input measurable space of dimension d and $\mathcal{C}$ the set of possible labels. $\mathcal{P}(\Omega)$ denotes the set of all probability measures over Ω. The standard learning paradigm assumes the existence of a set of training data $\mathbf{X}_s = \{\mathbf{x}_i^s\}_{i=1}^{N_s}$ associated with a set of class labels $\mathbf{Y}_s = \{y_i^s\}_{i=1}^{N_s}$, with $y_i^s \in \mathcal{C}$, and a testing set $\mathbf{X}_t = \{\mathbf{x}_i^t\}_{i=1}^{N_t}$ with unknown labels. In order to infer the set of labels $\mathbf{Y}_t$ associated with $\mathbf{X}_t$, one usually relies on an empirical estimate of the joint probability distribution $\mathbf{P}(\mathbf{x}, y) \in \mathcal{P}(\Omega \times \mathcal{C})$ from $(\mathbf{X}_s, \mathbf{Y}_s)$, and assumes that $\mathbf{X}_s$ and $\mathbf{X}_t$ are drawn from the same distribution $\mathbf{P}(\mathbf{x}) \in \mathcal{P}(\Omega)$.

2.2 Domain adaptation as a transportation problem

In domain adaptation problems, one assumes the existence of two distinct joint probability distributions $\mathbf{P}_s(\mathbf{x}^s, y)$ and $\mathbf{P}_t(\mathbf{x}^t, y)$, respectively related to a *source* and a *target* domains, noted as Ω_s and Ω_t. In the following, μ_s and μ_t are their respective marginal distributions over $\mathbf{X}$. We also denote f_s and f_t the true labeling functions, *i.e.* the Bayes decision functions in each domain.

At least one of the two following assumptions is generally made by most domain adaptation methods:

- **Class imbalance**: Label distributions are different in the two domains ($\mathbf{P}_s(y) \neq \mathbf{P}_t(y)$), but the conditional distributions of the samples with respect to the labels are the same ($\mathbf{P}_s(\mathbf{x}^s|y) = \mathbf{P}_t(\mathbf{x}^t|y)$);
- **Covariate shift**: Conditional distributions of the labels with respect to the data are equal ($\mathbf{P}_s(y|\mathbf{x}^s) = \mathbf{P}_t(y|\mathbf{x}^t)$, or equivalently $f_s = f_t = f$). However, data distributions in the two domains are supposed to be different ($\mathbf{P}_s(\mathbf{x}^s) \neq \mathbf{P}_t(\mathbf{x}^t)$). For the adaptation techniques to be effective, this difference needs to be small [2].

what is distribution? hard to understand

In real world applications, the drift occurring between the source and the target domains generally implies a change in both marginal and conditional distributions.

In our work, we assume that the domain drift is due to an unknown, possibly nonlinear transformation of the input space $\mathbf{T} : \boldsymbol{\Omega_s} \rightarrow \boldsymbol{\Omega_t}$. This transformation may have a physical interpretation (e.g. change in the acquisition conditions, sensor drifts, thermal noise, etc.). It can also be directly caused by the unknown process that generates the data. Additionnally, we also suppose that the transformation preserves the conditional distribution, *i.e.*

$$\mathbf{P}_s(y|\mathbf{x}^s) = \mathbf{P}_t(y|\mathbf{T}(\mathbf{x}^s)).$$

This means that the label information is preserved by the transformation, and the Bayes decision functions are tied through the equation $f_t(\mathbf{T}(\mathbf{x})) = f_s(\mathbf{x})$.

Another insight can be provided regarding the transformation $\mathbf{T}$. From a probabilistic point of view, $\mathbf{T}$ transforms the measure μ in its *image measure*, noted $\mathbf{T}\#\mu$, which is another probability measure over Ω_t satisfying

$$\mathbf{T}\#\mu(\mathbf{x}) = \mu(\mathbf{T}^{-1}(\mathbf{x})), \quad \forall \mathbf{x} \in \Omega_t \tag{1}$$

$\mathbf{T}$ is said to be a **transport map** or **push-forward** from μ_s to μ_t if $\mathbf{T}\#\mu_s = \mu_t$ (as illustrated in Figure 2.a). Under this assumption, $\mathbf{X}_t$ are drawn from the same PDF as $\mathbf{T}\#\mu_s$. This provides a principled way to solve the adaptation problem:

1) Estimate μ_s and μ_t from $\mathbf{X}_s$ and $\mathbf{X}_t$ (Equation (6))
2) Find a transport map $\mathbf{T}$ from μ_s to μ_t
3) Use $\mathbf{T}$ to transport labeled samples $\mathbf{X}_s$ and train a classifier from them.

Searching for $\mathbf{T}$ in the space of all possible transformations is intractable, and some restrictions need to be imposed. Here, we propose that $\mathbf{T}$ should be chosen so as to minimize a transportation cost $C(\mathbf{T})$ expressed as:

why this restriction is needed?

$$C(\mathbf{T}) = \int_{\Omega_s} c(\mathbf{x}, \mathbf{T}(\mathbf{x}))d\mu(\mathbf{x}), \tag{2}$$

where the cost function $c : \Omega_s \times \Omega_t \to \mathbb{R}^+$ is a distance function over the metric space Ω. $C(\mathbf{T})$ can be interpreted as the energy required to move a probability mass $\mu(\mathbf{x})$ from $\mathbf{x}$ to $\mathbf{T}(\mathbf{x})$.

The problem of finding such a transportation of minimal cost has already been investigated in the literature. For instance, the optimal transportation problem as defined by Monge is the solution of the following minimization problem:

$$\mathbf{T}_0 = \underset{\mathbf{T}}{\operatorname{argmin}} \int_{\Omega_s} c(\mathbf{x}, \mathbf{T}(\mathbf{x}))d\mu(\mathbf{x}), \quad \text{s.t. } \mathbf{T}\#\mu_s = \mu_t \tag{3}$$

The Kantorovitch formulation of the optimal transportation [30] is a convex relaxation of the above Monge problem. Indeed, let us define Π as the set of all probabilistic couplings $\in \mathcal{P}(\Omega_s \times \Omega_t)$ with marginals μ_s and μ_t. The Kantorovitch problem seeks for a general coupling $\boldsymbol{\gamma} \in \Pi$ between Ω_s and Ω_t:

$$\boldsymbol{\gamma}_0 = \underset{\boldsymbol{\gamma} \in \Pi}{\operatorname{argmin}} \int_{\Omega_s \times \Omega_t} c(\mathbf{x}^s, \mathbf{x}^t)d\boldsymbol{\gamma}(\mathbf{x}^s, \mathbf{x}^t) \tag{4}$$

In this formulation, $\boldsymbol{\gamma}$ can be understood as a joint probability measure with marginals μ_s and μ_t as depicted in Figure 2.b. $\boldsymbol{\gamma}_0$ is also known as **transportation plan** [43]. It allows to define the **Wasserstein distance** of order p between μ_s and μ_t. This distance is formalized as

$$\begin{aligned} W_p(\mu_s, \mu_t) &\overset{\text{def}}{=} \left(\inf_{\boldsymbol{\gamma} \in \Pi} \int_{\Omega_s \times \Omega_t} d(\mathbf{x}^s, \mathbf{x}^t)^p d\boldsymbol{\gamma}(\mathbf{x}^s, \mathbf{x}^t) \right)^{\frac{1}{p}} \\ &= \inf_{\boldsymbol{\gamma} \in \Pi} \left\{ \left(\mathop{\mathbb{E}}_{\mathbf{x}^s \sim \mu_s, \mathbf{x}^t \sim \mu_t} d(\mathbf{x}^s, \mathbf{x}^t)^p \right)^{\frac{1}{p}} \right\} \end{aligned} \tag{5}$$

where d is a distance and the corresponding cost function $c(\mathbf{x}^s, \mathbf{x}^t) = d(\mathbf{x}^s, \mathbf{x}^t)^p$. The Wasserstein distance is also known as the Earth Mover Distance in the computer vision community [41] and it defines a metric over the space of integrable squared probability measures.

In the remainder, we consider the squared ℓ_2 Euclidean distance as a cost function, $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$ for computing optimal transportation. As a consequence, we evaluate distances between measures according to the squared Wasserstein distance W_2^2 associated with the Euclidean distance $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$. The main rationale for this choice is that it experimentally provided the best result on average (as shown in the supplementary material). Nevertheless, other cost functions better suited to the nature of specific data can be considered, depending on the application at hand and the data representation, as discussed more in details in Section 3.4.

3 Regularized Discrete Optimal Transport

This section discusses the problem of optimal transport for domain adaptation. In the first part, we introduce the OT optimization problem on discrete empirical distributions. Then, we discuss a regularized variant of this discrete optimal transport problem. Finally, we address the question of how the resulting probabilistic coupling can be used for mapping samples from source to target domain.

3.1 Discrete optimal transport

When μ_s and μ_t are only accessible through discrete samples, the corresponding empirical distributions can be written as

$$\mu_s = \sum_{i=1}^{n_s} p_i^s \delta_{\mathbf{x}_i^s}, \quad \mu_t = \sum_{i=1}^{n_t} p_i^t \delta_{\mathbf{x}_i^t} \tag{6}$$

where $\delta_{\mathbf{x}_i}$ is the Dirac function at location $\mathbf{x}_i \in \mathbb{R}^d$. p_i^s and p_i^t are probability masses associated to the i-th sample and belong to the probability simplex, *i.e.* $\sum_{i=1}^{n_s} p_i^s = \sum_{i=1}^{n_t} p_i^t = 1$. It is straightforward to adapt the Kantorovich formulation of optimal transport problem to the discrete case. We denote $\mathcal{B}$ the set of probabilistic couplings between the two empirical distributions defined as:

$$\mathcal{B} = \left\{ \boldsymbol{\gamma} \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} \,|\, \boldsymbol{\gamma} \mathbf{1}_{\mathbf{n_t}} = \mu_{\mathbf{s}}, \boldsymbol{\gamma}^{\mathbf{T}} \mathbf{1}_{\mathbf{n_s}} = \mu_{\mathbf{t}} \right\} \tag{7}$$

where $\mathbf{1}_d$ is a d-dimensional vector of ones. The Kantorovitch formulation of the optimal transport [30] reads:

$$\boldsymbol{\gamma}_0 = \underset{\boldsymbol{\gamma} \in \mathcal{B}}{\operatorname{argmin}} \quad \langle \boldsymbol{\gamma}, \mathbf{C} \rangle_F \tag{8}$$

where $\langle ., . \rangle_F$ is the Frobenius dot product and $\mathbf{C} \geq 0$ is the cost function matrix, whose term $C(i, j) = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ denotes the cost to move a probability mass from $\mathbf{x}_i^s$ to $\mathbf{x}_j^t$. As previously detailed, this cost was chosen as the squared Euclidean distance between the two locations, i.e. $C(i, j) = ||\mathbf{x}_i^s - \mathbf{x}_j^t||_2^2$.

Note that when $n_s = n_t = n$ and $\forall i, j \quad p_i^s = p_j^t = 1/n$, $\boldsymbol{\gamma}_0$ is simply a permutation matrix. In this case, the optimal transport problem boils down to

an optimal assignment problem. In the general case, it can be shown that γ_0 is a sparse matrix with at most $n_s + n_t - 1$ non zero entries, equating the rank of the constraint matrix expressing the two marginal constraints.

Problem (8) is a linear program and can be solved with combinatorial algorithms such as the simplex methods and its network variants (successive shortest path algorithms, Hungarian or relaxation algorithms). Yet, the computational complexity was shown to be $O((n_s + n_t)n_s n_t log(n_s + n_t))$ [1, p. 472, Th. 12.2] at best, which dampens the utility of the method when handling large datasets. However, the regularization scheme recently proposed by Cuturi [15] presented in the next section, allows a very fast computation of a transportation plan.

3.2 Regularized optimal transport

Regularization is a classical approach used for preventing overfitting when few samples are available for learning. It can also be used for inducing some properties on the solution. In the following, we discuss a regularization term recently introduced for optimal transport problem.

Cuturi [15] proposed to regularize the expression of the optimal transport problem by the entropy of the probabilistic coupling. The resulting information-theoretic regularized version of the transport γ_0^λ is the solution of the minimization problem:

$$\gamma_0^\lambda = \underset{\gamma \in \mathcal{B}}{\operatorname{argmin}} \langle \gamma, \mathbf{C} \rangle_F + \lambda \Omega_s(\gamma), \tag{9}$$

where $\Omega_s(\gamma) = \sum_{i,j} \gamma(i,j) \log \gamma(i,j)$ computes the negentropy of γ. The intuition behind this form of regularization is the following: since most elements of γ_0 should be zero with high probability, one can look for a smoother version of the transport, thus lowering its sparsity, by increasing its entropy. As a result, the optimal transport γ_0^λ will have a denser coupling between the distributions. $\Omega_s(\cdot)$ can also be interpreted as a Kullback-Leibler divergence $KL(\gamma \| \gamma_u)$ between the joint probability γ and a uniform joint probability $\gamma_u(i,j) = \frac{1}{n_s n_t}$. Indeed, by expanding this KL divergence, we have $KL(\gamma \| \gamma_u) = \log n_s n_t + \sum_{i,j} \gamma(i,j) \log \gamma(i,j)$. The first term is a constant *w.r.t.* γ, which means that we can equivalently use $KL(\gamma \| \gamma_u)$ or $\Omega_s(\gamma) = \sum_{i,j} \gamma(i,j) \log \gamma(i,j)$ in Equation (9).

Hence, as the parameter λ weighting the entropy-based regularization increases, the sparsity of γ_0^λ decreases and source points tend to distribute their probability masses toward more target points. When λ becomes very large ($\lambda \to \infty$), the OT solution of Equation (9) converges toward $\gamma_0^\lambda(i,j) \to \frac{1}{n_s n_t}, \forall i,j$.

Another appealing outcome of the regularized OT formulation given in Equation (9) is the derivation of a computationally efficient algorithm based on Sinkhorn-Knopp's scaling matrix approach [31]. This efficient algorithm will also be a key element in our methodology presented in Section 4.

3.3 OT-based mapping of the samples

In the context of domain adaptation, once the probabilistic coupling γ_0 has been computed, source samples have to be transported in the target domain. For this purpose, one can interpolate the two distributions μ_s and μ_t by following the geodesics of the Wasserstein metric [49, Chapter 7], parameterized by $t \in [0,1]$. This defines a new distribution $\hat{\mu}$ such that:

$$\hat{\mu} = \underset{\mu}{\operatorname{argmin}} \quad (1-t)W_2(\mu_s, \mu)^2 + tW_2(\mu_t, \mu)^2. \tag{10}$$

Still following Villani's book, one can show that for a squared ℓ_2 cost, this distribution boils down to:

$$\hat{\mu} = \sum_{i,j} \gamma_0(i,j) \delta_{(1-t)\mathbf{x}_i^s + t\mathbf{x}_j^t}. \tag{11}$$

Since our goal is to transport the source samples onto the target distribution, we are mainly interested in the case $t = 1$. For this value of t, the novel distribution $\hat{\mu}$ is a distribution with the same support of μ_t, since Equation (11) reduces to

$$\hat{\mu} = \sum_j \hat{p}_j^t \delta_{\mathbf{x}_j^t}. \tag{12}$$

with $\hat{p}_j^t = \sum_i \gamma_0(i,j)$. The weights $\hat{p}_j^t$ can be seen as the sum of probability mass coming from all samples $\{\mathbf{x}_i^s\}$ that is transferred to sample $\mathbf{x}_j^t$. Alternatively, $\gamma_0(i,j)$ also tells us how much probability mass of $\mathbf{x}_i^s$ is transferred to $\mathbf{x}_j^t$. We can exploit this information to compute a transformation of the source samples. This transformation can be conveniently expressed with respect to the target samples as the following **barycentric mapping**:

$$\widehat{\mathbf{x}_i^s} = \underset{\mathbf{x} \in \mathbb{R}^d}{\operatorname{argmin}} \quad \sum_j \gamma_0(i,j) c(\mathbf{x}, \mathbf{x}_j^t). \tag{13}$$

where $\mathbf{x}_i^s$ is a given source sample and $\widehat{\mathbf{x}_i^s}$ is its corresponding image. When the cost function is the squared ℓ_2 distance, this barycenter corresponds to a weighted average and the sample is mapped into the convex hull of the target samples. For all source samples, this barycentric mapping can therefore be expressed as:

$$\hat{\mathbf{X}}_s = \mathbf{T}_{\gamma_0}(\mathbf{X}_s) = \operatorname{diag}(\gamma_0 \mathbf{1}_{n_t})^{-1} \gamma_0 \mathbf{X}_t. \tag{14}$$

The inverse mapping from the target to the source domain can also be easily computed from γ_0^T. Interestingly, one can show [17, Eq. 8] that this transformation is a first order approximation of the true n_s Wasserstein barycenters of the target distributions. Also note that when marginals μ_s and μ_t are uniform, one can easily derive the barycentric mapping as a linear expression:

$$\hat{\mathbf{X}}_s = n_s \gamma_0 \mathbf{X}_t \quad \text{and} \quad \hat{\mathbf{X}}_t = n_t \gamma_0^\top \mathbf{X}_s \tag{15}$$

need example

为什么插值?

这种等价表示有什么用呢

常数的好处?

for the source and target samples. Finally, remark that if $\gamma_0(i,j) = \frac{1}{n_s n_t}, \forall i,j$, then each transported source point converges toward the center of mass of the target distribution that is $\frac{1}{n_t}\sum_j \mathbf{x}_j^t$. This occurs when $\lambda \to \infty$ in Equation (9).

3.4 Discussing optimal transport for domain adaptation

We discuss here the requirements and conditions of applicability of the proposed method.

Guarantees of recovery of the correct transformation. Our goal for achieving domain adaptation is to uncover the transformation that occurred between source and target distributions. While the family of transformation that an OT formulation can recover is wide, we provide a proof that, for some simple affine transformations of discrete distributions, our OT solution is able to match source and target examples exactly.

Theorem 3.1: Let μ^s and μ^t be two discrete distributions with n Diracs as defined in Equation (6). If the following conditions hold

1) The source samples in μ^s are $\mathbf{x}_i^s \in \mathbb{R}^d, \forall i \in 1,\dots,n$ such that $\mathbf{x}_i^s \neq \mathbf{x}_j^s$ if $i \neq j$.
2) All weights in the source and target distributions are $\frac{1}{n}$.
3) The target samples are defined as $\mathbf{x}_i^t = \mathbf{A}\mathbf{x}_i^s + \mathbf{b}$ *i.e.* an affine tranformation of the source samples.
4) $\mathbf{b} \in \mathbb{R}^d$ and $\mathbf{A} \in \mathcal{S}^+$ is a strictly positive definite matrix.
5) The cost function is $c(\mathbf{x}^s, \mathbf{x}^t) = \|\mathbf{x}^s - \mathbf{x}^t\|_2^2$.

then the solution $\mathbf{T}_0$ of the optimal transport problem (8) is so that $\mathbf{T}_0(\mathbf{x}_i^s) = \mathbf{A}\mathbf{x}_i^s + \mathbf{b} = \mathbf{x}_i^t \quad \forall i \in 1,\dots,n$.

In this case, we retrieve the exact affine transformation on the discrete samples, which means that the label information are fully preserved during transportation. Therefore, one can train a classifier on the mapped samples with no generalization loss. We provide a simple demonstration in the supplementary material.

Choosing the cost function. In this work, we have mainly considered a ℓ_2-based cost function. Let us now discuss the implication of using a different cost function in our framework. A number of norm-based distances have been investigated by mathematicians [49, p 972]. Other types of metrics can also be considered, such as Riemannian distances over a manifold [49, Part II], or learnt metrics [16]. Concave cost functions are also of particular use in real life problems [21]. Each different cost function will lead to a different OT plan $\boldsymbol{\gamma}_0$, but the cost itself does not impact the OT optimization problem, *i.e.* the solver is independent from the cost function. Nonetheless, since $c(\cdot,\cdot)$ defines the Wasserstein geodesic, the interpolation between domains defined in Equation (10) leads to a different trajectory (potentially non-unique). Equation (11), which corresponds to $c(\cdot,\cdot)$, is a squared ℓ_2 distance, so it does not hold anymore. Nevertheless, the solution of (10) for $t = 1$ does not depend on the cost c and one can still use the proposed barycentric mapping (13). For instance if the cost function is based on the ℓ_1 norm, the transported samples will be estimated using a component-wise weighted median. Unfortunately, for more complex cost functions, the barycentric mapping might be complex to estimate.

4 Class-regularization for domain adaptation

In this section we explore regularization terms that preserve label information and sample neighborhood during transportation. Finally, we discuss the semi-supervised case and show that label information in the target domain can be effectively included in he proposed model.

4.1 Regularizing the transport with class labels

Optimal transport, as it has been presented in the previous section, does not use any class information. However, and even if our goal is unsupervised domain adaptation, class labels are available in the source domain. This information is typically used only during the decision function learning stage, which follows the adaptation step. Our proposition is to take advantage of the label information for estimating a better transport. More precisely, we aim at penalizing couplings that match source samples with different labels to same target samples.

To this end, we propose to add a new term to the regularized optimal transport, leading to the following optimization problem:

$$\min_{\boldsymbol{\gamma}\in\mathcal{B}} \quad \langle \boldsymbol{\gamma}, \mathbf{C}\rangle_F + \lambda\Omega_s(\boldsymbol{\gamma}) + \eta\Omega_c(\boldsymbol{\gamma}), \tag{16}$$

where $\eta \geq 0$ and $\Omega_c(\cdot)$ is a class-based regularization term.

In this work, we propose and study two choices for this regularizer $\Omega_c(\cdot)$. The first is based on group sparsity and promotes a probabilistic coupling $\boldsymbol{\gamma}_0$ where a given target sample receives masses from source samples which have same labels. The second is based on graph Laplacian regularization and promotes a locally smooth and class-regular structure in the source transported samples.

4.1.1 Regularization with group-sparsity

With the first regularizer, our objective is to exploit label information in the optimal transport computation. We suppose that all samples in the source domain have labels. The main intuition underlying the use of this group-sparse regularizer is that we would like each target sample to receive masses only from source samples that have the same label. As a consequence, we expect that a given target sample will be involved

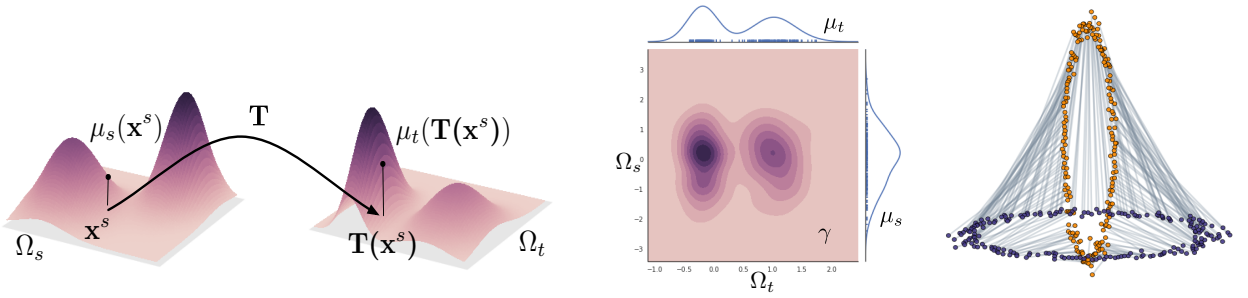


Fig. 2: Illustration of the optimal transport problem. (a) Monge problem over 2D domains. $\mathbf{T}$ is a push-forward from Ω_s to Ω_t. (b) Kantorovich relaxation over 1D domains: $\boldsymbol{\gamma}$ can be seen as a joint probability distribution with marginals μ_s and μ_t. (c) Illustration of the solution of the Kantorovich relaxation computed between two ellipsoidal distributions in 2D. The grey line between two points indicate a non-zero coupling between them.

in the representation of transported source samples as defined in Equation (14), but only for samples from the source domain of the same class. This behaviour can be induced by means of a group-sparse penalty on the columns of $\boldsymbol{\gamma}$.

This approach has been introduced in our preliminary work [14]. In that paper, we proposed a $\ell_p - \ell_1$ regularization term with $p < 1$ (mainly for algorithmic reasons). When applying a majoration-minimization technique on the $\ell_p - \ell_1$ norm, the problem can be cast as problem (9) and can be solved using the efficient Sinkhorn-Knopp algorithm at each iteration. However, this regularization term with $p < 1$ is non-convex and thus the proposed algorithm is guaranteed to converge only to local stationary points.

In this paper, we retain the convexity of the underlying problem and use the convex group-lasso regularizer $\ell_1 - \ell_2$ instead. This regularizer is defined as

$$\Omega_c(\boldsymbol{\gamma}) = \sum_j \sum_{cl} ||\boldsymbol{\gamma}(\mathcal{I}_{cl}, j)||_2, \tag{17}$$

where $||\cdot||_2$ denotes the ℓ_2 norm and $\mathcal{I}_{cl}$ contains the indices of rows in $\boldsymbol{\gamma}$ related to source domain samples of class cl. Hence, $\boldsymbol{\gamma}(\mathcal{I}_{cl}, j)$ is a vector containing coefficients of the jth column of $\boldsymbol{\gamma}$ associated to class cl. Since the jth column of $\boldsymbol{\gamma}$ is related to the jth target sample, this regularizer will induce the desired sparse representation in the target sample. Among other benefits, the convexity of the corresponding problem allows to use an efficient generic optimization scheme, presented in Section 5.

Ideally, with this regularizer we expect that the masses corresponding to each group of labels are matching samples of the source and target domains exclusively. Hence, for the domain adaptation problem to have a relevant solution, the distributions of labels are expected to be preserved in both the source and target distributions. We thus need to have $\mathbf{P}_s(y) = \mathbf{P}_t(y)$. This assumption, which is a classical assumption in the field of learning, is nevertheless a mild requirement since, in practice, small deviations of proportions do not prevent the method from working (see reference [48] for experimental results on this particular issue).

4.1.2 Laplacian regularization

This regularization term aims at preserving the data structure – approximated by a graph – during transport [20], [13]. Intuitively, we would like similar samples in the source domain to also be similar after transportation. Hence, denote as $\hat{\mathbf{x}}_i^s$ the transported source sample $\mathbf{x}_i^s$, with $\hat{\mathbf{x}}_i^s$ being linearly dependent on the transportation matrix $\boldsymbol{\gamma}$ through Equation (14). Now, given a positive symmetric similarity matrix $\mathbf{S_s}$ of samples in the source domain, our regularization term is defined as

$$\Omega_c(\boldsymbol{\gamma}) = \frac{1}{N_s^2} \sum_{i,j} S_s(i,j) \|\hat{\mathbf{x}}_i^s - \hat{\mathbf{x}}_j^s\|_2^2, \tag{18}$$

where $S_s(i,j) \geq 0$ are the coefficients of matrix $\mathbf{S}_s \in \mathbb{R}^{N_s \times N_s}$ that encodes similarity between pairs of source sample. In order to further preserve class structures, we can sparsify similarities for samples of different classes. In practice, we thus impose $S_s(i,j) = 0$ if $y_i^s \neq y_j^s$.

The above equation can be simplified when the marginal distributions are uniform. In that case, transported source samples can be computed according to Equation (15). Hence, $\Omega_c(\boldsymbol{\gamma})$ boils down to

$$\Omega_c(\boldsymbol{\gamma}) = \mathrm{Tr}(\mathbf{X}_t^\top \boldsymbol{\gamma}^\top \mathbf{L}_s \boldsymbol{\gamma} \mathbf{X}_t), \tag{19}$$

where $\mathbf{L}_s = \mathrm{diag}(\mathbf{S}_s \mathbf{1}) - \mathbf{S}_s$ is the Laplacian of the graph $\mathbf{S}_s$. The regularizer is therefore quadratic *w.r.t.* $\boldsymbol{\gamma}$.

The regularization terms (18) or (19) are defined based on the transported source samples. When a similarity information is also available in the target samples, for instance, through a similarity matrix $\mathbf{S}_t$, we can take advantage of this knowledge and a

symmetric Laplacian regularization of the form

$$\Omega_c(\boldsymbol{\gamma}) = (1-\alpha)\text{Tr}(\mathbf{X}_t^\top \boldsymbol{\gamma}^\top \mathbf{L}_s \boldsymbol{\gamma} \mathbf{X}_t) + \alpha \text{Tr}(\mathbf{X}_s^\top \boldsymbol{\gamma} \mathbf{L}_t \boldsymbol{\gamma}^\top \mathbf{X}_s) \tag{20}$$

can be used instead. In the above equation $\mathbf{L}_t = \text{diag}(\mathbf{S}_t \mathbf{1}) - \mathbf{S}_t$ is the Laplacian of the graph in the target domain and $0 \leq \alpha \leq 1$ is a trade-off parameter that weights the importance of each part of the regularization term. Note that, unlike the matrix $\mathbf{S}_s$, the similarity matrix $\mathbf{S}_t$ cannot be sparsified according to the class structure, since labels are generally not available for the target domain.

A regularization term similar to $\Omega_c(\boldsymbol{\gamma})$ has been proposed in [20] for histogram adaptation between images. However, the authors focused on displacements ($\hat{\mathbf{x}}_i^s - \mathbf{x}_i^s$) instead of on preserving the class structure of the transported samples.

4.2 Regularizing for semi-supervised domain adaptation

In semi-supervised domain adaptation, few labelled samples are available in the target domain [50]. Again, such an important information can be exploited by means of a novel regularization term to be integrated in the original optimal transport formulation. This regularization term is designed such that samples in the target domain should only be matched with samples in the source domain that have the same labels. It can be expressed as:

$$\Omega_{semi}(\boldsymbol{\gamma}) = \langle \boldsymbol{\gamma}, \mathbf{M} \rangle \tag{21}$$

where $\mathbf{M}$ is a $n_s \times n_t$ cost matrix, with $\mathbf{M}(i,j) = 0$ whenever $\mathbf{y}_i^s = \mathbf{y}_j^t$ (or j is a sample with unknown label) and $+\infty$ otherwise. This term has the benefit to be parameter free. It boils down to changing the original cost function $\mathbf{C}$, defined in Equation (8), by adding an infinite cost to undesired matches. Smooth versions of this regularization can be devised, for instance, by using a probabilistic confidence of target sample $\mathbf{x}_j^t$ to belong to class $\mathbf{y}_j^t$. Though appealing, we have not explored this latter option in this work. It is also noticeable that the Laplacian strategy in Equation (20) can also leverage on these class labels in the target domain through the definition of matrix $\mathbf{S}_t$.

5 Generalized conditional gradient for solving regularized OT problems

In this section, we discuss an efficient algorithm for solving optimization problem (16), that can be used with any of the proposed regularizers.

Firstly, we characterize the existence of a solution to the problem. We remark that regularizers given in Equations (17) and (18) are continuous, thus the objective function is continuous. Moreover, since the constraint set $\mathcal{B}$ is a convex, closed and bounded (hence compact) subset of $\mathbb{R}^d$, the objective function reaches its minimum on $\mathcal{B}$. In addition, if the regularizer is strictly convex that minimum is unique. This occurs for instance, for the Laplacian regularization in Equation (18).

Now, let us discuss algorithms for computing optimal transport solution of problem (16). For solving a similar problem with a Laplacian regularization term, Ferradans et al. [20] used a conditional gradient (CG) algorithm [4]. This approach is appealing and could be extended to our problem. It is an iterative scheme that guarantees any iterate to belong to $\mathcal{B}$, meaning that any of those iterates is a transportation plan. At each of these iterations, in order to find a feasible search direction, a CG algorithm looks for a minimizer of the objective function's linear approximation . Hence, at each iteration it solves a Linear Program (LP) that is presumably easier to handle than the original regularized optimal transport problem. Nevertheless, and despite existence of efficient LP solvers such as CPLEX or MOSEK, the dimensionality of the LP problem makes this LP problem hardly tractable, since it involves $n_s \times n_t$ variables.

In this work, we aim for a more scalable algorithm. To this end, we consider an approach based on a generalization of the conditional gradient algorithm [7] denoted as generalized conditional gradient (GCG).

The framework of the GCG algorithm addresses the general case of constrained minimization of composite functions defined as

$$\min_{\boldsymbol{\gamma} \in \mathcal{B}} \quad f(\boldsymbol{\gamma}) + g(\boldsymbol{\gamma}), \tag{22}$$

where $f(\cdot)$ is a differentiable and possibly non-convex function; $g(\cdot)$ is a convex, possibly non-differentiable function; $\mathcal{B}$ denotes any convex and compact subset of $\mathbb{R}^n$. As illustrated in Algorithm 1, all the steps of the GCG algorithm are exactly the same as those used for CG, except for the search direction part (Line 3). The difference is that GCG linearizes only part $f(\cdot)$ of the composite objective function, instead of the full objective function. This approach is justified when the resulting nonlinear optimization problem can be efficiently solved. The GCG algorithm has been shown by Bredies et al. [8] to converge towards a stationary point of Problem (22). In our case, since $g(\boldsymbol{\gamma})$ is differentiable, stronger convergence results can be provided (see supplementary material for a discussion on convergence rate and duality gap monitoring).

More specifically, for problem (16) we can set

$$f(\boldsymbol{\gamma}) = \langle \boldsymbol{\gamma}, \mathbf{C} \rangle_F + \eta \Omega_c(\boldsymbol{\gamma}) \quad \text{and} \quad g(\boldsymbol{\gamma}) = \lambda \Omega_s(\boldsymbol{\gamma}).$$

Supposing now that $\Omega_c(\boldsymbol{\gamma})$ is differentiable, step 3 of Algorithm 1 boils down to

$$\boldsymbol{\gamma}^\star = \underset{\boldsymbol{\gamma} \in \mathcal{B}}{\text{argmin}} \quad \langle \boldsymbol{\gamma}, \mathbf{C} + \eta \nabla \Omega_c(\boldsymbol{\gamma}^k) \rangle_F + \lambda \Omega_s(\boldsymbol{\gamma})$$

Interestingly, this problem is an entropy-regularized optimal transport problem similar to Problem (9) and

Algorithm 1 Generalized Conditional Gradient

1: Initialize $k = 0$ and $\boldsymbol{\gamma}^0 \in \mathcal{P}$
2: **repeat**
3: With $\mathbf{G} \in \nabla f(\boldsymbol{\gamma}^k)$, solve

$$\boldsymbol{\gamma}^\star = \underset{\boldsymbol{\gamma} \in \mathcal{B}}{\operatorname{argmin}} \quad \langle \boldsymbol{\gamma}, \mathbf{G} \rangle_F + g(\boldsymbol{\gamma})$$

4: Find the optimal step α^k

$$\alpha^k = \underset{0 \leq \alpha \leq 1}{\operatorname{argmin}} f(\boldsymbol{\gamma}^k + \alpha \Delta\boldsymbol{\gamma}) + g(\boldsymbol{\gamma}^k + \alpha \Delta\boldsymbol{\gamma})$$

with $\Delta\boldsymbol{\gamma} = \boldsymbol{\gamma}^* - \boldsymbol{\gamma}^k$
5: $\boldsymbol{\gamma}^{k+1} \leftarrow \boldsymbol{\gamma}^k + \alpha^k \Delta\boldsymbol{\gamma}$, set $k \leftarrow k + 1$
6: **until** Convergence

can be efficiently solved using the Sinkhorn-Knopp scaling matrix approach.

In our optimal transport problem, $\Omega_c(\boldsymbol{\gamma})$ is instantiated by the Laplacian or the group-lasso regularization term. The former is differentiable whereas the group-lasso is not when there exists a class cl and an index j for which $\boldsymbol{\gamma}(\mathcal{I}_{cl}, j)$ is a vector of 0. However, one can note that if the iterate $\boldsymbol{\gamma}^k$ is so that $\boldsymbol{\gamma}^k(\mathcal{I}_{cl}, j) \neq 0 \; \forall cl, \forall j$, then the same property holds for $\boldsymbol{\gamma}^{k+1}$. This is due to the exponentiation occurring in the Sinkhorn-Knopp algorithm used for the entropy-regularized optimal transport problem. This means that if we initialize $\boldsymbol{\gamma}^0$ so that $\boldsymbol{\gamma}^0(\mathcal{I}_{cl}, j) \neq 0$, then $\Omega_c(\boldsymbol{\gamma}^k)$ is always differentiable. Hence, our GCG algorithm can also be applied to the group-lasso regularization, despite its non-differentiability in 0.

6 Numerical Experiments

In this section, we study the behavior of four different versions of optimal transport applied to DA problem. In the rest of the section, **OT-exact** is the original transport problem (8), **OT-IT** the Information theoretic regularized one (9), and the two proposed class-based regularized ones are denoted **OT-GL** and **OT-Laplace**, corresponding respectively to the group-lasso (Equation (17)) and Laplacian (Equation (18)) regularization terms. We also present some results with our previous class-label based regularizer built upon an $\ell_p - \ell_1$ norm: **OT-LpL1** [14].

6.1 Two moons: simulated problem with controllable complexity

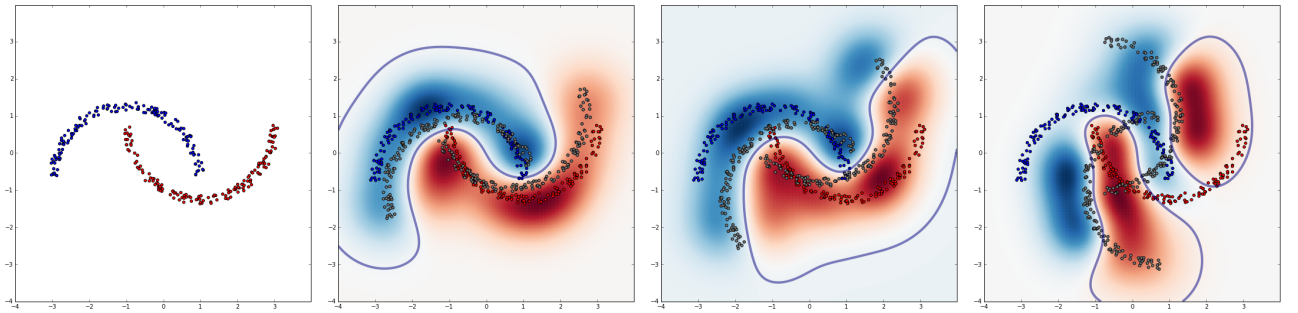
In the first experiment, we consider the same toy example as in [22]. The simulated dataset consists of two domains: for the source, the standard two entangled moons data, where each moon is associated to a specific class (See Figure 3(a)). The target domain is built by applying a rotation to the two moons, which allows to consider an adaptation problem with an increasing difficulty as a function of the rotation angle. This example is notably interesting because the corresponding problem is clearly non-linear, and because the input dimensionality is small, 2, which leads to poor performances when applying methods based on subspace alignment (*e.g.* [23], [34]).

Target rotation angle	10°	20°	30°	40°	50°	70°	90°
SVM (no adapt.)	0	0.104	0.24	0.312	0.4	0.764	0.828
DASVM [9]	0	0	0.259	0.284	0.334	0.747	0.82
PBDA [22]	0	0.094	0.103	0.225	0.412	0.626	0.687
OT-exact	0	0.028	0.065	0.109	0.206	0.394	0.507
OT-IT	0	0.007	0.054	0.102	0.221	0.398	0.508
OT-GL	0	0	0	0.013	0.196	0.378	0.508
OT-Laplace	0	0	0.004	0.062	0.201	0.402	0.524

TABLE 1: Mean error rate over 10 realizations for the two moons simulated example.

We follow the same experimental protocol as in [22], thus allowing for a direct comparison with the state-of-the-art results presented therein. The source domain is composed of two moons of 150 samples each. The target domain is also sampled from these two shapes, with the same number of examples. Then, the generalization capability of our method is tested over a set of 1000 samples that follow the same distribution as the target domain. The experiments are conducted 10 times, and we consider the mean classification error as comparison criterion. As a classifier, we used a SVM with a Gaussian kernel, whose parameters were set by 5-fold cross-validation. We compare the adaptation results with two state-of-the-art methods: the DA-SVM approach [9] and the more recent PBDA [22], which has proved to provide competitive results over this dataset.

Results are reported in Table 1. Our first observation is that all the methods based on optimal transport behave better than the state-of-the-art methods, in particular for low rotation angles, where results indicate that the geometrical structure is better preserved through the adaptation by optimal transport. Also, for large angle (*e.g.* $90°$), the final score is also significantly better than other state-of-the-art method, but falls down to a 0.5 error rate, which is natural since in this configuration a transformation of $-90°$, implying an inversion of labels, would have led to similar empirical distributions. This clearly shows the capacity of our method to handle large domain transformations. Adding the class-label information into the regularization also clearly helps for the mid-range angle values, where the adaptation shows nearly optimal results up to angles $< 40°$. For the strongest deformation ($> 70°$ rotation), no clear winner among the OT methods can be found. We think that, regardless of the amount and type of regularization chosen, the classification of test samples becomes too much tributary of the training samples. These ones mostly come from the denser part of μ_s and as a consequence, the less dense parts of this PDF are not satisfactorily transported. This behavior can be seen in Figure 3d.



(a) source domain (b) rotation=20° (c) rotation=40° (d) rotation=90°

Fig. 3: Illustration of the classification decision boundary produced by **OT-Laplace** over the two moons example for increasing rotation angles. The source domain is represented as coloured points. The target domain is depicted as points in grey (best viewed with colors).

6.2 Visual adaptation datasets

We now evaluate our method on three challenging real world vision adaptation tasks, which have attracted a lot of interest in recent computer vision literature [39]. We start by presenting the datasets, then the experimental protocol, and finish by providing and discussing the results obtained.

6.2.1 Datasets

Three types of image recognition problems are considered: digits, faces and miscellaneous objects recognition. This choice of datasets was already featured in [34]. A summary of the properties of each domain considered in the three problems is provided in Table 2. An illustration of some examples of the different domains for a particular class is shown in Figure 4.

Digit recognition. As source and target domains, we use the two digits datasets USPS and MNIST, that share 10 classes of digits (single digits $0-9$). We randomly sampled $1,800$ and $2,000$ images from each original dataset. The MNIST images are resized to the same resolution as that of USPS (16×16). The grey levels of all images are then normalized to obtain a final common feature space for both domains.

Face recognition. In the face recognition experiment, we use the PIE ("Pose, Illumination, Expression") dataset, which contains 32×32 images of 68 individuals taken under various pose, illumination and expressions conditions. The 4 experimental domains are constructed by selecting 4 distinct poses: PIE05 (C05, left pose), PIE07 (C07, upward pose), PIE09 (C09, downward pose) and PIE29 (C29, right pose). This allows to define 12 different adaptation problems with increasing difficulty (the most challenging being the adaptation from right to left poses). Let us note that each domain has a strong variability for each class due to illumination and expression variations.

Object recognition. We used the Caltech-Office dataset [42], [24], [23], [54], [39]. The dataset contains images coming from four different domains: *Amazon* (online merchant), the *Caltech-256* image collection [25], *Webcam* (images taken from a webcam) and *DSLR* (images taken from a high resolution digital SLR camera). The variability of the different domains come from several factors: presence/absence of background, lightning conditions, noise, etc. We consider two feature sets:

- SURF descriptors as described in [42], used to transform each image into a 800 bins histogram. These histograms are subsequently normalized and reduced to standard scores.
- two DeCAF deep learning features sets [19]: these features are extracted as the sparse activation of the neurons from the fully connected 6th and 7th layers of a convolutional network trained on imageNet and then fine tuned on the visual recognition tasks considered here. As such, they form vectors with 4096 dimensions.

Problem	Domains	Dataset	# Samples	# Features	# Classes	Abbr.
Digits	USPS	USPS	1800	256	10	U
	MNIST	MNIST	2000	256	10	M
Faces	PIE05	PIE	3332	1024	68	P1
	PIE07	PIE	1629	1024	68	P2
	PIE09	PIE	1632	1024	68	P3
	PIE29	PIE	1632	1024	68	P4
Objects	Calltech	Calltech	1123	800|4096	10	C
	Amazon	Office	958	800|4096	10	A
	Webcam	Office	295	800|4096	10	W
	DSLR	Office	157	800|4096	10	D

TABLE 2: Summary of the domains used in the visual adaptation experiment

6.2.2 Experimental setup

Following [23], the classification is conducted using a 1-Nearest Neighbor (1NN) classifier, which has the advantage of being parameter free. In all experiments, 1NN is trained with the adapted source data, and evaluated over the target data to provide a classification accuracy score. We compare our optimal transport solutions to the following baseline methods that are particularly well adapted for image classification:

- **1NN** is the original classifier without adaptation and constitutes a baseline for all experiments;

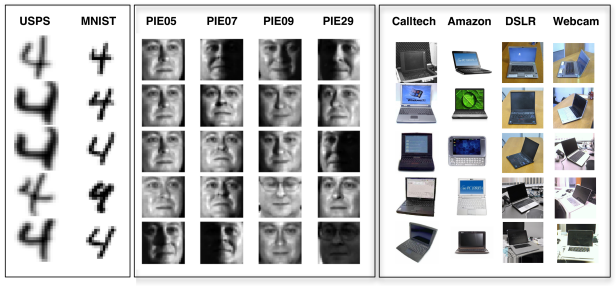


Fig. 4: Examples from the datasets used in the visual adaptation experiment. 5 random samples from one class are given for all the considered domains.

- **PCA**, which consists in applying a projection on the first principal components of the joint source/target distribution (estimated from the concatenation of source and target samples);
- **GFK**, Geodesic Flow Kernel [23];
- **TSL**, Transfer Subspace Learning [44], which operates by minimizing the Bregman divergence between the domains embedded in lower dimensional spaces;
- **JDA**, Joint Distribution Adaptation [34], which extends the Transfer Component Analysis algorithm [38];

In unsupervised DA no target labels are available. As a consequence, it is impossible to consider a cross-validation step for the hyper-parameters of the different methods. However, and in order to compare the methods fairly, we follow the following protocol. For each source domain, a random selection of 20 samples per class (with the only exception of 8 for the DSLR dataset) is adopted. Then the target domain is equivalently partitioned in a validation and test sets. The validation set is used to obtain the best accuracy in the range of the possible hyper-parameters. The accuracy, measured as the percent of correct classification over all the classes, is then evaluated on the testing set, with the best selected hyper-parameters. This strategy normally prevents overfitting on the testing set. The experimentation is conducted 10 times, and the mean accuracy over all these realizations is reported.

We considered the following parameter range : for subspace learning methods (**PCA**,**TSL**, **GFK**, and **JDA**) we considered reduced k-dimensional spaces with $k \in \{10, 20, \dots, 70\}$. A linear kernel was chosen for all the methods with a kernel formulation. For the all methods requiring a regularization parameter, the best value was searched in $\lambda = \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. The λ and η parameters of our different regularizers (Equation (16)), are validated using the same search interval. In the case of the Laplacian regularization (**OT-Laplace**), $\mathbf{S}_t$ is a binary matrix which encodes a nearest neighbors graph with a 8-connectivity. For the source domain, $\mathbf{S}_s$ is filtered such that connections between elements of different classes are pruned. Finally, we set the α value Equation (20) to 0.5.

6.2.3 Results on unsupervised domain adaptation

Results of the experiment are reported in Table 3 where the best performing method for each domain adaptation problem is highlighted in bold. On average, all the OT-based domain adaptation methods perform better than the baseline methods, except in the case of the PIE dataset, where **JDA** outperforms the OT-based methods in 7 out of 12 domain pairs. A possible explanation is that the dataset contains a lot of classes (68), and the EM-like step of **JDA**, which allows to take into account the current results of classification on the target, is clearly leading to a benefit. We notice that **TSL**, which is based on a similar principle of distribution divergence minimization, almost never outperforms our regularized strategies, except on pair A→C. Among the different optimal transport strategies, **OT-Exact** leads to the lowest performances. **OT-IT**, the entropy regularized version of the transport, is substantially better than **OT-Exact**, but is still inferior to the class-based regularized strategies proposed in this paper. The best performing strategies are clearly **OT-GL** and **OT-Laplace** with a slight advantage for **OT-GL**. **OT-LpL1**, which is based on a similar regularization strategy as **OT-GL**, but with a different optimization scheme, has globally inferior performances, except on some pairs of domains (e.g. C→A) where it achieves better scores. On both digits and objects recognition tasks, **OT-GL** significantly outperforms the baseline methods.

In the next experiment (Table 4), we use the same experimental protocol on different features produced by the DeCAF deep learning architecture [19]. We report the results of the experiment conducted on the Office-Caltech dataset, with the **OT-IT** and **OT-GL** regularization strategies. For comparison purposes, **JDA** is also considered for this adaptation task. The results show that, even though the deep learning features yield naturally a strong improvement over the classical SURF features, the proposed OT methods are still capable of improving significantly the performances of the final classification (up to more than 20 points in some case, e.g. D→A or A→W). This clearly shows how OT has the capacity to handle non-stationarity in the distributions that the deep architecture has difficulty handling. We also note that using the features from the 7th layer instead of the 6th does not bring a strong improvement in the classification accuracy, suggesting that part of the work of the 7th layer is already performed by the optimal transport.

6.2.4 Semi-supervised domain adaptation

In this last experiment, we assume that few labels are available in the target domain. We thus benchmark

TABLE 3: Overall recognition accuracies in % obtained over all domains pairs using the SURF features. Maximum values for each pair is indicated in bold font.

Domains	1NN	PCA	GFK	TSL	JDA	OT-exact	OT-IT	OT-Laplace	OT-LpLq	OT-GL
U→M	39.00	37.83	44.16	40.66	54.52	50.67	53.66	57.42	**60.15**	57.85
M→U	58.33	48.05	60.96	53.79	60.09	49.26	64.73	64.72	68.07	**69.96**
mean	48.66	42.94	52.56	47.22	57.30	49.96	59.20	61.07	**64.11**	63.90
P1→P2	23.79	32.61	22.83	34.29	**67.15**	52.27	57.73	58.92	59.28	59.41
P1→P3	23.50	38.96	23.24	33.53	56.96	51.36	57.43	57.62	58.49	**58.73**
P1→P4	15.69	30.82	16.73	26.85	40.44	40.53	47.21	47.54	47.29	**48.36**
P2→P1	24.27	35.69	24.18	33.73	**63.73**	56.05	60.21	62.74	62.61	61.91
P2→P3	44.45	40.87	44.03	38.35	**68.42**	59.15	63.24	64.29	62.71	64.36
P2→P4	25.86	29.83	25.49	26.21	49.85	46.73	51.48	**53.52**	50.42	52.68
P3→P1	20.95	32.01	20.79	39.79	**60.88**	54.24	57.50	57.87	58.96	57.91
P3→P2	40.17	38.09	40.70	39.17	65.07	59.08	63.61	**65.75**	64.04	64.67
P3→P4	26.16	36.65	25.91	36.88	52.44	48.25	52.33	**54.02**	52.81	52.83
P4→P1	18.14	29.82	20.11	40.81	**46.91**	43.21	45.15	45.67	46.51	45.73
P4→P2	24.37	29.47	23.34	37.50	**55.12**	46.76	50.71	52.50	50.90	51.31
P4→P3	27.30	39.74	26.42	46.14	**53.33**	48.05	52.10	52.71	51.37	52.60
mean	26.22	34.55	26.15	36.10	**56.69**	50.47	54.89	56.10	55.45	55.88
C→A	20.54	35.17	35.29	45.25	40.73	30.54	37.75	38.96	**48.21**	44.17
C→W	18.94	28.48	31.72	37.35	33.44	23.77	31.32	31.13	38.61	**38.94**
C→D	19.62	33.75	35.62	39.25	39.75	26.62	34.50	36.88	39.62	**44.50**
A→C	22.25	32.78	32.87	**38.46**	33.99	29.43	31.65	33.12	35.99	34.57
A→W	23.51	29.34	32.05	35.70	36.03	25.56	30.40	30.33	35.63	**37.02**
A→D	20.38	26.88	30.12	32.62	32.62	25.50	27.88	27.75	36.38	**38.88**
W→C	19.29	26.95	27.75	29.02	31.81	25.87	31.63	31.37	33.44	**35.98**
W→A	23.19	28.92	33.35	34.94	31.48	27.40	37.79	37.17	37.33	**39.35**
W→D	53.62	79.75	79.25	80.50	**84.25**	76.50	80.00	80.62	81.38	84.00
D→C	23.97	29.72	29.50	31.03	29.84	27.30	29.88	31.10	31.65	**32.38**
D→A	27.10	30.67	32.98	36.67	32.85	29.08	32.77	33.06	37.06	**37.17**
D→W	51.26	71.79	69.67	77.48	80.00	65.70	72.52	76.16	74.97	**81.06**
mean	28.47	37.98	39.21	42.97	44.34	36.69	42.30	43.20	46.42	**47.70**

TABLE 4: Results of adaptation by optimal transport using DeCAF features.

	Layer 6				Layer 7			
Domains	DeCAF	JDA	OT-IT	OT-GL	DeCAF	JDA	OT-IT	OT-GL
C→A	79.25	88.04	88.69	**92.08**	85.27	89.63	91.56	**92.15**
C→W	48.61	79.60	75.17	**84.17**	65.23	79.80	82.19	**83.84**
C→D	62.75	84.12	83.38	**87.25**	75.38	85.00	85.00	**85.38**
A→C	64.66	81.28	81.65	**85.51**	72.80	82.59	84.22	**87.16**
A→W	51.39	80.33	78.94	**83.05**	63.64	83.05	81.52	**84.50**
A→D	60.38	**86.25**	85.88	85.00	75.25	85.50	**86.62**	85.25
W→C	58.17	**81.97**	74.80	81.45	69.17	79.84	81.74	**83.71**
W→A	61.15	90.19	80.96	**90.62**	72.96	90.94	88.31	**91.98**
W→D	97.50	**98.88**	95.62	96.25	98.50	**98.88**	98.38	91.38
D→C	52.13	81.13	77.71	**84.11**	65.23	81.21	82.02	**84.93**
D→A	60.71	91.31	87.15	**92.31**	75.46	91.92	92.15	**92.92**
D→W	85.70	**97.48**	93.77	96.29	92.25	**97.02**	96.62	94.17
mean	65.20	86.72	83.64	**88.18**	75.93	87.11	87.53	**88.11**

TABLE 5: Results of semi-supervised adaptation with optimal transport using the SURF features.

	Unsupervised + labels		Semi-supervised		
Domains	OT-IT	OT-GL	OT-IT	OT-GL	MMDT [28]
C→A	37.0 ± 0.5	41.4 ± 0.5	46.9 ± 3.4	47.9 ± 3.1	**49.4 ± 0.8**
C→W	28.5 ± 0.7	37.4 ± 1.1	64.8 ± 3.0	**65.0 ± 3.1**	63.8 ± 1.1
C→D	35.1 ± 1.7	44.0 ± 1.9	59.3 ± 2.5	**61.0 ± 2.1**	56.5 ± 0.9
A→C	32.3 ± 0.1	36.7 ± 0.2	36.0 ± 1.3	**37.1 ± 1.1**	36.4 ± 0.8
A→W	29.5 ± 0.8	37.8 ± 1.1	63.7 ± 2.4	**64.6 ± 1.9**	**64.6 ± 1.2**
A→D	36.9 ± 1.5	46.2 ± 2.0	57.6 ± 2.5	**59.1 ± 2.3**	56.7 ± 1.3
W→C	35.8 ± 0.2	36.5 ± 0.2	38.4 ± 1.5	**38.8 ± 1.2**	32.2 ± 0.8
W→A	39.6 ± 0.3	41.9 ± 0.4	47.2 ± 2.5	47.3 ± 2.5	**47.7± 0.9**
W→D	77.1 ± 1.8	**80.2 ± 1.6**	79.0 ± 2.8	79.4 ± 2.8	67.0 ± 1.1
D→C	32.7 ± 0.3	34.7 ± 0.3	35.5 ± 2.1	**36.8 ± 1.5**	34.1 ± 1.5
D→A	34.7 ± 0.3	37.7 ± 0.3	45.8 ± 2.6	46.3 ± 2.5	**46.9 ± 1.0**
D→W	81.9 ± 0.6	**84.5 ± 0.4**	83.9 ± 1.4	84.0 ± 1.5	74.1 ± 0.8
mean	41.8	46.6	54.8	**55.6**	52.5

our semi-supervised approach on SURF features extracted from the Office-Caltech dataset. We consider that only 3 labeled samples per class are at our disposal in the target domain. In order to disentangle the benefits of the labeled target samples brought by our optimal transport strategies from those brought by the classifier, we make a distinction between two cases: in the first one, denoted as "Unsupervised + labels", we consider that the label target samples are available only at the learning stage, after an unsupervised domain adaptation with optimal transport. In the second case, denoted as "semi-supervised", labels in the target domain are used to compute a new transportation plan, through the use of the proposed semi-supervised regularization term in Equation (21)).

Results are reported in Table 5. They clearly show the benefits of the proposed semi-supervised regularization term in the definition of the transportation plan. A comparison with the state-of-the-art method of Hoffman and colleagues [28] is also reported, and shows the competitiveness of our approach.

7 Conclusion

In this paper, we described a new framework based on optimal transport to solve the unsupervised domain adaptation problem. We proposed two regularization schemes to encode class-structure in the source domain during the estimation of the transportation

plan, thus enforcing the intuition that samples of the same class must undergo similar transformation. We extended this OT regularized framework to the semi-supervised domain adaptation case, i.e. the case where few labels are available in the target domain. Regarding the computational aspects, we suggested to use a modified version of the conditional gradient algorithm, the generalized conditional gradient splitting, which enables the method to scale up to real-world datasets. Finally, we applied the proposed methods on both synthetic and real world datasets. Results show that the optimal transportation domain adaptation schemes frequently outperform the competing state-of-the-art methods.

We believe that the framework presented in this paper will lead to a paradigm shift for the domain adaptation problem. Estimating a transport is much more general than finding a common subspace, but comes with the problem of finding a proper regularization term. The proposed class-based or Laplacian regularizers show very good performances, but we believe that other types of regularizer should be investigated. Indeed, whenever the transformation is induced by a physical process, one may want the transport map to enforce physical constraints. This can be included with dedicated regularization terms. We also plan to extend our optimal transport framework to the multi-domain adaptation problem, where the problem of matching several distributions can be cast as a multi-marginal optimal transport problem.

ACKNOWLEDGMENTS

This work was partly funded by the Swiss National Science Foundation under the grant PP00P2-150593 and by the CNRS PEPS Fascido program under the Topase project.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[2] S. Ben-David, T. Luu, T. Lu, and D. Pál, "Impossibility theorems for domain adaptation." in *Artificial Intelligence and Statistics Conference (AISTATS)*, 2010, pp. 129–136.

[3] J.-D. Benamou and Y. Brenier, "A computational fluid mechanics solution to the monge-kantorovich mass transfer problem," *Numerische Mathematik*, vol. 84, no. 3, pp. 375–393, 2000.

[4] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[5] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, "Sliced and radon Wasserstein barycenters of measures," *Journal of Mathematical Imaging and Vision*, vol. 51, pp. 22–45, 2015.

[6] N. Bonneel, M. van de Panne, S. Paris, and W. Heidrich, "Displacement interpolation using Lagrangian mass transport," *ACM Transaction on Graphics*, vol. 30, no. 6, pp. 158:1–158:12, 2011.

[7] K. Bredies, D. A. Lorenz, and P. Maass, "A generalized conditional gradient method and its connection to an iterative shrinkage method," *Computational Optimization and Applications*, vol. 42, no. 2, pp. 173–193, 2009.

[8] K. Bredies, D. Lorenz, and P. Maass, *Equivalence of a generalized conditional gradient method and the method of surrogate functionals*. Zentrum für Technomathematik, 2005.

[9] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A dasvm classification technique and a circular validation strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 770–787, May 2010.

[10] T. S. Caetano, T. Caelli, D. Schuurmans, and D. Barone, "Grapihcal models and point pattern matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1646–1663, 2006.

[11] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola, "Learning graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 1048–1058, 2009.

[12] G. Carlier, A. Oberman, and E. Oudet, "Numerical methods for matching for teams and Wasserstein barycenters," Inria, Tech. Rep. hal-00987292, 2014.

[13] M. Carreira-Perpinan and W. Wang, "LASS: A simple assignment model with laplacian smoothing," in *AAAI Conference on Artificial Intelligence*, 2014.

[14] N. Courty, R. Flamary, and D. Tuia, "Domain adaptation with regularized optimal transport," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2014.

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

[16] M. Cuturi and D. Avis, "Ground metric learning," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 533–564, Jan. 2014.

[17] M. Cuturi and A. Doucet, "Fast computation of Wasserstein barycenters," in *International Conference on Machine Learning (ICML)*, 2014.

[18] H. Daumé III, "Frustratingly easy domain adaptation," in *Ann. Meeting of the Assoc. Computational Linguistics*, 2007.

[19] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: a deep convolutional activation feature for generic visual recognition," in *International Conference on Machine Learning (ICML)*, 2014, pp. 647–655.

[20] S. Ferradans, N. Papadakis, J. Rabin, G. Peyré, and J.-F. Aujol, "Regularized discrete optimal transport," in *Scale Space and Variational Methods in Computer Vision, SSVM*, 2013, pp. 428–439.

[21] W. Gangbo and R. J. McCann, "The geometry of optimal transportation," *Acta Mathematica*, vol. 177, no. 2, pp. 113–161, 1996.

[22] P. Germain, A. Habrard, F. Laviolette, and E. Morvant, "A PAC-Bayesian Approach for Domain Adaptation with Specialization to Linear Classifiers," in *International Conference on Machine Learning (ICML)*, Atlanta, USA, 2013, pp. 738–746.

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

[24] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *International Conference on Computer Vision (ICCV)*, 2011, pp. 999–1006.

[25] G. Griffin, A. Holub, and P. Perona, "Caltech-256 Object Category Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2007-001, 2007.

[26] J. Ham, D. Lee, and L. Saul, "Semisupervised alignment of manifolds," in *10th International Workshop on Artificial Intelligence and Statistics*, R. G. Cowell and Z. Ghahramani, Eds., 2005, pp. 120–127.

[27] J. Hoffman, E. Rodner, J. Donahue, K. Saenko, and T. Darrell, "Efficient learning of domain invariant image representations," in *International Conference on Learning Representations (ICLR)*, 2013.

[28] ——, "Efficient learning of domain-invariant image representations," in *International Conference on Learning Representations (ICLR)*, 2013.

[29] I.-H. Jhuo, D. Liu, D. T. Lee, and S.-F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2168–2175.

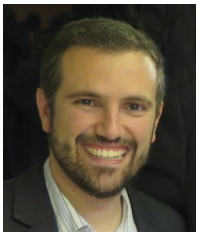
[30] L. Kantorovich, "On the translocation of masses," *C.R. (Doklady) Acad. Sci. URSS (N.S.)*, vol. 37, pp. 199–201, 1942.

[31] P. Knight, "The sinkhorn-knopp algorithm: Convergence and applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 1, pp. 261–275, 2008.

[32] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: domain adaptation using asymmetric kernel transforms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO, 2011.
[33] A. Kumar, H. Daumé III, and D. Jacobs, "Generalized multiview analysis: A discriminative latent space," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
[34] M. Long, J. Wang, G. Ding, J. Sun, and P. Yu, "Transfer feature learning with joint distribution adaptation," in *International Conference on Computer Vision (ICCV)*, Dec 2013, pp. 2200–2207.
[35] B. Luo and R. Hancock, "Structural graph matching using the em algorithm and singular value decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1120–1136, 2001.
[36] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," in *Conference on Learning Theory (COLT)*, 2009, pp. 19–30.
[37] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
[38] ——, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, pp. 199–210, 2011.
[39] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: an overview of recent advances," *IEEE Signal Processing Magazine*, vol. 32, no. 3, 2015.
[40] J. Rabin, G. Peyré, J. Delon, and M. Bernot, "Wasserstein barycenter and its application to texture mixing," in *Scale Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science, 2012, vol. 6667, pp. 435–446.
[41] Y. Rubner, C. Tomasi, and L. Guibas, "A metric for distributions with applications to image databases," in *International Conference on Computer Vision (ICCV)*, 1998, pp. 59–66.
[42] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European Conference on Computer Vision (ECCV)*, ser. LNCS, 2010, pp. 213–226.
[43] F. Santambrogio, "Optimal transport for applied mathematicians," *Birkäuser, NY*, 2015.
[44] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 929–942, July 2010.
[45] J. Solomon, R. Rustamov, G. Leonidas, and A. Butscher, "Wasserstein propagation for semi-supervised learning," in *International Conference on Machine Learning (ICML)*, 2014, pp. 306–314.
[46] M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Neural Information Processing Systems (NIPS)*, 2008.
[47] D. Tuia and G. Camps-Valls, "Kernel manifold alignment for domain adaptation," *PLoS One*, vol. 11, no. 2, p. e0148655, 2016.
[48] D. Tuia, R. Flamary, A. Rakotomamonjy, and N. Courty, "Multitemporal classification without new labels: a solution with optimal transport," in *8th International Workshop on the Analysis of Multitemporal Remote Sensing Images*, 2015.
[49] C. Villani, *Optimal transport: old and new*, ser. Grundlehren der mathematischen Wissenschaften. Springer, 2009.
[50] C. Wang, P. Krafft, and S. Mahadevan, "Manifold alignment," in *Manifold Learning: Theory and Applications*, Y. Ma and Y. Fu, Eds. CRC Press, 2011.
[51] C. Wang and S. Mahadevan, "Manifold alignment without correspondence," in *International Joint Conference on Artificial Intelligence (IJCAI)*, Pasadena, CA, 2009.
[52] ——, "Heterogeneous domain adaptation using manifold alignment," in *International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press, 2011, pp. 1541–1546.
[53] K. Zhang, V. W. Zheng, Q. Wang, J. T. Kwok, Q. Yang, and I. Marsic, "Covariate shift in Hilbert space: A solution via surrogate kernels," in *International Conference on Machine Learning (ICML)*, 2013.
[54] J. Zheng, M.-Y. Liu, R. Chellappa, and P. Phillips, "A Grassmann manifold-based domain adaptation approach," in *International Conference on Pattern Recognition (ICPR)*, Nov 2012, pp. 2095–2099.

Nicolas Courty is associate professor within University Bretagne-Sud since October 2004. He obtained his habilitation degree (HDR) in 2013. His main research objectives are data analysis/synthesis schemes, machine learning and visualization problems, with applications in computer vision, remote sensing and computer graphics. Visit http://people.irisa.fr/Nicolas.Courty/ for more information.

Rémi Flamary is Assistant Professor at Université Côte d'Azur (UCA) and a member of Lagrange Laboratory/Observatoire de la Côte d'Azur since 2012. He received a Dipl.-Ing. in electrical engineering and a M.S. degrees in image processing from the Institut National de Sciences Appliquées de Lyon in 2008 and a Ph.D. degree from the University of Rouen in 2011. His current research interest involve signal processing, machine learning and image processing.



Devis Tuia (S'07, M'09, SM'15) received the Ph.D. from University of Lausanne in 2009. He was a Postdoc at the University of Valéncia, the University of Colorado, Boulder, CO and EPFL Lausanne. Since 2014, he is Assistant Professor with the Department of Geography, University of Zurich. He is interested in algorithms for information extraction and data fusion of remote sensing images using machine learning. More info on http://devis.tuia.googlepages.com/

Alain Rakotomamonjy (M'15) is Professor in the Physics department at the University of Rouen since 2006. He obtained his Phd on Signal processing from the university of Orléans in 1997. His recent research activities deal with machine learning and signal processing with applications to brain-computer interfaces and audio applications. Alain serves as a regular reviewer for machine learning and signal processing journals.