Multi-Source Domain Adaptation through Wasserstein Barycenters

Eduardo Fernandes Montesuma

université PARIS-SACLAY

Multi-Source Domain Adaptation through Wasserstein Barycenters

Adaptation de Domaines Multi-Sources à l'aide des Barycentres de Wasserstein

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Informatique Mathématique
Graduate School : Informatique et sciences du numérique
Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Institut LIST (Université Paris-Saclay, CEA)**, sous la direction d'**Antoine SOULOUMIAC**, Directeur de Recherche au CEA-List, le co-encadrement de **Fred Maurice NGOLÈ MBOULA**, Ingénieur Chercheur au CEA-List

Thèse soutenue à Paris-Saclay, le 16 septembre 2024, par

Eduardo FERNANDES MONTESUMA

Composition du jury

Membres du jury avec voix délibérative

Gabriel PEYRE Directeur de recherche, HDR, ENS Ulm	Président
Ievgen REDKO Chercheur Scientifique Principal, HDR, Huawei	Rapporteur & Examinateur
Laëtitia CHAPEL Professeur, HDR, Institut Agro Rennes-Angers	Rapporteur & Examinateur
Jérome BOBIN Directeur de Recherche, HDR, Université Paris-Saclay	Examinateur
Rémi FLAMARY Professeur, HDR, École Polytechnique	Examinateur
Julie DELON Professeur, HDR, Université Paris Cité	Examinatrice

THESE DE DOCTORAT

NNT : 2024UPASG045

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Adaptation de Domaines Multi-Sources à l'aide des Barycentres de Wasserstein
Mots clés: Transport Optimal, Barycentres de Wasserstein, Adaptation de Domaines, Apprentissage par Transfert, Diagnostique de Défauts Inter-Domaines

Résumé: Les systèmes d'apprentissage automatique fonctionnent sous l'hypothèse que les conditions d'entraînement et de test ne changent pas. Néanmoins, cette hypothèse est rarement vérifiée en pratique. En conséquence, le système est entraîné avec des données qui ne sont plus représentatives des données sur lesquelles il sera testé: la mésure de probabilité des données évolue entre les périodes d'entraînement et de test. Ce scénario est connu dans la littérature sous le nom de *décalage de distribution* entre deux domaines : une source et une cible. Une généralisation évidente de ce problème considère que les données d'entraînement présentent elles-mêmes plusieurs décalages instrinsèques. On parle, donc, *d'adaptation de domaine à sources multiples* (MSDA). Dans ce contexte, le transport optimal est un outil de mathématique utile. En particulier, qui sert pour comparer et manipuler des mesures de probabilité. Cette thèse étudie les contributions du transport optimal à l'adaptation de domaines à sources multiples. Nous le faisons à travers des barycentres de Wasserstein, un objet qui définit une moyenne pondérée, dans l'espace des mesures de probabilité, des multiples domaines en MSDA. Basé sur ce concept, nous proposons : (i) une nouvelle notion de barycentre lorsque les mesures en question sont étiquetées, (ii) un nouveau problème d'apprentissage de dictionnaire sur des mesures de probabilité empiriques et (iii) de nouveaux outils pour l'adaptation de domaines via le transport optimal de modèles de mélanges Gaussiens. Nos méthodes améliorent les performances de l'adaptation de domaines par rapport aux méthodes existentes utilisant le transport optimal sur des benchmarks d'images et de diagnostic de défauts inter-domaines. Notre travail ouvre une perspective de recherche intéressante sur *l'apprentissage de l'enveloppe barycentrique* de mesures de probabilité.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Multi-Source Domain Adaptation through Wasserstein Barycenters
Keywords: Optimal Transport, Wasserstein Barycenters, Domain Adaptation, Transfer Learning, Cross-Domain Fault Diagnosis

Abstract: Machine learning systems work under the assumption that training and test conditions are uniform, i.e., they do not change. However, this hypothesis is seldom met in practice. Hence, systems are often trained with data that is no longer representative of the data it will be tested on. This case is represented by a shift in the probability measure generating the data. This scenario is known in the literature as *distributional shift* between two domains: a source, and a target. A straightforward generalization of this problem is when training data itself exhibit shifts on its own. In this case, one consider *Multiple Source Domain Adaptation* (MSDA). In this context, optimal transport is an useful field of mathematics. Especially, optimal transport serves as a toolbox, for comparing and manipulating probability measures. This thesis studies the contributions of optimal transport to MSDA. We do so through Wasserstein barycenters, an object that defines a weighted average, in the space of probability measures, for the multiple domains in MSDA. Based on this concept, we propose: (i) a novel notion of barycenter, when the measures at hand are equipped with labels, (ii) a novel dictionary learning problem over empirical probability measures and (iii) new tools for domain adaptation through the optimal transport of Gaussian mixture models. Through our methods, we are able to improve domain adaptation performance in comparison with previous optimal transport-based methods on image, and cross-domain fault diagnosis benchmarks. Our work opens an interesting research direction, on *learning the barycentric hull* of probability measures.

Synthèse en Français

Bien que cela ne soit pas explicite dans le titre, cette thèse traite de l'intelligence artificielle et de l'apprentissage automatique. Par conséquent, avant de présenter nos contributions, il convient de définir plus précisément ce que nous cherchons à accomplir. Ici, nous suivons la discussion sur l'histoire de l'intelligence artificielle de [1].

On peut dire que l'un des travaux pionniers sur l'intelligence artificielle est celui de [2], qui a défini, en termes mathématiques, un modèle artificiel du fonctionnement des neurones. Curieusement, ce travail est à la fois l'un des pionniers du concept de *réseau de neurones*, une architecture majeure pour les systèmes d'intelligence artificielle. Comme le mentionne [1], environ une décennie après ce travail révolutionnaire, un atelier d'été au Dartmouth College a mené la première étude sur l'intelligence artificielle. Parmi leurs objectifs, il y avait : *faire en sorte que les machines utilisent le langage*, *résoudre des problèmes réservés aux humains à l'époque*, et *s'améliorer elles-mêmes*. En résumé, l'idée était de décrire l'apprentissage et d'autres caractéristiques de l'intelligence de manière si précise qu'une machine pourrait *les simuler*.

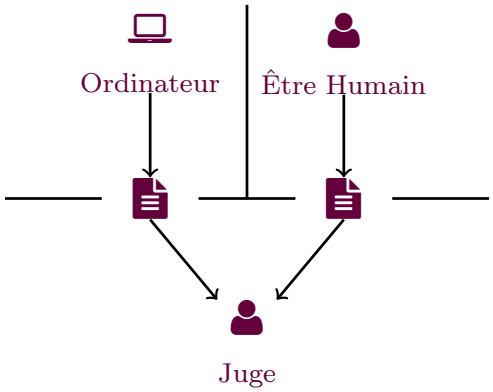


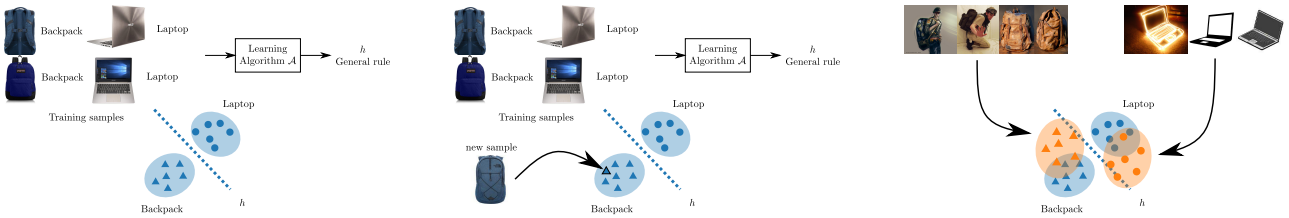
Figure 1 – **Illustration du test de Turing** dans lequel une machine et un humain répondent à des questions textuelles pour un juge humain. Un ordinateur est considéré comme intelligent s'il peut imiter un humain avec succès.

Voici une distinction importante. Dans ce travail initial, les machines n'étaient jamais censées *être intelligentes*, mais plutôt *imiter* l'intelligence. Cela fait écho au test de Turing, également connu sous le nom de *jeu de l'imitation* [3]. De manière générale, cette expérience de pensée fondamentale suppose un interrogateur humain, qui pose un ensemble de questions écrites. Un ordinateur réussit le test de Turing si l'interrogateur ne peut pas dire si les réponses écrites proviennent d'une personne ou d'un

ordinateur [1, Section 1.1.1].

Dans le domaine de l'intelligence artificielle, il existe plusieurs approches pour atteindre l'intelligence. Une manière intéressante de classer ces approches est entre les approches *rationalistes* et *empiristes*[1]. Du côté rationaliste, les praticiens réalisent généralement de l'*intelligence artificielle symbolique*. Cette approche se caractérise par la représentation des objets par des symboles de haut niveau, et la caractérisation de l'intelligence par la logique. Notez que cette perspective sur l'intelligence valorise des structures *a priori*[2] plutôt que l'expérience, car l'intelligence provient de la manière inhérente dont les objets sont représentés. Nous ne considérons pas ce type d'approches dans cette thèse.

En opposition à l'approche rationaliste, on trouve l'*apprentissage automatique*. Ce type de stratégie est dit *axé sur les données*, car plutôt que de s'appuyer sur des représentations expertes de haut niveau des objets du monde, il cherche des motifs dans les données disponibles. Ici, il est possible d'établir un parallèle entre les données et l'expérience humaine. Les *modèles* d'apprentissage automatique fonctionnent en définissant une fonction entre les entrées et les sorties souhaitées. En conséquence, il existe potentiellement plusieurs fonctions, qui sont classées selon une notion de *pertinence* par rapport aux données disponibles. Ainsi, un modèle est dit apprendre s'il maximise un certain critère de pertinence ou, de manière équivalente, s'il minimise une notion de risque. Ici, on établit un lien supplémentaire entre *apprentissage* et *optimisation*. Cette thèse adopte précisément cette approche de l'intelligence.



(a) Apprentissage. (b) Généralisation. (c) Décalage de distribution.

Figure 2 – **Le problème de l'induction et de l'adaptation.** En (a), un algorithme d'apprentissage apprend un motif sur un ensemble de données fini D. Ici, nous utilisons des exemples de reconnaissance d'objets, où l'on cherche à distinguer entre des ordinateurs portables et des sacs à dos. La règle générale, un classificateur h, est définie sur l'ensemble de l'espace latent. Ici, nous utilisons $\mathbb{R}^2$ pour simplifier l'exposition. En (b), nous illustrons l'idée de généralisation, c'est-à-dire la prédiction précise sur un nouvel exemple, non présent à l'origine dans l'ensemble d'apprentissage. En (c), nous montrons des images représentant les mêmes objets, mais avec des styles, poses et arrière-plans différents, ce qui induit un décalage dans l'apparition probable des données dans l'espace.

Ici, une question importante se pose. Étant donné que les modèles d'apprentissage automatique sont basés sur l'expérience, les connaissances qu'ils acquièrent sont *a posteriori*[3]. Un inconvénient

1. Cette distinction fait écho au débat philosophique de longue date entre ces deux champs
2. Nous employons ici le terme a priori dans son sens philosophique. Une proposition est a priori si elle est indépendante de l'expérience, c'est-à-dire si elle n'est pas contingente. Par exemple, l'énoncé : « un carré a quatre côtés » est universel. Il s'applique à tout carré possible.
3. Encore une fois, nous utilisons ce terme dans son sens philosophique. Les connaissances a posteriori sont

des connaissances *a posteriori* est le problème de l'induction [4]. En termes simples, ce problème peut être illustré comme suit : « étant donné que le soleil s'est levé tous les matins jusqu'à présent, se lèvera-t-il demain ? » Du point de vue de l'apprentissage automatique, cette question prend la forme de *généralisation*. Supposons qu'un modèle soit ajusté sur certaines données. Sera-t-il capable d'effectuer des prédictions fiables sur des données non vues ? Comme il se trouve, dans des conditions raisonnables, la réponse est oui, comme le pose la *théorie de l'apprentissage statistique* [5, 6]. Nous illustrons ces idées dans la Figure 2.

Cette thèse met la question de la généralisation à l'épreuve. À partir de l'apprentissage statistique, on peut s'attendre à ce que les modèles d'apprentissage automatique fonctionnent raisonnablement bien sur des échantillons similaires aux échantillons d'apprentissage. Dans la vie réelle, cependant, ces modèles sont souvent confrontés à des données différentes représentant les mêmes concepts. Prenons, par exemple, la tâche de l'analyse des sentiments, c'est-à-dire déterminer si un texte dénote un sentiment positif ou négatif. Supposons maintenant que vous disposiez de deux ensembles de données contenant des avis sur des produits. Le premier ensemble de données contient des avis sur des films. Le second contient des avis sur des produits ménagers. Ces deux ensembles de données essaient de réaliser la même tâche, mais présentent des particularités qui rendent la généralisation d'un ensemble à l'autre difficile (c'est-à-dire, un changement dans la distribution des mots). Il s'agit d'un exemple particulier d'un problème en apprentissage automatique connu sous le nom de *changement distributionnel* [7]. Un exemple est présenté dans la Figure 2.

En termes simples, l'objectif de cette thèse est de concevoir des méthodes capables de gérer le changement distributionnel. Nous le faisons par le biais de l'*apprentissage par transfert*. Là encore, nous nous appuyons sur l'intelligence humaine. Nous le faisons par le biais de l'*apprentissage par transfert*. Encore une fois, nous nous inspirons de l'intelligence humaine pour définir ce que devrait être l'apprentissage par transfert. En particulier, les humains peuvent s'adapter remarquablement bien à de nouvelles circonstances. Par exemple, il est plus facile pour un humain d'apprendre à parler français s'il maîtrise déjà une langue apparentée, comme le portugais, l'italien ou l'espagnol. Comme l'explique [8], le domaine de l'apprentissage par transfert "est motivé par le fait que l'être humain peut appliquer intelligemment des connaissances apprises auparavant pour résoudre de nouveaux problèmes plus rapidement ou avec de meilleures solutions".

Dans cette thèse, nous abordons un problème important de l'apprentissage par transfert, connu sous le nom d'*adaptation de domaine*. Dans ce problème, on a une tâche fixe (par exemple, l'analyse des sentiments), mais des domaines différents (par exemple, les films et les produits ménagers). De plus, on distingue le *domaine source* et le *domaine cible*. Le domaine source représente les connaissances acquises précédemment et solidement établies. Cela prend généralement la forme d'une quantité raisonnable de données étiquetées. Le domaine cible représente là où nous voulons que la généralisation ait lieu. L'adaptation de domaine impose généralement peu d'exigences sur le domaine cible. Par exemple, dans cette thèse, nous effectuons l'adaptation de manière non supervisée, c'est-à-dire qu'aucune donnée étiquetée n'est disponible dans le domaine cible. Comme nous l'avons déjà mentionné, nous adoptons une vue probabiliste ou distributionnelle de l'adaptation de domaine. Cela présente l'avantage de nous

contingentes au contexte particulier dans lequel les sensations sont acquises. Par exemple, l'énoncé : « la chaise est verte » est contingent à des exemples spécifiques de chaises. Il s'applique à l'expérience d'une chaise particulière, donc il est contingent.

permettre de dériver une caractérisation mathématique de la performance attendue lors de l'adaptation, en utilisant le cadre de l'apprentissage statistique.

Dans l'esprit de fournir un traitement probabiliste de l'adaptation de domaine, un cadre naturel pour ce faire est le transport optimal [9, 10]. Le transport optimal est un domaine des mathématiques qui s'intéresse au déplacement de masse avec le moindre effort. Depuis sa création [11], ce domaine a connu de nombreuses résurgences et reformulations. Du point de vue de cette thèse, ce cadre est utile car il nous fournit une boîte à outils rigoureuse pour manipuler les mesures de probabilité.

La pièce finale du puzzle que cette thèse tente de résoudre est la prise en compte de sources multiples. Dans les applications du monde réel, les utilisateurs sont souvent confrontés à de grands ensembles de données hétérogènes. Dans de tels cas, différentes approches peuvent être envisagées. D'une part, il est possible de sélectionner un sous-ensemble de données sources relativement homogène pour effectuer l'adaptation, au risque d'obtenir un modèle sous-optimal. D'autre part, on pourrait considérer tous les sous-ensembles hétérogènes au sein du jeu de données source comme une source unique. Ce choix néglige les décalages potentiels au sein du domaine source. Dans cette thèse, nous proposons des outils permettant d'utiliser les barycentres de Wasserstein dans l'adaptation de domaine à sources multiples.

D'un point de vue méthodologique, cette thèse peut être résumée en 3 contributions principales.

Barycentres de mesures étiquetées. Nos travaux [12, 13] et [14] ont été les premiers à considérer le problème du calcul d'un barycentre de Wasserstein de mesures avec des échantillons munis d'étiquettes. Cette question est complexe, car contrairement aux caractéristiques, les étiquettes ont une nature discrète. Dans cette thèse, nous représentons les étiquettes de manière continue via des probabilités floues.

Apprentissage de dictionnaires. Nos travaux [14] et [15] ont été les premiers à considérer un problème d'apprentissage de dictionnaires dans l'espace des mesures de probabilité au-delà des histogrammes.

Adaptation de domaine basée sur le modèle de mélange gaussien. Nos travaux [16] et [15] ont été les premiers à envisager l'utilisation du cadre de transport optimal basé sur le mélange gaussien de [17] pour l'adaptation de domaine.

De plus, cette thèse est structurée en trois parties, qui sont résumées ci-dessous.

Aperçu de la Partie I. La première partie de cette thèse vise à donner une vue d'ensemble sur les fondations théoriques de notre travail. Nous avons divisé ces fondations en trois chapitres. Le chapitre 2 couvre le transport optimal, le chapitre 3 traite des barycentres de mesures de probabilité et le chapitre 4 aborde l'adaptation de domaine. Différents aspects de ces chapitres sont plus ou moins abordés dans notre revue [18]. Nous détaillons ci-après les contenus de chaque chapitre.

Chapitre 2 : les multiples facettes du transport optimal. Ce chapitre introduit la théorie du transport optimal. Nos principales références ici sont [19], pour les concepts généraux du transport optimal, et [10], pour les méthodes computationnelles. Nous adoptons une approche pragmatique de la théorie du transport optimal, en introduisant un certain niveau d'abstraction tout en illustrant le transport optimal sur des espaces euclidiens, ce qui est nécessaire pour nos algorithmes. Nous nous concentrons principalement sur trois types de mesures : empiriques, gaussiennes et mélanges gaussiens.

Du point de vue computationnel, nous considérons des sujets récents, tels que le mini-batch, le transport optimal déséquilibré et partiel.

Chapitre 3 : barycentres de mesures de probabilité. Ce chapitre traite du barycentre, acteur principal de cette thèse. Pour cela, nous devons définir des distances dans l'espace des mesures de probabilité. Nous nous concentrons ici sur trois types de distances largement utilisées dans l'adaptation de domaine : la $\mathcal{H}-$distance, la discrepancy moyenne maximale, et la distance de Wasserstein. Comme nous traitons également des mesures empiriques, nous discutons de la manière d'estimer ces distances à partir d'échantillons finis. En outre, nous abordons les aspects computationnels du calcul des barycentres de Wasserstein, qui seront utiles dans les chapitres suivants.

Chapitre 4 : théorie de l'apprentissage et adaptation de domaine. Ce chapitre traite de deux problèmes. Le premier est la théorie de l'apprentissage, c'est-à-dire comment apprendre un classificateur à partir d'échantillons d'une mesure de probabilité. La question sous-jacente est celle de la généralisation, c'est-à-dire la capacité du classificateur à bien fonctionner sur des échantillons non vus *issus de la même mesure*. Notre référence principale ici est [5, 6], qui a établi les bases de la *théorie de l'apprentissage statistique*. Le deuxième problème est celui de l'adaptation de domaine, une généralisation du premier, qui consiste à apprendre dans des contextes où les données proviennent de mesures de probabilité différentes, c'est-à-dire sous un changement de distribution. Les principales références sont [8] et [20], qui ont décrit le problème en termes mathématiques, ainsi que le travail plus récent de [21, 22]. L'objectif de ce chapitre est d'établir les principales théories et algorithmes de l'adaptation de domaine.

Aperçu de la Partie II. Cette partie présente nos contributions méthodologiques à l'adaptation de domaine multi-source, c'est-à-dire lorsque le domaine source est composé de mesures hétérogènes. Nous la divisons en trois chapitres. Le chapitre 5 présente notre algorithme de Transport par Barycentre de Wasserstein, qui résout un problème multi-source en calculant d'abord un barycentre des mesures sources, puis en résolvant un problème à source unique entre le barycentre et la cible. Le chapitre 6 introduit notre nouveau cadre, *Dataset Dictionary Learning*, qui effectue un apprentissage de dictionnaire sur des mesures empiriques. Enfin, le chapitre 7 propose une reformulation paramétrique des algorithmes précédents. Ces chapitres s'appuient sur nos publications [12, 13, 14, 15, 16].

Chapitre 5 : Transport par Barycentre de Wasserstein. [12, 13] Dans ce chapitre, nous présentons une vue actualisée de nos articles [12, 13]. Nous introduisons notamment une nouvelle distance de type Wasserstein sur l'espace des distributions dans l'espace joint caractéristiques-étiquettes, appelée distance de joint-Wasserstein. Nous dérivons ensuite un nouvel algorithme, inspiré de l'algorithme à support libre de [26], pour calculer le barycentre de Wasserstein de mesures empiriques étiquetées.

Chapitre 6 : Dataset Dictionary Learning. [14] Ce chapitre présente un cadre d'apprentissage de dictionnaire basé sur le transport optimal, en étendant les travaux de [28] pour traiter des mesures empiriques à support libre. Nous proposons notre nouveau cadre d'apprentissage de dictionnaire, appelé *Dataset Dictionary Learning*.

Chapitre 7 : Adaptation de domaine avec mélanges gaussiens. [16, 15] Ce chapitre discute des outils pour calculer les barycentres de Wasserstein de mélanges gaussiens, et propose un nouvel algorithme rapide pour calculer ces barycentres. Nous introduisons également une distance de joint-

Wasserstein pour les mélanges gaussiens.

Aperçu de la Partie III. La dernière partie de cette thèse concerne l'expérimentation. Le chapitre 8 présente un nouveau benchmark en adaptation de domaine pour l'ingénierie chimique, basé sur le processus Tennessee Eastman [29]. Le chapitre 9 effectue un benchmarking de nos méthodes par rapport à d'autres algorithmes de pointe. Enfin, le chapitre 10 couvre nos expériences en distillation de jeux de données.

Chapitre 8 : Diagnostic de défaut inter-domaines [31]. Ce chapitre analyse le processus Tennessee Eastman et propose un benchmark pour l'adaptation de domaine dans le cadre de la détection de défauts.

Chapitre 9 : Benchmarking de l'adaptation de domaine multi-source. Ce chapitre compare nos méthodes basées sur le transport optimal à d'autres sur 5 benchmarks existants.

Chapitre 10 : Distillation de jeux de données [33]. Nous explorons ici la possibilité de compresser le domaine cible tout en maintenant une bonne performance en adaptation de domaine.

Acknowledgements

So many people have been fundamental to the realisation of this manuscript and these 3 years of thesis, that I am left with the feeling that those who carried me over the course of these years are deserving of the title of *doctor*, as much as me. Unfortunately, that is not how life works. I will try – not in vain, I hope - to do justice in these few lines.

I will begin by those closer to the evaluation of this thesis. I would like to recognize and thank the work done by the *rapporteurs*, especially their feedback, which helped me improve the quality of this manuscript. Furthermore, I would like to thank the members of the jury for the insightful discussion we had during the defence.

Of course, this thesis could not have been done without the wonderful direction of Antoine and Fred. It is very difficult to define what a perfect supervision is, but in my opinion, this comes very close to it. I am very grateful for the professional, and friendship links that we developed over the course of this thesis, and I cannot hope but feel a bit nostalgic when writing these verbose acknowledgments.

Now, moving a bit to the personal side, I would like to thank my family for the support, encouragement, and sacrifices made during these years. In this sense, I would like to express my deepest gratitude to my mother, my sister, and my wife, who deserves acknowledgments on its own. My wife supported me without hesitation over these 3 years, having to share me with my papers and this manuscript. You are, without any doubt, *my fixed point in this chaotic world*.

Furthermore, I would like to thank my friends, some of them became colleagues, others became co-authors. I am very happy to have you on my side. I would like to thank all my brazilian friends in France, Ana Vitória, André, Bia, Gabi, Geovana, Gustavo, Leo, Marcelo, Patrícia, Renan, Victor, Vitor. All my CEA friends, Arnaud, Fabiola, Baudouin, Antonin, Pierre, Stevan, Adel, and many others!

To conclude, I would like to thank the Commissariat à l'energie atomique, and more generally France, for providing me near optimal *material* and *immaterial* conditions for performing this work.

Contents

Chapter 1

Introduction

Il faut imaginer Sisyphe heureux

Albert Camus

Contents

Although not explicit from the title, this thesis treats Artificial Intelligence and machine learning, as a result, before laying out our contributions, a few words should be used to define more precisely what we are effectively trying to accomplish. Here, we follow the discussion on the history of artificial intelligence of [1].

Arguably, one of the pioneer works on artificial intelligence is [2], who defined, in mathematical terms, an artificial model for the workings of neurons. Curiously, this work is at the same time one of the pioneers of the concept of *neural network*, which is a prominent architecture for artificial intelligence systems. As [1] notes, around a decade after this groundbreaking work, a summer workshop at Dartmouth college carried out the first study in artificial intelligence. Among their goals, there were: *make machines use language*, *solve problems at the time reserved to humans*, and *improve themselves*. In a nutshell, the idea was to describe learning and other features of intelligence so precisely that a machine can be made to *simulate it*.

Here comes an important distinction. In this initial work, machines were never meant *to be intelligent*, but rather to *imitate* intelligence. This echoes Turing test, otherwise known as the *imitation game* [3]. Broadly speaking, this cornerstone thought experiment conjectures a human interrogator, which poses a set of written questions. A computer passes the Turing test if the interrogator cannot tell whether the written responses comes from a person, or a computer [1, Section 1.1.1].

Within artificial intelligence, one can take several stances on how to achieve intelligence. An inter-

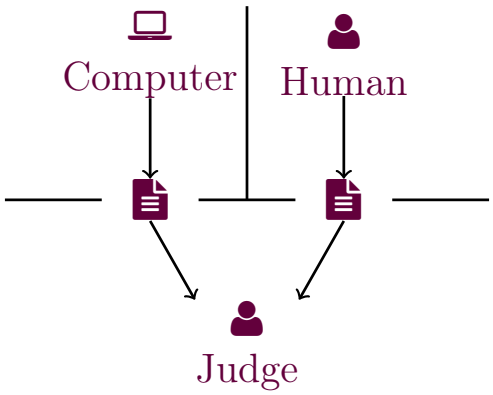


Figure 1.1 – **Illustration of the Turing test** in which a machine and a human answer textual questions for a human judge. A computer is said to be intelligent, if it can successfully imitate a human.

esting way to class these stances is between *rationalist* and *empiricist* approaches [1]. On the rationalist side, practicioners usually perform *symbolic artificial intelligence*. This approach is characterized by the representation of objects by high level symbols, and characterizing intelligence via logic. Note that this perspective on intelligence values *a priori* [2] structures over experience, as intelligence comes from the inherent way in which objects are represented. We do not consider these kinds of approaches in this thesis.

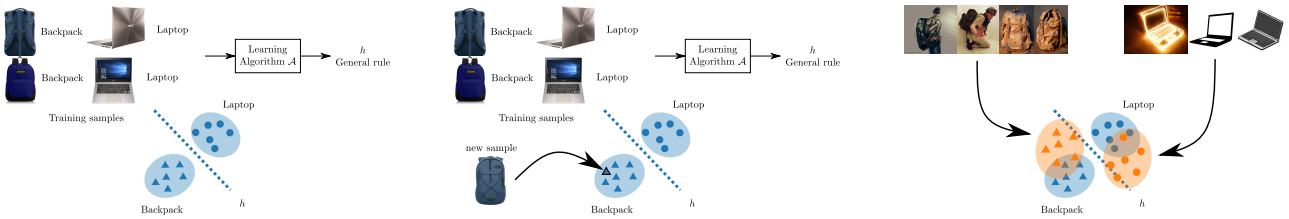
In opposition to the rationalist approach, one has *machine learning*. This kind of strategy is said to be *data-driven*, as rather than relying on high-level, expert-based representations for world objects, they seek for patterns in the available data. Here, it is possible to make a parallel between data, and human experience. Machine learning *models* work by defining a function between inputs and desired outputs. As a result, there are potentially many functions, which are ranked according a notion of *fitness* to the data at hand. As a result a model is said to learn, if it maximizes some criterion of fitness, or, equivalently, if it minimizes some notion of risk. Here, one draws a further connection between *learning* and *optimization*. This thesis takes precisely this approach to intelligence.

Here, an important issue comes into play. Since machine learning models are based on experience, the knowledge they acquire is *a posteriori* [3]. A downside of *a posteriori* knowledge is the problem of induction [4]. In rough terms, this problem can be exemplified as follows: "given that the sun has risen every morning so far, will it rise tomorrow?" From the perspective of machine learning, this question takes the form of *generalization*. Suppose a model is fit on some data. Will it be able to reliably perform predictions on unseen data? As it turns out, under reasonable conditions, the answer is yes, as laid out in *statistical learning theory* [5, 6]. We illustrate these ideas in Figure 1.2.

1. This distinction echoes the longstanding debate in Philosophy between these two fields

2. Here, we employ the term a priori in its philosophical sense. A proposition holds a priori if its independent from experience, i.e., if it is not contingent. For instance, the statement: "a square has four sides" is universal. It holds for any possible square.

3. Again, we employ this term in its philosophical sense. A posteriori knowledge is contingent on the particular context the sensations are acquired. For instance, the statement: "the chair is green" is contingent on specific examples of chairs. It holds upon the experience of a particular chair, hence it is contingent.



(a) Learning from samples. (b) Generalization. (c) Distributional Shift.

Figure 1.2 – **The problem of induction, and adaptation.** In (a), a learning algorithm learns a pattern over a finite dataset D. In this case, we use an object recognition examples, where one wants to distinguish between laptop computers and backpacks. The general rule, a classifier h, is defined over the whole latent space space. Here, we use $\mathbb{R}^2$ for simplicity of exposition. In (b), we show the idea of generalization, i.e., accurately predicting on a new example, not originally present in the training set. In (c), we show images that represent the same objects, but have different styles, poses and background, which induces a shift on where the data is likely to appear in space.

This thesis takes the question of generalization to a stress. From statistical learning, one can expect machine learning models to perform reasonably well on samples similar to training samples. In real life, however, these models are often confronted with different data representing the same concepts. Take, for instance, the task of sentiment analysis, i.e., determining if a piece of text denotes a positive or a negative sentiment. Now, suppose you have two datasets containing product reviews. The first dataset contains reviews from movies. The second, contains reviews from house products. Both of these datasets try to perform the same task, but they have particularities that make generalization to each other difficult (i.e., change in the distribution of words). This is a particular example of a problem in machine learning known as *distributional shift* [7]. An example is shown in Figure 1.2.

Put in simple terms, the goal of this thesis is to design methods that can handle distributional shift. We do so through *transfer learning*. Again, we recur to human intelligence for defining what transfer learning should be. Especially, humans can adapt to new circumstances remarkably well. For instance, for a human, it is easier to learn how to speak French, if you are already fluent in a related language, such a Portuguese, Italian or Spanish. As [8] puts, the field of transfer learning "is motivated by the fact that people can intelligently apply knowledge learned previously to solve new problems faster or with better solutions".

In this thesis, we tackle an important problem in transfer learning, known as *domain adaptation*. In this problem, one has a fixed task (e.g., sentiment analysis), but different domains (e.g., movie and houses). Furthermore, one makes a difference between *source domain* and *target domain*. The source domain represents previously acquired, solid knowledge. This usually takes the form of a reasonable amount of labeled data. The target domain represents where we want generalization to take place. Domain adaptation usually makes light requirements about the target domain. For instance, in this thesis, we perform adaptation in an unsupervised way, i.e., no labeled data is available in the target domain. As we hinted previously, we take a probabilistic or distributional view of domain adaptation. This is advantageous, as we are able to derive a mathematical characterization of how well we are

expected to perform adaptation, using the statistical learning framework.

In the spirit of providing a probabilistic treatment of domain adaptation, a natural candidate for framework is optimal transport [9, 10]. Optimal transport is a field within mathematics, that is concerned with the displacement of mass at least effort. Since its inception [11], this field has seen many resurgences and reformulations. From the perspective of this thesis, this framework is useful as it provides us a principled toolbox for manipulating probability measures.

The final piece in the puzzle this thesis is trying to solve is the consideration of multiple sources. In real world applications, users are often confronted with large, heterogeneous datasets. In such cases, one may consider different approaches. On one hand, it is possible to select a reasonably homogeneous sub-set of source data to perform adaptation, at the risk of having a sub-optimal model. On the other hand, one could consider all the heterogeneous subsets within the source dataset as a single source. This choice disregards the potential shifts occurring within the source domain. In this thesis we propose tools for leveraging Wasserstein barycenters in multi-source domain adaptation.

1.1 Summary of Contributions

From a methodological perspective, this thesis can be resumed to 3 main contributions.

Barycenters of labeled measures. Our works [12, 13] and [14] were the first to consider the problem of computing a Wasserstein barycenter of measures with samples equiped with labels. This question is challenging, as contrary to features, labels have a discrete nature. In this thesis, we represent labels continously via soft-probabilities.

Dictionary Learning. Our works [14] and [15] were the first to consider a dictionary learning problem in the space of probability measures besides histograms.

Gaussian Mixture Model-based Domain Adaptation. Our works [16] and [15] were the first to consider using the Gaussian mixture optimal transport framework of [17] for domain adaptation.

On top of these publications, we provided a set of 2 open-source libraries putting forth the reproducibility aspect of our works,

Python Distribution Learning (PyDiL) [4]**:** this library contains the implementation of our proposed algorithms.

Tennessee Eastman Domain Adaptation (TEP-DA) [5]**:** this library contains the code for building a new domain adaptation benchmark in the field of chemical engineering.

We further make public some of the code [6] for creating the figures in this thesis, and our publication [18]. This source code makes part of an effort to put forth illustrative examples of optimal transport in machine learning. Associated with these contributions, we have the following papers,

1. Eduardo Fernandes Montesuma, Fred Ngole Mboula, and Antoine Souloumiac. Recent advances in optimal transport for machine learning. arXiv preprint arXiv:2306.16156, 2023. (Chapters

4. https://github.com/eddardd/PyDiL
5. https://github.com/eddardd/tep-domain-adaptation
6. https://github.com/eddardd/examples-ot

2, 3 and 4)

2. Eduardo Fernandes Montesuma and Fred-Maurice Ngolè Mboula. Wasserstein barycenter transport for acoustic adaptation. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3405–3409. IEEE, 2021. (Chapter 5)
3. Eduardo Fernandes Montesuma and Fred Maurice Ngole Mboula. Wasserstein barycenter for multi-source domain adaptation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 16785–16793, 2021. (Chapter 5)
4. Eduardo Fernandes Montesuma, Fred Ngolè Mboula, and Antoine Souloumiac. Multi-source domain adaptation through dataset dictionary learning in wasserstein space. In 26th European Conference on Artificial Intelligence (ECAI), pages 1739–1746. IOS Press, 2023. (Chapter 6)
5. Eduardo Fernandes Montesuma, Fred Mboula, and Antoine Souloumiac. Lighter, better, faster multi-source domain adaptation with gaussian mixture models. Accepted at the Joint European Conference on Machine Learning and Knowledge Discovery in Databases 2024 (Chapter 7)
6. Eduardo Fernandes Montesuma, Fred Maurice Ngolè Mboula, and Antoine Souloumiac. Optimal transport for domain adaptation through gaussian mixture models [7]. arXiv:2403.13847, 2024. (Chapter 7)
7. Eduardo Fernandes Montesuma, Michela Mulas, Fred Ngolè Mboula, Francesco Corona, and Antoine Souloumiac. Benchmarking Domain Adaptation for Chemical Processes on the Tennessee Eastman Process [8]. Accepted at the Workshop Machine Learning for Chemistry and Chemical Engineering (ML4CCE) at the Joint European Conference on Machine Learning and Knowledge Discovery in Databases 2024 (Chapter 8)
8. Eduardo Fernandes Montesuma, Fred Ngolè Mboula, and Antoine Souloumiac. Multi-source domain adaptation meets dataset distillation through dataset dictionary learning. In ICASSP2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pages 5620–5624. IEEE, 2024. (Chapter 10)

The following paper are discussed in the conclusion as future works, but were not formally described in the body of the manuscript,

9. Eduardo Fernandes Montesuma, Fabiola Espinoza Castellon*, Fred Ngolè Mboula, Aurélien Mayoue, Antoine Souloumiac, and Cédric Gouy-Pailler. Federated dataset dictionary learning for multi-source domain adaptation. In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5610–5614. IEEE, 2024.
10. Eduardo Fernandes Montesuma, Fabiola Espinoza Castellon, Fred Ngolè Mboula, Aurélien Mayoue, Antoine Souloumiac, and Cédric Gouy-Pailler. Dataset Dictionary Learning in a Wasserstein Space for Federated Domain Adaptation.
11. Eduardo Fernandes Montesuma, Stevan le Stanc, and Fred Ngolè Mboula. Online multi-source domain adaptation through gaussian mixtures and dataset dictionary learning. Accepted at the IEEE 34rd International Workshop on Machine Learning for Signal Processing (MLSP)

7. This work was presented at the 55th Journée de Statistique de la Socieeté Française de Statistique.
8. An earlier version of this work was published at ArXiV under the title "Multi-source domain adaptation for cross-domain fault diagnosis of chemical processes".

Besides these, the following paper was produced during this thesis, but ultimately did not figure in the manuscript,

12. Eduardo Fernandes Montesuma, Michela Mulas, Francesco Corona, and Fred-Maurice Ngole Mboula. Cross-domain fault diagnosis through optimal transport for a cstr process. IFAC-PapersOnLine, 55(7):946–951, 2022.

Furthermore, during the period of this thesis, I participated as a reviewer in the following conferences and journals,

— 18th European Conference on Computer Vision
— IEEE Transactions on Image Processing
— Engineering Applications of Artificial Intelligence (Elsevier)
— 12th International Conference on Learning Representations
— 41st International Conference on Machine Learning
— IEEE/CVF Conference on Computer Vision and Pattern Recognition 2024
— 27th International Conference on Artificial Ingelligence and Statistics
— 20th International Conference on Learning Representations
— 37th Conference on Neural Information Processing Systems
— 26th European Conference on Artificial Intelligence
— 26th International Conference on Artificial Intelligence and Statistics (AISTATS). Top 10% Reviewers award [9].
— 2022 IEEE International Conference on Robotics and Automation (ICRA)

1.2 Thesis Structure

1.2.1 Part I

Part Overview. The first part of this thesis aims at giving a broad view on the theoretical foundations of our work. We divided the foundations in 3 chapters. Chapter 2 covers optimal transport, chapter 3 covers barycenters of probability measures and chapter 4 covers domain adaptation. Different aspects of these chapters can be more or less found in our review [18]. In the following, we detail the contents of each chapter.

Chapter 2: the many faces of Optimal Transport. This chapter introduces optimal transport theory. Here, our main references are [19], for general optimal transport concepts, and [10], for computational methods. We take a pragmatic approach to optimal transport theory, namely, while we introduce some level of abstraction, we illustrate optimal transport on Euclidean spaces, as this is what is needed for our algorithms. With respect to the kinds of measures that we treat, we mainly focus on three types: empirical, Gaussian and Gaussian mixtures. From the computational perspective, we consider recent topics in optimal transport, such as mini-batch, unbalanced and partial optimal transport.

Chapter 3: barycenters of probability measures. This chapter treats the main protagonist of this thesis, namely, barycenters. To that end, we need to define distances in the space of probability

9. http://aistats.org/aistats2023/reviewers.html

measures. Here, we focus on three kinds of distances, which are widely used in domain adaptation: $\mathcal{H}-$distance, maximum mean discrepancy, and the Wasserstein distance. Since we are also dealing with empirical measures, we discuss how to estimate these distances from finite samples. Furthermore, we provide a discusison on the computational aspects of computing Wasserstein barycenters, which will be useful in upcoming chapters.

Chapter 4: learning and domain adaptation theory. This chapter considers two problems. The first, is learning theory, i.e., how to learn a classifier, provided samples from a probability measure. The deeper question concerns generalization, that is, the hability of the trained classifier to perform well on unseen samples *from the same measure*. Here, the main reference is [5, 6], who established the foundations of *statistical learning theory*. The second problem, domain adaptation, can be seen as a generalization of the first. In a nutshell, it considers learning problems where data comes from different probability measures, that is, learning under distributional shift. For this problem, the main reference is [8] and [20], who described the problem in mathematical terms. A more recent, comprehensive survey was presented by [21, 22]. In this chapter, our goal is to establish the main theory and algorithms behind domain adaptation.

1.2.2 Part II

Part Overview. This part establishes our methodological contributions in multi-source domain adaptation, i.e., domain adaptation when the source domain is composed of multiple, heterogeneous measures. We divide it into three chapters. Chapter 5 covers our algorithm Wasserstein Barycenter Transport, which solves a multi-source problem by first calculating a barycenter of source measures, then solving a single-source problem between the barycenter and the target. Chapter 6 covers our new framework, *Dataset Dictionary Learning*, which performs dictionary learning over empirical measures. This framework approximates each measure in multi-source domain adaptation as a Wasserstein barycenter of learnable measures. Finally, Chapter 7 presents a parametric reformulation of the previous algorithms. These chapters are based on our publications [12, 13, 14, 15, 16].

Chapter 5: Wasserstein Barycenter Transport. [12, 13] In this chapter, we present an updated view of our papers [12, 13]. Especially, we provided a broader view on optimal transport between labeled measures. We introduce a new Wasserstein-like distance over the space of distributions over the feature-label joint space, called joint-Wasserstein distance. We further show that the first-order conditions of this metric correspond to the barycentric map and label propagation equations used by [23] and [24] (Theorem 9). We then establish a comparison between this metric, and the distance introduced by [25]. We then derive a new algorithm, inspired by the free-support algorithm of [26], for computing the Wasserstein barycenter of labeled empirical measures (algorithm 6).

Chapter 6: Dataset Dictionary Learning [14]. This chapter presents a review of optimal transport-based dictionary learning, especially the work of [27]. We then present the work of [28], who estimates a vector of barycentric coordinates. These coordinates give the measure, in the Wasserstein hull of source measures, that best approximates a target measure. We then start building our new dictionary learning framework, called *Dataset Dictionary Learning*. First, we extend the work of [28] to handle free-support empirical measures. We do so in algorithms 7 and 8. Then, we propose our new frame-

work in section 6.2, by learning a set of atoms that interpolate the measures in multi-source domain adaptation.

Chapter 7: Domain adaptation with Gaussian mixtures. [16, 15] In this chapter, our main reference is the work of [17], who first formalized a version of the optimal transport problem for Gaussian mixtures. These authors introduced a new metric in the space of Gaussian mixtures, called mixture-Wasserstein distance. We start by discussing how to map the parameters of Gaussian mixtures through optimal transport. This creates a new label propagation strategy for these mixtures, based on a transport plan between their components. We further discuss the estimation of a mapping between domains through optimal transport. These ideas correspond to our paper [16].

After this initial discussion, we build new tools for the computation of mixture-Wasserstein barycenters. So far, these barycenters were calculated through multi-marginal optimal transport, which scales poorly with the number of components in the mixtures. We do a similar step to [26], and propose a fast, fixed-point algorithm for computing the parameters of the barycentric mixture (section 7.3). Then, we introduce a new metric for Gaussian mixtures that have labeled components. We call this metric joint mixture-Wasserstein distance, in reference to the empirical case presented in Chapter 5. As in the empirical case, the first-order optimality conditions under this metric can be understood as barycentric maps of means, standard deviation vectors and labels. Based on this, we build Gaussian mixture versions of our previous algorithms. Especially, the Gaussian mixture version of dataset dictionary learning is lighter, faster and better than its empirical counterpart (see examples 24, 26 and 25, respectively).

1.2.3 Part III

Part Overview. The last part of this thesis is concerned with experimentation. In Chapter 8 we introduce a new benchmark in domain adaptation for chemical engineering, based on the Tennessee Eastman process [29]. Chapter 9 performs the benchmarking of our methods in comparison with other state-of-the-art algorithms. Finally, chapter 10 covers our experiments in dataset distillation [30].

Chapter 8: Cross-Domain Fault Diagnosis [31]. In this section, we analyze the Tennessee Eastman process of [29]. We then describe a series of pre-processing steps on the data of [32] for extending their simulations to a domain adaptation scenario. We then benchmark different methods in the domain adaptation literature on this benchmark, showing that multi-source domain adaptation methods based on Wasserstein barycenters outperform other kinds of methods.

Chapter 9: Benchmarking Multi-Source Domain Adaptation. In this chapter, we use 5 existing benchmarks in domain adaptation to compare optimal transport-based methods in single-source and multi-source domain adaptation. We further analyze the domain adaptation performance over the interpolation space defined by Wasserstein barycenters. Then analyze the performance of WBT, DaDiL and their GMM counterparts with respect to their hyper-parameters.

Chapter 10: Dataset Distillation [33]. In this chapter, we explore a feature of free-support Wasserstein barycenters, that is, the fact that the number of samples in their support is a free parameter. As a result, we show that we are able to perform domain adaptation while compressing the target domain. More surprisingly, in the context of the Tennessee Eastman benchmark, we are able to achieve

state-of-the-art performance while reducing 99.84% of the total amount of samples.

Part I

Theoretical Foundations

Chapter 2

The Many Faces of Optimal Transport

Contents

Optimal transport is a mathematical theory founded by the 18th century French Mathematician Gaspard Monge [11]. As discussed by [9], Monge's first description of the Optimal Transport (OT) problem was concerned with literal displacement of masses. Initially, one has two known places: a *source*, from which mass is taken and transported to a *target*, to be incorporated into a construction. While the places where the material needs to be extracted from, and where it should be transported to are known, one does not known *the optimal transport assignment* between the two. This assignment can be understood as an optimal transport map between the source and the target.

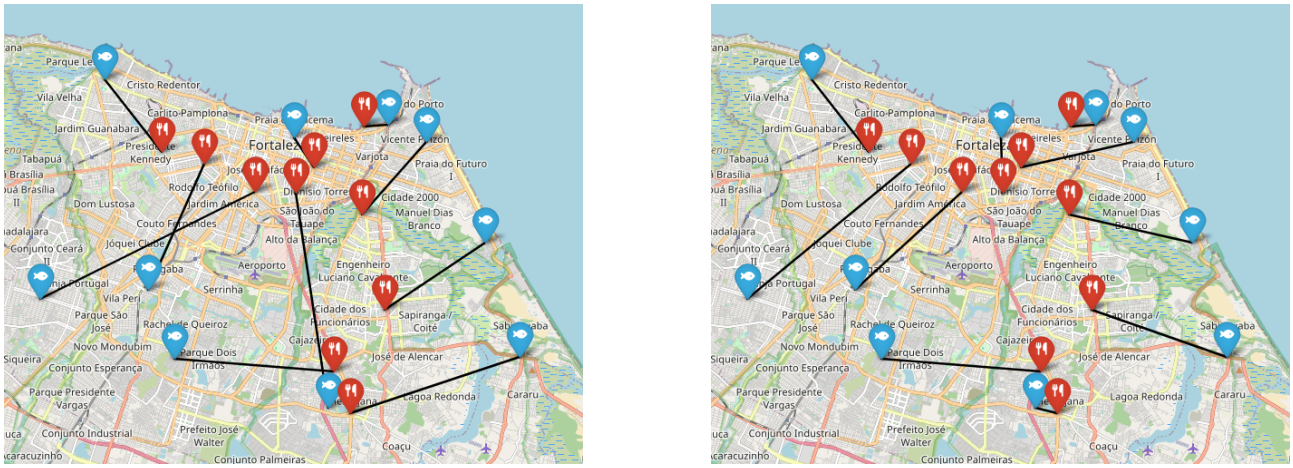
Two centuries later, the optimal transport problem was rediscovered, under a completely different guise, by Leonid Vitalyevich Kantorovich [34]. While Monge looked at optimal transportation from the perspective of an optimal transport map, Kantorovich formulated the problem in terms of mass allocation. Under Kantorovich, mass is not sent to a single place in the target, but rather distributed along possible locations. As a result, *one loses the notion of transport map, but gains the much more flexible notion of transport plan.* Before proceeding, we illustrate these ideas.

Example 1. *(Shipper's Problem) In Fortaleza, consider n restaurants located at $\mathbf{x}_1^{(P)}, \cdots, \mathbf{x}_n^{(P)}$, and m fish markets located at $\mathbf{x}_1^{(Q)}, \cdots, \mathbf{x}_m^{(Q)}$, where $\mathbf{x}_i^{(P)}, \mathbf{x}_j^{(Q)} \in \mathbb{R}^2$. Each fish market j has q_j units of fish, and each restaurant i has a demand of p_i in order to cook dishes for the day. The cost of shipping γ_{ij} units of fish from market j to restaurant i is $\gamma_{ij} C_{ij}$, where C_{ij} is called the* ground-cost. *In this example, we assume $C_{ij} = \|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2$, i.e., the Euclidean distance. The optimal transport problem*

consists on finding coefficients $\{\gamma_{ij}^{\star}\}_{i,j=1}^{n,m}$, *such that,*

$$\begin{aligned} \gamma^{\star} = \underset{\{\gamma_{ij}\}_{i,j=1}^{n,m}}{argmin} \sum_{i=1}^{n}\sum_{j=1}^{m}\gamma_{ij}\|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2, \\ subject\ to\ \sum_{i=1}^{n}\gamma_{ij} = q_j,\ \sum_{j=1}^{m}\gamma_{ij} = p_i,\ and\ \gamma_{ij} \geq 0, \end{aligned} \tag{2.1}$$

which is a linear program involving nm *variables (i.e., each* γ_{ij}*), and* $n+m-1$ *constraints. To illustrate these concepts, we let* $n = m = 10$ *restaurants over the city of Fortaleza, and* $p_i = q_j = 1$*, i.e., each restaurant needs to buy one fish, and each market needs to sell one fish. We compare optimal transport with a greedy strategy: starting from* $i = 1$*, each restaurant buys its fish from the nearest market not previously chosen by other restaurants. In these conditions, while the greedy strategy has a cost of* 5.6*, the optimal transport strategy has a cost of* 4.8*. An illustration for these strategies is shown in Figure 2.1.*



(a) Greedy strategy.

(b) Optimal transport strategy.

Figure 2.1 – Comparison of greedy (a) and optimal transport (b) strategies for matching fish markets and restaurants.

While the previous example illustrates optimal transport from an economical point of view, in the following our exposition focus on an probabilistic angle. As such, we present optimal transport theory from the perspective of probability measures. Following [35], given a set $\mathcal{X}$ and a $\sigma-$Algebra $\mathcal{F}$ on $\mathcal{X}$, the pair $(\mathcal{X}, \mathcal{F})$ is called a measurable space. A probability measure $P : \mathcal{F} \to [0, 1]$ is a function that assigns probabilities to subsets of $\mathcal{X}$, and that suffices,

$$P(\emptyset) = 0,\ P(\mathcal{X}) = 1,\ \text{and},\ P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i),$$

for a countable collection of disjoint sets $\{A_i\}_{i=1}^{\infty}$ in $\mathcal{F}$. To ground intuition about these notions, one can consider $\mathcal{X} = \mathbb{R}^d$, and $\mathcal{F} = \mathcal{B}(\mathbb{R}^d)$ as the collection of open sets of $\mathbb{R}^d$. Associated with P, one has

the notion of probability density,

$$P(A) = \int_A dP(x) = \int_A p(x)dx,$$

note that, henceforth, we use $dP(x) = p(x)dx$. Furthermore, we denote by $\mathbb{P}(\mathcal{X})$, the set of all probability measures over a set $\mathcal{X}$.

2.1 Optimal Transport Theory

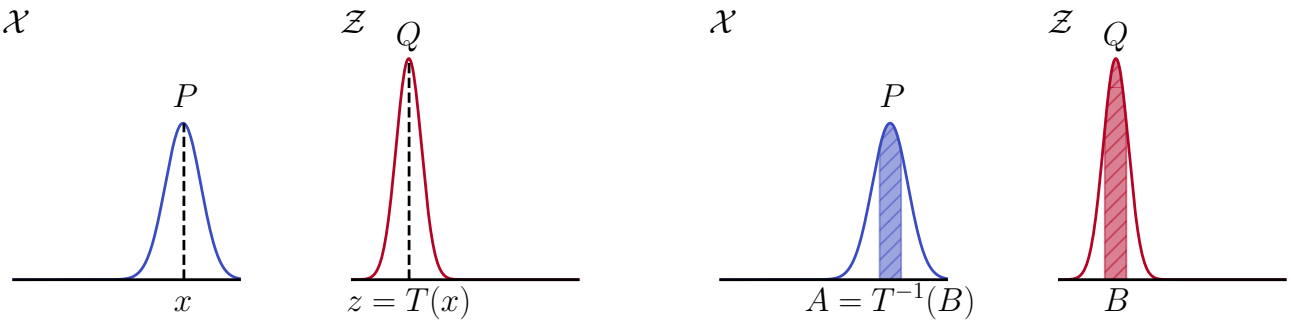
2.1.1 Monge Formulation

We start with the concept of push-forward of a probability measure.

Definition 1. *(Push-Forward Measure) Let $(\mathcal{X}, \mathcal{B}(\mathcal{X}), P)$ and $(\mathcal{Z}, \mathcal{B}(\mathcal{Z}), Q)$ be two probability spaces, and p, q be the corresponding densities of P and Q. For each measurable function $T : \mathcal{X} \to \mathcal{Z}$, there corresponds an operator $T_\sharp : \mathbb{P}(\mathcal{X}) \to \mathbb{P}(\mathcal{Z})$ such that $P \mapsto Q = T_\sharp P$ as follows,*

$$Q(B) = \int_B q(z)dz = (T_\sharp P)(B) = P(T^{-1}(B)) = \int_A p(x)dx \tag{2.2}$$

where $A = T^{-1}(B)$, for $A \in \mathcal{B}(\mathcal{X})$ and $B \in \mathcal{B}(\mathcal{Z})$.



(a) T (b) $T_\sharp$

Figure 2.2 – Comparison between the mapping $T : \mathcal{X} \to \mathcal{Z}$ and the push-forward $T_\sharp : \mathbb{P}(\mathcal{X}) \to \mathbb{P}(\mathcal{Z})$. As described by [10], while the transformation moves points $x \in \mathcal{X}$ to $z \in \mathcal{Z}$, the push-forward can be understood as moving the whole measure $P \in \mathbb{P}(\mathcal{X})$ to $Q \in \mathbb{P}(\mathcal{Z})$. Note that, if T maps $A \subset \mathcal{X}$ to $B \subset \mathcal{Z}$, then the push-forward ties the masses $P(A)$ and $Q(B)$.

Using the established concepts of probability measure and push-forward of a measure, the transportation problem between measures $P \in \mathbb{P}(\mathcal{X})$ and $Q \in \mathbb{P}(\mathcal{Z})$ can be understood as the search of a map $T : \mathcal{X} \to \mathcal{Z}$ such that $T_\sharp P = Q$ and that is optimal with respect to a transport effort. This effort is measured with respect to a *ground-cost*, $c : \mathcal{X} \times \mathcal{Z} \to [0, +\infty]$ between elements $x \in \mathcal{X}$ and $z \in \mathcal{Z}$. These are the ingredients of the Monge problem.

Definition 2. *(Monge Problem) Given two probability measures $P \in \mathbb{P}(\mathcal{X})$ and $Q \in \mathbb{P}(\mathcal{Z})$, and a ground-cost $c : \mathcal{X} \times \mathcal{Z} \to [0, +\infty]$, the OT mapping $T^\star : \mathcal{X} \to \mathcal{Z}$ is given by*

$$T^\star = \underset{T_\sharp P = Q}{arginf} \underset{x \sim P}{\mathbb{E}}[c(x, T(x))] := \int_{\mathcal{X}} c(x, T(x)) dP(x), \tag{2.3}$$

Using figure 2.2 as a support, we can understand the constraint $T_\sharp P = Q$ as a *probability mass conservation constraint*, i.e., when T maps $A \subset \mathcal{X}$ to $B \subset \mathcal{Z}$, the probability mass of these sets measured by P and Q should be equal. This bounds the idea that T moves the mass of P into Q.

At this point, it is worthwhile mentioning that a few technicalities that plague the Monge problem. First, dealing with transportation maps is very restrictive. To illustrate this issue, take $P = \delta(x_0) \in \mathbb{P}(\mathbb{R})$. From equation 2.2, it is immediate that $T_\sharp P = \delta(T(x_0))$. As a result, if Q is not of the form $\delta(z_0)$, the set $\{T : T_\sharp P = Q\} = \emptyset$, i.e., the Monge problem has no solutions. Second, for a sequence of maps $(T_n)_{n=1}^\infty$, each sufficing $T_{n,\sharp} P = Q$, and $T_n \to T$, one may have $T_\sharp P \neq Q$ [19, Chapter 1]. Before proceeding, let us state an example in which the Monge problem has closed form solution.

Example 2. *(Optimal Transport between Gaussian Measures) Here, we consider optimal transport with $c(\mathbf{x}, \mathbf{z}) = |\mathbf{x} - \mathbf{z}|^2$, and between distributions P and Q with densities $\mathcal{N}(\mathbf{x}|\mu_P, \Sigma_P)$ and $\mathcal{N}(\mathbf{z}|\mu_Q, \Sigma_Q)$, $\Sigma_P, \Sigma_Q \in \mathcal{S}^d = \{\mathbf{M} \in \mathbb{R}^{d \times d} : \mathbf{M} = \mathbf{M}^T, \mathbf{x}^T \mathbf{M} \mathbf{x} > 0, \forall \mathbf{x} \in \mathbb{R}^d \backslash \{\mathbf{0}\}\}$, the manifold of Symmetric Positive Definite (SPD) matrices and,*

$$\mathcal{N}(\mathbf{x}|\mu, \Sigma) = \frac{1}{\Big((2\pi)^d det(\Sigma)\Big)^{1/2}} \exp\Big(-1/2(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\Big).$$

In these conditions, the optimal transport map $T^\star$ exists and is affine,

$$T(\mathbf{x}) = (\mu_Q - \mathbf{A}\mu_P) + \mathbf{A}\mathbf{x}, \tag{2.4}$$

where $\mathbf{A} = \Sigma_P^{-1/2} (\Sigma_P^{1/2} \Sigma_Q \Sigma_P^{1/2})^{1/2} \Sigma_P^{-1/2}$ (see e.g., [10, Remark 2.31] or [36]). In Figure 2.3 we show an example of the map T between two Gaussian measures.

2.1.2 Kantorovich Formulation

Contrary to the Monge problem, Kantorovich consider optimal transport under transportation plans. Here, we consider transportation plans through the lens of probability theory. Given $P \in \mathbb{P}(\mathcal{X})$ and $Q \in \mathbb{P}(\mathcal{Z})$, a transportation plan is a measure $\gamma \in \mathbb{P}(\mathcal{X} \times \mathcal{Z})$ such that,

$$P(A) = \int_{\mathcal{Z}} \gamma(A, z) dz, \text{ and } Q(B) = \int_{\mathcal{X}} \gamma(x, B) dx. \tag{2.5}$$

Given P and Q, one can further consider the set of all transportation plans for which equation 2.5 holds, i.e., $\Gamma(P, Q)$. Note that this set is non-empty. Indeed, if p is the density of P and q is the density of Q, then $\gamma(x, z) = p(x)q(z) \in \Gamma(P, Q)$. Based on these ideas, the Kantorovich problem is,

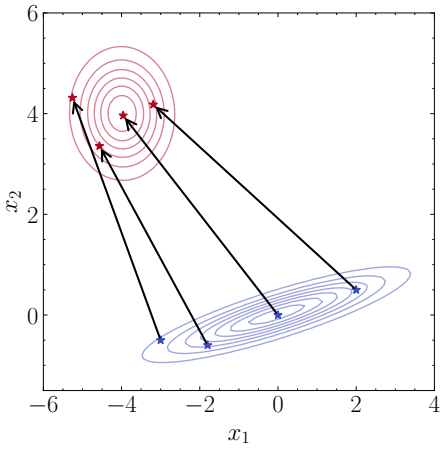


Figure 2.3 – Optimal transport map between Gaussian measures. Blue and red denote distributions P and Q respectively. The stars denote samples, whereas contours show the level sets of the densities of P and Q.

Definition 3. *(Kantorovich Problem) Let $P \in \mathbb{P}(\mathcal{X})$ and $Q \in \mathbb{P}(\mathcal{Z})$ be two probability measures, and let $c : \mathcal{X} \times \mathcal{Z} \to [0, +\infty]$ be a ground-cost. The OT plan $\gamma^\star$ is given by*

$$\gamma^\star = \underset{\gamma \in \Gamma(P,Q)}{\operatorname{arginf}} \underset{(x,z)\sim\gamma}{\mathbb{E}} [c(x, z)] := \int_{\mathcal{X}\times\mathcal{Z}} c(x, z) d\gamma(x, z) \tag{2.6}$$

Under the Kantorovich formulation, optimal transport becomes an *infinite-dimensional linear program*. Indeed, equations 2.6 and 2.5 are linear with respect $\gamma(x, z)$. As a side note, Kantorovich himself is one of the founding fathers of linear programming. From optimization theory, it is quite natural to consider a dual problem. Let $\phi : \mathcal{X} \to \mathbb{R}$ and $\psi : \mathcal{Z} \to \mathbb{R}$ be two continuous functions. We further define the set,

$$\Phi_c := \{(\phi, \psi) \in \mathcal{C}_0(\mathcal{X}) \times \mathcal{C}_0(\mathcal{Z}) : \phi(x) + \psi(z) \leq c(x, z)\}, \tag{2.7}$$

where $\mathcal{C}_0(\mathcal{X})$ is the set of continuous functions over $\mathcal{X}$, vanishing at infinity.

The dual Kantorovich problem revolves around ϕ and ψ, known in the literature as Kantorovich potentials. These variables have an interesting economical interpretation: ϕ and ψ can be understood as the cost of sending a resource from location $x \in \mathcal{X}$, and the cost of receiving a resource at a location $z \in \mathcal{Z}$, respectively. For this strategy to have a sense, the cost of shipping, $\phi(x) + \psi(z)$, must be at most $c(x, z)$, i.e., the cost of transportation from x to z.

Definition 4. *(Dual Kantorovich Problem) Let $P \in \mathbb{P}(\mathcal{X})$ and $Q \in \mathbb{P}(\mathcal{Z})$ be probability measures. For a ground-cost $c : \mathcal{X} \times \mathcal{Z} \to [0, +\infty]$, the optimal Kantorovich potentials are given by,*

$$(\phi^\star, \psi^\star) = \underset{(\phi,\psi)\in\Phi_c}{\operatorname{argsup}} \int_{\mathcal{X}} \phi(x) dP(x) + \int_{\mathcal{Z}} \phi(z) dQ(z). \tag{2.8}$$

An interesting simplification of equation 2.8 arises when $\mathcal{X} = \mathcal{Z} \subset \mathbb{R}^d$, and c is a metric over $\mathbb{R}^d$ (e.g., the Euclidean distance). In this case, one can use convex analysis to show that the constraint

$(\phi, \psi) \in \Phi_c$ can be translated into $\phi \in \text{Lip}_1 = \{f : \mathbb{R}^d \to \mathbb{R} : |f(\mathbf{x}) - f(\mathbf{z})| \leq c(\mathbf{x}, \mathbf{z})\}$, $\psi = -\phi$, and Lip_1 denotes the set of functions with unitary Lipschitz norm. As a result, one may rewrite the Dual Kantorovich formula into what is known as the Kantorovich-Rubinstein formulation,

Definition 5. *(Kantorovich-Rubinstein Formulation) Let $P, Q \in \mathbb{P}(\mathbb{R}^d)$, and c be a metric on $\mathbb{R}^d$. The dual Kantorovich problem in equation 2.8 admits a reformulation as,*

$$\phi^\star = \sup_{\phi \in Lip_1} \int_{\mathbb{R}^d} \phi d(P - Q) = \sup_{\phi \in Lip_1} \mathbb{E}_{\mathbf{x} \sim P}[\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim Q}[\phi(\mathbf{x})], \tag{2.9}$$

The Kantorovich-Rubinstein formulation made a significant impact in the machine learning community, especially through [37], in which the authors propose parametrizing ϕ through a neural net. This approach was further extended for domain adaptation by [38]. In both of these works, the constraint $\phi \in \text{Lip}_1$ is non-trivial to enforce, and is an active field of research in the community of generative modeling.

2.1.3 Metric Optimal Transport

When defining optimal transport, we discussed the notion of transportation effort. For instance, using Kantorovich's formulation, an optimal transport cost can be defined in terms of the OT plan,

$$\mathcal{T}_c(P, Q) = \inf_{\gamma \in \Gamma(P,Q)} \int_{\mathcal{X} \times \mathcal{Z}} c(x, z) d\gamma(x, z). \tag{2.10}$$

In Euclidean spaces (i.e., $\mathcal{X} = \mathcal{Z} \subset \mathbb{R}^d$), this notion takes a physical interpretation. Let $c(\mathbf{x}, \mathbf{z}) = \|\mathbf{x} - \mathbf{z}\|_2$. The plan γ can be interpreted as *how much mass* is moved from $\mathbf{x} \in \mathbb{R}^d$ to $\mathbf{z} \in \mathbb{R}^d$. As a result, the innermost part of the integral in equation 2.10 is proportional to the energy necessary to displace a differential of mass from P into Q.

These ideas motivate a notion of distance based on optimal transport, known as Wasserstein distance. Before delving into its definition, we define the notion of Wasserstein space,

Definition 6. *(Wasserstein Space [9]) Let $(\mathcal{X}, d)$ be a metric space, and $\alpha \in [1, +\infty)$. The $\alpha-$Wasserstein space, $\mathbb{W}_\alpha$ over $\mathcal{X}$ is a sub-set of $\mathbb{P}(\mathcal{X})$,*

$$\mathbb{W}_\alpha(\mathcal{X}) := \left\{ P \in \mathbb{P}(\mathcal{X}) : \mathop{\mathbb{E}}_{x \sim P}[d(x, x_0)^\alpha] = \int_{\mathcal{X}} d(x, x_0)^\alpha dP(\mathbf{x}) \leq +\infty \right\},$$

where $x_0 \in \mathcal{X}$ is arbitrary.

This restriction on the space of measures $\mathbb{P}(\mathcal{X})$ is needed for enforcing the definiteness of the following metric,

Definition 7. *(Wasserstein Distance) Let $(\mathcal{X}, d)$ be a metric space, and let $\alpha \in [1, +\infty)$. For any $P, Q \in \mathbb{P}(\mathcal{X})$, the Wasserstein distance of order α between P and Q is,*

$$\mathcal{W}_\alpha(P, Q) = \left(\inf_{\gamma \in \Gamma(P,Q)} \int_{\mathcal{X} \times \mathcal{X}} d(x, z)^\alpha d\gamma(x, z) \right)^{1/\alpha}. \tag{2.11}$$

The $\alpha-$Wasserstein distance defines a metric on $\mathbb{W}_\alpha$, directly associated with the cost of transportation $\mathcal{T}_c$. Indeed, $\mathcal{W}_\alpha = (\mathcal{T}_{d^\alpha})^{1/\alpha}$. Before proceeding, we present an informal reasoning about why $\mathcal{W}_\alpha$ is a metric [19].

First, note that $d(x,z) \geq 0$ and $\gamma(x,z) \geq 0$, $\forall x, z \in \mathcal{X}$. As a result, $\mathcal{W}_\alpha$ is a sum (an integral) of non-negative terms, which is itself non-negative. Second, $\mathcal{W}_\alpha(P,Q) = 0$ implies the existence of γ concentrated on $\{x, z \in \mathcal{X} : x = z\}$, which implies $P = Q$. Third, one needs to prove the triangle inequality, i.e. $\mathcal{W}_\alpha(P,Q) \leq \mathcal{W}_\alpha(P,Q') + \mathcal{W}_\alpha(Q',Q)$. This part is somewhat more technical, and relies on the notions of disintegration of measures and the *gluing lemma*. The idea is to construct $\gamma_0 \in \Gamma(P,Q)$ by gluing together $\gamma_1 \in \Gamma(P,Q')$ and $\gamma_2 \in \Gamma(Q',Q)$. A formal treatment of these results can be found in [19, Lemmas 5.4. and 5.5.].

Now, let us start to build some intuition about $\mathcal{W}_\alpha$. The Wasserstein distance is said to *lift* [39] the metric d over $\mathcal{X}$, to $\mathcal{W}_\alpha$ over $\mathbb{W}_\alpha(\mathcal{X})$, which is illustated in Figure 2.4. An interesting fact about this space is that it contains a copy of $\mathcal{X}$ [39]. Indeed, for $x_0, z_0 \in \mathcal{X}$, with $P(x) = \delta(x - x_0)$ and $Q(z) = \delta(z - z_0)$, $\mathcal{W}_\alpha(P,Q) = d(x_0,z_0)^\alpha$. In other words, with respect the Wasserstein distance, the mapping $x_0 \mapsto \delta(x - x_0)$ is isometric. These ideas help understanding how geometric concepts in $\mathcal{X}$, such as barycenters, may be transported to the much more abstract space $\mathbb{W}_\alpha$. In the next example, we analyze geometry over the manifold of Gaussian measures. For further comparisons, we refer readers to Chapter 3, section 3.1.3.

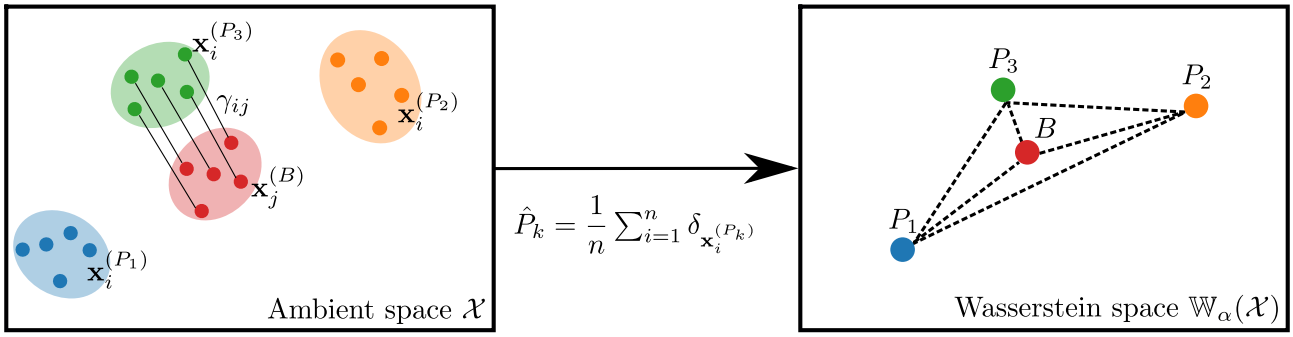


Figure 2.4 – **Comparison between the ambient space, and its associated Wasserstein space.** Given groups of points over an ambient space $\mathcal{X}$, the Wasserstein distance induces a geometry over the Wasserstein space, $\mathbb{W}_\alpha(\mathcal{X})$, associated with $\mathcal{X}$.

Example 3. *(Wasserstein distance between Gaussian Measures) In the same conditions of Example 2, let P and Q be Gaussian measures with μ_P, μ_Q, Σ_P, and Σ_Q. In this case, the $2-$Wasserstein distance with $c(\mathbf{x},\mathbf{z}) = \|\mathbf{x} - \mathbf{z}\|_2$ has closed form,*

$$\mathcal{W}_2(P,Q)^2 = \|\mu_P - \mu_Q\|^2 + Tr\Big(\Sigma_P + \Sigma_Q - 2(\Sigma_P^{1/2}\Sigma_Q\Sigma_P^{1/2})^{1/2}\Big). \tag{2.12}$$

Under especial circumstances, this expression can be further simplified. For instance, if $\Sigma_P = diag((\sigma_P^2)_i)$ and $\Sigma_Q = diag((\sigma_Q^2)_i)$ the distance becomes,

$$\mathcal{W}_2(P,Q)^2 = \|\mu_P - \mu_Q\|_2^2 + \|\sigma_P - \sigma_Q\|_2^2. \tag{2.13}$$

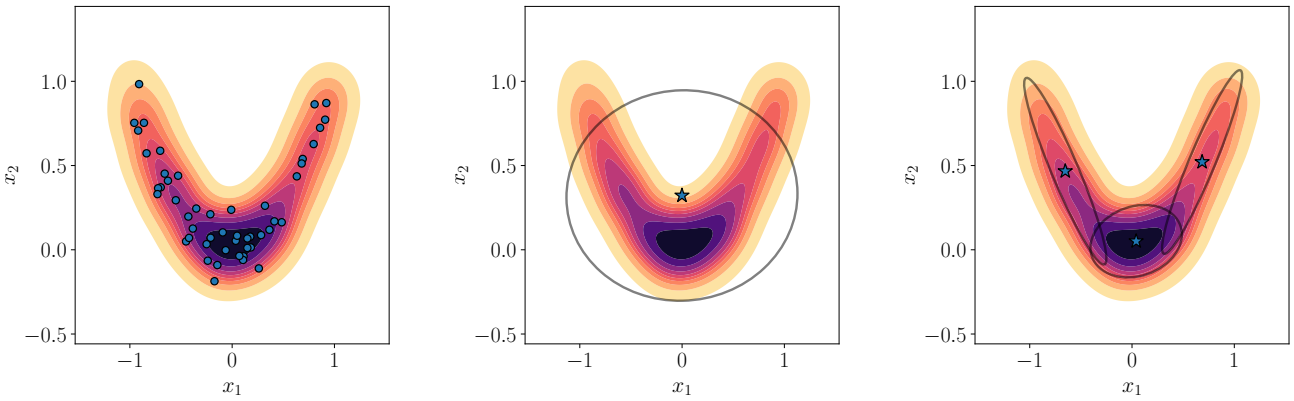
Otherwise, if $d = 1$, the Wasserstein distance yields an Euclidean geometry for $\mathbb{W}_2$,

$$\mathcal{W}_2(P, Q)^2 = (\mu_P - \mu_Q)^2 + (\sigma_P - \sigma_Q)^2. \tag{2.14}$$

This latter expression shows that, under the 2−Wasserstein metric, the manifold of Gaussian measures is isometric to the half positive plane $(\mu, \sigma) \in \mathbb{R} \times \mathbb{R}_+$ equipped with the Euclidean metric.

2.2 Computational Optimal Transport

Computational optimal transport is an active field of research within machine learning and computer science. We refer readers to [40, 10] for its foundations, and to [41, 42] for widely used software. Before treating computational optimal transport, we need to define how a computer can process its problem. In this thesis, we consider two strategies: (i) non-parametric, and (ii) parametric representation of measures. These strategies are illustrated in Figure 2.5, where we show two parametric models for the data, i.e., Gaussian and Gaussian mixtures. While non-parametric estimation offers flexibility, parametric models may be more computationally and statistically efficient [1].



(a) Empirical measure. (b) Gaussian measure. (c) Gaussian mixture measure.

Figure 2.5 – **Parametric and non-parametric approximation of measures.** In (a), blue points denote samples over an unknown measure P (underlying heatmap). In (b), we show the mean (blue star) and 1-standard deviation for a Gaussian measure estimated from samples. In (c), we show the means (blue stars) and covariances of a Gaussian mixture measure learned from data.

The first strategy relies on discretizations of $\mathcal{X}$. This idea may be sub-divided further into two schemes: (i) binning the ambient space, and (ii) approximating the measures from samples. In both cases, the next definition is useful.

1. Computational efficiency corresponds to running time and the amount of storage required by methods. Statistical efficiency refers to the number of samples needed to obtain an accurate prediction.

Definition 8. *(Empirical Measure) Let $\mathcal{X}$ be a set, $P \in \mathbb{P}(\mathcal{X})$, and $x_i^{(P)} \sim P$ with probability $p_i \geq 0$. The* empirical approximation $\hat{P}$ *of P is,*

$$\hat{P}(x) = \sum_{i=1}^{n} p_i \delta(x - x_i^{(P)}), \tag{2.15}$$

where $p_i \geq 0, \forall i$, and $\sum_i p_i = 1$. We may further denote $\mathbf{p} \in \Delta_n = \{\mathbf{a} \in \mathbb{R}_+^n : \sum_i a_i = 1\}$.

As follows, the binning strategy considers n fixed bins $X^{(P)} = [x_1^{(P)}, \cdots, x_n^{(P)}]$ over $\mathcal{X}$. The weights p_i represent the weight of the bin $x_i^{(P)}$ or the probability density function $p(x)$. Conversely, one can sample $x_i^{(P)} \overset{i.i.d.}{\sim} P$. In this case, $p_i := 1/n$ are fixed. These two strategies are often called fixed and free-support, respectively.

The second strategy relies on a parametric specification of $\hat{P}$. For instance, one can assume a Gaussian model for the density of P, i.e., $p_\theta = \mathcal{N}(\mathbf{x}|\mu, \Sigma)$, where $\theta = \{\mu, \Sigma\}$. With this choice, OT has closed form solution (see example 2), but the data may not follow an actual Gaussian distribution. A more flexible model consists on assuming that data follows a *mixture of Gaussian measures*, for which optimal transport has nice properties as well [17].

Definition 9. *(Gaussian Mixture Model) Let $K \geq 1$ be an integer. A Gaussian Mixture Model (GMM) of size K on $\mathcal{X} = \mathbb{R}^d$ is a probability measure $P \in \mathbb{P}(\mathcal{X})$ such that,*

$$P = \sum_{k=1}^{K} p_k \mathcal{N}(\mu_k, \Sigma_k), \tag{2.16}$$

where $\Sigma_k \in \mathcal{S}^d$, and $\mathbf{p} \in \Delta_K$.

In the following, we provide a high-level overview of *how to do optimal transport on a computer*, based on the two approaches presented above, i.e., using empirical distributions or GMMs. We call the former *empirical optimal transport*, and the latter *Gaussian mixture optimal transport*.

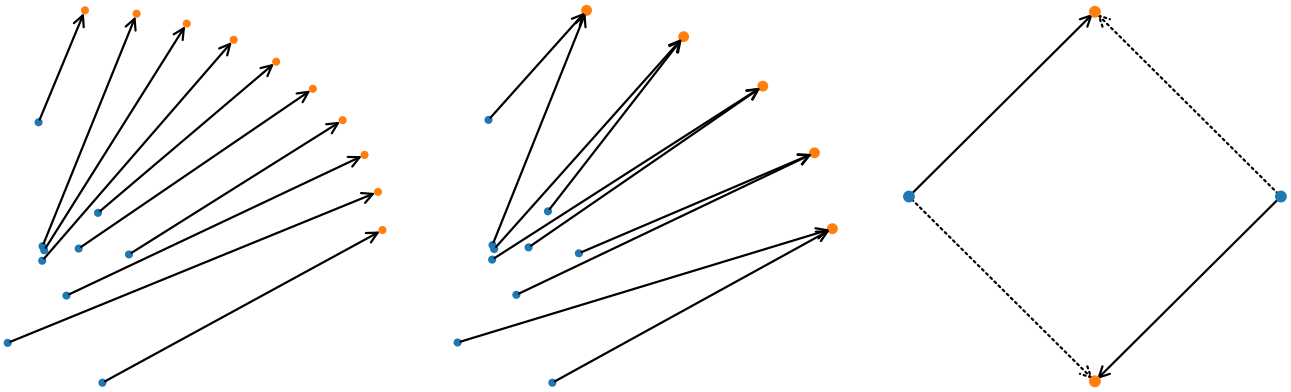
2.2.1 Empirical Optimal Transport

Let $\{x_i^{(P)}\}_{i=1}^n$ (resp. $\{x_j^{(Q)}\}_{j=1}^m$) sampled from P (resp. Q) with probability p_i (resp. q_j). The Monge problem seeks a mapping T, that is the solution of,

$$T^\star = \underset{T_\sharp \hat{P} = \hat{Q}}{\operatorname{argmin}} \sum_{i=1}^{n} c(\mathbf{x}_i^{(P)}, T(\mathbf{x}_i^{(P)})), \tag{2.17}$$

where the constraint implies $\sum_{i \in \mathcal{I}_j} p_i = q_j$, for $\mathcal{I}_j = \{i : x_j^{(Q)} = T(x_i^{(P)})\}$. In other words, one can view T as an assignment between indices $i = 1, \cdots, n$ and $j = 1, \cdots, m$, for which each i will be assigned to a single j. This illustrates an important feature of the Monge formulation: *the map T cannot split mass*. It is immediate to see that, when points have uniform mass, i.e., $p_i = n^{-1}$ and $q_j = m^{-1}$, the problem has no solution for $n < m$, since $k = m - n > 0$ points in Q will be left out in the transportation.

In Figure 2.6, we show three examples of Monge maps between point clouds with uniform mass (i.e., $p_i = n^{-1}$). Note that, in the discrete setting, one has an assignment between (i, j), rather than a function over all $\mathbb{R}^d$. This restricts the domain of definition of T to the points in the support of $\hat{P}$. As previously highlighted, such an assignment only exists for $n \geq m$, as shown in figures 2.6 (a) and (b). Furthermore, there may be more than one optimal assignment, as shown in Figure 2.6 (c).



(a) $n = m$. (b) $n > m$. (c) Non-unique optimal transport solution.

Figure 2.6 – **Three examples of Monge maps between point clouds with uniform mass**, which illustrate the fact that, under uniform masses, T exists only for $n \geq m$. In (c), we demonstrate the case where multiple optimal T may exist.

Conversely, the Kantorovich formulation seeks an OT *plan* $\gamma \in \mathbb{R}^{n \times m}$, where γ_{ij} denotes the amount of mass transported from sample i to sample j. In this case γ must minimize,

$$\hat{\gamma} = \underset{\gamma \in \Gamma(\mathbf{p}, \mathbf{q})}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_{ij} c(\mathbf{x}_i^{(P)}, \mathbf{x}_j^{(Q)}), \tag{2.18}$$

where $\Gamma(\mathbf{p}, \mathbf{q}) = \{\gamma \in \mathbb{R}^{n \times m} : \sum_i \gamma_{ij} = q_j \text{ and } \sum_j \gamma_{ij} = p_i\}$. As the shipment problem, equation 2.18 is a linear program on $n \times m$ variables, and $n + m - 1$ constraints. As a consequence [10, Proposition 3.4], $\hat{\gamma}$ has at most $n+m-1$ positive entries, i.e., it is a sparse matrix. Furthermore, optimal transport has complexity $\mathcal{O}(n^3 \log n)$, which becomes prohibitive for large scale distributions, i.e., with respect to the number of samples or bins n.

Remark 1. *When referring to complexities, we adopt O-notation, which characterizes upper bounds. Following [43], if $f(n) = \mathcal{O}(g(n))$, then there are constants c and n_0 such that $0 \leq f(n) \leq cg(n)$, for all $n \geq n_0$.*

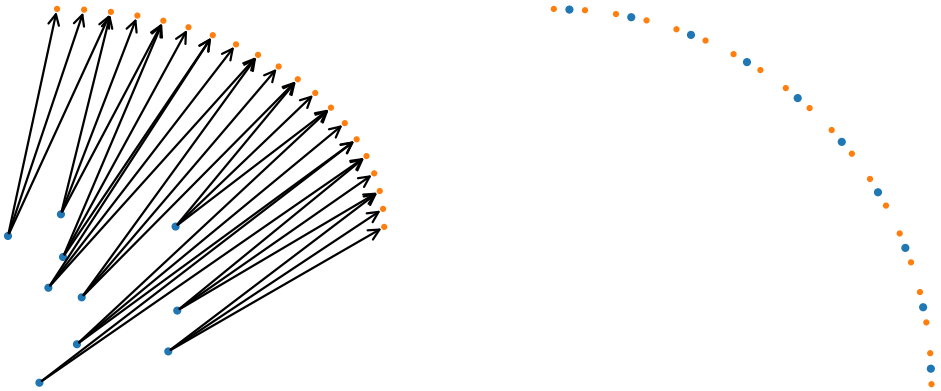
As we stated previously, the Kantorovich formulation offers more flexibility, and indeed the optimal transport problem is defined for $n < m$, as illustrated below. This flexibility comes with a cost: one loses the notion of map between P and Q. It is possible, however, to retrieve this notion, through the so-called *barycentric map*,

$$T_\gamma(\mathbf{x}_i^{(P)}) = \underset{\mathbf{x} \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{j=1}^{m} \gamma_{ij} c(\mathbf{x}, \mathbf{x}_j^{(Q)}). \tag{2.19}$$

which coincides with the Monge map whenever it exists (see [44] and [45] for a more detailed description of the equivalence). Note that, for the squared Euclidean distance, T_γ has a closed-form solution,

$$T_\gamma(\mathbf{x}_i^{(P)}) = \sum_{j=1}^{m} \frac{\gamma_{ij}}{p_i} \mathbf{x}_j^{(Q)}$$

In the discrete setting, similarly to the Monge map, the barycentric map is only defined on the support of $\hat{P}$. In Figure 2.7a, we show the optimal transport plan for $n = 10 < m = 20$, which shows that the mass from the source points is split across multiple target points. Likewise, in Figure 2.7b, we show the associated barycentric map.



(a) $n < m$. (b) Barycentric map.

Figure 2.7 – Kantorovich formulation of optimal transport. In (a), optimal transport plan from source points (blue) towards target points (orange). In (b), corresponding transported points (blue) through the barycentric mapping T_γ.

From convex analysis and linear programming theory, the Kantorovich problem in equation 2.18 is identified as a primal formulation of a linear optimization problem. As such, one has a dual transportation problem in terms of variables $(\mathbf{f}, \mathbf{g})$, with $\mathbf{f} \in \mathbb{R}^n$ and $\mathbf{g} \in \mathbb{R}^m$,

$$(\hat{\mathbf{f}}, \hat{\mathbf{g}}) = \underset{(\mathbf{f},\mathbf{g})\in\Phi(\mathbf{C})}{\operatorname{argmax}} \ \mathbf{f}^T\mathbf{p} + \mathbf{g}^T\mathbf{q}, \tag{2.20}$$

where $\Phi(\mathbf{C}) = \{(\mathbf{f}, \mathbf{g}) \in \mathbb{R}^{n\times m} : f_i + g_j \leq C_{ij}\}$. Equation 2.20 is a (dual) linear program on $n + m - 1$ variables, and $n \times m$ constraints. More importantly, the variables $(\mathbf{f}, \mathbf{g})$ play the role of Kantorovich potentials (ϕ, ψ) in equation 2.8.

Wasserstein Distance. As we discussed so far, approximating probability distributions through empirical ones leads to a discrete optimal transport problem, solvable by a computer when samples are available. Given these concepts, one may devise a discrete definition for the Wasserstein distance, based on the problems in equations 2.17, 2.18 and 2.20. For instance, using the Kantorovich formulation,

$$\mathcal{W}_\alpha(\hat{P}, \hat{Q})^\alpha = \min_{\gamma\in\Gamma(\mathbf{p},\mathbf{q})} \sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{ij} \|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2^\alpha, \tag{2.21}$$

where $\hat{P}$ and $\hat{Q}$ highlight the fact that the Wasserstein distance is taken between empirical distributions.

Remark 2. *(Computational Complexity) The complexity of computing the Wasserstein distance (c.f., equation 2.21) boils down to computing the OT γ, using equation 2.18. As a result, estimating the Wasserstein distance involves a $\mathcal{O}(n^3 \log n)$ complexity.*

An alternative to this strategy consists of assuming P and Q are a Gaussian measures. In this case, one can estimate $(\hat{\mu}^{(P)}, \hat{\Sigma}^{(P)}, \hat{\mu}^{(Q)}, \hat{\Sigma}^{(Q)})$ from samples, then compute $\mathcal{W}_2$ using equation 2.12. Here, the complexity boils down to the calculation of square root matrices, which rely no the diagonalization of the covariances. As a result, taking this route yields a $\mathcal{O}(d^3)$ complexity.

The bottom line is a trade-off. On one hand, using a non-parametric approximation for P and Q involves a sample-dependent *complexity, which grows with the number of samples. On the other hand, using a Gaussian, parametric approximation for P and Q involves a* dimension-dependent *complexity, plus possible modelling errors involved with the Gaussian assumption. We summarize these ideas in Figure 2.8.*

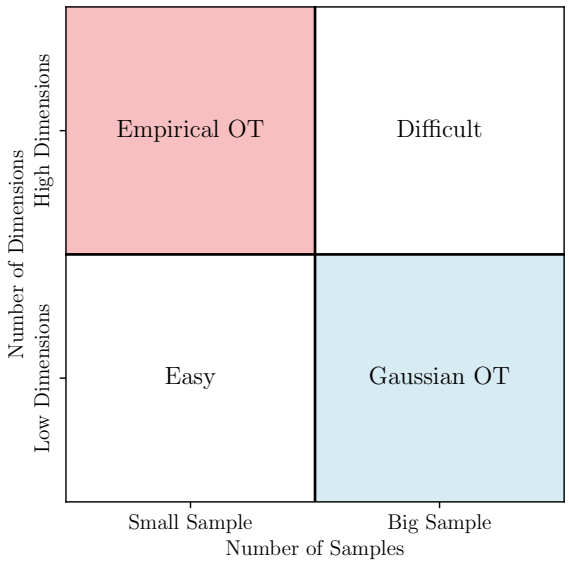


Figure 2.8 – **Computational complexity trade-off in OT**. One should note that this illustration does not take into account the statistical difficulty of performing OT, which we cover below. When n is small and d is large, performing empirical OT becomes computationally cheaper than Gaussian OT. Conversely, in low dimensions and big samples, using Gaussian OT becomes more efficient. Furthermore, on the one hand, the small sample - low dimensions regime is computationally easy to solve. On the other hand, the big sample - high dimensions case is computationally hard. We discuss later on strategies for the later case.

From the point of view of statistics, $\mathcal{W}_\alpha(\hat{P}, \hat{Q})$ is a random variable, depending on the choice of samples from P and Q respectively. This begs the question of *how well* $\mathcal{W}_\alpha(\hat{P}, \hat{Q})$ *approximates* $\mathcal{W}_\alpha(P, Q)$*?* To start an answer to this question, we use [46, Theorem 1.1],

Theorem 1. *Let P be a probability distribution over $\mathbb{R}^d$, so that for some $\alpha > 0$ we have that $\int_{\mathbb{R}^d} e^{\alpha||\mathbf{x}||^2} dP < \infty$ and $\hat{P}$ be its associated empirical approximation with support $\{\mathbf{x}_i^{(P)}\}_{i=1}^n$ drawn independently from P. Then, for any $d' > d$ and $\xi' < \sqrt{2}$ there is a constant n_0 depending on d' and some square exponential moment of P such that for any $\epsilon > 0$ and $n \geq n_0 max(\epsilon^{-(d'+2)}, 1)$,*

$$\mathbb{P}[\mathcal{W}_1(\hat{P}, P) > \epsilon] \leq \exp\Big(-\frac{\xi'}{2} n\epsilon^2 \Big),$$

where d' and ξ' can be calculated explicitly.

We can use this result to get a bound between $\mathcal{W}_1(P, Q)$ and its empirical estimator, i.e., $\mathcal{W}_1(\hat{P}, \hat{Q})$. We formalize this in the next corollary,

Corollary 1. *Let P be a probability distribution over $\mathbb{R}^d$, so that for some $\alpha > 0$ we have that $\int_{\mathbb{R}^d} e^{\alpha||\mathbf{x}||^2} dP < \infty$ and $\hat{P}$ be its associated empirical approximation with support $\{\mathbf{x}_i^{(P)}\}_{i=1}^n$ drawn independently from P. Then, for any $d' > d$ and $\xi' < \sqrt{2}$ there is a constant n_0 depending on d' and some square exponential moment of P such that for any $\epsilon > 0$ and $n \geq n_0 max(\epsilon^{-(d'+2)}, 1)$,*

$$|\mathcal{W}_1(P, Q) - \mathcal{W}_1(\hat{P}, \hat{Q})| \leq \underbrace{\sqrt{\frac{2 \log 1/\delta}{\xi'}} \Big(\sqrt{\frac{1}{n}} + \sqrt{\frac{1}{m}} \Big)}_{\mathcal{C}_{OT}(n,\delta)} \tag{2.22}$$

where d' and ξ' can be calculated explicitly, and $\mathcal{C}(n, \delta)$ denotes the statistical complexity *of optimal transport. Strictly speaking, $\mathcal{C}$ depends on the number of samples on P and Q. In the spirit of alleviating the notation, we denote $\mathcal{C}(n, \delta)$, for $n = max(n, m)$, where n is the number of samples from P (resp. m for Q).*

Henceforth, inequalities like the one in 2.22 will constitute our theoretical results, especially those in domain adaptation. These inequalities reflect an error in the empirical estimation of a quantity, in this case, the 1−Wasserstein distance. We loosely refer to the rate of convergence on the right hand side as *statistical complexity*, in analogy to computational complexity. We understand this term as the rate of decay (towards zero), as a function of number of samples. Indeed, assuming without loss of generality $n \geq m$, as $n \to \infty$ one has $|\mathcal{W}_1(P, Q) - \mathcal{W}_1(\hat{P}, \hat{Q})| \to 0$. A simple consequence of equation 2.22 is that estimating $\mathcal{W}_1$ has $\mathcal{O}(n^{-1/2})$ complexity.

While the discrete optimal transport problem is useful for mapping, and calculating distances between probability distributions, it is computationally complex. Furthermore, the fact that it is a linear program makes its implementation on a GPU complicated. Next, we discuss two alternatives for alleviating the cost of optimal transport: (i) adding entropic regularization and (ii) computing optimal transport between mini-batches.

Entropic Regularization. In [47], the authors proposed to solve a regularized optimal transport problem, by adding an entropy term to the Kantorovich formulation in equation 2.18,

$$\hat{\gamma}_\epsilon = \underset{\gamma \in \Gamma(\mathbf{p}, \mathbf{q})}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_{ij} c(\mathbf{x}_i^{(P)}, \mathbf{x}_j^{(Q)}) + \epsilon \underbrace{\Big(-\sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_{ij} (\log \gamma_{ij} - 1) \Big)}_{H(\gamma)}. \tag{2.23}$$

As a result, one can use the Lagrangian of the constrained optimization problem,

$$L(\gamma, \mathbf{f}, \mathbf{g}) = \langle \gamma, \mathbf{C} \rangle_F + \epsilon H(\gamma) - \mathbf{f}^T(\gamma \mathbf{1}_m - \mathbf{p}) - \mathbf{g}^T(\gamma^T \mathbf{1}_n - \mathbf{q})$$

which imply a closed form for $\hat{\gamma}_{\epsilon,i,j}$,

$$\hat{\gamma}_{\epsilon,i,j} = \exp\left(- \frac{C_{ij} - f_i - g_j}{\epsilon} \right), \text{ or, } \hat{\gamma}_\epsilon = \text{diag}(\mathbf{u}) \mathbf{K} \text{diag}(\mathbf{v}) \tag{2.24}$$

where $\mathbf{u} \in \mathbb{R}^n$, $\mathbf{v} \in \mathbb{R}^m$ and $\mathbf{K} \in \mathbb{R}^{n \times m}$, with $u_i = e^{f_i/\epsilon}$, $v_j = e^{g_j/\epsilon}$ and $K_{ij} = e^{-C_{ij}/\epsilon}$. Equation 2.24 can be solved for $\mathbf{u}$ and $\mathbf{v}$ using the celebrated Sinkhorn-Knopp matrix scaling algorithm [48]. Starting from an initialization (e.g. $u_i = 1/n$ and $v_j = 1/m$), this algorithm iteratively updates the marginals using,

$$\mathbf{u}^{(\ell+1)} = \frac{\mathbf{p}}{\mathbf{K}\mathbf{v}^{(\ell)}} \text{ and } \mathbf{v}^{(\ell+1)} = \frac{\mathbf{q}}{\mathbf{K}^T \mathbf{u}^{(\ell+1)}}, \tag{2.25}$$

which leads to a straightforward algorithm involving matrix multiplications and element-wise divisions. However, note that these iterations involve exponential terms (i.e., $K_{ij} = \exp(-C_{ij}/\epsilon)$), which may suffer from underflow given the values of C_{ij} and ϵ, especially for small values of ϵ. Based on this, [49] proposed *stabilized iterations* in place of equation 2.25,

$$\mathbf{f}^{(\ell+1)} = \epsilon \log \mathbf{p} - \epsilon \log \mathbf{K} e^{\mathbf{g}^{(\ell)}/\epsilon}, \text{ and } \mathbf{g}^{(\ell+1)} = \epsilon \log \mathbf{q} - \epsilon \log \mathbf{K}^T e^{\mathbf{f}^{(\ell+1)}/\epsilon},$$

Remark 3. *By rearranging terms in equation 2.23, one can express the entropic OT problem as,*

$$\hat{\gamma}_\epsilon = \underset{\gamma \in \Gamma(\mathbf{p},\mathbf{q})}{\text{argmin}} \, KL(\gamma | \mathbf{K}), \tag{2.26}$$

where KL is the Kullback-Leibler divergence (c.f., chapter 3). As a result one may see entropic OT as a projection process with respect the KL divergence. Furthermore, the Sinkhorn iterations can be understood as projections on the sets $\mathcal{C}_r$ and $\mathcal{C}_c$ of row and column normalized arrays. These ideas are described in [50], and will be useful in chapter 3.

Sinkhorn Divergences. Solving OT with finite samples provides an empirical estimator for $\mathcal{T}_c$ and W_p, i.e., $\mathcal{T}_c(\hat{P}, \hat{Q}) = \mathcal{L}_K(\hat{\gamma})$. Likewise, for $\gamma_\epsilon^\star$ one has $\mathcal{T}_{c,\epsilon}(\hat{P}, \hat{Q}) = \mathcal{L}_K(\gamma_\epsilon^\star)$. This approximation motivates the Sinkhorn divergence [51],

$$\mathcal{S}_{\alpha,\epsilon}(\hat{P}, \hat{Q}) = 2\mathcal{W}_{\alpha,\epsilon}(\hat{P}, \hat{Q}) - \left(\mathcal{W}_{\alpha,\epsilon}(\hat{P}, \hat{P}) + \mathcal{W}_{\alpha,\epsilon}(\hat{Q}, \hat{Q}) \right), \tag{2.27}$$

which has interesting properties, such as interpolating between the Maximum Mean Discrepancy (MMD) of [52] and the Wasserstein distance. Overall, entropic OT has two computational advantages w.r.t exact OT. Indeed, its calculations are GPU-friendly, and for $L \geq 1$ iterations, its complexity is $\mathcal{O}(Ln^2)$. In addition, $\mathcal{S}_{\alpha,\epsilon}$ is a smooth approximator of $\mathcal{W}_\alpha$ [53], and it enjoys better sample complexity [54]. Interestingly, the sample complexity of $\mathcal{S}_{\alpha,\epsilon}$ depends on the regularization parameter ϵ. We refer readers to [143] for further details.

Algorithm 1: Sinkhorn algorithm

```
function sinkhorn(p, q, C, ε)
    // Initialization.
    f^(0) ← 1_n/n
    g^(0) ← 1_m/m
    // Updates.
    for it = 1, ··· , N_iter do
        f^(it+1) = ε log p − ε log( K exp(g^(it)/ε) )
        g^(it+1) = ε log p − ε log( K^T exp(f^(it+1)/ε) )
    return f, g, γ_ε = diag(exp(f/ε))K diag(exp(g/ε))
```

Mini-batch Optimal Transport. A major challenge in OT is its time complexity. This motivated different authors [55, 51, 56] to compute the Wasserstein distance between mini-batches rather than complete datasets. Note that this is an effective solution to the *high dimensions – big sample* situation.

For a dataset with n samples, this strategy leads to a dramatic speed-up, since for K mini-batches of size $m \ll n$, one reduces the time complexity of OT from $\mathcal{O}(n^3 \log n)$ to $\mathcal{O}(Km^3 \log m)$ [56]. This choice is key when using OT as a loss in learning [57] and inference [58]. Henceforth we describe the mini-batch framework of [59], for using OT as a loss. Let $\mathcal{L}_{OT}$ denote an OT loss (e.g. $\mathcal{W}_\alpha$ or $\mathcal{S}_{\alpha,\epsilon}$). Assuming continuous distributions P and Q, the mini-batch optimal transport loss is given by,

$$\mathcal{L}_{\text{MBOT}}(P,Q) = \mathop{\mathbb{E}}_{(\mathbf{X}^{(P)},\mathbf{X}^{(Q)})\sim P^{\otimes m}\otimes Q^{\otimes m}}[\mathcal{L}_{OT}(\mathbf{X}^{(P)},\mathbf{X}^{(Q)})],$$

where $\mathbf{X}^{(P)} \sim P^{\otimes m}$ indicates $\mathbf{x}_i^{(P)} \sim P$, $i = 1, \cdots, m$. This loss inherits some properties from OT, i.e., it is positive and symmetric, but $\mathcal{L}_{\text{MBOT}}(P,P) > 0$, since different mini-batches may be sampled from the same measure.

In practice, let $\{\mathbf{x}_i^{(P)}\}_{i=1}^{n_P}$ and $\{\mathbf{x}_j^{(Q)}\}_{i=1}^{n_Q}$ be iid samples from P and Q respectively. Let $\mathcal{I}_m \subset \{1, \cdots, n_P\}^m$ denote a set of m indices. We denote by $\hat{P}_{\mathcal{I}_m}$ to the empirical approximation of P with iid samples $\mathbf{X}^{(P)} = \{\mathbf{x}_i^{(P)} : i \in \mathcal{I}_m\}$. Therefore,

$$\mathcal{L}_{\text{MBOT}}^{(k,m)}(\hat{P},\hat{Q}) = \frac{1}{k}\sum_{(\mathcal{I}_b,\mathcal{I}_b')\in\mathbb{I}_k} \mathcal{L}_{OT}(\hat{P}_{\mathcal{I}_b},\hat{Q}_{\mathcal{I}_b'}), \tag{2.28}$$

where $\mathbb{I}_k$ is a random set of k mini-batches of size m from P and Q. This constitutes an estimator for $\mathcal{L}_{MBOT}(P,Q)$, which converges as n and $k \to \infty$. We highlight 3 advantages that favor the mini-batch optimal transport for machine learning: (i) it is faster to compute and computationally scalable; (ii) the deviation bound between $\mathcal{L}_{\text{MBOT}}(P,Q)$ and $\mathcal{L}_{\text{MBOT}}^{(k,m)}$ does not depend on the dimensionality of the space; (iii) it has unbiased gradients, i.e., the expected gradient of the sample loss equals the gradient of the true loss [60]. Nonetheless, mini-batch OT brings new challenges. As [61] studies, the use of

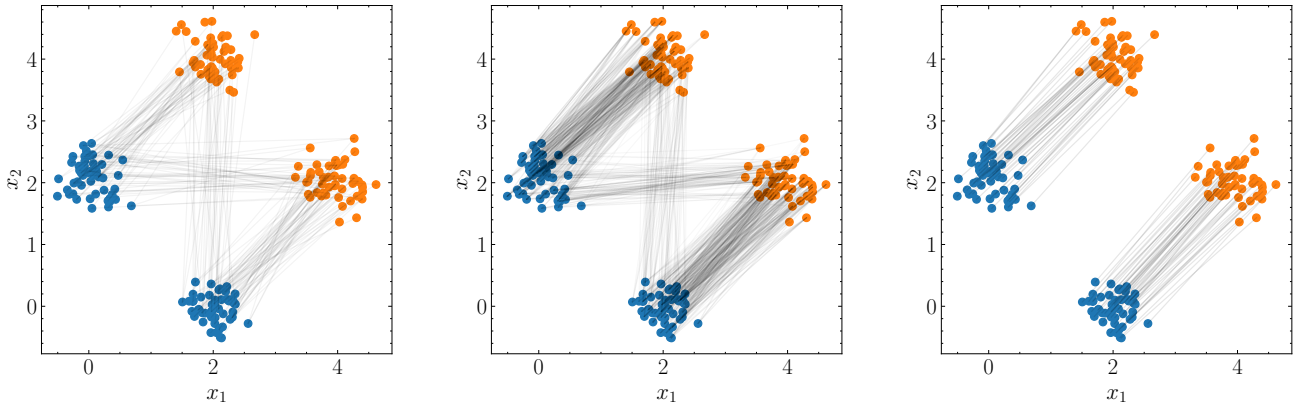


Figure 2.9 – Mini-batch OT between distributions P (in blue) and Q (in orange). As follows, an OT plan is calculated with mini-batches of 2 (a), 10 (b) and 100 (c) samples. (c) corresponds to the original OT problem. Overall, in mini-batch OT the plans become less sparse, due to OT being forced to transport all mass between mini-batches.

mini-batches introduces artifacts in OT plans, as they become less sparse. This issue is shown in Figure 2.9, which shows the OT plan in mini-batch OT.

Unbalanced OT is an extension to the original Kantorovich problem, which relaxes the mass conservation constraint [59]. The idea, as discussed in [62], is to replace the hard constraint $\gamma \in \Gamma(\mathbf{p}, \mathbf{q})$ by soft constraints in terms of a f-divergence (cf., eq. 3.1),

$$\hat{\gamma}_{\epsilon,\tau} = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{ij} c(\mathbf{x}_i^{(P)}, \mathbf{x}_j^{(Q)}) + \epsilon H(\gamma) + \tau(D_f(\gamma_1|\mathbf{p}) + D_f(\gamma_2|\mathbf{q})). \tag{2.29}$$

where $\gamma_1 = \sum_j \gamma_{ij}$ and $\gamma_2 = \sum_i \gamma_{ij}$. This alternative problem can be solved by easily adapting Algorithm 1. Indeed, as [63] shows, it suffices to show the iterations using $\eta = \epsilon/(\epsilon+\tau)$ in algorithm 1.

In analogy with the Sinkhorn divergence, unbalanced OT defines a divergence $\mathcal{S}_{\epsilon,\tau}$ as well. This extension has a few advantages. First, it can be easily implemented on top of the Sinkhorn algorithm [63]. Second, it is well defined for positive vectors $\mathbf{p} \in \mathbb{R}_+^n$, $\mathbf{q} \in \mathbb{R}_+^m$. Third, it is robust to outliers [61], which favors its application to mini-batch OT.

Partial OT defines an OT problem in which the transportation plan transports only a fraction, $0 \leq s \leq 1$, of the total mass. This defines a new set,

$$\Gamma_s(\mathbf{p}, \mathbf{q}) = \{\gamma : \sum_i \gamma_{ij} \leq q_j, \sum_j \gamma_{ij} \leq p_i, \sum_{i,j} \gamma_{ij} = s\},$$

which substitutes Γ in eq. 2.18. As [64] proposes, partial OT can be solved by adding dummy sink points to which the mass that is not transported, $1 - s$, will be sent to. Similarly to unbalanced OT, the partial extension is used to enhance mini-batch OT [65].

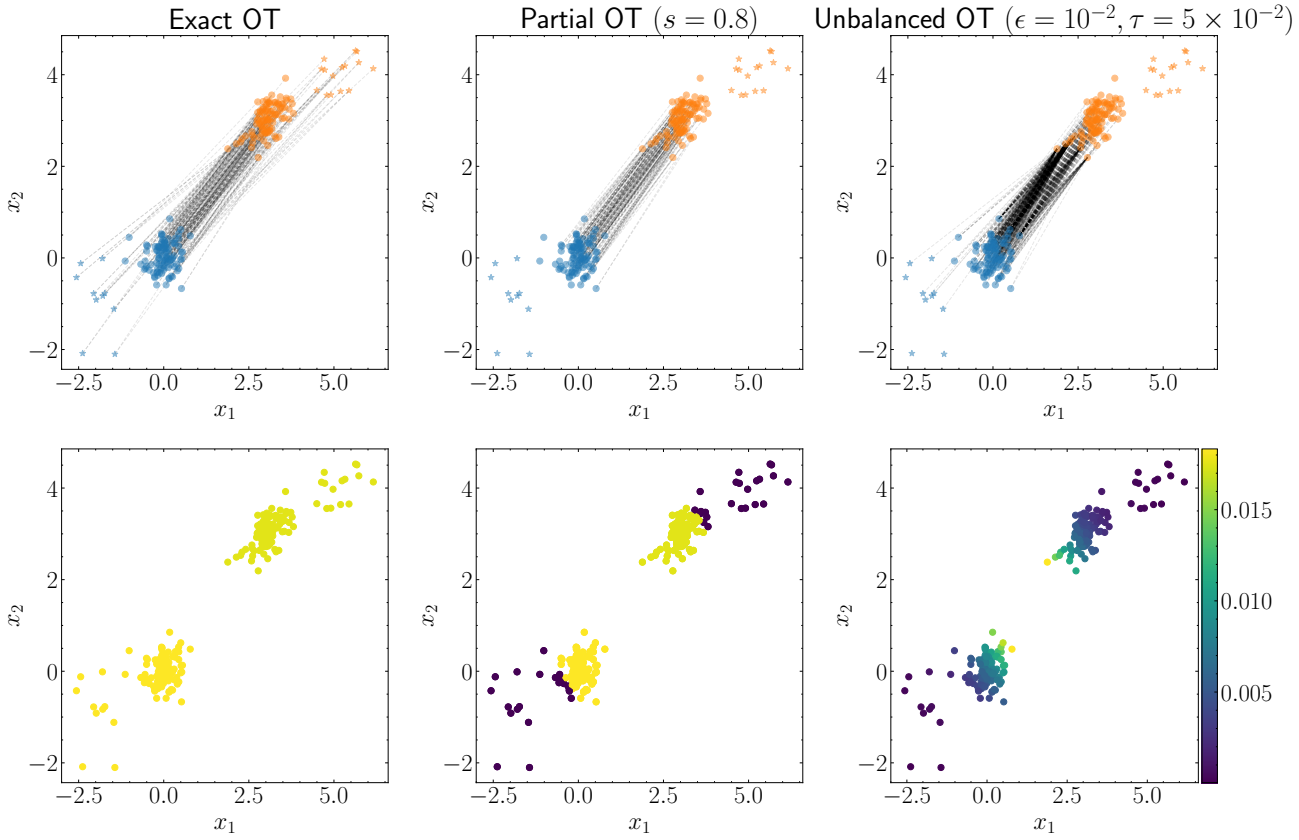


Figure 2.10 – Comparison between exact, partial and unbalanced OT. On top, we visualize the OT plans as lines joining points transporting mass. On bottom, we color samples by how much mass they send (source distribution) and receive (target distribution), which shows that most outliers in the distributions do not participate in partial nor unbalanced OT. This phenomenon highlights the advantage of these extensions for handling datasets with outliers.

2.2.2 Gaussian Mixture Optimal Transport

This section is primarily based on [66, 17]. Our goal is to present the *Gaussian Mixture Model-based Optimal Transport (GMM-OT) machinery* developed by the authors. This methodology will play a prominent role in chapter 7.

As we explored in the initial sections, when P and Q are Gaussian distributions, optimal transport has a closed-form, linear solution. This leads to efficient, large scale robust algorithms for estimating optimal transport [67], under the hypothesis that data is approximately Gaussian. Nonetheless, this is seldom the case. Especially, in the context of this thesis, data is rather supposed to be multi-modal, in which each mode represents a cluster or a class of similar examples.

As follows, one may relax the Gaussian hypothesis towards mixtures of Gaussian distributions. This relaxation gives more flexibility to the probabilistic model, while, as shown by [17], retaining the desirable feature of having closed-form solutions. However, even if P and Q are Gaussian mixtures, the optimal transport plan $\gamma \in \Gamma(P, Q)$ [17, Proposition 3], which imply that barycenters of P and Q

are not Gaussian mixtures either.

To solve the aforementioned issues, [17] proposes a Wasserstein-like distance between Gaussian mixtures, which restricts the set of admissible transport plans,

Definition 10. *(Mixture Wasserstein Distance) Let $P \in GMM_d(K_1)$ and $Q \in GMM_d(K_2)$ be two Gaussian mixtures. The Mixture-Wasserstein distance is given by,*

$$\mathcal{MW}_\alpha(P,Q)^\alpha = \min_{\gamma \in \Gamma(P,Q) \cap GMM_{2d}(\infty)} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \|\mathbf{x}-\mathbf{z}\|_2^\alpha d\gamma(\mathbf{x},\mathbf{z}) \tag{2.30}$$

This distance has many desirable properties. First, it is well-defined, as the trivial coupling $\gamma = P \otimes Q$ belongs to $\Gamma(P,Q) \cap \mathrm{GMM}_{2d}(\infty)$. This implies that the minimum in equation 2.30 exists. Furthermore, via [17, Proposition 4], it admits an equivalent discrete formulation,

$$\mathcal{MW}_\alpha(P,Q)^\alpha = \inf_{\omega \in \Gamma(\mathbf{p},\mathbf{q})} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \omega_{k_1,k_2} \mathcal{W}_\alpha(P_{k_1}, Q_{k_2})^\alpha, \tag{2.31}$$

where $P_{k_1} = \mathcal{N}(\mu_{k_1}^{(P)}, \Sigma_{k_1}^{(P)})$ and $Q_{k_2} = \mathcal{N}(\mu_{k_2}^{(Q)}, \Sigma_{k_2}^{(Q)})$ are the k_1-th and k_2-th components of mixtures P and Q, respectively. Equation 2.31 defines the $\mathcal{MW}$ distance as an hierarchical optimal transport problem, i.e., an optimal transport problem defined at two levels. On the inner part of equation 2.31, one needs to compute the Wasserstein distance between components P_{k_1} and Q_{k_2}. Based on all pairwise distances, one solves an outer optimal transport problem via linear programming.

Remark 4. *One should note that the optimization problem in equations 2.30 and 2.31 is defined for different variables. In the continuous case (equation 2.30), one optimizes with respect a transport plan* between samples, $\gamma(\mathbf{x},\mathbf{z})$. *This transport plan is further required to be a GMM. In the discrete case (equation 2.31), one optimizes with respect a transport plan $\omega \in \mathbb{R}^{K_P \times K_Q}$ between* components. *These two objects are linked through the density of γ, i.e.,*

$$\gamma(\mathbf{x}_1,\mathbf{x}_2) = \sum_{k_1=1}^{K_P} \sum_{k_2=1}^{K_Q} \omega_{k_1,k_2} \mathcal{N}(\mathbf{x}_1 | \mu_{k_1}^{(P)}, \Sigma_{k_1}^{(P)}) \delta(\mathbf{x}_2 - T_{k_1,k_2}(\mathbf{x}_1)). \tag{2.32}$$

Next, we present [17, Corollary 2] as a proposition. This result is a consequence of the equivalence between equations 2.30 and 2.31, i.e., [17, Proposition 4].

Proposition 1. *(Interpolation of Gaussian mixtures) Let $P \in GMM_d(K_1)$ and $Q \in GMM_d(K_2)$ be two Gaussian mixtures, and $\omega^\star$ be the solution of equation 2.31. For $t \in [0,1]$, the interpolation P_t between P and Q is given by,*

$$P_t = \pi_{t,\sharp}\gamma^\star = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \omega_{k_1,k_2}^\star P_{t,k_1,k_2},$$

where $P_{t,k_1,k_2} = ((1-t)Id + tT_{k_1,k_2})_\sharp P_{k_1}$ is the distribution obtained by interpolating between components P_{k_1} and Q_{k_2} of mixtures P and Q.

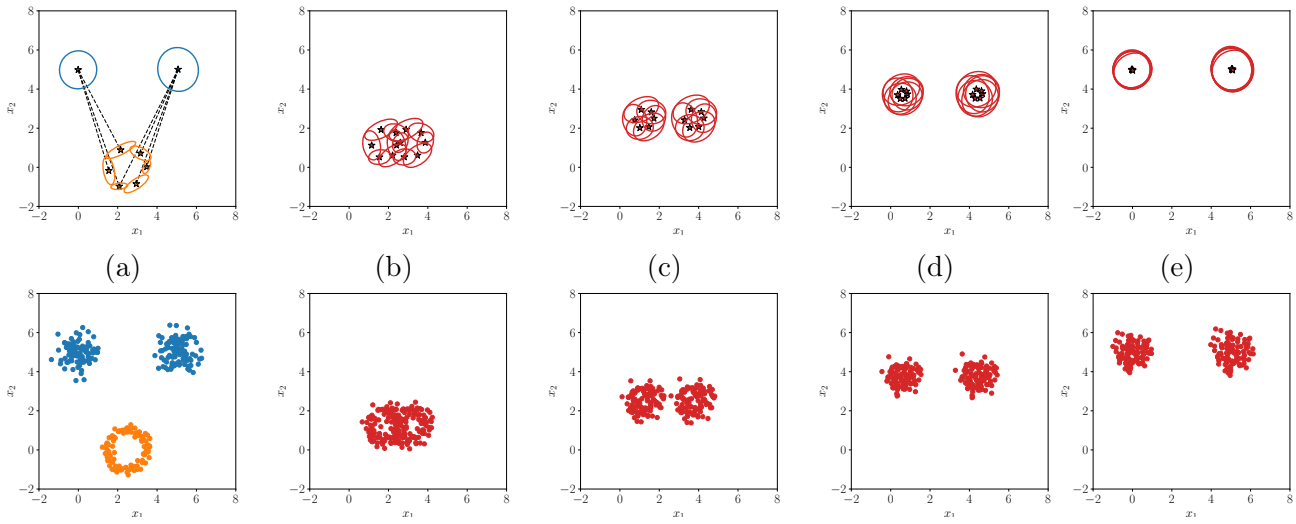


Figure 2.11 – Illustration of Gaussian mixture optimal transport. In (a), we show the optimal transport plan between components of P (orange) and Q (blue). In (f), we show samples drawn from P and Q. From (b) to (e), we show the components of the interpolated $P_{t,k_1,k_2} = ((1-t)Id+tT_{k_1,k_2})_\sharp P_{k_1}$. From (g) to (j) we show the action of the map $T_{t,k_1,k_2} = (1-t)Id+tT_{k_1,k_2}$ on samples from P.

An immediate consequence of this proposition is that one may retrieve the optimal transport map between P and Q by setting $t := 1$. As a consequence, the optimal transport map T between P and Q is a piece-wise linear map, i.e., it is linear inside each component P_{k_1} of P. In Figure 2.11 we show an illustration of the Gaussian Mixture transport plan ω between P and Q, as well as the interpolation P_t between P and Q.

2.3 Conclusion

In this chapter, we presented a pragmactic view of OT. We focus our exposition on Euclidean spaces, which is the usual space that Domain Adaptation (DA) is carried out. A major theme in this thesis is *computational optimal transport* [10], which assumes some *finite encoding* of probability measures. In this thesis we cover three strategies. First, one can directly use samples from probability measures, leading to empirical approximations. The second and third options rely on parametric models. For instance, one can assume that probability measures are Gaussians. In this case, as we shown in Example 2, OT has a closed form solution in the form of a Monge map. Furthermore, the Wasserstein distance can be readily estimated from the means and covariances, as shown in Example 3. A third strategy is to assume that probability measures are Gaussian mixture models. In this case, as [17] shows, restricting the OT plan to be a GMM itself leads to great simplifications. Indeed, as we discuss in Section 2.2.2, the continuous OT problem between GMMs has an equivalent discrete, hierarchical OT problem between their components.

Overall, this chapter serves as the foundation of Part II. For instance, empirical OT is at the basis of Chapters 5 and 6, which rely on empirical measures. Furthermore, the Gaussian and GMM formalisms are the basis of our GMM-Optimal Transport for Domain Adaptation (OTDA), GMM-Wasserstein Barycenter Transport (WBT) and GMM-Dataset Dictionary Learning (DaDiL) approaches in Chapter 7.

Chapter 3

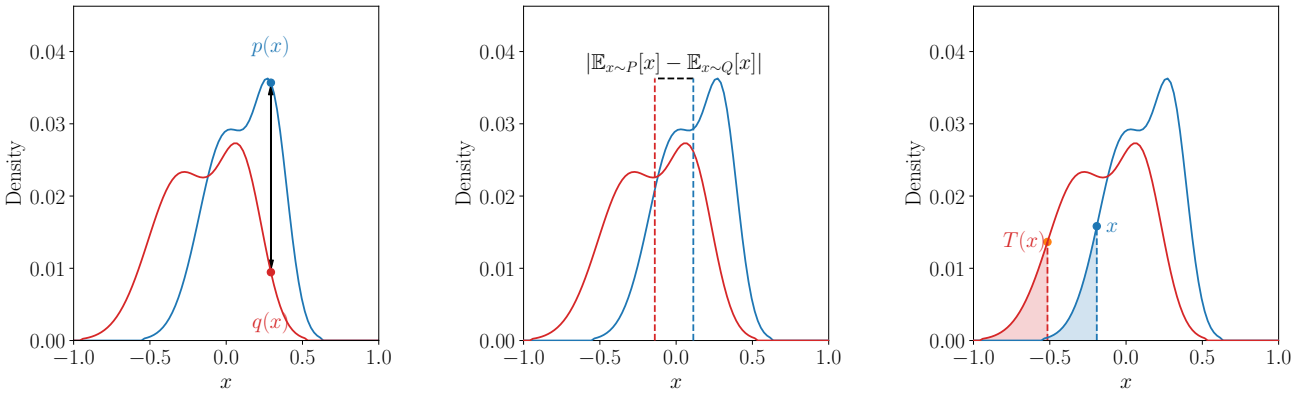
Barycenters of Probability Measures

Contents

In this chapter, we consider measures in the space of probability metrics. Note that, in the previous chapter, we introduced the notion of Wasserstein distance (c.f., Definition 2.11). We introduce two families of probability dissimilarities. First, $f-$divergences (see definition 11) are dissimilarities which are not symmetric nor suffice the triangle inequality. These are based on the ratio between the densities of P and Q. Second, Integral Probability Metrics (IPMs) (see definition 12) offer proper metrics on $\mathbb{P}(\mathcal{X})$. These are based on the average difference under P and Q over a family of functions.

The main goal of this section is to establish the notion of barycenters of probability measures. The main idea is to define these barycenters in analogy with barycenters in Euclidean geometry, i.e., a point that is equidistant to a set of points. We focus our attention on barycenters under two metrics: the MMD [52], and the Wasserstein distance.

The rest of this chapter is divided as follows. Section 3.1 defines $f-$divergences and integral probability metrics. Section 3.2 we discuss barycenters under metrics in $\mathbb{P}(\mathcal{X})$, with an emphasis on Wasserstein distances and multi-marginal OT. Finally, section 3.3 presents computational strategies for computing barycenters.



(a) $f-$divergences. (b) IPMs. (c) Optimal transport.

Figure 3.1 – **Conceptual comparison of families of divergences.** This figure is inspired by [19]. In (a), we show $f-$divergences, which work by computing the ratio between densities. In (b), we show IPMs, which work by calculating the difference beween mean embeddings. In (c), we show how optimal transport work, where x is transported to $T(x)$ such that the corresponding probability mass is preserved. As we cover in the following, under certain circumstances, *optimal transport is an IPM*.

3.1 Metrics and Divergences between Probability Measures

In this section, we present different notions of discrepancy between probability measures. These notions can be defined into two categories. First, one has proper distances, which suffice the usual conditions for metrics: (i) $\mathcal{D}(P,Q) = 0 \iff P = Q$, (ii) $\mathcal{D}(P,Q) \geq 0 \; \forall P, Q$, (iii) $\mathcal{D}(P,Q) = \mathcal{D}(Q,P)$, and (iv) $\mathcal{D}(P,Q) \leq \mathcal{D}(P,Q') + \mathcal{D}(Q',Q)$. Second, one has divergences, which come from information theory and for which (iii) and (iv) do not necessarily hold.

3.1.1 $f-$Divergences

We start our presentation with the notion of $f-$divergences [68], which evaluate the difference between two probability measures based on the ratio of their densities.

Definition 11. *($f-$divergence) Let $P, Q \in \mathbb{P}(\mathcal{X})$. For a convex, lower semi-continuous function $f : \mathbb{R}_+ \to \mathbb{R}$ with $f(1) = 0$,*

$$\mathcal{D}_f(P||Q) = \mathop{\mathbb{E}}_{x\sim Q}\left[f\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right)\right] = \int_{\mathcal{X}} f\left(\frac{p(x)}{q(x)}\right) q(x)dx, \tag{3.1}$$

where $\mathcal{D}_f = +\infty$ whenever $q(x) = 0$ for $p(x) > 0$.

Different divergences fall into the definition in equation 3.1, such as the Kullback Leibler (KL) divergence (for $f(t) = t\log t$), the Hellinger distance (for $f(t) = (\sqrt{t}-1)^2$) and the Total variation (for $f(t) = |t-1|$). An interesting remark is that the total variations is, at the same time, an $f-$divergence,

and an integral probability metric (see e.g. next section). In this thesis we are particularly interested on the KL divergence, which can be expressed as,

$$\mathrm{KL}(P||Q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} dx = \mathop{\mathbb{E}}_{x\sim P}\left[\log \frac{p(x)}{q(x)}\right] = \mathop{\mathbb{E}}_{x\sim P}[\log p(x)] - \mathop{\mathbb{E}}_{x\sim P}[\log q(x)]. \tag{3.2}$$

The KL divergence plays a prominent role in statistical inference, through the so-called maximum likelihood estimation. As a result, this divergence has an impact on various areas of machine learning, such as unsupervised learning and generative modeling. Next, we exemplify how it is used to learn GMMs.

Example 4. *(Expectation-Maximization) Let* $\{\mathbf{x}_i^{(P)}\}_{i=1}^n$ *be* n *i.i.d. samples from an unknown distribution* P*. We are interested in modeling these samples through a Gaussian mixture model (c.f., definition 9), that is,* $P_\theta = \sum_{k=1}^K \beta_k \mathcal{N}(\mu_k, \Sigma_k)$*. Here,* $\theta = \{(\beta_k, \mu_k, \Sigma_k)_{k=1}^K : \beta \in \Delta_K, \mu_k \in \mathbb{R}^d, \Sigma_k \in \mathcal{S}^d\}$*. We begin by introducing the likelihood function, which depends on* θ*,*

$$\mathcal{L}(\theta) = P(\mathbf{x}_1^{(P)}, \cdots, \mathbf{x}_n^{(P)}|\theta) = \prod_{i=1}^n P_\theta(\mathbf{x}_i^{(P)}).$$

One may interpret $\mathcal{L}(\theta)$ *as* how well θ *explains the data points* $\mathbf{x}_i^{(P)}$*. As such, in maximum likelihood, one wants to maximize* $\mathcal{L}$ *with respect* θ*. Equivalently, since* log *is a concave function, we may solve,*

$$\theta^\star = \operatorname*{argmax}_\theta \frac{1}{n} \log \mathcal{L}(\theta) = \operatorname*{argmax}_\theta \frac{1}{n} \sum_{i=1}^n \log P_\theta(\mathbf{x}_i^{(P)}). \tag{3.3}$$

As $n \to +\infty$*, this maximization problem is actually equivalent to minimizing* $KL(P||P_\theta)$*, since,*

$$\operatorname*{argmax}_\theta \frac{1}{n} \sum_{i=1}^n \log P_\theta(\mathbf{x}_i^{(P)}) \overset{n\to+\infty}{=} \operatorname*{argmax}_\theta \mathop{\mathbb{E}}_{\mathbf{x}\sim P}[\log P_\theta(\mathbf{x})] = \operatorname*{argmin}_\theta \mathop{\mathbb{E}}_{\mathbf{x}\sim P}[\log P(\mathbf{x})] - \mathop{\mathbb{E}}_{\mathbf{x}\sim P}[\log P_\theta(\mathbf{x})],$$

which, after some rearranging, is equivalent to $KL(P||P_\theta)$*. Back to equation 3.3, this maximization problem has no closed-form solution. However, an efficient procedure known as Expectation Maximization (EM) is capable of maximizing the log-likelihood [69]. The idea of the algorithm is to iteratively update the parameters of the GMM via two steps, called expectation, and maximization. Given* θ*, the expectation step consists of computing,*

$$G_{ik} = P(K = k|X = \mathbf{x}_i) = \frac{\beta_k P_k(\mathbf{x}_i)}{\sum_{k'} \beta_{k'} P_{k'}(\mathbf{x}_i)}, \tag{3.4}$$

also called reponsibility *of component* k *given sample* $\mathbf{x}_i$ *[70]. Indeed, the entries of* $G_{i,\cdot}$ *denote the probability that* $\mathbf{x}_i$ *comes from component* P_k*. Given* $\mathbf{G}$*, we can maximize the log-likelihood (equation 4) through the* maximization *step,*

$$\mu_k = \frac{1}{n_k} \sum_{i=1}^n G_{ik}\mathbf{x}_i,\ \Sigma_k = \frac{1}{n_k} \sum_{i=1}^n G_{ik}(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T, \text{ and } \beta_k = {}^{n_k}/_n, \tag{3.5}$$

where $n_k = \sum_{i=1}^n G_{ik}$ *corresponds to the number of samples assigned to component* k*. One can repeat equations 3.4 and 3.5 until convergence. These ideas are shown in Algorithm 2*

Algorithm 2: Expectation-Maximization

```
function em(X, {β_k^(0), μ_k^(0), Σ_k^(0)}_{k=1}^K)
    while not converged do
        // Expectation
        G_ik = β_k P_k(x_i) / Σ_{k'} β_{k'} P_{k'}(x_i)
        // Maximization
        μ_k = (1/n_k) Σ_{i=1}^n G_ik x_i,  Σ_k = (1/n_k) Σ_{i=1}^n G_ik (x_i − μ_k)(x_i − μ_k)^T,  and  β_k = n_k/n
    return {β_k^⋆, μ_k^⋆, Σ_k^⋆}_{k=1}^K
```

3.1.2 Integral Probability Metrics

In the previous section, we introduced divergences between probability measures based on the ratio of their densities. Here, we cover integral probability metrics [71, 72], which rely on expectations of functions under each measure. Besides being proper metrics, an immediate advantage of these distances is that they have a meaning even if P and Q do not share a support, a case for which, for instance, $f-$divergences are not finite.

Definition 12. *(Integral Probability Metric) Let $P, Q \in \mathbb{P}(\mathcal{X})$. For a family of functions $\mathcal{F}$, an IPM is given by,*

$$\mathcal{D}_{\mathcal{F}}(P,Q) = \sup_{f\in\mathcal{F}} \left| \int_{\mathcal{X}} f(x)dP(x) - \int_{\mathcal{X}} f(x)dQ(x) \right| = \sup_{f\in\mathcal{F}} \left| \mathop{\mathbb{E}}_{x\sim P}[f(x)] - \mathop{\mathbb{E}}_{x\sim Q}[f(x)] \right|.$$

From the perspective of machine learning, IPMs are widespread in generative modeling. Curiously, the original work of [73] used the Jensen-Shannon divergence, which is itself based on the KL divergence. However, many works since then considered IPMs, such as [74, 37], and [51]. Furthermore, in domain adaptation, works usually use IPMs for measuring distances between domains [20, 23, 25].

Now, we give some examples of IPMs relevant for this thesis. First, due to Kantorovich duality (c.f., definition 5), the $1-$Wasserstein distance is an IPM. This fact, however, is not true for other values of α. Next, we define a prominent IPM used in domain adaptation, called $\mathcal{H}$-distance [75].

Definition 13. *($\mathcal{H}$-distance) Let $\mathcal{X}$ be a set, and $P, Q \in \mathbb{P}(\mathcal{X})$. Let $\mathcal{H}$ be a family of functions from $\mathcal{X}$ to $\{0,1\}$. Let $\mathbb{1}_h$ denote the indicator function associated with $h \in \mathcal{H}$, i.e., $x \in \mathbb{1}_h \iff h(x) = 1$. In this case,*

$$\mathcal{D}_{\mathcal{H}}(P,Q) = 2 \sup_{h\in\mathcal{H}} |P(\mathbb{1}_h) - Q(\mathbb{1}_h)|.$$

The $\mathcal{H}-$distance is an IPM, when $\mathcal{F}$ as the set of indicator functions associated with functions $h \in \mathcal{H}$. An useful property of $\mathcal{D}_{\mathcal{H}}$ is that it may be easily estimated from finite samples,

Lemma 1. *Let $\mathcal{H}$ be a family of functions from $\mathcal{X}$ to $\{0,1\}$, such that for each $h \in \mathcal{H}$, its symmetric $1-h \in \mathcal{H}$. For samples $\mathbf{X}^{(P)}$ and $\mathbf{X}^{(Q)}$ of size n and m from P and Q, respectively,*

$$\mathcal{D}_{\mathcal{H}}(\hat{P},\hat{Q}) = 2\left(1 - \min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{h(\mathbf{x}_i^{(P)}) = 0\} + \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}\{h(\mathbf{x}_j^{(Q)}) = 1\}\right). \tag{3.6}$$

As discussed in [75, 20] and [76], the empirical $\mathcal{H}-$distance can be conveniently calculated once $\mathcal{H}$ has been defined. First, one constructs an artificially labeled dataset $\{(\mathbf{x}_i^{(P)},0)\}_{i=1}^n \cup \{(\mathbf{x}_j^{(Q)},1)\}_{j=1}^m$, then, one finds the hypothesis $h^\star = \mathcal{H}$ that minimizes the error on this dataset. Equation 3.6 is then equivalent to the generalization error of $h^\star$. Based on this empirical estimator [20] proves an error bound between $\mathcal{D}_{\mathcal{H}}(P,Q)$ and $\mathcal{D}_{\mathcal{H}}(\hat{P},\hat{Q})$,

Lemma 2. *Let $\mathcal{H}$ be a hypothesis class with VC-dimension $VC(\mathcal{H})$, and let $\mathbf{X}^{(P)}$ and $\mathbf{X}^{(Q)}$ be two samples of sizes n from P and Q. Then, for any $0<\delta<1$, with probability at least $1-\delta$,*

$$\mathcal{D}_{\mathcal{H}}(P,Q) \leq \mathcal{D}_{\mathcal{H}}(\hat{P},\hat{Q}) + \underbrace{4\sqrt{\frac{VC(\mathcal{H})\log(2n) + \log(2/\delta)}{n}}}_{\mathcal{C}_{\mathcal{H}}(n,\delta,\mathcal{H})}, \tag{3.7}$$

where $\mathcal{C}_{\mathcal{H}}$ is referred to as sample complexity of empirically approximating the $\mathcal{H}-$distance.

Due the fact that the $\mathcal{H}-$distance is inherently linked to classification, it should not appear as a surprise that its sample complexity is linked to the VC-dimension of the hypothesis space $\mathcal{H}$, which itself serves as a complexity measure for the hypothesis class. We give further details and discussion about this concept in Chapter 4, especially in section 4.1.

The next IPM is based on functional analysis, especially the notion of Reproducing Kernel Hilbert Space (RKHS) [77, Section 2.2.3].

Definition 14. *(Reproducing Kernel Hilbert Space) Let $\mathcal{X}$ be a set, and $\mathcal{F}$ be a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$. Then $\mathcal{F}$ is called a RKHS endowed with the dot product $\langle\cdot,\cdot\rangle_{\mathcal{F}}$ and the norm $\|h\| = \sqrt{\langle f, f\rangle_{\mathcal{F}}}$ if there exists a kernel $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ that satisfies the reproducing property,*

$$\langle f, k(\mathbf{x},\cdot)\rangle_{\mathcal{F}} = f(\mathbf{x}), \tag{3.8}$$

and spans $\mathcal{H}$, i.e., each $f \in \mathcal{F}$ can be expressed as $f = k(\cdot,\mathbf{x}')$ for $\mathbf{x}' \in \mathcal{X}$.

On an intuitive level, an RKHS define a space of functions based on a kernel k. This space is special due the reproducing property (equation 3.8), which guarantee that the evaluation functional, i.e., $f \mapsto f(\mathbf{x})$, is continuous [78, Section 2.2]. Back to our discussion of IPMs, [52, 78] define a metric based on RKHS, called the MMD. The principle behind this metric is to compare two distributions $P, Q \in \mathbb{P}(\mathcal{X})$ based on the *mean embedding* of P (resp. Q) into $\mathcal{H}$, that is $\mu_P = \mathbb{E}_{\mathbf{x}\sim P}[f(\mathbf{x})] = \langle f, \mu_P\rangle_{\mathcal{H}}$. In the next definition, we offer three equivalent forms of this metric, which are consequences of the reproducing property [78, Lemma 4, Lemma 6].

Definition 15. *(Maximum Mean Discrepancy) Let $P, Q \in \mathbb{P}(\mathcal{X})$, and $\mathcal{F}$ be a RKHS. The MMD is given by,*

$$MMD_{\mathcal{F}}(P,Q) = \sup_{\|f\|_{\mathcal{F}} \leq 1} \left| \mathop{\mathbb{E}}_{x\sim P}[f(x)] - \mathop{\mathbb{E}}_{x\sim Q}[f(x)] \right|, \tag{3.9}$$

or, equivalently,

$$MMD_{\mathcal{F}}(P,Q) = \|\mu_P - \mu_Q\|_{\mathcal{F}}, \tag{3.10}$$

and,

$$MMD_{\mathcal{F}}(P,Q) = \sqrt{\mathop{\mathbb{E}}_{x,x'\sim(P,P)}[k(x,x')] + \mathop{\mathbb{E}}_{x,x'\sim(Q,Q)}[k(x,x') - 2\mathop{\mathbb{E}}_{x,x'\sim(P,Q)}[k(x,x')]]}. \tag{3.11}$$

The discussion in [78] offers further intuition on the reasoning behind the definition of the RKHS and the MMD in equation 3.9. For instance, under IPMs (definition 12), it satisfies that $\mathcal{F} = \mathcal{C}_b(\mathcal{X})$, the space of bounded continuous functions, to have $P = Q \iff \mathcal{D}_{\mathcal{F}}(P,Q) = 0$. Nonetheless, under finite samples, this family of functions is not practical. As a result, one needs to search for a functional space that is, at the same time, rich enough and practical under finite samples. From definition 15 (or lemmas 4 and 6 of [78]), defining an RKHS satisfies this desiderata.

As discussed in [78, Lemma 6], when i.i.d. samples from P and Q are available, the MMD can be estimated by replacing the expectations in equation 3.11 by empirical averages,

$$\begin{aligned}\mathrm{MMD}_{\mathcal{F}}(\hat{P},\hat{Q}) = \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n} k(\mathbf{x}_i^{(P)},\mathbf{x}_j^{(P)}) + \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m} k(\mathbf{x}_i^{(Q)},\mathbf{x}_j^{(Q)}) \\ - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k(\mathbf{x}_i^{(P)},\mathbf{x}_j^{(Q)}).\end{aligned} \tag{3.12}$$

Similarly to lemma 2, we present [79, Theorem 4] as a lemma bounding the true and empirical MMDs,

Lemma 3. *Let $P, Q \in \mathbb{P}(\mathcal{X})$ be two probability measures, and $\mathcal{F}$ be an RKHS with kernel k, such that $\forall \mathbf{x}^{(P)}, \mathbf{x}^{(Q)}$, $|k(\mathbf{x}^{(P)}, \mathbf{x}^{(Q)})| \leq M$. Let $\mathbf{X}^{(P)}$ and $\mathbf{X}^{(Q)}$ be two samples of size n from P and Q. Then, for any $0 < \delta < 1$, with probability at least $1 - \delta$,*

$$|MMD_{\mathcal{F}}(P,Q) - MMD_{\mathcal{F}}(\hat{P},\hat{Q})| \leq \underbrace{4\sqrt{\frac{M}{n}} + \frac{1}{2}\sqrt{\frac{\log(2/\delta)}{4M}}}_{\mathcal{C}_{MMD}(n,\delta,k)}.$$

we refer to $\mathcal{C}_{MMD}(n,\delta,k)$ as the sample complexity of estimating the MMD.

3.1.3 A comparison of probability metrics

In this section, we compare how different metrics span geometries over the manifold of Gaussian measures, i.e., $\mathcal{M} = \{\mathcal{N}(\mu,\sigma^2) : \mu \in \mathbb{R}, \sigma \in \mathbb{R}_+\}$. We thus denote the parameters of this family as $\theta = (\mu,\sigma)$. This family of measures is convenient for two reasons. First, we may associate points

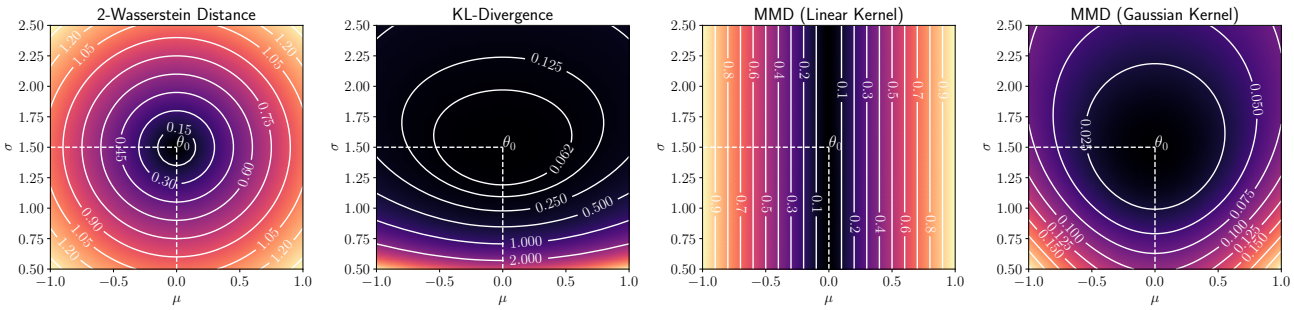


Figure 3.2 – Geometry on the family of Gaussian measures given by the Wasserstein distance, Kullback-Leibler divergence and the MMD under a linear and Gaussian kernels.

(μ, σ) in the positive half-plane $\mathbb{R} \times \mathbb{R}_+$ with Gaussian measures. As a result, given a fixed θ_0, we can visualize and interpret the distance $\mathcal{D}(\theta, \theta_0)$ through heat maps. Furthermore, this manifold is convenient, as most metrics have a closed-form solution for the Gaussian family (e.g., the Wasserstein distance or the Kullback-Leibler divergence).

For our comparison, we fix $\theta_0 = (0.0, 1.5)$, then we compare the metrics $\mathcal{D}(\theta, \theta_0)$ for $\theta \in \mathbb{R} \times \mathbb{R}_+$. We focus on the metrics discussed in the previous sections, namely, the $2-$Wasserstein distance, the KL divergence, and the MMD under a linear, and a Gaussian kernel. We show these ideas in figure 3.2.

From figure 3.2, a first striking remark is how different the geometries under different discrepancies are. Especially, the 2-Wasserstein distance yields an Euclidean geometry over $\mathcal{M}$, while the KL divergence is associated with an hyperbolic geometry (see e.g., [10, Remark 8.2.] for further details). Furthermore, the MMD depends on the kernel choice. Under a linear kernel, this metric cannot differentiate past 1st order moments. This highlights the idea that one should choose a complex enough kernel for comparing distributions [78, Theorem 5], such as universal kernels [80].

In general, the choice of discrepancy for comparing measures is essential to machine learning algorithms. For instance, when modeling a probability measure through a parametric model P_θ, it may be advantageous to use IPMs over $f-$divergences, as these are defined and have a gradient even if the $\text{supp}(P_\theta)$ and $\text{supp}(P)$ are different. From the perspective of this thesis, we are particularly interested in the interpolation of probability measures. In this sense, the Wasserstein distance is advantageous, since Wasserstein barycenters are widely studied and capture the data geometry.

3.2 Barycenters of Probability Measures

In physics, a barycenter is the center of mass of a group of objects. This perspective reflects its etymology, as the word *barycenter* comes from Greek and can be roughly translated as *heavy center*. For instance, for a set of n objects positioned at $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^d$, with masses $m_1, \cdots, m_n \geq 0$, the barycenter of these n objects is the point $\bar{\mathbf{x}}$ such that,

$$\bar{\mathbf{x}} = \underset{\mathbf{x} \in \mathbb{R}^d}{\text{argmin}} \sum_{i=1}^{n} m_i \|\mathbf{x} - \mathbf{x}_i\|_2^2. \tag{3.13}$$

This idea can be generalized in various ways. First, since all masses are positive, one can normalize

them by $M = \sum_i m_i$. As a result, the barycenter $\bar{\mathbf{x}}$ depends on the relative masses $\lambda = \{m_i/M\}_{i=1}^n$ which belongs to the simplex $\Delta_n = \{\mathbf{a} \in \mathbb{R}_+^n : \sum_i a_i = 1\}$. Next, equation 3.13 can be generalized to other metric spaces. These elements motivate the next definition, due to [81].

Definition 16. *(Fréchet Means) Let $(\mathcal{X}, d)$ be a metric space. Let $\{x_i\}_{i=1}^N$ be points in M, with weights $\lambda_i \geq 0$ such that $\sum_{i=1}^N \lambda_i = 1$. For any point $x \in \mathcal{X}$, the Fréchet variance is the function $\Psi : \mathcal{X} \to \mathbb{R}$ given by,*

$$\Psi(x) = \sum_{i=1}^{N} \lambda_i d(x, x_i)^2.$$

As follows, the Fréchet mean, or barycenter *is a point $x^\star \in M$ that minimizes Ψ,*

$$x^\star = \underset{x \in M}{\operatorname{argmin}}\, \Psi(x). \tag{3.14}$$

As follows, when we equip $\mathbb{P}(\mathcal{X})$ or $\mathbb{W}_p(\mathcal{X})$ with a probability metric (e.g., an IPM or the Wasserstein distance), we can extend the geometric notion of mean or barycenter to probability measures. This extension leads, for instance, to the concept of Wasserstein barycenters [82], which were widely studied and applied in machine learning, but other metrics can be used as well. Based on these ideas, we give the following definition.

Definition 17. *(Barycenter of Probability Measures) Let $(\mathbb{P}(\mathcal{X}), \mathcal{D})$ is a metric space. Let $\mathcal{P} = \{P_k\}_{k=1}^K$ be a set of probability measures with $P_k \in \mathbb{P}(\mathcal{X})$ and $\lambda \in \Delta_K$ be a set of barycentric coordinates. The barycenter of $\mathcal{P}$, with respect λ and $\mathcal{D}$ is,*

$$B^\star = \mathcal{B}_{\mathcal{D}}(\lambda, \mathcal{P}) = \underset{B \in \mathbb{P}(\mathcal{X})}{\operatorname{argmin}}\, \Psi(B) = \sum_{k=1}^{K} \lambda_k \mathcal{D}(B, P_k). \tag{3.15}$$

We further define the barycentric hull [1] *of $\mathcal{P}$ as the set,*

$$\mathcal{M}_{\mathcal{D}}(\mathcal{P}) = \{\mathcal{B}_{\mathcal{D}}(\lambda, \mathcal{P}) : \alpha \in \Delta_K\}. \tag{3.16}$$

Henceforth we refer to $\mathcal{B}$ as *barycentric operator*. It maps the pair $(\lambda, \mathcal{P})$ to the barycentric measure in $\mathbb{P}(\mathcal{X})$. While this definition is much more general, the main focus of this thesis are $2-$Wasserstein barycenters, especially when $\mathcal{D} = \mathcal{W}_2^2$.

3.2.1 Multi-Marginal Optimal Transport

Throughout this section, we assume $\mathcal{D} = \mathcal{W}_2^2$. As we previously discussed, these kinds of barycenters were first introduced in [82]. The goal of this section is to define an extension of OT, known as Multi-Marginal OT (MMOT) [83], and to explain its connection to the barycenter problem.

1. We name this set in reference to the notion of convex hull.

Definition 18. *(Multi-Marginal Optimal Transport) Let $\mathcal{X}$ be a set, $\mathcal{P} = \{P_1, \cdots, P_K\}$ and Q be $K+1$ probability measures in $\mathbb{P}(\mathcal{X})$. Let $\bar{c} : \mathcal{X}^{K+1} \to \mathbb{R}$ be a ground-cost. The MMOT problem is,*

$$\gamma^\star = \underset{\gamma \in \Gamma(P_1, \cdots, P_K, Q)}{\operatorname{arginf}} \int_{\mathcal{X}^{K+1}} \bar{c}(\mathbf{x}_1, \cdots, \mathbf{x}_K, \mathbf{x}) d\gamma(\mathbf{x}_1, \cdots, \mathbf{x}_K, \mathbf{x}), \tag{3.17}$$

where, similarly to the Kantorovich formulation (c.f., definition 3), Γ is a subset of $\mathbb{P}(\mathcal{X}^{K+1})$ of measures with marginals $P_1, \cdots, P_K, Q$.

The connection between equations 3.15 (with $\mathcal{D} = \mathcal{W}_2^2$) and 3.17 comes from the use of the *gluing lemma* [9, Chapter 1], which is a measure theoretic tool that allows us to "glue" together probability measures. Intuitively, if γ_1 is an OT plan between P_1 and B, and γ_2 is an OT plan between P_2 and B, it is possible to glue γ_1 and γ_2 together into γ, which is an OT plan between P_1, P_2, and B. As a result, one may glue the OT plans $\gamma_1, \cdots, \gamma_K$, in which γ_k is an OT plan between P_k and B, and define the following cost,

$$\bar{c}(\mathbf{x}_1, \cdots, \mathbf{x}_K, \mathbf{x}) = \sum_{k=1}^{K} \lambda_k c(\mathbf{x}, \mathbf{x}_k).$$

With these tools, one may cast the barycenter problem into the MMOT problem (see e.g., [82, Section 4] or [84, Section 6]), which involves minimizing equation 3.17 with respect the last marginal $Q = B$. This minimization process can be carried out by defining an equivalent problem with respect $P_1, \cdots, P_K$. As such, let $\tilde{c}(\mathbf{x}_1, \cdots, \mathbf{x}_K) = \sum_k \lambda_k c(\mathbf{x}_k, T_\lambda(\mathbf{x}_1, \cdots, \mathbf{x}_k))$. The mapping T_λ is the barycentric mapping (c.f., equation 2.19), for weights $\lambda \in \Delta_K$. This defines an MMOT problem over K marginals,

$$\gamma^\star = \underset{\gamma \in \Gamma(P_1, \cdots, P_K)}{\operatorname{arginf}} \int_{\mathcal{X}^K} \sum_{k=1}^{K} \lambda_k c(\mathbf{x}_k, T_\lambda(\mathbf{x}_1, \cdots, \mathbf{x}_k)) d\gamma(\mathbf{x}_1, \cdots, \mathbf{x}_K). \tag{3.18}$$

If the probability measures $P_1, \cdots, P_K$ are empirical, it is possible to solve equation 3.18 through linear programming. In this case, $\gamma^\star$ is be a $K-$order sparse tensor of shape $(n_1, \cdots, n_k)$ and $\sum_k n_k - K + 1$ non-zero entries [10, Remark 10.2]. In this case, the Wasserstein barycenter is explicitly defined:

$$\hat{B}^\star(\mathbf{x}) = \sum_{(i_1, \cdots, i_K)} \gamma_{i_1, \cdots, i_K} \delta(\mathbf{x} - T_\lambda(\mathbf{x}_{i_1}^{(P_1)}, \cdots, \mathbf{x}_{i_K}^{(P_K)})). \tag{3.19}$$

Even though MMOT offers an explicit form for Wasserstein barycenters, it scales poorly with the number of samples in the supports of P_k's, and the number of marginals K. These limitations motivate the next section, which takes a different route in minimizing 3.15 with respect an empirical $\hat{B}$.

3.3 Computational Methods

Before proceeding to computational methods, let us highlight a few nuances with the calculation of barycenters of probability metrics. In the following, we divide the computational strategies into 2 classes of methods: empirical and parametric (Gaussian or Gaussian mixture) barycenters. Let us

focus on the first case, where one introduces empirical approximations for the distributions $P_1, \cdots, P_K$ (see e.g., section 2.2.1). The empirical approximation of P is divided into 2 strategies, namely fixed and free support. In the first case, $\{P_k\}_{k=1}^K$ share a common support. These measures differ in their weights $\mathbf{p}_k \in \Delta_n$. In the second case, one assumes $\{\mathbf{x}_i^{(P_k)}\}_{i=1}^{n_k}$, where $\mathbf{x}_i^{(P_k)} \overset{i.i.d.}{\sim} P_k$, i.e., the measures do not share a support anymore.

The fixed and free-support strategies determine which metrics are applicable. While the Wasserstein distance can be used in both free and fixed support cases, $f-$divergences assume that $\text{supp}(P) \subset \text{supp}(Q)$. As a result, free-support approximations of measures are not suitable. In the same line of reasoning, the fixed and free-support strategies offer different interpretations for data. In the first case, data takes the form of histograms over the fixed, shared support, in the sense that the vectors $\mathbf{p}_k$ are positive and sum to one. In the second case, one has data matrices $\mathbf{X}^{(P_k)} \in \mathbb{R}^{n_k \times d}$, in which each row represents a sample from P_k.

3.3.1 Empirical Wasserstein Barycenters

This section is primarily based on [26], as we introduce the computational tools for the calculation of empirical Wasserstein barycenters. For empirical $\hat{P}_k$, with support $\mathbf{X}^{(P_k)}$, the Wasserstein barycenter $\hat{B}$ of $\mathcal{P} = \{\hat{P}_k\}_{k=1}^K$ is an empirical distribution with support $\mathbf{X}^{(B)} \in \mathbb{R}^{n_B \times d}$ and weights $\mathbf{b} \in \Delta_{n_B}$. These parameters may be determined through optimization (equation 3.14),

$$(\mathbf{b}^\star, \mathbf{X}^{(B^\star)}) = \underset{\substack{\mathbf{b} \in \Delta_{n_B} \\ \mathbf{X}^{(B)} \in \mathbb{R}^{n_B \times d}}}{\text{argmin}} \sum_{k=1}^K \lambda_k \mathcal{W}_\alpha \left(\sum_{i=1}^{n_B} b_i \delta(\mathbf{x} - \mathbf{x}_i^{(B)}), \hat{P}_k \right)^\alpha. \tag{3.20}$$

In this context, there are three strategies for solving equation 3.20, depending on whether the support of $\hat{P}_k$ is free or not. The first strategy [26, Algorithm 1], called fixed-support Wasserstein barycenter, assumes $\mathbf{X}^{(B)}$ given and fixed. In this case, one solves equation 3.20 for $\mathbf{b} \in \Delta_{n_B}$. The second strategy, called free-support, assumes $\mathbf{b} \in \Delta_{n_B}$ given and fixed, and solves for $\mathbf{X}^{(B)}$. The third strategy [26, Algorithm 2] mixes the first two by alternating optimization. One first fixes $\mathbf{X}^{(B)}$ and solves equation 3.20 for $\mathbf{b}$, then fixes $\mathbf{b}$ and solves for $\mathbf{X}^{(B)}$. We now describe each of these strategies.

In its primal formulation, the sample weights appear indirectly in $\mathcal{W}_\alpha$, through the constraints $\Gamma(\mathbf{p}, \mathbf{q})$. It is thus more useful to use the dual formulation (c.f., equation 2.20), in which they appear explicitly in the formula. As such, one may rewrite equation 3.20 as,

$$\mathcal{L}(\mathbf{b}) = \underset{\mathbf{f}_k, \mathbf{g}_k \in \Phi(\mathbf{C})}{\text{argmax}} \sum_{k=1}^K \lambda_k (\mathbf{f}_k^T \mathbf{b} + \mathbf{g}_k^T \mathbf{p}_k),$$

which is convex, since $\mathcal{L}(\mathbf{b})$ is the maximum of a set of affine functions. As a result, let $(\mathbf{f}_k^\star, \mathbf{g}_k^\star)$ be solutions to the dual Kantorovich problem, the sub-gradient of $\mathcal{L}$ is $\partial\mathcal{L} = \sum_{k=1}^K \lambda_k \mathbf{f}_k^\star$. Note that, from the sub-gradient, one cannot extract an useful first-order optimality condition. As a result one needs to initialize $\mathbf{b}$, then optimize it by iteratively solving OT and updating its value. This is the essence of [26, Algorithm 1], shown in algorithm 3 below.

Alternatively, if in place of $\mathcal{W}_\alpha$ one has the Sinkhorn divergence $\mathcal{W}_{\alpha,\epsilon}$, one can derive an alternative algorithm called Iterative Bregman Projections (IBP) [50], which relies on the link between entropic OT

Algorithm 3: Fixed-Support Wasserstein barycenter algorithm.

```
function fixed_support_wbary(C, P, λ, η, τ)
    // Initialization.
    b^(0) ← 1_n/n
    ε = +∞
    // Updates.
    while ε > τ do
        for k = 1, ··· , K do
            f_k^(it) = argmax_{f,g∈Φ(C)} f^T a_k + g^T b_i
        b^(it+1) = P_{Δ_n}( b^(it) e^{−Σ_{k=1}^K λ_k f_k^(it)} )
        ε = ‖b^(it+1) − b^(it)‖
    return b*
```

and a projection under the KL divergence (c.f., remark 3). In this setting, the Wasserstein barycenter problem is equivalent to,

$$\gamma^\star = \operatorname{argmin} \left\{ \sum_{k=1}^{K} \mathrm{KL}(\gamma_k | \xi_k); \forall k,\ \gamma_k^T \mathbb{1} = \mathbf{p}_k \text{ and } \exists \mathbf{p} \in \Delta_n \text{ s.t. } \gamma_k \mathbb{1} = \mathbf{p} \right\}. \tag{3.21}$$

As [50, Proposition 2] shows, this problem can be solved with Sinkhorn-like iterations. Indeed, let $\mathbf{v}_k^{(0)} = \mathbb{1}_n/n$ and $\xi_k = e^{-\mathbf{C}_k/\epsilon}$. The iterations read as,

$$\mathbf{u}_k^{(\ell)} = \frac{\mathbf{b}^{(\ell)}}{\xi_k \mathbf{v}_k^{(\ell)}}, \text{ and, } \mathbf{v}_k^{(\ell+1)} = \frac{\mathbf{p}_k}{\xi_k^T \mathbf{u}_k^{(\ell)}},$$

where $\mathbf{b}^{(\ell)}$ is the current estimate of the Wasserstein barycenter, i.e.,

$$\mathbf{b}^{(\ell)} = \prod_{k=1}^{K} (\mathbf{u}_k^{(\ell)} \odot (\xi_k \mathbf{v}_k^{(\ell)}))^{\lambda_k}$$

Example 5. *(Fixed-support barycenters of images) In this example, we illustrate the calculation of fixed-support barycenters. A (gray-scale) image is a 2-D array* $I \in \mathbb{R}^{h \times w}$ *with* h *rows and* w *columns. A pair* $\mathbf{x} = (i, j)$*, such that* $0 \leq i \leq h-1$ *and* $0 \leq j \leq w-1$ *is called* a pixel. *Note that* $I(\mathbf{x}) = I(i, j) \in \{0, \cdots, 255\}$ *is called pixel intensity, and is shown as a shade of red in our example.*

In this setting, we normalize the images so as to represent them as histograms, i.e., we divide I *by* $\sum_{i=0}^{h-1} \sum_{j=0}^{w-1} I(i, j)$*. Furthermore, we consider a regular grid over images of the same size, and we use an Euclidean ground-cost for comparing pixels. An example of these ideas is shown in figure 3.3.*

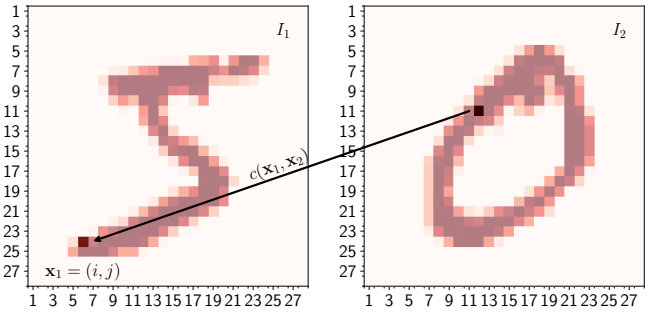


Figure 3.3 – Optimal transport between two images, I_1 and I_2, understood as histograms. The ground-cost is computed between pairs of pixels $\mathbf{x}_1 = (i, j)$ and $\mathbf{x}_2 = (k, \ell)$.

As follows, in figure 3.4 we compare barycenters under the Wasserstein metric and Sinkhorn divergence, and barycenters under the Euclidean distance, i.e., $I_t = (1-t)I_1 + tI_2$*. Note that, while the interpolation under the Euclidean metric mixes the two digits, the Wasserstein and Sinkhorn barycenters better preserves the local geometric structures of the images.*

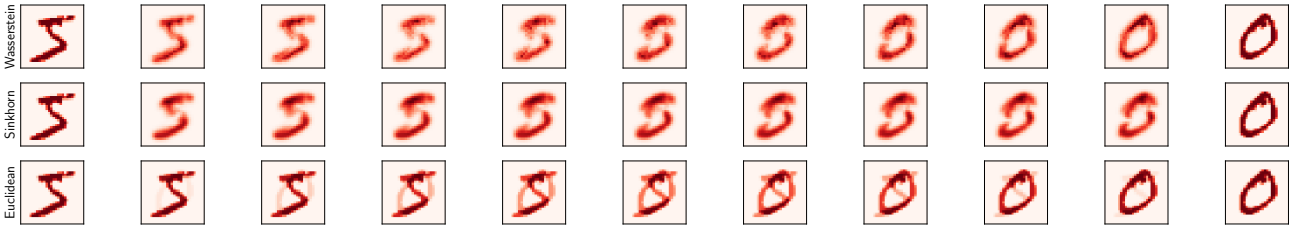


Figure 3.4 – Wasserstein, Sinkhorn and Euclidean barycenters of two images.

Overall, this example illustrates an important property of OT: it captures some properties of the underlying geometry of data. This feature comes from the fact that OT relies on distances in the ambient space $\mathcal{X}$ *of* $\mathbb{P}(\mathcal{X})$*, i.e., OT lifts metrics on* $\mathcal{X}$ *towards* $\mathbb{P}(\mathcal{X})$*.*

As we mentioned in the beginning of this section, in the empirical case one divides the the calculation of Wasserstein barycenters into fixed and free-support methods. As we covered so far, the first focuses on calculating $\mathbf{b}$ for a fixed, given $\mathbf{X}^{(B)}$. Now, we turn our free-support problem, which calculates $\mathbf{X}^{(B)}$ for a given, fixed $\mathbf{b}$.

Let $\mathbf{X}^{(B)}$ be a given initialization for the barycenter support, and let $\gamma_k^\star = \mathrm{OT}(\mathbf{X}^{(B)}, \mathbf{X}^{(P_k)})$. One

may express the Wasserstein barycenter objective in equation 3.20 as,

$$\mathcal{L}(\mathbf{x}_1^{(B)}, \cdots, \mathbf{x}_n^{(B)}) = \sum_{k=1}^{K} \lambda_k \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma^\star_{k,i,j} \|\mathbf{x}_i^{(B)} - \mathbf{x}_j^{(P_k)}\|_2^2. \tag{3.22}$$

From equation 3.22, it is immediate that the first order conditions of $\mathcal{L}$ with respect each $\mathbf{x}_i^{(B)}$ correspond to a barycentric mapping, as in the proposition below.

Proposition 2. *Let $\mathcal{P} = \{\hat{P}_1, \cdots, \hat{P}_K\}$ be $K \geq 1$ empirical probability measures with weights $\mathbf{p}_k \in \Delta_{n_k}$ and support $\mathbf{X}^{(P_k)} \in \mathbb{R}^{n_k \times d}$. Given a number of samples $n_B \in \mathbb{N}$, let $\hat{B}^\star$ be the empirical measure minimizer of $B \mapsto \sum_k \lambda_k \mathcal{W}_2(B, \hat{P}_k)^2$, with weights $\mathbf{b} \in \Delta_{n_B}$ and support $\mathbf{X}^{(B)} \in \mathbb{R}^{n_B \times d}$. Then, the support of $\hat{B}^\star$ satisfies,*

$$\mathbf{X}^{(B)} = diag(1/\mathbf{b}) \sum_{k=1}^{K} \lambda_k \gamma_k^\star \mathbf{X}^{(P_k)} = \sum_{k=1}^{K} \lambda_k T_{\gamma_k^\star}(\mathbf{X}^{(B)}), \tag{3.23}$$

where γ_k is the OT plan between $\hat{B}^\star$ and $\hat{P}_k$.

Proof. Taking derivatives of equation 3.22 with respect $\mathbf{x}_i^{(B)}$,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}_i^{(B)}} = 2 \sum_k \lambda_k \sum_j \gamma^\star_{k,i,j} (\mathbf{x}_i^{(B)} - \mathbf{x}_j^{(P_k)}).$$

Solving for $\mathbf{x}_i^{(B)}$ gives us $\mathbf{x}_i^{(B)} = b_i^{-1} \sum_{k=1}^K \lambda_k \sum_j \gamma^\star_{k,i,j} \mathbf{x}_j^{(P_K)} = \sum_{k=1}^K \lambda_k T_{\gamma_k^\star}(\mathbf{x}_i^{(B)})$. Grouping the samples into the support matrix gives us the formula in equation 3.23. □

This proposition was first proven by Cuturi and Doucet in [26, Section 4.4]. The authors thus propose to calculate the support $\mathbf{X}^{(B)}$ of $\hat{B}^\star$ by iterating,

$$\mathbf{X}_{it+1}^{(B)} = (1-\theta)\mathbf{X}_{it}^{(B)} + \theta \sum_{k=1}^{K} \lambda_k T_{\gamma_k^\star}(\mathbf{X}_{it}^{(B)}),\ \theta \in [0,1], \tag{3.24}$$

where θ is found via line-search. In [14], however, we argue in favor of $\theta := 1$, so that the barycenter iterations resemble the fixed-point iterations of [85],

$$\mathbf{X}_{it+1}^{(B)} = \sum_{k=1}^{K} \lambda_k T_{\gamma_k^\star}(\mathbf{X}_{it}^{(B)}).$$

We give our version of the strategy in algorithm 4, keeping in mind that its extension to support the version of [26] (i.e., equation 3.24) is straightforward. Its extension to entropic OT is equally straightforward, as the entropic regularization only affects the calculation of γ. Next, we give an example of barycenters of free-support empirical measures, which we also call *point clouds*.

Algorithm 4: Free-Support Wasserstein barycenter algorithm.

function `free_support_wbary`$(\{\mathbf{p}_k\}_{k=1}^K, \{\mathbf{X}^{(P_k)}\}_{k=1}^K, \lambda, \epsilon, \tau)$

  // Initialization.

  $\mathbf{x}_i^{(B)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$

  $L_0 = 0$

  // Updates.

  while $|L_{it} - L_{it-1}| > \tau$ **do**

    for $k = 1, \cdots, K$ **do**

      $\gamma^{(k,it)} = \mathrm{OT}_\epsilon\left(\mathbf{X}_{it}^{(B)}, \mathbf{X}^{(P_k)}\right)$

    $L_{it} = \sum_{k=1}^K \lambda_k \langle \gamma^{(k,it)}, \mathbf{C}^{(k)} \rangle_F$

    $\mathbf{X}_{it+1}^{(B)} = \sum_{k=1}^K \lambda_k T_{\gamma^{(k,it)}}(\mathbf{X}_{it}^{(B)})$

  return $\mathbf{X}^{(B)}$

Example 6. *(Free-support Wasserstein barycenters of point clouds) In this example, we illustrate free-support Wasserstein barycenters over point clouds. We assume empirical measures* $\hat{P}_1, \cdots, \hat{P}_K$, $K \geq 1$, *such that each has its own support* $\mathbf{X}^{(P_k)} \in \mathbb{R}^{n_k \times d}$, *and uniform weights* $p_{k,i} = 1/n_k$. *In principle, it is possible to estimate the importance weights* $p_{k,i}$ *from the samples, but for the purposes of this example, we assume uniform weights. We illustrate the empirical measures used in this example in Figure 3.5.*

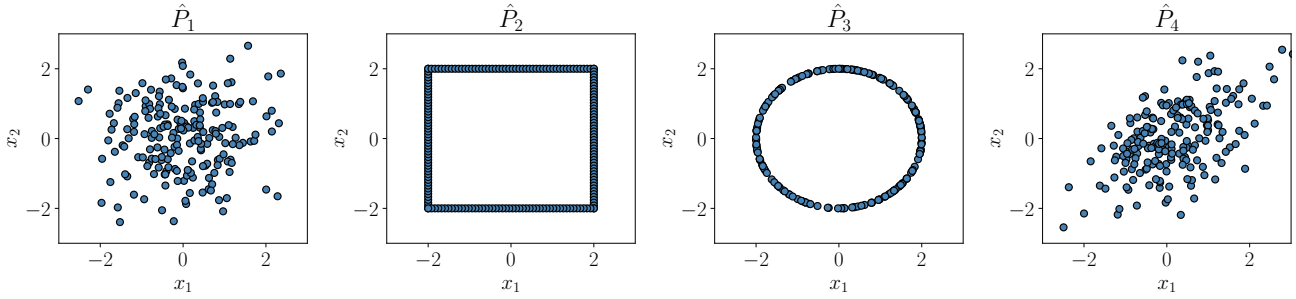


Figure 3.5 – Empirical measures over $\mathcal{X} = \mathbb{R}^2$ with uniform weights.

Next, we illustrate the interpolation between pairs of measures $(\hat{P}_k, \hat{P}_{k'})$, $k = 1 \cdots, K$, $k' \neq k$. We do so by calculating $\mathcal{B}([t, 1-t]; \mathcal{P})$, $t = 0, 1/3, 2/3, 1$. We show the results in Figure 3.6. Note that the barycenters interpolate between the shape and form of the empirical measures. This remark agrees with our previous example, in which we calculated (fixed-support) Wasserstein barycenters of digits. Note that, in Figure 3.6, we use exact OT. It would have been possible to use entropic regularization, but one should be careful about the tuning of the penalty ϵ. We show in Figure 3.7 a case where entropic regularization yields a Wasserstein barycenter that does not capture the shape of the interpolated measures.

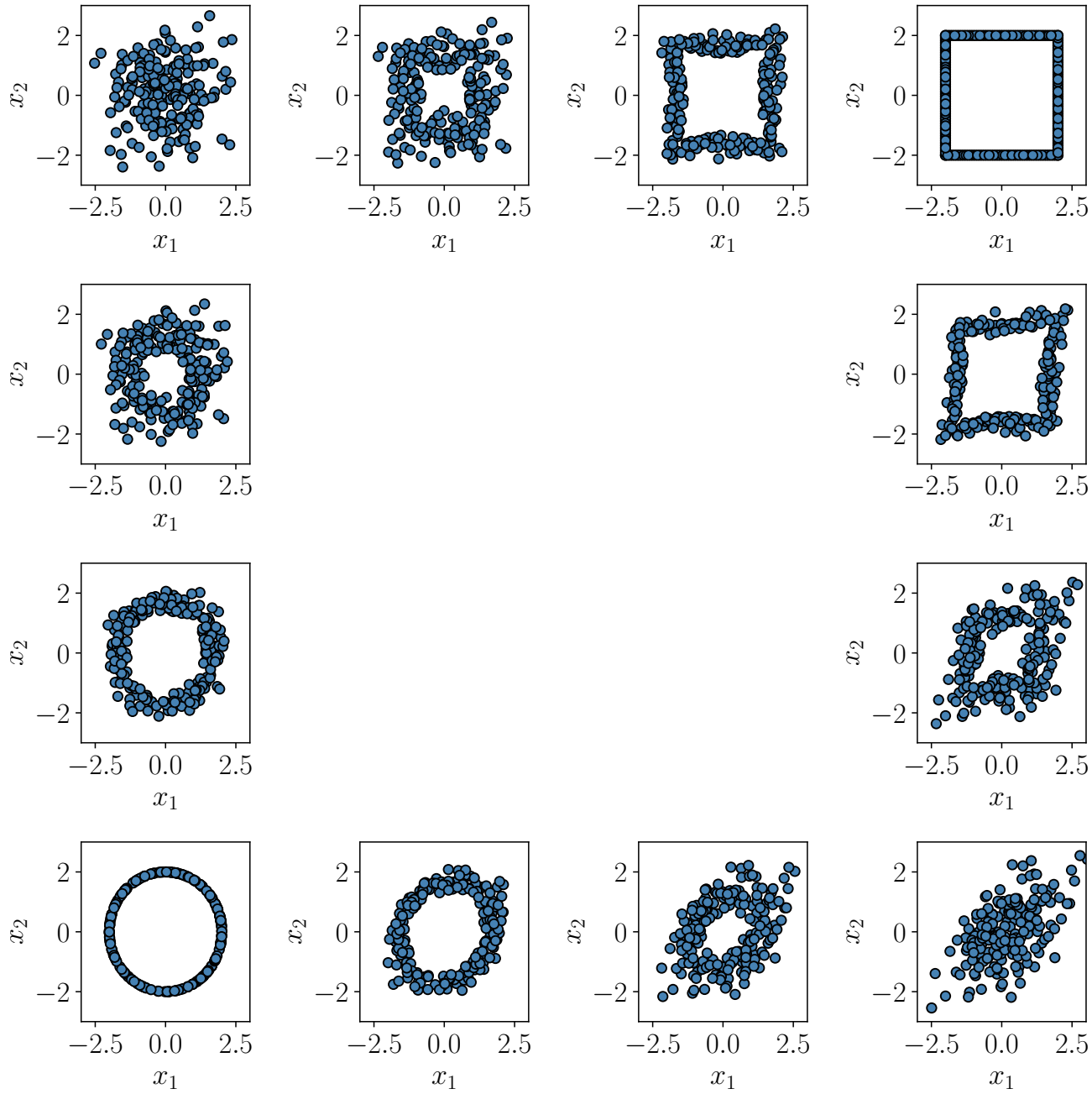


Figure 3.6 – Interpolation of point clouds through Wasserstein barycenters

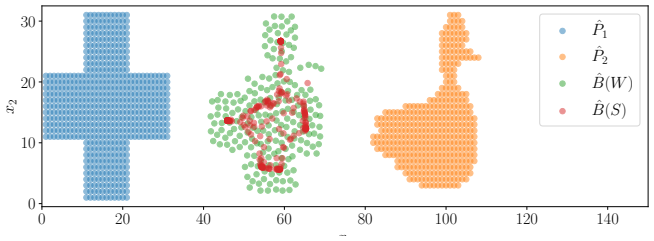


Figure 3.7 – Wasserstein and Sinkhorn barycenters between two geometrical shapes. While the Wasserstein barycenter gives us a point cloud that mixes the geometric properties of $\hat{P}_1$ and $\hat{P}_2$, the Sinkhorn barycenter has artifacts due the entropic regularization.

3.3.2 Wasserstein Barycenters of Gaussians and Gaussian Mixtures

In this section, we explore Wasserstein barycenters of specific families of probability measures, such as Gaussians and Gaussian mixtures. We begin with the first, as they serve as the foundation for the second. These kinds of barycenters were analyzed by [82] and [85].

Theorem 2. *Let $P_1, \cdots, P_K$ be $K \geq 2$ Gaussian measures with parameters (μ_k, Σ_k), $k = 1, \cdots, K$. The barycenter $\mathcal{B}(\lambda; \mathcal{P})$ with respect the squared $2-$Wasserstein metric $\mathcal{W}_2^2$ has parameters $\bar{\mu} = \sum_{k=1}^K \lambda_k \mu_k$ and $\bar{\Sigma}$ such that,*

$$\bar{\Sigma} = \sum_{k=1}^{K} \lambda_k (\bar{\Sigma}^{1/2} \Sigma_k \bar{\Sigma}^{1/2})^{1/2}, \tag{3.25}$$

where powers should be understood in a matricial sense.

While the calculation of the mean vector of $\mathcal{B}(\lambda; \mathcal{P})$ is straightforward, the computation of its covariance matrix is somewhat more involved. In this case, [85] describes an iterative method for solving 3.25, which is outlined in algorithm 5.

Algorithm 5: Barycenter of Gaussian measures

function `gauss_wbary`$(\{\mu_k, \Sigma_k\}_{k=1}^K, \lambda, S_0, N_{iter}, \tau)$

  // Initialization.

  $\delta = \infty$

  // Updates.

  while $\delta > \tau$ **and** $it \leq N_{iter}$ **do**

    $S_{it+1} = S_{it}^{-1/2} \left(\sum_{k=1}^K \lambda_k (S_{it}^{1/2} \Sigma_j S_{it}^{1/2})^{1/2} \right)^2 S_{it}^{-1/2}$

    $\delta = \|S_{it+1} - S_{it}\|_F$

  return $\bar{\mu} = \sum_{k=1}^K \lambda_k \mu_k$, $\bar{\Sigma} = S_{it}$

Now, we turn our attention to barycenters of Gaussian mixtures. Note that, as we discussed in chapter 2 and section 2.2.2, under the mixture-Wasserstein metric, the OT problem is based on

the distance between the components of the Gaussian mixture. As a result, it appears natural that a barycenter of Gaussian mixtures depends on barycenters of its components. In [17], the authors introduce the Gaussian mixture barycenter as a MMOT problem,

$$\gamma^\star = \underset{\gamma \in \Gamma(P_1,\cdots,P_K)\cap \mathrm{GMM}_{Kd}(\infty)}{\operatorname{arginf}} \int \sum_{k=1}^{K} \lambda_k c(\mathbf{x}_k, T_\lambda(\mathbf{x}_1,\cdots,\mathbf{x}_K)) d\gamma(\mathbf{x}_1,\cdots,\mathbf{x}_K). \tag{3.26}$$

In light of the results of [17], which were presented in section 2.2.2 of chapter 2, the *continuous* problem in 3.26 is equivalent to a *discrete* problem,

$$\mathbf{w}^\star = \underset{w\in\Gamma(\mathbf{p}_1,\cdots,\mathbf{p}_K)}{\operatorname{arginf}} \sum_{i_1,\cdots,i_K} w_{i_1,\cdots,i_K} \mathrm{mm}\mathcal{W}_2(P_{1,i_1},\cdots,P_{K,i_K})^2, \tag{3.27}$$

where P_{k,i_k} corresponds to the i_k-th component of the $k-$th measure. As with the empirical MMOT problem, the solution $\mathbf{w}^\star$ is $K-$th order tensor of shape $(n_1,\cdots,n_K)$, and $mm\mathcal{W}_2$ is given by,

$$\mathrm{mm}\mathcal{W}_2(P_{1,i_1},\cdots,P_{K,i_K})^2 = \sum_{k=1}^{K} \lambda_k \mathcal{W}_2(P_{k,i_k}, \mathcal{B}(\lambda;\{P_{1,i_1},\cdots,P_{K,i_K}\}))^2. \tag{3.28}$$

The MMOT formulation of a GMM barycenter offers the same advantage as the empirical MMOT problem, i.e., it has closed form solution,

$$B^\star = \mathcal{B}(\lambda;\mathcal{P}) = \sum_{i_1,\cdots,i_K} w_{i_1,\cdots,i_K} \mathcal{B}(\lambda;\{P_{1,i_1},\cdots,P_{K,i_K}\}),$$

which is a mixture of $\sum_k n_k - K + 1$ components. However, as we mentioned previously, MMOT does not scale well with the number of points in the support of empirical distributions. In the case of GMMs, this translates to the number of components in these mixtures. To give an order of magnitude, in [17] the authors use around 10 components per mixture. This choice is necessary to have reasonable computations, but as we discuss in upcoming sections, in the context of this thesis we need to scale up this strategy, even at the expense of an accurate density model.

A possible strategy for scaling up the GMM-MMOT strategy is deriving similar algorithms to those of [26]. We indeed do so in our publication [15]. We discuss this in further details in chapter 7.

3.4 Conclusion

In this chapter, we presented different notions of divergences and distances in the space of probability measures. The correct choice of metric for $\mathcal{P}(\mathcal{X})$ is essential. Indeed, as we show in Figure 3.2, the notion of distance induces a geometry in the space of probability metrics. If one performs an optimization procedure in this space, the associated geometry influences the minimizers found by the algorithm. This problem is prominent in the field of generative modeling (see, e.g., our discussion in [18]), and as we show in the next chapter, also plays an important role in domain adaptation.

From the perspective of this thesis, defining a distance in $\mathcal{P}(\mathcal{X})$ allows for the definition of a barycenter of probability measures. Arguably, one of the better understood notions of barycenter is

associated with the Wassserstein distance [82]. For this metric, there are well defined algorithms [83, 26, 50, 85] for free, and fixed-support empirical measures. The same can be said for some parametric families of measures, such as Gaussians [85], and Gaussian mixtures [17].

Overall, this chapter serves as a cornerstone for Part II. Especially, in Chapter 5 directly relies on section 3.3, as one of our contributions is extending the free-support Wasserstein barycenter algorithm of [26] to handle labeled measures. Furthermore, Chapter 7 extends the same algorithm for GMMs, thus serving as an alternative to the expensive calculations of MMOT used in [17].

Chapter 4

Learning and Domain Adaptation Theory

> I freely confess: it was the objection of David Hume which first, many years ago, interrupted my dogmatic slumber
>
> Immanuel Kant

Contents

At the core of this thesis, we deal with the problem of generalization. From the perspective of machine learning, this property can be loosely defined as the ability of performing accurate predictions on unseen samples. Here, one can make a parallel with a student preparing for a test. The training set consists of questions for which they have the answers (supervised learning). Based on their act of studying, or training to conform with the usual jargon, they need to correctly answer a set of unseen questions, that is, the test. Here, one supposes that the same patterns or principles are present in the

training, and test sets of questions, otherwise the training process undergone by the student would be meaningless. Imagine that someone mistakenly studied English literature for a physics exam, which would likely result in a very poor test performance.

To continue and stress our example, imagine now that our unfortunate student mistook physics for mathematics instead. Arguably, physics is much closer to mathematics than to literature. As a result, one could expect that, even under a *shift* in the concepts approached by these areas, some level of knowledge could still be re-used, or is shared between the two domains.

In the past two paragraphs we described, on an informal level, some principles behind transfer learning. In the same spirit of the introduction of this thesis (c.f., Chapter 1), and the imitation game of Turing [3], we can define the goal of transfer learning as trying to mimic the remarkable ability of human intelligence to seamlessly adapt to new, related context or tasks, given a repertory of already known concepts. This area of machine learning aims at re-using knowledge on closely related datasets, so that the learning task on a target becomes easier. In this thesis we deal with a specific type of transfer learning, known as domain adaptation. This problem supposes that train and test, or source and target to adequate our jargon, share the same task. The difference between source and target comes in the form of a *shift* in the probability distributions of the elements we are trying to predict.

Before delving into the technical description of the problem, let us go back to the problem of generalization. Another way of looking at this problem, is to see it as a process of induction [4], that is, to *define a general rule out of specific examples*. As it is, this problem is largely considered in philosophy, for instance in metaphysics and philosophy of science. For instance, here is an example of inductive reasoning: *From the fact that the sun has risen every day thus far, we conclude that it will rise again tomorrow*. Besides the philosophical validity of inductive reasoning, it is necessary to assume, on some degree, some uniformity about the phenomenon being analyzed. Here, David Hume gives us an argument for which induction is valid, known as principle of uniformity [4],

> if Reason determin'd us, it would proceed upon that principle that instances, of which we have had no experience, must resemble those, of which we have had experience, and that the course of nature continues always uniformly the same. [86]

Going back to machine learning, one can make a parallel between the principle of uniformity, and the resemblance between training and test data. As a consequence, at the heart of standard learning techniques, such as risk minimization, all data must be generated by the same process, that is, they must come from the same probability distribution. This assumption, however, is seldom verified in practice. Take, for instance, the example shown in figure 4.1, where training data comes from two heterogeneous sources (internet and synthetic), and adaptation must be done towards images taken *in the wild*, that is, in an uncontrolled environment.

The rest of this chapter is organized as follows. Section 4.1 introduces the basics of generalization in machine learning under the empirical risk minimization framework. Section 4.2 presents the concepts behind transfer learning and domain adaptation. Section 4.3 discusses some mathematical results behind domain adaptation. Finally, section 4.4 presents strategies in domain adaptation, with a focus on optimal transport.

(a) Train data 1.

(b) Train data 2.

(c) Test data.

Figure 4.1 – Example of data being generated by different processes. For instance, in (a), one has product images coming from internet. In (b), one has CAD generated images. In (c), test data comes from *in the wild conditions*.

4.1 Empirical Risk Minimization

In this section, we introduce the tool that allows us to derive a general rule from finite examples. This tool is known as empirical risk minimization, and it was first presented by [5]. Our exposition is based on [87]. Before proceeding, let us establish some notions,

- **Input Space** is denoted by $\mathcal{X}$. It is the set of objects that we want to make predictions with. In principle, this space may be composed of complex objects (images, audio, text), but for the context of this section, we assume a vector representation, called feature vectors. We explore the construction of feature vectors from data in upcoming sections. As such, $\mathcal{X} \subset \mathbb{R}^d$.
- **Output Space** is denoted by $\mathcal{Y}$. It is the set of predictions made by the rule. In regression, $\mathcal{Y} = \mathbb{R}$. In this thesis, we focus on *classification*, i.e., $\mathcal{Y} = \{1, \cdots, n_c\}$.
- **Experience** or **training data**, is a finite set of examples $\{(x_i, y_i)\}_{i=1}^n$ from $\mathcal{X} \times \mathcal{Y}$.
- **Hypothesis** is, in the context of our previous discussion, the general rule $h : \mathcal{X} \to \mathcal{Y}$. We assume $h \in \mathcal{H}$, for a family of hypothesis $\mathcal{H}$. The algorithm that decides h is called *the learner*.
- **Data Generation Process.** We assume the existence of a ground-truth labeling function $h_0 : \mathcal{X} \to \mathcal{Y}$. Note that it may be that $h_0 \notin \mathcal{H}$. We further assume a probability measure Q for which $\mathbf{x}_i^{(Q)} \overset{i.i.d.}{\sim} Q$. The labels correspond to $y_i^{(Q)} = h_0(\mathbf{x}_i^{(Q)})$.

Given all of these concepts, the study of *statistical learning theory* revolves around the notion of risk between two hypothesis. Intuitively, the risk of $h \in \mathcal{H}$ with respect $h' \in \mathcal{H}$ corresponds to the expected disagreement between h and h' when predicting on samples $\mathbf{x}^{(Q)} \sim Q$. To measure the disagreement, one needs a loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to [0, +\infty]$.

Definition 19. *(True Risk) Let $h, h' \in \mathcal{Y}^{\mathcal{X}}$. The risk of h with respect h' and a distribution Q is,*

$$\mathcal{R}_Q(h, h') = \mathop{\mathbb{E}}_{\mathbf{x} \sim Q}[\mathcal{L}(h(\mathbf{x}), h'(\mathbf{x}))] = \int_{\mathcal{X}} \mathcal{L}(h(\mathbf{x}), h'(\mathbf{x})) dQ(\mathbf{x}), \tag{4.1}$$

Following [20], in learning theory one measures the risk of h with respect the ground-truth, i.e., h_0. In this case we adopt the short-hand $\mathcal{R}_Q(h) = \mathcal{R}_Q(h, h_0)$. Based on the true risk, one may be tempted to define an optimal hypothesis based on an optimization problem,

$$h^\star = \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{R}_Q(h), \tag{4.2}$$

but this requires explicit knowledge of Q and h_0. As we mentioned previously, one only has access to samples coming from Q, and labeled according to h_0. This leads to an empirical approximation of the expectation in equation 4.1.

Definition 20. *(Empirical Risk) Let $h : \mathcal{X} \to \mathcal{Y}$ be a hypothesis belonging to $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, and let $\{(\mathbf{x}_i^{(Q)}, y_i^{(Q)})\}_{i=1}^n$ be a finite sample with $\mathbf{x}_i^{(Q)} \overset{i.i.d.}{\sim} Q$ and $y_i^{(Q)} = h_0(\mathbf{x}_i^{(Q)})$. The* empirical risk *of h is,*

$$\hat{\mathcal{R}}_Q(h) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(h(\mathbf{x}_i^{(Q)}), y_i^{(Q)}). \tag{4.3}$$

Unlike the true risk, the empirical risk is subject to randomness with respect the choice of samples $\{(\mathbf{x}_i^{(Q)}, y_i^{(Q)})\}_{i=1}^n$ from Q. Furthermore, in analogy to equation 4.2, one can estimate the hypothesis h from finite data, by minimizing the empirical risk,

$$\hat{h} = \underset{h\in\mathcal{H}}{\operatorname{argmin}}\, \hat{\mathcal{R}}_Q(h) = \underset{h\in\mathcal{H}}{\operatorname{argmin}}\, \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(h(\mathbf{x}_i^{(Q)}), y_i^{(Q)}). \tag{4.4}$$

In the next example, we illustrate these concepts, and give some intuition about how $\hat{\mathcal{R}}_Q$ converges to $\mathcal{R}_Q$ when an increasing number of samples is available.

Example 7. *In this example we consider a binary classification problem, i.e., $\mathcal{Y} = \{0, 1\}$. We assume the measure Q is given by,*

$$Q(\mathbf{x}) = 0.5Q(\mathbf{x}|y = 0) + 0.5Q(\mathbf{x}|y = 1), \tag{4.5}$$

where $Q(\mathbf{x}|y = 0) = \mathcal{N}(\mu_0, \Sigma_0)$ and $Q(\mathbf{x}|y = 1) = \mathcal{N}(\mu_1, \Sigma_1)$. In this example,

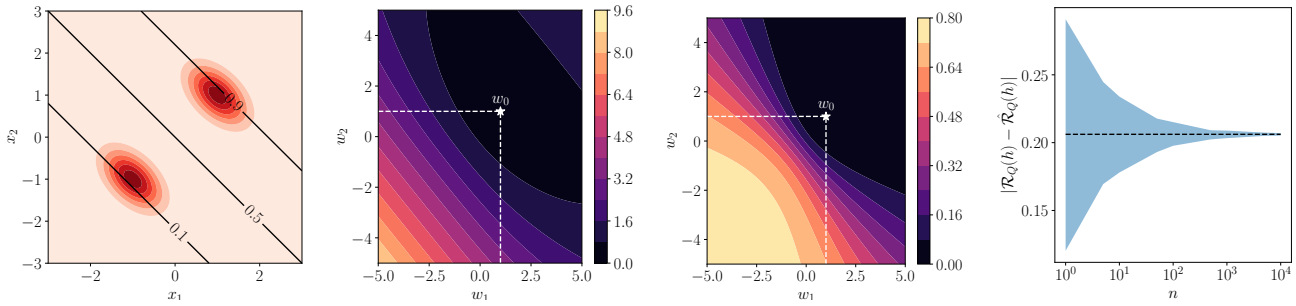
$$\Sigma_0 = \begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.2 \end{bmatrix}, \mu_0 = \begin{bmatrix} -1. & -1. \end{bmatrix}, \Sigma_1 = \begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.2 \end{bmatrix}, \text{ and } \mu_1 = \begin{bmatrix} 1. & 1. \end{bmatrix}. \tag{4.6}$$

We show the underlying measure Q, and h_0 in Figure 4.2 (a). We assume that $h \in \mathcal{H}$ is parametrized by vectors $\mathbf{w} \in \mathbb{R}^2$, i.e., $\mathcal{H} = \left\{ h(\mathbf{x}) = \dfrac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}} : \mathbf{w} \in \mathbb{R}^2 \right\}$, which is a parametrization for linear classifiers. Especifically, since $h(\mathbf{x}) \in [0, 1]$, we can understand this prediction as the probability of $\mathbf{x}$ having label 1. Note that this choice of parametrization allows us assign a classifier to each point in $\mathbb{R}^2$. As a result, we show in Figures 4.2 (b) and (c), the true risk for two choices of losses. Note that, here, we are exploiting the encoding of classifiers. The plot actually shows $\mathbf{w} \mapsto \mathcal{R}_Q(h_\mathbf{w})$.

Next, given $h \in \mathcal{H}$, parametrized by $\mathbf{w}$, we explore how well $\hat{\mathcal{R}}_Q(h)$ approximates $\mathcal{R}_Q(h)$ as a function of the number of samples n drawn from Q. To do so, given n, we sample $\mathbf{X}^{(Q)} = [\mathbf{x}_1^{(Q)}, \cdots, \mathbf{x}_n^{(Q)}]$, where $\mathbf{x}_i^{(Q)} \overset{i.i.d.}{\sim} Q$, then we label $y_i^{(Q)} = h_0(\mathbf{x}_i^{(Q)})$. We then evaluate equation 4.3 using the acquired samples. Since the estimated quantity is a random variable that depends on the samples, we repeat this process over 20 times from which we calculate the mean and standard deviation of $\hat{\mathcal{R}}_Q(h)$. This experiment is shown in figure 4.2 (d).

This last example should convice the reader that, in order to evaluate the true risk, one needs additional information about the closed-form of the underlying probability measure, and the ground-truth labeling function. As we highlighted previously, this assumption is not feasible in practice. Hence the need for Empirical Risk Minimization (ERM).

In the following, we describe the Vapnik-Chervonenkis (VC) framework of [5, 6]. This framework considers the convergence of the empirical risk towards the true risk from the perspect of the *worst-case*



(a) Q alongside h_0 (b) BCE(y, y') (c) $|y - y'|^2$ (d) Convergence of $\hat{\mathcal{R}}_Q$.

Figure 4.2 – Binary classification problem on a mixture of two Gaussian measures for $\mathbf{w}_0 = \begin{bmatrix} 1 & 1 \end{bmatrix}$ (a). Since $\mathcal{H}$ is parametrized by $\mathbf{w} = \begin{bmatrix} w_1 & w_2 \end{bmatrix}$, we plot on (b) and (c) the risk of hypothesis in $\mathcal{H}$ for two different choices of loss functions $\mathcal{L}$. In (d), we show the convergence of the empirical risk as $n \to \infty$.

scenario, i.e., the worst hypothesis $h \in \mathcal{H}$. As such, it is often criticized from its pessimism, and the fact that the derived bounds are not dependent on the data at hand. While since the proposition of this framework other ideas have emerged (see [22] for a recent review), we decide to present its concepts, as they often play a prominent role in the proofs associated with domain adaptation.

The cornerstone of VC theory is the concept of VC-dimension of a family of hypothesis $\mathcal{H}$. This dimension measures the complexity of the classifiers $h \in \mathcal{H}$. As the next definition shows, this notion is combinatorial in nature, as the VC-dimension is identified as *the maximum number of points that $h \in \mathcal{H}$ can perfectly classify.*

Definition 21. *(VC-Dimension) The VC dimension $VC(\mathcal{H})$ of a given hypothesis class $\mathcal{H}$ for the problem of binary classification is defined as the largest possible cardinality of some subset $\mathcal{X}' \subset \mathcal{X}$ for which there exists $h \in \mathcal{H}$ that perfectly classifier elements from $\mathcal{X}'$ whatever their labels are, i.e.,*

$$VC(\mathcal{H}) = max\{|\mathcal{X}'| : \forall y_i \in \{-1, +1\}^{|\mathcal{X}'|}, \exists h \in \mathcal{H} \text{ so that } \forall \mathbf{x}_i \in \mathcal{X}', h(\mathbf{x}_i) = y_i\}.$$

For instance, the VC-Dimension of the set of linear classifiers in $\mathbb{R}^d$ is $d - 1$ [21, Section 1.2.1., Figure 1.3]. However, even simple classifiers such as nearest neighbors have an infinite VC-dimension, which makes the learning bounds here presented vacuous (i.e., they hold trivially).

Theorem 3. *Let $\mathcal{Y} = \{-1, +1\}$ be the output space, $Q \in \mathbb{P}(\mathcal{X})$, and h_0 be a ground-truth labeling function. Let $(\mathbf{X}^{(Q)}, \mathbf{Y}^{(Q)})$ be a finite sample of size n, i.i.d. from Q, such that $y_i^{(Q)} = h_0(\mathbf{x}_i^{(Q)})$. Let $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class of VC-dimension $VC(\mathcal{H})$. For any $\delta \in (0, 1]$, with probability at least $1 - \delta$ over the choice of samples $\mathbf{X}^{(Q)} \sim (Q)^n$, the following holds,*

$$\forall h \in \mathcal{H},\ \mathcal{R}_Q(h) \leq \hat{\mathcal{R}}_Q(h) + \underbrace{\sqrt{\frac{4}{n}\left(VC(\mathcal{H}) \log \frac{2en}{VC(\mathcal{H})} + \log \frac{4}{\delta}\right)}}_{\mathcal{C}_{ERM}(n,\delta,\mathcal{H})}, \tag{4.7}$$

where $\mathcal{C}_{ERM}(n, \delta, \mathcal{H})$ denotes the sample *complexity of ERM.*

This result provides a statistical framework for classification. Indeed, it guarantees that the empirical risk is close to the true risk, plus an error term $\mathcal{O}(n^{-1/2}\log n)$. As we discussed in the context of corollary 1 in Chapter 2, terms depending on the number of samples n are referred to as *statistical complexity*, that is, the number of samples needed to accurately estimate a term.

4.2 Domain Adaptation and its Cousins

In the previous section, we described a formal framework for generalization to unseen samples. These results were acquired under the assumption that new data comes from the same measure as the training data. Unfortunately, such an hypothesis is restrictive. As we discussed in the beginning of this chapter, the *principle of uniformity* may not hold, that is, nature may change. Nonetheless, it is desirably that a machine learning model is still able to adapt to such cases, as humans are. Motivated by these ideas, the field of transfer learning emerged as a sub-field of machine learning.

Transfer learning works under the assumption that *something changes*. The first formalization for this idea was laid out by [8], who introduced the concepts of domain and task.

Definition 22. *(Domains and Tasks) A domain is a pair $(\mathcal{X}, Q)$ of a feature space $\mathcal{X}$, and a feature marginal measure $Q \in \mathbb{P}(\mathcal{X})$. A task is a pair $(\mathcal{Y}, h(\cdot))$ of a label space $\mathcal{Y}$ and a ground-truth predictive function $h : \mathcal{X} \to \mathcal{Y}$.*

With the definitions of domains and tasks at hand, one can view transfer learning as a generalization problem under a domain or task shift.

Definition 23. *(Transfer Learning) Given a source domain $(\mathcal{X}_S, Q_S)$, a source task $(\mathcal{Y}_S, h_S(\cdot))$, a target domain $(\mathcal{X}_T, Q_T)$ and a target task $(\mathcal{Y}_T, h_T(\cdot))$, transfer learning aims to help improving the learning of h_T given knowledge from the source, when either $(\mathcal{X}_S, Q_S) \neq (\mathcal{X}_T, Q_T)$ or $(\mathcal{Y}_S, h_S(\cdot)) \neq (\mathcal{Y}_T, h_T(\cdot))$.*

Different problems fall under the transfer learning definition, such as multi-task learning [88] and domain adaptation. This thesis is particularly interested in the latter, which we now present.

Domain adaptation is concerned with learning problems in which domains differ, i.e., $(\mathcal{X}_S, Q_S) \neq (\mathcal{X}_T, Q_T)$. Naturally, two cases can occur: (i) $\mathcal{X}_S \neq \mathcal{X}_T$, known as *heterogeneous domain adaptation* [89] and (ii) $Q_S \neq Q_T$, known as *covariate shift* [90]. In the context of this thesis, we assume the second scenario, i.e., we fix $\mathcal{X}_S = \mathcal{X}_T = \mathbb{R}^d$. A few supplementary hypothesis are made. First, one assumes the same task, i.e., $\mathcal{Y}_S = \mathcal{Y}_T$ and $h_S = h_T$. Second, we assume that *unlabeled data* is available in the target domain. This sets up the *unsupervised domain adaptation* scenario.

Definition 24. *(Unsupervised Domain Adaptation) Given a source $(\mathcal{X}, Q_S)$ and a target $(\mathcal{X}, Q_T)$ domain, and a task $(\mathcal{Y}, h_0(\cdot))$,* unsupervised domain adaptation *seeks to improve the performance of $(\mathcal{Y}, h_0(\cdot))$ on $(\mathcal{X}, Q_T)$, labeled samples from the source domain, and unlabeled samples from the target domain.*

Let us briefly discuss the 3 previous hypothesis. First, $\mathcal{X}_S = \mathcal{X}_T$ is relatively straight-forward. For instance, one can think of a fixed set of features being extracted from images. Indeed, heterogeneous domain adaptation is a relatively specific application [91, 89]. Second, to assume $\mathcal{Y}_S = \mathcal{Y}_T$ is a

standard assumption in domain adaptation, but it can be relaxed with $\mathcal{Y}_T \subset \mathcal{Y}_S$ (partial domain adaptation [92]) and $\mathcal{Y}_S \subset \mathcal{Y}_T$ (open set domain adaptation [93]). Third the unsupervised setting enhances the practicality of domain adaptation, as the target domain is associated with the test set. In some settings, such as cross-domain fault diagnosis [94], acquiring labeled target domain data may be dangerous or expensive. This assumption further requires $h_S = h_T$, as one needs to adapt to the target domain with only information about $Q_T(X)$.

Since we assume $\mathcal{X} = \mathbb{R}^d$ fixed, we characterize domain adaptation as a *distributional shift* phenomenon, i.e., $Q_S \neq Q_T$. With this concept at hand, one can further distinguish 2 two kinds of domain adaptation. The first, *single-source* domain adaptation, assumes source data is distributed according a single P. The second, *multi-source* domain adaptation, assumes a family of probability distributions $\{Q_{S_\ell}\}_{\ell=1}^{N_S}$. While the main contributions of this thesis are in multi-source domain adaptation, we also propose, in Chapter 7, a new method for single-source domain adaptation.

The multi-source case arises naturally, especially when learning from heterogeneous databases. For instance, as the pioneer work of [95] discusses, one can think of learning the preferences of different users in the web, i.e., a classifier that predicts what sites an user might find interesting. On the one hand, one can train a classifier for each individual user, thus having a model *personalized* to each users' data. On the other hand, one can harness data from similar users to improve the model for a given user. In a more extreme case, one can think of a target user whose preferences where not yet labeled. Even though such an user cannot learn a model with their own data, they can use the data or models of other users, especially those similar theirs, to build a classifier.

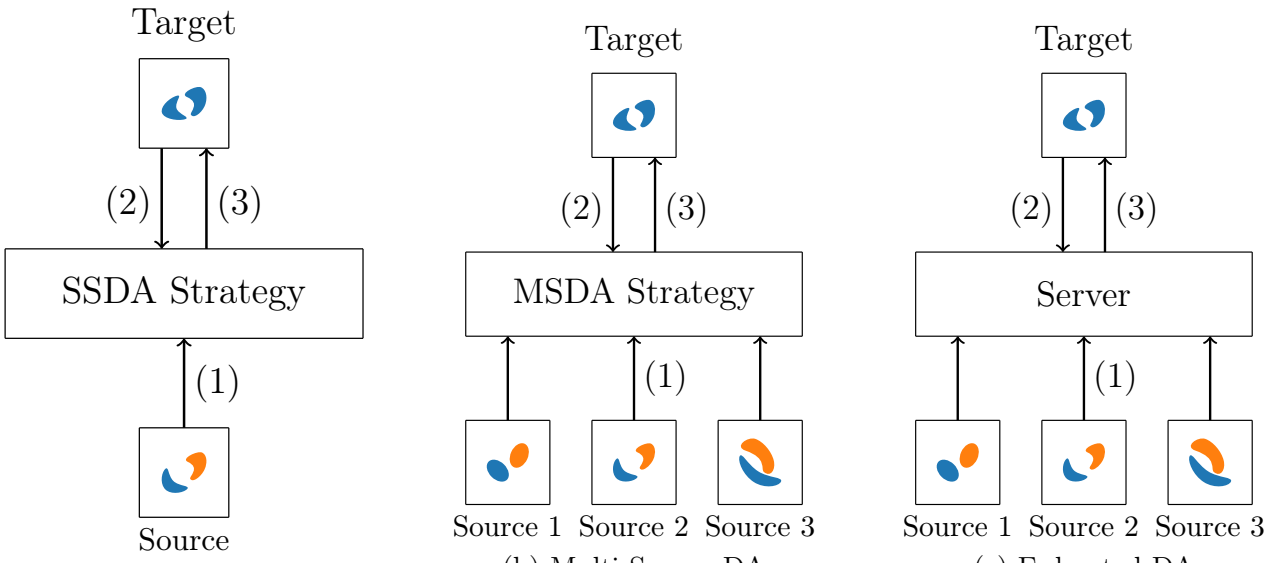
A similar case in which it is necessary to consider multiple probability distributions is *federated* domain adaptation, first proposed in [96]. In this setting, data is supposed to be held by multiple clients, which do not want to share nor centralize their data for training. This problem is a particular case of the non-i.i.d. federated learning setting [97]. An illustration of both multi-source and federated domain adaptation is shown in figure 4.3. We further discuss this setting in the future works in Chapter 11

4.3 Domain Adaptation Theory

This section serves as a brief introduction on the theoretical analysis of DA, especially through OT. The idea of most theoretical results in DA is to bound the error under the target domain measure, by the error under the source domain measure. If the bound is tight, one may determine a classifier that works on the target domain by minimizing the risk on the source domain. Intuitively, these bounds should be tight if the two measures are close.

Here, chapters 2 and 3 come into play, as one uses probability metrics to estimate domain differences, thus formalizing the notion of *closeness*. Two additional constraints are also relevant, since we are dealing with *unsupervised* domain adaptation and we do not know the actual probability measures that generate the data. First, one needs to bound the risks under the two measures without access to its labels. Second, one must be able to accurately estimate the distance between distributions from finite samples.

While different papers established theoretical results in DA, [20] is a landmark paper that puts together many of the different results. A first bound in DA may be derived using the total variation



(a) Single Source DA (b) Multi-Source DA (c) Federated DA

Figure 4.3 – Three settings of domain adaptation. In (a), (1), (2) and (3) correspond to labeled source domain data, unlabeled source domain data and the resulting, adapted classifier. In (b), the source domain data in (1) comes from multiple sources, each with its own distribution. In (c), each domain is understood as a client. Adaptation is coordinate by a server, and client data is not explicitly communicated throughout adaptation.

$\mathcal{D}_{TV}(Q_S, Q_T) = 2\sup_{B\in\mathcal{B}}|Q_S(B) - Q_T(B)|$, where $\mathcal{B}$ is the set of measurable sets under Q_S and Q_T. As discussed in [98] and [99], estimating the total variation from finite samples is in most cases impossible, since the number of samples grows exponentially with the desired precision.

This limitation motivated [99] to consider restricting the set $\mathcal{B}$ to *sets that the user care about*. This intuition ultimately led [20] to propose the $\mathcal{H}-$distance, which we previously discussed in chapter 3. Compared to the total variation, $\mathcal{D}_{\mathcal{H}}$ can be estimated from finite samples, and it has a simple empirical estimator (e.g., Lemma 1). In this context, we present a first theoretical result bounding the disagreement between two hypothesis in terms of this probability metric. In the following, we denote $\mathcal{R}_S = \mathcal{R}_{Q_S}$ (resp. T) in short.

Lemma 4. *For any two hypothesis $h, h' \in \mathcal{H}$,*

$$|\mathcal{R}_T(h, h') - \mathcal{R}_S(h, h')| \leq \frac{1}{2}\mathcal{D}_{\mathcal{H}}(Q_S, Q_T).$$

The proof of this lemma is a straightforward application of definitions 19 and 13, thus highlighting the convenience of restricting the set upon which one computes the total variation. This bound formalizes the intuition that, as $Q_S \to Q_T$ under $\mathcal{D}_{\mathcal{H}}$, the $\mathcal{R}_S(h, h') \to \mathcal{R}_T(h, h')$. Based on the previous bound, [20] derives a fundamental theorem for domain adaptation,

Theorem 4. *Let $\mathcal{H}$ be a hypothesis space with finite VC-dimension $VC(\mathcal{H}) < \infty$. Let $\mathbf{X}^{(Q_S)}$ and $\mathbf{X}^{(Q_T)}$ be two unlabeled samples of size n and m drawn from Q_S and Q_T respectively, then for any*

$\delta(0,1)$, *with probability at least* $1-\delta$ *(over the choice of the samples), for every* $h \in \mathcal{H}$,

$$\mathcal{R}_T(h) \leq \hat{\mathcal{R}}_S(h) + \mathcal{C}_{ERM}(n,\delta,\mathcal{H}) + \frac{1}{2}\mathcal{D}_{\mathcal{H}}(\hat{Q}_S,\hat{Q}_T) + \mathcal{C}_{\mathcal{H}}(n,\delta) + \underbrace{\min_{h\in\mathcal{H}} \mathcal{R}_T(h) + \mathcal{R}_S(h)}_{\mathcal{C}_{DA}(Q_S,Q_T)}, \tag{4.8}$$

where $\mathcal{C}_{\mathcal{H}}$ *is the sample complexity of estimating* $\mathcal{D}_{\mathcal{H}}$ *(c.f., lemma 2), and* $\mathcal{C}_{DA}(Q_S,Q_T)$ *refers to the* domain adaptation complexity.

With respect the ERM bound in equation 4.7, one adds three terms: (i) how close P and Q are, (ii) how complex it is to estimate $\mathcal{D}$ and (iii) how informative P is to Q. We now present two additional results, one for the MMD (section 3.1.2) and the $1-$Wasserstein distance. Furthermore, note that, in classification, one minimizes $\hat{\mathcal{R}}_S$, i.e., the risk over labeled data. The term $\mathcal{D}_{\mathcal{H}}$ is minimized through DA. The remaining terms (i.e., $\mathcal{C}_{ERM}$, $\mathcal{C}_{\mathcal{H}}$ and $\mathcal{C}_{DA}$) are approximation errors or the instrinsic difficulty of performing DA, thus, these cannot be explicitly minimized.

Lemma 5. *Let* $\mathcal{F} = \{h \in \mathcal{H}_k : \|h\|_{\mathcal{H}_k} \leq 1\}$, *where* $\mathcal{H}_k$ *is a RKHS with its associated kernel* k. *Let* $\mathcal{L}(h(\mathbf{x}), h'(\mathbf{x}))$ *be a convex loss function with parametric form* $|h(\mathbf{x}) - h'(\mathbf{x})|^q$, *for some* $q > 0$ *and defined* $\forall h, h' \in \mathcal{F}$ *such that* $\mathcal{L}$ *obeys the triangle inequality. Then, if* $\|\mathcal{L}\|_{\mathcal{H}_{k^q}} \leq 1$,

$$\mathcal{R}_T(h,h') \leq \mathcal{R}_S(h,h') + MMD(Q_S,Q_T).$$

With this result, [21] derives a bound similar to that in equation 4.8 with the MMD,

Theorem 5. *With the same assumptions from Lemma 4, let* $\mathbf{X}^{(Q_S)}$ *and* $\mathbf{X}^{(Q_T)}$ *be two samples of size* n *and* m, *drawn i.i.d. from* Q_S *and* Q_T, *respectively. Then, for* $\delta \in (0,1)$, *with probability at least* $1-\delta$ *(over the choice of the samples), the following holds,*

$$\mathcal{R}_T(h) \leq \hat{\mathcal{R}}_S(h) + \mathcal{C}_{ERM}(n,\delta,\mathcal{H}) + MMD(\hat{Q}_S,\hat{Q}_T) + \mathcal{C}_{MMD}(n,\delta,k) + \mathcal{C}_{DA}(Q_S,Q_T).$$

where $\mathcal{C}_{MMD}(n,\delta,k)$ *denotes the complexity of estimating the MMD empirically (c.f., lemma 3), and it depends on the number of samples* n, *the precision* δ *and the kernel* k.

Now, for the Wasserstein distance, [100] uses RKHS theory, that is, the authors choose a ground-cost that comes from a kernel. As the authors comment, while this may appear restrictive, the theoretical result of [101, Lemma 12] guarantee an equivalence between positive semi-definite kernels and distances.

Lemma 6. *Let* Q_S *and* Q_T *be two probability distributions over* $\mathbb{R}^d$. *Assume that the cost function* $c(\mathbf{x}^{(Q_S)}, \mathbf{x}^{(Q_T)}) = \|\phi(\mathbf{x}^{(Q_S)}) - \phi(\mathbf{x}^{(Q_T)})\|_{\mathcal{F}}$, *where* $\mathcal{F}$ *is a RKHS equipped with kernel* $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ *induced by* $\phi : \mathbb{R}^d \to \mathcal{F}$ *and* $k(\mathbf{x}^{(Q_S)}, \mathbf{x}^{(Q_T)}) = \langle \phi(\mathbf{x}^{(Q_S)}), \phi(\mathbf{x}^{(Q_T)}) \rangle_{\mathcal{F}}$. *Assume that the kernel* $k \in \mathcal{F}$ *is square-root integrable w.r.t. both* Q_S *and* Q_T *and* $0 \leq k(\mathbf{x}^{(Q_S)}, \mathbf{x}^{(Q_T)}) \leq M$, $\forall \mathbf{x}^{(Q_S)}, \mathbf{x}^{(Q_T)} \in \mathbb{R}^d$. *Then the following holds,*

$$\mathcal{R}_T(h,h') \leq \mathcal{R}_S(h,h') + \mathcal{W}_1(Q_S,Q_T). \tag{4.9}$$

Based on the bound for the disagreement between two hypothesis, one has the main theoretical result in optimal transport for domain adaptation.

Theorem 6. *Let $\mathbf{X}^{(Q_S)} \in \mathbb{R}^{n\times d}$ and $\mathbf{X}^{(Q_T)} \in \mathbb{R}^{m\times d}$ be i.i.d. samples from Q_S and Q_T. Then, for any $d' > d$ and $\xi' < \sqrt{2}$ there exists some constant n_0 depending on d' s.t. for $\delta \in (0,1)$ and $min(n_P, n_Q) \geq n_0 max(\delta^{-(d+2)}, 1)$ with probability at least $1-\delta$ for all h,*

$$\mathcal{R}_T(h) \leq \hat{\mathcal{R}}_S(h) + \mathcal{C}_{ERM}(n,\delta,\mathcal{H}) + \mathcal{W}_1(\hat{P},\hat{Q}) + \mathcal{C}_{OT}(n,\delta) + \mathcal{C}_{DA}(Q_S,Q_T), \tag{4.10}$$

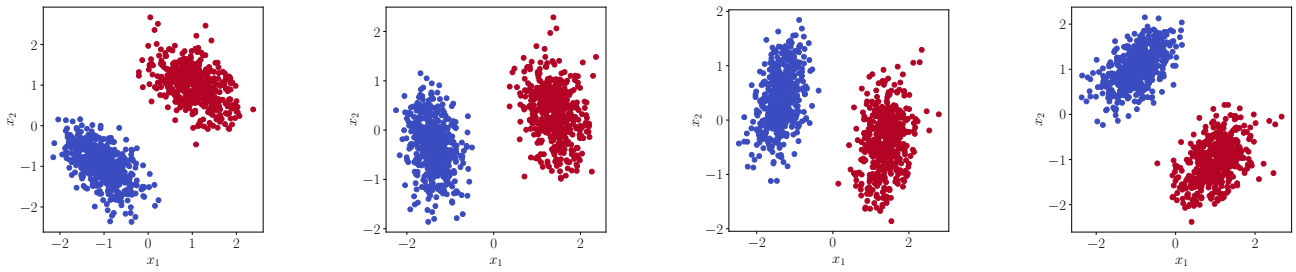
where $\mathcal{C}_{OT}(n,\delta)$ is the sample complexity of estimating $\mathcal{W}_1(P,Q)$ (equation 2.22).

In the next example, we illustrate the Wasserstein-based DA bound and the possible difficulties of the *unsupervised* DA problem.

Example 8. *In this example, we re-use the distribution Q described in example 7 and equations 4.5 and 4.6. In particular, we let $Q_S := Q$. Next, we create an artificial distribution shift via a rotation matrix,*

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \text{ and, } \hat{Q}_{T,\theta} = T_{\theta,\sharp}\hat{Q}_S,$$

where $T_\theta(\mathbf{x}) = \mathbf{R}_\theta\mathbf{x}$. This corresponds to rotating each sample in $\hat{Q}_S$, as shown in figure 4.4.



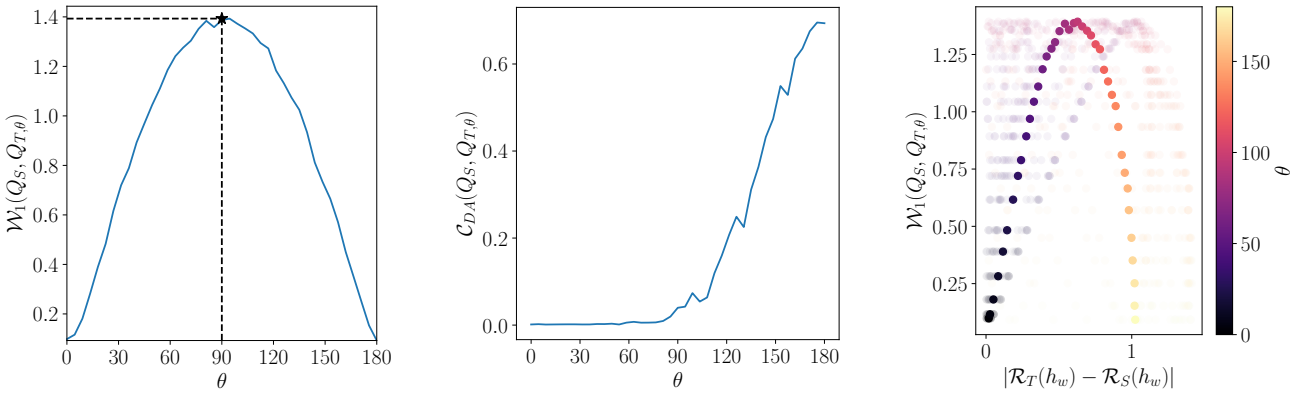
(a) $\theta = 0$. (b) $\theta = 30$. (c) $\theta = 60$. (d) $\theta = 90$.

Figure 4.4 – Rotated distributions for domain adaptation analysis.

This example is interesting for two aspects. First, the domain adaptation difficulty increases with θ. Indeed, for small rotations, it is likely that a classifier fit with source data is able to generalize to the new distribution. Second, without the labels, due to the symmetry of the class clusters, shifts for $\theta > 90$ are equivalent to those of $\theta' = 90 - \theta$. However, there is an inversion in the labels of the two domains. This means that, while there is a decrease in the Wasserstein distance, the difference $\mathcal{R}_T(h) - \mathcal{R}_S(h)$ continues to grow. This phenomenon illustrates why one needs to take into account $\mathcal{C}_{DA}(Q_S, Q_{T,\theta})$ besides the discrepancy between marginals into the DA bounds.

4.3.1 Multi-Source Domain Adaptation

Besides the standard theory of Multi-Source DA (MSDA) covered throughout this section, a special care must be given to the case of multiple sources. Especially, one needs to consider the possible combinations of sources, possibly with a weighting strategy over the domains, in the composition of the empirical risk. Arguably, a pioneering work in this regard is [102], who considered a learning



(a) (b) (c)

Figure 4.5 – In (a), we plot the 1−Wasserstein distance against the rotation parameter θ. Due to the symmetry of the adaptation problem, for $\theta > 90$, there is a decrease in the Wasserstein distance between source and target. This, however, is not reflected in the DA complexity (b), which grows as there is an inversion in the labels from source to target for $\theta > 90$. In (c), we show the same pattern for $\mathbf{w} \sim \mathbb{S}_2$ (shadowed dots). The solid scatter plot corresponds to $\mathbb{E}_{\mathbf{w}\sim\mathbb{S}_2}[|\mathcal{R}_T(h_w) - \mathcal{R}_S(h_w)|]$, which exhibits the same pattern as (a) and (b).

problem in which the target domain was assumed to be a linear combination of the source domains. In terms of densities, these authors assumed,

$$q_T(x) = \sum_{k=1}^{K} \lambda_k q_{S_k}(x).$$

In this case, as the authors show, a natural weighting scheme for DA is

$$h_T(\mathbf{x}) = \sum_{k=1}^{K} \left(\frac{q_{S_k}(\mathbf{x})}{\sum_{k'=1}^{K} q_{S_k}(\mathbf{x})} \right) h_k(\mathbf{x}).$$

Note, however, that this would imply knowing the densities q_{S_k}, which is not feasible in the context of this thesis due the high dimensionality of the input space. We thus turn our attention on the risk under the target domain of combining classifiers fit on each sources.

Definition 25. *(Combined Sources Risk) Let $K \geq 1$ be K source domains, with probability measures $\mathcal{Q}_S = \{Q_{S_k}\}_{k=1}^{K}$. Let $\lambda \in \Delta_K$ be weights, where λ_k denotes the weight of the $k-th$ domain with respect the others. The $\lambda-$weighted risk of $h \in \mathcal{H}$ is given by,*

$$\mathcal{R}_\lambda(h) = \sum_{k=1}^{K} \lambda_k \mathcal{R}_{Q_{S_k}}(h).$$

Given samples $\{\{\mathbf{x}_i^{(Q_{S_k})}, y_i^{(Q_{S_k})}\}_{i=1}^{n_{S_k}}\}_{k=1}^{K}$, we have an associated notion of $\lambda-$weighted empirical risk,

$$\hat{\mathcal{R}}_\lambda(h) = \sum_{k=1}^{K} \frac{\lambda_k}{n_{S_k}} \sum_{i=1}^{n_{S_k}} \mathcal{L}(h(\mathbf{x}_i^{(Q_{S_k})}), y_i^{(Q_{S_k})}).$$

Based on this notion of combined error, [20] proved a generalization error for the hypothesis learned by minimizing $\hat{h} = \text{argmin}_{h\in\mathcal{H}}\hat{\mathcal{R}}(h)$, which we now state.

Theorem 7. *Let $\{\{\mathbf{x}_i^{(Q_{S_k})}, y_i^{(Q_{S_k})}\}_{i=1}^{n_{S_k}}\}_{k=1}^{K}$ be i.i.d. samples from $Q_{S_k} \in \mathcal{Q}_S$. Let $n = \sum_k n_{S_k}$, and $\rho_k = {}^{n_{S_k}}/_{n}$. Let $\hat{h}_k = \text{argmin}_{h\in\mathcal{H}}\hat{\mathcal{R}}_{S_k}(h)$ be the empirical minimizer of the $\lambda-$weighted source domains, and let $h_T^\star = \text{argmin}_{h\in\mathcal{H}}\mathcal{R}_{Q_T}(h)$ be the target risk minimizer. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\mathcal{R}_{Q_T}(\hat{h}) \leq \mathcal{R}_{Q_T}(h_T) + \mathcal{C}_{MS-\mathcal{H}}(\lambda,\rho,\mathcal{H},\delta) + \sum_{k=1}^{K}\lambda_k(\mathcal{C}_{DA}(Q_{S_k},Q_T) + \mathcal{D}_{\mathcal{H}}(Q_{S_k},Q_T)), \tag{4.11}$$

where $\mathcal{C}_{MS-\mathcal{H}}$ is a complexity term associated with MSDA,

$$\mathcal{C}_{MS-\mathcal{H}}(\lambda,\rho,\mathcal{H},\delta) = 2\sqrt{\left(\sum_{k=1}^{K}\frac{\alpha_k^2}{\beta_k}\right)\left(\frac{VC(\mathcal{H})\log(2n) - \log(\delta)}{2n}\right)}$$

An interesting thing about the bound in equation 4.11 is that it involves a new complexity term, corresponding to how the sources are weighted through λ, and the proportion of samples from each domain, given by ρ. As the next result, as [100] shows, a similar scenario occurs when bounding these terms using OT,

Theorem 8. *(Due to [100]) Let $\{\{\mathbf{x}_i^{(Q_{S_k})}, y_i^{(Q_{S_k})}\}_{i=1}^{n_{S_k}}\}_{k=1}^{K}$ be i.i.d. samples from $Q_{S_k} \in \mathcal{Q}_S$. Let $n = \sum_k n_{S_k}$, and $\{\mathbf{x}_i^{(Q_T)}\}_{i=1}^{n_T}$, be i.i.d. samples from Q_T. Let $\hat{Q}_{S_k}$ and $\hat{Q}_T$ be their respective empirical measures. Let $\rho_k = {}^{n_{S_k}}/_{n}$. If $\hat{h}_\lambda$ is the empirical minimizer of $\hat{\mathcal{R}}_\lambda$ and $h_T^\star$ is the minimizer of $\mathcal{R}_{Q_T}$, then for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\mathcal{R}_{Q_T}(\hat{h}_\lambda) \leq \mathcal{R}_{Q_T}(h_T^\star) + \mathcal{C}_{MS-OT}(n,\lambda,\rho,\delta) + 2\sum_{\ell=1}^{N_S}\lambda_\ell(\mathcal{W}_1(\hat{Q}_{S_\ell},\hat{Q}_T) + \mathcal{C}_{DA}(Q_{S_\ell},Q_T) + \mathcal{C}_{OT}(n,\delta)), \tag{4.12}$$

where $\mathcal{C}_{MS-OT}(n,\lambda,\rho,\delta)$ is the following sample complexity,

$$\mathcal{C}_{MS-OT}(n,\lambda,\rho,\delta) = 2\sqrt{\frac{2M\sum_{\ell=1}^{N}{}^{\lambda_\ell}/_{\rho_\ell}\log(2/\delta)}{n}} + 2\sqrt{\sum_{\ell=1}^{N_S}\frac{M\lambda_\ell}{\rho_\ell}}.$$

where M is defined as in Lemma 6.

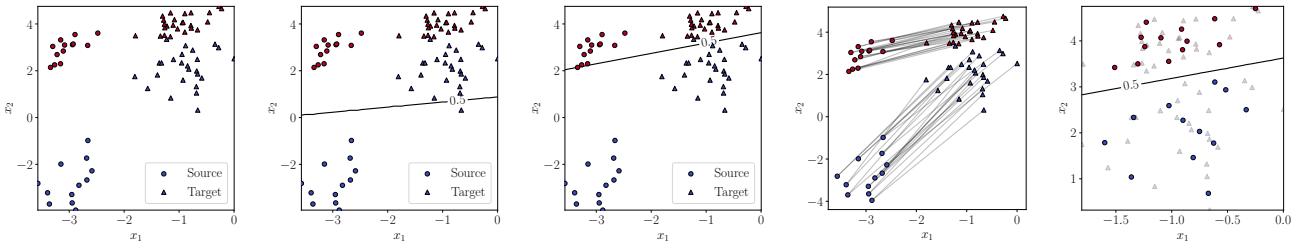
4.4 Domain Adaptation Practice

In this section, we discuss some algorithmic ideas in domain adaptation. The list here presented is far from exhaustive. We focus on two principles: distribution matching, and importance weighting. These two concepts are behind the main baselines related to our contributions in Part II. The motivation for matching *source* and *target* measures comes from the bound in equation 4.10. Assuming the target

domain distribution fixed, one may try to minimize the right-hand-side of the aforementioned equation by minimizing a probability metric with respect to $\hat{Q}_S$. Naturally, there are many ways of doing so, both in terms of *how* to minimize $\hat{Q} \mapsto \mathcal{D}(\hat{Q}, \hat{Q}_T)$, and in terms of *which* metric $\mathcal{D}$ to choose. In the following, we focus on the 2−Wasserstein distance.

4.4.1 Barycentric Mapping

The first OT-inspired domain adaptation algorithm was proposed in the seminal works of [103, 23], and is generally called OTDA. The choice $\mathcal{D} = \mathcal{W}_2$ leads to a principled approach, in which source domain data is transformed under a least effort principle, that is, one seeks for a minimal transformation that matches both measures. The OTDA strategy starts with the assumption that distributional shift is caused by an unknown, possibly non-linear transformation acting on $\mathcal{X}$, i.e., $T : \mathcal{X} \to \mathcal{X}$. In real-case scenarios, such a transformation may have a physical interpretation. For instance in [104], T is caused by a change in the chemical reactions inside a reactor. One further assumes that T preserves the conditional distributions, i.e., $Q_S(Y|X) = Q_T(Y|T(X))$, which guarantees that the label information from samples is preserved under T. This hypothesis will play an important role in the method, as we describe in the following.



(a) Source and Target samples. (b) Source Classifier. (c) Target Classifier. (d) Source and Target match. (e) Transported Samples.

Figure 4.6 – In (a), source (circles) and target (triangles) distributions differ in their marginal. In (b) and (c) we show the classifiers $\hat{h}_S$ and $\hat{h}_T$, estimated with source and target domain data respectively. In (d) and (e), the optimal transport solution between points in the two domains using feature information only. In (f), the transported source domain points carry their labels to the target domain, so a classifier can be fit in this domain.

At this point, there may exist many transformations such that, when applied to samples of Q_S, provide samples of Q_T. Indeed, the set of admissible transformations is exactly those such that $T_\sharp Q_S = Q_T$, as described by the Monge formulation of OT in chapter 2. The motivation for a least effort map between these distributions effectively leads to the Monge problem. Here, a major limitation comes, as one usually has samples $\mathbf{X}^{(Q_S)}$ and $\mathbf{X}^{(Q_T)}$, rather than a closed form for these distributions. As a result, without further heuristics or assumptions, one cannot estimate T through equation 2.17.

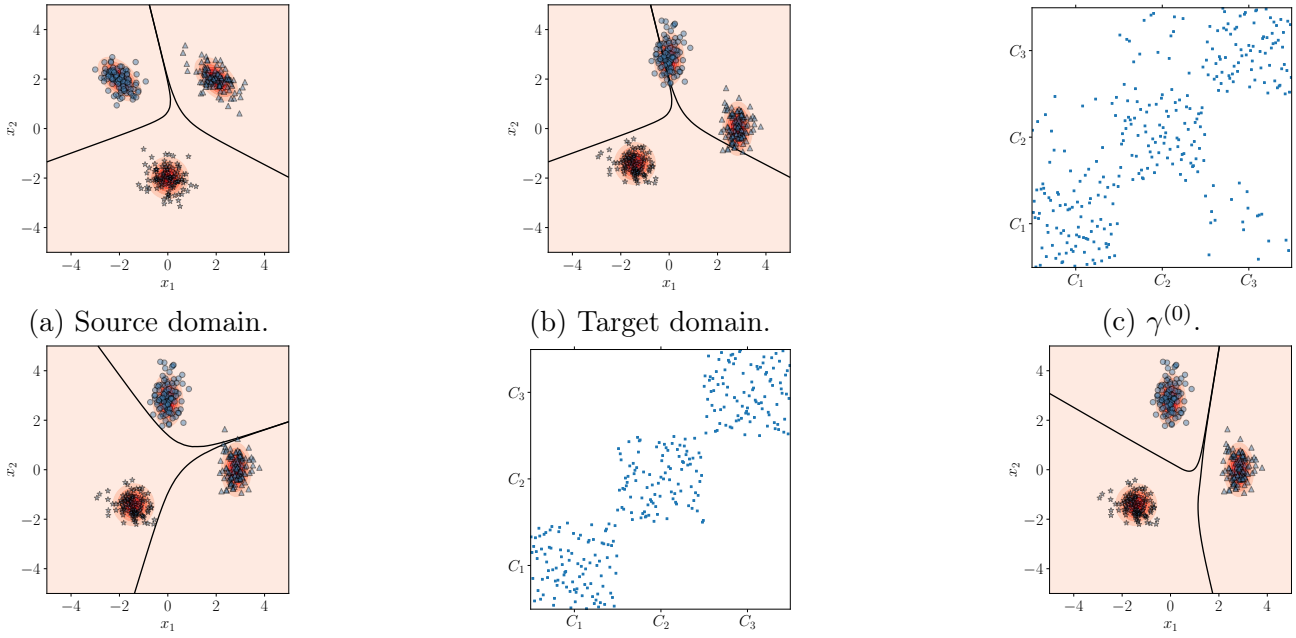
A workaround comes from the barycentric map (equation 2.19), which provides an empirical approximation for the Monge map based on the optimal transport plan $\gamma^\star$. This leads to the following strategy: (i) Estimate $\gamma \in \mathbb{R}^{n \times m}$ between Q_S and Q_T; (ii) Construct a novel dataset $\{T_{\gamma^\star}(\mathbf{x}_i^{(Q_S)}), y_i^{(Q_S)}\}_{i=1}^n$.

This new dataset will be distributed according Q_T, but it carries the labels of the samples $\mathbf{x}_i^{(Q_S)}$. For this strategy to be valid, one must assume $y_i^{(Q_S)} = h_S(\mathbf{x}_i^{(Q_S)}) = h_T(T_\gamma(\mathbf{x}_i^{(Q_S)}))$, as we previously highlighted. We illustrate in Figure 4.6 an example of the application of this strategy.

4.4.2 Joint Distribution Optimal Transport

In the OTDA approach, one must assume that $Q_S(\mathbf{y}|\mathbf{x}) = Q_T(\mathbf{y}|T(\mathbf{x}))$, otherwise transportation would not *carry the labels* to the target domain. As discussed in [105], there is no clear reason on why this assumption must hold. More generally, one may assume $Q_S(\mathbf{x}, \mathbf{y}) \neq Q_T(\mathbf{x}, \mathbf{y})$. Upon this new assumption, one has a major shortcoming, as the estimation of a probability metric $\mathcal{D}(Q_S, Q_T)$ would involve the labels of the target domain, which are inaccessible in unsupervised domain adaptation. However, note that for a classifier $h \in \mathcal{H}$, it is possible to artificially label the points in the target domain. This leads to a *proxy distribution*,

$$\hat{Q}_T^{(h)}(\mathbf{x}, \mathbf{y}) = \frac{1}{m} \sum_{j=1}^{m} \delta\Big((\mathbf{x}, \mathbf{y}) - (\mathbf{x}_j^{(Q)}, h(\mathbf{x}_j^{(Q)})) \Big).$$



(a) Source domain. (b) Target domain. (c) $\gamma^{(0)}$.

(d) $h^{(0)}$. (e) $\gamma^{(1)}$. (f) $h^\star = h^{(1)}$.

Figure 4.7 – In Joint Distribution Optimal Transport (JDOT), one starts with a transportation plan between $\hat{Q}_S$ and $\hat{Q}_T$, then uses it to obtain a classifier $h^{(0)}$. This classifier is then re-used to obtain $\gamma^{(1)}$ between the measures $\hat{Q}_S$ and $\hat{Q}_T$, leading to a classifier $h^\star$ that correctly classifies the target domain data.

This workaround is at the heart of the JDOT strategy, as one seeks to minimize the (joint) Wasserstein distance $\mathcal{W}_1(\hat{Q}_S, \hat{Q}_T^{(h)})$ with respect $h \in \mathcal{H}$. In other words, the JDOT objective may be written

as,

$$h^\star = \underset{h\in\mathcal{H}}{\operatorname{argmin}}\, \mathcal{W}_1(\hat{Q}_S, \hat{Q}_T^{(h)}),$$
$$= \min_{h\in\mathcal{H},\gamma\in\Gamma(\hat{Q}_S,\hat{Q}_T^{(h)})} \sum_{i=1}^{n}\sum_{j=1}^{m} \underbrace{\left(\alpha\|\mathbf{x}_i^{(Q_S)} - \mathbf{x}_j^{(Q_T)}\|_2^2 + \mathcal{L}(y_i^{(Q_S)}, h(\mathbf{x}_j^{(Q_T)}))\right)}_{\text{ground-cost } C_{ij}} \gamma_{ij}, \tag{4.13}$$

here, note that the ground-cost C_{ij} is a function of features $(\mathbf{x}_i^{(Q_S)}, \mathbf{x}_j^{(Q_T)})$ and labels $(y_i^{(Q_S)}, h(\mathbf{x}_j^{(Q_T)}))$, and $\mathcal{L}$ is a loss function between labels (e.g., the Hinge loss, or the cross-entropy). An interesting aspect of the double optimization in 4.13 is that it may be performed in blocks, i.e., with respect to h for a fixed γ, and vice-versa. This leads to a two steps iterative procedure. For an iteration k,

$$\gamma^{(k+1)} = \min_{\gamma\in\Gamma(\hat{Q}_S,\hat{Q}_T^{(h)})} \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\alpha\|\mathbf{x}_i^{(Q_S)} - \mathbf{x}_j^{(Q_T)}\|_2^2 + \mathcal{L}(y_i^{(Q_S)}, h^{(k)}(\mathbf{x}_j^{(Q_T)}))\right)\gamma_{ij},$$
$$h^{(k+1)} = \min_{h\in\mathcal{H}} \sum_{i=1}^{n}\sum_{j=1}^{m} \mathcal{L}(y_i^{(Q_S)}, h(\mathbf{x}_j^{(Q_T)}))\gamma_{ij}^{(k)}$$

we illustrate the iterative procedure in figure 4.7.

4.4.3 Hierarchical Optimal Transport

A key advantage in OT is taking into account the data geometry, notably through the ground cost c. However, besides the data geometry, the previous methods are fairly agnostic with respect the structure of the data. Indeed, these methods are non-parametric, meaning that no implicit assumptions are made about the underlying measures. At one hand, this modeling choice offers freedom in representing various, heterogeneous kinds of data. On the other hand, the model is issensitive to some additional structure that the data may have. For classification, a reasonable assumption is that data is clustered among the classes.

This initial remark motivates the Hierarchical OT (HOT) strategy of [106]. The idea is to leverage clustering for proposing an *Hierarchical* OT problem. For the source domain, there is a straightforward cluster structure, namely, the classes. Let $\{(\mathbf{x}_i^{(Q_S)}, y_i^{(Q_S)})\}_{i=1}^{n_S}$. These data induce n_c empirical measures. For $c = 1, \cdots, n_c$,

$$\hat{Q}_{S,c} = \frac{1}{|i : y_i^{(Q_S)} = c|} \sum_{i:y_i^{(Q_S)}=c} \delta(\mathbf{x} - \mathbf{x}_i^{(Q_S)}).$$

The structure for the target domain is not so straightforward, as there are no labels for the target domain data. Here, [106] proposes to take advantage of clustering algorithms. Let $\tilde{y}_j^{(Q_T)}$ be the assignment of each $\mathbf{x}_j^{(Q_T)}$ to cluster $c = 1, \cdots, n_c$ by a clustering algorithm (e.g., $k-$means). There is no correspondence between the class index c and the cluster index c. However, it is possible to solve an OT problem between classes and clusters,

$$\gamma^\star = \underset{\gamma\in\Gamma(\mathbf{q}_S,\mathbf{q}_T)}{\operatorname{argmin}} \sum_{c_1=1}^{n_c}\sum_{c_2=1}^{n_c} \gamma_{c_1,c_2} W_{c_1,c_2} = \underset{\gamma\in\Gamma(\mathbf{q}_S,\mathbf{q}_T)}{\operatorname{argmin}} \sum_{c_1=1}^{n_c}\sum_{c_2=1}^{n_c} \gamma_{c_1,c_2} \mathcal{W}_2(\hat{Q}_{S,c_1}, \hat{Q}_{T,c_2}), \tag{4.14}$$

where W_{c_1,c_2} is an *hierarchical ground-cost matrix*, which computes the empirical Wasserstein distance between class c_1 and cluster c_2. Furthermore, the vector $\mathbf{q}_S = \{q_{S,c_1}\}_{c_1=1}^{n_c}$ (resp. $\mathbf{q}_T$) correspond to the proportion of samples in class c_1 (resp., cluster c_2). Based on $\gamma^\star$, one can derive the corresponding class for each cluster, that is,

$$c_2 \sim c_1 \iff c_1 = \underset{c=1,\cdots,n_c}{\operatorname{argmax}} \gamma_{c,c_2},$$

where $a \sim b$ means that a is equivalent to b. Once a correspondence between classes and clusters has been established, once can transport source class c to its corresponding target cluster through the barycentric mapping.

4.4.4 Information Maximizing Optimal Transport

In the same spirit of inducing additional structure on the transportation problem, [107] proposed a method for enriching OT with a Kernel Density Estimation (KDE) approach. Their approach relies on the notion of *mutual information* [108]. For random variables $X^{(Q_S)}$ and $X^{(Q_T)}$, this quantity is,

$$I(X^{(Q_S)}; X^{(Q_T)}) = \int_{\mathcal{X}_T}\int_{\mathcal{X}_S} q_{S,T}(\mathbf{x}_S, \mathbf{x}_T) \log\left(\frac{q_{S,T}(\mathbf{x}_S, \mathbf{x}_T)}{q_S(\mathbf{x}_S) q_T(\mathbf{x}_T)}\right) d\mathbf{x}_S d\mathbf{x}_T$$

where $q_{X,Z}$ denotes the density of the joint probability measure between $X^{(Q_S)}$ and $X^{(Q_T)}$. Likewise, q_S (resp., q_T) represent the probability measure of $X^{(Q_S)}$ (resp., $X^{(Q_T)}$). Evaluating $I(X^{(Q_S)}; X^{(Q_T)})$ requires knowledge about the density of these variables, which is not readily available. Given a paired set of samples $\{\mathbf{x}_i^{(Q_S)}, \mathbf{x}_i^{(Q_T)}\}_{i=1}^n$ i.i.d. from the measures Q_S and Q_T, and a kernel $k : \mathbb{R} \to \mathbb{R}$, one can estimate the densities using KDE,

$$\begin{aligned}
q_S(\mathbf{x}_S) &= \frac{1}{n}\sum_{i=1}^n k(d_{\mathcal{X}_S}(\mathbf{x}_S, \mathbf{x}_i^{(Q_S)})), \\
q_T(\mathbf{x}_T) &= \frac{1}{n}\sum_{i=1}^n k(d_{\mathcal{X}_T}(\mathbf{x}_T, \mathbf{x}_i^{(Q_T)})), \\
q_{S,T}(\mathbf{x}_S, \mathbf{x}_T) &= \frac{1}{n}\sum_{i=1}^n k(d_{\mathcal{X}_S}(\mathbf{x}_S, \mathbf{x}_i^{(Q_S)}))k(d_{\mathcal{X}_T}(\mathbf{x}_T, \mathbf{x}_i^{(Q_T)})),
\end{aligned} \tag{4.15}$$

note that q_S and q_T can be seen as smooth versions of the marginal vectors $\mathbf{q}_S \in \Delta_n, \mathbf{q}_T \in \Delta_m$ in empirical OT. Furthermore, the notion of kernel in KDE is slightly different from that used in the MMD. For instance, in [107] the authors use,

$$k_h(d_{\mathcal{X}_S}(\mathbf{x}_S, \mathbf{x}'_S)) = \frac{1}{Z_h}\exp\left(-\frac{d_{\mathcal{X}_S}(\mathbf{x}_S, \mathbf{x}'_S)}{2\sigma^2 h}\right), \tag{4.16}$$

where Z_h is a normalization constant making $\int_{\mathcal{X}_S} q_S(\mathbf{x}_S) d\mathbf{x}_S$ interegrate to 1. In the context of OT, adopting the notion in equation 4.16 is convenient, as the densities only depend on the intra-domain distances. As a result, one can have different feature spaces $\mathcal{X}_S \neq \mathcal{X}_T$. While interesting, this possibility is beyond the scope of this thesis.

Nonetheless, note that the construction of densities in equation 4.15 requires a paired set of samples from the measures Q_S and Q_T. This requirement is seldom met in practice, as samples are acquired in an i.i.d. fashion. For unpaired samples $\{\mathbf{x}_i^{(Q_S)}\}_{i=1}^n$ and $\{\mathbf{x}_i^{(Q_T)}\}_{j=1}^m$, one can find an alignment $\gamma \in \Gamma(\mathbf{q}_S, \mathbf{q}_T)$, so that,

$$q_\gamma(\mathbf{x}_S, \mathbf{x}_T) = \sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{ij} k(d_{\mathcal{X}_S}(\mathbf{x}_S, \mathbf{x}_i^{(Q_S)}))k(d_{\mathcal{X}_T}(\mathbf{x}_T, \mathbf{x}_j^{(Q_T)})).$$

With this estimator, the mutual information can be calculated as,

$$I_\gamma(X_S; X_T) = \sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{ij} \log \underbrace{\frac{nm \sum_{i'=1}^{n}\sum_{j'=1}^{m} \gamma_{i'j'} k(d_{\mathcal{X}_S}(\mathbf{x}_i^{(Q_S)}, \mathbf{x}_{i'}^{(Q_S)}))k(d_{\mathcal{X}_T}(\mathbf{x}_j^{(Q_T)}, \mathbf{x}_{j'}^{(Q_T)}))}{\left(\sum_{i'=1}^{n} k(d_{\mathcal{X}_S}(\mathbf{x}_i^{(Q_S)}, \mathbf{x}_{i'}^{(Q_S)}))\right)\left(\sum_{j'=1}^{m} k(d_{\mathcal{X}_T}(\mathbf{x}_j^{(Q_T)}, \mathbf{x}_{j'}^{(Q_T)}))\right)}}_{R_{ij}}.$$

An interesting property of this estimator is that, when $\sigma \to 0$, $I_\gamma \to -H(\gamma)$ (see [107, Lemma 4.2]). This remark motivates using $I_\gamma(X_S; X_T) = \langle \gamma, \log \mathbf{R} \rangle_F$ as a regularizer in empirical OT, that is,

$$\gamma^\star = \underset{\gamma \in \Gamma(\mathbf{q}_S, \mathbf{q}_T)}{\operatorname{argmin}} \langle \gamma, \mathbf{C} - \epsilon \log \mathbf{R} \rangle, \tag{4.17}$$

for $\epsilon > 0$. This problem can be solved through the conjugated gradient method introduced in [105]. Besides the novelty of the OT problem in equation 4.17, an important advantage of performing KDE is having a density for the source, target and joint measures. This facts allows for out-of-sample mapping, using,

$$\begin{aligned} T(\mathbf{x}_S) &= \mathop{\mathbb{E}}_{\mathbf{x}_T \sim Q(\mathbf{x}_T | X_S = \mathbf{x}_S)}[\mathbf{x}_T] = \mathop{\mathbb{E}}_{\mathbf{x}_T \sim Q_T}\left[\frac{q_{T|S}(\mathbf{x}_T)}{q_T(\mathbf{x}_T)}\mathbf{x}_T\right] = \mathop{\mathbb{E}}_{\mathbf{x}_T \sim Q_T}\left[\frac{q_{S,T}(\mathbf{x}_S, \mathbf{x}_T)}{q_S(\mathbf{x}_S) q_T(\mathbf{x}_T)}\mathbf{x}_T\right], \\ &\approx \frac{1}{Z}\sum_{j=1}^{m} \frac{q_\gamma(\mathbf{x}, \mathbf{x}_j^{(Q_T)})}{q_S(\mathbf{x}) q_T(\mathbf{x}_j^{(Q_T)})}\mathbf{x}_j^{(Q_T)}. \end{aligned} \tag{4.18}$$

Equation 4.18 serves the same purpose of the barycentric mapping (c.f., equation 2.19).

4.4.5 Invariant Representation Learning

Besides manipulating the labeled data, it is possible to design learning algorithms that, throughout training, perform adaptation towards the target domain. This kind of approach is prominent in deep learning, in which the overall architecture is divided into an encoder $\phi : \mathcal{X} \to \mathcal{Z}$ and a classifier $h : \mathcal{Z} \to \mathcal{Y}$. Here, $\mathcal{Z}$ is a latent, or representation space, that is, the intermediate space of the deep neural net layers. Looking at neural nets from this perspective is useful, since $\mathcal{Z}$ often carries semantic information about the data points. Furthermore, the process of building a rich latent space is often called *representation learning* [109]. These ideas are shown in Figure 4.8

The principle behind *Invariant Representation Learning (IRL)* is to use labeled source domain data, and unlabeled target domain data through the training of $\phi \circ h$. The idea is to use unlabeled

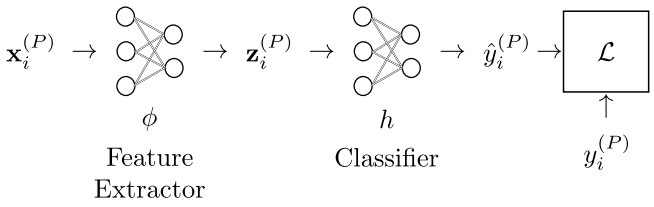


Figure 4.8 – **Architecture of a deep neural net.** In this case, the classifier is implemented through the composition of a feature extractor ϕ, and a (feature) classifier h, which are parametrized via neural networks.

target data to align, in the latent space, the distributions Q_S and Q_T. As such, the goal is to minimize, alongside the standard ERM, an additional alignment loss $\mathcal{D}(\phi_\sharp Q_S, \phi_\sharp Q_T)$, that is,

$$(\theta_\phi^\star, \theta_h^\star) = \underset{\theta_\phi, \theta_h}{\operatorname{argmin}} \mathcal{R}_{Q_S}(\phi \circ h) + \beta \mathcal{D}(\phi_\sharp Q_S, \phi_\sharp Q_T),$$

where θ_ϕ and θ_h are the parameters of the encoder and classifier, and $\beta > 0$ is an importance factor.

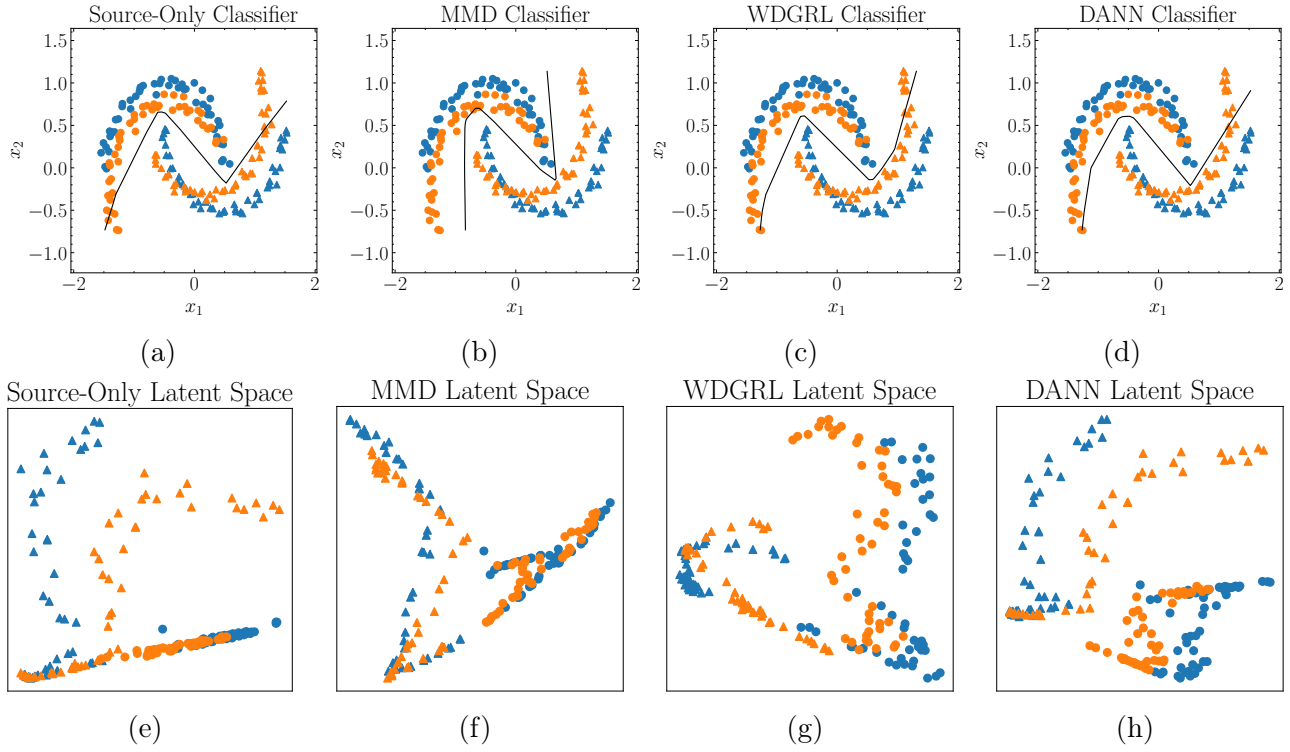


Figure 4.9 – Comparison of Deep DA strategies, based on the MMD (b, f) and OT (c, d, g, h). Overall, (e – h) show the PCA (2 components) of the latent space of a neural net.

The first approaches on IRL used the MMD (see section 3.1.2) for learning invariant features with deep neural nets. This is the case of [110], which minimized the MMD between representations at the last layer of a convolutional neural net, and [111], who used the MMD between various intermediate

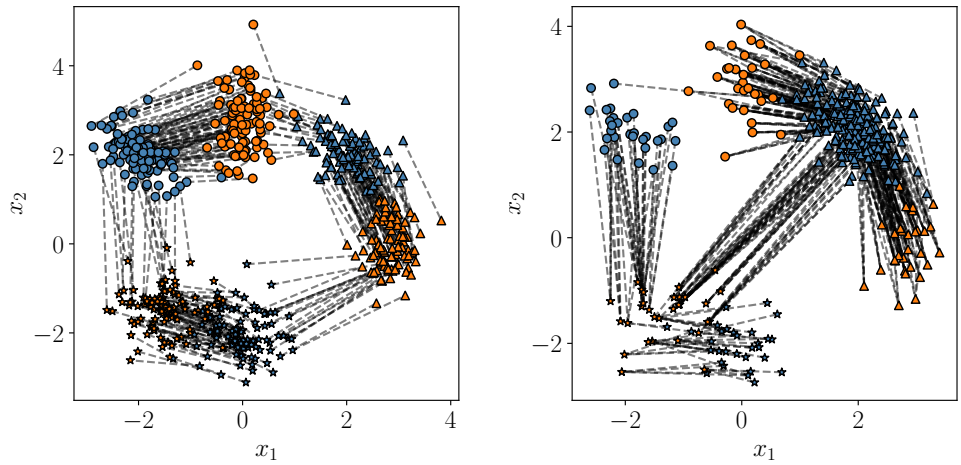
representations. Here, it is noteworthy that the same idea was applied by [112] for the multi-source scenario.

Second, one of the most influential works in IRL consists of *domain adversarial training* [76]. The idea is to learn an encoder such that $\phi_\sharp Q_S$ and $\phi_\sharp Q_T$ are close together under the $\mathcal{H}$ distance. In other words, one wants to learn representations such that a classifier cannot distinguish from which domain each sample comes. This approach bears an interesting link to the principle behind Generative Adversarial Nets [73], which tries to generate synthetic data that is indistinguishable from true data. On the same reasoning, [38] proposes an approach based on the $1-$Wasserstein distance, thus making a parallel with the Wasserstein GAN strategy of [37]. We show a comparison of these different strategies in figure 4.9, for the two moons toy example.

4.4.6 Multi-Source Domain Adaptation

In this section, we discuss OT-based MSDA algorithms [24, 113]. These 2 methods share a key similarity: they rely on a notion of aggregation for representing the multiple source domains. In the case of [24], the idea relies on fixed-support Wasserstein barycenters, whereas [113] linearly weights source domains, which can be seen as a barycenter under the MMD metric (see chapter 3). As such, these methods share an interesting link with our proposed strategy, Wasserstein barycenter transport [12, 13].

So far, we considered two kinds of distribution shifts, namely $Q_S(X) \neq Q_T(X)$ and $Q_S(X, Y) \neq Q_T(X, Y)$. In [24], the authors tackle another problem, known as *target shift*, that is $Q_S(Y) \neq Q_T(Y)$. As we are dealing with classification in this thesis, this shift can be understood as a change in the balance of classes across domains. As a consequence of target shift and the fact that vanilla optimal transport is forced to preserve mass, the transport plan γ will match points from different classes. We illustrate this problem in figure 4.10.



(a) Uniform and equal proportions. (b) Target shift.

Figure 4.10 – Optimal transport plan for two datasets. In (a), source (blue points) and target (orange points) domains have the same class proportions (uniform). In (b), the class proportions are not the same. As a result, OT is forced to match samples from different classes.

Inspired by the aforementioned challenge, [24] proposes to perform multi-source domain adaptation under target shift by estimating the proportion of classes in the target domain through Wasserstein barycenters. The starting point is considering $Q_S^{(\pi_S)} = \sum_{c=1}^{n_c} \pi_{S,c} Q_{S,c}$, where $Q_{S,c}$ is the distribution of the $c-$class, and $\pi_S \in \Delta_{n_c}$ are the proportions of each class in the source domain. One may do the same decomposition for the target domain, except that, since the target is unlabeled, π_T is unknown. However, as shown in [24, Propsition 3], one can estimate π_T through an optimization problem,

$$\pi_T = \underset{\pi \in \Delta_{n_c}}{\operatorname{argmin}} \mathcal{W}_1(Q_S^{(\pi)}, Q_T),$$

which naturally extends to the multi-source case,

$$\pi_T = \underset{\pi \in \Delta_{n_c}}{\operatorname{argmin}} \sum_{k=1}^{K} \lambda_k \mathcal{W}_1(Q_{S_k}^{(\pi)}, Q_T). \tag{4.19}$$

As a result, the JCPOT strategy can be summarized as follows. First, one solves equation 4.19 for π_T. This further leads to K transport plans $\gamma^{(1)}, \cdots, \gamma^{(K)}$, between each source and the target. These transport plans have the advantage of taking the proportions of classes into account, as a result, they are not equivalent to independently solving K Kantorovich problems. The next step consists of using $\gamma^{(k)}$ for mapping the sources towards the target domain using the barycentric mapping. An alternative strategy consists of mapping the labels $\hat{Y} = \sum_k \lambda_k \gamma^{(k)} \mathbf{Y}^{(P_k)}$, where $\mathbf{Y}^{(Q_{S_k})}$ is the matrix of one-hot encoded labels of the $k-$th source domain.

Weighted JDOT goes in another direction, and considers a convex combination of source domain distributions. Let $\hat{Q}_{S_1}, \cdots, \hat{Q}_{S_K}$ be given, each with n_k samples. For $\lambda \in \Delta_K$, the authors define

$$\hat{Q}_S^{(\lambda)}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{K} \lambda_k \hat{Q}_{S_k} = \sum_{k=1}^{K} \frac{\lambda_k}{n_k} \sum_{i=1}^{n_k} \delta\Big((\mathbf{x}, \mathbf{y}) - (\mathbf{x}_i^{(Q_{S_k})}, \mathbf{y}_i^{(Q_{S_k})})\Big),$$

which is a new empirical distribution, reweighed by λ_k, which contains $n = \sum_k n_k$ samples. The insight behind WJDOT is to apply JDOT between distributions Q_S^{λ} and $Q_T^{(h)}$,

$$(\lambda^\star, h^\star) = \underset{h \in \mathcal{H}, \lambda \in \Delta_K}{\operatorname{argmin}} \mathcal{W}_1(Q_S^{(\lambda)}, Q_T^{(h)}). \tag{4.20}$$

Like JDOT, the optimization problem in equation 4.20 can be solved iteratively via gradient descent. The procedure described in [113, Algorithm 1] parametrizes h through θ (e.g., through a neural network), then takes steps with respect θ, for a fixed λ, and vice-versa.

4.5 Domain Adaptation Benchmarks

In this section, we present some domain adaptation benchmarks that will be used as examples throughout this thesis. A further advantage of this section is presenting a concise review of existing datasets used by the community. As we covered so far, while not mandatory, DA techniques are often designed for classification problems, with a few exceptions (e.g., [105]). Within classification, applications range from different fields, such as object recognition [114], audio processing [12, 113],

fault diagnosis [104] and sentiment analysis [115]. While upcoming chapters treat the problem of fault diagnosis under DA in more details (e.g., chapter 8), in our methodological chapters we use some benchmarks from other fields to illustrate our methods.

Digits 5. This benchmark was first proposed in [116]. It consists of 0 to 9 digits from 5 domains: MNIST (mt) [117], MNIST-M (mm) [76], SVHN (sv) [118], USPS (up) [119] and Synthetic Digits (sy) [76]. From each dataset, except USPS, 25000 samples are drawn for training, and 9000 for testing. For USPS, the entire dataset is used as a domain. In comparison with other benchmarks, *Digits 5* is relatively large scale with respect its number of samples (approximately 180000 samples). However, the task itself (digit recognition) is relatively simple, and the number of classes is small in comparison with other benchmarks. This benchmark is considered a good toy example for deep DA methods, which rely on large amounts of data. We show in Figure 4.11 an overview of samples from the domains of this benchmark.

Office-like datasets. An important benchmark in DA was proposed by [114], and consists of a dataset with 3 domains: Amazon (A), dSLR (D) and Webcam (W). These domains contain 2817, 498 and 795 images respectively, from a total of 31 objects. While images from amazon are collected automatically from the web, dSLR and Webcam are high and low resolution images from the same objects, respectively. As such, one knows *a priori* that these domains are highly similar. One should note, however, that due the automatic annotation process of the amazon domain, the domain adaptation process is subject to *label noise*. Indeed, [120] provides a closer look at this issue, and propose clean and adapted versions for this classic benchmark.

Based on the Office 31 benchmark, another DA benchmark was proposed by [121]: the Caltech-Office benchmark. The idea is to merge the domains of Office 31 with the Caltech dataset [122], which has 256 classes of objects. These two datasets have 10 classes in common. After the intersection of these datasets, one forms the Caltech-Office benchmark, with four domains: A, D, W and Caltech (C). These domains have 958, 157, 295 and 1123 samples each, for a total of 2533 samples. Note that, as a result of filtering classes not present in the Caltech benchmark, the domains A, D and W have a reduced number of samples. Furthermore, the label noise originally present in the Office 31 benchmark is present in the Caltech-Office benchmark as well.

Both Office 31 and Caltech-Office benchmarks have a reduced number of samples. For instance, these benchmarks represent approximately 2.28% and 1.41% of the total amount of samples in the Digits 5 benchmark. At the same time, they pose a more realistic adaptation problem. For instance, object recognition is not as artificial as digit recognition. Furthermore, the resolution of images in these benchmarks (418×418 pixels on average) is larger than those in Digits 5 (32×32 pixels). While the problem considered by these benchmarks is more interesting, they pose a problem for deep DA methods, that commonly rely on large amounts of data for training a neural encoder and a classifier. This motivates [123] to propose a larger *Office-like* benchmark, called Office-Home.

The Office-Home benchmark is larger than the Office 31 and Caltech-Office benchmarks. It consists of 4 domains: Art (Ar), Clipart (Cl), Product (Pr) and Real World (Rw), with 2427, 4365, 4439 and 4357 samples each, for a total of 15588 samples. We show samples from these domains in Figure 4.11. Note that, due the diverse nature of images, and the larger number of classes, this benchmark is more challenging than previous benchmarks. Especially, one can expect a lower classification accuracy on

challenging domains such as art and clipart.

Besides object recognition, existing bechmarks explore other problems. A non-exhaustive list of tasks include face recognition [124], animal recognition [125] and semantic segmentation [126].

(a) Digits-5 (b) Caltech-office (c) Office-home

Figure 4.11 – Illustration of visual adaptation benchmarks. The digits 5 benchmark is composed of digits (0 to 9) from 5 different domains, as shown in (a). The Caltech-Office benchmark (b) is composed of images of 10 objects, of 4 different domains. Note that domains D and W are highly similar, as their only difference is the image of the camera that captured the object images. The Office 31 benchmark is also illustrated in (b), as it corresponds to the domains A, D and W. Finally, in (c) we show the office-home benchmark, also composed of 4 domains. The main difference between (b) and (c) is the number of samples and classes in these benchmarks.

Amazon Reviews. One of the earliest applications of DA is for sentiment classification of different types of products. This is the principle behind the amazon reviews benchmark [115], which contains product reviews from products from 4 categories: books, dvd, electronics and kitchen. In natural language, features work differently than image processing, due to the inherently discrete nature of text. For instance, if one considers term frequency features, the terms associated with a positive review for a book (e.g., insightful), might not be the same for an electronic product (e.g., efficient).

(a) Books (b) DVDs (c) Electronics (d) Kitchen

Figure 4.12 – Word cloud based on the frequency of occurence of words in positive in the Amazon reviews benchmark on different domains. Note that words used to positively describe DVDs (life, performance, classic) are quite different from Electronics (work, price, easy), even though some words are shared (good, quality). This illustrates qualitatively the distributional shift occuring in text benchmarks.

4.6 Conclusion

In this chapter, we review learning theory and the theoretical principles of domain adaptation. In this sense, we base our discussion on previous reviews of domain adaptation, such as [127] and [21, 22]. In particular, we use the ERM framework [5, 6] as our main theoretical cornerstone. The main result of this theory is that, given enough samples, the empirical approximation of a classifier's risk (i.e., its probability of making a mistake) is close to the true risk. In mathematical terms, this result takes the form of a bound, that is, the difference of the risks depends on a term that is $\mathcal{O}(n^{-1/2}\log n)$. Other constants are involved in this term, such as the complexity of the family of classifiers being considered. In this sense, we call this term *learning complexity*. These ideas give a statistical foundation for the notion of generalization, i.e., the ability to reliably predict on unseen samples.

Moving on the notion of domain adaptation, one needs to cope to learning under different probability measures. Here, theory is fairly intuitive. If data comes from similar probability measures, generalization should take place as the ERM framework. For dissimilar measures, there is no hope for a successful adaptation. As a result, the notion of distance between probability measures comes into play. Indeed, some of the distances presented in Chapter 3 are used to bound the risks under different domains. However, there is an additional caveat. Since these distances only take into account how *featuers are distributed* (i.e., the distances are over $\mathbb{P}(\mathcal{X})$), one should beware of taking labels into account. Here, the notion of *adaptation complexity* comes into play [20], as it takes the form as the minimum error a classifier is able to achieve, when having access to labeled data from both domains. To summarize, theoretical results in domain adaptation usually take the following form. For $\delta \in (0, 1)$, with probability $1 - \delta$,

$$\hat{\mathcal{R}}_{Q_T}(h) \leq \hat{\mathcal{R}}_{Q_S}(h) + \mathcal{D}(\hat{Q}_S, \hat{Q}_T) + \mathcal{C}_{ERM}(n, \delta, \mathcal{H}) + \mathcal{C}_{\mathcal{D}}(n, \delta) + \mathcal{C}_{DA}(Q_S, Q_T),$$

where $\mathcal{C}_{ERM}$ is the learning complexity, $\mathcal{C}_{\mathcal{D}}$ is the error associated with the estimation of $\mathcal{D}(Q_S, Q_T)$ with finite samples, and $\mathcal{C}_{DA}(Q_S, Q_T)$ is the instrinsic complexity of domain adaptation.

Based on these ideas, we present in section 4.4 several algorithms that perform domain adaptation by minimizing the Wasserstein distance between source and target domain measures. These algorithms are further used, in Chapter 9, for comparing how different strategies rank in domain adaptation benchmarks. To further discuss domain adaptation practice, we presented in section **??** a brief description of popular benchmarks used by the community.

Overall, optimal transport has made a significant impact in the field of domain adaptation, being a useful tool for building algorithms that match source and target probability measures. Furthermore, since [100], optimal transport is an important tool in the theoretical analysis of domain adaptation algorithms.

Part II

Methodological Contributions

Chapter 5

Wasserstein Barycenter Transport

Contents

This chapter presents our first contributions to MSDA through Wasserstein barycenters, namely [12] and [13]. While very similar in content, we update some of its language to better reflect our notation and discussion throughout this thesis.

As we discussed in chapter 3, the idea of Wasserstein barycenter was introduced by [82]. The idea is to extend the geometric notion of barycenter in Euclidean spaces - a point equidistant to a group of points - to probability measures. This notion is then defined based on the Wasserstein metric over $\mathbb{P}(\mathcal{X})$. As such, in section 3.2 we discussed these ideas extensively in the case of discrete probability measures.

Nonetheless, one should be careful when applying the notion of Wasserstein barycenters in domain adaptation, as probability measures carry label information (e.g., classes, as in classification). As a result, one needs to make sure that the calculation of a barycenter does not mixes classes. In the following discussion, we build some theory guaranteeing class consistent barycenters. One should note, however, that this requires labeled distributions (i.e., feature-label pairs). In MSDA - the main focus of this thesis - this is true for source domain data. However, this is not true for target domain data.

The rest of this chapter is divided as follows. Section 5.1 discusses previous strategies for obtaining class-sparse transport plans. We then introduce, in section 5.1.2 the distance we adopt throughout many of ours works over $\mathcal{X} \times \mathcal{Y}$. This distance allows us to retrieve the barycentric mapping and label propagation formulas as first-order optimality conditions of the associated Wasserstein-like distance. Section 5.2 introduces our algorithm for MSDA, namely WBT. Section **??** presents the theoretical guarantees of WBT, namely [13, Theorem 1]. Finally, 5.3 concludes this chapter.

5.1 Optimal Transport with Labeled Distributions

We divide our discussion into 3 parts. Section 5.1.1 discusses class-based regularizers. Section 5.1.2 discusses optimal transport between feature-label joint measures. Section 5.1.3 covers optimal transport dataset distances. As a reminder, we consider in this section discrete OT, that is, a linear program of size $n_S \times n_T$ such that,

$$\gamma^\star = \underset{\gamma \in \Gamma(\mathbf{q}_S, \mathbf{q}_T)}{\operatorname{argmin}} \langle \gamma, \mathbf{C} \rangle_F + \epsilon H(\gamma), \tag{5.1}$$

where for $\epsilon > 0$ one has *entropic OT*, and for $\epsilon = 0$ one has *exact OT*. See e.g., chapter 2 and section 2.2.1 for further details.

5.1.1 Class-Regularized Optimal Transport

As remarked in [23, Section 4.1], the discrete OT problem does not take into account the class information that is available in domain adaptation. An example of such problems occurs when γ *mixes classes* during transportation. Recalling the barycentric mapping strategy for domain adaptation (section 4.4.1), the source samples $\mathbf{x}_i^{(Q_S)}$ are transported to (see e.g., equation 2.19),

$$\hat{\mathbf{x}}_i^{(Q_S)} = T_\gamma(\mathbf{x}_i^{(Q_S)}) = \sum_{j=1}^{n_T} \frac{\gamma_{ij}}{q_{S,i}} \mathbf{x}_j^{(Q_T)}.$$

Now, since $\sum_{j=1}^{n_T} \gamma_{ij}/q_{S,i} = 1$, $\hat{\mathbf{x}}_i^{(Q_S)}$ lies on the convex hull of $\{\mathbf{x}_j^{(Q_T)} : \gamma_{ij} > 0\}$. As a result, if γ mixes classes, the transported samples $\hat{\mathbf{x}}_i^{(Q_S)}$ may end up on the wrong side of the classification boundary. Here, we also make a parallel with theoretical results in DA (see e.g., theorem 6). If γ mixes the classes, it means that even though $\mathcal{W}_1(\hat{Q}_S, \hat{Q}_T)$ is minimized, the joint error $\mathcal{C}_{DA}(T_{\gamma,\sharp}\hat{Q}_S, \hat{Q}_T)$ grows, due the mixing of classes.

This discussion suggests that having transport plans that do not mix classes is a strong desiderata for DA. In light of [23], we call this property *class sparsity*,

Definition 26. *(Class Sparse Transport Plans) Let Q_S and Q_T be two probability measures in $\mathbb{P}(\mathcal{X} \times \mathcal{Y})$. Let $(\mathbf{X}^{(Q_S)}, \mathbf{Y}^{(Q_S)})$ and $(\mathbf{X}^{(Q_T)}, \mathbf{Y}^{(Q_T)})$ be two samples of size n_S and n_T from Q_S and Q_T respectively. A transport plan $\gamma \in \Gamma(\mathbf{q}_S, \mathbf{q}_T)$ is called* class sparse *if,*

$$\gamma_{ij} > 0 \iff y_i^{(P)} = y_j^{(Q)}.$$

Note that, in order to assess class sparsity, it necessary to have labels in the target domain. In unsupervised DA, this is not always the case. A closely-related problem is semi-supervised DA, in which a few labeled samples are available in the target domain. In this case, a straightforward way for inducing class sparsity is,

$$\gamma^\star = \underset{\gamma \in \Gamma(\mathbf{q}_S, \mathbf{q}_T)}{\operatorname{argmin}} \langle \gamma, \mathbf{C} \rangle_F + \epsilon H(\gamma) + \beta \sum_{i=1}^{n_S} \sum_{j=1}^{n_T} \gamma_{ij} \delta(y_i^{(P)} - y_j^{(Q)}), \tag{5.2}$$

where β is a penalty term that expresses how costly it is to mix classes. For $\beta = 0$, one has standard OT, whereas for $\beta \to +\infty$ OT plans are necessarily class sparse. An alternative view on this penalty comes from incorporating $\beta\delta(y_i^{(P)} - y_j^{(Q)})$ into the ground-cost, that is [1],

$$C_{ij} = \|\mathbf{x}_i^{(Q_S)} - \mathbf{x}_j^{(Q_T)}\|_2^2 + \beta\delta(y_i^{(P)} - y_j^{(Q)}). \tag{5.3}$$

With such a ground-cost, the OT problem is now defined over the joint ambient space, that is, features and labels $\mathcal{X} \times \mathcal{Y}$. We revisit this idea in the next section.

Unfortunately, for unsupervised DA the labels $y_j^{(Q)}$ are not available. As a result, it is necessary to rely on heuristics. [23] introduced a series of strategies for inducing class sparsity without $y_j^{(Q)}$, i.e., group-sparsity and Laplacian regularization. In the first case, the goal is to induce a sparse representation of transported source samples. This is achieved by inducing sparsity on the columns of γ_{ij}. That way, $\hat{\mathbf{x}}_i^{(Q_S)}$ is represented with a few target domain samples. Hence,

$$\gamma^\star = \underset{\gamma \in \Gamma(\mathbf{q}_S, \mathbf{q}_T)}{\operatorname{argmin}} \langle \gamma, \mathbf{C} \rangle_F + \epsilon H(\gamma) + \beta \sum_{j=1}^{n_T} \sum_{c=1}^{n_c} \|\gamma_{\mathcal{I}_c, j}\|_2,$$

where $\gamma_{\mathcal{I}_c, j} = (\gamma_{i_1, j}, \cdots, \gamma_{i_{n'}, j})$ corresponds to the vector of γ_{ij}'s corresponding to source domain samples with class c.

Conversely, one may represent the relationship between samples through a graph. Given positive and symmetric similarity matrices for the source and target, i.e., $\mathbf{S}_S$ and $\mathbf{S}_T$, the idea consists of regularizing the transport plan based on,

$$\Omega(\gamma) = (1-\alpha)\mathrm{Tr}((\mathbf{X}^{(Q_T)})^T \gamma^T \mathbf{L}_S \gamma \mathbf{X}^{(Q_T)}) + \alpha \mathrm{Tr}((\mathbf{X}^{(Q_S)})^T \gamma \mathbf{L}_T \gamma^T \mathbf{X}^{(Q_T)}),$$

which forces similar points to be transported closed together. These ideas highlight an important aspect of OT for DA: besides distances between features, it is important to take into account label information. Hence, OT plans are required to have *additional* structure.

5.1.2 Joint Optimal Transport

Arguably, the first work to propose a ground-cost on $\mathcal{X} \times \mathcal{Y}$ was [23]. Note that this idea was further refined in [105], in which the authors consider an arbitrary loss $\mathcal{L}$ between labels. This choice offers an advantage over equation 5.3 because δ is not differentiable with respect its inputs. A different approach, that we take in our recent works [14, 33, 31, 15] is to consider proper distances on the joint space $\mathcal{X} \times \mathcal{Y}$ that are, at the same time, smooth with respect its inputs. This especially important for our dictionary learning problem, which we cover in chapter 6. This is, in itself, a challenge, as the space $\mathcal{Y} = \{1, \cdots, n_c\}$ is discrete for classification. We get around this issue by letting $\mathcal{Y} = \Delta_{n_c} \subset \mathbb{R}^{n_c}$. As such, vectors $\mathbf{y} \in \mathcal{Y}$ are *soft probability vectors*. Using this choice, a distance on $\mathcal{X} \times \mathcal{Y}$ appears naturally,

$$d\Big((\mathbf{x}_i^{(Q_S)}, \mathbf{y}_i^{(Q_S)}); (\mathbf{x}_j^{(Q_S)}, \mathbf{y}_j^{(Q_T)})\Big) = \sqrt{\|\mathbf{x}_i^{(Q_S)} - \mathbf{x}_j^{(Q_T)}\|_2^2 + \beta \|\mathbf{y}_i^{(Q_S)} - \mathbf{y}_j^{(Q_T)}\|_2^2}, \tag{5.4}$$

1. In our initial publications [12] and [13], we used the ground-cost given by equation 5.3. Later, in our publication [14], we needed a continuous ground-cost for performing dictionary learning. As a result, we adopted the Euclidean distance in place of the delta, which proved to work better for domain adaptation.

where β has the same interpretation as in equation 5.3. With this cost, we define a Wasserstein-like distance on $\mathbb{P}(\mathcal{X} \times \mathcal{Y})$,

Definition 27. *(Joint Wasserstein Distance) Let $\mathcal{X}$ be a feature space, and $\mathcal{Y}$ be a label space. Let d be the distance over $\mathcal{X} \times \mathcal{Y}$ given by equation 5.4. For $P, Q \in \mathbb{P}(\mathcal{X} \times \mathcal{Y})$, the $(\alpha, \beta)-$Joint Wasserstein distance between P and Q is,*

$$\mathcal{JW}_{\alpha,\beta}(P,Q)^\alpha = \inf_{\gamma \in \Gamma(P,Q)} \int_{(\mathcal{X}\times\mathcal{Y})^2} d\Big((x_1,y_1),(x_2,y_2)\Big)^\alpha d\gamma((x_1,y_1),(x_2,y_2)). \tag{5.5}$$

Furthermore, given finite samples $(\mathbf{X}^{(P)}, \mathbf{Y}^{(P)})$ from P and $(\mathbf{X}^{(Q)}, \mathbf{Y}^{(Q)})$, the empirical Joint Wasserstein loss is given by,

$$\mathcal{JW}_{\alpha,\beta}(\hat{P},\hat{Q})^\alpha = \min_{\gamma \in \Gamma(\mathbf{p},\mathbf{q})} \sum_{i=1}^{n_P}\sum_{j=1}^{n_Q} C_{ij}\gamma_{ij},, \tag{5.6}$$

where $C_{ij} = d\Big((\mathbf{x}_i^{(Q_S)}, \mathbf{y}_i^{(Q_S)}); (\mathbf{x}_j^{(Q_S)}, \mathbf{y}_j^{(Q_T)})\Big)^\alpha$.

While the $\alpha-$Wasserstein distance $\mathcal{W}_\alpha$ is a distance over $\mathbb{P}(\mathcal{X})$, the (α, β)-Joint Wasserstein distance is a distance over $\mathbb{P}(\mathcal{X} \times \mathcal{Y})$. In the following, we focus on $\mathcal{JW}_{2,\beta}(\hat{P},\hat{Q})^2$, which allows us to derive two rules for transporting features and labels between domains based on the first-order conditions of $\mathcal{JW}_{2,\beta}(\hat{P},\hat{Q})^2$,

Theorem 9. *(First order conditions of $\mathcal{JW}_{2,\beta}$) Let P and Q be probability measures over $\mathbb{P}(\mathcal{X} \times \mathcal{Y})$, and $(\mathbf{X}^{(P)}, \mathbf{Y}^{(P)})$, $(\mathbf{X}^{(Q)}, \mathbf{Y}^{(Q)})$ be samples from P and Q. Let $\gamma^\star$ be the solution of,*

$$\begin{aligned}\gamma^\star &= JOT\Big((\mathbf{X}^{(P)}, \mathbf{Y}^{(P)}), (\mathbf{X}^{(Q)}, \mathbf{Y}^{(Q)})\Big), \\ &= \underset{\gamma \in \Gamma(\mathbf{p},\mathbf{q})}{argmin} \sum_{i=1}^{n_P}\sum_{j=1}^{n_Q} \gamma_{ij}(\|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2^2 + \beta\|\mathbf{y}_i^{(P)} - \mathbf{y}_j^{(Q)}\|_2^2),\end{aligned} \tag{5.7}$$

then, the first order conditions of $(\mathbf{x}_i, \mathbf{y}_i) \mapsto \sum_{i=1}^{n_P}\sum_{j=1}^{n_Q} \gamma_{ij}^\star(\|\mathbf{x}_i - \mathbf{x}_j^{(P)}\|_2^2 + \beta\|\mathbf{y}_i - \mathbf{y}_j^{(P)}\|_2^2)$ are,

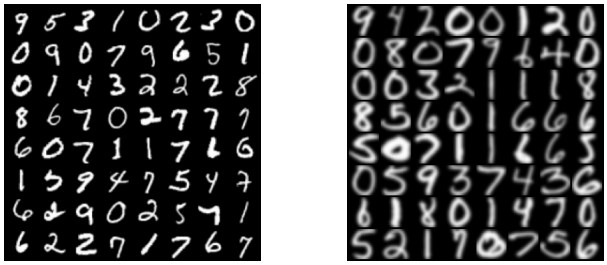
$$\mathbf{x}_i^\star = T_\gamma(\mathbf{x}_i^{(P)}) = \sum_{j=1}^{n_Q} \frac{\gamma_{ij}^\star}{p_i}\mathbf{x}_j^{(Q)}, \text{ and } \mathbf{y}_i^\star = T_\gamma(\mathbf{y}_i^{(P)}) = \sum_{j=1}^{n_Q} \frac{\gamma_{ij}^\star}{p_i}\mathbf{y}_j^{(Q)}, \tag{5.8}$$

especially, note that since $T_\gamma(\mathbf{y}_i^{(P)})$ is a convex combination of $y_j^{(Q)} \in \Delta_{n_c}$, it is also a soft probability vector.

Remark 5. *In the previous theorem, we introduced a rule for mapping features and labels between probability measures. While our goal was initially concerned with classification, this technique can be applied to* continuous labels *in general (e.g., regression).*

The consequence of theorem 9 is that, after calculating $\gamma^\star$, the transport of the support of P into Q can be performed by applying the barycentric mapping on the features **and** labels. In the latter case, this corresponds to the label propagation strategy proposed in [24].

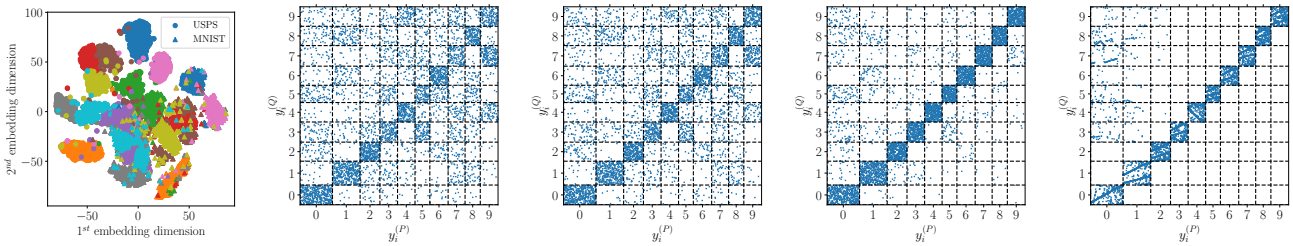
Example 9. *(MNIST and USPS) In this example, we evaluate the distributional shift occurring between two handwritten digit recognition datasets, namely MNIST [128] and USPS [119]. We show 64 samples from these two datasets in figure 5.1.*



(a) MNIST. (b) USPS.

Figure 5.1 – Handwritten digit recognition datasets used in this example. Note that USPS digits tend to occupy a larger space in the 32×32 image grid. Furthermore, these images are blurred due to up-scaling from 16×16 to 32×32. These operations make a distributional shift between the two datasets.

To analyze the distributional shift occurring between the two datasets, we first downsample each dataset to 5000 *samples. Then, we use the normalized pixel values between* $(0, 1)$ *as features. This generates* $1024-$*dimensional feature vectors. We then use t-SNE [129] for visualizing the data over* $\mathbb{R}^2$ *in figure 5.2 (a). From* $(b-e)$*, we show the transport plan* $\gamma^\star$ *obtained using the feature-label joint OT problem (equation 5.7) under a variable* β*. For* $\beta = 0$*, one retrieves the classical Kantorovich problem, which transports mass between classes. As* β *increases, transporting mass between different classes becomes increasingly costly, resulting in a more class-sparse transport plan. If* $\beta \to \infty$*, the feature term in the ground-cost becomes negligible, yielding sub-optimal matchings within the classes.*



(a) t-SNE. (b) $\beta = 0$. (c) $\beta = 1$. (d) $\beta = 10$. (e) $\beta = 10^{20}$.

Figure 5.2 – Visualization and optimal transport between MNIST and USPS.

5.1.3 Optimal Transport Dataset Distance

In the previous section, we considered a Wasserstein-like distance in the space $\mathbb{P}(\mathcal{X} \times \mathcal{Y})$. While using the squared of the Euclidean distance over soft labels allows us to derive interesting rules for mapping the labels between probability measures, this notion of dissimilarity does not take into account how the

different classes may be related. To further make this point, consider a 3-class classification problem. Then $d_{\mathcal{Y}}([1,0,0],[0,1,0]) = d_{\mathcal{Y}}([1,0,0],[0,0,1]) = d_{\mathcal{Y}}([0,1,0],[0,0,1])$, regardless of the relationship between these three classes.

A first solution to the aforementioned issue consists of using semantic relationships between the labels [130], where the authors explore learning with a Wasserstein objective. The distance between labels, i.e., predicted and ground-truth, are determined by the textual embeddings of their description. Arguably, this strategy is not widely applicable, as classes may not have a meaningful textual embedding. This motivates a second solution proposed by [25].

As discussed in [25], rather than relying on labels alone, a meaningful way of constructing a distance over $\mathcal{Y}$ is to consider the relationship between $\mathbf{x}$ and $\mathbf{y}$. This led the authors to consider the an embedding $y \mapsto P(X|Y=y)$, i.e., an embedding from $\mathcal{Y}$ into $\mathbb{P}(\mathcal{X})$. As a result, one may define $d_{\mathcal{Y}}$ through distances in $\mathbb{P}(\mathcal{X})$ (e.g., the $\alpha-$Wasserstein distance). As a result, one has the Optimal Transport Dataset Distance (OTDD),

Definition 28. *(Optimal Transport Dataset Distance [25]) Let $(\mathcal{X}, d_{\mathcal{X}})$ be a metric feature space, and $\mathcal{Y}$ be label space. Let $P, Q \in \mathbb{P}(\mathcal{X} \times \mathcal{Y})$ for which $P_y = P(X|Y=y)$ and $Q_y = Q(X|Y=y) \in \mathbb{P}(\mathcal{X})$ exist $\forall y \in \mathcal{Y}$. The OTDD is given by,*

$$OTDD_\alpha(P,Q)^\alpha = \min_{\gamma \in \Gamma(P,Q)} \int_{(\mathcal{X}\times\mathcal{Y})} (d_{\mathcal{X}}(x_1,x_2)^\alpha + \mathcal{W}_\alpha(P_{y_1}, Q_{y_2})^\alpha) d\gamma((x_1,y_1),(x_2,y_2)). \tag{5.9}$$

There is a challenge associated with equation 5.9, as the definition of the OTDD is hierarchical: one needs to solve sub-OT problems before calculating the OTDD itself. A straightforward idea is to approximate P_y empirically (c.f., Definition 8), which leads to a computational complexity of $\mathcal{O}(n^5 \log n)$. Alternatively, one can use Gaussians, for which OT has closed-form solution (c.f., Example 2), resulting in a complexity $\mathcal{O}(d^3)$. Choosing a Gaussian approximation leads to *sample-size free* complexity, at the cost of two, potentially important caveats: (i) modelling errors and (ii) estimation errors. The first comes from the fact that data is not necessarily Gaussian, and the second comes from the estimation of mean vectors and covariance matrices. *In a big data scenario*, one alleviates the second caveat and the sample-size free complexity becomes much more attractive.

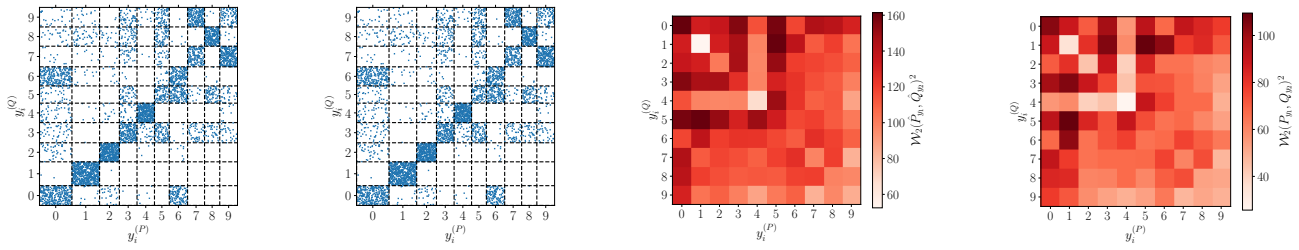
Example 10. *(MNIST and USPS, contd.) We continue on the MNIST and USPS example, illustrating the OT plan obtained by the OTDD, for which we test the empirical and Gaussian approximations of conditional measures P_y and Q_y. We show our overall comparison in figure 5.3.*

5.2 Multi-Source Domain Adaptation

In this section, we present the WBT algorithm [12, 13], which is a strategy for MSDA relying on the labeled Wasserstein barycenter of source domains. Differently from those publications, we incorporate recent advances proposed by us in [14].

We start our presentation of WBT by considering barycenters of labeled distributions, that is, given $\lambda = (\lambda_1, \cdots, \lambda_C) \in \Delta_C$ and $\mathcal{P} = \{\hat{P}_c\}_{c=1}^C$, with,

$$\hat{P}_c(\mathbf{x},\mathbf{y}) = \frac{1}{n_c} \sum_{i=1}^{n_c} \delta\Big((\mathbf{x},\mathbf{y}) - (\mathbf{x}_i^{(P_c)}, \mathbf{y}_i^{(P_c)}) \Big),$$



(a) γ (empirical) (b) γ (Gauss) (c) $\mathcal{W}_2(\hat{P}_{y_1}, \hat{Q}_{y_2})^2$ (d) $\mathcal{W}_2(P_{y_1}, Q_{y_2})^2$

Figure 5.3 – Comparison of OTDD plans for empirical (a, c) and Gaussian (b, d) approximations of probability measures.

we want to determine an empirical measure $B^\star$ with support $\mathbf{X}^{(B^\star)} \in \mathbb{R}^{n_B \times d}$ and $\mathbf{Y}^{(B^\star)} \in (\Delta_{n_c})^{n_B}$ such that,

$$(\mathbf{X}^{(B^\star)}, \mathbf{Y}^{(B^\star)}) = \underset{\substack{\mathbf{X}^{(B)} \in \mathbb{R}^{n_B \times d} \\ \mathbf{Y}^{(B)} \in (\Delta_{n_c})^{n_B}}}{\operatorname{argmin}} \sum_{c=1}^{C} \lambda_c \mathcal{JW}_{2,\beta} \left(\frac{1}{n_B} \sum_{i=1}^{n_B} \delta_{(\mathbf{x}_i^{(B)}, \mathbf{y}_i^{(B)})}, \hat{P}_c \right)^2.$$

As in chapter 3, section 3.3, we consider the cost function,

$$\mathcal{L}(\mathbf{x}_1^{(B)}, \cdots, \mathbf{x}_{n_B}^{(B)}, \mathbf{y}_1^{(B)}, \cdots, \mathbf{y}_{n_B}^{(B)}) = \sum_{c=1}^{C} \lambda_c \sum_{i=1}^{n_B} \sum_{j=1}^{n_c} \gamma^\star_{c,i,j} \left(\|\mathbf{x}_i^{(B)} - \mathbf{x}_i^{(P_c)}\|_2^2 + \beta \|\mathbf{y}_i^{(B)} - \mathbf{y}_i^{(P_c)}\|_2^2 \right) \quad (5.10)$$

Proposition 3. *Let $\mathcal{P} = \{\hat{P}_1, \cdots, \hat{P}_C\}$ be $C \geq 1$ empirical probability measures with weights $\mathbf{p}_c \in \Delta_{n_c}$ and support $(\mathbf{X}^{(P_c)}, \mathbf{Y}^{(P_c)})$ with $\mathbf{X}^{(P_c)} \in \mathbb{R}^{n_c \times d}$ and $\mathbf{Y}^{(P_c)} \in (\Delta_{n_c})^{n_c}$. Given a number of samples $n_B \in \mathbb{N}$ and $\beta > 0$, let $\hat{B}^\star$ be the empirical measure minimizer of $B \mapsto \sum_c \lambda_c \mathcal{JW}_{2,\beta}(B, \hat{P}_c)^2$, with weights $\mathbf{b} \in \Delta_{n_B}$ and support $\mathbf{X}^{(B)} \in \mathbb{R}^{n_B \times d}$. Then, the support of $\hat{B}^\star$ suffices,*

$$\begin{cases} \mathbf{X}^{(B)} = \sum_{c=1}^{C} \lambda_c T_{\gamma_c^\star}(\mathbf{X}^{(B)}), \\ \mathbf{Y}^{(B)} = \sum_{c=1}^{C} \lambda_c T_{\gamma_c^\star}(\mathbf{Y}^{(B)}) \end{cases} \quad (5.11)$$

where γ_c is the OT plan between $\hat{B}^\star$ and $\hat{P}_c$.

Proof. The proof is straightforward, and it takes the same steps as the proof of proposition 2 in chapter 3. Note that due the nature of the ground-cost C_{ij}, the terms depending on $\mathbf{y}_i^{(B)}$ are independent from $\mathbf{x}_i^{(B)}$, so they do not affect the feature terms. The first-order optimality conditions with respect $\mathbf{y}_i^{(B)}$ are,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{y}_i^{(B)}} = 2\beta \sum_c \lambda_c \sum_j \gamma^\star_{c,i,j} (\mathbf{y}_i^{(B)} - \mathbf{y}_j^{(P_c)}).$$

Solving for $\mathbf{y}_i^{(B)}$ gives us $\mathbf{y}_i^{(B)} = b_i^{-1} \sum_{c=1}^{C} \lambda_c \sum_j \gamma^\star_{c,i,j} \mathbf{y}_j^{(P_C)} = \sum_{c=1}^{C} \lambda_c T_{\gamma_c^\star}(\mathbf{y}_i^{(B)})$. Grouping the samples into the support matrix gives us the formula in equation 5.11. Note that since $b^{-1} \sum_j \gamma^\star_{c,i,j} \mathbf{y}_j^{(P_C)}$ is a convex combination of vectors in the simplex Δ_{n_c}, the resulting vector is in the simplex as well. □

As in chapter 3 and [26], proposition 3 results in a strategy for calculating empirical barycenters. One first initializes $\mathbf{X}^{(B_0)}$ and $\mathbf{Y}^{(B_0)}$ at random. Then, the following steps are repeated: (i) find $\gamma^{(c,it)}$ using joint OT. (ii) Update $\mathbf{X}^{(B_{it+1})}$ and $\mathbf{Y}^{(B_{it+1})}$ using equation 5.11. These steps are repeated until the loss in equation 5.10 is not updated or the algorithm reaches a maximum number of iterations. Similarly to [85], we also remark that this algorithm usually converges fast, as we analyze in the next example. The overall method is shown in Algorithm 6.

Algorithm 6: Labeled Wasserstein Barycenter

function `free_support_wbary`$(\{\mathbf{p}_c\}_{c=1}^{C}, \{\mathbf{X}^{(P_c)}, \mathbf{Y}^{(P_c)}\}_{c=1}^{C}, \lambda, \epsilon, \tau)$

`// Initialization.`

$\mathbf{x}_i^{(B)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $\mathbf{y}_i^{(B)} \sim \text{randint}(n_c)$, $L_0 = 0$

`// Updates.`

while $|L_{it} - L_{it-1}| > \tau$ **do**

  for $c = 1, \cdots, C$ **do**

    $\gamma^{(c,it)} = \text{JOT}\Big((\mathbf{X}^{(P_c)}, \mathbf{Y}^{(P_c)}); (\mathbf{X}^{(B_{it})}, \mathbf{Y}^{(B_{it})})\Big)$

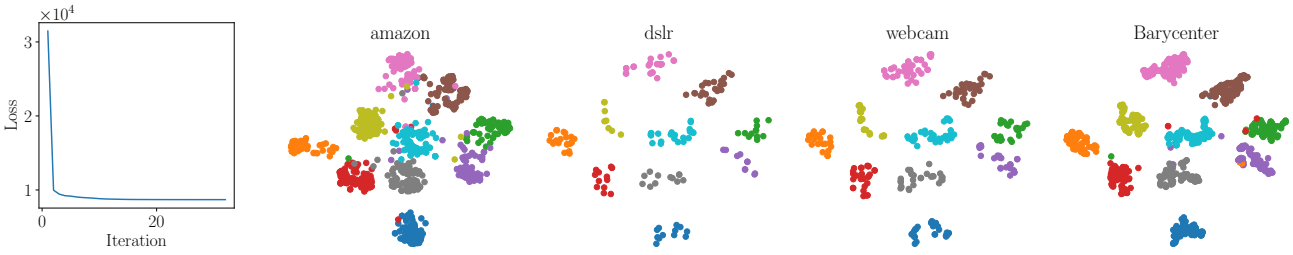
  $L_{it} = \sum_{c=1}^{C} \lambda_c \langle \gamma^{(c,it)}, \mathbf{C}^{(c)} \rangle_F$

  $\mathbf{X}^{(B_{it+1})} = \sum_{c=1}^{C} \lambda_c T_{\gamma^{(c,it)}}(\mathbf{X}^{(B_{it})})$

  $\mathbf{Y}^{(B_{it+1})} = \sum_{c=1}^{C} \lambda_c T_{\gamma^{(c,it)}}(\mathbf{Y}^{(B_{it})})$

return $(\mathbf{X}^{(B)}, \mathbf{Y}^{(B)})$

Example 11. *(Barycenter of Caltech-Office domains) In this example, we use the Caltech-Office 10 benchmark to illustrate the calculation of Wasserstein barycenters. We encode each image in this benchmark using DeCAF [131], which are 4096-dimensional feature vectors containing the activations of the 7-th fully connected layer. Here, we take the barycenter of the domains (A, D, W), where A stands for Amazon, D for dSLR, and W for Webcam. We visualize these doamins, as well as their barycenter, using t-SNE [129] on the concatenated feature matrices. We show our results in Figure 5.4.*



(a) Loss. (b) t-SNE embeddings.

Figure 5.4 – **Barycenter calculation on Caltech-Office 10 domains.** In (a), we show the loss L_{it} per iteration. In (b), we show the t-SNE embeddings for the 3 domains, (A, D, W), and their barycenter. In general, through our labeled Wasserstein barycenter, the barycentric measure retains the class separation observed in the various domains.

Example 12. *(MNIST-USPS barycenters) Continuing our example on digits, we calculate barycenters between these two datasets. We interpolate them through barycenters* $\mathcal{B}(\lambda; [\hat{Q}_M, \hat{Q}_U])$*, where* M *refers to MNIST and* U *to USPS. We show our results in Figure 5.5.*

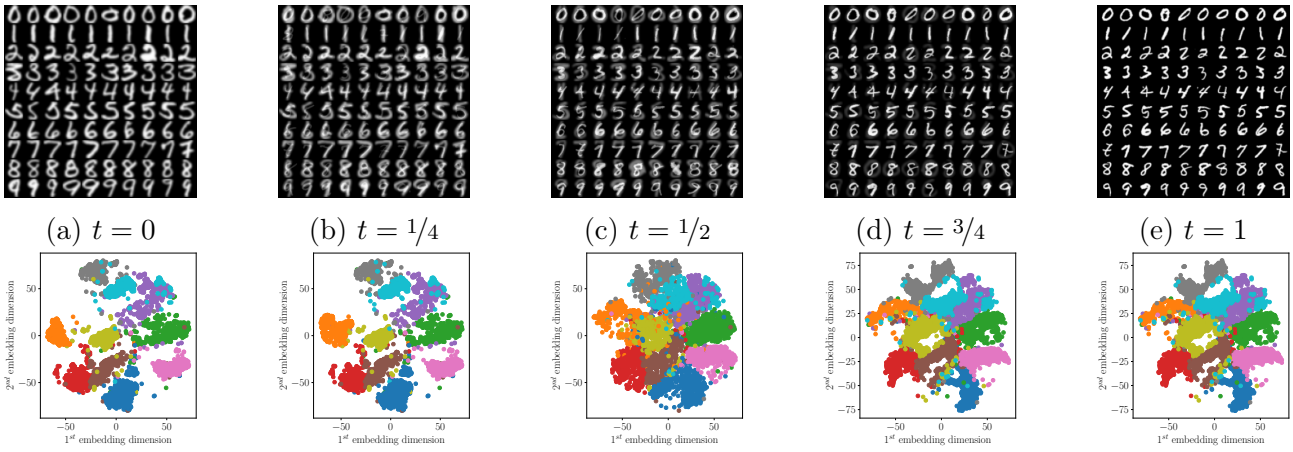


Figure 5.5 – Interpolation between MNIST and USPS through class-regularized Wasserstein Barycenters in image (a - e) and feature (f - j) spaces.

Furthermore, we compare the results acquired with labeled Wasserstein barycenters (Figure 5.5 and Algorithm 6) with those obtained through [26, Algorithm 2], which we show in Figure 5.6. Looking at the interpolations (a-e) in figure 5.6, one may note that the interpolation between MNIST and USPS mixes different digits.

The WBT algorithm takes as input N_S labeled probability measure in the form of its samples, that is, $\{\{(\mathbf{x}_i^{(Q_{S_\ell})}, \mathbf{y}_i^{(Q_{S_\ell})})\}_{i=1}^{n_\ell}\}_{\ell=1}^{N_S}$, and one unlabeled target probability measure, $\{\mathbf{x}_i^{(Q_T)}\}_{i=1}^{n_T}$. This yields $N_S + 1$ empirical measures. We then use the source domain samples to calculate the empirical Wasserstein barycenter $\hat{B} = \mathcal{B}(N_S^{-1}\mathbb{1}_{N_S}, \mathcal{Q}_S)$, where $\mathcal{Q}_S = \{\hat{Q}_{S_\ell}\}_{\ell=1}^{N_S}$. Note that we use uniform weights because we want each domain to contribute equally to the barycentric measure. However, $\hat{B}$ is not necessarily close to the target domain $\hat{Q}_T$. As a result, we employ a second step for mapping the support of $\hat{B}$ into that of $\hat{Q}_T$, e.g., using the barycentric mapping (c.f., equation 2.19),

$$T_\gamma(\mathbf{x}_i^{(B)}) = n_B \sum_{j=1}^{n_T} \gamma_{ij} \mathbf{x}_j^{(Q_T)}, \tag{5.12}$$

where $\gamma = \mathrm{OT}(\mathbf{X}^{(B)}, \mathbf{X}^{(Q_T)})$.

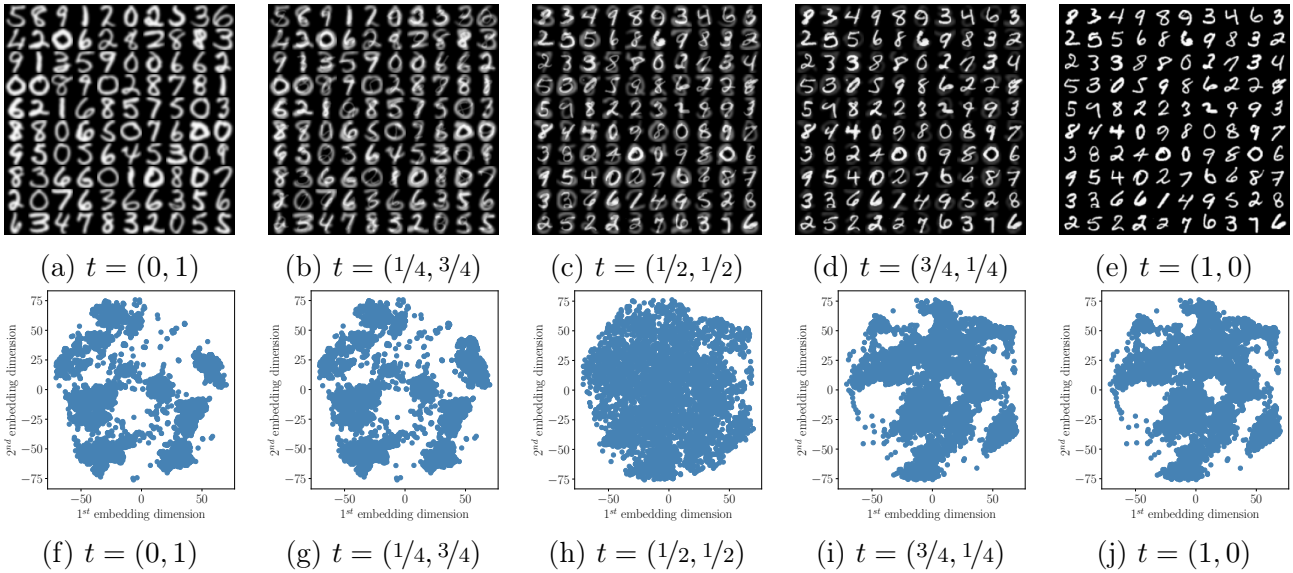


Figure 5.6 – Interpolation between MNIST and USPS through Wasserstein Barycenters. (a-e) interpolation in image space. (f-j) interpolation visualized through t-SNE.

Example 13. *(Gaussian Clusters) In this example, we illustrate how WBT works for MSDA. We consider a problem with 3 source domains, and a target domain, generated by applying an affine transformation to an initial measure with two classes, represented by two colors: blue (0) and red (1). While we show the labels from $\hat{Q}_T$ in Figure 5.7, these are not available for domain adaptation.*

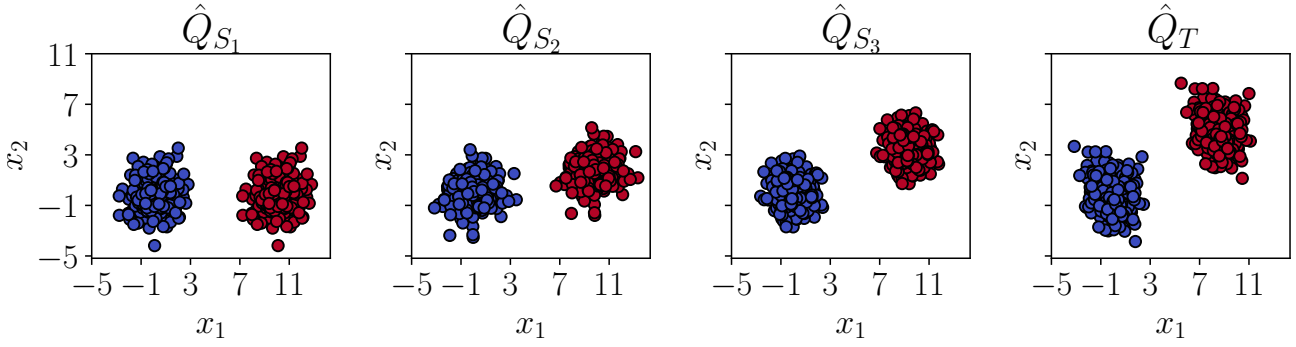
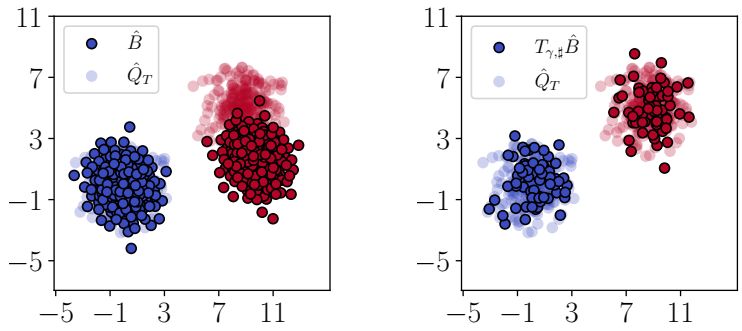


Figure 5.7 – Samples from the probability measures used in the toy example for MSDA.

The idea of WBT consists of two steps. First, one calculates the Wasserstein barycenter for source domain measures. The target domain measure is not included, since it does not have labeled samples. Note, however, that one still has a distributional shift between $\hat{B} = \mathcal{B}(N_S^{-1}\mathbb{1}_{N_S}, \mathcal{Q}_S)$ and $\hat{Q}_T$, that is, $\mathcal{W}_2(\hat{B}, \hat{Q}_T) \neq 0$. As a result, a second step is needed, i.e., applying 5.12 is needed. We exemplify our discussion in Figure 5.8.

Theoretical Justification. An interesting consequence of the bound in equation 4.12 is that it justifies using the Wasserstein barycenter as an intermediate domain in domain adaptation. First,



(a) $\hat{B}$ (b) $T_{\gamma,\sharp}\hat{B}$

Figure 5.8 – Illustration of the WBT method. In (a), we compare $\hat{Q}_T$ and $\hat{B}$ to the barycenter of source domain measures. Note that there is a residual shift between $\hat{B}$ and $\hat{Q}_T$. As a result, one needs to perform a second step, transporting the samples of $\hat{B}$ towards $\hat{Q}_T$ using the barycentric map T_γ, which is shown in (b).

note that $\mathcal{W}_1(P,Q) \leq \mathcal{W}_2(P,Q)$ [9]. Second, we can apply the triangle inequality on $\mathcal{W}_2$ so that,

$$\sum_{\ell=1}^{N_S} \lambda_\ell \mathcal{W}_2(\hat{Q}_{S_\ell}, \hat{Q}_T) \leq \sum_{\ell=1}^{N_S} \lambda_\ell \mathcal{W}_2(\hat{Q}_{S_\ell}, \hat{B}) + \mathcal{W}_2(\hat{B}, \hat{Q}_T).$$

Minimizing this bound with respect the empirical measure $\hat{B}$ corresponds exactly to the WBT strategy.

5.3 Conclusion

In this chapter, we introduced optimal transport distances that take into account label information associated with samples. This leads to a Wasserstein-like distance on the space $\mathbb{P}(\mathcal{X} \times \mathcal{Y})$, i.e., the space of joint probability measures over features $\mathbf{x} \in \mathcal{X}$ and labels $\mathbf{y} \in \mathcal{Y}$. Especially, we covered two distances: the joint Wasserstein distance $\mathcal{JW}_{\alpha,\beta}$ and the OTDD.

On one hand, our works rely primarily on the $\mathcal{JW}_{2,\beta}$, which uses one-hot encoded vectors for categorical class values. This choice offers two advantages. First, the resulting space of one-hot encoded vectors is continuous, i.e., the simplex Δ_{n_c}. In this sense, we are using soft-labels rather than categorical values, which provides us some smoothness in the label definition. Second, using the usual Euclidean distance over vectors in Δ_C allows us to retrieve the barycentric mapping of [23] and the label propagation strategy of [24] as first-order optimality conditions of $\mathcal{JW}^2_{2,\beta}$. However, the joint Wasserstein distance does not take into account the actual geometry of the label space. For instance, in a 3-class problem, the classes 0, 1 and 2 are all equidistant; regardless of how its samples are distributed.

On the other hand, the OTDD uses conditional distributions $P_y = P(X|Y=y)$ to represent labels. As a result, the labels are encoded via the map $y \mapsto P_y$, which maps $\mathcal{Y}$ to $\mathbb{P}(\mathcal{X})$. The OTDD distances are then calculated using the usual Euclidean distance over features, in addition to the Wasserstein $\mathcal{W}_2(P_{y_1}, P_{y_2})$. In contrast with the joint Wasserstein distance, the OTDD actually takes into account the geometry of classes, by considering how samples are distributed conditionally to its class label.

This remark is the main advantage of this distance. However, the OTDD considers discrete labels, which renders optimization with respect this metric challenging [132].

Based on probability metrics over the joint space $\mathcal{X} \times \mathcal{Y}$, we were able to extend previous algorithms [26, 85] for the calculation of the empirical Wasserstein barycenter. As we highlighted in chapter 3, this calculation seeks to determine the **support** of the barycenter. As a result, we need to determine matrices $\mathbf{X}^{(B)} \in \mathbb{R}^{n_B \times d}$ and $\mathbf{Y}^{(B)} \in (\Delta_{n_c})^{n_B}$, where n_B is the number of samples in the barycenter support, d is the dimensionality of features, and n_c is the number of classes. Note that, as previously mentioned, labels are treated continuously as soft-assignments, i.e., probabilities of belonging to a given class. These matrices are determined through fixed-point-like iterations based on the barycentric mapping (features) and label propagation (labels), which is shown in algorithm 3.

Based on our **labeled** barycenter algorithm, we devise an algorithm for domain adaptation. This strategy consists of two steps. First, we reduce the complexity of the adaptation task, by calculating the joint Wasserstein barycenter of source domain measures. Note that, upon doing this step, we are not guaranteed to retrieve the target domain measure. Therefore, there exists a residual domain shift between the barycenter and the target domain. We solve this shift via the OTDA strategy of [23], presented in chapter 2.

Chapter 6

Dataset Dictionary Learning

> Les mathématiciens n'étudient pas des objets, mais des relations entre objets; il leur est donc indifférent de remplacer ces objets par d'autre, pourvu que les relations ne changent pas. La matière ne leur importe pas, la forme seule les intéresse.
>
> Henri Poincaré

Contents

In this chapter, we introduce our technique called DaDiL, proposed in [14]. Given a set of probability measures (e.g., domains in MSDA), we want to learn how to interpolate between them in a Wasserstein space. We do so through *dictionary learning*, i.e., we learn a set of atoms that are weighted to get each measure in the learning set. Since we are dealing primarily with empirical measures, our set of atoms will be a set of synthetic, empirical measures, weighted in a Wasserstein space through Wasserstein barycenters.

Besides our DaDiL framework, we introduce tools for performing barycentric coordinates regression over empirical measures. This corresponds to an adaptation of the problem first introduced in [28],

for histograms. With respect our paper [14], this chapter includes a set of examples illustrating our methods. From a more general perspective, we introduce a novel problem in OT, where we learn a set of probability measure such that their *Wasserstein hull* (see definition 17 in Chapter 3) contains the probability measures at hand. We illustrate this idea in Figure 6.1.

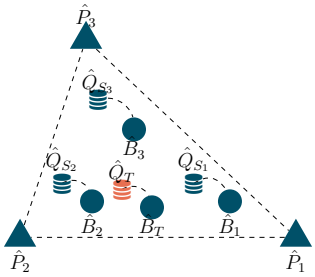


Figure 6.1 – Dataset dictionary learning framework. We approximate probability measures $\mathcal{Q} = \{Q_\ell\}_{\ell=1}^N$ as interpolations of learned atoms $\mathcal{P} = \{P_c\}_{c=1}^C$.

This chapter is divided as follows. Section 6.1 presents previous works on dictionary learning and barycentric coordinates regression over histograms. Section 6.2 presents our tools for dataset dictionary learning and barycentric coordinates regression, proposed by us in [14]. Finally, section 6.3 concludes this chapter.

6.1 Histogram Dictionary Learning and Coordinates Regression

In this section we present the works of [28] and [27]. The first paper introduced the concept of barycentric coordinates regression, i.e., given a set of measures $\mathcal{P} = \{P_c\}_{c=1}^C$ and another measure Q, how to estimate $\lambda^\star$ such that $Q \overset{\mathcal{W}_2}{\approx} \mathcal{B}(\lambda^\star, \mathcal{P})$. The second paper introduced a strategy for expressing probability measures $\mathcal{Q} = \{Q_\ell\}_{\ell=1}^N$ as a Wasserstein barycenter of learned atoms $\mathcal{P} = \{P_c\}_{c=1}^C$.

Originally, these works were designed to work over histograms, i.e., empirical probability measures with a fixed-support. As we present in section 6.2, our main contribution is defining an analogous problem with respect free-support empirical measures. As we cover in the upcoming section, this shift in paradigm completely changes the optimization problem.

6.1.1 Barycentric Coordinates Regression

Given a set of probability measures $\mathcal{P} = \{P_c\}_{c=1}^C$, $P_c \in \mathbb{W}_2(\mathcal{X})$, Wasserstein barycenters can be understood as a map between the $C-$simplex Δ_C and the space of probability measures over $\mathcal{X}$, $\mathbb{W}_2(\mathcal{X})$. In [28], the authors proposed to study the inverse problem. Given a fixed family $\mathcal{P}$ of measures in $\mathbb{W}_2(\mathcal{X})$, and a new $Q \in \mathbb{W}_2(\mathcal{X})$, the authors seek to determine,

$$\lambda^\star = \underset{\lambda \in \Delta_C}{\operatorname{argmin}} \mathcal{W}_2(Q, \mathcal{B}(\lambda, \mathcal{P}))^2.$$

The authors in [28] considered this problem over fixed-support empirical probability measures, i.e., histograms. Hence, for a fixed support $\mathbf{X} \in \mathbb{R}^{n \times d}$, the measures $\mathcal{P}$ are represented through a matrix $\mathbf{P} \in (\Delta_n)^N$, and Q is represented through a vector $\mathbf{q} \in \Delta_n$. This simplifies the previous optimization problem to,

$$\lambda^\star = \underset{\lambda \in \Delta_C}{\operatorname{argmin}} \mathcal{W}_2(\mathbf{q}, \mathcal{B}(\lambda, \mathbf{P})), \tag{6.1}$$

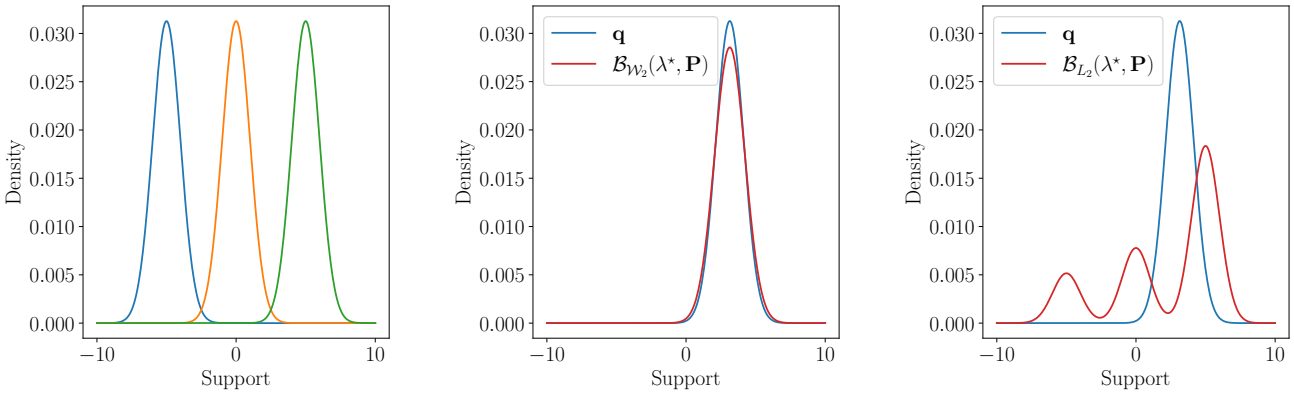
where $\mathcal{B}(\lambda, \mathbf{P})$ is calculated using a fixed-support barycenter algorithm. In [28], equation 6.1 is efficiently minimized by using the Sinkhorn algorithm [47] in place of $\mathcal{W}_2$, and using IBP [50] for computing barycenters. A limitation of the approach in equation 6.1 is that, when Q is far from the *Wasserstein barycentric hull* (see e.g., Definition 17) $\mathcal{M}(\mathcal{P})$, the Wasserstein Barycentric coordinates Regression (WBR) algorithm may lead to poor reconstructions of the original histograms. A possible solution to this issue is to learn the set of measures $\mathcal{P}$ through dictionary learning.

Example 14. *(Barycentric Regression over Histograms) In this example, we consider a problem where* $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$*, where* $\mathcal{X} = [-10, 10]$*, discretized uniformly over* 256 *bins, and,*

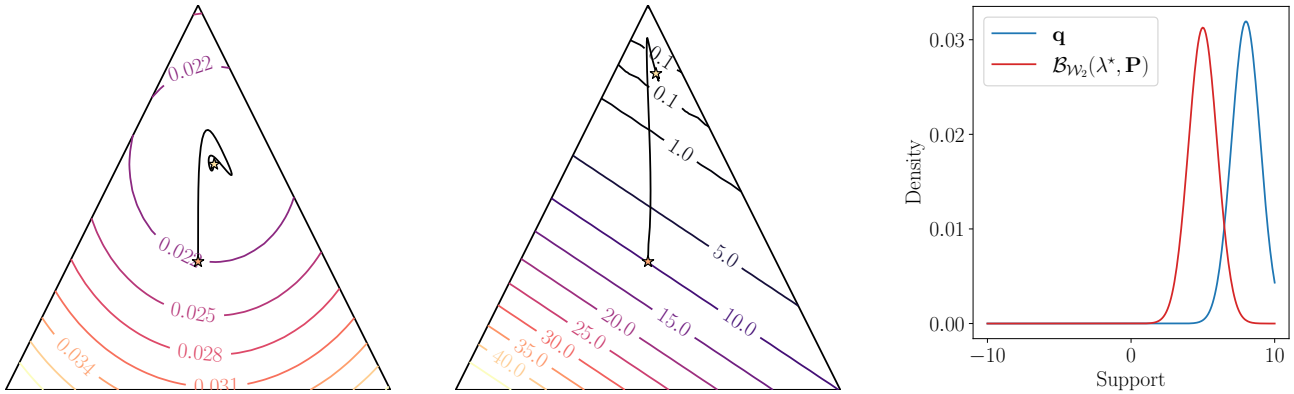
$$p_{\ell,i} = \frac{1}{Z_\ell} e^{-(x_i - \mu_\ell)^2}, \text{ and } Z_\ell = \sum_{i=1}^{n} p_{\ell,i},$$

for $\mu_1 = -7$*,* $\mu_2 = 0$*, and* $\mu_3 = +7$*. The measures in* $\mathbf{P}$ *are shown in figure 6.2 (a). We optimize the barycentric regression problem in equation 6.1 through gradient descent with a change of variables, i.e., we optimize with respect* $\tilde{\lambda}$ *such that* $\lambda = softmax(\tilde{\lambda})$*.*

We compare the $\mathcal{W}_2-$*barycentric regression problem to a* L_2 *barycentric regression, where* L_2 *denotes the squared deviation between* P *and* Q*, i.e.,* $L_2(P, Q) = \int_{\mathcal{X}} (dP - dQ)^2$*. For both problems, we use gradient descent with the Adam optimizer algorithm [133]. In the discrete case this corresponds to the squared Euclidean distance between their histograms, i.e.,* $\|\mathbf{p} - \mathbf{q}\|_2^2$*. Naturally, as we show in Figure 6.2 (c), this geometry does not yield faithful reconstructions based on its barycenter. We show in Figures 6.2 (d, e) a summary of the optimization path taken by the WBR optimization process, and in (b, c), the corresponding reconstructions. As we mentioned previously, if* $\mathbf{q}$ *does not belong to the barycentric hull of* $\mathbf{P}$*, the WBR technique may yield poor reconstructions, as we show in Figure 6.2 (f).*



(a) Measures $\mathbf{P}$ (b) Faithful reconstruction. (c) L_2-barycenter



(d) Optimization (L_2). (e) Optimization ($\mathcal{W}_2$). (f) Unfaithful reconstruction.

Figure 6.2 – Overview of the barycentric regression problem with respect $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3)$ (a) and a measure $\mathbf{q}$ (b, in blue). Using a Wasserstein geometry effectively reconstructs the measure $\mathbf{q}$, in contrast with the L_2 geometry. In (d) and (e), we show the optimization path of λ as equation 6.1 is minimized with respect $\mathcal{W}_2$ and L_2. In (f), we show an example of $\mathbf{q}$ that cannot be reconstructed using $\mathcal{B}(\lambda, \mathbf{P})$.

6.1.2 Dictionary Learning

Dictionary learning is a representation learning technique that expresses a collection of vectors $\{\mathbf{x}_\ell\}_{\ell=1}^N$, $\mathbf{x}_\ell \in \mathbb{R}^d$ through a set of atoms $\mathbf{P} \in \mathbb{R}^{c\times d}$ and weights $\mathbf{A} \in \mathbb{R}^{n\times c}$, $\lambda_\ell \in \mathbb{R}^C$. Mathematically,

$$\underset{\mathbf{A},\mathbf{P}}{\operatorname{argmin}} \mathcal{L}(\mathbf{X}, \mathbf{AP}) + \Omega(\mathbf{A}, \mathbf{P}), \tag{6.2}$$

where ℓ denotes a loss such as the Frobenius norm between $\mathbf{X}$ and $\mathbf{AP}$. In turn, $\Omega(\mathbf{A}, \mathbf{P})$ denotes a regularization term involving the representations and atoms. A common choice for regularization is the nuclear norm $\|\mathbf{A}\|_0$, which induces sparse representations for the data matrix. For instance, a popular problem that fits into the framework of equation 6.2 is the Principal Component Analysis (PCA). We exemplify these ideas on MNIST, in Example 15.

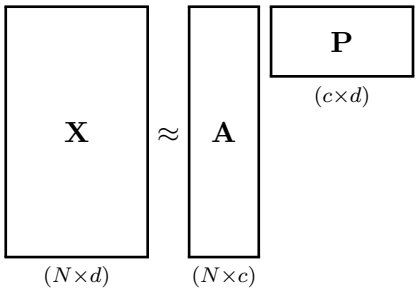


Figure 6.3 – Linear decomposition of a matrix $\mathbf{X}$ in terms of a dictionary $\mathbf{P} \in \mathbb{R}^{d\times c}$. Each row $\mathbf{a}_i$ of the matrix $\mathbf{A}$ corresponds to the representation for each row $\mathbf{x}_i$.

Example 15. *(PCA as Dictionary Learning on MNIST) The PCA is a dimensionality reduction technique that seeks a projection matrix $\mathbf{W} \in \mathbb{R}^{c\times d}$ retaining as much data variance as possible. From a projection-reconstruction perspective, this problem can be phrased in terms of the quadratic error,*

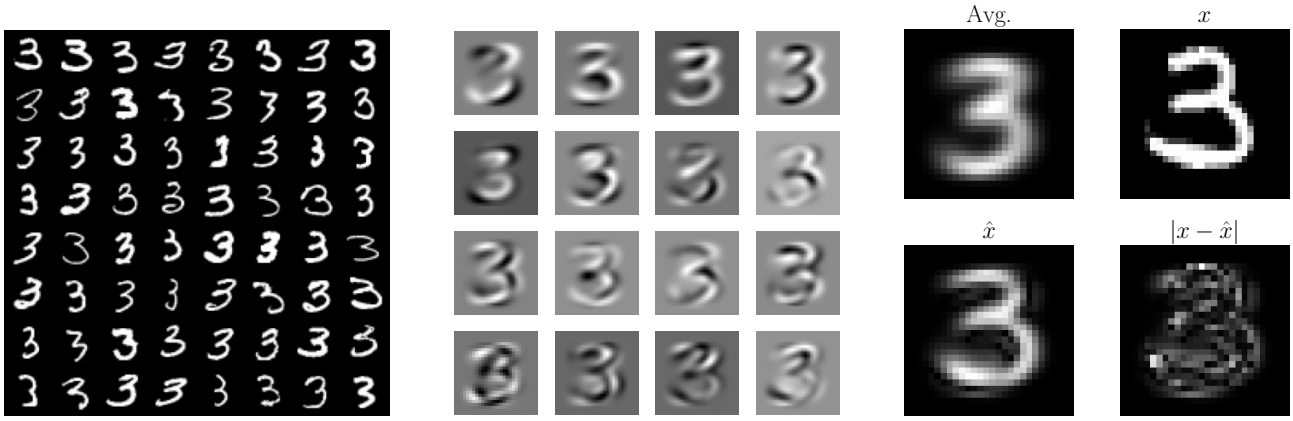
$$\underset{\mathbf{W}\mathbf{W}^T=\mathbf{I}_c}{\operatorname{argmin}} \mathcal{L}(\mathbf{X}, \mathbf{W}) := \|\mathbf{X} - \mathbf{X}\mathbf{W}^T\mathbf{W}\|_F^2. \tag{6.3}$$

Here, one may identify $\mathbf{AP} = (\mathbf{X}\mathbf{W}^T)\mathbf{W}$, which means that the matrix of representations consists of $\mathbf{A} = \mathbf{X}\mathbf{W}^T \in \mathbb{R}^{n\times c}$ and the dictionary is the projection matrix, $\mathbf{W} \in \mathbb{R}^{c\times d}$. From a geometric perspective, the PCA creates a dictionary consisting of the $c-$eigenvectors with the largest corresponding eigenvalues. The representations correspond to the projections of data points into the $c-$dimensional linear subspace. Next we exemplify these ideas on the MNIST dataset in figure 6.4.

Due its nature, optimal transport intersects the problem in equation 6.2 when the data points are understood as probability measures. In the literature [134, 27], this is the case when $\mathbf{x}_\ell$ takes the form of a histogram, i.e., $\mathbf{x}_\ell \in \Delta_d$. In this case, other kinds of losses may preferred over the squared error in equation 6.3, such as the Kullback-Leibler divergence. A solution coming from OT consists of using the Wasserstein distance or the Sinkhorn divergence.

A second axis of complexity when data are histograms comes from the reconstruction process, i.e., $\mathbf{X} \approx \mathbf{AP}$. If the elements in $\mathbf{X}$ are histogram, this matrix has an intrinsic structure: it is non-negative, and its rows sum to one. As a result, one needs to impose additional constraints, i.e., $\mathbf{AP} \in (\Delta_d)^N$. Minimizing equation 6.2 under these constraints is known in the literature as Nonegative Matrix Factorization (NMF) [135]. In this context, the linear operation $\mathbf{AP}$ may lose its sense as well.

If we assume $\mathbf{A} \in (\Delta_d)^N$, i.e., $\mathbf{a}_\ell \in \Delta_d$, one can interpret the matrix product $\mathbf{AP}$ as a weighted sum of dictionary elements, i.e., $\hat{\mathbf{x}}_\ell = \sum a_{\ell,j}\mathbf{p}_j$. Since $\sum_c a_{\ell,c} = 1$, this operation can be understood as



(a) Images used for PCA (b) Dictionary $\mathbf{P} = \mathbf{W}$ (c) Reconstruction of a digit.

Figure 6.4 – Summary of the application of PCA to learn a dictionary over MNIST digits "3".

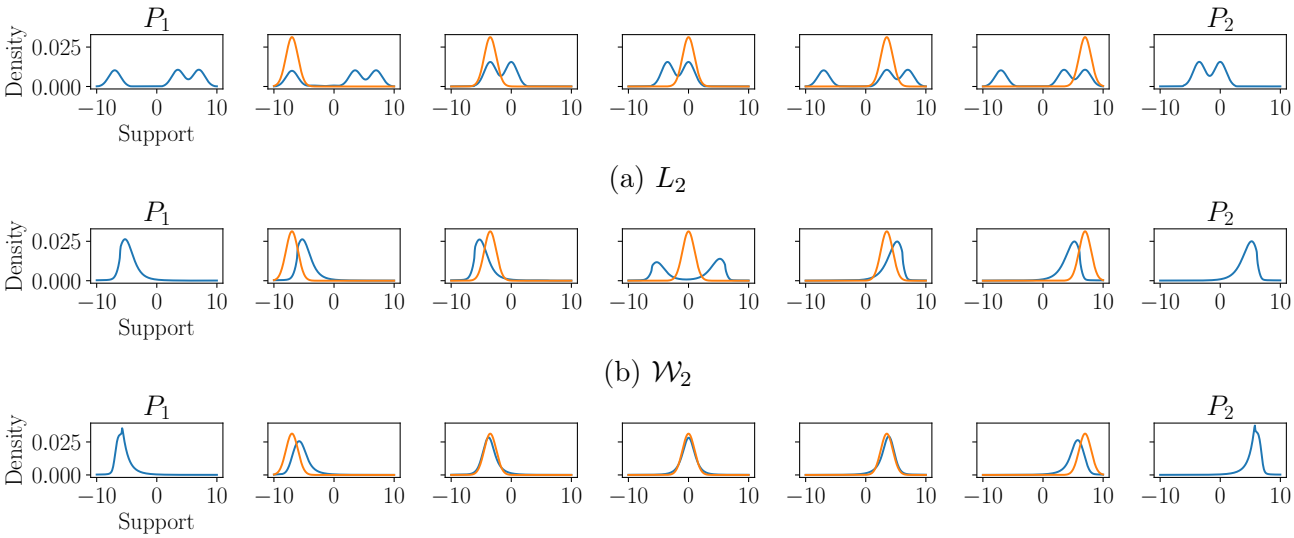
a barycenter in an Euclidean space. A natural extension of this principle is considering a Wasserstein geometry over the histograms, i.e. [27]

$$\underset{\mathbf{A},\mathbf{P}}{\operatorname{argmin}} \sum_{\ell=1}^{N} \mathcal{W}_2(\mathbf{x}_\ell, \mathcal{B}(\mathbf{a}_\ell, \mathbf{P})) + \Omega(\mathbf{A}, \mathbf{P}), \tag{6.4}$$

where $\mathcal{B}(\mathbf{a}, \mathbf{P})$ denotes the Wasserstein barycenter of histograms in $\mathbf{P}$, weighted by $\mathbf{a}$ (see chapter 3). Due the inherent geometrical nature of equation 6.4, the representations are now understood as *barycentric coordinate vectors*, i.e., they determine the position of the barycenter $\mathcal{B}(\mathbf{a}_\ell, \mathbf{P})$ with respect the atoms $\mathbf{P} = (\mathbf{p}_1, \cdots, \mathbf{p}_c)$. We now present a comparison of dictionary learning methods to exemplify why using a Wasserstein geometry is beneficial.

Example 16. *(Dictionary learning over histograms) Here, we extend the previous example by considering a dictionary learning problem over a set of* 5 *probability measures* $\mathbf{q}_\ell$*, with means taken uniformly over* $[-7, 7]$*. We compare the learning of 3 types of dictionaries. For the first two methods, we consider the problem in equation 6.2 with losses* $\mathcal{L}(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_2^2$ *and* $\mathcal{L}(\mathbf{p}, \mathbf{q}) = \mathcal{W}_2(\mathbf{p}, \mathbf{q})$ *using the dual Kantorovich formulation (see equation 2.20). The third method consists of using the Wasserstein Dictionary Learning (WDL) strategy described in the previous discussion, especially equation 6.4.*

We implement these methods using Pytorch and POT [41]. The resulting dictionaries and reconstructions are shown in Figure 6.5. Naturally, WDL has an advantage over the other two methods, as the Wasserstein geometry is the most adequate to express the diversity in the measure set $\mathcal{Q}$*. This remark illustrates an important challenge related to dictionary learning, i.e., how to adequately describe the objects participating in the learning process. Arguably, using OT bias the kinds of dictionaries learned by the dictionary learning problem.*



(a) L_2

(b) $\mathcal{W}_2$

(c) WDL

Figure 6.5 – Summary of dictionary learning methods. In (a) and (b), we decompose the signals in $\mathbf{X}$ as the product $\mathbf{AP}$. This choice yields poor reconstructions of the data in $\mathbf{X}$. In contrast, using the Wasserstein geometry (c) yields the correct reconstructions for the data.

6.2 Dataset Dictionary Learning and Coordinates Regression

Given this initial overview of dictionary learning, we are now interested on how to apply these ideas for domain adaptation. Here, an important paradigm shift is needed. Indeed, in MSDA one has free-support measures rather than histograms, that is, we represent these measures via their samples, rather than weights over a fixed grid. In this sense, our proposed algorithm makes a series of analogies with the work of [27] and dictionary learning in general.

Table 6.1 – Overview of analogies between different works in dictionary learning.

Concept	Symbol	Classic DiL	WDL [27]	DaDiL (ours)
Data	$\mathbf{x}_\ell$, or $\hat{Q}_\ell$	Vectors	Histograms	Point Clouds
Atom	$\mathcal{P}$	Vectors	Histograms	Point Clouds
Representation	$\mathcal{A}$	Vectors	Barycentric Coordinates	Barycentric Coordinates
Reconstruction	$\mathcal{B}$	Vectors	Histograms	Point Clouds

6.2.1 Barycentric Coordinates Regression

Our first step into building an intuition for DaDiL consists on the Barycentric Coordinates Regression (BCR) problem. This problem was previously considered in section 6.1.1. As we discuss in

definition 17, a set of measures $\mathcal{P}$ in $\mathbb{W}_2(\mathcal{X})$ define a *barycentric hull* $\mathcal{M}(\mathcal{P}) = \{\mathcal{B}(\lambda, \mathcal{P}) : \lambda \in \Delta_C\}$. From the perspective of MSDA, we assume a set of empirical measures $\mathcal{Q}_S = \{\hat{Q}_{S_\ell}\}_{\ell=1}^{N_S}$, so that,

$$\hat{Q}_{S_\ell}(\mathbf{x}) = \frac{1}{n}\sum_{\ell=1}^{N_S} \delta(\mathbf{x} - \mathbf{x}_i^{(Q_{S_\ell})}).$$

We can then search the space $\mathcal{M}(\mathcal{Q}_S)$ for the empirical measure that best approximates $\hat{Q}_T$, that is,

$$\lambda^\star = \underset{\lambda \in \Delta_C}{\operatorname{argmin}}\, \mathcal{W}_2(\hat{Q}_T, \mathcal{B}(\lambda, \mathcal{Q}_S)). \tag{6.5}$$

For finding $\lambda^\star$, we design a simple projected gradient descent algorithm based on the Wasserstein distance between $\hat{Q}_T$ and the barycenter $\mathcal{B}(\lambda, \mathcal{Q}_S)$ calculated using Algorithm 6. This is shown in Algorithm 7 below.

Algorithm 7: WBR

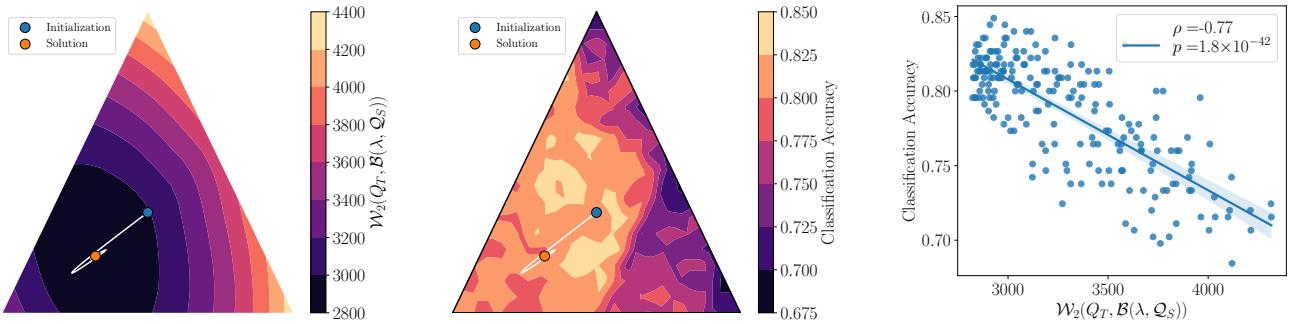
```
function fullbatch_wbr(Q_S, η, N_iter)
    // Initialization
    λ_0 = N_S^{-1}
    // Optimization
    for it = 1, ··· , N_iter do
        B̂ ← B(λ, Q_S)
        L ← W_2(Q̂_T, B̂)
        λ_{it+1} ← π_{Δ_{N_S}}(λ_it − η ∂L/∂λ)
    return λ*
```

As noted by [28], the barycentric regression algorithm leads to poor reconstructions, when $\hat{Q}_T$ is far from the distributions in $\mathcal{Q}_S$. Instead of using $\mathcal{Q}_S$ in the BCR problem, an alternative is to learn λ jointly with a set of atoms $\mathcal{P}$. This motivates our main contribution in [14]. We illustrate this remark on the next example.

Example 17. *(WBR on the Caltech-Office benchmark) In this example, we explore the BCR problem over a high-dimensional dataset. Especially, we explore the generalization of barycenters $\mathcal{M}_{\mathcal{JW}_{2,\beta}} = \{\mathcal{B}(\lambda, \mathcal{Q}_S) : \lambda \in \Delta_{N_S}\}$ to the target domain $\hat{Q}_T$. For this experiment we use the Caltech-Office benchmark presented in chapter 4. As features, we do as in [13] and consider $4096-$dimensional DeCAF feature vectors [131].*

The Caltech-Office benchmark has 4 domains in total. We use the domain Caltech as the target, and the other 3 domains (Amazon, Webcam and dSLR) as sources. This allows for an easy visualization of the simplex Δ_3 in $\mathbb{R}^2$. In this experiment, we are interested in two quantities, namely, the generalization performance, and the distance between measures. For the first case, we measure the classification accuracy of a $1-$neighbors classifier trained with data from the barycenters $\mathcal{B}(\lambda, \mathcal{Q}_S)$. In the second case, we measure the Wasserstein distance between $\hat{Q}_T$ and $\mathcal{B}(\lambda, \mathcal{Q}_S)$.

We show our results in Figure 6.6. Note that, while the level curves of objective function are quadratic, the classification accuracy is much more complex. This is somewhat expected, as the



(a) Loss. (b) Classification accuracy. (c) Regression analysis.

Figure 6.6 – Barycentric coordinates regression results on the Caltech-Office benchmark. We use as adaptation task $(A, D, W) \rightarrow C$. In (a), we show the optimization loss taken by the BCR algorithm, alongside a heatmap indicating the loss over the simplex Δ_3. In (b), we show the associated classification accuracy on Q_T, using data from $\mathcal{B}(\lambda, \mathcal{P})$. In (c), we show a correlation analysis between the 2−Wasserstein distance between $\hat{Q}_T$ and $\mathcal{B}(\lambda, \mathcal{Q}_S)$, and the classificatoin accuracy of $\mathcal{B}(\lambda, \mathcal{Q}_S)$ over Q_T.

2−*Wasserstein distance does not take into account the labels of the measures. Nonetheless, one can expect some level of correlation between the two quantities, as hinted by theoretical results (e.g., theorem 6), as is shown in Figure 6.6 (c).*

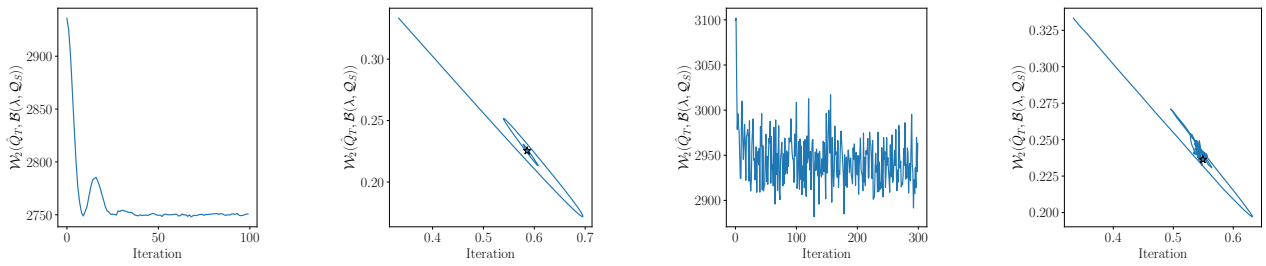
Alternatively, if the measures $\hat{Q}_T$ and in $\mathcal{Q}_S$ have many samples, it may be interesting to perform the optimization in equation 6.5 via mini-batches. This strategy is detailed in Algorithm 8.

Algorithm 8: Mini-batch WBR

```
function minibatch_wbr(Q_S, η, N_iter)
    // Initialization
    λ_0 = N_S^{-1}
    // Optimization
    for it = 1, ··· , N_iter do
        for B = 1, ··· , M do
            // Sampling (sources)
            for ℓ = 1, ··· , N_S do
                X^{(Q_{S_ℓ})} ← {x_i^{(Q_{S_ℓ})}}_{i=1}^{n_b}
            // Barycenter
            X^{(B)} ← B(λ, Q_S)
            // Sampling (target)
            X^{(Q_T)} ← {x_i^{(Q_T)}}_{i=1}^{n_b}
            L ← W_2(Q̂_T, B̂)
            λ_{it+1} ← π_{Δ_{N_S}}(λ_it − η ∂L/∂λ)
    return λ*
```

The main idea of algorithm 8 is estimating the Wasserstein distance in equation 6.1 using a handful of samples that fit into memory. Naturally, the same issues that come with mini-batch OT also affect the mini-batch version of the WBR problem. We refer readers to Chapter 2 and [61] for further discussion. In the next example, we exemplify the possible issues with using mini-batches.

Example 18. *(Mini-batch WBR on Caltech-Office benchmark) Next, we consider the effect of using mini-batch OT in the optimization process of the WBR problem. We show our results in Figure 6.7. We use a mini-batch of size* 300*. Note that, by using mini-batches, the loss becomes much more noisy. This reflects itself in the trajectory taken by the barycentric coordinates, but ultimately, the algorithm converge to close-by points.*



(a) Loss. (b) Trajectory. (c) Loss. (d) Trajectory.

Figure 6.7 – Fullbatch vs. Minibatch optimization strategies for solving the WBR problem. (a) and (b) show the loss and optimization trajectory of fullbatch WBR, whereas (c) and (d) show the respective figures for mini-batch OT.

6.2.2 Dictionary Learning

Given the previous discussion, we propose a novel dictionary learning problem involving empirical probability measures with a free-support. In this sense, we have a set of atoms $\mathcal{P} = \{\hat{P}_c\}_{c=1}^{C}$, each parametrized by its support, $\mathbf{X}^{(P_c)} \in \mathbb{R}^{n \times d}$ and $\mathbf{Y}^{(P_c)} \in \mathbb{R}^{n \times n_c}$, so that,

$$\hat{P}_c(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{c=1}^{C} \delta\Big((\mathbf{x}, \mathbf{y}) - (\mathbf{x}_i^{(P_c)}, \mathbf{y}_i^{(P_c)})\Big),$$

where $n \in \mathbb{N}$ and $C \in \mathbb{N}$ are hyper-parameters that control the dictionary's complexity.

Let $\mathcal{Q} = \mathcal{Q}_S \cup \{\hat{Q}_T\}$, for $\mathcal{Q}_S = \{\hat{Q}_{S_\ell}\}_{\ell=1}^{N_S}$. Our goal is to approximate each $\hat{Q}_\ell \in \mathcal{Q}$ through a barycenter of the atoms in $\mathcal{P}$. To do so, we introduce a set of barycentric coordinate vectors $\Lambda = \{\lambda_\ell\}_{\ell=1}^{N_S+1}$, which will locate each measure inside the interpolation space $\mathcal{M}(\mathcal{P}) = \{\mathcal{B}(\lambda, \mathcal{P}) : \alpha \in \Delta_C\}$. We thus define our dictionary $(\Lambda^\star, \mathcal{P}^\star)$ through,

$$(\Lambda^\star, \mathcal{P}^\star) = \underset{\Lambda, \mathcal{P}}{\operatorname{argmin}}\, \mathcal{L}(\Lambda, \mathcal{P}) := \mathcal{W}_2(\hat{Q}_T, \mathcal{B}(\lambda_T, \mathcal{P}))^2 + \sum_{\ell=1}^{N_S} \mathcal{JW}_{2,\beta}(\hat{Q}_{S_\ell}, \mathcal{B}(\lambda_\ell, \mathcal{P}))^2. \tag{6.6}$$

Note that the loss $\mathcal{L}(\Lambda, \mathcal{P})$ is a function of the barycentric coordinate matrix $\Lambda \in (\Delta_C)^{N_S+1}$, as well as the supports of measures in $\mathcal{P}$. Before discussing how to optimize equation 6.6, let us remark that we need to enforce two types of constraints, as $\mathbf{y}_i^{(P_c)} \in \Delta_{n_c}$ and $\lambda_\ell \in \Delta_C$. We choose to enforce each of these differently [1].

On the one hand, in the first case, we use a change of variables,

$$y_{ic}^{(P_c)} = \text{softmax}(\mathbf{u}_i^{(P_c)})_c = \frac{e^{u_{ic}^{(P_c)}}}{\sum_{c'=1}^{n_c} e^{u_{ic'}^{(P_c)}}},$$

where $y_{ic}^{(P_c)}$ denotes the probability of $i-$th sample of $\hat{P}_c$ belonging to the $c-$class. This strategy is similar to what was done in [27]. See, for instance [27, equation 13]. On the other hand, in the second case, similarly to [113], we project λ_ℓ into the simplex after each optimization step,

$$\text{proj}_{\Delta_C}(\lambda_\ell) = \underset{\lambda \in \Delta_C}{\text{argmin}} \|\lambda - \lambda_\ell\|_2.$$

In the following, we discuss two strategies for optimizing equation 6.6: full, and mini-batch DaDiL. Note that, in our publication [14], we employ mini-batch OT [136].

Algorithm 9: DaDiL

```
function dadil(Q_S, Q̂_T, N_it, C, η)
    // Initialization
    x_j^(P_c) ~ N(0, I_d)
    u_j^(P_c) ~ N(0, I_{n_c})
    λ_ℓ ← 1_C / C
    for it = 1, ··· , N_it do
        L ← 0
        // Change of variables
        Y^(P_c) ← softmax(U^(P_c))
        // Dictionary learning (sources)
        for ℓ = 1, ··· , N_S do
            (X^(B_ℓ), Y^(B_ℓ)) ← B(λ_ℓ, P)
            L ← L + JW_{2,β}(Q̂_{S_ℓ}, B̂_ℓ)^2
        // Dictionary learning (target)
        (X^(B_T), Y^(B_T)) ← B(λ_T, P)
        L ← L + W_2(Q̂_T, B̂_T)^2
        x_j^(P_c) ← x_j^(P_c) − η ∂L/∂x_j^(P_c)
        u_j^(P_c) ← u_j^(P_c) − η ∂L/∂u_j^(P_c)
        λ_ℓ ← π_{Δ_C}(λ_ℓ − η ∂L/∂λ_ℓ)
    return Λ*, P*
```

1. Initially, we tried to enforce $\lambda_\ell \in \Delta_C$ through a change of variables, as in [27]. While in some cases this worked, we oftentimes verified a collapse of weight vectors towards a single atom. Using an orthogonal projection on this variable yielded more stable results overall.

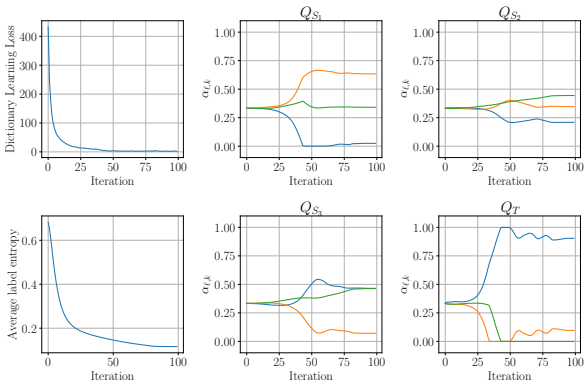
Full-batch strategy. Under this strategy, we consider minimizing equation 6.6 by estimating the loss $\mathcal{L}(\Lambda, \mathcal{P})$ over the complete available data. This choice implies a few things. Let us fix, without loss of generality, the number of samples in the support of atoms $\mathcal{P}$, $n \geq \max\{n_{S_1}, \cdots, n_T\}$. We then calculate a barycenter $\hat{B}_\ell = \mathcal{B}(\lambda_\ell, \mathcal{P})$ with n points in its support as well. This step has computational, and storage complexity of $\mathcal{O}(Cn^3 \log n)$ and $\mathcal{O}(Kn^2)$, respectively. Based on this barycenter, we calculate its contribution to the loss, i.e., $\mathcal{W}_2(\hat{Q}_T, \hat{B}_T)$, or $\mathcal{JW}_{2,\beta}(\hat{Q}_{S_\ell}, \hat{B}_{S_\ell})$. These steps have computational, and storage complexity of $\mathcal{O}(n^3 \log n)$ and $\mathcal{O}(n^2)$, for an overall computational, and storage complexity of $\mathcal{O}(Nn^3 \log n)$ and $\mathcal{O}(Nn^2)$. The final resulting strategy, shown in Algorithm 9, has a computational, and storage complexity of $\mathcal{O}(NKn^3 \log n)$ and $\mathcal{O}(KNn^2)$. For large values of N, C, or n, both the computational and storage complexities of DaDiL become prohibitive. Therefore, we need a scalable algorithm for computing the loss in equation 6.6.

Algorithm 10: Minibatch DaDiL

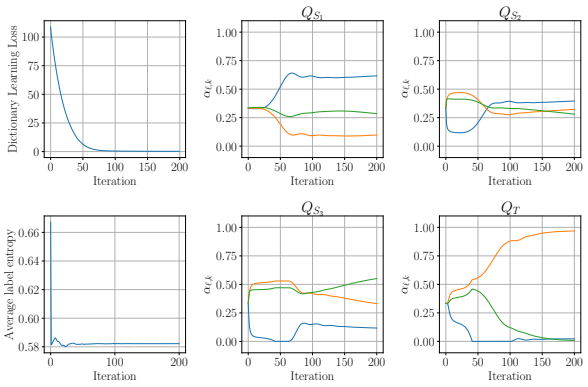
function dadil($\mathcal{Q}_S, \hat{Q}_T, N_{it}, n_b, C, \eta$)
  // Initialization
  $\mathbf{x}_j^{(P_c)} \sim \mathcal{N}(0, \mathbf{I}_d)$
  $\mathbf{u}_j^{(P_c)} \sim \mathcal{N}(0, \mathbf{I}_{n_c})$
  $\lambda_\ell \leftarrow \mathbb{1}_C / C$
  for $it = 1, \cdots, N_{it}$ do
    $L \leftarrow 0$
    // Sampling (Atoms)
    $\mathbf{X}^{(P_c)} \leftarrow \{\mathbf{x}_i^{(P_c)}\}_{i=1}^{n_b}$
    $\mathbf{u}^{(P_c)} \leftarrow \{\mathbf{u}_i^{(P_c)}\}_{i=1}^{n_b}$
    // Change of variables
    $\mathbf{Y}^{(P_c)} \leftarrow \text{softmax}(\mathbf{U}^{(P_c)})$
    // Dictionary learning (sources)
    for $\ell = 1, \cdots, N_S$ do
      // Sampling (Datasets)
      $\mathbf{X}^{(Q_{S_\ell})} \leftarrow \{\mathbf{x}_i^{(Q_{S_\ell})}\}_{i=1}^{n_b}$
      $\mathbf{Y}^{(Q_{S_\ell})} \leftarrow \{\mathbf{y}_i^{(Q_{S_\ell})}\}_{i=1}^{n_b}$
      $(\mathbf{X}^{(B_\ell)}, \mathbf{Y}^{(B_\ell)}) \leftarrow \mathcal{B}(\lambda_\ell, \mathcal{P})$
      $L \leftarrow L + \mathcal{JW}_{2,\beta}(\hat{Q}_{S_\ell}, \hat{B}_\ell)^2$
    // Dictionary learning (target)
    $\mathbf{X}^{(Q_T)} \leftarrow \{\mathbf{x}_i^{(Q_T)}\}_{i=1}^{n_b}$
    $(\mathbf{X}^{(B_T)}, \mathbf{Y}^{(B_T)}) \leftarrow \mathcal{B}(\lambda_T, \mathcal{P})$
    $L \leftarrow L + \mathcal{W}_2(\hat{Q}_T, \hat{B}_T)^2$
    $\mathbf{x}_j^{(P_c)} \leftarrow \mathbf{x}_j^{(P_c)} - \eta \partial L / \partial \mathbf{x}_j^{(P_c)}$
    $\mathbf{u}_j^{(P_c)} \leftarrow \mathbf{u}_j^{(P_c)} - \eta \partial L / \partial \mathbf{u}_j^{(P_c)}$
    $\lambda_\ell \leftarrow \pi_{\Delta_C}(\lambda_\ell - \eta \partial L / \partial \lambda_\ell)$
  return $\Lambda^\star, \mathcal{P}^\star$

Mini-batch strategy. Given the previous limitation of the full-batch approach, we can resort to mini-batch OT (see chapter 2, section 2.2.1). Our strategy consists on dividing the atom samples into $M-$minibatches of size n_b. Naturally, $M = {}^{n}/_{n_b}$, where for convenience we choose n to be a multiple of the batch size. The final resulting strategy is shown in Algorithm 10. In this version of DaDiL, OT is computed between mini-batches – including the Wasserstein barycenter, which has n_b samples in its support.

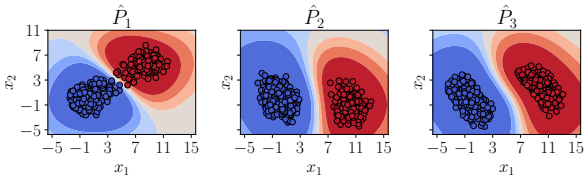
As we covered in section 2.2.1 of chapter 2, mini-batch OT poses new challenges, as at the level of a mini-batch, OT may match samples that would not be matched at a global level. This issue is even more sensitive, when not all classes are present in the sampled mini-batch. In this case, one cannot hope to have a *class-sparse* (see chapter 5). Here, we resort to a strategy introduced in [57], where the authors have sampled *balanced mini-batches*. This means that $spc = {}^{n_b}/_{n_c}$ samples per class are obtained from each class, in a stratified fashion. A second possible solution is to use unbalanced OT, so that, as discussed in [61], OT can automatically select to not transport some samples, especially outliers. This, however, introduces additional hyper-parameters in our algorithm, such as entropic regularization ϵ, and the penalty coefficient τ.



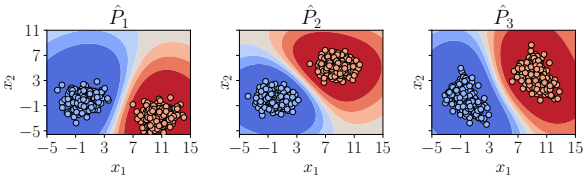
(a) Mini-batch strategy.



(b) Full-batch strategy.



(c) Mini-batch strategy.



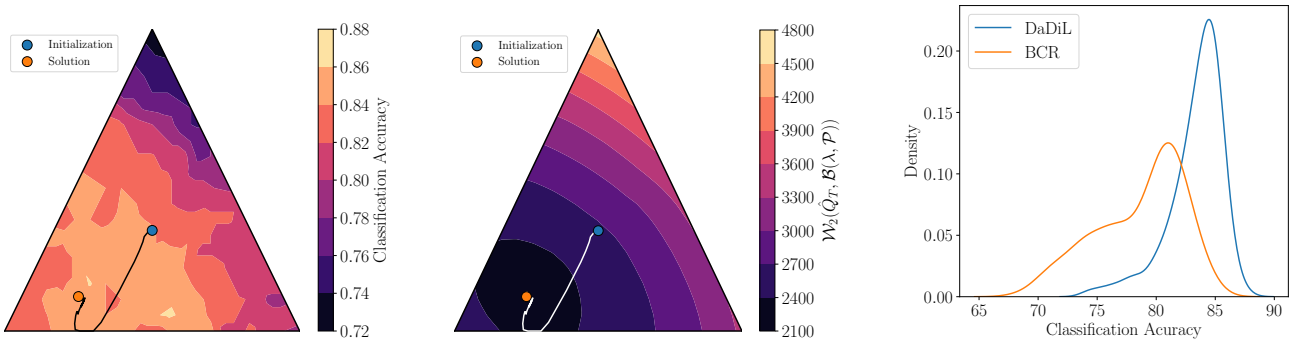
(d) Full-batch strategy.

Figure 6.8 – Comparison between DaDiL optimization strategies.

Example 19. *(Gaussian Clusters) In this example, we illustrate DaDiL using the toy dataset introduced in example 13. In the case of this section, we want to learn a set of atoms and barycentric coordinates that express each measure as a barycenter of atoms. Furthermore, we are particularly interested in comparing the mini-batch with the full-batch strategy. We present a comparison between these strategies in Figure 6.8.*

Beyond performance, Figure 6.8 illustates a few advantages of the mini-batch strategy. First, it converges faster, as can be seen in the dictionary learning loss curve in (a) and (b). Furthermore, the mini-batch strategy converges towards labels that have lower entropy (i.e., they are more "certain", with respect the class), as is shown in the average label entropy in (a) and (b), and the color intensities shown in (c) and (d).

Example 20. *(Caltech-Office benchmark) In this example we illustrate DaDiL on the Caltech-Office benchmark. We especially draw a comparison with our results on BCR, obtained in example 17. We show a summary of our results in Figure 6.9. We compare DaDiL with hyperparameters* $n_b = 200$, $n = 1000$, $\eta = 10^{-1}$ *and* $N_{it} = 50$. *For an ease of comparison with respect BCR, we fix* $C = 3$, *which allows us to compare the simplices* Δ_3 *with vertices* $\mathcal{P}$ *and* $\mathcal{Q}_S$, *for DaDiL and BCR respectively.*



(a) (b) (c)

Figure 6.9 – Summary of DaDiL on the Caltech-Office benchmark. In (a) and (b), we show the classification accuracy and loss of DaDiL over the simplex Δ_3. We particularly plot the evolution of weights in the simplex in these figures. In (c), we compare DaDiL and BCR, which shows that, on average, DaDiL is better over the interpolation space than BCR, indicating the advantage of performing dictionary learning.

One should compare Figures 6.9 with 6.6, in particular the classification accuracy and loss over the simplex Δ_3. *For DaDiL, we are showing the Wasserstein hull of atoms, i.e.,* $\mathcal{M}(\mathcal{P})$, *whereas for BCR we show* $\mathcal{M}(\mathcal{Q}_S)$. *In Figure 6.9 (c), we explore the distribution of classification accuracy over these interpolation spaces. Our main finding is that* $\mathcal{M}(\mathcal{P})$ *accuracy is higher over a larger region of* Δ_3, *as evidenced by Figure 6.9 (c), in comparison with* $\mathcal{M}(\mathcal{Q}_S)$. *This illustrates* why *perform dictionary learning, i.e., why learn* $(\Lambda, \mathcal{P})$.

6.2.3 Strategies for Domain Adaptation

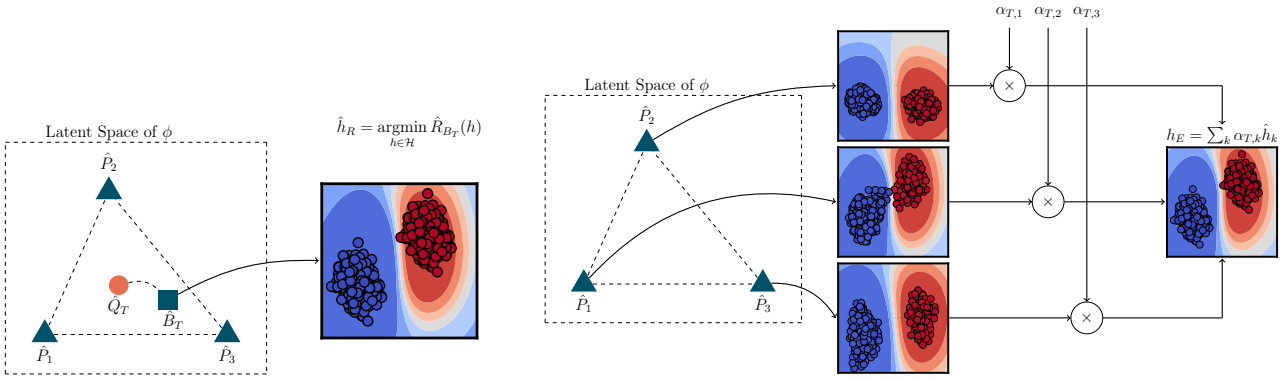
We propose 2 ways of using our dictionary for MSDA. Our first strategy, called DaDiL-R, consists on computing $\hat{B}_T = \mathcal{B}(\lambda_T; \mathcal{P})$, i.e., the distribution in $\mathcal{M}(\mathcal{P})$ closest to $\hat{Q}_T$. Since each $\hat{P}_c$ has a labeled support, algorithm 6 yields matrices $\mathbf{X}^{(B_T)}$ and $\mathbf{Y}^{(B_T)}$ corresponding to the support of $\hat{B}_T$. Then,

$$\hat{h}_R = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \hat{\mathcal{R}}_{B_T}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(h(\mathbf{x}_i^{(B_T)}), y_i^{(B_T)}) \tag{6.7}$$

We theoretically justify it using Theorem 2 of [100], which was previously presented in chapter 4 (Theorem 6). We apply this result for the residual shift $\mathcal{W}_2(\hat{Q}_T, \hat{B}_T)$,

$$\mathcal{R}_{Q_T}(h) \leq \mathcal{R}_{B_T}(h) + \mathcal{W}_2(\hat{Q}_T, \hat{B}_T) + \mathcal{C}_{OT}(n, \delta) + \mathcal{C}_{DA}(\hat{B}_T, \hat{Q}_T). \tag{6.8}$$

Here, we employ the $\mathcal{W}_2$ rather than $\mathcal{W}_1$. This is valid since $\mathcal{W}_1$ is the weakest of $\alpha-$Wasserstein distances. As discussed in [100], 3 factors play a role in the success of DA, namely, $\mathcal{W}_2(\hat{P}, \hat{Q})$, $\mathcal{R}_{B_T}(h)$, and $\mathcal{C}_{DA}$. The first term is the reconstruction error, and is directly minimized in algorithm 10. The second term is the risk of h in B_T, which is minimized when learning the classifier $\hat{h}_R = \text{argmin } \hat{\mathcal{R}}_{B_T}(h)$. This term depends on the separability of classes in $\hat{B}_T$, which is enforced by considering labels in the ground-cost (eqn. 5.4). The last term is the DA complexity $\mathcal{C}_{DA}$, i.e., the minimum error of a classifier learned with data from Q_T and B_T. This term is difficult to bound, as no labels in $\hat{Q}_T$ are available, but, under the hypothesis $Q_T(Y|X) = B_T(Y|T(X))$, this term is low. This was similarly assumed by [23, 100]. DaDiL-R is illustrated in Figure 6.10a.



(a) Reconstruction (b) Ensembling

Figure 6.10 – Conceptual illustration of the 2 methods, based on DaDiL, for MSDA.

Our second strategy, called DaDiL-E, is based on ensembling. Since each of our atoms is labeled, i.e., each $\mathbf{x}_i^{(P_c)}$ has an associated $\mathbf{y}_i^{(P_c)}$, we may learn a set of C classifiers, $\hat{h}_c = \text{argmin}_{h \in \mathcal{H}} \hat{\mathcal{R}}_{P_c}(h)$, one for each atom. Naturally, one may use $\lambda \in \Delta_C$ for weighting predictions of atom classifiers. We weight the $\hat{h}_c$'s using λ_T, which is theoretically justified in theorem 10,

$$\hat{h}_E(\mathbf{x}_j^{(Q_T)}) = \sum_{c=1}^{C} \lambda_{T,c} \hat{h}_c(\mathbf{x}_j^{(Q_T)}), \tag{6.9}$$

Theorem 10. *Let* $\{\mathbf{X}^{(P_c)}\}_{c=1}^C$, $\mathbf{X}^{(P_c)} \in \mathbb{R}^{n_c \times d}$ *and* $\mathbf{X}^{(Q_T)} \in \mathbb{R}^{n_T \times d}$ *be i.i.d. samples from* P_c *and* Q_T. *Let* $\hat{h}_c$ *be the minimizer of* $\mathcal{R}_{P_c}$ *and* $\mathcal{R}_\lambda(h) = \sum_{c=1}^C \lambda_c \mathcal{R}_{P_c}(h)$. *Assume* $\mathcal{L}$ *satisfies the triangle inequality. Under the same conditions of theorem 6, and for* $\delta \in (0, 1)$, *with probability at least* $1 - \delta$,

the following holds,

$$\mathcal{R}_{Q_T}(\hat{h}_\lambda) \leq \mathcal{R}_\lambda(\hat{h}_\lambda) + \mathcal{W}_2(\mathcal{B}(\lambda,\mathcal{P}),\hat{Q}_T) + \underbrace{\sum_{c=1}^{C}\lambda_c\mathcal{W}_2(\hat{P}_c,\mathcal{B}(\lambda,\mathcal{P}))}_{\mathcal{V}(\mathcal{P})} + \sum_{c=1}^{C}\lambda_c\mathcal{C}_{OT}(n,\delta) + \sum_{c=1}^{C}\lambda_c\mathcal{C}_{DA}(P_c,Q_T),$$

where $\mathcal{V}(\mathcal{P})$ denotes the Wasserstein variance *of $\mathcal{P}$. Note that λ_T minimizes the terms in the r.h.s., as, by design, it minimizes the term $\lambda \mapsto W_2(\mathcal{B}(\lambda,\mathcal{P}),\hat{Q}_T)$. DaDiL-E is illustrated in figure 6.10b.*

Proof. Before proving our result, we define the following risks,

$$\mathcal{R}_{Q_T}(h) = \mathop{\mathbb{E}}_{\mathbf{x}\sim Q_T}[\mathcal{L}(h(\mathbf{x}),h_{Q_T,0}(\mathbf{x}))],\ \mathcal{R}_{P_c}(h) = \mathop{\mathbb{E}}_{\mathbf{x}\sim P_c}[\mathcal{L}(h(\mathbf{x}),h_{P_c,0}(\mathbf{x}))],\text{ and }\mathcal{R}_\lambda(h) = \sum_{k=1}^{K}\lambda_c\mathcal{R}_{P_c}(h),$$

which are the risk under the target, under each of the atom distributions, and the combined risk weighted by λ, respectively. $h_{Q_T,0}$ and $h_{P_c,0}$ are the ground-truth labeling functions of the target distribution, and atom k. Likewise, we define the following classifiers,

$$\begin{aligned} h^\star_{T,k} &= \text{argmin}_{h\in\mathcal{H}}\mathcal{R}_{Q_T}(h) + \mathcal{R}_{P_c}(h),\\ \hat{h}_c &= \text{argmin}_{h\in\mathcal{H}}\mathcal{R}_{P_c}(h),\\ \hat{h}_\lambda(\mathbf{x}) &= \sum_c \lambda_c\hat{h}_c(\mathbf{x}). \end{aligned}$$

Our proof relies on the triangle inequality for the risk. With our previous definitions,

$$\mathcal{R}_{Q_T}(\hat{h}_\lambda) \leq \mathcal{R}_{Q_T}(h^\star_{T,k}) + \mathcal{R}_{Q_T}(h^\star_{T,k},\hat{h}_\lambda).$$

Now, we add and subtract $\mathcal{R}_{P_c}(h^\star_{T,k},\hat{h}_\lambda)$,

$$\begin{aligned} \mathcal{R}_{Q_T}(\hat{h}_\lambda) &\leq (\mathcal{R}_{Q_T}(h^\star_{T,k},\hat{h}_\lambda) - \mathcal{R}_{P_c}(h^\star_{T,k},\hat{h}_\lambda)) + (\mathcal{R}_{Q_T}(h^\star_{T,k}) + \underbrace{\mathcal{R}_{P_c}(h^\star_{T,k},\hat{h}_\lambda)}_{\leq\mathcal{R}_{P_c}(h^\star_{T,k})+\mathcal{R}_{P_c}(\hat{h}_\lambda)}),\\ &\leq \underbrace{(\mathcal{R}_{Q_T}(h^\star_{T,k},\hat{h}_\lambda) - \mathcal{R}_{P_c}(h^\star_{T,k},\hat{h}_\lambda))}_{\leq\mathcal{W}_2(P_c,Q_T)\text{ by Lemma 1.}} + \underbrace{(\mathcal{R}_{Q_T}(h^\star_{T,k}) + \mathcal{R}_{P_c}(h^\star_{T,k}))}_{=\lambda_c\text{ by Def.}} + \mathcal{R}_{P_c}(\hat{h}_\lambda) \end{aligned}$$

where, in the first line, we used the triangle inequality again. This result leads to,

$$\mathcal{R}_{Q_T}(\hat{h}_\lambda) \leq \mathcal{R}_{P_c}(\hat{h}_\lambda) + \mathcal{W}_2(P_c,Q_T) + \mathcal{C}_{DA}(P_c,Q_T).$$

Now, summing over k, weighted by λ, one has $\mathcal{R}_{Q_T}(\hat{h}_\lambda) = \sum_c \lambda_c\mathcal{R}_{Q_T}(\hat{h}_\lambda)$. We can bound this latter term as follows,

$$\begin{aligned} \sum_c \lambda_c\mathcal{R}_{Q_T}(\hat{h}_\lambda) &\leq \sum_c \lambda_c\mathcal{R}_{P_c}(\hat{h}_\lambda) + \sum_c \lambda_c(\mathcal{W}_2(P_c,Q_T) + \mathcal{C}_{DA}(P_c,Q_T)),\\ &= \mathcal{R}_\lambda(\hat{h}_\lambda) + \sum_c \lambda_c(\mathcal{W}_2(P_c,Q_T) + \mathcal{C}_{DA}(P_c,Q_T)),\\ &\leq \mathcal{R}_\lambda(\hat{h}_\lambda) + \sum_c \lambda_c(\mathcal{W}_2(\hat{P}_c,\hat{Q}_T) + \mathcal{C}_{DA}(P_c,Q_T) + \mathcal{C}_{OT}(n,\delta)), \end{aligned} \tag{6.10}$$

From this last inequality, we use the triangle inequality between $\hat{P}_c$, $\mathcal{B}(\lambda, \mathcal{P})$ and $\hat{Q}_T$,

$$\mathcal{W}_2(\hat{P}_c, \hat{Q}_T) \leq \mathcal{W}_2(\hat{P}_c, \mathcal{B}(\lambda, \mathcal{P})) + \mathcal{W}_2(\mathcal{B}(\lambda, \mathcal{P}), \hat{Q}_T).$$

Summing over k, and noting that,

$$\sum_c \lambda_c \mathcal{W}_2(\mathcal{B}(\lambda, \mathcal{P}), \hat{Q}_T) = \mathcal{W}_2(\mathcal{B}(\lambda, \mathcal{P}), \hat{Q}_T)$$

one has,

$$\sum_c \lambda_c \mathcal{W}_2(\hat{P}_c, \hat{Q}_T) \leq \mathcal{W}_2(\mathcal{B}(\lambda, \mathcal{P}), \hat{Q}_T) + \mathcal{V}(\mathcal{P}).$$

Plugging this result back into equation 6.10, one has the desired result. □

6.3 Conclusion

In this chapter, we introduced tools for the manipulation of multiple, heterogeneous datasets in a Wasserstein space. Our tools are especially designed to handle *point clouds*, i.e., empirical measures with a free-support. In this sense, we present new, analogous methods to prior art [28, 27], who focused on histograms, that is, empirical measures with a fixed-support. As a consequence, we started this chapter by reviewing the BCR and WDL problems introduced by these papers.

In addition, we introduced our methods, called BCR and DaDiL. In the first case, we find the barycentric coordinate vector $\lambda \in \Delta_{N_S}$ that minimizes the reconstruction error $\mathcal{W}_2(\hat{Q}_T, \mathcal{B}(\lambda, \mathcal{Q}_S))$. We provided two strategies for this problem, a mini-batch and a full-batch version, in algorithms 8 and 9, respectively. We did the same for our DaDiL strategy, which learns to express each measure in MSDA as a barycenter $\mathcal{B}(\lambda, \mathcal{P})$, parametrized by λ. As before, we have a mini-batch and full-batch versions, in algorithms 10 and 9, respectively.

Based on our DaDiL algorithm, we proposed 2 solutions for MSDA. The first, DaDiL-R, reconstructs labeled data that is close in measure to the target domain. With this labeled data, we are able to learn a classifier on the target domain. The second, DaDiL-E, relies on our learned atoms. These are labeled measures, hence we can learn a classifier h_c with data from each P_c. We can then weight the predictions of these classifiers using λ_T, for predicting on the target domain. These 2 methods are theoretically grounded, as we discuss in section 6.2.3.

Overall, we propose a novel framework for learning a Wasserstein hull, $\mathcal{M}(\mathcal{P})$, such that each measure in MSDA belongs to this interpolation space. As we show in further chapters, this framework can be extended to parametric models using GMMs [16, 15] (Chapter 7). Furthermore, our optimization algorithms can be extended to a federated scenario [137] (Chapter 9), in which different clients hold data from each domain, and do not want to centralize their data for adaptation. Finally, DaDiL and Wasserstein barycenters in general can be used to reduce or compress the number of samples in datasets [33].

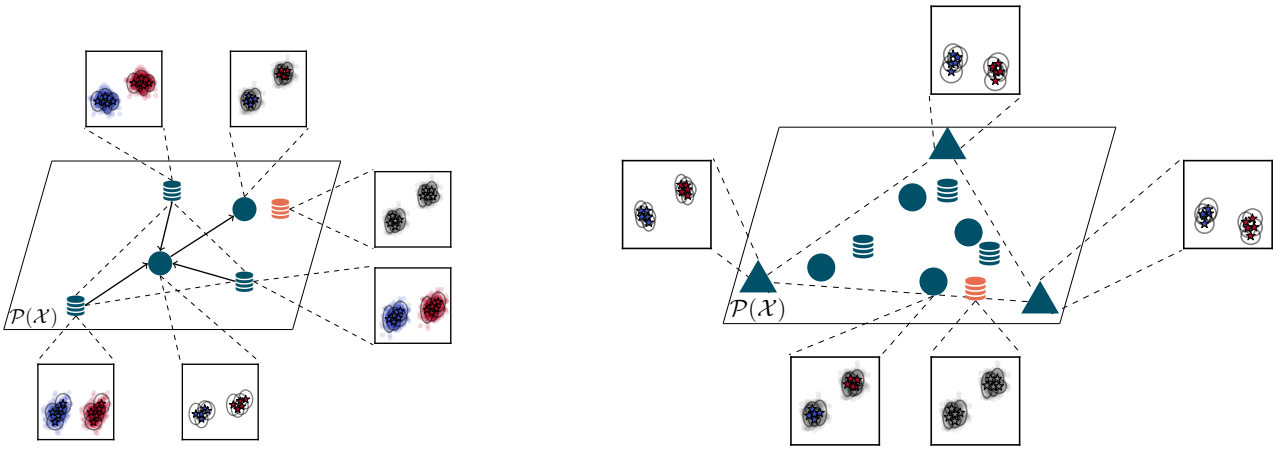
Chapter 7

Domain Adaptation with Gaussian Mixture Models

Contents

In the previous chapters, we proposed new tools for domain adaptation based on empirical OT. While these tools are effective, they suffer from some drawbacks. First, there is the curse of dimensionality [138, 139, 45], i.e., the speed of convergence of OT, with respect the number of samples, becomes slower as the dimension of data grows. Second, the empirical representation of measures rely on the entirety of datasets' samples, which is expensive with respect to computation, and storage complexity. In this chapter, we offer an alternative route based on the GMM-OT framework of [17], previously presented in Chapter 2. Instead of using mixtures of Diracs, we use mixtures of Gaussians for representing probability measures, which gives us parametric versions for the algorithms in [23], [13] and [14]. We show a conceptual illustration in Figure 7.1.

This chapter is divided as follows. Section 7.1 introduces a simple strategy for labeling the components of GMMs. Section 7.2 discusses our contribution [16] of using GMM-OT for DA. The remainder is based on [15]. Section 7.3 introduces a novel technique for calculating barycenters of GMMs. Section 7.4 proposes novel MSDA algorithms based on GMM-OT. Finally, section 7.5 concludes this chapter.



(a) GMM-WBT (b) GMM-DaDiL

Figure 7.1 – Conceptual illustration of GMM-WBT and GMM-DaDiL strategies. ⛁ represents datasets, circles represent barycenters and triangles represent learned atoms. Blue and orange elements represent labeled and unlabeled measures, respectively. GMM-WBT calculates a barycenter of source domain GMMs, then transport it to the target domain. GMM-DaDiL learns to express each GMM as a barycenter of learned atoms, which are themselves GMMs.

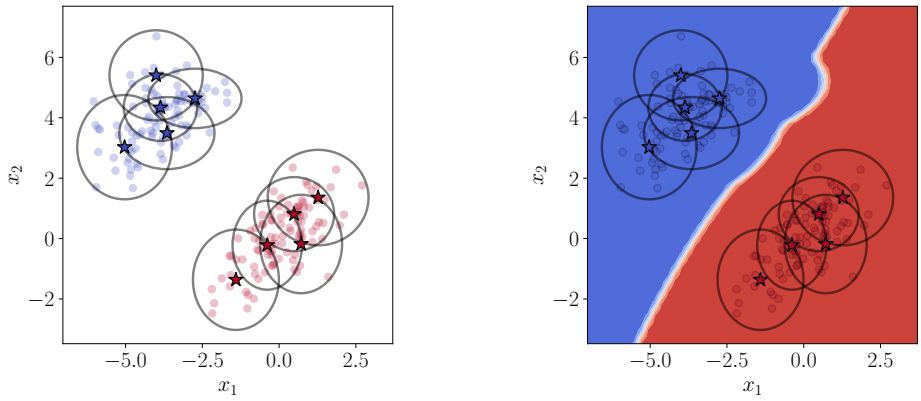
7.1 Supervised Gaussian Mixture Models

In this chapter, we consider supervised learning problems. As such, it is necessary to equip the components of GMMs with labels that represent the classes in the datasets. We propose doing so through a simple heuristic, especially, we model $P(\mathbf{x}|y)$ through a GMM. Especially, we use EM (Algorithm 2) on samples $\{\mathbf{x}_i^{(P)} : y_i^{(P)} = y\}$, for $y = 1, \cdots, n_c$. We then concatenate the n_c obtained GMMs, and assign, for the $k-$th GMM of the $y-$th class. Through this heuristic, we construct a vector for each component $\nu_{k,y'}^{(P)} = \delta(y' - y)$ with n_c entries, such that it has 1 on the $y-$th entry if the $k-$th component was fitted with data of the $y-$th class. We can assure that the resulting weights sum to 1 by dividing their value by $\sum_{y=1}^{n_c}\sum_{k=1}^{K} p_{k,y}$, where $p_{k,y}$ corresponds to the weight of the $k-$th component of the $y-$th GMM.

Given a GMM $\{p_k, \mu_k^{(P)}, \Sigma_k^{(P)}, \nu_k^{(P)}\}_{k=1}^K$, we define a classifier through 2 strategies. First, we can sample $\{(\mathbf{x}_i^{(P)}, y_i^{(P)})\}_{i=1}^n$, where $y_i^{(P)}$ inherits the label from the component it was sampled from. As a result, a classifier can be learned with these samples (e.g., using a neural net). A second strategies consists of using Maximum a Posteriori (MAP) estimation, that is,

$$\hat{h}_{MAP}(\mathbf{x}) = \underset{y=1,\cdots,n_c}{\operatorname{argmax}} P(y|\mathbf{x}) = \underset{y=1,\cdots,n_c}{\operatorname{argmin}} \sum_{k=1}^{K} P_\theta(k|\mathbf{x})P(y|k) = \underset{y=1,\cdots,n_c}{\operatorname{argmin}} \sum_{k=1}^{K} \frac{p_k P_k(\mathbf{x})}{\sum_{k'} p_{k'} P_{k'}(\mathbf{x})} \nu_{k,y}^{(P)}. \quad (7.1)$$

We show an example of the result of the conditional fit of GMMs, as well as the associated classifier obtained through MAP estimation in Figure 7.2.



(a) Labeled GMM (b) MAP classifier.

Figure 7.2 – **Class-conditional fit of GMMs**. In (a), we show an example in which a GMM is fit to each each class, then concatenated into a single GMM. In (b), we show the decision boundary of a classifier defined through MAP estimation.

7.2 Gaussian Mixture Domain Adaptation

In this section, we introduce a DA technique analogous to the OTDA proposed by [23] (see chapter 4). Let Q_S and Q_T be the probability measures for the source and target domain. We assume these measures are approximated through GMMs $Q_{S,\theta}$ and $Q_{T,\theta'}$. With a slight abuse of notation, we drop the θ and θ' from these GMMs. Since we have labeled data from Q_S, we can use our fitting strategy described in the previous section. In [16], we introduced a series of techniques for: (i) determining, based on Q_S, the labels for the target domain, i.e., $\nu_k^{(Q_T)}$, (ii) finding an OT map between Q_S and Q_T. We divided these methods into *label propagation* and *mapping estimation*.

Label propagation. Inspired by [24], we introduce a label propagation strategy for determining $\nu_k^{(Q_T)}$. We do so through the GMM-OT framework (see chapter 2 and equation 2.31),

$$\omega^\star = \underset{\omega \in \Gamma(\mathbf{p},\mathbf{q})}{\operatorname{argmin}} \sum_{k_1=1}^{K_P} \sum_{k_2=1}^{K_Q} \omega_{k_1,k_2} \mathcal{W}_2(Q_{S,k_1}, Q_{T,k_2})^2.$$

One can interpret $\omega^\star_{k_1,k_2}$ as the joint probability $P(k_1, k_2)$. As a result, we can apply the law of total probabilities for estimating $\nu^{(Q_T)}_{k_2,y} = Q_T(y|k_2)$,

$$\nu^{(Q_T)}_{k_2,y} = Q_T(y|k_2) = \sum_{k_1=1}^{K_P} P(k_1|k_2) Q_S(y|k_1) = \frac{1}{q_{T,k_2}} \sum_{k_1=1}^{K_P} \omega^\star_{k_1,k_2} \nu^{(Q_S)}_{k_1}. \tag{7.2}$$

In this last equation, we use an expression similar to the barycentric map defined in equation 5.8. We recall that, with a labeled GMM in the target domain, we can define a classifier either by sampling from Q_T, or by using MAP estimation.

Mapping Estimation. Given GMMs Q_S and Q_T for the source and target domain, our goal is to devise a transport map from Q_S to Q_T. As discussed in [17], this is challenging. As we covered in

remark 4, given $\omega^\star = \text{GMMOT}(Q_S, Q_T)$ the OT plan between samples of Q_S and Q_T is a GMM given by equation 2.32. Unfortunately, γ is not on the form $(Id, T)_\sharp Q_S$, so the definition of a transport map is not as straightforward. In [17], the authors propose 2 heuristics for devising a mapping.

The first strategy is similar to the notion of barycentric mapping in empirical OT, and consists of defining a map based on an expectation, $T_{mean}(\mathbf{x}_1) = \mathbb{E}_\gamma[X_2|X_1 = \mathbf{x}_1]$. This choice, however, may pose problems. As [17] exemplifies, given $Q_S = \mathcal{N}(0,1)$ and $Q_T = \frac{1}{2}\mathcal{N}(-a,1) + \frac{1}{2}\mathcal{N}(a,1)$, $T_{mean} = Id$. As a result, $T_{mean,\sharp}Q_S = Q_S$, and one may make the difference $\mathcal{MW}_2(T_{mean,\sharp}Q_S, Q_T)$ arbitrarily large by letting $a \to +\infty$.

A second strategy consists of mapping samples $\mathbf{x}_1 \sim Q_S$ randomly, that is,

$$T_{rand}(\mathbf{x}_1) = T_{k_1,k_2}(\mathbf{x}_1) \text{ with probability } p_{k_1,k_2}(\mathbf{x}_1) = \omega^\star_{k_1,k_2} \frac{\mathcal{N}(\mathbf{x}_1|\mu_{k_1}^{(Q_S)}, \Sigma_{k_1}^{(Q_S)})}{\sum_k q_{S,k}\mathcal{N}(\mathbf{x}_1|\mu_k^{(Q_S)}, \Sigma_k^{(Q_S)})}, \tag{7.3}$$

where T_{k_1,k_2} corresponds to the Monge map between components k_1 from Q_S and k_2 of Q_T (c.f., equation 2.4, in example 2 in Chapter 2). Using T_{rand} is less problematic from the point of view of mapping Q_S into Q_T, at the cost of regularity. Indeed, since T_{rand} is subject to the randomness of choosing T_{k_1,k_2}, two close-by samples may end up being mapped to different parts of the target measure. In [16] we introduced a deterministic alternative to T_{rand}, which we call T_{weight}.

Our intuition is twofold. First, we can increase the regularity of T_{rand}, by estimating the component k_1 that most likely originated $\mathbf{x}^{(P)}$, that is,

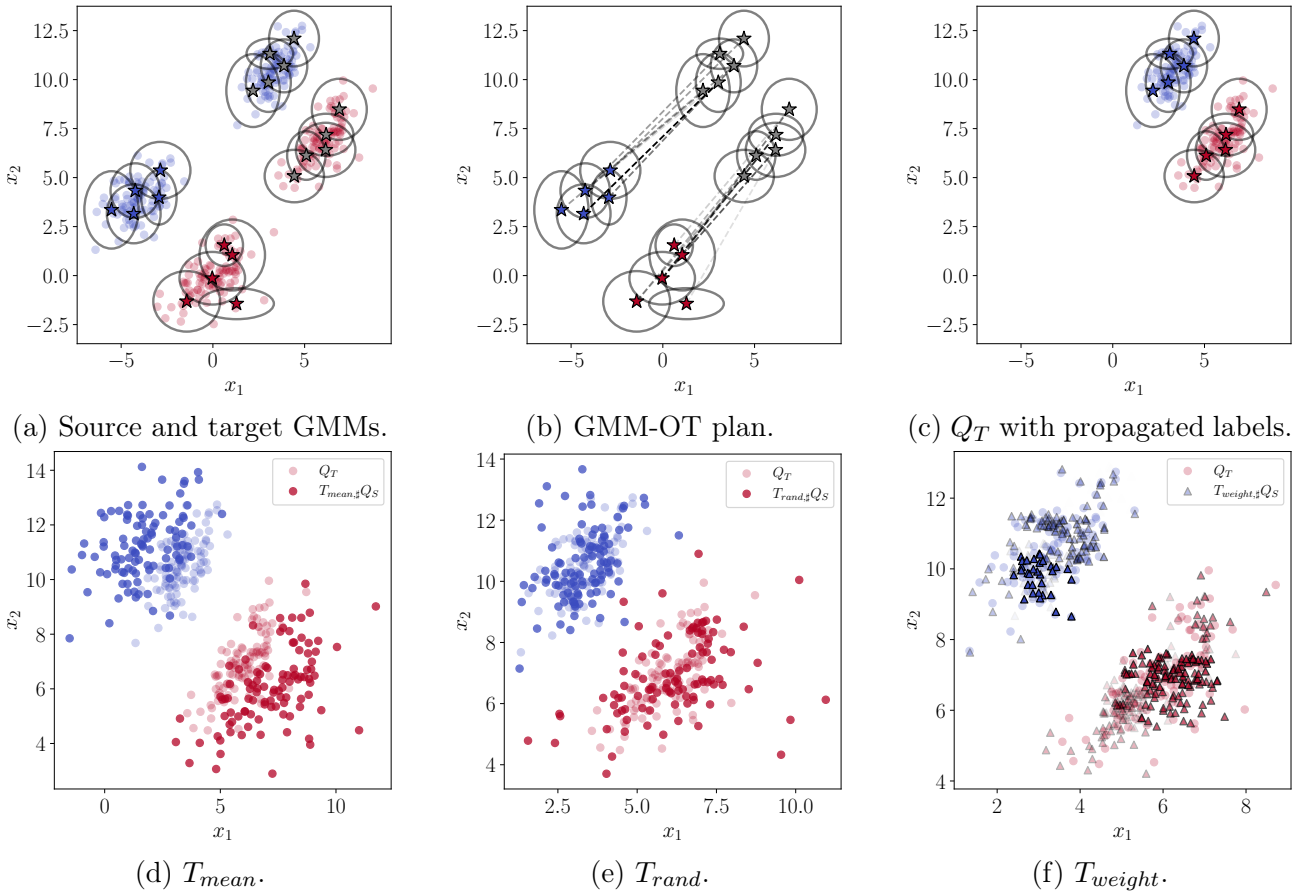
$$k_1 := \underset{k=1,\cdots,K_P}{\text{argmax}}\ P(K = k|X = \mathbf{x}^{(P)}) = \frac{p_k P_k(\mathbf{x}^{(P)})}{\sum_{k'=1}^{K_P} p_{k'} P_{k'}(\mathbf{x})}.$$

Second, we map $\mathbf{x}^{(P)}$ into the components of Q. Note that, even if $K_P = K_Q$, the optimal transport plan ω may split the mass of P_{k_1} into several $Q_{k_2'}$. This is due the different marginals $\mathbf{p}$ and $\mathbf{q}$. As a result, we produce $\{T_{k_1,k_2}(\mathbf{x}^{(P)})\}_{k_2:\omega_{k_1,k_2}\geq\tau}$, i.e., we map $\mathbf{x}^{(P)}$ to all components Q_{k_2} such that $\omega_{k_1,k_2} \geq \tau \geq 0$. In principle, one may choose $\tau = 0$ and filter only the components that are not matched with P_{k_1}. Third, we further weight the importance of generated samples, by using ω_{k_1,k_2} as sample weights. At the end, we generate a weighted dataset $\{(\omega_{k_1,k_2}, T_{k_1,k_2}(\mathbf{x}_i^{(P)}), y_i^{(P)})\}_{i=1}^m$, where m is the total amount of samples generated. We call our overall mapping T_{weight}.

These techniques were proposed in [16] for single source DA, and can be seen as extensions, using the GMM-OT framework of [17], of the works of [24] and [23]. We illustrate this methodology in Example 21. We can move towards MSDA by considering extensions of our own previous works, such as [12, 13] and [14]. We do so in the next sections.

Example 21. *(GMM-OTDA) Similarly to the previous examples in MSDA, we consider a binary classification problem. Since the methods explored in this section deal primarily with single source DA, we explore a problem with only two domains. We show an illustration of the GMM-OTDA strategies in Figure 7.3.*

As we show in Figure 7.3, based a labeled source GMM, we can propagate its labels towards an unlabeled target GMM using equation 7.2. This equation relies on the GMM-OT plan, $\omega^\star$, shown in Figure 7.3 (b), which defines a correspondence between the source and target GMM components. As



(a) Source and target GMMs. (b) GMM-OT plan. (c) Q_T with propagated labels.

(d) T_{mean}. (e) T_{rand}. (f) T_{weight}.

Figure 7.3 – Illustration of GMM-OTDA. In (a), 2 GMMs are learned, for the source and target domain. Since labels are not available in the target domain, the GMM is initially unlabeled (gray stars). Through the GMM-OT plan, in (b), we are able to propagate the labels of Q_S towards Q_T, as shown in (c). In $(d-f)$ we show different GMM-OT mapping strategies.

we previously discussed, the situation for estimating a map between Q_S and Q_T is more challenging. On one hand, as we show in (d) and (e), T_{mean} and T_{rand} do not produce samples that follow Q_T. On the other hand, our proposed strategy T_{weight} effectively maps the points of Q_S correctly.

In the following, we cover our technical contributions towards MSDA. These composed our publication [15], and rely on new tools for computing Wasserstein barycenters of axis-aligned GMMs.

7.3 Gaussian Mixture Barycenters

Before proceeding with our proposed methods for MSDA, we need to define efficient algorithms for calculating mixture-Wasserstein barycenters of GMMs. We previously covered a way of calculating such barycenters in chapter 3, section 3.3.2. However, as we analyzed in the aforementioned section, this approach is not scalable with respect the number of components in the GMMs in $\mathcal{P}$. Given this shortcoming, we proposed in [15] an algorithm *à la Cuturi* [26] for solving,

$$\mathcal{B}(\lambda, \mathcal{P}) = \underset{B \in \text{GMM}_d(K_B)}{\text{argmin}} \sum_{c=1}^{C} \lambda_c \mathcal{MW}_2(B, P_c)^2.$$

Remark 6. *Throughout this chapter, we assume axis-aligned GMMs, i.e., we assume diagonal covariance matrices $\Sigma_k^{(P)}$. As a result, we use standard deviation vectors $\sigma_k^{(P)} \in \mathbb{R}_+^d$. While this assumption is restrictive, we need it to enforce the numerical stability and the correct estimation of GMMs. Furthermore, our methods can be extended to the case of complete covariances, but the computational complexity, and stability of estimating a barycenter high-dimensional Gaussian distributions oftentimes becomes prohibitive.*

Next, we develop a theory for the efficient calculation of mixture-Wasserstein, and joint mixture-Wasserstein barycenters. This theory essentially extends the algorithms in previous works [26, 12, 13, 14] to the GMM scenario.

7.3.1 Mixture Wasserstein Barycenters

Our first step is exploring how we can leverage the discrete OT plan between GMMs for mapping the components of P into those of Q. Note that, in GMM-OT, γ is an OT plan between samples, which is a GMM. Associated with γ, we have a discrete OT plan, ω, between components of P and Q. In the next result, we show that we can map the components of P into those of Q through barycentric maps (see equation 2.19), but here *applied over components of* P.

Theorem 11. *Let P and Q be two GMMs with components $P_{k_1} = \mathcal{N}(\mu_{k_1}^{(P)}, (\sigma_{k_2}^{(P)})^2)$ (resp. Q_{k_2}) and $\omega^\star$ be the solution of equation 2.31. The first-order optimality conditions of $\mathcal{MW}_2^2$, with respect $\mu_{k_1}^{(P)}$ and $\sigma_{k_1}^{(P)}$ are given by,*

$$\mu_{k_1}^{(P)} = T_{\omega^\star}(\mu_{k_1}^{(P)}) = \sum_{k_2=1}^{K_Q} \frac{\omega_{k_1,k_2}^\star}{p_{k_1}} \mu_{k_2}^{(Q)}, \text{ and } \sigma_{k_1}^{(P)} = T_{\omega^\star}(\sigma_{k_1}^{(P)}) = \sum_{k_2=1}^{K_Q} \frac{\omega_{k_1,k_2}^\star}{p_{k_1}} \sigma_{k_2}^{(Q)}, \tag{7.4}$$

Proof. Under the GMM-OT framework, the 2−mixture Wasserstein distance is given by,

$$\mathcal{MW}_2(P, Q)^2 = \sum_{k_1=1}^{K_P} \sum_{k_2=1}^{K_Q} \omega_{k_1,k_2}^\star \mathcal{W}_2(P_{k_1}, Q_{k_2})^2,$$

where $\omega_{k_1,k_2}^\star$ is the mass transported from component i in P to component j in Q by the solution to the GMM-OT problem. Since P_{k_1} and Q_{k_2} are axis-aligned Gaussian measures,

$$\mathcal{W}_2(P_{k_1}, Q_{k_2})^2 = \|\mu_{k_1}^{(P)} - \mu_{k_2}^{(Q)}\|_2^2 + \|\sigma_{k_1}^{(P)} - \sigma_{k_2}^{(Q)}\|_2^2,$$

where $\mu_{k_1}^{(P)}$ and $\sigma_{k_1}^{(P)}$ are the free parameters we want to determine. By linearity,

$$\frac{\partial \mathcal{MW}_2^2}{\partial \mu_{k_1}^{(P)}} = 2\sum_{k_2=1}^{K_Q} \omega_{k_1,k_2}^{\star}(\mu_{k_1} - \mu_{k_2}^{(Q)}) = 2\left(p_{k_1}\mu_{k_1} - \sum_{k_2=1}^{K_Q} \omega_{k_1,k_2}^{\star}\mu_{k_2}^{(Q)}\right),$$

equating this last term to 0, one gets the desired equality. The derivation for $\sigma_{k_1}^{(P)}$ is analogous. □

Remark 7. *In equation 7.4, we express $\sigma_{k_1}^{(P)}$ as a combination of $\sigma_{k_2}^{(Q)}$, which are all positive vectors. Furthermore, since this is a convex combination (due to $\sum_{k_2} {}^{\omega_{k_1,k_2}^{\star}}/_{p_{k_1}} = 1$), $\sigma_{k_2}^{(Q)} \in \mathbb{R}_+^d$.*

Remark 8. *In theorem 11, we are interested in the derivatives of the function,*

$$\mathcal{MW}_2(P,Q)^2 = \min_{\omega \in \Gamma(\mathbf{p},\mathbf{q})} \sum_{k_1=1}^{K_P}\sum_{k_2=1}^{K_Q} \omega_{k_1,k_2}\mathcal{W}_2(P_{k_1},Q_{k_2})^2,$$

which is the pointwise minimum of convex functions (w.r.t. $\mu_{k_1}^{(P)}$ and $\sigma_{k_1}^{(P)}$). Unfortunately, this function is concave *[140, Section 3.2.3]. Nonetheless, from the perspective of optimization, we use the envelope theorem [141, 142] and differentiate $\mathcal{MW}_2(P,Q)^2$ at optimality, i.e.,*

$$\mathcal{MW}_2(P,Q)^2 = \sum_{k_1=1}^{K_P}\sum_{k_2=1}^{K_Q} \omega_{k_1,k_2}^{\star}\mathcal{W}_2(P_{k_1},Q_{k_2})^2,$$

where $\omega^{\star}$ is the GMM-OT plan between P and Q.

On a more general note, analyzing OT at optimality is convenient. First, quantities such as the Wasserstein distance $\mathcal{W}_2$ and the mixture-Wasserstein distance $\mathcal{MW}_2$ become convex functions of the parameters of P and Q, since these are convex combinations of convex functions. Second, differentiating at optimality avoid the intricate calculation of gradients through linear programming, or the Sinkhorn algorithm. As previously reported in [143], differentiating at optimality is preferable than differentiating through the inner optimization problem in OT. While deep and relevant, these questions are out of the scope of this thesis.

Now that we laid out our main tools, we can define an efficient algorithm for calculating mixture-Wasserstein barycenters. Our analysis relies on the map $B \mapsto \sum_c \lambda_c \mathcal{MW}_2(B,P_c)$ with respect $\mu_{k_1}^{(B)}$ and $\sigma_{k_1}^{(B)}$. We formalize these ideas in the next theorem.

Theorem 12. *Let $\lambda \in \Delta_C$ be a vector of barycentric coordinates, and $\mathcal{P} = \{P_c\}_{c=1}^C$ be a set of GMMs. Given $K_B \geq 1$, the first-order optimality conditions of,*

$$\begin{aligned}\mathcal{L}(\{\mu_{k_1}^{(B)}\}_{k_1=1}^{K_B}, \{\sigma_{k_1}^{(B)}\}_{k_1=1}^{K_B}) &= \sum_{c=1}^{C} \lambda_c \mathcal{MW}_2(B,P_c)^2, \\ &= \sum_{c=1}^{C} \lambda_c \sum_{k_1=1}^{K_B}\sum_{k_2=1}^{K_{P_c}} \omega_{c,k_1,k_2}^{\star}\left(\|\mu_{k_1}^{(B)} - \mu_{k_2}^{(P_c)}\|_2^2 + \|\sigma_{k_1}^{(B)} - \sigma_{k_2}^{(P_c)}\|_2^2\right), \end{aligned} \tag{7.5}$$

are given by,

$$\mu_{k_1}^{(B)} = \sum_{c=1}^{C} \lambda_c T_{\omega_c^\star}(\mu_{k_1}^{(B)}), \text{ and, } \sigma_{k_1}^{(B)} = \sum_{c=1}^{C} \lambda_c T_{\omega_c^\star}(\sigma_{k_1}^{(B)})$$

Proof. The proof roughly follows the same logic of theorem 11. We do the derivations for $\mu_{k_1}^{(B)}$, as those for $\sigma_{k_1}^{(B)}$ are analogous. Taking derivatives of equation 7.5,

$$\begin{aligned}\frac{\partial \mathcal{MW}_2(B, P_c)^2}{\partial \mu_{k_1}^{(B)}} &= 2\sum_{c=1}^{C} \lambda_c \Big(\sum_{k_2=1}^{K_{P_c}} \omega_{c,k_1,k_2}^\star (\mu_{k_1}^{(B)} - \mu_{k_2}^{(P_c)}) \Big), \\ &= \frac{2}{K_B}\mu_{k_1}^{(B)} - 2\sum_{c=1}^{C} \lambda_c \sum_{k_2=1}^{K_{P_c}} \omega_{c,k_1,k_2}^\star \mu_{k_2}^{(P_c)},\end{aligned}$$

equating this last equation to 0 gives the desired result. □

Algorithm 11: Mixture Wasserstein Barycenter

function `mw_barycenter`$(\{\{\mu_k^{(P_c)}\}_{k=1}^K\}_{c=1}^C,\ \{\{\sigma_k^{(P_c)}\}_{k=1}^K\}_{c=1}^C,\ \lambda,\ \tau,\ N_{it})$

```
// Initialization.
```

$\mu_k^{(B)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$

$\sigma_k^{(B)} \leftarrow \mathbb{1}_d$

while $|L_{it} - L_{it-1}| \geq \tau$ **and** $it \leq N_{it}$ **do**

```
// Computes GMM-OT transport plans
```

for $c = 1, \cdots, C$ **do**

$\omega_c^{(it)} = \text{GMMOT}(B, P_c)$

```
// Note:  W_2(B_{k_1}, P_{c,k_2})^2 = ||μ_{k_1}^{(B)} − μ_{k_2}^{(P_c)}||_2^2 + ||σ_{k_1}^{(B)} − σ_{k_2}^{(P_c)}||_2^2
```

$L_{it} = \sum_{c=1}^{C} \lambda_c \sum_{k_1=1}^{K_B} \sum_{k_2=1}^{K_P} \omega_{c,k_1,k_2}^{(it)} \mathcal{W}_2(B_{k_1}, P_{c,k_2})^2$

```
// Update barycenter parameters
```

$\mu_{k_1}^{(B)} = \sum_{c=1}^{C} \lambda_c T_{\omega_c^{(it)}}(\mu_{k_1}^{(B)})$

$\sigma_{k_1}^{(B)} = \sum_{c=1}^{C} \lambda_c T_{\omega_c^{(it)}}(\sigma_{k_1}^{(B)})$

return $\{(\mu_k^{(B)}, \sigma_k^{(B)})\}_{k=1}^K$

In comparison with Cuturi's algorithm [26] for the computation of free-support empirical Wasserstein barycenters, our algorithm for GMMs includes two barycentric maps, i.e., for the means and standard deviation vectors. Our strategy is shown in Algorithm 11.

In what follows, we need to explicitly take into account the labels of GMMs components. That way, the GMM obtained through the mixture-Wasserstein barycenter is labeled as well. In this sense, we take a similar step to what we did in Chapter 5, that is, we compute the Euclidean distance between soft-label vectors, which introduces a new metric over the space of GMMs.

7.3.2 Joint Mixture Wasserstein Barycenters

So far, we have introduced novel tools for the calculation of barycenters of GMMs. These tools, however, do not take into account the labels, $\nu_k^{(P)}$. Note that, analogously to the empirical case, labeled GMMs are measures over the space $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y} = \Delta_{n_c}$ is the space of soft-labels. Therefore, we introduce a novel Wasserstein-like distance for labeled GMMs, inspired by [17] and [14],

Definition 29. *(Joint mixture-Wasserstein distance) Let $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \Delta_{n_c}$ be the feature and label spaces. For GMMs P and Q with parameters $\{\mu_k^{(P)}, \nu_k^{(P)}, \Sigma_k^{(P)}\}_{k=1}^{K_P}$ (resp. Q), the $\beta-$Joint mixture-Wasserstein distance is,*

$$\mathcal{JMW}_\beta(P,Q)^2 = \min_{\omega \in \Gamma(\mathbf{p},\mathbf{q})} \sum_{k_1=1}^{K_P} \sum_{k_2=1}^{K_Q} \omega_{k_1,k_2}(\mathcal{W}_2(P_{k_1}, Q_{k_2})^2 + \beta \|\nu_{k_1}^{(P)} - \nu_{k_2}^{(Q)}\|_2^2). \tag{7.6}$$

Furthermore, we denote the associated OT problem as,

$$JGMMOT(P,Q) = \operatorname*{argmin}_{\omega \in \Gamma(\mathbf{p},\mathbf{q})} \sum_{k_1=1}^{K_P} \sum_{k_2=1}^{K_Q} \omega_{k_1,k_2}(\mathcal{W}_2(P_{k_1}, Q_{k_2})^2 + \beta \|\nu_{k_1}^{(P)} - \nu_{k_2}^{(Q)}\|_2^2). \tag{7.7}$$

Note that, in the previous definition, we use $2-$Wasserstein distances between Gaussians. We fix this choice, as this distance can be readily computed from the parameters of the components. As in Chapter 5 (see Theorem 9), the use of the Euclidean distance between soft-probability vectors allows us to conveniently express the first order conditions of $\mathcal{JMW}$ through label propagation,

Theorem 13. *Under the same conditions of theorem 11, let P_{k_1} and Q_{k_2} be equipped with labels $\nu_{k_1}^{(P)}$ and $\nu_{k_2}^{(Q)}$, both in Δ_{n_c}. Let $\omega^\star = JGMMOT(P,Q)$. The first order optimality conditions of $\mathcal{JMW}_\beta$ with respect μ_{k_1} and σ_{k_2} are given by equation 7.4. Furthermore, for ν_{k_1},*

$$\nu_{k_1} = T_{\omega^\star}(\nu_{k_1}^{(P)}) = \sum_{k_2=1}^{K_Q} \frac{\omega^\star_{k_1,k_2}}{p_i} \nu_{k_2}^{(Q)}. \tag{7.8}$$

An immediate consequence of this result is that we may update Algorithm 11 to take into account the labels of the GMMs in $\mathcal{P}$ with minor modifications. It suffices to substitute *GMMOT* for *JGMMOT* in line 6, and add an update of the barycenter GMM labels, after line 9. The resulting strategy is shown in Algorithm 12.

Next, we introduce methods for MSDA. We propose two contributions: (i) GMM-WBT, and (ii) GMM-DaDiL. These two methods leverage the tools introduced in this chapter for finding a labeled GMM on the target domain. We emphasize again that, with a labeled GMM at hand, it is possible to find a classifier on the target domain, either by sampling from the GMM, or using MAP estimation (c.f., equation 7.1).

7.4 Multi-Source Domain Adaptation Strategies

In this section, we detail two contributions for MSDA based on GMM-OT: GMM-WBT and GMM-DaDiL. In both cases, we suppose access to N labeled source measures $\mathcal{Q}_S = \{Q_{S_\ell}\}_{\ell=1}^N$ and an unlabeled

Algorithm 12: Mixture Wasserstein Barycenter

function `jmw_barycenter`$(\{\{\mu_k^{(P_c)}\}_{k=1}^K\}_{c=1}^C,\ \{\{\sigma_k^{(P_c)}\}_{k=1}^K\}_{c=1}^C,\ \{\{\nu_k^{(P_c)}\}_{k=1}^K\}_{c=1}^C,\ \lambda,\ \tau,\ N_{it})$

// Initialization.

$\mu_k^{(B)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $\sigma_k^{(B)} \leftarrow \mathbb{1}_d$, $\nu_k^{(B)} \leftarrow \mathbb{1}_{n_c}/n_c$

while $|L_{it} - L_{it-1}| \geq \tau$ **and** $it \leq N_{it}$ **do**

// Computes GMM-OT transport plans

for $c = 1, \cdots, C$ **do**

$\omega_c^{(it)} = \text{JGMMOT}(B, P_c)$

// Note: $\mathcal{W}_2(B_{k_1}, P_{c,k_2})^2 = \|\mu_{k_1}^{(B)} - \mu_{k_2}^{(P_c)}\|_2^2 + \|\sigma_{k_1}^{(B)} - \sigma_{k_2}^{(P_c)}\|_2^2$

$L_{it} = \sum_{c=1}^C \lambda_c \sum_{k_1=1}^{K_B} \sum_{k_2=1}^{K_P} \omega_{k_1,k_2}^{(c,it)} (\mathcal{W}_2(B_{k_1}, P_{c,k_2})^2 + \beta \|\nu_{k_1}^{(B)} - \nu_{k_2}^{(P_c)}\|_2^2)$

// Update barycenter parameters

$\mu_{k_1}^{(B)} = \sum_{c=1}^C \lambda_c T_{\omega_c^{(it)}}(\mu_{k_1}^{(B)})$

$\sigma_{k_1}^{(B)} = \sum_{c=1}^C \lambda_c T_{\omega_c^{(it)}}(\sigma_{k_1}^{(B)})$

$\nu_{k_1}^{(B)} = \sum_{c=1}^C \lambda_c T_{\omega_c^{(it)}}(\nu_{k_1}^{(B)})$

return $\{(\mu_k^{(B)}, \nu_k^{(B)}, \sigma_k^{(B)})\}_{k=1}^K$

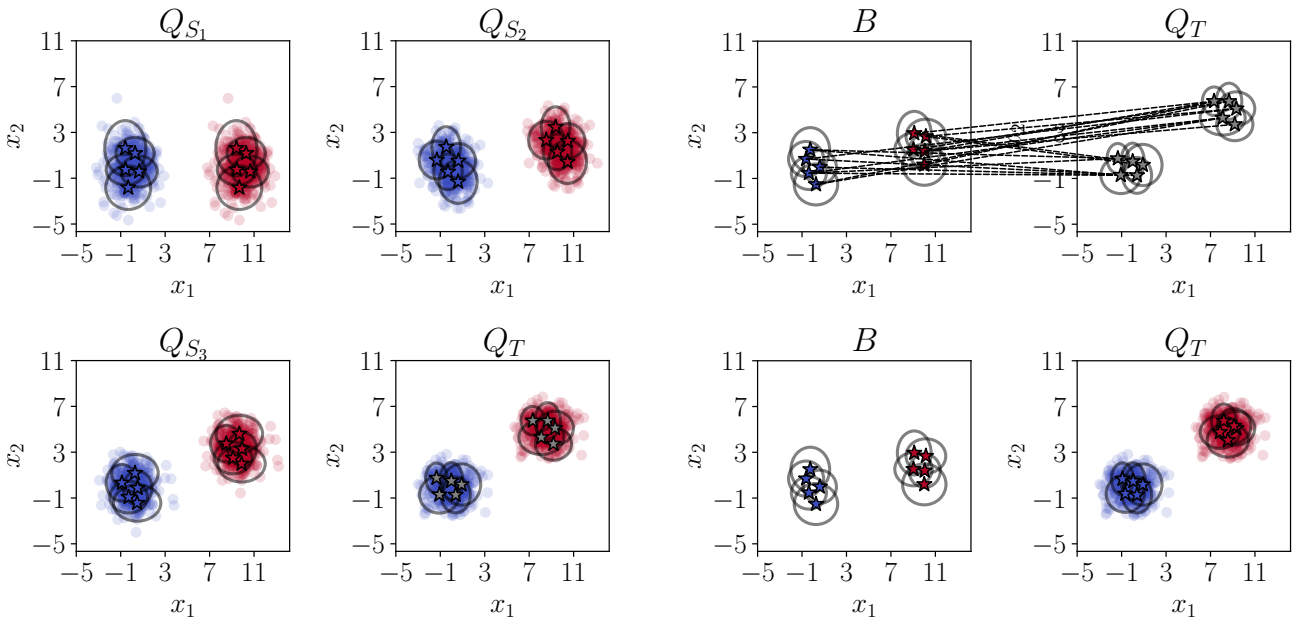
target measure Q_T. Our first method, GMM-WBT, adapts the WBT algorithm to the GMM framework. We start by calculating a Wasserstein barycenter of $B = \mathcal{B}(\mathbb{1}_N/N, \mathcal{Q}_S)$. After this step, WBT solves a single-source problem between B and Q_T. When each Q_{S_ℓ} is a GMM, the parameters of B are estimated through algorithm 12. Next, one solves for $\omega^{(T)} = \text{GMMOT}(B, Q_T)$, so that the parameters of B are transported towards Q_T using theorems 11 and 13,

$$\hat{\mu}_{k_1}^{(Q_T)} = K_B \sum_{k_2=1}^{K_T} \omega_{k_1,k_2}^{(T)} \mu_{k_2}^{(Q_T)}, \text{ and, } \hat{\sigma}_{k_1}^{(Q_T)} = K_B \sum_{k_2=1}^{K_T} \omega_{k_1,k_2}^{(T)} \sigma_{k_2}^{(Q_T)}. \tag{7.9}$$

With a labeled GMM, $\{\hat{\mu}_k^{(Q_T)}, \hat{\sigma}_k^{(Q_T)}, \nu_k^{(B)}\}_{k=1}^{K_B}$, on the target domain, we can learn a classifier on the target domain as explained in section 7.1.

Example 22. *(GMM-WBT on Gaussian Clusters) In this example, we continue the illustration of our proposed methods on the Gaussian clusters toy example (see e.g., example 13). The first step in our methodology is fitting a GMM on each domain, which is shown in Figure 7.4 (a). Note that, since the target domain has no labels, the learned GMM is not labeled as well.*

Based on the parameters of these GMMs, we want to find a labeled GMM on Q_T through GMM-WBT. We do so by first calculating the Wasserstein barycenter of $\mathcal{Q}_S$, shown in Figure 7.4 (b). We then calculate the GMM-OT plan $\omega^{(T)}$ between B and Q_T, which is shown as the lines connecting the components of B and Q_T on the top part of Figure 7.4 (b). Based on that, we may propagate the labels of the barycenter towards the target GMM, as is shown in the bottom right of Figure 7.4 (b).



(a) Domain GMMs. (b) GMM-WBT.

Figure 7.4 – In (a), we show the initial setting for GMM-WBT, i.e., a GMM learned on each domain. Since we do not have access to labeled data on the target domain, Q_T is not labeled (gray points). In (b), we show a summary of the GMM-WBT methodology. A labeled GMM is estimated (left part). We then calculate a GMM-OT plan (top) between B and Q_T. Based on this plan, we propagate the labels of B towards Q_T.

Our second algorithm consists of a parametric version for the DaDiL algorithm of [14]. The idea is to replace the atoms in $\mathcal{P} = \{\hat{P}_c\}_{c=1}^C$ by GMMs parametrized through $\Theta_P = \{\{(\mu_k^{(P_c)}, \sigma_k^{(P_c)}, \nu_k^{(P_c)})\}_{k=1}^K\}_{c=1}^C$. Learning a dictionary is thus equivalent to estimating these parameters, that is,

$$(\Lambda^\star, \Theta_P^\star) = \underset{\Lambda, \Theta_P}{\operatorname{argmin}}\, \mathcal{MW}_2(Q_T, \mathcal{B}(\lambda_T, \mathcal{P}))^2 + \sum_{\ell=1}^{N} \mathcal{SMW}_2(Q_\ell, \mathcal{B}(\lambda_\ell, \mathcal{P}))^2. \tag{7.10}$$

An important feature of this strategy is that the barycenters $\mathcal{B}$ inherit the labels from the atoms $\mathcal{P}$. As a result, we can estimate a labeled GMM for Q_T in one step through $\mathcal{B}(\lambda_T, \mathcal{P})$.

While equation 7.10 does not have a closed-form solution, we optimize it through gradient descent. An advantage of the GMM modeling is that this optimization problem involves far less variables than DaDiL, hence we do not resort to mini-batches. We detail our strategy in Algorithm 13. Note that we need to enforce 3 kinds of constraints: (i) $\sigma_k^{(P_c)} \in \mathbb{R}_+^d$, (ii) $\lambda_\ell \in \Delta_C$ and (iii) $\nu_k^{(P_c)} \in \Delta_{n_{cl}}$. For (i) and (ii), we use orthogonal projections into $\mathbb{R}_+^d$ and Δ_C respectively. We additionally set $\sigma_k^{(P_c)} \geq \sigma_{min}$ for numerical stability. For (iii), we perform a change of variables $\nu_k^{(P_c)} = \text{softmax}(\tilde{\nu}_i^{(P_c)})$.

Once the dictionary $(\Lambda, \mathcal{P})$ is learned, we are able to reconstruct the domains in MSDA via the barycenter $\mathcal{B}(\lambda, \mathcal{P})$. We are especially interested in the target reconstruction λ_T, i.e., $\mathcal{B}(\lambda_T, \mathcal{P})$. This barycenter is a labeled GMM. As a result, we can obtain labeled samples from this GMM, then use them to train a classifier that works on the target domain.

Algorithm 13: GMM-Dataset Dictionary Learning

function `gmm_dadil`($\{\mathbf{M}^{(Q_{S_\ell})}\}_{\ell=1}^N$, $\{\mathbf{S}^{(Q_{S_\ell})}\}_{\ell=1}^N$, $\{\mathbf{Y}^{(Q_{S_\ell})}\}_{\ell=1}^N$, $\{(\mu_k^{(Q_T)}, \sigma_k^{(Q_T)}, \nu_k^{(Q_T)})\}_{k=1}^K$, N_{it}, η)

// Initialization.

$\mu_k^{(P_c)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $\sigma_k^{(P_c)} \leftarrow 1$, $\tilde{\nu}_k^{(P_c)} \leftarrow \mathbb{1}_{n_c}/n_c$, $\lambda_\ell \leftarrow 1/K$

for $it = 1, \cdots, N_{it}$ **do**

  $L \leftarrow 0$

  $\nu_k^{(P_c)} \leftarrow \text{softmax}(\tilde{\nu}_k^{(P_c)})$

  // Evaluate supervised loss on sources

  for $\ell = 1, \cdots, N$ **do**

    $L \leftarrow L + \mathcal{SMW}_2(Q_{S_\ell}, \mathcal{B}(\lambda_\ell, \mathcal{P}))^2$

  // Evaluate unsupervised loss on targets

  $L \leftarrow L + \mathcal{MW}_2(Q_T, \mathcal{B}(\lambda_T, \mathcal{P}))^2$

  // Gradient step

  $\mu_k^{(P_c)} \leftarrow \mu_k^{(P_c)} - \eta \partial L / \partial \mu_k^{(P_c)}$

  $\tilde{\nu}_k^{(P_c)} \leftarrow \tilde{\nu}_k^{(P_c)} - \eta \partial L / \partial \tilde{\nu}_k^{(P_c)}$

  // Note: we project variables σ and λ.

  $\sigma_k^{(P_c)} \leftarrow \text{proj}_{\mathbb{R}_+^d}(\sigma_k^{(P_c)} - \eta \partial L / \partial \sigma_k^{(P_c)})$

  $\lambda_\ell \leftarrow \text{proj}_{\Delta_K}(\lambda_\ell - \eta \partial L / \partial \lambda_\ell)$

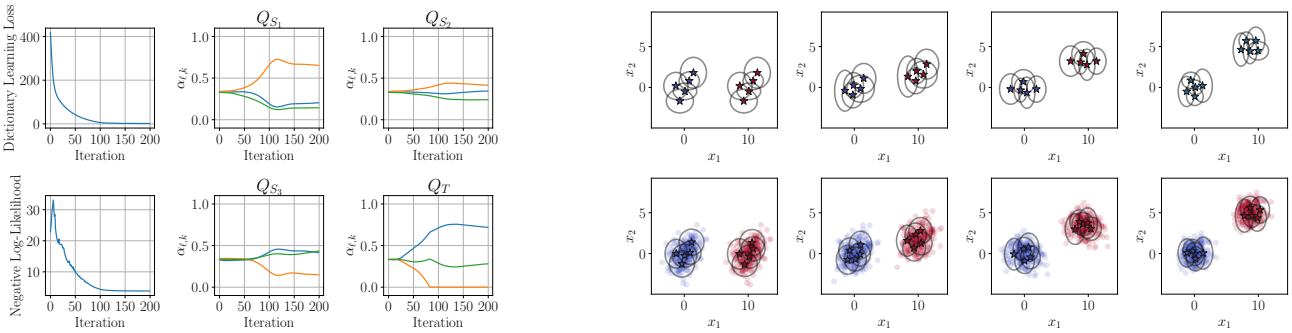
return $\mathcal{P}, \Lambda$

Remark 9. *(Faster Dataset Dictionary Learning.) The computational complexity of an optimization step of algorithm 13 corresponds to $\mathcal{O}(N \times N_{it} \times C \times K^3 \log K)$, i.e., we calculate N barycenters of C atoms. One should compare this complexity with that of DaDiL, i.e., $\mathcal{O}(N \times N_{it} \times M \times C \times n_b^3 \log n_b)$, where $M = \lceil n/n_b \rceil$ is the number of mini-batches sampled at each iteration. As a result,* ***we achieve a speed-up on the order of M while solving an unbatched OT problem****.*

Example 23. *(GMM-DaDiL on Gaussian Clusters) Here, we continue our discussion on the Gaussian clusters toy example, introduced in Example 13. Readers should compare the results here presented with Examples 13, 19, and 22. We start by showing in Figure 7.5 a summary of the optimization history of GMM-DaDiL (a), and the final reconstruction of domain GMMs (b).*

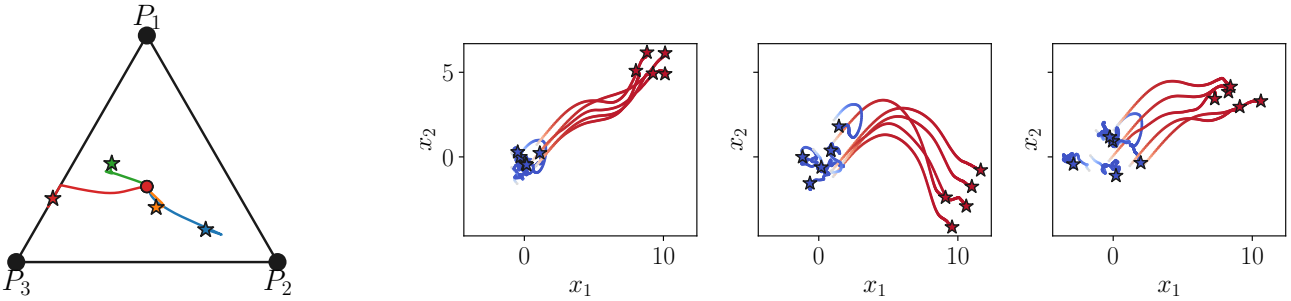
Based on Figure 7.5, we see that the GMM-DaDiL objective (equation 7.10) is effectively minimized by gradient descent. We further track the negative log-likelihood of each reconstruction, i.e., $\mathcal{B}(\lambda_\ell, \mathcal{P})$ with respect the data of each domain. Although this quantity is not directly minimized by GMM-DaDiL, there is a decrease of it as the optimization proceeds. Curiously, there is a peak at the beginning of the optimization process, due the shrinkage of standard deviations.

We now analyze the evolution of our dictionary $(\Lambda, \mathcal{P})$ over the course of gradient descent, as shown in Figure 7.6. In (a), we se that the barycentric coordinates converge as the optimization progresses. In (b), note that the atoms components $(\mu_k^{(P_c)}, \nu_k^{(P_c)})$ are split into the two classes of the classification



(a) Optimization summary. (b) Reconstructions.

Figure 7.5 – Overview of GMM-DaDiL. In (a), we show a summary of the gradient descent history, with the dictionary learning loss (objective function), barycentric coordinates, and negative log-likelihood. In (b), we show the reconstruction of domain GMMs at the end of the optimization process.



(a) Evolution of Λ. (b) Evolution of $\mathcal{P}$.

Figure 7.6 – Evolution of dictionary variables. In (a), we show the evolution of barycentric coordinates over Δ_3. In (b), we show the evolution of each atom as the gradient descent process progresses. The positions are encoded by the points in $\mathbb{R}^2$. The labels are encoded through the probability of assignment of class 1, which ranges from 0 (blue), to 1 (red).

problem. Their soft labels, $\nu_k^{(P_c)}$, get increasingly confident with the iterations.

Here, we highlight two advantages with respect empirical DaDiL. First, due the compressive effect of using GMMs to represent probability measures, we do not need to use mini-batches, which leads to simpler optimization trajectories. Second, while using the complete GMMs during optimization, GMM-DaDiL does not suffer from label uncertainty as fullbatch DaDiL (e.g., Figure 6.8).

Example 24. *(Lighter MSDA) In this example, we illustrate that, through the GMM-OT framework, we are able to achieve better performance than WBT and DaDiL.*

Throughout this example we use the Office-Home benchmark (see Chapter 4, section 4.5). We compare methods based on their domain adaptation performance, i.e., the average classification accuracy while leaving one domain out (e.g., $(Ar, Cl, Pr) \rightarrow Rw$). We thus measure this quantity for GMM-WBT and GMM-DaDiL, as a function of number of components K in the GMMs. This parameter is analogous to the number of samples in WBT and DaDiL. We therefore compare these four algorithms for K and $n \in \{65, 130, 195, \cdots, 910\}$. We show our results in Figure 7.7. From this figure, we see that GMM-DaDiL surpasses all other methods over the entire range of K, which showcases the advantage of using GMMs and dictionary learning. Curiously, the performance of GMM-WBT and WBT are quite similar. Indeed, recent studies [33] show that Wasserstein barycenters are effective in compressing probability measures with respect the number of their samples. As a result, in this adaptation task, the GMM version of WBT has similar performance to the empirical version.

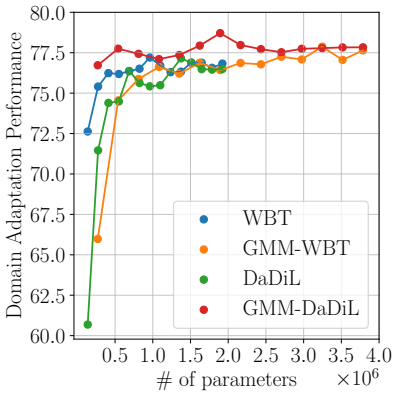


Figure 7.7 – Comparison of domain adaptation performance and number of parameters in GMM-WBT, GMM-DaDiL and its empirical counterparts.

Example 25. *(Better MSDA) In this example, we illustrate the advantages of dictionary learning for MSDA. Especially, we ablate the learning of a set of atoms $\mathcal{P}$.*

We compare the classification accuracy on the target domain of the mixture-Wasserstein hull $\mathcal{M}(\mathcal{P})$ and $\mathcal{M}(\mathcal{Q}_S)$. Furthermore, since Q_T might not actually belong to $\mathcal{M}(\mathcal{Q}_S)$, we further transport the measures $Q \in \mathcal{M}(\mathcal{Q}_S)$ towards Q_T, using equation 7.9. This corresponds to applying GMM-WBT with barycentric coordinates λ. Here, we focus on the Office 31 benchmark, wich has only 3 domains. We use the adaptation task $(W, D) \rightarrow A$. As such, we can parametrize the measures in $\mathcal{M}(\mathcal{P})$ through a variable $t \in [0, 1]$, so that $\lambda = (t, 1-t)$. We show a summary of our results in Figure 7.8. Here, we note two immediate conclusions. First, GMM-DaDiL learns atoms that are different from the source domains. This is evidenced by the different classification accuracies at $t = 0$ and $t = 1$. Indeed, one of the atoms learned by DaDiL has better performance than both of the source domains. Second, the mixture-Wasserstein hull $\mathcal{M}(\mathcal{P})$ learned by GMM-DaDiL is better at adapting towards the target, for a wide

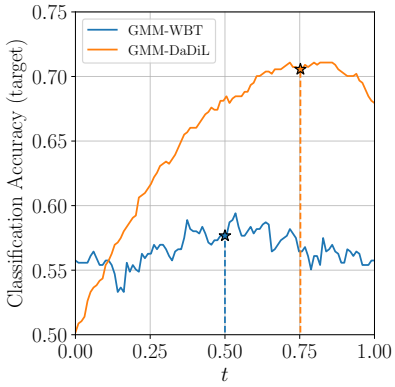


Figure 7.8 – Classification accuracy on the taget domain over the mixture-Wasserstein hull of GMM-WBT (blue) and GMM-DaDiL (orange).

range of values of t. Overall, GMM-DaDiL manages to learn GMMs whose interpolation are better than actual source domain GMMs.

Example 26. *(Faster MSDA) In this example, we show that GMM-DaDiL is faster than empirical DaDiL.*

We plot the running time of these methods on the Office 31 benchmark, for the $(D, W) \to A$ adaptation task. The variables that influence the complexity of GMM and empirical DaDiL are the number of components K and the batch size n_b, respectively. For GMM-DaDiL, we simply measure its running time for 5 independent runs of the algorithm (blue curve) for each $K \in \{31, 62, \cdots, 217\}$. For DaDiL, we set $n_b \in \{31, 62, \cdots, 217\}$, and set $n = M \times n_b$, where M is the number of mini-batches. We measure the performance over 5 independent runs as well. Other than these parameters, we fix $N_{iter} = 50$ and $C = 3$. As shown in Figure 7.9 (c), the running time of GMM-DaDiL and DaDiL are essentially equivalent for $M = 1$. For $M > 1$, we have a speedup that is proportional to M.

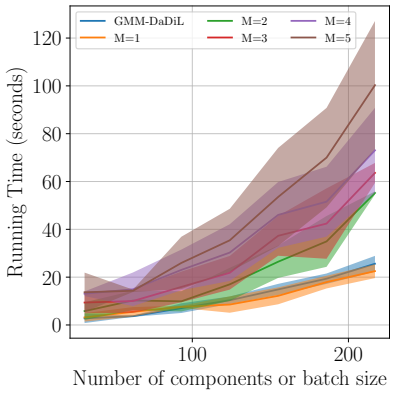


Figure 7.9 – Classification accuracy on the taget domain over the mixture-Wasserstein hull of GMM-WBT (blue) and GMM-DaDiL (orange).

7.5 Conclusion

In this chapter, we presented a series of contributions of the GMM-OT framework of [17] for DA. We started with 2 simple single source DA strategies, relying on label propagation of GMM components, and a mapping between GMM samples. These 2 approaches can be seen as an extesion of label propagation [24] and barycentric mapping [23], already proposed in the context of empirical OT. The use of GMMs presents a few advantages. First, it may need less parameters to encode probability measures, especially when using axis-aligned Gaussians. Second, the use of GMMs naturally promotes a transport that maps close samples together to the target domain. This result is due the grouping of points into GMM components.

Besides our strategies for single source DA, we introduced 2 GMM counterparts of our algorithms introduced in Chapters 5 and 6. These are the GMM-WBT and GMM-DaDiL strategies. To effectively implement those strategies, we proposed 2 components that were missing in the literature. First, we introduced the so-called joint mixture-Wasserstein distance, $\mathcal{JMW}$, that computes a Wasserstein-like distance between labeled GMMs. Second, we needed an efficient algorithm for calculating mixture and joint mixture-Wasserstein barycenters. As we covered in Chapter 3, [17] previously proposed a strategy, based on MMOT, for mixture-Wasserstein barycenters. However, in our applications, solving a MMOT problem is unfeasible. We thus resorted to a strategy similar to what was proposed in [26], through the 1-st order analysis of the mixture-Wasserstein distance.

Overall, this chapter introduced a GMM dictionary notion that is faster than empirical DaDiL. As a result, we use standard gradient descent for optimizing the objective function of GMM-DaDiL, with respect the parameters of atoms (means, standard deviations and labels). Besides being faster (Remark 9 and Example 26), GMM-DaDiL is also lighter than its empirical version (Example 24), and learning atoms actually induces an interpolation space more appropriate to DA than the actual source domain data (Example 25).

Part III

Applications

Chapter 8

Cross-Domain Fault Diagnosis

Contents

As discussed by [144], within process supervision, faults are unpermitted deviations of a characteristic property or variables of a system. Furthermore, there is an increasing demand on reliability and safety of technical plants, leading to the necessity of methods for supervision and monitoring. These are Fault Detection and Diagnosis (FDD) methods, which comprise the *detection*, i.e., if and when a fault has occurred, and the *diagnosis*, i.e., the determination of *which fault* has occurred. In this chapter, we focus on Automatic Fault Diagnosis (AFD) systems, assuming that faults were previously detected accordingly.

In parallel, machine learning is a field of artificial intelligence, that defines predictive models based on data. Nonetheless, these models make an implicit assumption, that training and test data come from the same probability distribution, which is seldom verified in practice [7], as both training and test data may be collected under heterogeneous conditions that drive shifts in probability distributions. This phenomenon motivates the field of transfer learning [145] to propose algorithms that are robust to *distributional shift*[1].

There is a straightforward link between machine learning and AFD systems, as one can understand fault diagnosis as a classification problem. In this sense, one uses sensor data (e.g., temperature, concentration, flow-rate) as inputs to a classifier, which predicts the corresponding fault, or its absence [94]. Further, transfer learning is a broad field within machine learning, in which knowledge is

1. Our code is available at https://github.com/eddardd/tep-domain-adaptation, and our data is hosted on Kaggle, https://www.kaggle.com/datasets/eddardd/tennessee-eastman-process-domain-adaptation

transferred from a source to a target context. Within transfer learning, DA is a common framework where one has access to labeled data from a source domain, and unlabeled data from a target domain. Thus, DA seeks improving classification accuracy on target domain data. In many cases, source data is itself heterogeneous, following multiple probability distributions. This setting is known as MSDA.

In this chapter, we propose a new benchmark, based on the Tennessee Eastman Process (TEP) [29, 32], a complex, large-scale chemical process used by the chemical engineering community for benchmarking control systems, as well as FDD techniques. This process is interesting for DA, as it may operate at different modes of production. As we show in our case study (section 8.1), the different modes of production have different probability distributions, thus the need for DA techniques for improving generalization. We further benchmark existing techniques in DA, either based on pre-extracted features (shallow DA), or through deep learning (deep DA).

The rest of this chapter is divided as follows. Section 8.1 presents our case study, the TEP. Section 8.2 details our experiments with the TEP benchmark, especially on domain adaptation. Finally, section 8.3 concludes this chapter.

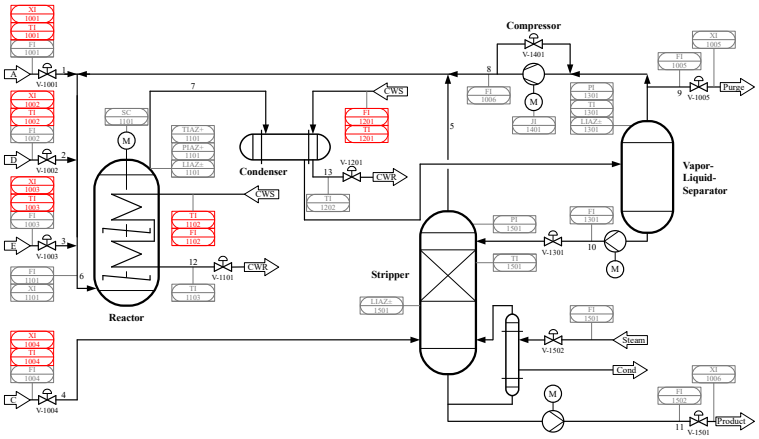


Figure 8.1 – **P&ID diagram for the TEP.** Figure reproduced from [147], which shows the main components of the process. Measurements originally introduced by [29] are shown in gray, whereas the measurements introduced by [147] are shown in red. A simulation environment, based on this diagram, is described in [32].

8.1 Case Study

The TEP was first introduced by [29] as a realistic benchmark for the design of control and monitoring systems. From the perspective of fault detection and diagnosis [146], this system is widely used by the academic community. Henceforth, we follow the description of [32]. The TEP consists on the production of two liquid product components, noted as G and H, from 4 gaseous reactants A, C, D and E, with an additional inert B and a byproduct F. These components are related through 4

exothermic and irreversible reactions,

$$\begin{aligned} A(\text{g}) + C(\text{g}) + D(\text{g}) &\rightarrow G(\text{liq}) \quad \text{Product 1,} \\ A(\text{g}) + C(\text{g}) + E(\text{g}) &\rightarrow H(\text{liq}) \quad \text{Product 2,} \\ A(\text{g}) + E(\text{g}) &\rightarrow F(\text{liq}) \quad \text{Byproduct,} \\ 3D(\text{g}) &\rightarrow 2F(\text{liq}) \quad \text{Byproduct.} \end{aligned} \tag{8.1}$$

The chemical plant is composed by five major process units: reactor, product condenser, vapor-liquid separator, recycle compressor and product stripper (c.f., Figure 8.1). Based on the reactions in equation 8.1, there are 6 different *modes of operation*, which correspond to 3 different G/H mass ratios, and a desired product rate. The different modes of operation are shown in Table 8.1.

Table 8.1 – TEP operation modes, as described in [29]. In our experiments, each mode of operation corresponds to a different domain.

Mode	Mass Ratio	Production rate
1	50/50	7038 kg h^{-1} G and 7038 kg h^{-1} H
2	10/90	1408 kg h^{-1} G and 12,669 kg h^{-1} H
3	90/10	10,000 kg h^{-1} G and 1111 kg h^{-1} H
4	50/50	maximum production rate
5	10/90	maximum production rate
6	90/10	maximum production rate

From the perspective of DA, each mode of operation induces changes in the statistical properties of the data. As a result, a model learned with historical data from a set of operation modes (e.g., $1, \cdots, 5$) may not generalize to a new operation mode (e.g., 6). At the same time, collecting labeled data at the new operation mode is costly. MSDA is thus a natural solution, where one leverages historical data from previous modes to learn a better model on the new mode, only requiring unlabeled data on the new operation conditions (c.f., Table 8.1).

Table 8.2 – Description of process variables of the TEP. Variables are divided into measurements (XME) and manipulated (XMV).

Variable	Description	Variable	Description	Variable	Description	Variable	Description
XME(1)	A Feed (kscmh)	XME(15)	Stripper Level (%)	XME(29)	Component A in Purge (mol %)	XMV(2)	E Feed (%)
XME(2)	D Feed (kg/h)	XME(16)	Stripper Pressure (kPa gauge)	XME(30)	Component B in Purge (mol %)	XMV(3)	A Feed (%)
XME(3)	E Feed (kg/h)	XME(17)	Stripper Underflow (m^3/h)	XME(31)	Component C in Purge (mol %)	XMV(4)	A & C Feed (%)
XME(4)	A & C Feed (kg/h)	XME(18)	Stripper Temp (°C)	XME(32)	Component D in Purge (mol %)	XMV(5)	Compressor recycle valve (%)
XME(5)	Recycle Flow (kscmh)	XME(19)	Stripper Steam Flow (kg/h)	XME(33)	Component E in Purge (mol %)	XMV(6)	Purge valve (%)
XME(6)	Reactor Feed rate (kscmh)	XME(20)	Compressor Work (kW)	XME(34)	Component F in Purge (mol %)	XMV(7)	Separator liquid flow (%)
XME(7)	Reactor Pressure (kscmh)	XME(21)	Reactor Coolant Temp (°C)	XME(35)	Component G in Purge (mol %)	XMV(8)	Stripper liquid flow (%)
XME(8)	Reactor Level (%)	XME(22)	Separator Coolant Temp (°C)	XME(36)	Component H in Purge (mol %)	XMV(9)	Stripper steam valve (%)
XME(9)	Reactor Temperature (°C)	XME(23)	Component A to Reactor (mol %)	XME(37)	Component D in Product (mol %)	XMV(10)	Reactor coolant (%)
XME(10)	Purge Rate (kscmh)	XME(24)	Component B to Reactor (mol %)	XME(38)	Component E in Product (mol %)	XMV(11)	Condenser Coolant (%)
XME(11)	Product Sep Temp (°C)	XME(25)	Component C to Reactor (mol %)	XME(39)	Component F in Product (mol %)	XMV(12)	Agitator Speed (%)
XME(12)	Product Sep Level (%)	XME(26)	Component D to Reactor (mol %)	XME(40)	Component G in Product (mol %)		
XME(13)	Product Sep Pressure (kPa gauge)	XME(27)	Component E to Reactor (mol %)	XME(41)	Component H in Product (mol %)		
XME(14)	Product Sep Underflow (m^3/h)	XME(28)	Component F to Reactor (mol %)	XMV(1)	D Feed (%)		

To build an AFD system, we need to collect data from a set of sensors, then categorize the data into a set of faults. In this chapter, we use the data provided by [32]. In their simulations, there are

53 sensors in the overall plant, corresponding to different physical and chemical quantities. We group these variables into measurements (denoted XME(i), for the $i-$th measurement) and manipulated (denoted XMV(j), for the $j-$th manipulation). In this dataset, the TEP system is simulated for a 100 hours, with a sampling rate of 3 minutes. As such, we use each simulation as a sample in our MSDA benchmark. In each simulation, faults are introduced after 600 time steps, i.e., 30 hours. Concerning the type of faults, in their initial publication, [29] presents 20 types of process disturbances, affecting different process variables. In addition to these initial faults, [147] proposed 8 additional faults under the type *random variation*. These faults are shown in Table 8.3.

Table 8.3 – Description and types of faults for the TEP in the simulation environment of [32]. Faults are grouped into 4 types: step, random variation (RV), sticking and unknown.

Fault	Variable	Type	Fault Class	Variable	Type
1	A/C feed ratio, B composition constant	Step	15	Water outlet temperature (separator)	Sticking
2	B composition, A/C ratio constant	Step	16	Variation coefficient of the steam supply of the heat exchange of the stripper	RV
3	D feed temperature	Step	17	Variation coefficient of heat transfer (reactor)	RV
4	Water inlet temperature (reactor)	Step	18	Variation coefficient of heat transfer (condenser)	RV
5	Water inlet temperature (condenser)	Step	19	Unknown	Unknown
6	A feed loss	Step	20	Unknown	RV
7	C header pressure loss	Step	21	A feed temperature	RV
8	A/B/C composition of stream 4	RV	22	E feed temperature	RV
9	D feed temperature 4	RV	23	A feed flow	RV
10	C feed temperature	RV	24	D feed flow	RV
11	Water outlet temperature (reactor)	RV	25	E feed flow	RV
12	Water outlet temperature (separator)	RV	26	A & C feed flow	RV
13	Reaction kinetics	RV	27	Water flow (reactor)	RV
14	Water outlet temperature (reactor)	Sticking	28	Water flow (condenser)	RV

8.1.1 Benchmark preparation

Data Cleaning. For each mode, the simulations provided by [32] are divided into 3 groups: set-point variation, mode transitions and single fault. In the first case, the authors change the initial simulation set-point using a step or ramp function. In the second case, the simulation changes from one mode to another at an instant in time. In the third case, as previously mentioned, a fault is introduced at time step 600, i.e., after 30 hours of simulation. For each fault, there are multiple intensities available (e.g., 25%, 50%, 75% and 100% fault magnitude). For magnitudes 25%, 50% and 75%, the system is simulated 100 times, whereas for 100%, the system is simulated 200 times. As a result, for the

single-fault scenario only, the data provided by [32] contains,

$$28 \text{ faults} \times 6 \text{ modes} \times 500 \text{ simulations} = 84000 \text{ simulations}.$$

Nonetheless, one should note that some simulations terminate earlier than $100h$, due to forced plant-shutdown. We filter out these simulations. As a result, we adopt the following step: for each fault, we keep the first 100 simulations of highest magnitude that terminate successfully. For each selected simulation, we crop the signal into two parts. The first 30h (600 time steps) correspond to the steady state, determined by the set point of the mode of operation. This first part of the signal characterizes the healthy state of the system (i.e., faultless state). We further sub-sample the number of faultless state signals to keep a balanced dataset (i.e., 100 per mode of operation). The second part consists on the next 30h of simulation. Since faults are introduced exactly at the 601th time step, the second part of the signal characterize each fault. This process generates a slightly imbalanced dataset of 17289 samples. We summarize the division of samples among modes of operation in Table 8.4.

Table 8.4 – Number and percentage of samples from each mode of operation.

Mode of Operation	# of Samples	% of Samples
1	2900	16.77
2	2845	16.45
3	2899	16.76
4	2865	16.57
5	2883	16.67
6	2897	16.75
Total	17289	100

Variable selection and pre-processing. Out of the 53 variables presented in Table 8.2 some of these variables are not continuous (e.g., XME(23) through XME(41)). Given this remark, we follow [32], and consider a sub-set of 34 continuous signals as input to our neural nets. These are measurements XME(1) through XME(22), and manipulated variables XMV(1) through XMV(12). We thus have multi-variate time series of shape $(34, 600)$, where 34 is the number of sensor readings (i.e., considered variables), and 600 corresponds to the number of time steps (T). Before training a neural net on the benchmark, we further perform a standardization along each variable, i.e.,

$$x_{i,j,t}^{(P_\ell)} = \frac{x_{i,j,t}^{(P_\ell)} - \mu_j^{(P_\ell)}}{\sigma_j^{(P_\ell)}}, \; \mu_j^{(P_\ell)} = \frac{1}{n_\ell T} \sum_{i=1}^{n_\ell} \sum_{t=1}^{T} x_{i,j,t}^{(P_\ell)}, \; \sigma_{i,j,t}^{(P_\ell)} = \sqrt{\frac{1}{n_\ell T} \sum_{i=1}^{n_\ell} \sum_{t=1}^{T} (x_{i,j,t}^{(P_\ell)} - \mu_j^{(P_\ell)})^2},$$

where n_ℓ is the number of samples for a given mode (c.f., Table 8.4).

Neural Network Backbone. In DA, it is common to choose a backbone upon which methods will rely on. For instance, in image processing, residual networks [148] are widely used. In the context of time series, general purpose backbones are not as straightforward. For the TEP data, different kinds architectures can be considered, such as recurrent, convolutional nets or transformers [149]. In the following, we employ a Fully Convolutional Network (FCN) [150, 151, 152], which consists on three

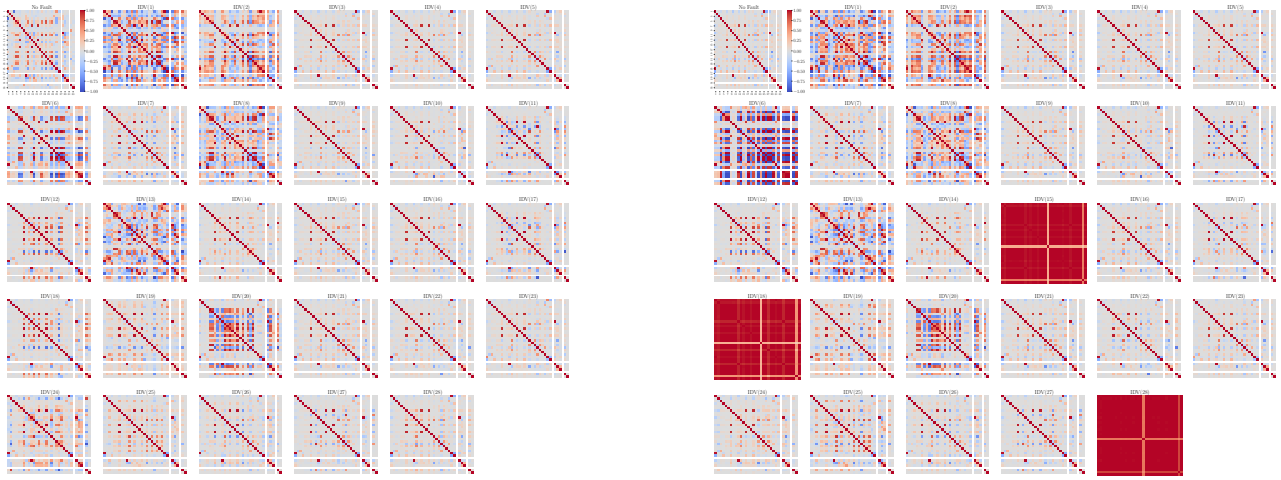
convolutional blocks followed by a Global Average Pooling (GAP) layer. Each convolutional block has a convolutional layer, and a normalization layer. In our experiments we verified that instance normalization [153] improves stability and performance over other normalization layers such as batch normalization [154].

8.2 Experiments

8.2.1 Exploratory Data Analysis

We perform an exploratory data analysis to better understand the operations performed on the data of [32]. We establish the distributional shift occurring between the different modes of production (c.f., Table 8.1).

Qualitative Analysis. We start by analyzing the correlations between pairs of variables, conditioned on the type of fault, that is, IDV(1) through IDV(28), and the no-fault scenario. In Fig. 8.2, we illustrate a change in the pattern of correlations between variables, conditioned on the fault type, for modes 1 and 2. In comparison, the correlation patterns drastically change for faults 15, 18 and 28, corresponding to a sticking fault on the water outlet temperate on the separator, a random variation on the heat transfer coefficient on the condenser and a random variation on the water flow on the condenser, respectively. As a result, the mode of production deeply impacts the dynamic of the system, which creates a shift in distribution between data from these modes.

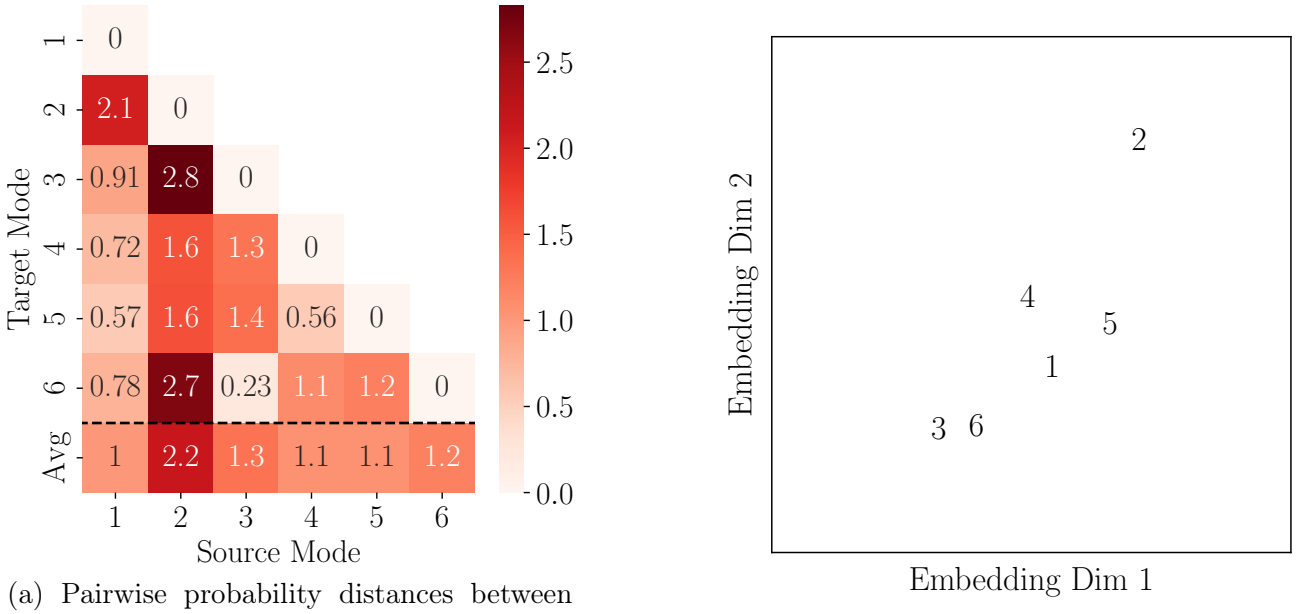


(a) Mode 1 (b) Mode 2

Figure 8.2 – **Qualitative analysis of distributional shift.** In (a) and (b), we show the correlation between different variables in TEP, for modes of production 1 and 2, conditioned on the type of fault, i.e., IDV(1) through IDV(28). On each correlation matrix, the coefficient $\rho_{jj'}$ corresponds to the Pearson correlation coefficient between $\{x_{j,t}\}_{t=1}^{600}$ and $\{x_{j',t}\}_{t=1}^{600}$ across simulations.

Quantitative Analysis. We quantify the shift between pairs of modes through the probability

metrics introduced in Chapter 3. We estimate the pairwise Wasserstein distances between modes using equation 2.12. We show our quantitative results in Fig. 8.3. In Fig. 8.3 (a), we show the pairwise distance in probability distribution between different modes. On one hand, the most different mode with respect others is Mode 2, which is especially far from modes 1, 3 and 5. On the other hand, the most similar modes are 3 and 6. We can have a better picture about the level of similarity of these different domains by embedding them on the plane, as shown in Fig. 8.3 (b). We obtain these embeddings through Multi-Dimensional Scaling (MDS) [155], which defines points in $\mathbb{R}^2$ while preserving the pairwise distances between the embeddings.



(a) Pairwise probability distances between different modes.

(b) MDS embeddings for different modes.

Figure 8.3 – **Quantitative analysis of distributional shift**. In (a), we measure the Wasserstein distance between pairs of modes. In (b), we obtain embeddings for the modes based on their pairwise distances.

From our qualitative and quantitative analysis, we expect lower performances with respect the adaptation towards mode 2, as it is the most dissimilar from other modes (c.f., Fig. 8.3 (a), average row). In contrast, adaptation between modes $(3, 6)$, and $(1, 4, 5)$ should work well as these modes share statistical characteristics. We verify these indications empirically in the next sections.

8.2.2 Single-Source Domain Adaptation

In this section, we explore single-source DA, i.e., when adaptation is done from a single source mode, to a single target mode. On the one hand, we refer to *generalization*, to the ability of a classifier to perform well on unseen data from an unseen domain. On the other hand, we refer to *adaptation*, when a classifier performs well on unseen data from the target domain.

In this context, we have 2 baselines. The first, *source-only*, considers that a classifier is learned exclusively with source domain data (i.e., no adaptation). This corresponds to the off-diagonals of

Figure 8.4 (a). Second, we have *target-only*, which trains and evaluates models on the target domain (i.e., no distributional shift). This corresponds to the diagonal of Figure 8.4 (a). With respect these scenarios, we verify our previous remarks, i.e., generalization towards mode 2 is much more difficult than other modes, and the clusters of similar modes (e.g., $(3, 6)$ and $(1, 4, 5)$) generalize well.

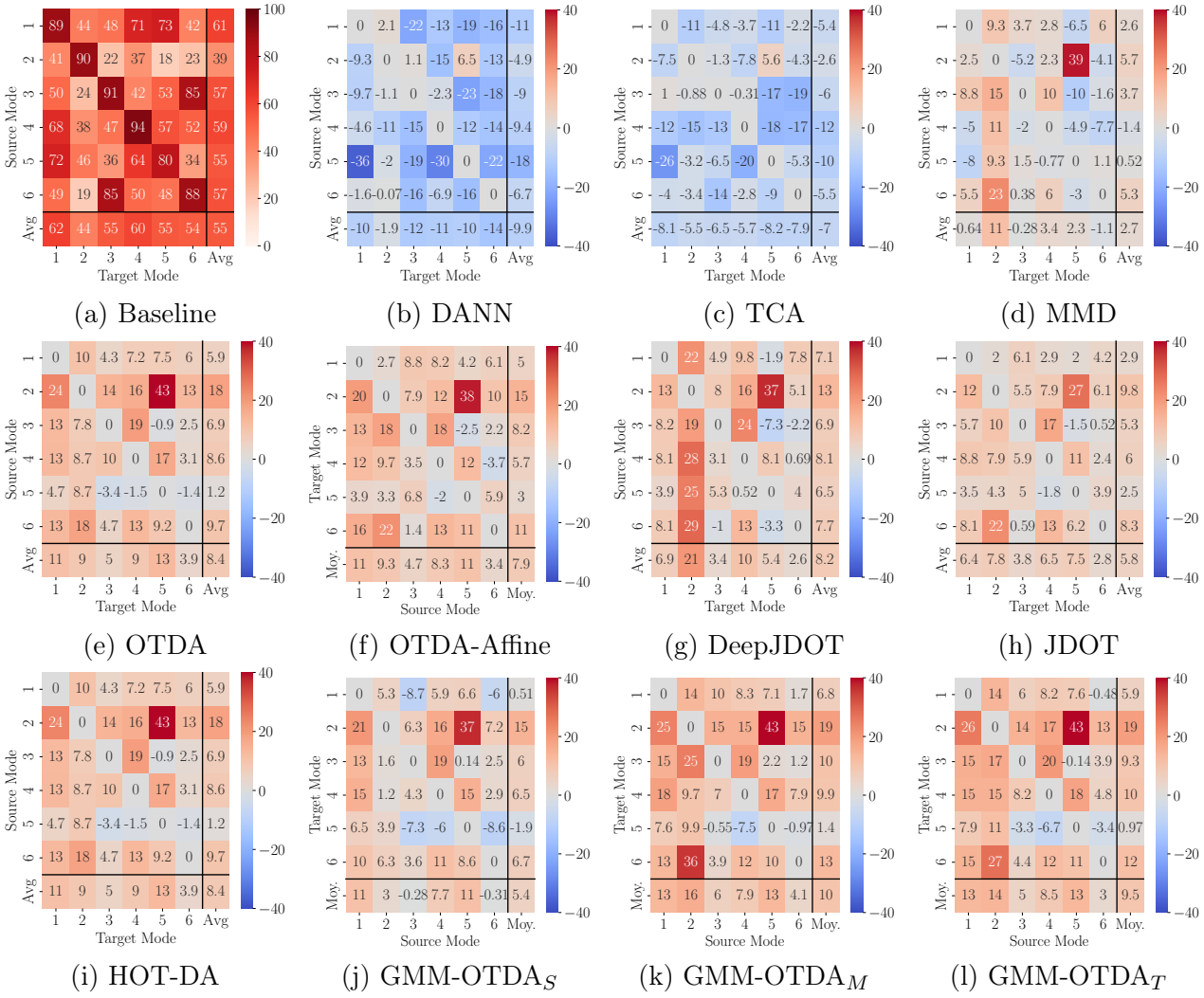


Figure 8.4 – **Results on single-source domain adaptation.** Performance of source-only (a), and domain adaptation algorithms (b - l). For domain adaptation algorithms, the entries show the different in performance with respect the source-only baseline. As such, values can be negative (negative transfer), or positive (successful adaptation).

We further compare the single-source DA methods presented in Chapter 4, which are shown in Fig. 8.4 (b) through (l). Overall, we find that OT-based methods have a higher performance than other metrics (e.g., the MMD). This is similar to previous findings on smaller scale problems, such as [104]. The best performing method is GMM-OTDA [16], showing that GMM modeling improves the mapping estimation. Furthermore, the labeled GMM acquired in the target domain outperforms

other methods. Nonetheless, one should be mindful of *negative transfer* [156] between similar modes (e.g., $3 \rightarrow 5$), which may result in performance degradation.

8.2.3 Multi-Source Domain Adaptation

In this section, we explore multi-source DA, i.e., when adaptation is done from multiple source domains towards a single target. Here, note that the models have much more labeled available data, as all domains are considered at once. We start our discussion by comparing the performance of single, source-only baselines, and the corresponding multi-source baseline for each target mode, which is shown in Fig. 8.5. Overall, the multi source-only baseline improve over single-only for the same target. These baselines have similar performance when there are pairs of highly similar modes (e.g., modes 3 and 6), showing that extra data from additional modes is not as informative for generalization.

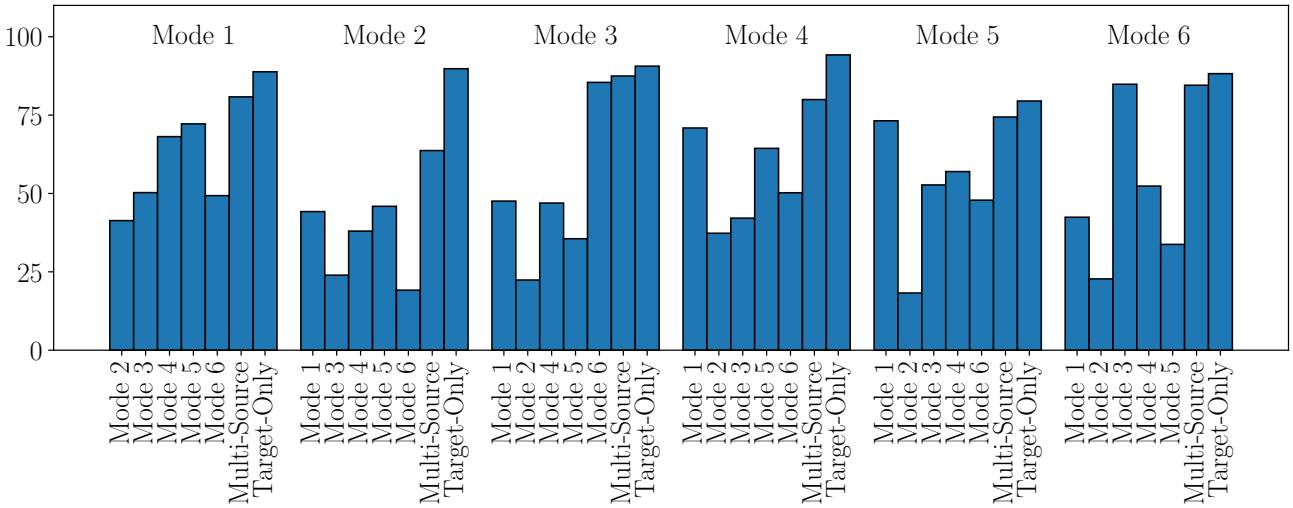


Figure 8.5 – **Multi and single-source baseline comparison.** On top, we show the target domain. In the abscissa, we show the corresponding baseline. The multi-source scenario generally improves over the single source-only case.

We now consider the performance of DA algorithms in the multi-source setting. Besides native MSDA algorithms, i.e., algorithms that suppose the source as composed by different domains, we also consider single-source algorithms with access to the concatenation of all source domains. Our comparison is shown in Fig. 8.6. A first question is whether access to additional data is beneficial to adaptation. For instance, in single-source DA, methods exhibited negative transfer in the task $3 \rightarrow 5$. When provided access to data from all domains, all single-source adaptation method performance improved over the single-source baseline. As a result, even though data from multiple domains may not improve generalization, it does improves adaptation.

With respect Fig. 8.6, from the perspective of MSDA, methods that weight sources in a linear space, such as WJDOT, or in a Wasserstein space, such as WBT and DaDiL outperform the weighting of classifiers' predictions, such as M3SDA$_\beta$ [112]. On the one hand, WJDOT can filter undesirable information during adaptation by assigning small weights to domains and samples. On the other

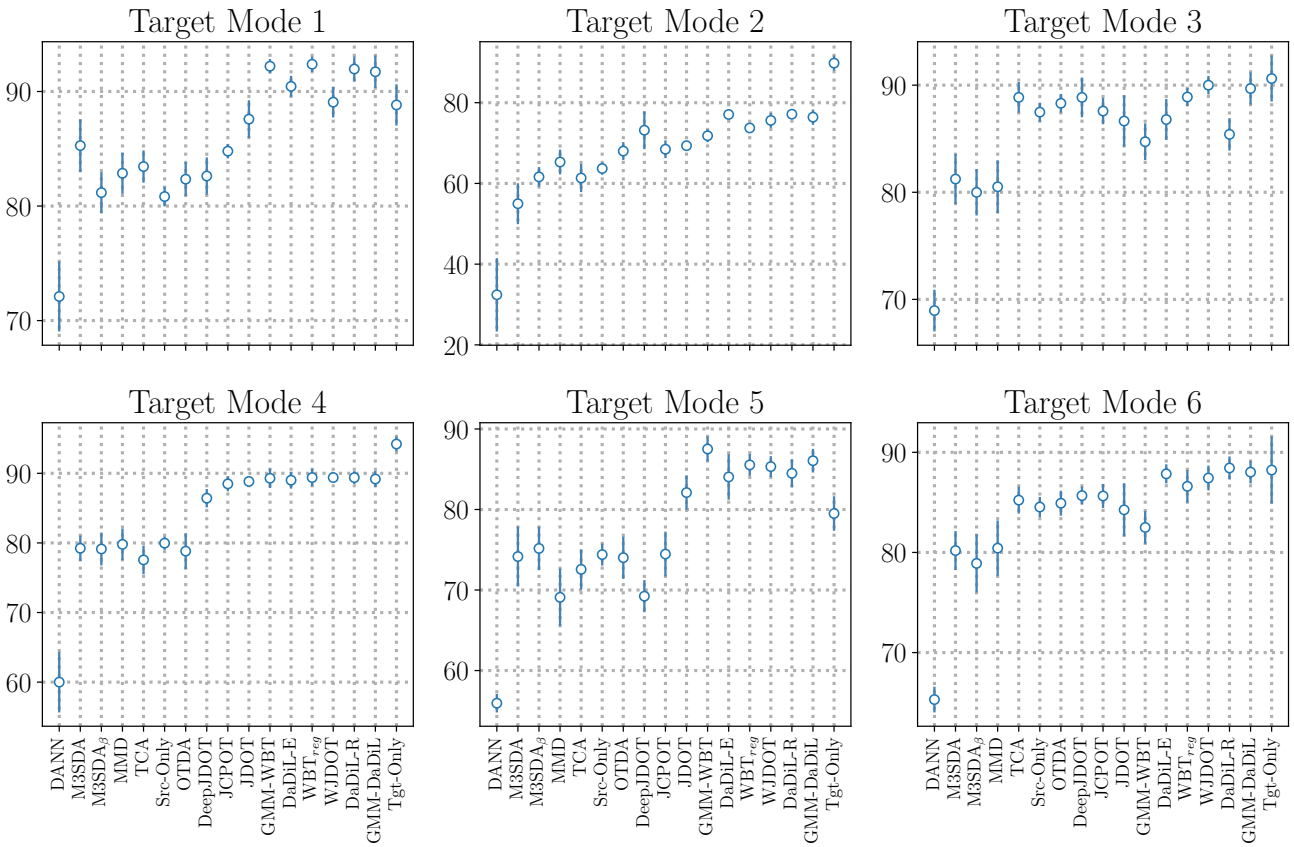


Figure 8.6 – **Multi-source domain adaptation results.** We compare all algorithms with access to labeled data from all source domains, except the target mode, from which we have access to unlabeled data. Methods in the abscissa are ordered by average performance on all modes.

hand, WBT and DaDiL combine the information in the sources non-linearly. These two strategies are effective in domain adaptation.

From Fig. 8.7, we can see that shallow DA methods (e.g., JDOT) generally improve over deep DA methods (e.g., DeepJDOT). Indeed, deep DA methods learn features that are invariant to the domain shift between different modes. As a result, these features may be less useful for classification. In a general note (both single, and multi-source methods), OT-based techniques outperform methods based on other distances, such as the MMD and $d_{\mathcal{H}}$.

Finally, we note that, through our GMM modeling, that is, GMM-DaDiL, we manage to achieve the best average domain adaptation performance across domains, without actually having the best performance on any particular domain. This is a result of the *performance stability*, across different intensities of distributional shift. For instance, while DaDiL under-performs other methods in Target Mode 3, GMM-DaDiL approaches the target-only performance.

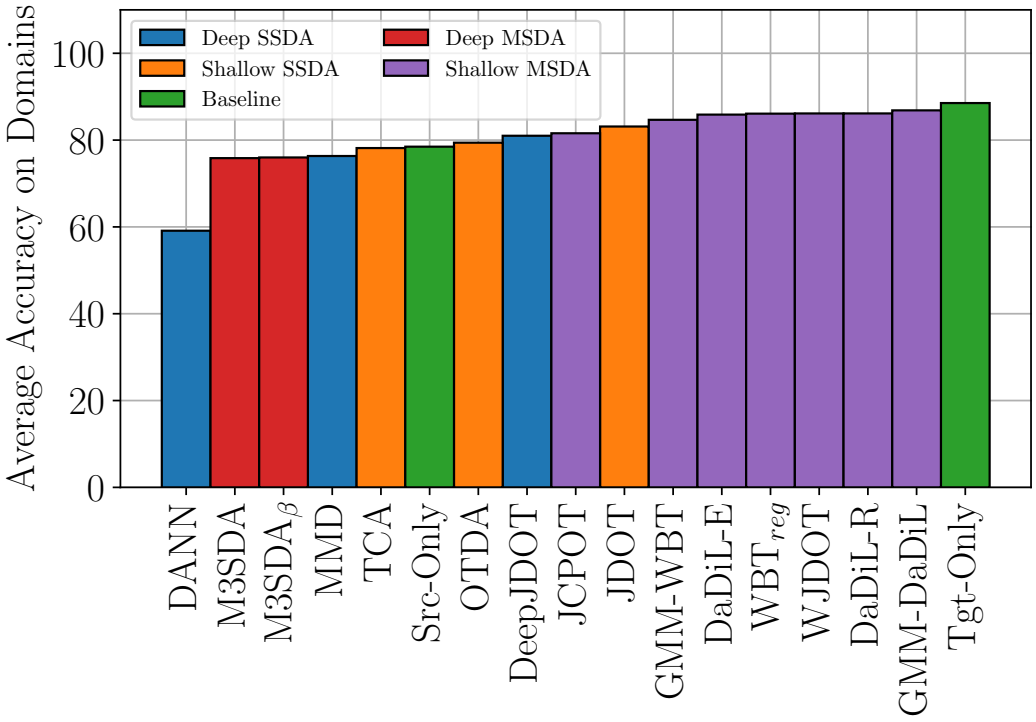


Figure 8.7 – Comparison of average adaptation performance of DA algorithms in the multi-source setting.

8.3 Conclusion

In this chapter, we introduce a new benchmark for domain adaptation algorithms based on the Tenessee Eastman process [29]. The present benchmark is created by applying pre-processing steps on the simulations provided by [32] (c.f., section 8.1.1), thus creating a large scale dataset of time series. These time series are associated with different modes of production. Based on each mode of production, the statistical properties of the time series change (c.f., Fig. 8.2) creating a shift in the data probability distribution (c.f., Fig. 8.3). As a result, data trained on a specific mode may not generalize well to other modes of production, thus the need for domain adaptation. Through a series of experiments with single-source and multi-source domain adaptation methods, we show that OT-based methods outperform methods that rely on the maximum mean discrepancy, and $\mathcal{H}-$distances, which agrees with previous findings on smaller scale systems [104]. Besides providing the open source code for the reproduction of our benchmark, with this work we hope to encourage research on the intersection between domain adaptation and fault diagnosis [94].

Chapter 9

Benchmarking Domain Adaptation

Contents

In this section, we cover our experiments with DA. We focus on three settings: single source, multi source DA (see, e.g., Figure 4.3 for an overview of the differences),

Single Source DA. Algorithms are provided with labeled data from a single domain, Q_S, and must adapt towards a target domain, Q_T, based on unlabeled data from this measure.

Multi Source DA. Algorithms are provided with labeled data from N domains, $\{Q_{S_\ell}\}_{\ell=1}^N$, and must adapt towards a target domain, Q_T, based on unlabeled data from this measure.

On top of these categories, one may further make a distinction based on how adaptation is carried out, namely, between shallow and deep DA. In both cases, let us assume that the classifier $f : \mathcal{X} \to \mathcal{Y}$ is divided into an encoder, $\phi : \mathcal{X} \to \mathcal{Z}$, where $\mathcal{Z}$ is a latent space, and a feature classifier, $h : \mathcal{Z} \to \mathcal{Y}$. In this case, f is implemented by the composition $f(\mathbf{x}) = h(\phi(\mathbf{x}))$. f is further *parametrized* by the weights θ_ϕ and θ_h. As a result,

Shallow DA. One fixes θ_ϕ, then extracts $\{\mathbf{z}_i^{(Q_S)}\}_{i=1}^{n_S}$ and $\{\mathbf{z}_j^{(Q_T)}\}_{j=1}^{n_T}$. Adaptation is then carried out between matrices $\mathbf{Z}^{(Q_S)} \in \mathbb{R}^{n_S \times d}$ and $\mathbf{Z}^{(Q_T)} \in \mathbb{R}^{n_T \times d}$, where d is the dimensionality of the latent space.

Deep DA. Rather than fixing θ_ϕ, these kinds of methods push the encoder to learn features that are invariant to the distributional shift between Q_S and Q_T. In other words, they minimize an additional term $\mathcal{D}(\phi_\sharp Q_S, \phi_\sharp Q_T)$, which accounts for the *domain discrepancy* after encoding.

We refer readers to section 4.4.5 in Chapter 4, where we presented a brief description of deep DA. Furthermore, Figure 4.8 illustrate the aforementioned concepts.

In either case, DA methods must choose a *backbone*, i.e., an architecture for ϕ and h, so that methods are compared fairly for a fixed architecture. In comparison with deep DA, shallow DA methods tend to demand less memory, as gradients do not need to be back-propagated through the layers of the encoder network. Alternatively, it is possible to use hand-crafted features (e.g., SURF features [157] for images). In these cases, a comparison between shallow and deep methods is no longer possible. In this chapter, we use Residual Networks (ResNets) [148] for visual adaptation benchmarks (e.g., Office-like benchmarks), and a Multilayer Perceptron (MLP) [1] for signal processing ones (e.g., Case Western Reserve University (CWRU)).

Once the backbone has been defined, one is faced with 2 choices. First, one may train the network *from scratch*, that is, from a random initialization. This choice is highly challenging, as most DA benchmarks do not have enough images for pre-training large neural nets. For instance, Office-Home is a fairly sized benchmark in DA, however, it represents only 0.11% of the total amount of images in ImageNet [158]. Second, ony may use a pre-trained network on a general image recognition benchmark, such as ImageNet. This is the most practical case. Indeed, as previous works have shown [159, 109, 160], networks trained on such benchmarks learn general purpose filters, which can be re-used in downstream tasks (e.g., recognizing a handful of object classes).

From a *reproducibility perpsective*, it is of key importance to disclose, and even publish the weights (deep DA) or pre-extracted features (shallow DA) used in the experiments with DA algorithms. One can understand this points as follows. In shallow DA, algorithms rely on pre-extracted features for performing adaptation. If the features are not the same, a fair comparison cannot be established. In deep DA, training starts from a checkpoint. As a result, the overall results are sensitive to a change in checkpoint. A further complication includes the use, or not, of data augmentation.

Choice of Benchmarks. We evaluate OT-based DA methods on 4 image benchmarks; ImageCLEF, Caltech-Office 10, Office 31, and Office-Home; and one signal processing benchmark, CWRU. These were previously discussed in Chapter 4. With respect these benchmarks, they are ranked with an increasing number of samples, ranging from small benchmarks, such as ImageCLEF and Caltech-Office 10, which have around 2×10^3 images, to larger benchmarks, such as Office-Home and CWRU, which have around 1.5×10^4 and 2.4×10^4 respectively. We provide an overview of these benchmarks in Table 9.1.

9.1 Single-Source Domain Adaptation

Based on the baselines obtained in the previous section, we compare our GMM-OTDA strategy with 6 different strategies. A review of these methods was presented in Chapter 4, section 4.4.

— OTDA [124] (exact and Sinkhorn), consist of mapping source domain points to the target domain through the barycentric mapping.

1. In the case of a MLP with L layers, we consider the first $L-1$ layers as the encoder network, and the last layer as the classifier. The architecture consists of $2048 \to 1024 \to 512 \to 256$ with ReLU activations.

Benchmark	Domains	Backbone	# Samples	# Classes
ImageCLEF	Caltech (C) Bing (B) ImageNet (I) Pascal (P) Total	ResNet 50	600 600 600 600 2400	12
Caltech-Office 10	Amazon (A) dSLR (D) Webcam (W) Caltech (C) Total	ResNet 101	958 157 295 1123 2533	10
Office 31	Amazon (A) dSLR (D) Webcam (W) Total	Resnet 50	2817 498 795 4110	31
Office-Home	Art (Ar) Clipart (Cl) Product (Pr) Real World (Rw) Total	ResNet 101	2427 4365 4439 4357 15588	65
CWRU	1772rpm (A) 1750rpm (B) 1730rpm (C) Total	MLP	8000 8000 8000 24000	10

Table 9.1 – Overview of Domain Adaptation benchmarks

— Affine [67], consists of estimating an affine Monge map between source and target domain, by assuming that their respective measures are multivariate Gaussian measures.
— InfoOT [2] [107], uses KDE and information theory for defining a robust OT problem.
— HOT [3] [106], which defines an OT problem based hierarchically on data clusters.

Evaluation Protocol. As we shown in table 9.1, we consider benchmarks composed of multiple domains. We consider each pair of different domains within a given benchmark as an adaptation task. For instance, in the Office 31 benchmark, we have the adaptation task $A \to D$, which corresponds to using Amazon as the source domain, and dSLR as the target. As a result the ImageCLEF, Caltech-Office 10, Office 31, Office-Home and CWRU benchmarks have, respectively, 12, 12, 6, 12 and 6 tasks

2. https://github.com/chingyaoc/InfoOT
3. https://github.com/MouradElHamri/HOT-DA

each. For each adaptation task, we measure the classification accuracy (in %), computed as follows,

$$\% \text{ Accuracy}(y, \hat{y}) = 100\% \times \frac{1}{n} \sum_{i=1}^{n} \delta(y_i - \hat{y}_i). \tag{9.1}$$

This metric measures the percentage of correct predictions made by a classifier. For ranking different adaptation algorithms, we use the average classification accuracy over all tasks.

Hyper-parameters and model selection. Some strategies among the 6 compared methods have hyper-parameters that need tuning. This is the case of OTDA$_{Sink}$, InfoOT, HOT-DA and our GMM-OTDA. Note that OTDA$_{EMD}$ and OTDA$_{affine}$ do not have hyper-parameters. For OTDA$_{Sink}$, HOT-DA and GMM-OTDA, we tune the entropic penalty $\epsilon \in \{0, 10^{-2}, 10^{-1}\}$. Note that, for these methods, whenever $\epsilon > 0$ we normalize the cost matrix, i.e., $\tilde{C}_{ij} = C_{ij}/\max_{i,j} C_{ij}$. For InfoOT, we use the authors' source code which does not normalize the cost. As a result, we tune $\epsilon \in \{1, 5, 10\}$. For InfoOT, we tune the kernel hyper-parameter $h \in \{0.1, ..., 1.0\}$. For HOT-DA, we tune the choice of clustering algorithm among K-Means [161] and spectral clustering [162]. For GMM-OTDA, we tune the number of components K.

Discussion. We show an overview of our results in Table 9.2. Based on this table, we can see that our GMM-OTDA approach outperforms its empirical counterpart [23] (i.e., OTDA$_{EMD}$ and OTDA$_{Sink}$) over all benchmarks. This result highlights the advantage of using GMM modeling. At one hand, using the GMM-OT framework forces samples from the same component to be transported together. At the other hand, the transport between source and target components takes the form of an affine map, which is easier to estimate accurately than the barycentric map.

Now, we consider GMM-OTDA in comparison with OTDA$_{affine}$ [67]. Note that this latter method assumes that source and target domain data are Gaussian. This can be seen as a particular case of our algorithm, when $K = 1$. As we highlited in Chapter 7, the Gaussian hypothesis is not adequate for classification problem, as data is likely multi-modal, where each mode represents a class. As a result, using the GMM modeling improves over OTDA$_{affine}$ on all benchmarks, except Office-Home.

Furthermore, we compare GMM-OTDA with HOT [106]. In this case, both methods rely on clustering techniques for performing adaptation. However, HOT employs empirical OT for transporting between pairs of matched clusters. This means that OT estimation is still subject to the curse of dimensionality [45]. A further limitation of HOT is that the number of clusters is fixed to the number of classes. This is a naive assumption, as each class may be itself multi-modal. In this sense, our GMM-OTDA method is flexible with respect the number of clusters or components of each GMM representing each class.

Benchmark	Task	Baseline	OTDA$_{EMD}$	OTDA$_{Sink}$	OTDA$_{affine}$	InfoOT$_b$	InfoOT$_c$	HOT-DA	GMM-OTDA$_{MAP}$	GMM-OTDA$_T$
Caltech-Office	$A \to D$	87.10	77.42	87.10	93.55	93.55	83.87	96.77	93.55	80.65
	$A \to W$	91.53	93.22	96.61	96.61	96.61	94.92	96.61	93.22	91.53
	$A \to C$	88.44	91.56	73.33	91.11	87.11	90.67	74.67	88.44	88.44
	$D \to A$	88.54	90.10	93.23	92.19	91.15	92.71	96.88	96.35	95.83
	$D \to W$	98.31	93.22	93.22	94.92	100.0	91.53	94.92	100.0	98.31
	$D \to C$	75.56	67.11	17.33	71.56	68.44	75.11	51.11	86.67	71.56
	$W \to A$	85.94	82.81	15.62	81.25	60.94	86.46	69.79	87.50	84.90
	$W \to D$	100.0	93.55	93.55	96.77	90.32	93.55	90.32	96.77	93.55
	$W \to C$	84.89	87.56	87.56	88.44	87.56	87.56	84.89	87.56	88.00
	$C \to A$	98.44	96.88	98.44	98.44	98.44	96.88	98.44	98.44	98.44
	$C \to D$	93.55	87.10	90.32	93.55	87.10	87.10	100.0	93.55	80.65
	$C \to W$	91.53	94.92	94.92	94.92	93.22	93.22	98.31	96.61	94.92
	Avg.	90.32	87.95	78.44	91.11	87.87	89.46	87.72	93.22	88.90
ImageCLEF	$B \to C$	92.50	95.00	95.00	94.17	97.50	96.67	96.67	95.00	94.17
	$B \to I$	90.00	89.17	89.17	91.67	94.17	91.67	95.00	95.00	93.33
	$B \to P$	68.33	69.17	70.00	71.67	73.33	75.83	74.17	72.50	74.17
	$C \to B$	65.00	65.83	65.83	65.00	51.67	62.50	62.50	65.00	66.67
	$C \to I$	89.17	96.67	96.67	95.00	92.50	97.50	95.83	94.17	96.67
	$C \to P$	71.67	74.17	73.33	71.67	75.00	75.83	72.50	70.83	75.83
	$I \to B$	68.33	70.00	68.33	70.00	65.00	66.67	61.67	67.50	70.00
	$I \to C$	93.33	95.83	95.83	95.83	95.83	96.67	95.83	95.00	95.83
	$I \to P$	71.67	74.17	75.00	73.33	73.33	71.67	72.50	73.33	75.00
	$P \to B$	67.50	69.17	66.67	68.33	57.50	65.83	62.50	62.50	64.17
	$P \to C$	95.00	95.83	95.83	95.00	96.67	96.67	96.67	95.00	95.00
	$P \to I$	90.83	92.50	92.50	90.83	95.83	95.83	95.00	94.17	95.00
	Avg.	80.28	82.29	82.01	81.88	80.69	82.78	81.74	81.67	82.99
Office 31	$A \to D$	66.07	68.75	69.64	69.64	75.89	76.79	72.32	69.64	72.32
	$A \to W$	76.02	74.27	80.12	80.12	79.53	79.53	73.68	76.61	80.70
	$D \to A$	65.68	65.85	67.77	66.90	67.60	66.20	61.15	68.29	73.52
	$D \to W$	94.15	95.32	98.25	98.25	95.91	97.08	84.80	98.83	95.32
	$W \to A$	63.41	66.90	67.42	65.51	67.60	67.60	61.67	66.38	65.68
	$W \to D$	96.43	90.18	92.86	95.54	87.50	91.96	81.25	91.96	91.07
	Avg.	76.96	76.88	79.34	79.32	79.00	79.86	72.48	78.62	79.77
Office-Home	$Ar \to Cl$	55.10	54.98	54.87	56.24	17.41	53.95	47.88	53.95	57.96
	$Ar \to Pr$	70.95	68.69	71.96	71.96	30.97	70.27	67.23	74.89	74.10
	$Ar \to Rw$	79.68	79.68	80.83	80.71	40.53	80.25	76.00	77.96	82.43
	$Cl \to Ar$	63.51	60.62	63.09	62.68	31.34	62.68	53.81	59.79	64.33
	$Cl \to Pr$	69.26	66.89	68.81	70.72	41.78	68.92	63.51	70.05	71.73
	$Cl \to Rw$	72.68	69.92	71.18	72.33	38.81	71.18	67.97	68.66	74.63
	$Pr \to Ar$	66.80	62.47	64.12	66.39	32.16	64.33	55.88	57.73	62.89
	$Pr \to Cl$	36.88	38.83	25.32	38.83	8.59	30.93	23.71	30.70	31.62
	$Pr \to Rw$	78.76	77.84	79.22	79.22	47.99	78.30	71.64	73.59	80.94
	$Rw \to Ar$	72.99	71.96	72.37	73.81	51.13	70.72	62.47	66.39	69.69
	$Rw \to Cl$	53.15	57.85	57.39	56.93	37.00	56.59	47.65	50.63	56.36
	$Rw \to Pr$	82.32	80.86	81.87	82.21	64.86	81.31	72.30	80.41	82.09
	Avg.	66.84	65.88	65.92	67.67	36.88	65.79	59.17	63.73	67.40
CWRU	$A \to B$	51.12	72.00	75.19	78.12	-	-	69.88	79.75	79.75
	$A \to C$	62.88	94.12	100.00	95.62	-	-	100.00	99.94	100.00
	$B \to A$	42.50	76.12	78.50	75.88	-	-	79.75	80.00	80.00
	$B \to C$	37.44	77.62	78.88	75.38	-	-	79.81	79.56	79.94
	$C \to A$	52.81	98.38	99.25	94.12	-	-	98.75	99.12	99.88
	$C \to B$	55.62	70.25	74.50	75.50	-	-	83.12	79.75	80.00
	Avg.	50.40	81.42	84.39	82.44	-	-	85.22	86.35	86.59

Table 9.2 – **Single-source domain adaptation results.** We compare 8 methods over 5 benchmarks, with a total of 48 adaptation tasks. We do not report the performance of InfoOT over CWRU, because this method did not converge.

9.2 Multi-Source Domain Adaptation

9.2.1 Benchmarking Results

In this section, we explore our benchmarking results. Here, we focus on OT-based MSDA algorithms. In total, we compare 9 methods: Weighted JDOT (WJDOT) [113], Joint Class Proportion and Optimal Transport (JCPOT) [24] (described in Chapter 4, Section 4.4), WBT [12, 13] (covered in Chapter 5), WBR-E and R, DaDiL-E and R [14] (detailed in Chapter 6), GMM-WBT and GMM-DaDiL [15] (described in Chapter 7). Before proceeding, we give a high-level description of these algorithms,

- WJDOT weights the source domains through an importance vector $\lambda = (\lambda_1, \cdots, \lambda_N)$. This generates a combined source measure $Q_S^{(\lambda)}$ which is matched with the proxy target measure $Q_T^{(h)}$.
- JCPOT estimates the class proportions of the target domain measure via the Wasserstein barycenter of source domain class proportions.
- WBT generates labeled data in the target domain, by first computing a free-support Wasserstein barycenter of source domain measures, then transporting it to the target, via barycentric mapping.
- WBR estimates the barycentric coordinates of the measure in the Wasserstein hull of source domain measures, $\mathcal{M}_{\mathcal{W}_2}(\mathcal{Q}_S)$, that best approximates the target domain measure, Q_T.
- DaDiL learns a set of atoms and barycentric coordinates, so that each measure in MSDA is expressed as a Wasserstein barycenter of atom measures.

The GMM versions of WBT and DaDiL corresponds to substituting empirical measures by GMMs. Furthermore, on one hand, suffix E refer to *ensembling methods*. For WBR, this means that a classifier is learned on each source domain. For DaDiL, this means that a classifier is learned on each atom. Then, the predictions of these classifiers are weighted via their barycentric coordinates with respect these measures (e.g., $\lambda_T = (\lambda_{T,1}, \cdots, \lambda_{T,N})$ for WBR). On the other hand, suffix R refers to *reconstruction methods*, i.e., methods that reconstruct the target measure through a Wasserstein barycenter. In the case of WBR, one uses $\mathcal{B}(\lambda_T, \mathcal{Q}_S)$. For DaDiL, one uses $\mathcal{B}(\lambda_T, \mathcal{P}^\star)$.

These methods have a few hyper-parameters that must be tuned. For WJDOT, we tune the entropic regularization ϵ (see Chapter 2, section 2.2), as well as the label distance importance β (see Chapter 4, section 4.4). For JCPOT, the only hyper-parameter that needs tuning is the entropic regularization ϵ, which should be positive, as this method relies on the IBP algorithm (see Chapter 3, section 3.3.1). For WBT and WBR, we tune ϵ and the number of samples n in the barycenter support. For DaDiL, we tune the number of samples in the atoms' support n, the number of atoms C, the batch size n_b. For GMM-WBT, we tune the number of components in the GMMs K, and the entropic regularization ϵ. For GMM-DaDiL, we tune C and the number of components in the GMMs, K.

Remark 10. *For DaDiL and GMM-DaDiL, we do not use entropic regularization. As we explore in section 9.3, we did not verified any gain in using the Sinkhorn algorithm.*

Evaluation Protocol. For MSDA, we adopt the leave-one-domain-out evaluation strategy. In a nutshell, the idea is to treat, one by one, each available domain as the target. Each other domain is

then considered as an individual source. For instance, in Office 31, one has (A, D, W), hence, there are 3 adaptation tasks. The first adaptation task corresponds to $(D, W) \rightarrow A$, where D and W are considered are the source domains, and A is the target. Note that, for each benchmark, this generates N metrics for each algorithm. We treat source domain data as training data. This means that all available source data is seen during training (i.e., no validation nor test partitions for these domains). In the spirit of having an unbiased evaluation, we consider training and test target data. The training target data is seen during training by DA algorithms, whereas the test target data is only used for evaluation. We rank algorithms by the *average adaptation performance*, i.e., the average accuracy (see equation 9.1) over all possible target domains. An overview of our results is shown in Table 9.3.

Algorithm	B	C	I	P	Avg. ↑
Source-Only	63.33	96.66	93.33	78.34	82.91
WJDOT	66.67	95.83	94.17	75.83	83.12
JCPOT	68.34	95.83	95.00	78.34	84.37
WBT	63.33	**99.16**	96.66	**81.66**	85.21
WBR-E	68.33	96.66	95.83	75.00	84.16
WBR-R	66.66	94.16	95.83	74.16	82.71
DaDiL-E	69.16	96.66	95.00	75.83	84.37
DaDiL-R	**70.83**	96.66	**97.50**	76.66	**85.41**
GMM-WBT	67.50	96.66	96.66	77.50	84.58
GMM-DaDiL	**70.83**	97.50	96.66	76.66	**85.41**

(a) ImageCLEF.

Algorithm	A	D	W	Avg. ↑
Source-Only	67.50	95.00	96.83	86.40
WJDOT	68.29	99.11	95.91	87.77
JCPOT	55.75	**100.00**	98.24	84.66
WBT	67.94	98.21	97.66	87.93
WBR-E	66.37	**100.00**	97.66	88.01
WBR-R	65.15	**100.00**	98.24	87.80
DaDiL-E	71.60	**100.00**	98.83	90.14
DaDiL-R	71.42	**100.00**	98.83	90.08
GMM-WBT	70.13	99.11	96.49	88.54
GMM-DaDiL	**72.48**	**100.00**	**99.41**	**90.63**

(b) Office 31.

Algorithm	Ar	Cl	Pr	Rw	Avg. ↑
Source-Only	72.90	62.20	83.70	85.00	75.95
WJDOT	74.28	63.80	83.78	84.52	76.59
JCPOT	74.28	63.68	84.90	83.48	76.58
WBT	75.72	63.80	84.23	84.63	77.09
WBR-E	76.95	61.39	82.99	84.86	76.54
WBR-R	72.22	60.48	81.19	81.53	73.85
DaDiL-E	**77.16**	64.95	85.47	84.97	78.14
DaDiL-R	75.92	64.83	85.36	**85.32**	77.86
GMM-WBT	75.31	64.26	**86.71**	85.21	77.87
GMM-DaDiL	**77.16**	**66.21**	86.15	**85.32**	**78.81**

(c) Office-Home.

Algorithm	A	B	C	Avg. ↑
MLP	70.90 ± 0.40	79.76 ± 0.11	72.26 ± 0.23	74.31
WJDOT	99.96 ± 0.02	98.86 ± 0.55	**100.0 ± 0.00**	99.60
JCPOT	93.44 ± 0.51	86.81 ± 0.39	97.31 ± 0.28	92.52
WBT	100.00 ± 0.00	99.85 ± 0.05	100.00 ± 0.00	99.95
WBR-E	71.47 ± 0.86	79.73 ± 0.15	73.50 ± 0.87	74.90
WBR-R	72.66 ± 3.06	77.48 ± 1.77	76.65 ± 4.14	75.60
DaDiL-E	93.71 ± 6.50	83.63 ± 4.98	99.97 ± 0.05	92.33
DaDiL-R	99.86 ± 0.21	99.85 ± 0.08	**100.00 ± 0.00**	99.90
GMM-WBT	**100.00 ± 0.00**	**99.95 ± 0.07**	**100.00 ± 0.00**	**99.98**
GMM-DaDiL	**100.00 ± 0.00**	**99.95 ± 0.04**	**100.00 ± 0.00**	**99.98**

(d) CWRU.

Table 9.3 – **Multi-source domain adaptation results.** For image benchmarks, evaluation is done on the fixed partition of train and test samples for the target domain. For CWRU, we show the average performance ± 2σ over 5−folds.

Model Selection is an open problem in DA [163, 164]. The challenge comes from, theoretically, not having labels on the target domain. However, with the leave-one-domain-out strategy, one indeed has access to labeled target domain data, even though algorithms *are not allowed to use the labels* for training. In principle, these labeled samples cannot be used for selecting hyper-parameters. Practicioners are then faced with a few alternatives,

1. Using unsupervised proxy metrics. In this case, one defines a metric that is a good indicative if a set of hyper-parameters should be chosen.

2. Selecting the model that performs the best. This is problematic, as validation is done with test data.
3. Validate on a fixed domain, then use these parameters for all domains.

The first approach is a good candidate for a solution, but using a unsupervised metric, might not reflect the underlying quantity we want to maximize, that is, generalization. In the case of DaDiL, an example of proxy is the dictionary learning loss at optimality, but one may fit the positions in space (i.e., the $\mathbf{x}_i^{(B)}$) with wrong labels. For GMM-DaDiL, another candidate is the Negative Log-Likelihood (NLL) of $\mathcal{B}(\lambda_T^\star, \mathcal{P}^\star)$, but it might suffer from the same problem. As we show in section 9.3, neither of these choices is guaranteed to lead to the best possible model (see Figures 9.8 and 9.11). In the following experiments, we choose models that maximize the performance (i.e., classification accuracy) on the training target domain data for *all models* (including the competitors). As a result, we use an impractical but fair metric for model selection. We highlight that, while we maximize the performance on *training* target domain data, models are ranked with respect the *test* target domain data.

Discussion. We start our discussion by comparing different domain weighting strategies. For instance, WJDOT weights source domains linearly, by assigning importances to their samples. In comparison, WBT weights source domains in a Wasserstein space, that is, it computes a Wasserstein barycenter of source domains. Over all tested benchmarks, WBT improves over WJDOT, which implies that combining sources through Wasserstein barycenters is advantageous. Furthermore, note that we also improve over JCPOT, which uses fixed-support Wasserstein barycenters for estimating label proportions in the target domain.

Now, we focus on DaDiL. We start by comparing DaDiL with WBR. Note that, as we highlighted in Chapter 6, WBR employs the same principle of DaDiL, but with the source domain as atoms. Indeed, WBR finds the barycentric coordinates of the target domain with respect the Wasserstein hull of source domains, $\mathcal{M}_{\mathcal{W}_2}(\mathcal{Q}_S)$. Over all benchmarks, DaDiL outperforms WBR-E and R. We further compare these approaches in section 9.2.2. Furthermore, DaDiL generally improves over WBT, while being a one step MSDA method. This means that DaDiL can better cope with distributional shift when reconstructing the target domain.

Finally, we compare our GMM methods with their empirical counterparts. In particular, GMM-DaDiL improves over DaDiL in all benchmarks. This highlights the advantage of the GMM modeling. At one hand, we have a parametric model for the measures in MSDA, which leads to a simpler optimization problem, without the need for mini-batching. At the other hand, assuming that data follows a GMM leads to better statistical estimation. A similar analysis can be done for GMM-WBT, but in some benchmarks empirical WBT outperforms it (e.g., ImageCLEF).

9.2.2 Exploring the Interpolatoin Space

In this experiment, we revisit examples 17 and 20, now using the Office Home benchmark. This benchmark is convenient, as there are 3 source domain measures and a target. As a result, we can easily visualize the barycentric hull through the $3-$simplex Δ_3.

In our experiment, we explore the interpolation space generated by the source domains, with that generated by DaDiL. We further extend this comparison, to comprise the GMM versions of these algorithms. Recalling definition 17, for a family of measures $\mathcal{P} \subset \mathbb{P}(\mathcal{X})$ and a metric $\mathcal{D}$ on $\mathbb{P}(\mathcal{X})$, one has the so-called barycentric hull $\mathcal{M}_{\mathcal{D}}(\mathcal{P})$ corresponding to all measures expressed as $\mathcal{B}(\lambda, \mathcal{P})$, for the barycentric coordinates $\lambda \in \Delta_C$. In this case, the vector λ serves as a *coordinate system* for the manifold $\mathcal{M}_{\mathcal{D}}(\mathcal{P})$. Hence, we consider 6 cases,

— **WBR.** Corresponds to the Wasserstein hull of source domain measures, i.e., $\mathcal{M}_{\mathcal{W}_2}(\mathcal{Q}_S)$.

— **GMM-WBR.** Corresponds to the mixture-Wasserstein hull of source domain measures, i.e., $\mathcal{M}_{\mathcal{MW}_2}(\mathcal{Q}_S)$.

— **WBT.** Corresponds to the *transported* Wasserstein hull of $\mathcal{Q}_S$, that is,

$$\mathcal{TM}_{\mathcal{W}_2}(\mathcal{Q}_S) = \{(T_\gamma)_\sharp \mathcal{B}_{\mathcal{W}_2}(\alpha, \mathcal{Q}_S) : \alpha \in \Delta_K, \gamma = OT(B, Q_T)\},$$

where T_γ is the barycentric mapping between $\mathcal{B}(\alpha, \mathcal{Q}_S)$ and Q_T.

— **GMM-WBT.** Corresponds to the *transported* mixture-Wasserstein hull of source domain measures,

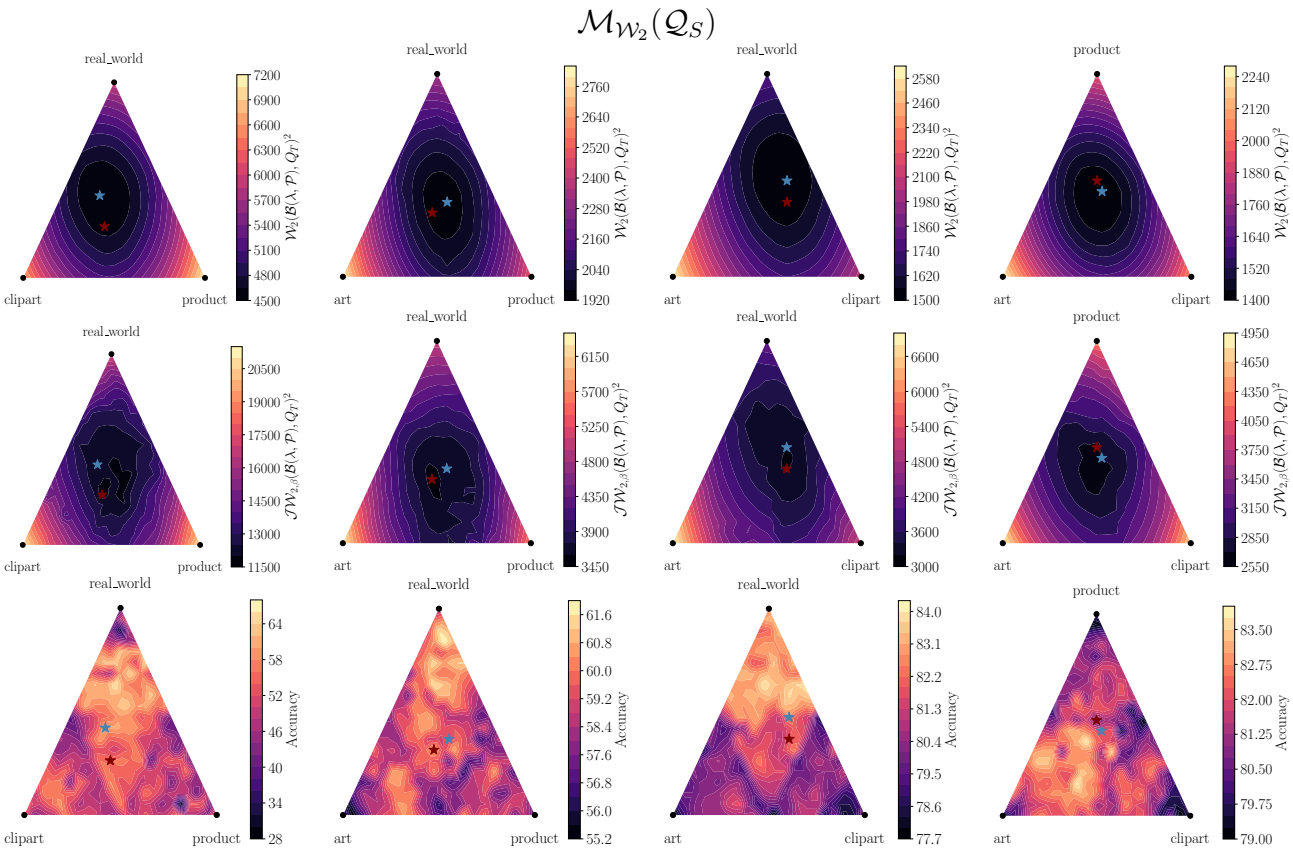
$$\mathcal{TM}_{\mathcal{MW}_2}(\mathcal{Q}_S) = \{(T_\omega)_\sharp \mathcal{B}_{\mathcal{MW}_2}(\alpha, \mathcal{Q}_S) : \alpha \in \Delta_K, \omega = GMMOT(B, Q_T)\},$$

— **DaDiL.** Corresponds to the Wasserstein hull of atoms, i.e., $\mathcal{M}_{\mathcal{W}_2}(\mathcal{P}^\star)$.

— **GMM-DaDiL.** Corresponds to the mixture-Wasserstein hull of atoms, i.e., $\mathcal{M}_{\mathcal{MW}_2}(\mathcal{P}^\star)$.

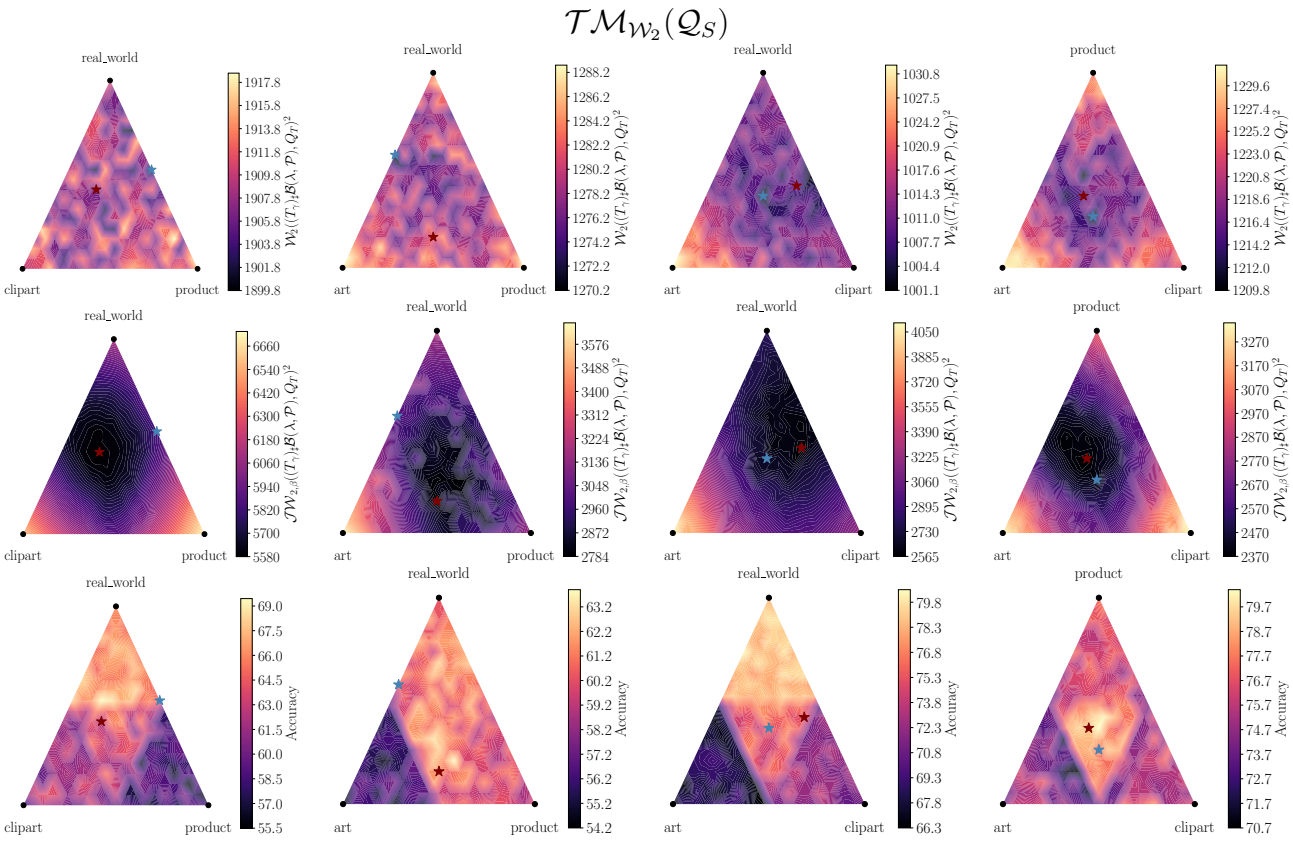
We show our results and discussion in Figures 9.1 through 9.6. Here, we provide some highlights of our results. First point is that DaDiL manages to better reconstruct the target domain, that is, it finds $\mathcal{B}(\lambda^\star, \mathcal{P}^\star)$ that is closer to $\hat{Q}_T$ with respect the Wasserstein distance $\mathcal{W}_2$. This result was already hinted in examples 17 and 20. From the perspective of GMM methods, we remark that interpolations of source domain GMMs fail to generalize to the target domain. Transporting the barycentric GMM to the target domain does improve performance, but it remains largely below its empirical counterpart. GMM-DaDiL solves these issues, achieving state-of-the-art performance. Furthermore, with respect the NLL, interpolating atoms generates measures with lower NLL, and the NLL has quadratic level curves with respect λ. This finding is surprising, since GMM-DaDiL is not explicitly trained to minimize the NLL.

The main takeaway of our experiments is the following. The relantionship between probability metrics and DA performance is complex. Indeed, if a measure is close to each other, one *can have* better generalization, but this is not guaranteed. This result is well known in the DA literature [21, 165]. In some cases, due the missing information (target domain labels), DA is a hopeless problem.



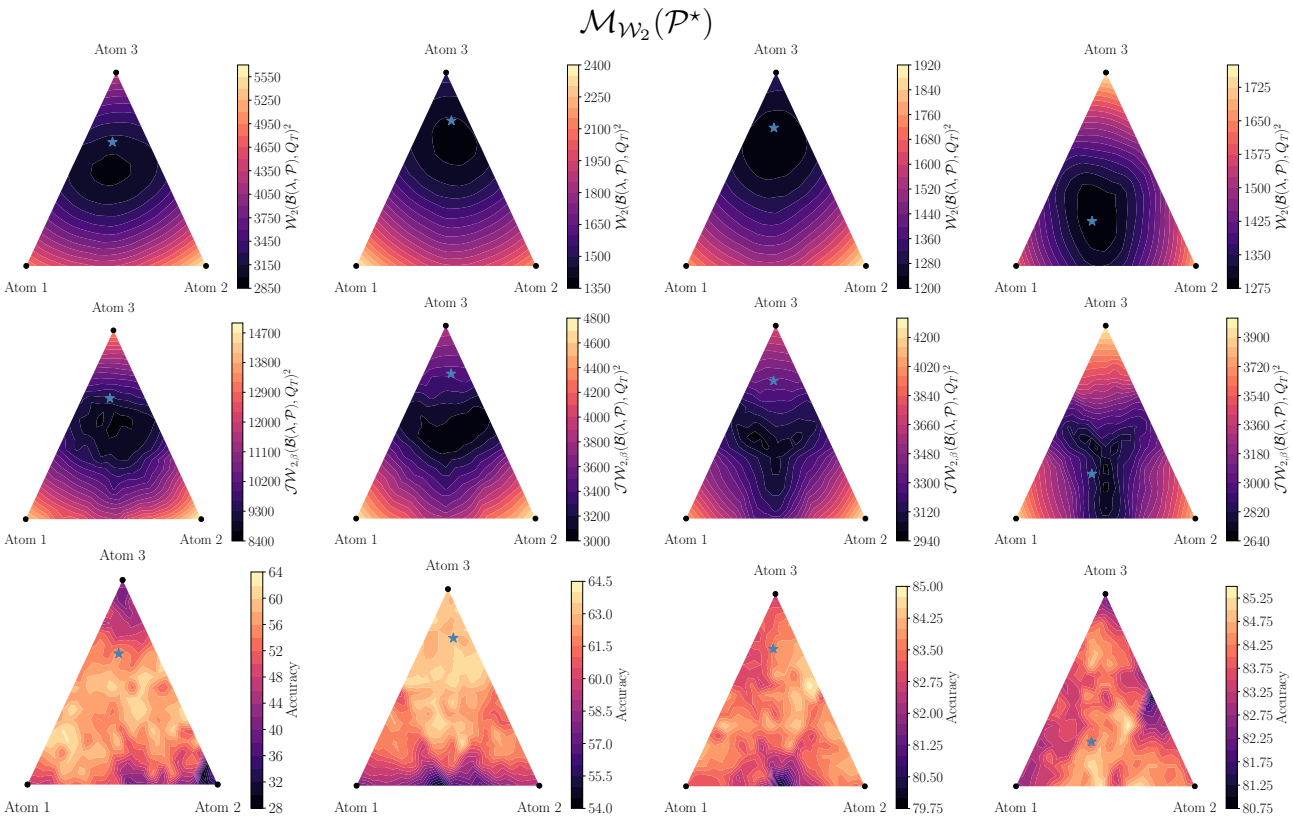
(a) Art (b) Clipart (c) Product (d) Real World

Figure 9.1 – **Wasserstein hull of source domain measures.** Points inside the triangle represent barycentric coordinate vectors $\lambda \in \Delta_3$. On each figure, we mark the minimizers $\lambda^\star_{\mathcal{W}}$ and $\lambda^\star_{\mathcal{JW}}$ by a blue, and a red star in the simplex. Overall, these minimizers are close together, however, taking the labels of measures into account changes the landscape of the Wasserstein hull. While the level curves of $\lambda \mapsto \mathcal{W}_2(\mathcal{B}(\lambda, \mathcal{Q}_S), \hat{Q}_T)$ are approximately quadratic, those for the $\mathcal{JW}_{2,\beta}$ are not. Finally, the level curves of the classification accuracy over the target domain (bottom row) of a 1-nearest neighbor classifier fit with $\mathcal{B}(\lambda, \mathcal{Q}_S)$ are actually much more irregular than both losses. This result is simillar to our remarks in examples 17 and 20, for the Caltech-Office benchmark.



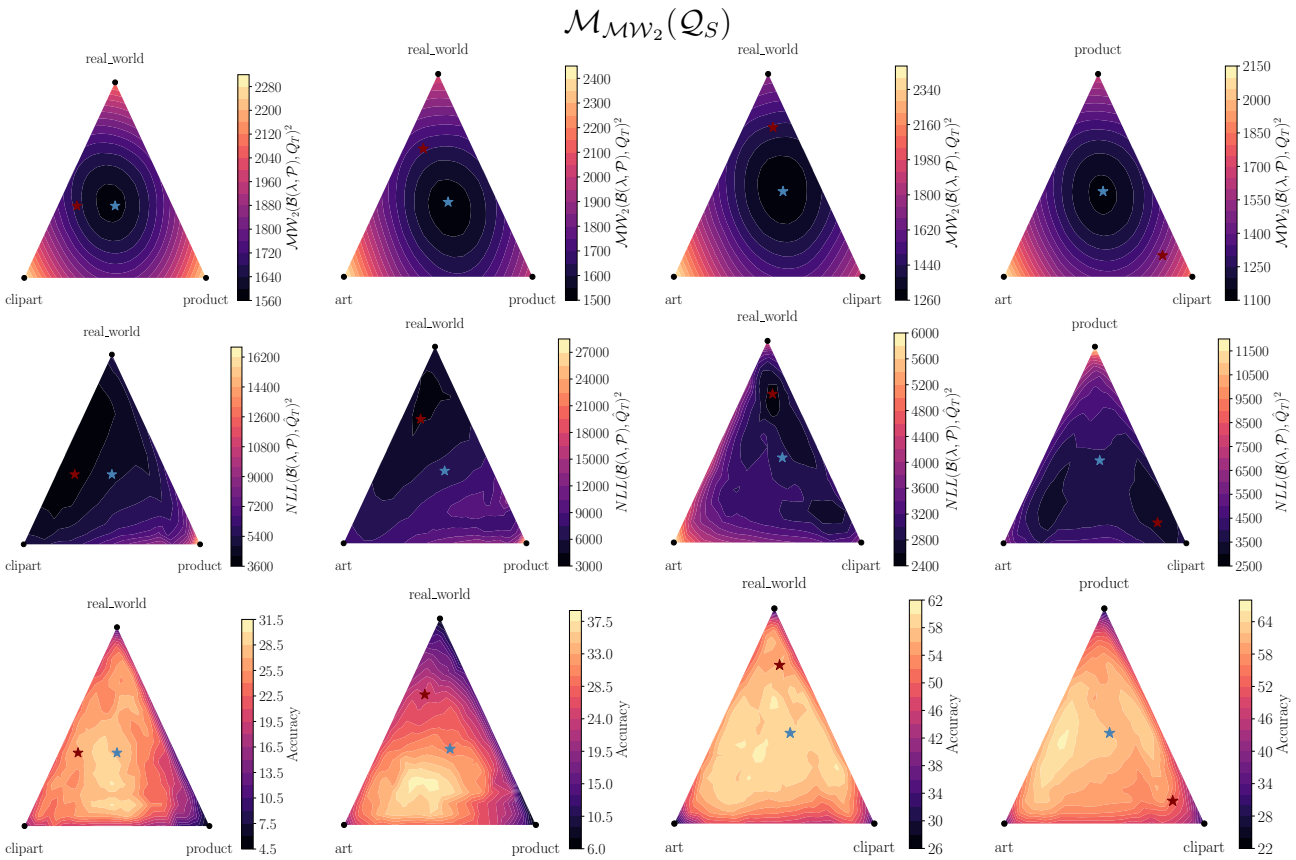
(a) Art (b) Clipart (c) Product (d) Real World

Figure 9.2 – **Transported Wasserstein hull of source domains.** In this case, for each $\lambda \in \Delta_3$, we transport the barycenter $\mathcal{B}(\lambda, \mathcal{Q}_S)$ towards the target domain $\hat{Q}_T$ using the barycentric mapping T_γ. As we can observe by the first and second rows, this decreases both the Wasserstein distance $\mathcal{W}_2$ and joint Wasserstein distance $\mathcal{JW}_{2,\beta}$ (compare the ranges of the colorbar with the previous image). As a result, the first row essentially display the randomness associated with the calculation of the Wasserstein barycenter (i.e., initialization in algorithm 4). Meanwhile, there is still some residual $\mathcal{JW}$ as shown in the level curves on the second row. This is a result of the fact that the target domain labels are not taken into account when calculating T_γ. In either case, note that performing an additional transportation step towards the target generally improves classification performance, as seen in the third row. This is in line with previous theoretical results in DA.



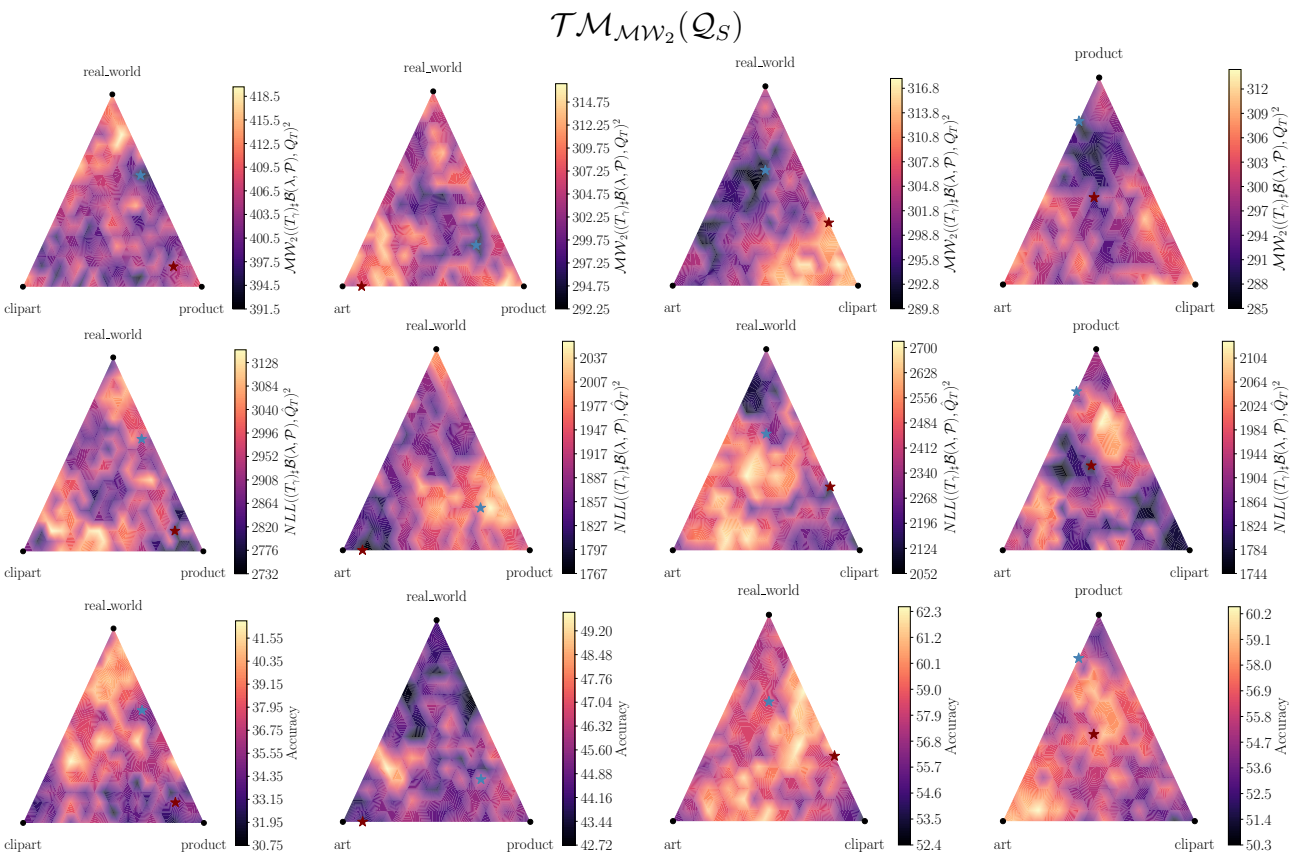
(a) Art (b) Clipart (c) Product (d) Real World

Figure 9.3 – **Wasserstein hull of atoms learned by DaDiL.** In this case, note that the level curves are much more simillar with the Wasserstein hull of sources, i.e., the level curves of $\mathcal{W}_2$ are quadratic. In comparison with the Wasserstein hull of sources, the hull of atoms achieves are closer to the target (compare the ranges in the first row with that of Figure 9.1). The situation is simillar for the $\mathcal{JW}_{2,\beta}$. This suggests that the interpolation space learned by DaDiL better expresses the target domain. Note, however, that transportting the barycenters towards the target usually achieves closer measures. Nonetheless, DaDiL manages to generate an interpolation space that achieves better adaptation performance (e.g., performance on product and real world domains).



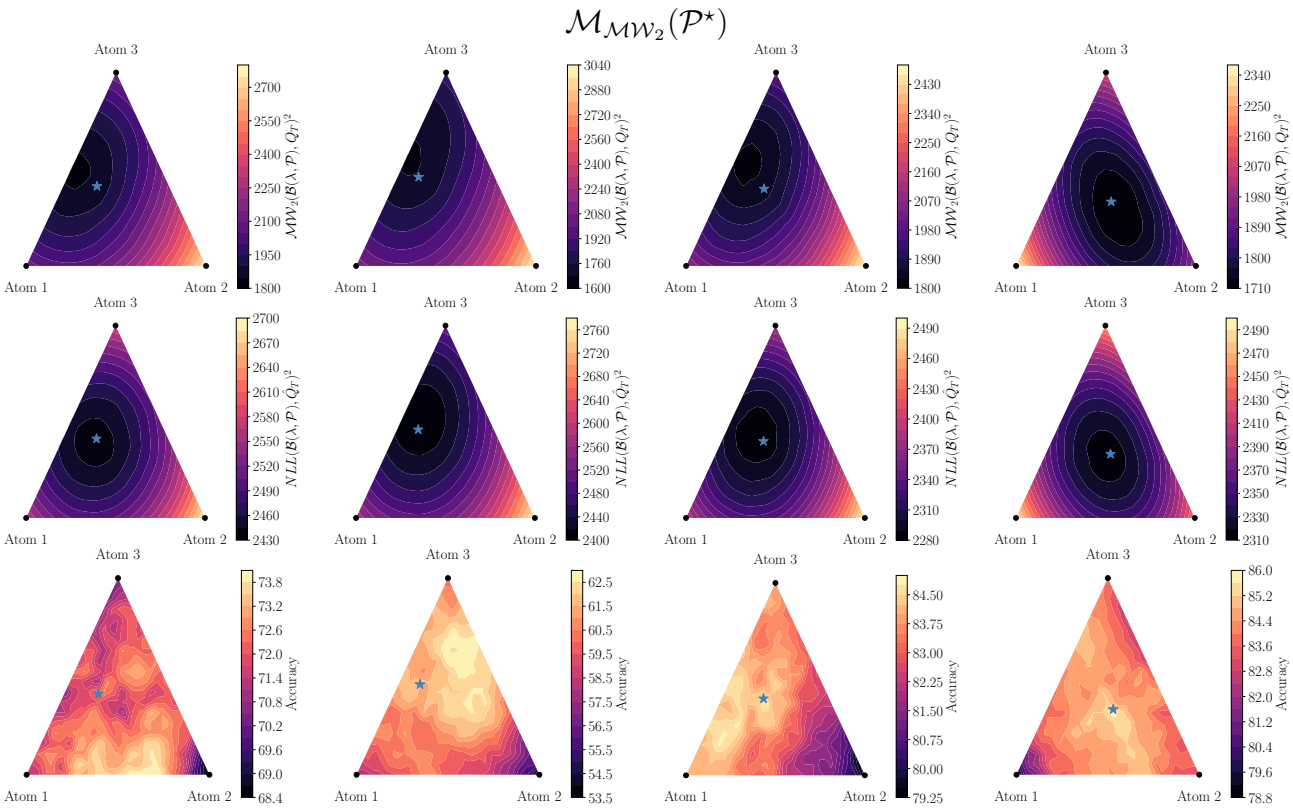
(a) Art (b) Clipart (c) Product (d) Real World

Figure 9.4 – **Mixture-Wasserstein hull of source GMMs**. In contrast with empirical methods, here source domains are represented through GMMs. As a result, the first and second rows measure different quantities from previous figures, namely, the $\mathcal{MW}_2^2$ and the negative log-likelihood of $\mathcal{B}(\lambda, \mathcal{Q}_S)$ on target domain data. The third row measures the accuracy of the MAP classifier (c.f., equation 7.1) obtained by the GMM $\mathcal{B}(\lambda, \mathcal{Q}_S)$. The situation for DA dramatically changes, as interpolations of source domain GMMs cannot generalize to the target domain. Furthermore, the NLL is highly irregular, with its minimizers being generally far from the $\mathcal{MW}_2$. This latter metric is quadratic with respect $\lambda \in \Delta_3$.



(a) Art (b) Clipart (c) Product (d) Real World

Figure 9.5 – **Transported Mixture-Wasserstein hull of source GMMs**. In addition to computing $\mathcal{B}_{\mathcal{MW}_2}(\lambda, \mathcal{Q}_S)$, we further transport the parameters of the barycentric GMM to match those of the target domain. As a result, we decrease both the mixture-Wasserstein distance and the NLL with respect the target domain (c.f., first and second rows), which exhibit a simillar noise pattern to what was verified in the empirical case. Furthermore, performance generally increases, however, it remains sub-optimal with respect the empirical methods (c.f., third row). Overall, we conclude that source domain GMMs generalize poorly to the target domain.



(a) Art (b) Clipart (c) Product (d) Real World

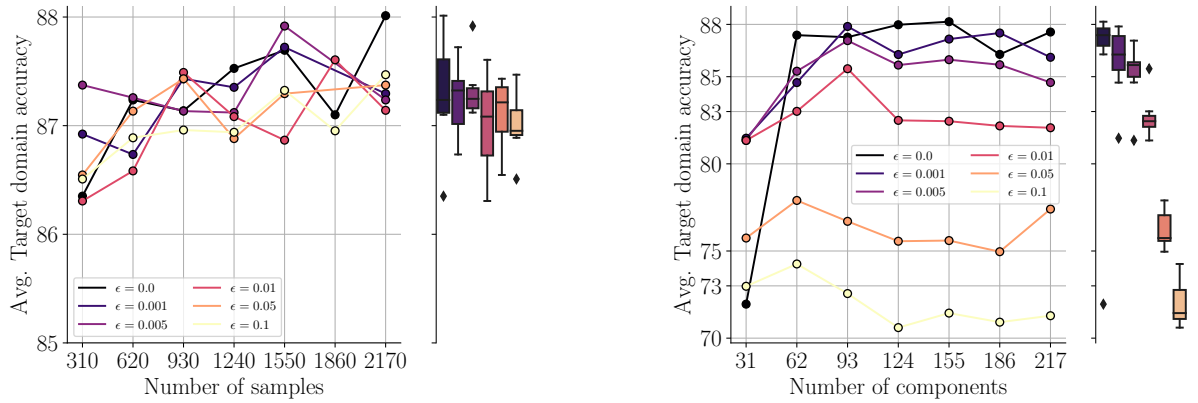
Figure 9.6 – **Mixture-Wasserstein hull of atoms.** By running GMM-DaDiL, we achieve a series of impressing results. First, unlike the empirical case, the barycenters in $\mathcal{M}_{\mathcal{MW}_2}(\mathcal{P}^\star)$ do not achieve, necessarily, a smaller mixture-Wasserstein distance with respect the target. Second, the level curves of the NLL with respect λ are actually quadratic, which is very different from the patterns in Figure 9.1. On the same note, barycenters of atoms generally achieve a lower NLL (c.f., second row). This is surprising, as GMM-DaDiL is not explicitly trained to minimize the NLL. From the perspective of classification (c.f., third row), GMM-DaDiL achieves state-of-the-art performance, largely surpassing the performance of empirical methods.

9.3 Hyper-parameter Sensitivity

In this section, we investigate the impact of hyper-parameters over WBT, GMM-WBT, DaDiL and GMM-DaDiL. Our following results are illustrated over the Office 31 benchmark. We use the adaptation performance, averaged over the 3 adaptation tasks. In the following, we tune 2 hyper-parameters for WBT: number of samples n, and the entropic penalty ϵ. For DaDiL, on top of these parameters, we have the number of atoms C, and the batch size n_b. For GMM-WBT and GMM-DaDiL, we have the number of components in the GMMs, K, which is somewhat equivalent to the number of samples. An advantage of GMM-DaDiL is not using mini-batches, as a result, we do not need to tune this hyper-parameter.

9.3.1 Wasserstein Barycenter Transport

In this section, we evaluate the performance of WBT and GMM-WBT with respect the number of samples n, components K and entropic regularization ϵ. Concerning the range of hyper-parameters, we evaluate $n \in \{310, 620, \cdots, 2170\}$, $K \in \{31, 62, \cdots, 217\}$ and $\epsilon \in \{0.0, 0.001, 0.005, 0.01, 0.05, 0.1\}$. Our results are shown in Figure 9.7

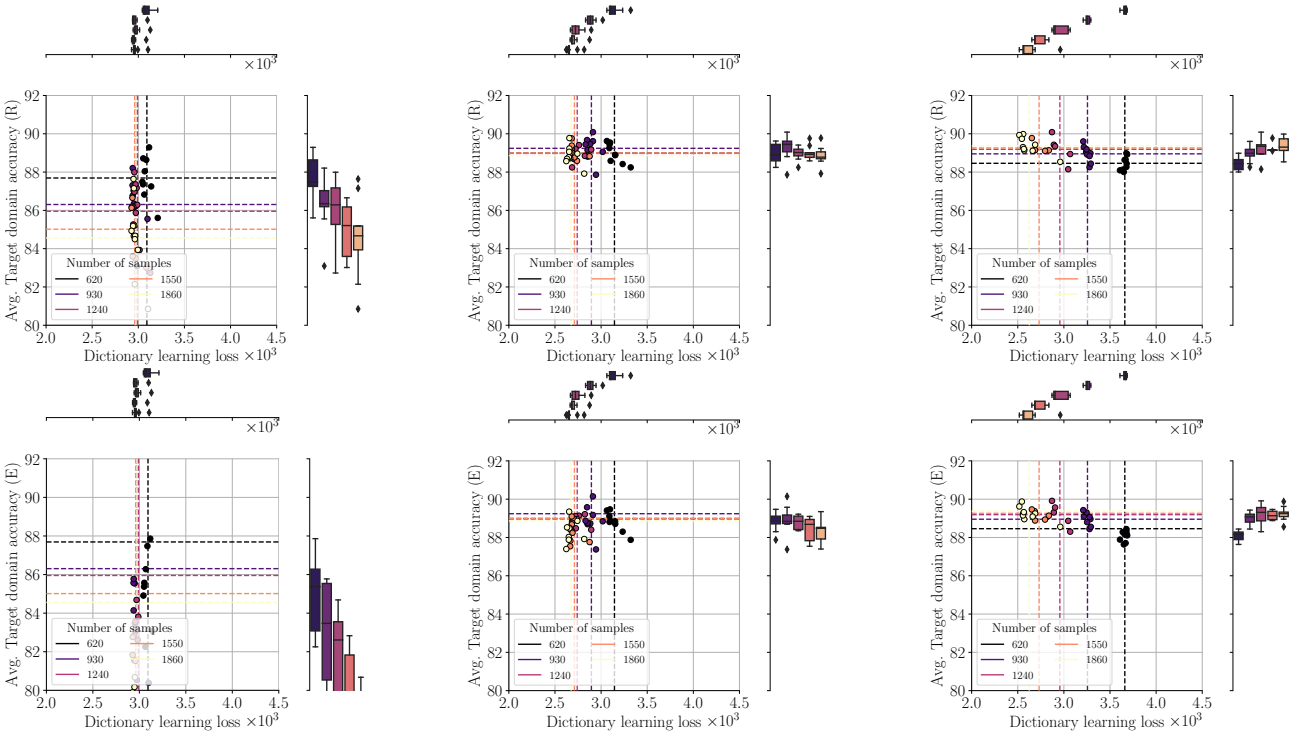


(a) WBT. (b) GMM-WBT.

Figure 9.7 – **WBT and GMM-WBT performance with respect number of samples, number of components and entropic regularization.** Overall, we find that it is better to use $\epsilon = 0$ on both methods. For WBT, performance tends to improve with more samples. For GMM-WBT, performance remains stable over $K > 31$.

For WBT, increasing the number of samples tends to improve performance. This is expected, as growing n reduces the approximation error of the empirical risk and the Wasserstein distance (c.f., $\mathcal{C}_{ERM}$ and $\mathcal{C}_{OT}$ in equation 4.10). The situation is similar for GMM-WBT. For instance, using $K = 31$ assumes that each class is represented by a single Gaussian measure. As a result, growing K leads to a smaller approximation error of the underlying measure. Furthermore, we verified that using entropic regularization (i.e., $\epsilon > 0$) degrades performance in both cases.

9.3.2 Dataset Dictionary Learning



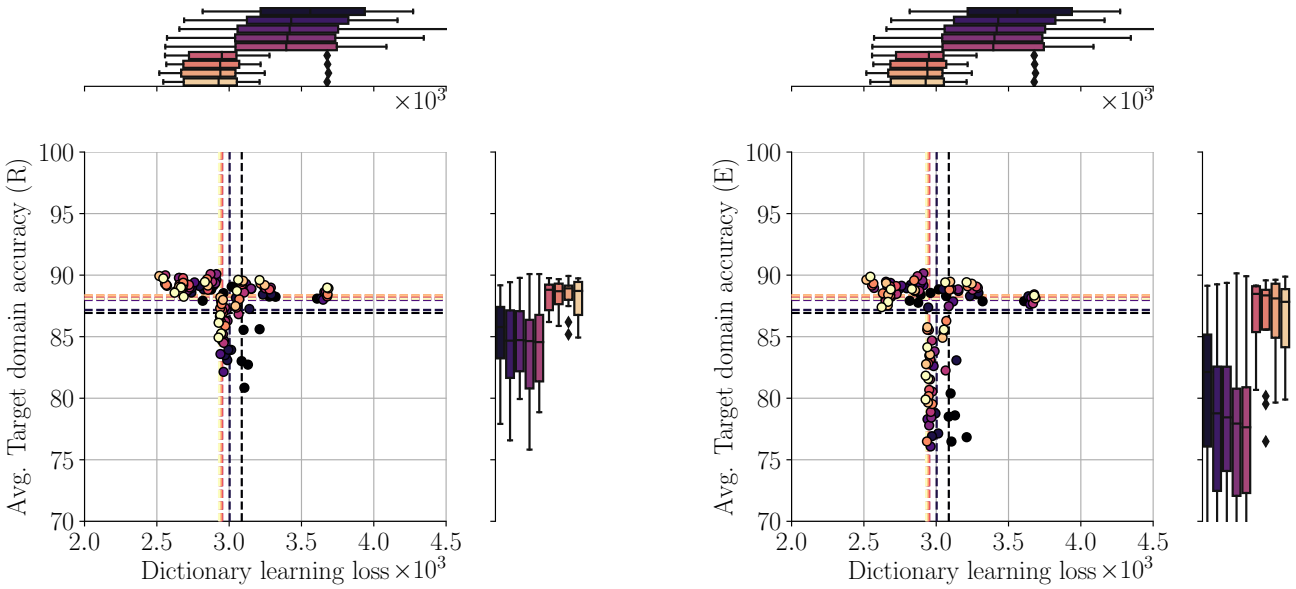
(a) $n_b = 62$ (b) $n_b = 155$ (c) $n_b = 310$

Figure 9.8 – **Analysis of DaDiL performance with respect n and n_b.** Dashed lines mark the average dictionary learning loss and DA performance over a specific group of experiments (e.g., $n = 1860$). Overall, using smaller batch sizes increases the variance with respect DaDiL-R and E performance. Using larger batch sizes with larger number of samples usually leads to more stable performance.

Number of samples n and batch size n_b. We show an analysis of DaDiL's performance with respect to the batch size n_b, and number of samples n in Figure 9.8. On the one hand, the batch size has a heavy effect on the success of DA. Take, for instance, $n_b = 62$, for which the average target domain accuracy of DaDiL-R and E are both under 88%. In comparison, for $n_b = 310$, the same metric is above 88%. This illustrate a potential issue with using mini-batches with unregularized OT. Indeed, as we covered in Chapter 2, section 2.2, computing OT between mini-batches introduces artifacts in the transportation process (c.f., Figure 2.9). Due to mass conservation, mini-batch OT is forced to match samples that would not have been match when given access to all samples from the probability measures [136, 61]. As a result, training DaDiL with small batch sizes leads to poor DA performance.

On the other hand, dictionary learning loss is robust to the use of mini-batches. For instance, for $n_b = 62$, most dictionary solutions concentrate over a loss value of 3.0×10^3. In contrast, for $n_b = 310$, some hyper-parameter choices yield higher losses (e.g., $n = 620$, $n_b = 310$, for which the loss is approximately 3.6×10^5). Overall, we found that the best results with respect dictionary learning

loss and adaptation performance were obtained for higher values of n and n_b. In this context, it is helpful to set it as a multiple of the number of classes, i.e., $n_b = spc \times n_c$, so that one should expect spc samples from each class in each mini-batch. Performing stratified mini-batch sampling may further help with this issue. A further guideline is to set n as a multiple of n_b, i.e., $n = M \times n_b$. For instance, as we remarked, if n is close to n_b, the mini-batch DaDiL optimization problem resembles the full-batch problem.

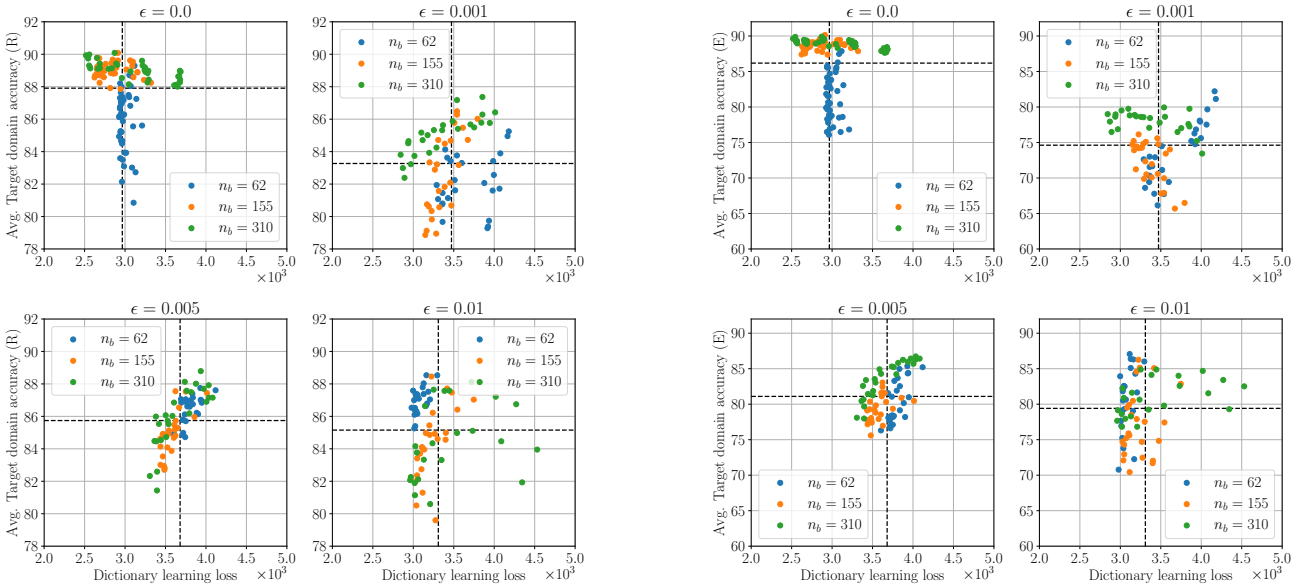


(a) DaDiL-R. (b) DaDiL-E

Figure 9.9 – **Analysis of DaDiL performance with respect C.** As in the previous figure, dashed lines mark the average dictionary learning loss and target performance for a group of experiments (e.g., $C = 2$). Overall, on one hand DaDiL-R is robust with respect the choice of number of atoms. Nonetheless, one should be mindful as for increasing values of C, the DaDiL optimization problem becomes large. DaDiL-E, on the other hand, has a more variable performance.

Number of atoms C. We show an overview of our results in Figure 9.9, which is divided into DaDiL-R (a) and DaDiL-E (b). In both cases, performance is approximately stable with respect to number of components. However, for both methods we verify an increase in adaptation performance for $C \geq 7$, as well as a decrease in dictionary learning loss. This finding is rather surprising, as we are actually using much more atoms than domains. Especially, in this benchmark, the number of domains used in dictionary learning is 3 (2 sources, and one target).

Entropic regularization. Our final analysis concerns the use of entropic regularization in dictionary learning. We show our results in Figure 9.10. Here, we note a few things. First, using unregularized OT outperforms entropic OT. For instance, when $\epsilon = 0$, the average adaptation performance is well above the value of $\epsilon > 0$. However, we do verify that using entropic OT leads to some robustness with respect to batch size (e.g., DaDiL-R performance for $\epsilon = 0.005$).



(a) DaDiL-R. (b) DaDiL-E.

Figure 9.10 – **Analysis of DaDiL with respect entropic regularization ϵ.** Dashed lines indicate the average target domain accuracy and dictionary learning loss over all experiments on each plot. In general, DaDiL performs better with unregularized OT.

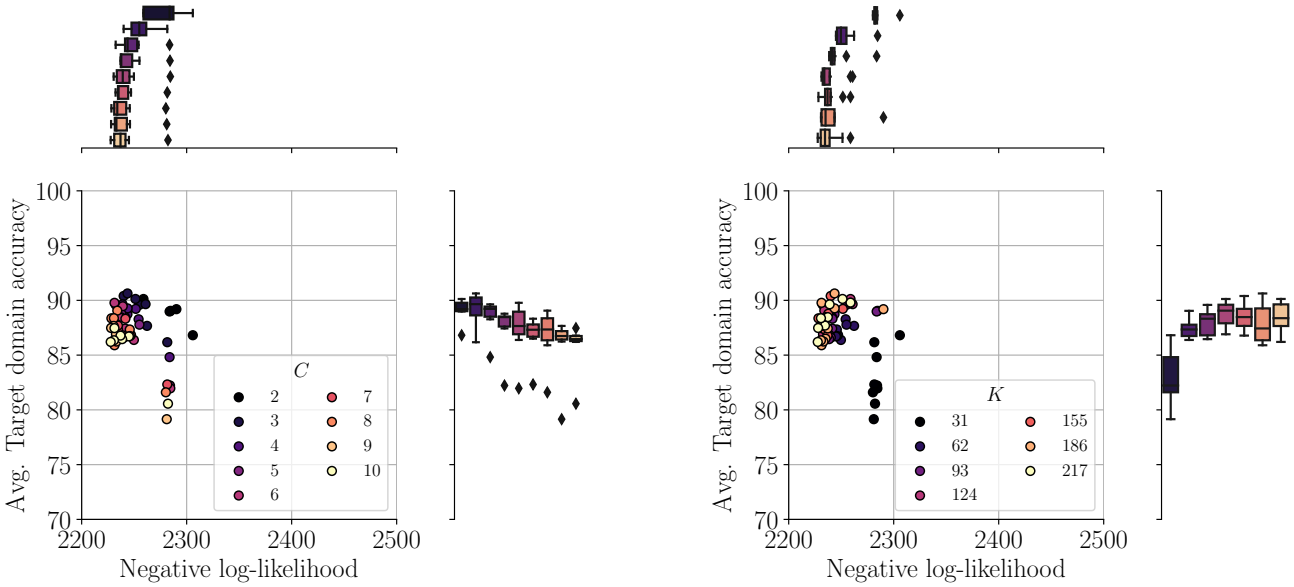
9.3.3 Gaussian Mixture Dataset Dictionary Learning

Now, moving to GMM-DaDiL, we have two hyper-parameters of interest. As before, we have the number of atoms C, however, since the atoms are now GMMs, we have the number of components K. One can see K as analogous to the number of samples in DaDiL. Here, a major advantage is that, since there is no mini-batching in the optimization problem, we do not have to tune n_b. As previously established (see Figure 9.8), this is a major source of variability in DaDiL.

We show an overview of our results in Figure 9.11, where we condition the relantionship between the NLL and DA performance on C and K. For C, choosing less atoms generally improves performance. This result is simillar to empirical DaDiL, but GMM-DaDiL is more sensitive to the overestimation problem. Furthermore, when conditioned to C, we can see that the NLL and the DA performance are negatively correlated. This remark agrees with our general intuition, as models with a smaller NLL represent better the data. For K, the situation is different. Except for $K = 31$, the DA performance remains approximately stable. Note that, for $K = 31$, one actually underfits the data from each domain in MSDA. Indeed, this choice generally yields higher NLL, and lower DA performance.

9.4 Conclusion

This chapter assembled a comparison of our methods along 5 benchmarks in visual domain adaptation, and signal processing. We started in section 9.1, where we compared our proposed GMM-OTDA with methods from the state-of-the-art, that estimate a mapping between source and target domain.



(a) Analysis of number of atoms C.

(b) Analysis of number of components K

Figure 9.11 – **Analysis of GMM-DaDiL performance with respect C and K.** On both plots, we show a KDE joint estimate of the target domain accuracy and the NLL of GMM-DaDiL. On the left, we condition our analysis on the number of atoms C. Overall, choosing less atoms leads to better performance in terms of NLL and DA performance. On the right, we condition our analysis on the number of GMM components K. Except for $K = 31$, GMM-DaDiL performance is stable across different values of K.

We show, in table 9.2, that our method is better or competitive with previous state-of-the-art methods, while being lightweight due the GMM modeling.

Next, we considered multi-source domain adaptation, in section 9.2. In this section we compared our methods relying on empirical measures, that is, WBR, WBT, and DaDiL, with the previously proposed methods WJDOT and JCPOT. We show that our methods manage to outperform these methods across all benchmarks. Furthermore, the GMM modeling further enhances the performance of DaDiL, and WBT. We then explored the inteporlation space defined by the source domains and the atoms of DaDiL, showing that, in general, interpolations in the Wasserstein hull of atoms has better adaptation performance than the Wasserstein hull of source domains.

Finally, we analyze the sensitivity of DaDiL and GMM-DaDiL with respect their hyper-parameter. For DaDiL, not surprisingly, using a higher batch size leads to stabler dictionaries with respect adaptation accuracy. Furthermore, performance is generally robust with respect the number of atoms. For GMM-DaDiL, performance is stable with respect the number of components in the GMMs, K, and number of atoms C. Overall, we highlight that using the GMM-OT framework of [17] consistently leads to better domain adaptation methods.

Chapter 10

Dataset Distillation

Contents

In modern Machine Learning (ML) practice, researchers face the challenge of reasoning about large-scale, heterogeneous datasets. This situation is challenging, as intuitive geometric concepts lose sense in high dimensions, and the computational cost of processing large amounts of data is often prohibitive. As such, [166] proposed Dataset Distillation (DD), a novel field of ML that seeks to synthesize a small dataset summary while retaining as much information as possible.

Nonetheless, current works in DD still need to consider the heterogeneity present in datasets. An example of such a phenomenon occurs in MSDA [95], where datasets contain multiple domains that follow different but related probability distributions. In this context, previous algorithms [12, 13, 14] leverage Wasserstein barycenters [82] for performing MSDA. As we argue in this paper, this mechanism can be used for distillation.

In this paper, we propose to bridge MSDA and DD, i.e., performing MSDA while summarizing the target domain. We call this new problem MSDA-DD. To this end, we adapt previous MSDA methods, such as WBT [12, 13] and DaDiL [14], and DD method Distribution Matching (DM) [167] to our setting. To the best of our knowledge, this is the first paper considering MSDA and DD simultaneously.

In the following, section 10.1 discusses previous work on DD and MSDA. Section 10.2 presents our methodology. Section 10.3 shows and discusses our experiments. Finally, section 10.4 concludes this chapter.

10.1 Dataset Distillation and Coresets

In [30], the authors informally define the problem of DD as: *approaches that aim to synthtesize tiny and high-fidelity data summaries which distill the most important knowledge from a given target dataset.*

Such distilled summaries are optimized to serve as effective drop-in replacements of the original dataset for efficient and accurate data-usage applications like model training, inference, architecture search, etc.

DD is an emerging field of ML, founded by [166], whose goal is to synthesize a small set of samples, which retain the information of the whole original dataset. Based on this idea, we can provide a definition of what a data summary is, and what it should accomplish,

Definition 30. *($\epsilon-$approximate data summary [167]) Let $\mathcal{X}$ be a feature space, $\mathcal{Y}$ be a label space, and $Q \in \mathbb{P}(\mathcal{X})$ be a probability measure. Let Θ be a parameter space, and $\mathcal{H} = \{h(\cdot;\theta) : \theta \in \Theta\}$ be a family of functions $h(\cdot;\theta) : \mathcal{X} \to \mathcal{Y}$ parametrized by θ. Let $\mathcal{A} : \cup_{n=1}^{\infty}(\mathcal{X} \times \mathcal{Y})^n \to \Theta$ be some learning algorithm. For $m \in \mathcal{N}$, the summary $D' = \{(\mathbf{x}_j, y_j)\}_{j=1}^m$ is an $\epsilon-$approximate data summary for $D = \{(\mathbf{x}_i^{(Q)}, y_i^{(Q)})\}_{i=1}^n$ if,*

$$|\mathcal{R}_Q(h(\cdot;\theta_D^\star)) - \mathcal{R}_Q(h(\cdot;\theta_{D'}^\star))| \leq \epsilon, \tag{10.1}$$

where $\theta_D^\star = \mathcal{A}(D)$ and $\theta_{D'}^\star = \mathcal{A}(D')$. Therefore, dataset distillation seeks for,

$$\begin{aligned} D_{syn}^\star = \underset{D'=\{\mathbf{x}_i,y_i\}_{i=1}^m}{\operatorname{argmin}} & \left(|\mathcal{R}_Q(h(\cdot;\theta_D^\star)) - \mathcal{R}_Q(h(\cdot;\theta_{D'}^\star))|\right) \\ & \text{subject to } \theta_{D'}^\star = \mathcal{A}(D'). \end{aligned} \tag{10.2}$$

This previous definition gives a rather general setting for dataset distillation. In practice, $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{1, \cdots, n_c\}$, $\mathcal{H}$ is the space of neural networks and $\mathcal{A}$ is gradient descent, i.e.,

$$\mathcal{A}(\{(\mathbf{x}_i, y_i)\}_{i=1}^n) = \underset{\theta\in\Theta}{\operatorname{argmin}}\, \hat{\mathcal{R}}_Q(h(\cdot;\theta)) = \underset{\theta\in\Theta}{\operatorname{argmin}}\, \frac{1}{n}\sum_{i=1}^n \mathcal{L}(h(\mathbf{x}_i;\theta), y_i).$$

Remark 11. *(On the definition of dataset distillation) In this thesis, we use the DD definition given by [167], as it is the work closest to ours. However, one should note that the definition for DD and distillation in general is blurry. For instance, a recent survey [30] proposes a different objective than equation 10.1,*

$$\sup_{\mathbf{x}\in\mathcal{X}, y\in\mathcal{Y}} \left\{ |\mathcal{L}(h(\mathbf{x};\theta_D^\star), y) - \mathcal{L}(h(\mathbf{x};\theta_{D'}^\star), y)| \right\} \leq \epsilon, \tag{10.3}$$

i.e., they seek for summaries that uniformly bound the loss function over all instances *of the feature-label joint space.*

Existing strategies. A few examples of DD methods include meta-learning [166], matching the gradients on the two datasets [168] and distribution matching [167]. This latter strategy is motivated similarly to DA. Indeed, while minimizing the objective function in equation 10.2, one effectively has two datasets, D and D'. Associated with these, one has different measures, $\hat{Q}$ and $\hat{Q}'$, which may be different, i.e., $\mathcal{D}(\hat{Q}, \hat{Q}') \neq 0$ for some metric over $\mathbb{P}(\mathcal{X})$. Based on our discussion in Chapter 4, one may minimize the difference between $\hat{\mathcal{R}}_Q$ and $\hat{\mathcal{R}}_{Q'}$ by minimizing $\mathcal{D}(\hat{Q}, \hat{Q}')$, that is, by matching these two measures. These ideas motivate the following definition,

Definition 31. *(Distribution Matching) Let $\hat{Q}$ be an empirical measures associated with samples $\{(\mathbf{x}_i^{(Q)}, \mathbf{y}_i^{(Q)})\}_{i=1}^n$. Given a metric $\mathcal{D}$ over $\mathbb{P}(\mathcal{X} \times \mathcal{Y})$, distribution matching seeks to construct a summary $D' = \{(\mathbf{x}_j^{(Q')}, \mathbf{y}_j^{(Q')})\}_{j=1}^m$, for $m \ll n$, such that the empirical measure $\hat{Q}'$ associated with D' suffices,*

$$\hat{Q}^\star = \underset{D'=\{(\mathbf{x}_j^{(Q')}, \mathbf{y}_j^{(Q')})\}_{j=1}^m}{\operatorname{argmin}} \mathcal{D}(\hat{Q}, \hat{Q}'). \tag{10.4}$$

From the perspective of this thesis, our proposed algorithms, namely WBT and DaDiL, perform distribution matching under the Wasserstein metric. In the following, we explore the idea of performing distillation and adaptation jointly. More precisely, given a set of labeled source datasets, and an unlabeled target dataset, we want to find a labeled summary for the target domain. We illustrate the idea in Figure 10.1.

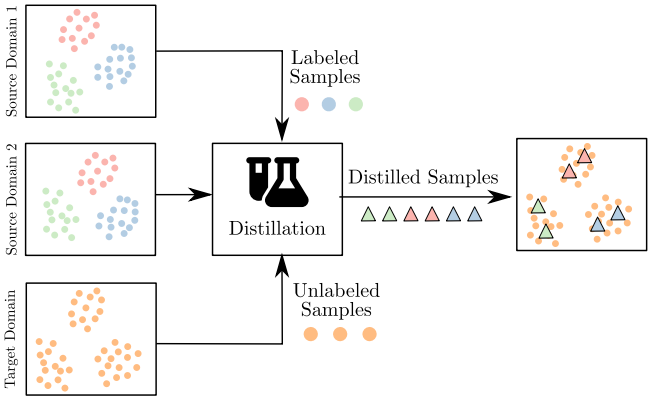


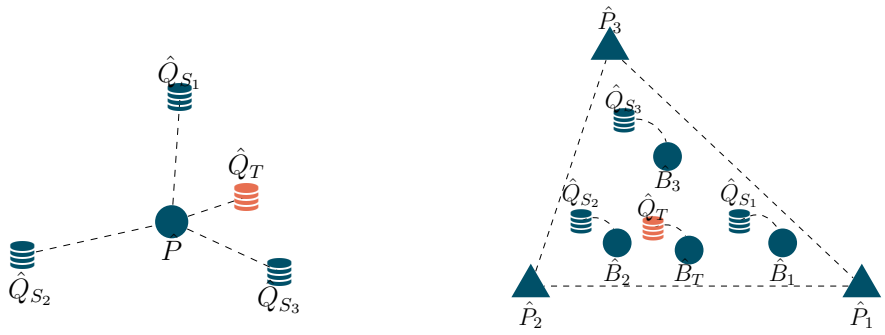
Figure 10.1 – **Illustration multi-source domain adaptation-distillation.** Given a set of labeled source domain datasets, and an unlabeled target domain dataset, we want to generate a small summary of labeled samples that match the target domain empirical measure.

10.2 MSDA through Dataset Distillation

An advantage of free-support Wasserstein barycenters (Chapter 3, Algorithm 4), is that the support $\mathbf{X}^{(B)}$ of $\hat{B}$ has a free number of samples. In this thesis, we previously treated this property as an hyper-parameter we needed to tune (e.g., Figure 9.8 in Chapter 9). In this chapter, we investigate this feature through the lens of DD, as it can be straightforwardly used to compress domains in MSDA. In what follows, we describe 3 adaptations of previously proposed algorithms for MSDA-DD.

Wasserstein Barycenter Transport. (Chapter 5) uses Wasserstein barycenters and the barycentric mapping for transporting source domain data to the target domain. This algorithm minimizes the following objective in two steps,

$$\hat{B} = \underset{\{\mathbf{x}_i^{(B)}, y_i^{(B)}\}_{i=1}^m}{\operatorname{argmin}} \mathcal{W}_2(\hat{B}, \hat{Q}_T) + \sum_{\ell=1}^{N_S} \mathcal{JW}_{2,\beta}(\hat{B}, \hat{Q}_{S_\ell}),$$



(a) MSDA-DM and WBT (b) DaDiL

Figure 10.2 – Conceptual illustration of MSDA-DD methods, where sources $\hat{Q}_{S_\ell}$ are labeled, and the target $\hat{Q}_T$ is unlabeled. The distillation is done from true datasets (⛁) towards data summaries (blue circles). We denote DaDiL's atoms by triangles. While MSDA-DM and WBT move the barycenter of true datasets towards the target domain, DaDiL learns to express datasets as Wasserstein barycenters of atoms.

where $\mathcal{W}_2$ and $\mathcal{JW}_{2,\beta}$ refer to Wasserstein and joint-Wasserstein distances (c.f., definitions 7 and 27). While the Wasserstein distance uses the squared Euclidean ground-cost, i.e., $C_{ij} = \|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2$, the joint-Wasserstein distance uses a feature-label joint cost, $C_{ij} = \|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2^2 + \beta\|\mathbf{y}_i^{(P)} - \mathbf{y}_j^{(Q)}\|_2^2$. As in previous chapters, we fix $\beta := \max_{i,j}\|\mathbf{x}_i^{(P)} - \mathbf{x}_j^{(Q)}\|_2^2$.

Recalling our discussion in Chapter 5, WBT first calculates a Wasserstein barycenter of sources, $\hat{B} = \mathcal{B}(\lambda; \mathcal{Q}_S)$, with uniform weights $\lambda_c = N_S^{-1}$, then transports the barycenter to the target domain through a barycentric mapping (c.f., equation 5.8). As a result, the compression happens at the barycenter step.

Dataset Condensation. We adapt the framework of [167] for MSDA. Instead of minimizing the distance between the summary $\hat{B}$ and a single dataset $\hat{Q}$, the goal is to minimize,

$$\hat{B} = \underset{\{\mathbf{x}_i^{(B)}\}_{i=1}^m}{\operatorname{argmin}} \ \mathrm{MMD}(\hat{B}, \hat{Q}_T) + \sum_{\ell=1}^{N_S} \mathrm{MMD}_c(\hat{B}, \hat{Q}_{S_\ell}), \text{ where } \mathrm{MMD}_c(\hat{P}, \hat{Q}) = \sum_{y=1}^{n_c} \mathrm{MMD}(\hat{P}_y, \hat{Q}_y),$$

As we show conceptually in Fig. 10.2a, MSDA-DM and WBT are conceptually close, in which they move the barycenter of labeled distributions towards the target, using the MMD, and W_2 respectively.

Dataset Dictionary Learning. (Chapter 6) uses dictionary learning. As such, DaDiL learns a set of atoms $\mathcal{P} = \{\hat{P}_c\}_{c=1}^C$ and barycentric coordinates $\Lambda = \{\lambda_\ell\}_{\ell=1}^{N_S+1}$, $\lambda_\ell \in \Delta_K$ and $\lambda_T := \lambda_{N_S+1}$, so that,

$$(\Lambda^\star, \mathcal{P}^\star) = \underset{\mathcal{P},\Lambda}{\operatorname{argmin}} \, \mathcal{W}_2(\hat{Q}_T, \mathcal{B}(\lambda_T, \mathcal{P}))^2 + \sum_{\ell=1}^{N_S} \mathcal{JW}_{2,\beta}(\hat{Q}_\ell, \mathcal{B}(\lambda_T, \mathcal{P}))^2.$$

We illustrate DaDiL conceptually in Fig. 10.2b. Effectively, DaDiL learns how to express each dataset $\hat{Q}_\ell$ as a Wasserstein barycenter of atoms, i.e., $\mathcal{B}(\lambda_\ell; \mathcal{P})$. As a consequence, one can directly compress $\hat{Q}_T$ by calculating $\hat{B}_T = \mathcal{B}(\alpha_T; \mathcal{P})$. In the following, we parametrize the number of samples in the support of barycenters by the number of Samples per Class (SPC), i.e., $n = \mathrm{SPC} \times n_c$.

10.3 Experiments and Discussion

In the following, we compare methods on 4 MSDA benchmarks: (i) Continuous Stirred Tank Reactor (CSTR) [169, 104], (ii) TEP [32, 31], (iii) CWRU [1] and (iv) Caltech-Office (CO) [114, 122]. While (i-iii) are fault diagnosis benchmarks, (iv) is a standard benchmark in visual DA. An overview is presented in table 10.1. The goal of tested algorithms is producing a small synthetic summary for the target domain. For classification, we use a Support Vector Machine (SVM) over extracted features. For the CSTR, we use the norm of the power spectrum of each sensor data [104]. For the TEP, CWRU and CO we use activations of neural networks, as in [31] and [14] respectively. All features are standardized to zero mean and unit variance.

Table 10.1 – Overview of benchmarks used in our experiments.

Benchmark	# Samples	# Domains	# Classes	# Features
CSTR	2860	7	13	7
TEP	17289	6	29	128
CWRU	24000	3	10	256
Caltech-Office 10	2533	4	10	4096

In this setting, we compare 5 methods for MSDA-DD: (i) random sampling (source-only), (ii) random sampling (target-only), (iii) WBT, (iv) MSDA-DM and (v) DaDiL. On one hand, methods (iii-v) constitute our proposed adaptations for MSDA-dd. On the other hand, (i,ii) are standard baselines in MSDA and DD. For (i), no adaptation is done towards the target, thus it is an intrinsically worst-case scenario. For (ii), there is no distribution shift, which characterizes it as a best case scenario. In both (i,ii), we randomly sample $n = SPC \times n_c$ samples from the overall data.

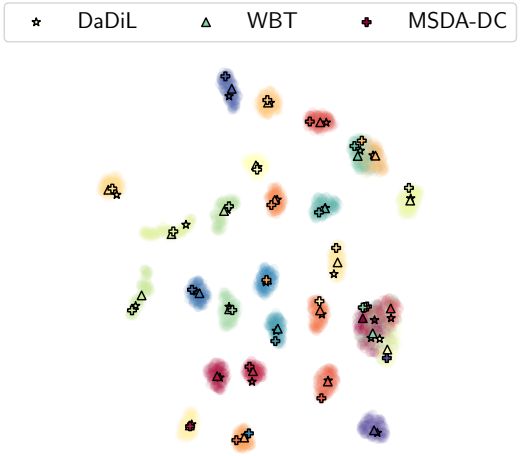


Figure 10.3 – UMAP projections of the TEP data for mode 1, where each color represent a different class.

First, we present an overview of our results in Fig. 10.4, as a function of SPC. Globally, DaDiL and WBT are largely superior to the baseline and MSDA-DM, especially in the fault diagnosis benchmarks. Surprisingly, the performance gap is more marked for small values of SPC, indicating that these methods are able to provide better generalization. This remark shows that MSDA is possible, even with as little as 1 sample for each class in the target domain. Second, unlike DD [166, 167], for an increasing SPC, MSDA-DD methods do not converge to random sampling. Indeed, random sampling the source domain do not tackle the

1. https://engineering.case.edu/bearingdatacenter/download-data-file

distributional shift problem. On the other hand one can expect random sampling the target domain to be optimal for large SPC (e.g., CWRU and CO in Fig. 10.4). Nonetheless, as demonstrated in Fig. 10.4, MSDA-DD are *sample efficient*, in the sense that they achieve high performance with as little as $SPC = 1$, which represents 0.04%, 0.16%, 0.39%, and 0.45% of the overall number of samples in CWRU, TEP, CO and CSTR benchmarks.

Next, we detail results only on the TEP benchmark. We explore how adaptation evolves for various values of SPC in the context of the 5 methods. Contrary to other benchmarks, on TEP performance remains stable over the range $SPC \in \{1, \cdots, 50\}$. On one hand, WBT and DaDiL have nearly equivalent performance, which agrees with previous research on this benchmark [31]. On the other hand, we are able to reach state-of-the-art, and to improve over the optimistic target-only scenario with only 1% of target domain samples, or 0.16% of the overall number of samples.

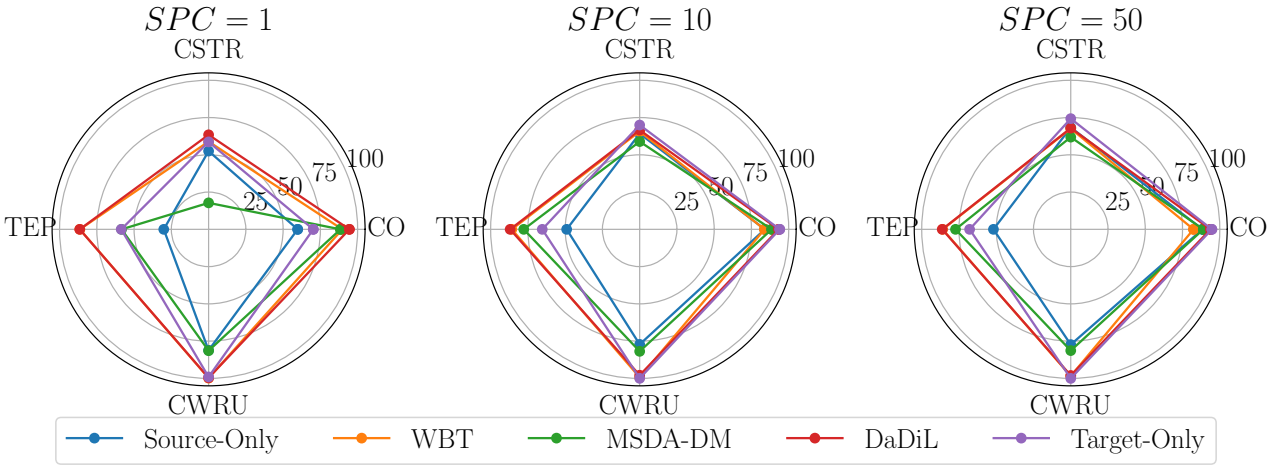


Figure 10.4 – Global comparison of MSDA methods in a distillation setting, for 1, 10 and 50 SPC.

Next, we focus on the performance gap between WBT, DaDiL and MSDA-DM. In Fig. 10.3 we show the UMAP [170] of target domain data, where the synthesized samples are highlighted. While WBT and DaDiL yield synthetic samples close to each other, MSDA-DM generates synthetic samples positioned in the wrong class cluster. This phenomenon indicates that the Wasserstein distance is a better candidate for DD. Indeed, while the *linear* MMD is only able to match 1^{st} order moments, the Wasserstein distance handles more complex distribution mismatch.

10.4 Conclusion

In this chapter, we bridge two fields of ML: MSDA and DD. We propose a new problem, called MSDA-DD, were concurrently to MSDA one also seeks to summarize the unlabeled target domain with labeled samples, while retaining as much information as possible. To that end, we adapt 3 state-of-the-art methods, [12, 13],[14], and [167] to our setting. Data summaries generated by these methods capture knowledge from the multiple labeled source domains and the unlabeled target domain itself.

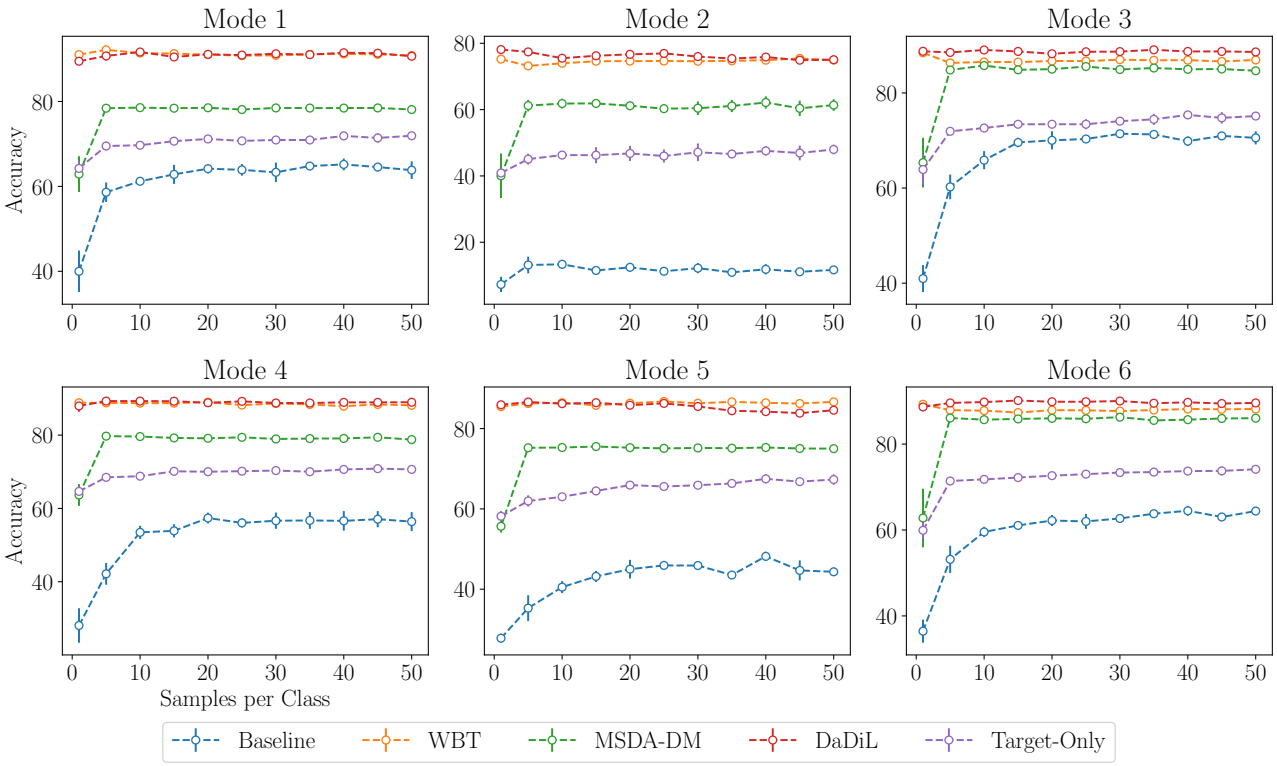


Figure 10.5 – Classification accuracy as a function of SPC, for the 6 domains in the TEP benchmark. Error bars indicate 95% confidence intervals.

We experiment extensively on 3 fault diagnosis benchmarks (CSTR, TEP, and CWRU) and 1 visual DA benchmark (Caltech-Office 10).

Our experiments show a series of intriguing results. First, we achieve state-of-the-art performance through WBT [12, 13] and DaDiL [14] with only 1 sample per class. For instance, in the context of the TEP benchmark, this represents only 1% of the samples in the target domain and 0.16% of the overall number of samples. Second, unlike previous studies in standard DD [166, 167], MSDA with DD is not equivalent to random sampling when SPC is large. This remark is due to the distributional shift phenomenon involved in MSDA.

Our work opens an interesting line of research, combining MSDA and DD. Future works include domain-incremental learning and considering label shifts between the different domains in MSDA.

Chapter 11

Conclusion

> É preciso sair da ilha para ver a ilha.
>
> José Saramago

Contents

11.1 Overview of Contributions

Wasserstein barycenters in domain adaptation. This thesis has proposed methods for domain adaptation relying on the averaging, in a Wasserstein space, of probability measures. A major challenge on the application of these ideas to domain adaptation is taking the labels of samples in the support of measures into account. In our work, we proposed a theoretically ground way of computing labeled barycenters, through the barycentric map of features, and labels. This way, the barycenter is determined as a convex combination of measures' features and labels. In this sense, we consider labels as soft probability vectors. For the source domain measures, the labels are known. As a result, they are represented as one-hot encoded vectors, that is, there is no uncertainty to which class a given sample belongs to. After computing the barycenter, the convex combination preserves membership to the n_c-simplex, i.e., the resulting vector contains soft probabilities.

Dictionary learning of free-support measures. Based on Wasserstein barycenters, we can optimize $\mathcal{B}(\lambda, \mathcal{Q}_S)$ with respect to the vector of barycentric coordinates λ so that $Q_T \overset{\mathcal{W}_2}{\approx} \mathcal{B}(\lambda, \mathcal{Q}_S)$. This corresponds to the barycentric coordinates regression problem, first proposed for histograms [28], and described in Chapter 6 for free-support measures. However, one should note that Q_T might not be approximated through barycenters of the source domains. Here, we leverage the notion of Wasserstein dictionary learning introduced by [28]. Our main contribution is defining the dictionary learning problem for free-support measures rather than histograms. As we show in example 20 and section 9.2.2, approximating Q_T with interpolations $\mathcal{B}(\lambda, \mathcal{P})$ of atoms $\mathcal{P}$ is better than using the source domains.

Gaussian-mixture based optimal transport. A limitation of DaDiL is relying on empirical measures. For instance, these measures rely directly on samples to approximate the underlying probability meausres of domains. For complex measures, one should expect a large number of samples in order to accurately represent them. Based on this idea, we leverage the GMM-OT framework of [17], for defining parametric methods for domain adaptation. For single source domain adaptation, our method rely on a new heuristic for devising a transport map between source and target mixtures. This is effectively a Gaussian mixture version of the OTDA method of [23]. For multi-source, we need to define efficient tools for computing barycenters of Gaussian mixtures. We do so in Chapter 7, where we define methods similar to the empirical case. We show through a series of experiments (e.g., Examples 24, 25, 26 and section 9.2.2) that using Gaussian mixtures is beneficial to domain adaptation performance.

11.2 Chalenges

11.2.1 Curse of Dimensionality

Let P and Q be measures on $\mathbb{P}(\mathcal{X})$, and let $\{\mathbf{x}_i^{(P)}\}_{i=1}^n$ and $\{\mathbf{x}_j^{(Q)}\}_{j=1}^n$, each i.i.d. from their respective measure. A straightforward way of estimating the distance $\mathcal{W}_\alpha(P, Q)$ is estimating $\mathcal{W}_\alpha(\hat{P}, \hat{Q})$. However, it is well known since [138], that this estimation is not so easy. Especially,

$$|\mathcal{W}_1(P, Q) - \mathcal{W}_1(\hat{P}, \hat{Q})| \sim \mathcal{O}(n^{-1/d}).$$

This result means that, as the dimensionality of the data grows, the convergence of the empirical estimator becomes increasingly slower. We illustrate the rate of convergence with respect to the dimensionality of the data in Figure 11.1.

This issue becomes dramatic in machine learning application, as the data and feature spaces are high dimensional (see e.g., Table 9.1). The same applies to other optimal transport related quantities, such as the barycentric map [45]. A possible workaround this issue is considering a parametric model for the data. For instance, using a Gaussian model allows one to derive dimension-free convergence rates, that is, $\mathcal{O}(n^{-1/2})$.

Given these ideas, our work relying on empirical optimal transport is itself subject to the curse of dimensionality. However, our methods relying on the GMM-OT framework may offer a workaround the curse of dimensionality. Especially, so far no analysis of the rate of convergence of the mixture-Wasserstein distance has been conducted. Meanwhile, our results seem to point in the direction that the statistical estimation of Gaussian mixture models is easier than empirical measures. This would

explain why GMM-OTDA and GMM-DaDiL perform well in comparison their empirical counterparts. We leave these questions for future works.

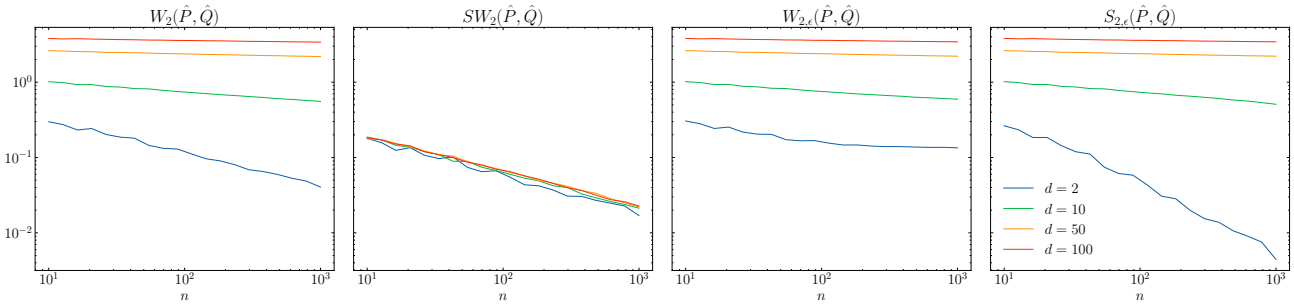


Figure 11.1 – Estimation of Wasserstein distances with finite samples as a function of number of samples n, and dimensions d. Overall, the plug-in empirical estimator $\mathcal{W}_2(\hat{P},\hat{Q})$ suffers from the curse of dimensionality, as estimation becomes harder in high dimensions. This issue can be alleviated through alternative estimators, such as sliced Wasserstein or entropic OT.

11.2.2 Class Imbalance

Throughout the chapters in Part II, our methods considered taking barycenters over measures in $\mathbb{P}(\mathcal{X}\times\mathcal{Y})$. This offers some flexibility, as measures can have shifts on $P(X)$, as well as $P(Y)$. It should be noted, however, that if the measures in $\mathcal{P}$ have different $P(Y)$, OT is forced to match samples from different classes. This issue is considered, for instance, in JCPOT [24].

To verify the possible issues associated with class imbalance, we adapt the example used in Part II for a class imbalance scenario (see, e.g., example 13 in Chapter 5). Note that, as we have two classes in these problems, we denote the proportions by a vector $\pi \in \Delta_2$. As such, we consider $\pi_{S_1} = [0.2, 0.8]$, $\pi_{S_2} = [0.4, 0.6]$, $\pi_{S_3} = [0.6, 0.4]$ and $\pi_T = [0.8, 0.2]$. The resulting measures are shown in Figure 11.2

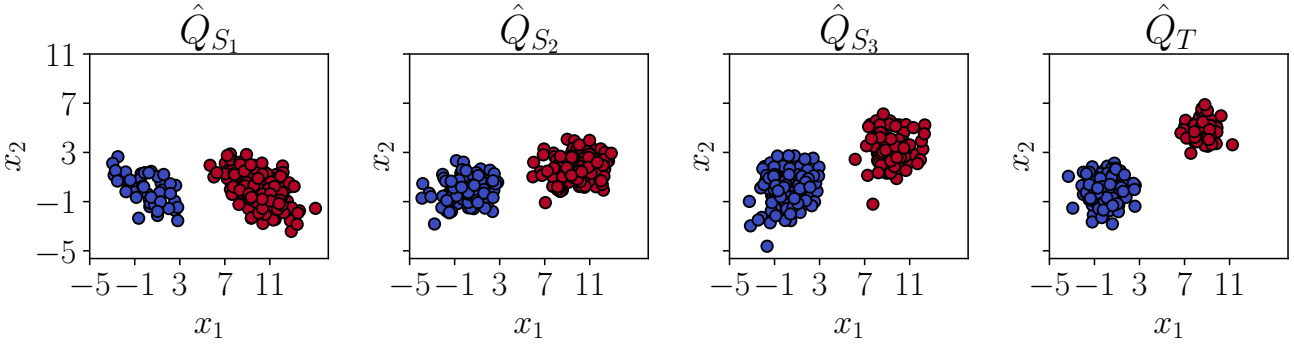


Figure 11.2 – Probability measures considered in our illustration of issues with class imbalance.

As we mentioned earlier, computing a barycenter among the different $\mathcal{Q}_S = \{\hat{Q}_{S_1}, \hat{Q}_{S_2}, \hat{Q}_{S_3}\}$ means that we need to compute a transport plan between each of these measures. However, as each of these have different class proportions, OT is forced to match points from different classes. Naturally, what

happens is that the labels get mixed in the process, as we are computing convex combinations of one-hot encoded vectors. This is evidenced in Figure 11.3, which shows that WBT cannot handle class imbalance.

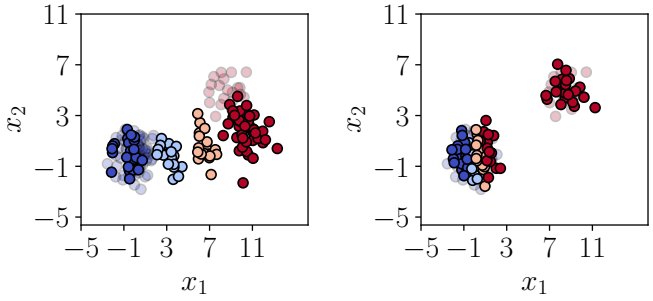


Figure 11.3 – On the left, barycenter of class-imbalanced empirical measures. On the right, result of transporting the barycenter towards the target. The resulting barycentric measure has fuzzy labels, as a result of combining one-hot encoded label vectors from different classes.

It would be an immediate conclusion to say that DaDiL suffers from the same problem as WBT. The situation, however, is more intricate, as we are actually computing atomic measures. The barycenters are computed according to these measures, rather than the sources. Since the sources are labeled, we can ensure that the mini-batching process samples balanced proportions. As a result, we expect that the atoms do not have fuzzy labels. This is indeed verified, as shown in Figure 11.4. Note, however, that some of the atomic measures mix the classes.

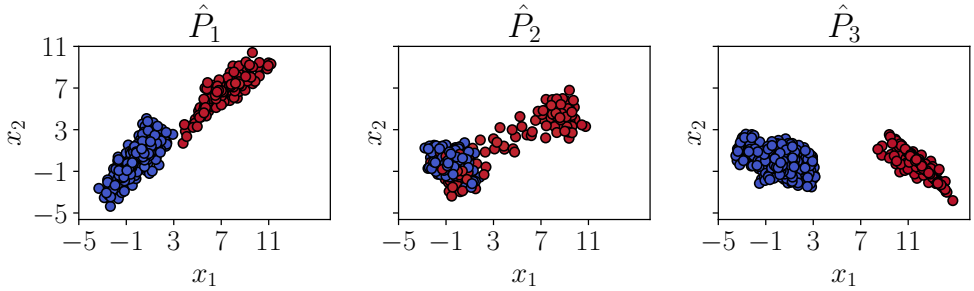


Figure 11.4 – Atoms learned by DaDiL, in an imbalanced scenario.

As a result, the issue in DaDiL is not necessarily with respect the computation of barycenters. One can expect the issue of OT between imbalanced classes happening at the level of $\mathcal{W}_2(Q_T, \mathcal{B}(\lambda_T, \mathcal{P}))^2$, exclusively. Overall, Figure 11.4 seems to indicate a relative robustness to class imbalance, which would help explaining its superior performance.

On the class imbalance issue, we consider a few solutions. First, one could design a mechanism similar to unbalanced OT, in which samples from the majority class are not forced to match with samples from the minority class. The challenge, here, is overcoming the regularization artifacts in the barycenter. Second, one could devise a proportion estimation mechanism similar to [24].

11.3 Perspectives and Future Works

11.3.1 Generative Modelling

While we did not explicitly discuss this point, our work has a non-negligible intersection with the field of generative modelling. We recall that such models are defined in opposition to discriminative models. While the former models the joint measure $P(X, Y)$, the latter models the conditional $P(Y|X)$. In our empirical methods we are *generating synthetic samples*. For instance, when calculating Wasserstein barycenters through Algorithm 6, we are generating new points in the support of the barycentric measure. For DaDiL, we are synthesizing samples in the atoms support. As a result, one can see these approaches doing a similar task as generative neural nets [73], even though we generate a fixed amount of data points. To further stress our argument, the generative side of our work becomes clearer through Gaussian mixtures, which are popular generative models.

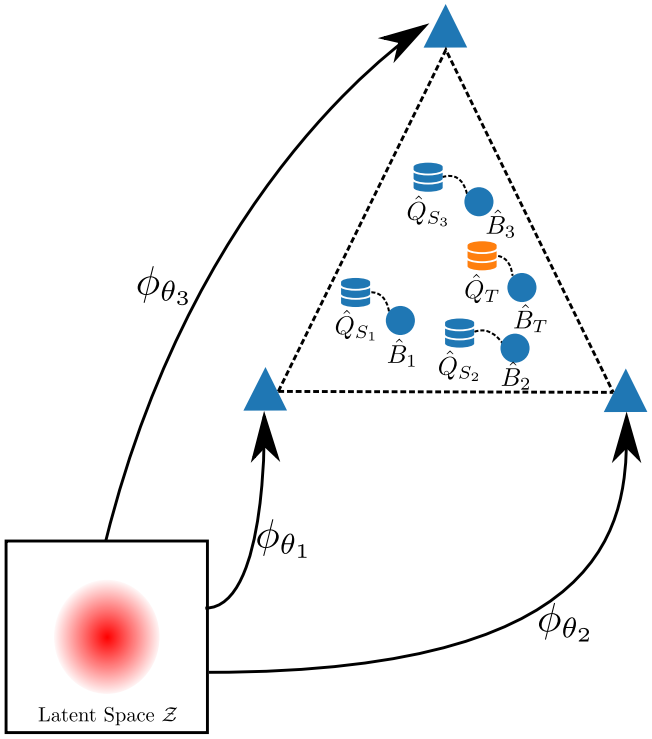


Figure 11.5 – Generative estimation of atoms in DaDiL. In this case, the atoms are represented through $(\phi_{\theta_c})_\sharp P_0$, where P_0 is a simple probability measure in a latent space $\mathcal{Z}$.

It is natural to ask oneself if rather than considering empirical measures or some model such as GMMs, one could use neural nets. In this case, DaDiL would be equivalent to the following minimization problem,

$$(\Lambda, \Theta) = \underset{\Lambda,\Theta}{\operatorname{argmin}} \mathcal{W}_2(Q_T, \mathcal{B}(\lambda_T, \mathcal{P}_\Theta))^2 + \sum_{\ell=1}^{N_S} \mathcal{JW}_{2,\beta}(Q_{S_\ell}, \mathcal{B}(\lambda_\ell, \mathcal{P}_\Theta)^2), \tag{11.1}$$

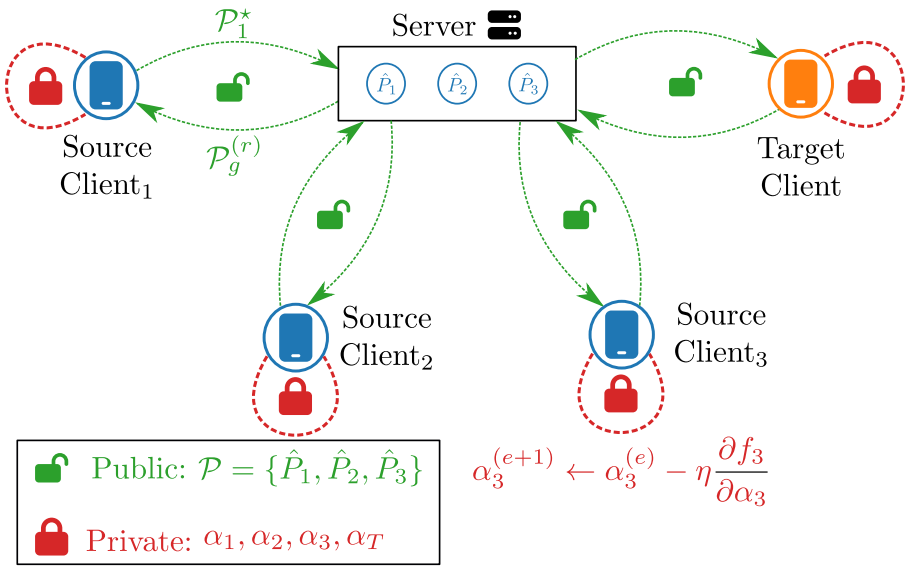


Figure 11.6 – Federated learning of atoms by clients coordinated by a server, as we studied in [137]. Future works can focus on whether atoms can be publicly disclosed without revealing clients' sensitive information.

where $\mathcal{P}_\Theta = \{(\phi_{\theta_c})_\sharp P_0\}_{c=1}^C$, for a neural net $\phi_\theta : \mathcal{Z} \to \mathcal{X}$, mapping some latent space $\mathcal{Z}$ to the feature space $\mathcal{X}$, and P_0 is some simple probability measure (e.g. $\mathcal{N}(\mathbf{0}, \mathbf{I})$). This idea is illustrated in Figure 11.5

11.3.2 Label Encoding

As we covered in chapter 5, in this thesis we took the decision to interpret labels as soft-probability vectors, that is, as elements of the n_c-simplex. In our theoretical results, it seemed natural to assume an weighted Euclidean metric for the labels, that is,

$$d(\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2) = \sqrt{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 + \beta \|\mathbf{y}_1 - \mathbf{y}_2\|_2^2}.$$

However, this choice disregards the geometry of the classes in the learning problem. Indeed, take a problem with 3 classes. With this modeling choice, these classes are equidistant, that is, $d_{\mathcal{Y}}(\mathbf{e}_1, \mathbf{e}_2) = d_{\mathcal{Y}}(\mathbf{e}_1, \mathbf{e}_3) = d_{\mathcal{Y}}(\mathbf{e}_2, \mathbf{e}_3)$, where $\mathbf{e}_i$ is a vector of zeros with 1 in its $i-$th position. In Chapter 5, we considered a geometry-aware metric introduced by [25], the Optimal Transport Dataset Distance. However, this distance relies on a categorical notion of labels, rather than a continuum. This makes it difficult to adapt our methods (e.g., DaDiL) to work with this metric. Future works could consider a more general way of embedding labels in an Euclidean space, so that these can be continuously differentiable and the geometry of their classes may be taken into account.

11.3.3 Federated Learning and Differential Privacy

A future work perspective is the study of our methods in a federated context. This is the case of our publications [137] and [171], who explored the optimization of DaDiL without the need for communicating domain data explicitly. Federated learning is a learning scenario where learning happens in a decentralized way, across different clients. In federated domain adaptation, each domain data is held by a different client, and adaptation must be carried out towards a target without explicitly communicating clients' data. The main idea is to decouple the optimization of the atoms $\mathcal{P}$ from that of the barycentric coordinates Λ. In the context of DaDiL, the barycentric coordinates work as a *key*, that allows clients to retrieve their own data given that they have the correct vector of barycentric coordinates. As a result, the atoms can be exchanged freely, as they do not necessarily reveal clients' data. An illustration of these ideas is shown in Figure 11.6.

Besides the challenge of designing a federated training strategy, a parallel direction is analyzing the privacy capabilities of DaDiL. For instance, an open question in [137] and [171] is whether DaDiL fits into the differential privacy framework of [172]. A further question is, given a malicious client in federated learning, how much of clients' information can it access given its access to the public atoms.

11.3.4 Online and Incremental Domain Adaptation

Our final perspective concerns online domain adaptation. In this case, at least one domain (e.g., the target domain) is available in a stream, rather than being stored offline. In this context, a couple of options are available. First, one can do optimization steps in the DaDiL objective as batches of data become available. This idea is problematic, as the method might not converge as the stream terminates. Second, one can perform an online fit of some probabilistic model with the batches of data. A possible way of doing so is through GMMs [173]. We explore this idea in [174], where we show that GMM-DaDiL can be fit online, with sometimes better performance than its offline version.

A further research direction is considering incremental learning. In this case, on top of samples arriving on a stream, they arrive in a particular order. For instance, in *task incremental* learning, classes may appear one after the other. In *domain incremental* learning, one domain appears after the other. The goal is to have a model that do not forget, once it is retrained with new data, what it has learned before. Here, one can make a nice link with our work in data distillation. Indeed, the small summaries can serve as a memory of tasks or domains, so that one can use this memory as a replay buffer.

Bibliography

[1] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, 2016.

[2] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[3] A. M. TURING. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, 10 1950.

[4] Leah Henderson. The Problem of Induction. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2022 edition, 2022.

[5] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

[6] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[7] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

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

[9] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

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

[11] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

[12] Eduardo Fernandes Montesuma and Fred-Maurice Ngolè Mboula. Wasserstein barycenter transport for acoustic adaptation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3405–3409. IEEE, 2021.

[13] Eduardo Fernandes Montesuma and Fred Maurice Ngole Mboula. Wasserstein barycenter for multi-source domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16785–16793, 2021.

[14] Eduardo Fernandes Montesuma, Fred Ngolè Mboula, and Antoine Souloumiac. Multi-source domain adaptation through dataset dictionary learning in wasserstein space. In *26th European Conference on Artificial Intelligence (ECAI)*, pages 1739–1746. IOS Press, 2023.

[15] Eduardo Fernandes Montesuma, Fred Mboula, and Antoine Souloumiac. Lighter, better, faster multi-source domain adaptation with gaussian mixture models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 341–356. Springer, 2024.

[16] Eduardo Fernandes Montesuma, Fred Maurice Ngolè Mboula, and Antoine Souloumiac. Optimal transport for domain adaptation through gaussian mixture models. *arXiv preprint arXiv:2403.13847*, 2024.

[17] Julie Delon and Agnes Desolneux. A wasserstein-type distance in the space of gaussian mixture models. *SIAM Journal on Imaging Sciences*, 13(2):936–970, 2020.

[18] Eduardo Fernandes Montesuma, Fred Ngole Mboula, and Antoine Souloumiac. Recent advances in optimal transport for machine learning. *arXiv preprint arXiv:2306.16156*, 2023.

[19] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[20] 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.

[21] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younes Bennani. *Advances in domain adaptation theory*. Elsevier, 2019.

[22] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv preprint arXiv:2004.11829*, 2020.

[23] 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.

[24] Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia. Optimal transport for multi-source domain adaptation under target shift. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 849–858. PMLR, 2019.

[25] David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 33:21428–21439, 2020.

[26] 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.

[27] Morgan A Schmitz, Matthieu Heitz, Nicolas Bonneel, Fred Ngole, David Coeurjolly, Marco Cuturi, Gabriel Peyré, and Jean-Luc Starck. Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning. *SIAM Journal on Imaging Sciences*, 11(1):643–678, 2018.

[28] Nicolas Bonneel, Gabriel Peyré, and Marco Cuturi. Wasserstein barycentric coordinates: histogram regression using optimal transport. *ACM Trans. Graph.*, 35(4):71–1, 2016.

[29] James J Downs and Ernest F Vogel. A plant-wide industrial process control problem. *Computers & chemical engineering*, 17(3):245–255, 1993.

[30] Noveen Sachdeva and Julian McAuley. Data distillation: A survey. *Transactions on Machine Learning Research*, 2023. Survey Certification.

[31] Eduardo Fernandes Montesuma, Michela Mulas, Fred Ngolè Mboula, Francesco Corona, and Antoine Souloumiac. Multi-source domain adaptation for cross-domain fault diagnosis of chemical processes. *arXiv e-prints*, pages arXiv–2308, 2023.

[32] Christopher Reinartz, Murat Kulahci, and Ole Ravn. An extended tennessee eastman simulation dataset for fault-detection and decision support systems. *Computers & Chemical Engineering*, 149:107281, 2021.

[33] Eduardo Fernandes Montesuma, Fred Ngolè Mboula, and Antoine Souloumiac. Multi-source domain adaptation meets dataset distillation through dataset dictionary learning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5620–5624. IEEE, 2024.

[34] L Kantorovich. On the transfer of masses (in russian). In *Doklady Akademii Nauk*, volume 37, pages 227–229, 1942.

[35] Donald L Cohn. *Measure theory*, volume 5. Springer, 2013.

[36] Asuka Takatsu. Wasserstein geometry of Gaussian measures. *Osaka Journal of Mathematics*, 48(4):1005 – 1026, 2011.

[37] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[38] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[39] Benoit Kloeckner. A geometric study of wasserstein spaces: Euclidean spaces. *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze*, 9(2):297–323, 2010.

[40] Bruno Lévy and Erica L Schwindt. Notions of optimal transport theory and how to implement them on a computer. *Computers & Graphics*, 72:135–148, 2018.

[41] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *J. Mach. Learn. Res.*, 22(78):1–8, 2021.

[42] Marco Cuturi, Laetitia Meng-Papaxanthos, Yingtao Tian, Charlotte Bunne, Geoff Davis, and Olivier Teboul. Optimal transport tools (ott): A jax toolbox for all things wasserstein. *arXiv preprint arXiv:2201.12324*, 2022.

[43] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

[44] Vivien Seguy, Bharath Bhushan Damodaran, Remi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large scale optimal transport and mapping estimation. In *International Conference on Learning Representations*, 2018.

[45] Nabarun Deb, Promit Ghosal, and Bodhisattva Sen. Rates of estimation of optimal transport maps using plug-in estimators via barycentric projections. *Advances in Neural Information Processing Systems*, 34:29736–29753, 2021.

[46] François Bolley, Arnaud Guillin, and Cédric Villani. Quantitative concentration inequalities for empirical measures on non-compact spaces. *Probability Theory and Related Fields*, 137(3-4):541–593, 2007.

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

[48] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[49] Bernhard Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019.

[50] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[51] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617. PMLR, 2018.

[52] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alexander J Smola. A kernel approach to comparing distributions. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 1637. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

[53] Giulia Luise, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. Differential properties of sinkhorn approximation for learning with wasserstein distance. *Advances in Neural Information Processing Systems*, 31, 2018.

[54] Aude Genevay, Lénaic Chizat, Francis Bach, Marco Cuturi, and Gabriel Peyré. Sample complexity of sinkhorn divergences. In *The 22nd international conference on artificial intelligence and statistics*, pages 1574–1583. PMLR, 2019.

[55] Grégoire Montavon, Klaus-Robert Müller, and Marco Cuturi. Wasserstein training of restricted boltzmann machines. In *NIPS*, pages 3711–3719, 2016.

[56] Max Sommerfeld, Jörn Schrieber, Yoav Zemel, and Axel Munk. Optimal transport: Fast probabilistic approximation with exact solvers. *J. Mach. Learn. Res.*, 20:105–1, 2019.

[57] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 447–463, 2018.

[58] Espen Bernton, Pierre E Jacob, Mathieu Gerber, and Christian P Robert. On parameter estimation with the wasserstein distance. *Information and Inference: A Journal of the IMA*, 8(4):657–676, 2019.

[59] Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Learning with minibatch wasserstein: asymptotic and gradient properties. In *AISTATS*, 2020.

[60] Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

[61] Kilian Fatras, Thibault Séjourné, Rémi Flamary, and Nicolas Courty. Unbalanced minibatch optimal transport; applications to domain adaptation. In *International Conference on Machine Learning*, pages 3186–3197. PMLR, 2021.

[62] Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3):969–1117, 2018.

[63] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018.

[64] Laetitia Chapel, Mokhtar Z Alaya, and Gilles Gasso. Partial optimal tranport with applications on positive-unlabeled learning. *Advances in Neural Information Processing Systems*, 33:2903–2913, 2020.

[65] Khai Nguyen, Dang Nguyen, Tung Pham, Nhat Ho, et al. Improving mini-batch optimal transport via partial transportation. In *International Conference on Machine Learning*, pages 16656–16690. PMLR, 2022.

[66] Yongxin Chen, Tryphon T Georgiou, and Allen Tannenbaum. Optimal transport for gaussian mixture models. *IEEE Access*, 7:6269–6278, 2018.

[67] Rémi Flamary, Karim Lounici, and André Ferrari. Concentration bounds for linear monge mapping estimation and optimal transport domain adaptation. *arXiv preprint arXiv:1905.10155*, 2019.

[68] Imre Csiszár. Information-type measures of difference of probability distributions and indirect observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

[69] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

[70] Christopher M Bishop. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[71] Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in applied probability*, 29(2):429–443, 1997.

[72] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. On integral probability metrics,\phi-divergences and binary classification. *arXiv preprint arXiv:0901.2698*, 2009.

[73] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[74] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30, 2017.

[75] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

[76] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[77] Alex J Smola and Bernhard Schölkopf. *Learning with kernels*, volume 4. The MIT Press, 1998.

[78] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[79] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.

[80] Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of machine learning research*, 2(Nov):67–93, 2001.

[81] Maurice Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. In *Annales de l'institut Henri Poincaré*, volume 10, pages 215–310, 1948.

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

[83] Wilfrid Gangbo and Andrzej Święch. Optimal maps for the multidimensional monge-kantorovich problem. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 51(1):23–45, 1998.

[84] Guillaume Carlier and Ivar Ekeland. Matching for teams. *Economic theory*, 42(2):397–418, 2010.

[85] Pedro C Álvarez-Esteban, E Del Barrio, JA Cuesta-Albertos, and C Matrán. A fixed-point approach to barycenters in wasserstein space. *Journal of Mathematical Analysis and Applications*, 441(2):744–762, 2016.

[86] David Hume. *A treatise of human nature*. Clarendon Press, 1896.

[87] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[88] Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

[89] Ievgen Redko, Titouan Vayer, Rémi Flamary, and Nicolas Courty. Co-optimal transport. *Advances in Neural Information Processing Systems*, 33(17559-17570):2, 2020.

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

[91] Yi-Ren Yeh, Chun-Hao Huang, and Yu-Chiang Frank Wang. Heterogeneous domain adaptation and classification by exploiting the correlation subspace. *IEEE Transactions on Image Processing*, 23(5):2009–2018, 2014.

[92] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 135–150, 2018.

[93] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 153–168, 2018.

[94] Huailiang Zheng, Rixin Wang, Yuantao Yang, Jiancheng Yin, Yongbo Li, Yuqing Li, and Minqiang Xu. Cross-domain fault diagnosis using knowledge transfer strategy: a review. *IEEE Access*, 7:129260–129290, 2019.

[95] Koby Crammer, Michael Kearns, and Jennifer Wortman. Learning from multiple sources. *Journal of Machine Learning Research*, 9(8), 2008.

[96] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. In *International Conference on Learning Representations*, 2019.

[97] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

[98] Tugkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D Smith, and Patrick White. Testing that distributions are close. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 259–269. IEEE, 2000.

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

[100] 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.

[101] Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The annals of statistics*, pages 2263–2291, 2013.

[102] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. *Advances in neural information processing systems*, 21, 2008.

[103] Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*, pages 274–289. Springer, 2014.

[104] Eduardo Fernandes Montesuma, Michela Mulas, Francesco Corona, and Fred-Maurice Ngole Mboula. Cross-domain fault diagnosis through optimal transport for a cstr process. *IFAC-PapersOnLine*, 55(7):946–951, 2022.

[105] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in Neural Information Processing Systems*, 30, 2017.

[106] Mourad El Hamri, Younes Bennani, and Issam Falih. Hierarchical optimal transport for unsupervised domain adaptation. *Machine Learning*, 111(11):4159–4182, 2022.

[107] Ching-Yao Chuang, Stefanie Jegelka, and David Alvarez-Melis. Infoot: Information maximizing optimal transport. In *International Conference on Machine Learning*, pages 6228–6242. PMLR, 2023.

[108] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[109] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[110] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[111] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[112] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.

[113] Rosanna Turrisi, Rémi Flamary, Alain Rakotomamonjy, et al. Multi-source domain adaptation via weighted joint distributions optimal transport. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.

[114] 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.

[115] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447, 2007.

[116] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3964–3973, 2018.

[117] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[118] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *Workshop on Deep Learning and Unsupervised Feature Learning*, 2016.

[119] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[120] Tobias Ringwald and Rainer Stiefelhagen. Adaptiope: A modern benchmark for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 101–110, 2021.

[121] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

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

[123] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[124] 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.

[125] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[126] Gabriela Csurka, Riccardo Volpi, and Boris Chidlovskii. Unsupervised domain adaptation for semantic image segmentation: a comprehensive survey. *arXiv preprint arXiv:2112.03241*, 2021.

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

[128] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[129] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[130] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. *Advances in neural information processing systems*, 28, 2015.

[131] 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. PMLR, 2014.

[132] David Alvarez-Melis and Nicolò Fusi. Dataset dynamics via gradient flows in probability space. In *International Conference on Machine Learning*, pages 219–230. PMLR, 2021.

[133] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[134] Antoine Rolet, Marco Cuturi, and Gabriel Peyré. Fast dictionary learning with a smoothed wasserstein loss. In *Artificial Intelligence and Statistics*, pages 630–638. PMLR, 2016.

[135] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[136] Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Minibatch optimal transport distances; analysis and applications. *arXiv preprint arXiv:2101.01792*, 2021.

[137] Fabiola Espinoza Castellon, Eduardo Fernandes Montesuma, Fred Ngolè Mboula, Aurélien Mayoue, Antoine Souloumiac, and Cédric Gouy-Pailler. Federated dataset dictionary learning for multi-source domain adaptation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5610–5614. IEEE, 2024.

[138] Richard Mansfield Dudley. The speed of mean glivenko-cantelli convergence. *The Annals of Mathematical Statistics*, 40(1):40–50, 1969.

[139] Jonathan Weed and Francis Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *Bernoulli*, 25(4A):2620 – 2648, 2019.

[140] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[141] SN Afriat. Theory of maxima and the method of lagrange. *SIAM Journal on Applied Mathematics*, 20(3):343–357, 1971.

[142] Dimitri Bertsekas. *Network optimization: continuous and discrete models*, volume 8. Athena Scientific, 1998.

[143] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690. PMLR, 2019.

[144] Rolf Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

[145] 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.

[146] Afrânio Melo, Maurício M Câmara, Nayher Clavijo, and José Carlos Pinto. Open benchmarks for assessment of process monitoring and fault diagnosis techniques: A review and critical analysis. *Computers & Chemical Engineering*, 165:107964, 2022.

[147] Andreas Bathelt, N Lawrence Ricker, and Mohieddine Jelali. Revision of the tennessee eastman process model. *IFAC-PapersOnLine*, 48(8):309–314, 2015.

[148] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[149] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[150] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[151] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.

[152] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[153] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[154] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[155] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[156] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2022.

[157] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 404–417. Springer, 2006.

[158] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[159] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[160] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523, 2020.

[161] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, number 14, pages 281–297. Oakland, CA, USA, 1967.

[162] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[163] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. Unsupervised domain adaptation: A reality check. *arXiv preprint arXiv:2111.15672*, 2021.

[164] Jianfei Yang, Hanjie Qian, Yuecong Xu, Kai Wang, and Lihua Xie. Can we evaluate domain adaptation models without target-domain labels? In *The Twelfth International Conference on Learning Representations*, 2024.

[165] 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. JMLR Workshop and Conference Proceedings, 2010.

[166] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

[167] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023.

[168] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021.

[169] Karl Ezra Salgado Pilario and Yi Cao. Canonical variate dissimilarity analysis for process incipient fault detection. *IEEE Transactions on Industrial Informatics*, 14(12):5308–5315, 2018.

[170] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[171] Eduardo Fernandes Montesuma, Fabiola Espinoza Castellon, Fred Ngolè Mboula, Aurélien Mayoue, Antoine Souloumiac, and Cédric Gouy-Pailler. Decentralized multi-source domain adaptation through dataset dictionary learning. *arXiv preprint arXiv:2306.16156*, 2024.

[172] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

[173] Juan M Acevedo-Valle, Karla Trejo, and Cecilio Angulo. Multivariate regression with incremental learning of gaussian mixture models. In *CCIA*, pages 196–205, 2017.

[174] Eduardo Fernandes Montesuma, Stevan Le Stanc, and Fred Ngolè Mboula. Online multi-source domain adaptation through gaussian mixtures and dataset dictionary learning. In *2024 IEEE 34rd International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 5620–5624. IEEE, 2024.