Wasserstein Barycenter for Multi-Source Domain Adaptation

Eduardo Fernandes Montesuma
Universidade Federal do Ceará
Fortaleza, Brazil
eduardomontesuma@alu.ufc.br

Fred Maurice Ngolè Mboula
Université Paris-Saclay, Institut LIST, CEA
F-91120 Palaiseau, France
fred-maurice.ngole-mboula@cea.fr

Abstract

Multi-source domain adaptation is a key technique that allows a model to be trained on data coming from various probability distribution. To overcome the challenges posed by this learning scenario, we propose a method for constructing an intermediate domain between sources and target domain, the Wasserstein Barycenter Transport (WBT). This method relies on the barycenter on Wasserstein spaces for aggregating the source probability distributions. Once the sources have been aggregated, they are transported to the target domain using standard Optimal Transport for Domain Adaptation framework. Additionally, we revisit previous single-source domain adaptation tasks in the context of multi-source scenario. In particular, we apply our algorithm to object and face recognition datasets. Moreover, to diversify the range of applications, we also examine the tasks of music genre recognition and music-speech discrimination. The experiments show that our method has similar performance with the existing state-of-the-art.

1. Introduction

Standard data-driven algorithms are based upon the hypothesis that training and test data follows the same probability distribution. When this assumption does not hold, a case also known as distributional shift, these algorithms may suffer from performance degradation. In this case, many predictive models need to be re-trained on the new data, ignoring previous knowledge.

There are various examples of distributional shift in areas of application of machine learning, such as image [22][23][27], natural language [3][2], and speech [15][24] processing. This has been known in the literature as transfer learning.

Transfer learning may be further categorized according to the nature of distributional shift. First, the features distribution change between training and test sets, that is $P_s(X) \neq P_t(X)$. This case is commonly referred as domain adaptation [17]. Additionally, the shift may also occur on the labels distribution $P_s(Y) \neq P_t(Y)$, or in the conditional distribution $P_s(Y|X) \neq P_t(Y|X)$. These cases are known respectively as target and conditional shift.

In this paper, a focus is given to unsupervised domain adaptation. Hence, we refer to training and test data as coming from different domains, respectively a source $\mathcal{D}_s$ and target domain $\mathcal{D}_t$. According to [17], the goal of unsupervised domain adaptation is to help the learning of a predictive model on $\mathcal{D}_t$, using knowledge learned in $\mathcal{D}_s$, without using any labels in the target domain.

In addition, as first remarked by [3], the distance between probability distributions play an important role on the model's performance on the target domain. This inspired various algorithms to explore statistical divergence minimization strategies for domain adaptation. In particular, as presented in the seminal works of [4], optimal transport can be used to do so.

Optimal transport is a mathematical theory that was originally devised in the context of transportation of masses under least effort [26]. Since there is a natural association between masses and probability distributions, optimal transport is suited for devising transformations that match different distributions. This theory has two contributions to domain adaptation: the definition of a transport map T that matches P_s and P_t, and the definition of a notion of distance between distributions, the Wasserstein distance.

Among the contributions of optimal transport to domain adaptation, the Wasserstein distance is of particular interest in this work. Indeed, it metrizes the space of probability distributions, hence geometric concepts such as the idea of barycenters can be extended to this space [1]. This latter notion is particularly useful for the Multi-Source Domain Adaptation (MSDA) setting.

The multi-source case in domain adaptation corresponds to when one has access to data coming from various domains $\{\mathcal{D}_{s_k}\}_{k=1}^{N}$. This case is challenging because one needs to minimize the distance from each P_{s_k} to P_t jointly. In the following, we provide our contributions, the intuition for our algorithm, and the paper structure.

Contributions. Our contributions are twofold: (1) we pro-

pose a new method for MSDA, the Wassertein Barycenter Transport (WBT). (2) A comparison with the state-of-the-art, revisiting datasets used for single-source domain adaptation which can be adapted to the MSDA scenario. In particular, the algorithms are evaluated on acoustic and visual adaptation datasets.

Intuition. We propose an algorithm for solving MSDA based on the Wasserstein barycenter. The intuition is to aggregate all source domains $\{\mathcal{D}_{s_k}\}_{k=1}^{N}$ into a single domain, $\mathcal{D}_b$ through the Wasserstein barycenter. Once the aggregation step is done, standard domain adaptation may be employed.

Paper structure. The rest of this paper is organized as follows: section 2 presents the related work in the fields of domain adaptation, optimal transport, acoustic and visual recognition. Section 3 details the WBT algorithm. Section 4 details the numerical experiments and its results, as well as it discusses the findings. Finally, section 5 concludes the paper.

2. Related Work

In the section we cover the state-of-the-art in domain adaptation and Optimal Transport for Domain Adaptation (OTDA). Moreover, a particular focus will be given to domain adaptation in the multi-source context.

Domain adaptation is a sub-topic of transfer learning, as first defined by [17]. The first work to consider a mismatch in the data distributions was [13], who proposed a method for the detection and estimation of change. Since then, this approach was formalized in [3], [7] and [2]. This latter work provided a solid theoretical ground for both single and multi-source domain adaptation.

The application of optimal transport to the latter subject is very recent, and it was first introduced in the seminal works of [4], which proposed solving the domain adaptation problem through the minimization of the Wasserstein distance between source and target distributions. This approach was then formalized in [20], which also commented on the possibility of using the Wasserstein barycenter for solving the MSDA case.

To the best of our knowledge, only two optimal transport based approaches for solving the MSDA scenario exists. These are the Joint Class Proportion and Optimal Transport (JCPOT) [19] and Weighted Joint Distribution Optimal Transport (WJDOT) [24]. The first algorithm was devised to solve a MSDA problem when there is a mismatch between the target distributions, that is, $P_s(Y) \neq P_t(Y)$. The second one proposes to solve the standard MSDA problem by using the joint distribution $P(X, Y)$, and the estimation of a weighting vector α. In this latter case, as the target labels are not available, the authors have proposed to substitute $P_t(X, Y)$ for $P^f(X, Y) = P_t(X, f(X))$, where f is the predictive function.

Our approach bears some similarity to both of these approaches. First, JCPOT estimates the class proportions through the Wasserstein barycenter. Second, WJDOT may be viewed as a barycenter between the distributions P_{s_k} and P^f, using α as the barycenter weights. Our approach, however, is different in the sense that the Wasserstein barycenter is used to build an intermediate domain for the transportation from sources to target.

Concerning the theory of barycenter in Wasserstein spaces, it was first formalized by [1], which presented the Wasserstein barycenter as an optimization problem. The treatment made by the authors was rather theoretical. Using their work as a foundation [6] proposed a fast algorithm to compute Wasserstein barycenters, using the Sinkhorn algorithm [5]. Despite the availability of new approaches, such as [14], we use the approach presented by [6] due to its simplicity.

3. Proposed Approach

In this section we discuss the WBT algorithm and its applications for MSDA. In this context, we present the theoretical background of optimal transport in section 3.1, the class-based regularization approaches to OTDA in section 3.2, the concept of Wasserstein barycenter in 3.3 and the WBT algorithm in section 3.4. Finally, in Section 3.5 we provide a theoretical discussion of our algorithm, based on the theoretical results of [20].

In the following discussion, we will adopt a discrete approach for optimal transport, as it is better suited for most of machine learning applications. In this setting, one does not know the source and target distributions μ_s and μ_t, but has access to samples $\mathbf{X}_s = \{\mathbf{x}_i^s\}_{i=1}^{n_s}$ and $\mathbf{X}_t = \{\mathbf{x}_j^t\}_{j=1}^{t}$. These samples induce empirical distributions $\hat{\mu}_s$ and $\hat{\mu}_t$. Each of these distributions is uniquely defined by its support $\mathbf{X}_s$ (resp. $\mathbf{X}_t$) and sample weights ω_i^s (resp. ω_j^t). We will further assume that ω^s (resp. ω^t) is uniform, that is, $\omega_i^s = n_s^{-1}$ (resp. n_t^{-1}), as in [4]. Therefore, $\hat{\mu}_s$ (resp. $\hat{\mu}_t$) may be expressed as,

$$\hat{\mu}_s(\mathbf{x}) = \frac{1}{n_s} \sum_{i=1}^{n_s} \delta(\mathbf{x} - \mathbf{x}_i^s), , \tag{1}$$

where δ is the delta dirac function.

3.1. Background

In this section, we formalize the problem of transfer learning with focus on domain adaptation, and detail how optimal transport can be used to solve it.

Formally, a domain is a pair $\mathcal{D} = (\mathcal{X}, \mu)$ [17], where $\mathcal{X}$ is a feature space and μ is its probability distribution. In domain adaptation, therefore, one has $\mathcal{D}_s = (\mathcal{X}_s, \mu_s)$, $\mathcal{D}_t = (\mathcal{X}_t, \mu_t)$ with $\mu_s \neq \mu_t$. Moreover, it is assumed that the labels conditional distribution $P(Y|X)$, is preserved

across the domains, that is, $P_s(Y|X) = P_t(Y|X)$. This corresponds the *covariate shift* hypothesis. For simplicity, we may assume that $\mathcal{X}_s \subset \mathbb{R}^d$ and $\mathcal{X}_t \subset \mathbb{R}^d$ are Euclidean spaces.

The application of optimal transport in the context of domain adaptation was first proposed by [4], in which the authors supposed that $\mu_s \neq \mu_t$ due to an unknown transformation $T : \mathcal{X}_s \to \mathcal{X}_t$. In addition, T is forced to preserve mass. In terms of empirical distributions, this is equivalent to the condition,

$$\omega_j^t = \sum_{i:T(\mathbf{x}_s^i)=\mathbf{x}_j^t} \omega_i^s.$$

This latter condition may be expressed through the push-forward operator, $T_\#$, so that $T_\#\hat{\mu}_s = \hat{\mu}_t$ [4]. Moreover, given $\hat{\mu}_s$ and $\hat{\mu}_t$, T may be determined through the following minimization problem [18],

$$T^* = \underset{T_\#\mu_s=\mu_t}{\operatorname{argmin}} \sum_i c(\mathbf{x}_i^s, T(\mathbf{x}_i^s)), \tag{2}$$

where $c : \mathbb{R}^d \times \mathbb{R}^t \to \mathbb{R}$ is a cost functional. This problem is known as Monge formulation of optimal transport. A couple of technical difficulties arise with this definition: (1) Equation 2 may not have solutions for empirical distributions [18]. (2) Equation 2 is not convex.

The issue (1) is particularly problematic for machine learning, since μ_s and μ_t are not known *a priori*. In this case, one only has access to empirical distributions $\hat{\mu}_s$ and $\hat{\mu}_t$. Moreover, (2) is problematic from an optimization perspective. These difficulties can be overcame with the so-called Kantorovich relaxation [26]. This is done by reformulating Equation 2 in terms of a coupling γ,

$$\gamma^* = \underset{\gamma\in\Pi(\hat{\mu}_s,\hat{\mu}_t)}{\operatorname{argmin}} \sum_{i,j} c(\mathbf{x}_i^s, \mathbf{x}_j^t)\gamma_{ij} = \langle C, \gamma\rangle_F, \tag{3}$$

where $\Pi(\hat{\mu}_s, \hat{\mu}_t)$ is the set of all couplings between $\hat{\mu}_s$ and $\hat{\mu}_t$, $\langle\cdot,\cdot\rangle_F$ is the Frobenius inner product of matrices, and $C_{ij} = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ is the cost matrix.

The Kantorovich formulation has nicer properties than Monge's formulation. First, Equation 3 has at least one solution, namely, the trivial coupling $\gamma_{ij} = \mu_s(\mathbf{x}_i^s)\mu_t(\mathbf{x}_j^t)$. Second, it is a convex problem on γ. Indeed, Equation 3 is a linear program on the matrix elements γ_{ij}.

Nonetheless, the linear programming approach for solving optimal transport problems is costly. By noticing that Equation 3 is a linear program of $n_s \times n_t$ variables, it scales poorly as the number of samples on each domain grows. To solve this drawback, [5] proposed an alternative optimization problem,

$$\gamma^* = \underset{\gamma\in\Pi(\hat{\mu}_s,\hat{\mu}_t)}{\operatorname{argmin}} \langle C, \gamma\rangle_F - \epsilon H(\gamma), \tag{4}$$

where $H(\gamma)$ is the entropy of matrix γ. This is an approximation to the original optimization problem, but has the desirable property of linear convergence and relies on matrix-vector operations.

Moreover, the theory of optimal transportation presents a second contribution to machine learning. Let $C_{ij} = ||\mathbf{x}_i^s - \mathbf{x}_j^t||_p^p$, Equation 3 further defines the so-called Wasserstein distance,

$$W_p^p(\mu_s, \mu_t) = \underset{\gamma\in\Pi(\hat{\mu}_s,\hat{\mu}_t)}{\operatorname{minimize}} \sum_{i,j} C_{ij}\gamma_{ij}, \tag{5}$$

which is a distance over the space of probability distributions with finite moments of order p, $P_p(\mathcal{X})$. This space is called a Wasserstein space [26]. Common values considered for p are 1, and 2, thus using the ℓ_1 and ℓ_2 norms as the cost $c(\mathbf{x}^s, \mathbf{x}^t)$. These cases result in the Wasserstein distances W_1 and W_2, respectively.

In addition, analogously to Equations 3 and 5, Equation 4 is associated with a distance called Sinkhorn distance [5],

$$S(\hat{\mu}_s, \hat{\mu}_t) = \underset{\gamma\in\Pi(\hat{\mu}_s,\hat{\mu}_t)}{\operatorname{minimize}} \langle C, \gamma\rangle_F - \epsilon H(\gamma).$$

As [4] remarks, once γ^* has been estimated, the source domain samples may be transported to the target domain by following the geodesics defined either by the Wasserstein or the Sinkhorn distances. This corresponds to an optimization problem for each source sample,

$$\hat{\mathbf{x}}_i^s = \underset{\mathbf{x}\in\mathcal{X}_s}{\operatorname{argmin}} \sum_j \gamma_{ij} c(\mathbf{x}, \mathbf{x}_j^t),$$

This problem has a closed form for the ℓ^2-cost, called Barycentric mapping [4]. In this case, the transported source samples is given by $\hat{\mathbf{X}}_s = n_s\gamma\mathbf{X}_t$.

3.2. Class-based Regularization

In the context of the application of optimal transport for domain adaptation, it has been verified in practice that using class-based regularization additionally to entropic regularization yields better performance [4]. This has been also supported theoretically by [20]. As follows, we describe a class regularizer that is relevant to our work. It corresponds to the Laplacian regularizer, presented by [4].

The Laplacian regularizer is based on a similarity matrix $S_s(i,j) \in \mathbb{R}^{n_s\times n_s}$ between samples $\mathbf{x}_i^s$ and $\mathbf{x}_j^s$. In practice, we proceed as [4], and consider $S_s(i,j)$ as the adjacency matrix of a nearest neighbors graph, that is,

$$S_s(i,j) = \begin{cases} 1 & \text{if } \mathbf{x}_i^s \text{ is in the k nearest neighbors of } \mathbf{x}_j^s \\ 0 & \text{otherwise} \end{cases},$$

moreover, the condition $S_s(i,j) = 0$ for i,j with $y_i^s \neq y_j^s$ is enforced for class-sparsity. The penalty can be calculated

using the following formula,

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

This equation is equivalent to $\Omega_c(\gamma) = Tr(\mathbf{X}_t^T\gamma^T L_s\gamma\mathbf{X}_t)$ for $L_s = \text{diag}(S_s\mathbf{1}) - \mathbf{S}_s$, the Laplacian of the adjacency matrix S_s. Finally, repeating the procedure without enforcing class-sparsity for $\mathbf{S}_t$ yields the following penalty term,

$$\begin{aligned}\Omega_c(\gamma) = (1-\alpha)Tr(\mathbf{X}_t^T\gamma^T L_s\gamma\mathbf{X}_t) \\ + \alpha Tr(\mathbf{X}_s^T\gamma L_t\gamma^T\mathbf{X}_s).\end{aligned} \tag{6}$$

Notice that this penalty term penalizes samples that are close to each other in the source domain from being transported to distant points in the target domain, whenever these belong to the same class. The same reasoning can be applied to the inverse transport using $\mathbf{S}_t$.

With the theory presented in the previous two sections, we may state the general OTDA framework for single-source domain adaptation, which consists in the minimization problem posed by,

$$\gamma_\epsilon^* = \underset{\gamma\in\Pi(\hat{\mu}_s,\hat{\mu}_t)}{\text{argmin}} \langle C,\gamma\rangle_F - \epsilon H(\gamma) + \eta\Omega_c(\gamma), \tag{7}$$

we will moreover denote by S_c to its associated distance between distributions, in the same spirit of the Wasserstein and Sinkhorn distances.

3.3. Barycenters on Wasserstein Spaces

Since the p-Wasserstein distance metrizes the space $P_p(\mathcal{X})$, it allows for the extension of geometric concepts such as the notion of barycenter. In this context, the Wasserstein barycenter was first formally defined by [1] as,

Definition 1 *Given $\{\hat{\mu}_{s_k}\}_{k=1}^N$, with $\hat{\mu}_{s_k} \in P_p(\mathcal{X})$, $\forall k$, and given positive constants $\{\lambda_k\}_{k=1}^N$ such that $\sum_k \lambda_k = 1$, the barycenter of $\{\hat{\mu}_{s_k}\}_{k=1}^N$, with weights $\{\lambda_k\}_{k=1}^N$ is denoted by $\hat{\mu}_b$, and is the solution of,*

$$\hat{\mu}_b = \underset{\mu\in P_p(\mathcal{X})}{\text{argmin}} \sum_{k=1}^N \lambda_k W_p^p(\hat{\mu}_{s_k}, \hat{\mu}_b)..$$

The problem of calculating the Wasserstein barycenter of empirical measures $\{\hat{\mu}_{s_k}\}_{k=1}^N$ was first considered by [6], and was applied in two contexts: (1) Centroid of histograms for the visualization of perturbed images, and (2) Clustering with uniform centroids. Despite their relevance for data analysis, these applications are not related to supervised learning.

An important distinction has been made by [6] about the calculation of the Wasserstein barycenter on empirical measures. One has two different optimization problems, namely the fixed-support and free-support barycenters. The first corresponds to fixing the support $\mathbf{X}_b$ of $\hat{\mu}_b$, and solving Equation 8 for the sample weights ω_b. The second corresponds to fixing ω_b (which is considered uniform, or found by the aforementioned method) and optimizing Equation 8 with respect to the support $\mathbf{X}_b$.

3.4. Wasserstein Barycenter Transport

The goal of WBT algorithm is to aggregate all source domains in a intermediate domain, which will later be transported to the target domain. This aggregation procedure is done through the Wasserstein barycenter.

For the calculation of the barycenter, two assumptions are made: (1) $\hat{\mu}_b$ is supported on $n_b = \sum_{k=1}^N n_{s_k}$ points, and (2) the weights ω_b are fixed and uniform, equal to n_b^{-1}. The first assumption seeks to ensure that each source data point is represented on the barycenter domain, and the second assumption is the standard for applying optimal transport for domain adaptation.

Note that since one has a point in the barycenter domain for each point in the source domains, we may as well artificially label the support of $\mathbf{X}_b$, with the concatenation of vectors $\{y_{s_k}\}_{k=1}^N$, as follows,

$$y_j^b = y_i^{s_k}, \text{ for } j = k\times i. \tag{8}$$

This step is crucial for the success of domain adaptation, since the transportation sources $\rightarrow$ barycenter $\rightarrow$ target should preserve the class structure of the data. Moreover, since the weights are fixed, only the free-support barycenter optimization problem needs to be solved.

To preserve the class structure across different domains, a class-based penalty inspired on [4] is used. This penalty term is defined as follows,

$$\begin{aligned}\Gamma_{ij}(y_s, y_b) &= \begin{cases} L & \text{if } y_i^s \neq y_j^b \\ 0 & \text{if } y_i^s = y_j^b \end{cases}, \\ \Omega_{cl}(\gamma; y_s, y_b) &= \sum_{i,j}\Gamma_{ij}\gamma_{ij}.\end{aligned} \tag{9}$$

where y_b denotes the labels artificially assigned to barycenter's points, and $L \gg max_{i,j}C_{ij}$ is a hyperparameter. This penalty has been originally proposed for semi-supervised optimal transport, where Ω_{cl} was computed using source and target labels. We remark the context of WBT, no information about the target labels is needed since Equation 9 is computed with respect to the barycenter labels.

The penalty from Equation 9 is added to the barycenter calculation cost function, leading to Equation 10 below. This can be interpreted as considering points from different

classes as far apart in the feature space. Hence, barycenter calculation is done through the following minimization problem,

$$\begin{aligned} S_{cl}(\hat{\mu}_{s_k}, \hat{\mu}_b) = & \underset{\gamma \in \Pi(\hat{\mu}_s, \hat{\mu}_t)}{\text{minimize}} \ \langle C, \gamma \rangle_F - \epsilon H(\gamma) \\ & + \Omega_{cl}(\gamma; y_s, y_b), \\ \hat{\mu}_b = & \underset{\mu \in P_p(\mathcal{X})}{\text{argmin}} \sum_{k=1}^{N} \lambda_k S_{cl}(\hat{\mu}_{s_k}, \hat{\mu}_b). \end{aligned} \tag{10}$$

In this context, we propose a method for solving the MSDA problem by transporting the Wasserstein barycenter of sources to the target domain. This corresponds to the following optimization problem:

$$\hat{\mu}^* = \underset{\mu \in P_p(\mathcal{X})}{\text{minimize}} \sum_{k=1}^{N} \lambda_k S_{cl}(\hat{\mu}_{s_k}, \mu) + S_c(\hat{\mu}_t, \mu). \tag{11}$$

This minimization problem corresponds to two independent steps: (1) the calculation of the Wasserstein Barycenter, corresponding to minimizing the summation in the right hand side of Equation 11, and (2) the transportation of $\hat{\mu}_b$ into $\hat{\mu}_t$, corresponding to minimizing $S_c(\hat{\mu}_t, \hat{\mu}_b)$. These two steps are summarized in Algorithm 1, which is an adapted version of Algorithm 2 of [6].

Algorithm 1 Wassertein Barycenter Transport

Require: Source domains samples $\mathbf{D}_{s_k} = \{(\mathbf{x}_i^{s_k}, y_i^{s_k})\}_{i=1}^{n_{s_k}}$, target domain samples $\mathbf{D}_t = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$, initial barycenter support $\mathbf{X}_b^0$, barycenter labels $\mathbf{y}_b$ and weights $\{\lambda_k\}_{k=1}^N$
  while $\mathbf{X}_b$ has not converged **do**
    for $k = 1, \cdots N$ **do**
      $\gamma_k \leftarrow$ Solution of Eq. 10.
    end for
    $\mathbf{X}_b \leftarrow n_b \sum_{k=1}^N \lambda_k \gamma_k^T \mathbf{X}_{s_k}$
  end while
  $\gamma \leftarrow$ Solution of Eq. 4 between $\hat{\mu}_b$ as source and $\hat{\mu}_t$ as target.
  $\hat{\mathbf{X}}_s \leftarrow n_s \gamma \mathbf{X}_t$
Ensure: Transported source samples $\hat{\mathbf{X}}_s$.

3.5. Theoretical Guarantees

In this Section we provide the theoretical insight for justifying our approach. Following [2], this corresponds to proving that our procedure minimizes a target error bound. We begin by defining the notion of error of a hypothesis h, on a domain $\mathcal{D}$,

Definition 2 *(Due to [2]) The error of a hypothesis function $h \in \mathcal{H}$ with respect to a labeling function f on a domain $\mathcal{D}$ is given by,*

$$\epsilon_{\mathcal{D}}(h, f) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[|h(\mathbf{x}) - f(\mathbf{x})|]. \tag{12}$$

Given a domain equipped with a ground labeling function $(\mathcal{D}_s, f_s)$, we will adopt for now on the abbreviation $\epsilon_s(h) = \epsilon_{\mathcal{D}_s}(h, f_s)$. The notion of error can be extended naturally to the case of many source domains. Given a vector $\alpha \in \mathbb{R}^N$, such that $\sum_{j=1}^N \alpha_j = 1$, the $\alpha-$weighted error of h is given by,

$$\epsilon_\alpha(h) = \sum_{j=1}^{N} \alpha_j \epsilon_{s_j}(h).$$

Notice that Equation 12 can be approximated by substituting the expectation operator by an empirical mean. In this case, we will denote the empirical error functional as,

$$\hat{\epsilon}_\alpha(h) = \sum_{j=1}^{N} \frac{\alpha_j}{n_j} \sum_{i=1}^{n_{s_j}} |h(\mathbf{x}_i) - f_j(\mathbf{x}_i)|.$$

Moreover, for the purposes of the next theorem, we will suppose that $n_{s_j} = \beta_j n$, for $\sum_{j=1}^N \beta_j = 1$. In these conditions, we may re-state Theorem 4 of [20],

Theorem 1 *Let $\mathbf{X}_{s_j}$, $j \in \{1, \cdots, N\}$ and $\mathbf{X}_t$ be $N+1$ samples of size n_{s_j} and n_t drawn i.i.d. from μ_{s_j} and μ_t respectively. Let $\hat{\mu}_{s_j}$ and $\hat{\mu}_{s_t}$ be the respective empirical measures. If $\hat{h}_\alpha$ is the empirical minimizer of $\hat{\epsilon}_\alpha$ and $h_t^* = \underset{h \in \mathcal{H}}{\text{minimize}}\ \epsilon_t(h)$, then for any fixed α and $\delta \in (0, 1)$, with probability at least $1 - \delta$ (over the choice of samples),*

$$\begin{aligned} \epsilon_t(\hat{h}_\alpha) \leq & \ \epsilon_t(h_T^*) + c_1 \\ & + 2 \sum_{j=1}^{N} \alpha_j (W_1(\hat{\mu}_{s_j}, \hat{\mu}_t) + \lambda_j + c_2), \end{aligned} \tag{13}$$

where,

$$c_1 = 2\sqrt{\frac{2K \sum_{j=1}^N \frac{\alpha_j^2}{\beta_j} \log(2/\delta)}{n}} + 2\sqrt{\sum_{j=1}^{N} \frac{K \alpha_j}{\beta_j}},$$

$$c_2 = \sqrt{2\log(1/\delta)/\xi'} \left(\sqrt{\frac{1}{n_{s_j}}} + \sqrt{\frac{1}{n_t}} \right),$$

$$\lambda_j = \underset{h \in \mathcal{H}}{\textit{minimize}}\ \epsilon_{s_j}(h) + \epsilon_t(h).$$

The proof is outlined in [20], and follows the same principle of Theorem 4 of [2]. For completeness, it is presented on the supplementary material. In the discussion of the last theorem, [20] proves that minimizing the Wasserstein distance on the right-hand-side of Equation 13 is equivalent to the following optimization problem,

$$\mu = \underset{\hat{\mu} \in \mathcal{P}_p(\mathcal{X})}{\text{minimize}} \frac{1}{N} \sum_{j=1}^{N} \alpha_j W_1(\hat{\mu}_{s_j}, \hat{\mu}) + W_1(\hat{\mu}, \hat{\mu}_t),$$

which is the unregularized version of Equation 11. Moreover, it is important to mention that, as argued by [20], the minimization of the Wasserstein distance is not sufficient for an adaptation that improves target error. Indeed, λ_j plays an important role in bound 13, and class-based regularization is important for controlling it.

In a nutshell, if points with different classes $\mathbf{x}_i^b$ and $\mathbf{x}_j^b$ on the barycenter are transported to the same target $\mathbf{x}_k^t$, there is an increase in the joint error. The same reasoning can be applied for the transport of sources to the barycenter. Hence, class-based regularization yields better results in the context of domain adaptation, as it has been verified in practice [4].

4. Experiments and Discussion

4.1. Datasets

To establish a comparison between the selected algorithms, four domain adaptation tasks were chosen: Music Genre Recognition (MGR), Music-Speech Discrimination (MSD), object and face recognition. These datasets were chosen based on their relevance in the transfer learning literature. For instance, MSD was previously considered in the work of Turrisi, et al. [24], face recognition was explored in [4], and object recognition is one of the most used benchmarks in domain adaptation. Below each dataset is described, and a summary of each task is shown in Table 1.

Task	Domains	# Samples	# Features	# Classes
Music Genre Recognition and Music-Speech Discrimination	Clean	1000	56	10/2
	F16	1000	56	10/2
	Buccaneer2	1000	56	10/2
	Destroyerengine	1000	56	10/2
	Factory2	1000	56	10/2
Object Recognition	Caltech	1123	4096	10
	Webcam	295	4096	10
	Amazon	958	4096	10
	DSLR	157	4096	10
Facial Recognition	PIE05	3332	1024	68
	PIE07	1629	1024	68
	PIE09	1632	1024	68
	PIE29	1632	1024	68

Table 1. Summary for each task considered in the MSDA scenario.

MGR and MSD: The original MGR dataset [25] is a classification dataset, consisting music samples of 10 distinct music genres (blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock), having 100 samples each. The MSD dataset [10], on the other hand, is composed by 64 music samples and 64 speech samples. No distinction upon the music genre is made. In both datasets, each sample is a *.wav* file with 30 seconds of duration. To simulate a domain adaptation scenario, each audio sample is overlaid with a specific noise, chosen from a noise dataset[1] using the Pydub library [21]. A total of 56 features were extracted on both datasets using the Librosa library [9], including the mean and variance of chromagram, root mean squared error, spectral bandwidth and rolloff, zero crossing rate, harmonic and percussive components, tempo and 20 Mel-Frequency Cepstral Coefficient (MFCC).

Object Recognition: For the object recognition task we use the Office-Caltech dataset, which is constituted by the Office dataset [22] and the Caltech-256 dataset [12]. The first dataset has three domains: Amazon, dslr and webcam. Each of these domains present different resolution and acquisition conditions. The second dataset is composed by images of various categories downloaded from Google and Picsearch. In the context of Office-Caltech dataset, these two datasets are merged by considering only the classes they have in common.

The resulting dataset has 2533 samples, for which DeCAF [8] features were extracted. These features correspond to the 7th layer activation of a convolutional neural network trained on imageNet, and then fine-tuned for object recognition [4].

Face Recognition: For this task, the CMU Pose, Illumination, Expression (PIE) dataset was used [23], which consists of over 40,000 images of size 32×32 from 68 individuals. The face recognition task corresponds to identifying the individual on each one of the images. From the total number of images, four different cameras were selected for the adaptation task: c05 (left pose), c07 (upward pose), c09 (downward pose) and c29 (right pose). Each camera is considered as an individual domain, and is denoted as PIEX, where X corresponds to the camera number.

Single vs Multi-source Domain Adaptation: The visual adaptation datasets were previously considered in [4], in the context of single-source domain adaptation. Both of these datasets have four domains, resulting in 12 single-source domain adaptation experiments. Each experiment corresponds to each possible pair source vs. target between distinct domains. When considered in the MSDA context, four experiments are possible, choosing one domain as target, and leaving all others as sources.

4.2. Acoustic Adaptation

For the acoustic adaptation experiments, 4 algorithms are compared with WBT: (1) Kernel Mean Matching (KMM) [11], which consists on a kernel-based importance estimation technique, (2) Transfer Component Analysis (TCA) [16], which is an algorithm that learns a subspace through the minimization of the Maximum Mean Discrepancy (MMD) metric, (3) Optimal Transport [4] through the Sinkhorn algorithm, with and without the additional Laplacian regularization (Equation 6). These are denoted, respectively, as OT-Laplace and OT-IT. (4) JCPOT [19], with its variant JCPOT-LP, which corresponds to the use of label propagation [19].

Regarding the results in Table 2, we remark that two methods have very low performance with comparison to the others: KMM, and the non-regularized version of WBT. In

[1] http://spib.linse.ufsc.br/noise.html

Task	MGR					MSD				
Method	Buccaneer2	Destroyerengine	F16	Factory2	Average	Buccaneer2	Destroyerengine	F16	Factory2	Average
Baseline	22.90 ± 0.84	38.25 ± 0.91	51.57 ± 1.11	47.80 ± 0.34	40.13 ± 11.07	82.43 ± 1.75	51.57 ± 2.56	88.89 ± 2.72	50.02 ± 2.21	68.23 ± 17.59
KMM	21.75 ± 0.99	39.25 ± 0.66	49.81 ± 1.69	47.37 ± 0.71	39.54 ± 10.99	87.12 ± 2.79	52.35 ± 2.94	74.86 ± 5.58	50.41 ± 2.17	66.18 ± 15.44
TCA	58.95 ± 1.27	60.67 ± 2.07	68.75 ± 2.11	59.82 ± 0.50	62.04 ± 3.91	90.43 ± 1.40	87.14 ± 4.99	**95.12 ± 2.02**	84.76 ± 3.30	89.36 ± 3.88
OT-IT	56.35 ± 0.84	61.92 ± 1.64	66.72 ± 1.86	61.77 ± 1.65	61.69 ± 3.67	89.26 ± 1.56	82.84 ± 2.78	84.97 ± 3.09	91.21 ± 2.04	87.07 ± 3.32
OT-Laplace	58.02 ± 1.45	60.47 ± 1.75	66.55 ± 1.60	63.87± 1.51	62.23 ± 3.24	87.28 ± 2.97	84.38 ± 1.76	86.14 ± 2.79	90.61 ± 1.68	87.10 ± 2.27
JCPOT	35.87 ± 0.41	48.47 ± 2.97	51.92 ± 3.25	51.95 ± 1.75	47.05 ± 6.60	92.55 ± 2.11	87.89 ± 1.39	88.67 ± 1.67	82.41 ± 2.22	87.88 ± 3.61
JCPOT-LP	36.40 ± 0.39	52.92 ± 1.32	56.30 ± 0.37	51.52 ± 2.28	49.28 ± 7.62	89.06 ± 1.38	84.97 ± 3.23	90.24 ± 1.71	86.13 ± 1.88	87.13 ± 2.13
WBT	21.37 ± 2.25	24.30 ± 2.71	25.30 ± 6.02	22.70 ± 2.25	23.41 ± 1.50	56.88 ± 9.54	56.63 ± 6.88	56.63 ± 6.56	59.38 ± 2.61	57.38 ± 1.15
WBT$_{reg}$	**70.60 ± 1.27**	**83.05 ± 0.97**	**84.40 ± 1.71**	**90.17 ± 0.46**	**82.05 ± 7.13**	**96.27 ± 1.60**	**92.98 ± 1.38**	94.92 ± 0.68	**96.87 ± 0.94**	**95.26 ± 1.49**
Target-only	67.43 ± 1.43	67.96 ± 2.91	66.86 ± 2.00	68.37 ± 1.87	67.41 ± 0.56	90.51 ± 3.98	93.07 ± 3.81	89.23 ± 4.25	92.30 ± 3.62	91.27 ± 1.50

Table 2. Results for MGR and MSD tasks. The results are grouped by task, and each column represents an individual experiment taking the column name as target domain. The values shown in the table are the obtained accuracies, in percentage. These values were averaged using a 5-fold cross validation procedure, and the interval of $\pm\sigma$ is shown.

Task	Object Recognition					Face Recognition				
Method	Amazon	dslr	webcam	Caltech	Average	PIE05	PIE07	PIE09	PIE29	Average
Baseline	90.55 ± 1.36	96.83 ± 1.33	88.36 ± 1.33	82.95 ± 1.26	89.67 ± 4.97	26.57 ± 2.51	42.99 ± 2.02	55.45 ± 2.75	31.73 ± 2.07	39.18 ± 11.11
PCA	91.67 ± 1.39	98.09 ± 3.81	92.61 ± 4.81	83.58 ± 1.45	91.49 ± 5.17	26.75 ± 2.55	43.15 ± 2.23	55.45 ± 2.81	31.73 ± 2.42	39.27 ± 11.07
TCA	86.83 ± 4.71	89.32 ± 1.33	**97.51 ± 1.18**	80.79 ± 2.65	88.61 ± 6.00	19.67 ± 2.54	12.74 ± 1.90	12.23 ± 2.98	11.91 ± 3.10	14.13 ± 3.21
OT-IT	69.31 ± 2.77	74.26 ± 0.66	74.69 ± 2.23	73.08 ± 0.87	72.83 ± 2.12	49.50 ± 3.75	59.41 ± 4.26	57.48 ± 3.92	52.78 ± 4.53	54.79 ± 3.89
OT-Laplace	70.50 ± 2.22	75.59 ± 1.68	75.39 ± 3.69	74.89 ± 0.48	74.09 ± 2.09	**51.13 ± 3.74**	63.24 ± 4.37	60.41 ± 4.38	56.42 ± 4.73	61.57 ± 5.77
JCPOT	79.23 ± 3.09	81.77 ± 2.81	93.93 ± 0.60	77.91 ± 0.45	83.21 ± 6.34	43.11 ± 4.56	70.44 ± 4.18	73.21 ± 5.01	58.98 ± 4.96	61.43 ± 11.85
JCPOT-LP	83.45 ± 0.15	81.51 ± 1.65	91.35 ± 1.91	79.65 ± 0.54	83.99 ± 4.45	42.43 ± 4.91	73.90 ± 2.64	77.27 ± 3.68	60.35 ± 3.56	63.49 ± 13.71
WJDOT	**94.24 ± 0.90**	**100.00 ± 0.00**	89.33 ± 2.91	**85.93 ± 2.07**	92.37 ± 5.30	Not Available	Not Available	Not Available	Not Available	Not Available
WBT	59.86 ± 2.48	60.99 ± 2.15	64.13 ± 2.38	62.80 ± 1.61	61.99 ± 1.64	7.90 ± 0.92	12.50 ± 1.88	11.02 ± 2.27	12.49 ± 1.45	10.98 ± 1.87
WBT$_{reg}$	92.74 ± 0.45	95.87 ± 1.43	96.57 ± 1.76	85.01 ± 0.84	**92.55 ± 4.58**	51.10 ± 4.44	**80.66 ± 3.93**	**79.58 ± 4.04**	**66.74 ± 5.73**	**69.52 ± 13.81**
Target-only	94.98 ± 1.29	94.74 ± 3.71	96.78 ± 2.36	91.43 ± 2.06	94.48 ± 1.93	97.67 ± 1.86	95.66 ± 4.65	98.02 ± 2.33	98.11 ± 1.64	97.36 ± 0.99

Table 3. Results for object and face recognition. The organization and notation of this table is similar to that of Table 2.

the first case, a major assumption made for the importance estimation procedure is that the support of $P_t(X)$ is contained in the support of $P_s(X)$. This is not necessarily true, since the type of noise can create a very different signal with respect to the extracted features.

Moreover, notice that WBT$_{reg}$ is the best performing algorithm among the tested methods, improving the baseline by 41.91% on average for MGR, and by 27.03% for MSD. For MGR, when compared to the second best method (OT-Laplace), it presents an average improvement of 19.82%. Moreover, it has even improved the target-only case, where one assumes that a classifier is trained and evaluated only on labeled target data. The average improvement in accuracy is 14.64%. The same consideration is valid for MSD, but in this case the second best was TCA, with a performance gap of 5.9%. When compared to the target-only case, WBT$_{reg}$ has a performance gain of 3.99%. This evidence the fact that the source domains carry information that may improve classification.

4.3. Visual Adaptation

For the visual adaptation experiments, 6 algorithms are compared with WBT: (1) Principal Component Analysis (PCA), which consists on projecting the whole dataset (sources and target) on a few principal components [4], (2) TCA, (3) OT-IT and OT-Laplace, (4) JCPOT and JCPOT-LP, and (5) WJDOT [24]. For WJDOT, since the code was not publicly available, its results are only shown for the object recognition task, for which the results reported in [24] are shown in Table 3.

Object Recognition: the results for the Caltec-Office dataset are shown on the left side of Table 2. For this task, WJDOT is the method with higher accuraccy across the different strategies tested. Our approach is the best on average, with a tight performance advantage of 0.18% with respect to WJDOT. Especially, when comparing these two methods, for this specific task WJDOT uses information about only one source domain, as reported in [24] and shown in their supplementary material. In contrast, WBT and WBT$_{reg}$ use information about all source domains equally (uniform weights). As it turns out, our approach manages to acquire a higher performance on webcam and caltech domains, indicating that the information carried by other domains is still useful in the context of the barycenter calculation.

When considered against other optimal transport methods, WBT$_{reg}$ shows a considerable higher performance. For instance, by merging all source domains into a single domain, and using the standard OTDA framework yields at best an accuracy of 74.09% (OT-Laplace), 17.71% lower than WBT$_{reg}$'s performance. This highlights the importance of distinguishing across domains when performing the transport. Moreover, when compared to JCPOT, its performance is 7.81% lower than ours. Additionally, it is noteworthy that even though JCPOT is designed for MSDA, it is used here outside its hypothesis (target shift, as discussed in [19]), justifying its lower performance.

Face Recognition: the results concerning the PIE dataset

Figure 1. Facial adaptation results example (best seen on screen). Each row corresponds to an individual domain. Especially, the first row shows samples from the three source domains (upward, left and downward poses), while the last row shows the target domain (right pose). In between, the domains generated by various various algorithms is shown.

are shown on the right side of Table 3. For this task, WBT$_{reg}$ is the best performing method, with an improvement of 6.3% on the average across targets with respect to the second best, JCPOT.

Since the features used for the transport in this example are raw images, it allows a concrete visualization of the Wasserstein barycenter. Figure 1 shows a comparative between the various domains involved in the adaptation of the PIE dataset. In this example, the camera 29, corresponding to the right pose, was used as target domain. It is visible that the intermediate domain constructed by the barycenter (second row) consists mostly on frontal poses, while the transported samples (third row) display right poses, as expected.

Additionally, PCA and TCA show poor performances when compared to the other methods. As can be seen in Figure 1, the projections generated by these two methods is a constant image. This is a clear contrast with other adaptation tasks which used extracted features rather than raw signals, indicating that in the latter case, there may not exist a sub-space that the images share common characteristics.

4.4. General Remarks

Two remarks can be made about Table 2 and 3 simultaneously. First, the performance of WBT, without class-based regularization, is considerably lower than that of the other methods. The reason for this drop in performance was briefly discussed in Section 3.5.

When there is no class-based regularizer in Equation 10, the WBT algorithm manages to minimize only the Sinkhorn distance between sources and the barycenter. However, λ_j remains unbounded as the support points are free to move. In the worst case, it may happens that points from different classes are moved to the same point in the barycenter, corresponding to a situation where λ_j grows. Our experiments agrees with the previous literature [4] [20], in the sense that using class-based regularization greatly improves our results.

A second remark may be made according the field of application. Specially, WBT$_{reg}$ manages to improve the target-only case for acoustic adaptation, while for for visual adaptation this is not always the case. We remark that between these two tasks, there is a significant gap in complexity on the optimization problem posed by Equation 11, since there are 56 features for acoustic adaptation, 4096 for object recognition, and 1024 for facial recognition.

5. Conclusion

We propose a novel approach for MSDA using barycenters of Wasserstein spaces, the WBT algorithm. Our method fits into the OTDA framework, in the sense that it does domain adaptation through the transport of samples from sources to target. To do so, we build an intermediate domain through the Wasserstein barycenter, then transport the barycenter into the target domain.

The usage of the intermediate domain built by the Wasserstein barycenter has shown state-of-the-art performance in both acoustic and visual adaptation tasks. When the number of features is small, our method manages to largely improve the most optimistic case, when a classifier is trained and evaluated on the target domain, indicating that the information carried by source domains is useful. When the number of features on the task is high, our method still ranks among the best, but its results are less outstanding.

Moreover, our results support the claim that class-based regularization is important for a successful application of optimal transport for domain adaptation. Especially, whenever this kind of regularizer was omitted during the barycenter calculation, the performance dropped significantly.

Finally, we remark that our approach can be further generalized, supporting any kind of single-source domain adaptation algorithm for the transport barycenter $\rightarrow$ target.

References

[1] Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[3] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[4] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

[5] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

[6] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 685–693, Bejing, China, 22–24 Jun 2014. PMLR.

[7] Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 129–136, 2010.

[8] 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 *International conference on machine learning*, pages 647–655, 2014.

[9] Brian McFee et al. librosa/librosa: 0.8.0, July 2020.

[10] Tzanetakis George, Essl Georg, and Cook Perry. Automatic musical genre classification of audio signals. In *Proceedings of the 2nd international symposium on music information retrieval, Indiana*, 2001.

[11] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

[12] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

[13] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *VLDB*, volume 4, pages 180–191. Toronto, Canada, 2004.

[14] Giulia Luise, Saverio Salzo, Massimiliano Pontil, and Carlo Ciliberto. Sinkhorn barycenters with free support via frank-wolfe algorithm. In *Advances in Neural Information Processing Systems*, pages 9322–9333, 2019.

[15] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. A multi-device dataset for urban acoustic scene classification. In *Scenes and Events 2018 Workshop (DCASE2018)*, page 9, 2018.

[16] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010.

[17] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[18] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[19] Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia. Optimal transport for multi-source domain adaptation under target shift. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 849–858. PMLR, 16–18 Apr 2019.

[20] Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical analysis of domain adaptation with optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 737–753. Springer, 2017.

[21] James Robert, Marc Webbie, et al. Pydub. *Github*, 2011.

[22] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[23] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination and expression database of human faces. *Carnegie Mellon University Technical Report CMU-RI-TR-OI-02*, 2001.

[24] Rosanna Turrisi, Rémi Flamary, Alain Rakotomamonjy, and Massimiliano Pontil. Multi-source domain adaptation via weighted joint distributions optimal transport. *arXiv preprint arXiv:2006.12938*, 2020.

[25] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

[26] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[27] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.