Fundação Getulio Vargas
Escola de Matemática Aplicada

Davi Sales Barreira

Optimal Transport for Machine Learning: Theory and Applications

Rio de Janeiro
2021

Davi Sales Barreira

Optimal Transport for Machine Learning: Theory and Applications

Dissertação submetida à Escola de Matemática Aplicada como requisito parcial para a obtenção do grau de Mestre em Modelagem Matemática da Informação.

Área de Concentração: Data Science

Orientador: Eduardo Fonseca Mendes

Rio de Janeiro
2021

Dados Internacionais de Catalogação na Publicação (CIP)
Ficha catalográfica elaborada pelo Sistema de Bibliotecas/FGV

Barreira, Davi Sales
Optimal transport for machine learning: theory and applications / Davi Sales Barreira. – 2021.
137 f.

Dissertação (mestrado) - Fundação Getulio Vargas, Escola de Matemática Aplicada.
Orientador: Eduardo Fonseca Mendes.
Inclui bibliografia.

1. Problemas de transporte (Programação). 2. Aprendizado do computador. 3. Otimização matemática. 4. Análise combinatória. I. Mendes, Eduardo Fonseca. II. Fundação Getulio Vargas. Escola de Matemática Aplicada. III. Título.

CDD – 006.31

Elaborada por Kelly Ayala – CRB-7/7007

DAVI SALES BARREIRA

“OPTIMAL TRANSPORT FOR MACHINE LEARNING: THEORY AND APPLICATIONS”.

Dissertação apresentado(a) ao Curso de Mestrado em Modelagem Matemática do(a) Escola de Matemática Aplicada para obtenção do grau de Mestre em Modelagem Matemática.

Data da defesa: 25/03/2021

ASSINATURA DOS MEMBROS DA BANCA EXAMINADORA

Presidente da Comissão Examinadora: Profº **Eduardo Fonseca Mendes**

Eduardo Fonseca Mendes
Orientador

Alberto Paccanaro
Membro Interno

Roberto Imbuzeiro Moraes Felinto de Oliveira
Membro Externo

Nos termos da Lei nº 13.979 de 06/02/20 - DOU nº 27 de 07/02/20 e Portaria MEC nº 544 de 16/06/20 - DOU nº 114 de 17/06/20 que dispõem sobre a suspensão temporária das atividades acadêmicas presenciais e a utilização de recursos tecnológicos face ao COVID-19, as apresentações das defesas de Tese e Dissertação, de forma excepcional, serão realizadas de forma remota e síncrona, incluindo-se nessa modalidade membros da banca e discente.

César Leopoldo Camacho Manco
Diretor

Antonio de Araujo Freitas Junior
Pró-Reitor de Ensino, Pesquisa e Pós-Graduação FGV

Antonio Freitas, PhD
Pró-Reitor de Ensino, Pesquisa e Pós-Graduação
Fundação Getúlio Vargas

Instrução Normativa nº 01/19, de 09/07/19 - Pró-Reitoria FGV
Em caso de participação de Membro(s) da Banca Examinadora de forma não-presencial*, o Presidente da Comissão Examinadora assinará o documento como representante legal, delegado por esta I.N.
*Skype, Videoconferência, Apps de vídeo etc

Agradecimentos

Agradeço à minha família, namorada e amigos, por todo o suporte emocional, que me foi tão necessário para concluir este trabalho. Também agradeço ao meu orientador, Eduardo Fonseca Mendes, por todo o suporte na elaboração desta dissertação e por me ajudar a manter o foco.

Abstract

In recent years, advances in Optimal Transport have led to a surge of applications in fields such as Economics, Quantitative Finance and Signal Processing, among others. One area in which it has been found particularly successful is Machine Learning. The development of computationally efficient methods for solving Optimal Transport problems opened doors for creating Machine Learning algorithms using concepts from Optimal Transport. These new algorithms encompass many different sub-areas such as Transfer Learning, Clustering, Dimensionality Reduction, Generative Models, just to name some.

This work provides an overview of the different ways in which Optimal Transport has been used in Machine Learning, thus helping Machine Learning researchers to better understand its impact in the field and how to use it. This thesis first introduces the main theoretical and computational aspects of Optimal Transport theory in an accessible way to Machine Learning researchers, followed by a semi-systematic literature review focusing on the main uses of Optimal Transport in Machine Learning.

Keywords: Optimal Transport, Wasserstein distance, Machine Learning, Literature Review.

Contents

Notation

The following symbols are used in the text without always recalling their meaning.

- $\mathcal{M}(X), \mathcal{M}_+(X)$: Space of finite measures and finite positive measures on X, respectively.
- $\mathcal{P}(X), \mathcal{P}_p(X)$: Space of probability measures and space of probability measures with pth finite moment, respectively.
- $\mathbb{1}_A(x)$: Indicator function of set A, i.e. $\mathbb{1}_A(x) = 1$ if $x \in A$ and 0 otherwise.
- $\mathbf{1}_n$: n dimensional vector of ones.
- id: Identity operator, i.e. $id(x) = x$.
- $\oplus$: For $\phi : X \to \mathbb{R}$, $\psi : Y \to \mathbb{R}$, then $(\phi \oplus \psi)(x, y) = \phi(x) + \psi(y)$.
- π_X: Projection operator on X, i.e. for $\pi_X : X \times Y \to X$, then $\pi_X(x, y) = x,\ \forall (x, y) \in X \times Y$.
- $\Pi(\mu, \nu)$: Coupling of measures μ and ν.
- $\mathbb{R}_+$: Positive real number greater or equal than 0.
- $\overline{\mathbb{R}}$: Real numbers extended to include $+\infty$ and $-\infty$.
- $C(X)$: Set of functions $f : X \to \mathbb{R}$, where f is continuous.
- $C_b(X)$: Set of functions $f : X \to \mathbb{R}$, where f is continuous and bounded.

- $C_0(X)$: Set of functions $f : X \to \mathbb{R}$, where f is continuous and goes to zero at infinity.
- $C_c(X)$: Set of functions $f : X \to \mathbb{R}$, where f is continuous and has compact support.
- $\mu_n \rightharpoonup \mu$: Measure μ_n converges weakly to μ.
- $OT_c(\mu, \nu)$: Optimal Transport cost between measures μ and ν for a ground cost function c.
- $OT_{c,\varepsilon}(\mu, \nu)$, $\overline{OT}_{c,\varepsilon}(\mu, \nu)$: Entropic Optimal Transport distance and the Entropic Optimal Transport cost between measures μ and ν for a ground cost function c, respectively.
- W_p, $W_{p,\varepsilon}$, $S_{c,\varepsilon}$, SW, GW: Wasserstein distance, Entropic Wasserstein distance, Sinkhorn divergence, Sliced-Wasserstein distance and Gromov-Wasserstein distance, respectively.
- KL: Kullback-Leibler divergence.

Chapter 1

Introduction

Optimal Transport (OT) is a field of Mathematics that studies the problem of transporting "mass" from one configuration to another while minimizing the cost of transportation. The problem is usually posed as the following: given two probability measures $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ and a cost function $c : X \times Y \to [0, +\infty]$, solve:

$$\inf \left\{ \int_{X \times Y} c(x, y) d\gamma \ : \ \gamma \in \Pi(\mu, \nu) \right\}, \tag{1.1}$$

where $\gamma \in \Pi(\mu, \nu)$ implies that $\gamma \in \mathcal{P}(X, Y)$ such that the marginal distribution of γ with respect to X is equal to μ and the marginal with respect to Y is equal to ν.

From this seemingly simple problem, the field of Optimal Transport has expanded immensely, and recent advances in both the theory and computational methods have lead to new possibilities for applying it to Machine Learning (ML).

1.1 Context and Motivation

One of the main uses of Optimal Transport on Machine Learning consists in measuring the distance between probability distributions with the so called Wasserstein distance. The Wasserstein distance is the minium transport cost between these two probability distributions when the cost of transportation is given by a distance metric[1]. This distance has the interesting property

[1]This is only an informal definition. A more correct definition is (2.17).

of preserving the ground geometry of where the probability distributions are defined, hence, even when the distributions don't share the same support, a meaningful distance can still be obtained.

One of the main problems of using such metric in Machine Learning comes from the fact that solving an OT problem can be difficult. For example, to compute the Wasserstein distance between empirical distributions with n samples has $O(n^3)$ complexity [28]. Since ML algorithms are usually applied on high-dimensional scenarios with large amounts of data, the use of OT had been dormant, until new and more efficient methods started to emerge. Since then, the use of OT with ML has been gaining traction in academia, as can be seen in Figure 1.1.

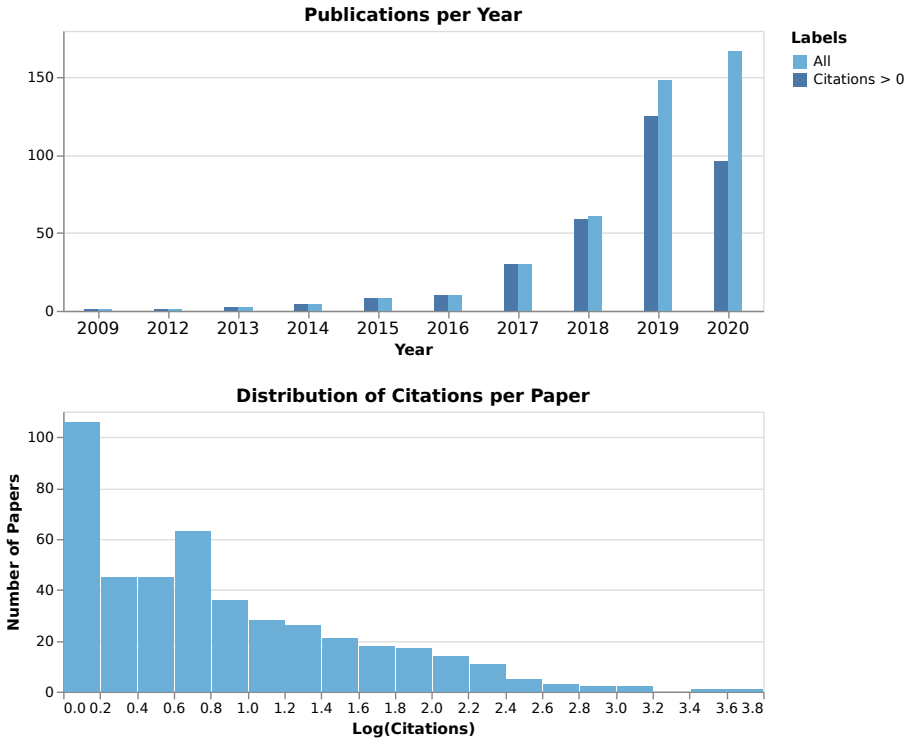

Figure 1.1: On the top, the evolution of the number of publications through the years, revealing a growing trend. On the bottom, the distribution of the number of citations, suggesting the relevance of many of these publications.

The use of OT has impacted several areas in Machine Learning, such as Transfer Learning, Clustering, Dimensionality Reduction, Generative Mod-

els, and others. Some of these applications have shown promising results, up to the point of becoming standard implementations in the field, as the case with the Wasserstein Generative Adversarial Networks (WGAN). Despite such steep growth in research, still no comprehensive review has been done highlighting the different ways in which ML and OT have been used together.

1.2 Objectives

This dissertation main goal is to provide detailed review of the applications of Optimal Transport to Machine Learning, examining the connections that have been drawn between these two subjects, and providing a clear view of the advantages and challenges. Thus, helping Machine Learning researchers to better understand how OT has impacted their field and how it can be used to improve and devise ML algorithms. To achieve this goal, the following objectives must be met:

- Introduce the theory of Optimal Transport necessary to understand how and when it can be appropriately utilized;
- Present the main computational methods that have been used to solve OT problems and which allowed the use of OT with Machine Learning;
- Review the literature on Optimal Transport for Machine Learning and highlight the main contributions in the different areas of ML.

1.3 Methodology

Snyder [81] defines three different approaches to conduct a literature review:

- *Systematic* - Mostly used in the medical sciences, this approach serves as a way to synthesize research findings, providing a reproducible way of conducting the review. Hence, it is useful for comparing different studies and making objective claims regarding a scientific field (e.g. "most of the scientific literature in the topic agrees that ...");
- *Semi-systematic* - This approach, also called *Narrative review*, might be used when the literature regarding a topic is just too broad for one to

perform a systematic review. Also, this method is mostly used when the goal is to provide an overview of a research topic, highlighting relevant findings, instead of objectively comparing research results across the literature;

- *Integrative* - The aim here is to analyze, synthesize and critique the literature in the topic of interest, in such a way as to generate new knowledge, by, for example, proposing new frameworks or developing new perspectives. In this approach, there need not be a systematic way of collecting information, thus allowing for a more "creative" collection methodology.

For this dissertation, we will be using the semi-systematic approach, since our aim is to provide an overview of the recent advances regarding Optimal Transport and Machine Learning, and not to compare the performance of the different algorithms. Therefore, we establish a methodology for collecting and filtering the data, but leave some leeway to add interesting research that might not be covered under our systematic collection.

1.3.1 Collection Methodology

The relevant literature was collected using the Microsoft Academic platform [1]. It was noted that the search results obtained through Microsoft Academic, were very similar to the ones obtained using other platforms such as Google Scholar and Dimensionality.ai, hence, only Microsoft Academic was used. The data for this dissertation (e.g. papers, authors, publications, number of citations) are from March 4th of 2021.

Three different search terms were used:

- "Optimal Transport" "Machine Learning";
- "Wasserstein" "Machine Learning";
- "Wasserstein" "GAN";

Two different search methods were used. The first one consisted in using the search engine of the platform without any filtering, while the other consisted of searching results only on selected publications. The reason for this is that filtering by publications allows for a more assertive search, while

searching without filtering allows for more broad results, which might still be relevant.

The publications used were:

- "Neural Information Processing Systems"(NeurIps);
- "International Conference on Machine Learning" (ICML);
- "International Conference on Artificial Intelligence and Statistics" (AISTATS);
- "International Conference on Learning Representations" (ICLR);
- "Journal of Machine Learning Research" (JMLR);
- "arXiv: Learning";
- "arXiv: Machine Learning".

Note that besides the Machine Learning conferences and journals, two arXiv publications were used. This was done in order to obtain relevant research that might have not yet been published.

1.3.2 Selection Methodology

A total of 444 different papers were originally collected using the methodology specified in the section above. This initial collection was intentionally broad, and another methodology was applied for selecting the works to be included in the review. This selection methodology consisted in two main criteria, namely: impact on the field and thematic relevance.

The impact on the field consisted in evaluating where the paper was published and number of citations[2]. Papers published in conferences such as "NeurIPS" were deemed more relevant in comparison to papers published on arXiv.

The thematic relevance consisted in evaluating how much the content of each paper was aligned with the goals of this review. Hence, the papers were classified in four main categories:

[2]Number of citations was obtained from Microsoft Academic.

1. **Algorithms for ML using OT**- This category is the main interest in this review. It consists of the papers that introduce new algorithms for Machine Learning using Optimal Transport. One prominent example in this category is the paper "Wasserstein GAN" by Arjovsky et al. [5];
2. **Computational Optimal Transport** - This is comprised of papers dealing with computational methods for solving Optimal Transport problems. One of the main challenges in using OT for Machine Learning is the computational efficiency. Hence, developing efficient methods for solving OT problems is necessary if one wants to develop Machine Learning algorithms with OT. In this category, the book Computational Optimal Transport by Peyré et al. [61] is perhaps the most important example;
3. **Theoretical Optimal Transport and Machine Learning** - This comprise papers dealing with theoretical aspects such as proof of learning bounds using Optimal Transport inequalities, reformulation of Machine Learning problems as OT problems, etc. Examples in this category consists of papers such as "A geometric view of optimal transportation and generative model" by Lei et al. [48] and "On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport" by Chizat and Bach [16];
4. **Applications** - This is comprised of papers focused on specific applications of Machine Learning algorithms that use OT. An example in this category is the paper "Low-Dose CT Image Denoising Using a Generative Adversarial Network With Wasserstein Distance and Perceptual Loss" by Yang et al. [98].

Since the goal of this review is to highlight the different usages of OT for Machine Learning applications, we restricted our review to papers of the first category. Some papers of the second category are present in Chapter 3, where we introduce the computational aspects of Optimal Transport. But our goal was merely to explain some of the main methods for solving OT that are used in the Machine Learning applications. Also, as noted above, a book [61] has recently been published on the subject.

Although we only focus on the category of "Algorithms for ML with OT", this category is also the one with the most papers, and it has a relevant number of citations per paper[3], as shown in Figure 1.2.

[3]Category "Algorithms for ML with OT" has a median of 4 citations per paper.

From the 200 papers in this category, a total of 70 were selected for the literature review. As can be seen in Table 1.1, the selected papers covered 94% of the total number of citations. Some Machine Learning subcategories, such as "Privacy Learning" and "Representation Learning" ended up not having any papers selected, and for this reason they are not present in Chapter 4.

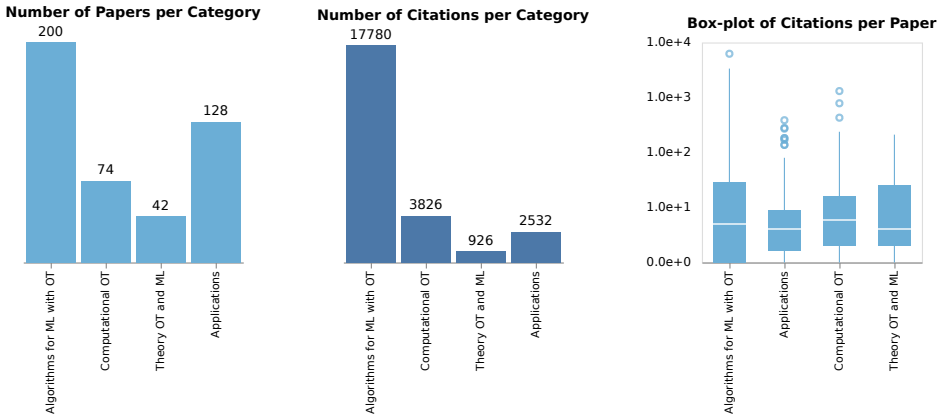

Figure 1.2: Visualization showing the number of papers and citations per paper category, and a box-plot of the distribution of the number of citations.

ML Category	**ML Subcategory**	**Papers**		**Citations**		**% Covered**	
		Selected	*Total*	*Selected*	*Total*	*Papers*	*Citations*
Semisupervised		**17**	**47**	**1.102**	**1.193**	**36%**	**92%**
	Reinforcement Learning	0	3	0	3	0%	0%
	Transfer Learning	17	44	1.102	1.190	39%	93%
Supervised		**23**	**42**	**2.781**	**2.935**	**55%**	**95%**
	Adversarial Robustness	3	12	421	460	25%	92%
	Classification	7	9	1.978	1.979	78%	100%
	Fairness Learning	2	5	37	83	40%	45%
	Privacy Learning	0	1	0	1	0%	0%
	Ranking	4	4	9	9	100%	100%
	Regression	7	11	336	403	64%	83%
Unsupervised		**30**	**111**	**12.819**	**13.652**	**27%**	**94%**
	Generative Modelling	22	60	1.2286	12.388	37%	99%
	Representation Learning	8	51	533	1.264	16%	42%
Total		**70**	**200**	**16.702**	**17.780**	**35%**	**94%**

Table 1.1: Comparison between collected versus selected papers under category "Algorithms for ML with OT".

1.3.3 A Taxonomy of Optimal Transport for Machine Learning

Papers in the category "Algorithms for ML using OT" were classified both according o their Machine Learning taxonomy (Figure 1.3) and the way Optimal Transport was utilized (Figure 1.4). Note that these two are parallel classifications, as ML algorithms in the same classification may use OT differently.

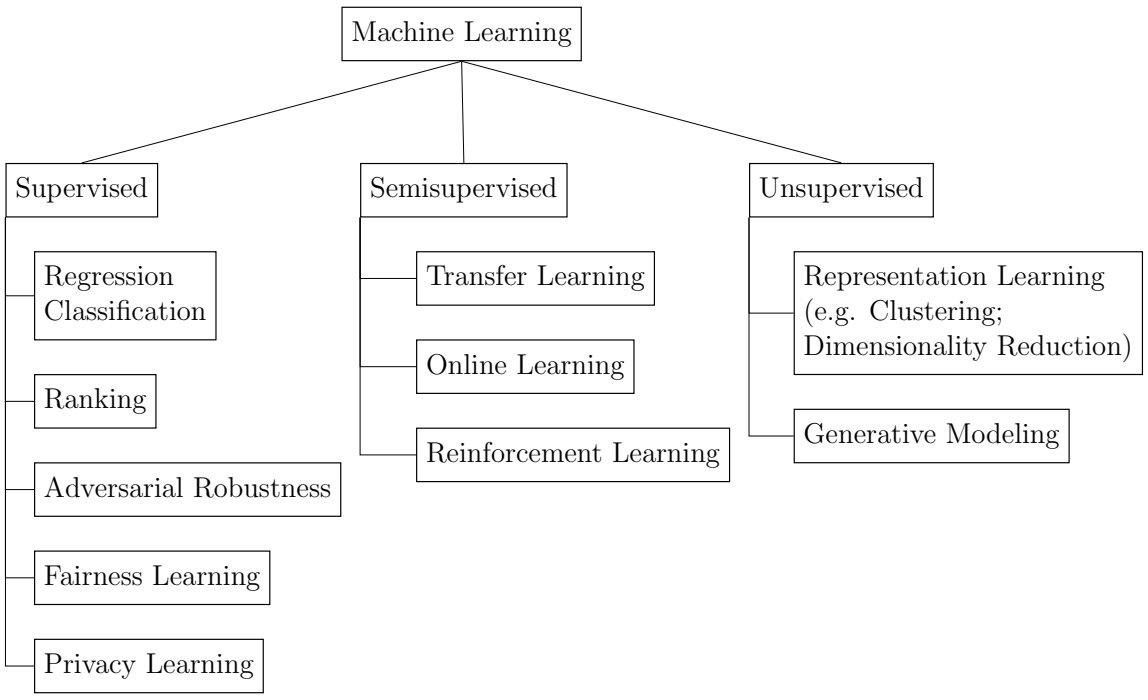

Figure 1.3: Taxonomy for Machine Learning Algorithms according to learning paradigm.

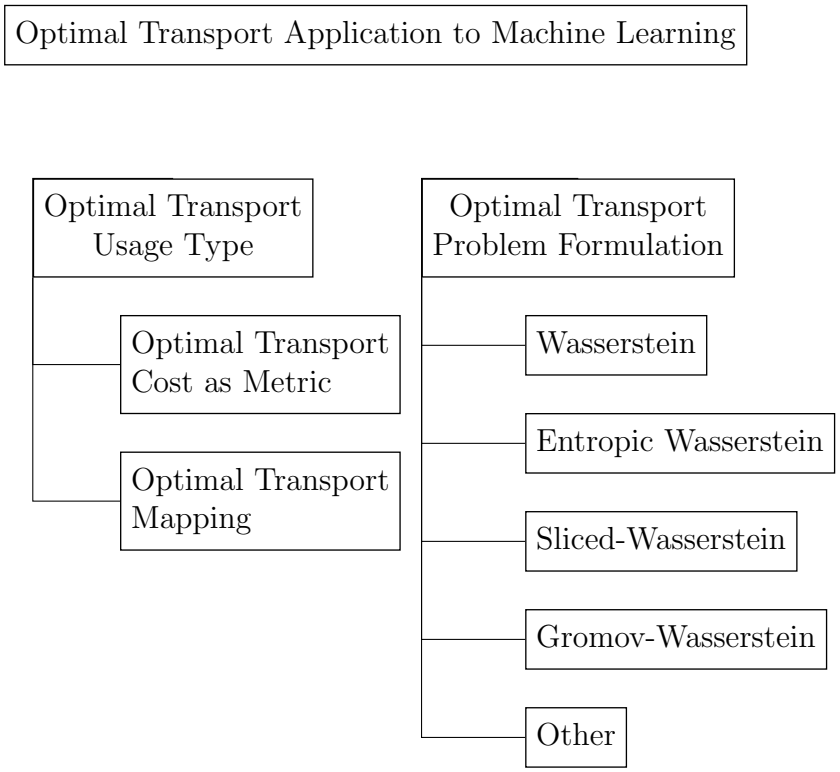

Figure 1.4: Taxonomy for the different ways that OT is used in Machine Learning.

As shown above in Figure 1.4, the usage of Optimal Transport for Machine Learning has two parallel main subdivisions.

The category **Optimal Transport Usage Type** classifies the Machine Learning applications according to the way they utilize OT. Applications usually make use of either the optimal transport plan or the optimal transport cost, we divide them in the following two subcategories:

- **Optimal Transport Cost as Metric** - Consists of those Machine Learning applications that make use of the optimal transport cost as a metric. Hence, this category contains applications such as Wasserstein Generative Adversarial Networks, in which the Wasserstein distance is used as a loss function where the output of the model is a probability distribution;

- **Optimal Transport Mapping** - Consists of those Machine Learning

applications that make use of the optimal transport plan. These transport plans are used, for example, to do dataset alignment in Transfer Learning by transporting the source dataset distribution to the target dataset distribution.

Optimal Transport Problem Formulation

The Optimal Transport Problem can be quite computationally costly to solve making it unusable on Machine Learning applications. Thus, many variations have been proposed, which modify the problem formulation making it easier to solve. We categorized the ML algorithms according the OT formulation they use. We propose the 5 following subcategories:

- **Wasserstein** - Uses the original OT problem formulation;
- **Entropic Wasserstein** - Ads an Entropic Regularization term to the Optimal Transport problem;
- **Sliced-Wasserstein** - Projects the probability distributions in 1-D and solves the OT problem;
- **Gromov-Wasserstein** - Modifies the OT problem in order to meaningfully compute a transport plan between different spaces (e.g. measures from $\mathbb{R}^1$ to $\mathbb{R}^2$);
- **Others** - All variations not present in the former subcategories;

Note that in the literature, it is common to refer to the Wasserstein distance as analogous to the minimum cost of an OT problem, but this is not 100% accurate. The actual definition of the Wasserstein distance requires that the cost function for the problem be given by $c(x, y) = d(x, y)^p$, where d is a metric (Definition 6.2.1) and $p \in [1, +\infty)$. In many situations, authors call their discrepancy function "Wasserstein distance", even though this would not be accurate. In the classification above, we do the same "semantic overloading" with the term, hence, we classify applications as "Wasserstein" even if the cost function is not of the form $d(x, y)^p$.

Finally, Figure 1.5 shows the distribution of papers based on ML task and the OT usage type. We observe that the use of OT on Machine Learning is most prevalent for Unsupervised Learning and the use of "OT as metric" is

also more common, with the "OT Mapping" more concentrated on Transfer Learning.

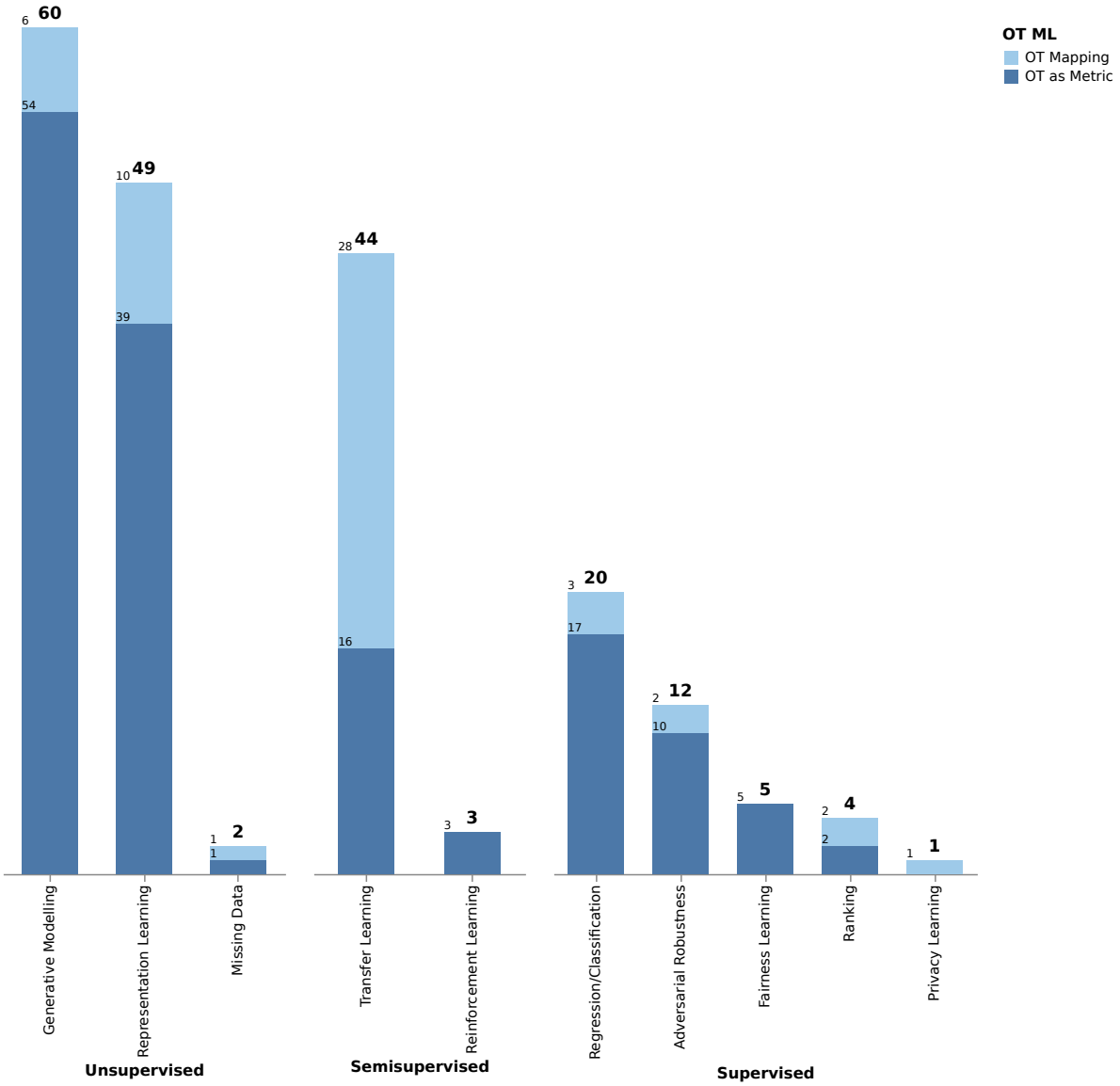

Figure 1.5: Distribution of all collected papers under "Algorithms for Machine Learning using Optimal Transport".

1.3.4 Structure of the Dissertation

This dissertation is composed of three main chapters, the theory of Optimal Transport, the computational methods for solving Optimal Transport problems, and the applications to Machine Learning.

The first chapter correspond to a brief, yet dense, introduction to the mathematical theory of Optimal Transport. The focus is on establishing some fundamental results regarding OT such as existence conditions, the relation of the Monge Problem and the Kantorovich Problem, Duality theorems and properties of the Wasserstein distance. The results presented are fundamental for a clear understanding of the subject, and cover most of the theoretical results and definitions usually present in the literature of Optimal Transport applied to Machine Learning. Most of the results presented are proved in the Appendix, with Santambrogio [72] serving as the main source for the chapter.

The second chapter focuses on presenting the computational methods used for solving Optimal Transport problems. This is necessary due to the fact that we need to solve OT problems if we wish to use OT in our Machine Learning algorithms. Hence, we explain some of the main methods used when coupling OT with ML. Besides these methods, we also present some variations to the Wasserstein distance that are commonly used in the literature, such as Sliced-Wasserstein and Gromov-Wasserstein. Peyré et al. [61] is the main reference in this topic, and we recommend it for any reader interested in a more complete exposition of Computational Optimal Transport.

Finally, the third chapter constitutes the actual survey of the many applications of Optimal Transport for Machine Learning. Since our focus is on ML researchers, we divided the survey according to the Machine Learning categories. The goal is to be comprehensive, covering the most relevant works in each area. Much of the literature share common concepts, hence, we expanded the explanation on some of the most relevant papers in order to clarify the applications as a whole. Readers already comfortable with Optimal Transport theory and computation may want to skip Chapters 2 and 3, and go straight to Chapter 4.

Chapter 2

Optimal Transport Theory

The field of Optimal Transport has grown quite substantially in recent years[1], and going through the theory in order understand how it applies to Machine Learning can be a challenging task for ML researchers not acquainted with the field. Hence, we have filtered the main theoretical results necessary for understanding the applications of Optimal Transport to Machine Learning presented in this dissertation.

This section is mainly based on the book "Optimal Transport for Applied Mathematicians" by Santambrogio [72].

2.1 A Brief Introduction to Optimal Transport

Before delving into formal definitions, theorems and proofs, let's give an informal overview of what is Optimal Transport, what are the main results we are interested in and how they relate to Machine Learning applications.

The main subject of study of Optimal transport theory is the problem of optimally transporting quantities from one configuration to another given a cost function. Although it may seem like a very narrow subject, this seemingly simple problem has a plethora of variations and can be significantly hard not only to solve, but to even prove that a solution exists.

The origin of the field of Optimal Transport is usually attributed to Gaspard Monge (1746-1818), a French mathematician, who was interested in the

[1] Villani [90] is roughly a thousand pages of theoretical results on OT.

problem of "what is the optimal way to transport soil extracted from one location and move to another where it will be used, for example, on a construction?"[2][90]. Monge studied this problem restricting the transportation assignment to deterministic maps, i.e. the soil extracted from location x should be moved entirely to an specific location y (see Figure 2.1), a condition that is known as "non-mass splitting". Monge also considered that the cost of transportation was proportional to the distance traveled (i.e. $c(x, y) = |x - y|$), but different cost functions can be used.

Although it has been considered the founding problem of Optimal Transport, the Monge Problem is not actually the most common formulation when it comes to applications in Machine Learning. The formulation most used when referring to the Optimal Transport problem is actually due to Leonid Kantorovich (1912-1986), a Russian mathematician. Kantorovich proposed a relaxation of the non-mass splitting condition, such that the optimal transportation solution could now transport the mass "excavated" from x to many locations (see Figure 2.2).

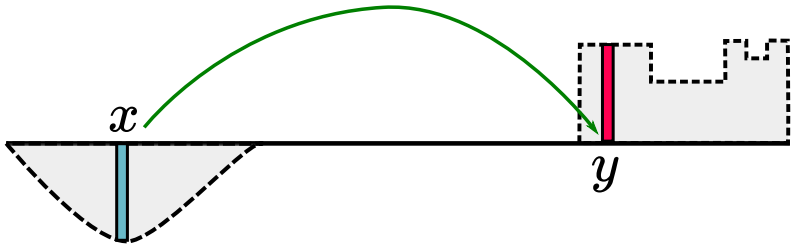

Figure 2.1: The figure illustrates the original Monge Problem, where all the mass is excavated from location x is transported to a deterministic location y. The transport assignment map is represented by the arrow in green.

The transportation assignment that solves the Monge Problem is called the Optimal Transport **map**, while the solution to the Kantorovich Problem is called the Optimal Transport **plan**. As we will show in the following sections, if the Monge Problem has a solution so does the Kantorovich Problem, but the contrary is not always true. From here on out, every time we refer to the OT problem, we'll be implicitly referring to the Kantorovich formulation, unless stated otherwise.

[2]This is not a quote from Monge.

Although the original OT problem is about soil excavation, we can apply it to abstract mathematical objects such as probability distributions. Consider two 1-dimensional probability distributions μ and ν, and define an Optimal Transport problem where the objective is to transport distribution μ to ν with $c(x, y) = |x - y|^p$ for $p \in [1, +\infty)$. Note that, if the OT problem has a solution, then there exists a minimum total cost. This minimum cost of transporting μ to ν is known as the Wasserstein distance ($W_p(\mu, \nu)$). The use of the Wasserstein distance to measure the discrepancy between probability distributions is one of the main applications of OT on Machine Learning.

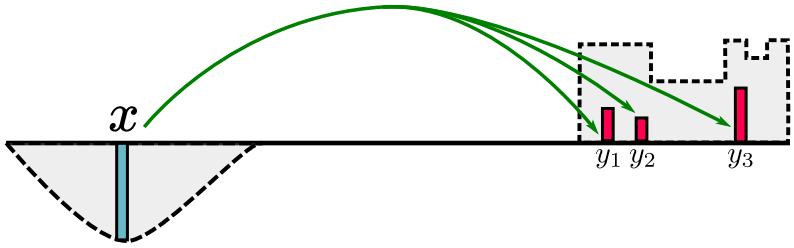

Figure 2.2: The figure illustrates the Optimal Transport Problem with the Kantorovich relaxation. The transportation assignment now can split the mass in blue, transporting it to many positions.

If we want to use the Wasserstein distance, then many questions have to be answered:

- Does the transport plan exist?
- If the transport plan exists, how does one obtain it and then calculates the Wasserstein distance?
- If the Wasserstein distance between two probability distributions goes to zero, does this imply convergence in probability?

The field of Optimal Transport has addressed these types of questions, thus the importance of understanding the theory before using it on real applications.

We end this brief introduction to OT with a description of the contents addressed in each of the following sections:

(i) **Monge Problem** - We formally define the Monge Problem;

(ii) **Kantorovich Problem** - We formally define the Kantorovich Problem and the notion of *relaxation*. Then, we prove that under certain conditions the Kantorovich Problem is a relaxation of the Monge Problem;

(iii) **On the Existence of Transport Plans** - This section focuses on the existence of solutions to the Optimal Transport problem. We prove the existence for compact metric spaces with continuous cost functions, and leave the proof of the more general cases to the Appendix;

(iv) **Duality Results** - The Kantorovich Problem admits a dual formulation, which, under some conditions, yields the same optimal cost as the primal formulation (i.e. Strong Duality). This section focuses on formally introducing the Dual Problem and stating the conditions under which Strong Duality holds. We close the section with the celebrated Kantorovich-Rubinstein Duality Theorem, which is used in Machine Learning applications such as WGANs;

(v) **Wasserstein Distance** - We formally define the Wasserstein distance and present some of its properties. Next, we state topological results of the Wasserstein space. We end the section with comments on the properties of the Wasserstein distance and why it is useful to fields like Machine Learning.

2.2 Monge Problem

Let's start by providing some definitions that will be used throughout this section.

Definition 2.2.1. Given $(\Omega, \mathcal{F})$ where $\mathcal{F}$ is a σ-algebra, then, $\mu : \mathcal{F} \to [0, +\infty]$ is a measure if:

i) $\mu(\varnothing) = 0$

ii) $(A_n)_{n\in\mathbb{N}} \subset \mathcal{F}$ with $A_j \cap A_i = \varnothing$, $\forall i, j \in \mathbb{N} \implies \mu(\cup_{n\in\mathbb{N}} A_n) = \sum_{n\in\mathbb{N}} \mu(A_n)$

We say that μ is a probability measure if besides the two properties above, we also have $\mu(\Omega) = 1$.

Definition 2.2.2. We call $\mathcal{P}(X)$ the space of probability measures defined on $(X, \mathcal{F})$, where the σ-algebra $\mathcal{F}$ is implicit and usually refers to the Borel σ-algebra.

Definition 2.2.3. (Pushforward) Let $(X, \mathcal{F})$ and $(Y, \mathcal{G})$ be measurable spaces, $T : X \to Y$ a $\mathcal{F}/\mathcal{G}$-measurable map and $\mu \in \mathcal{P}(X)$. We call $T_{\#}\mu$ the push-forward of μ, where:

$$T_{\#}\mu(B) = \mu(T^{-1}(B)), \quad \forall B \in \mathcal{G} \tag{2.1}$$

With these definitions, we can state the so called Monge Problem, which is known as the motivating problem that gave birth to the field of Optimal Transport.

Definition 2.2.4. (Monge Problem) Given two probability measures $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ and a cost function $c : X \times Y \to [0, +\infty]$, solve:

$$(MP) \qquad \inf \left\{ \int_X c(x, T(x)) d\mu \quad : \quad T_{\#}\mu = \nu \right\} \tag{2.2}$$

In the Monge Problem, no mass can be split. Therefore, one can easily come up with situations in which there is no solution to the problem, as shown in 2.3. A viable solution T to MP is called a **Transport Map**.

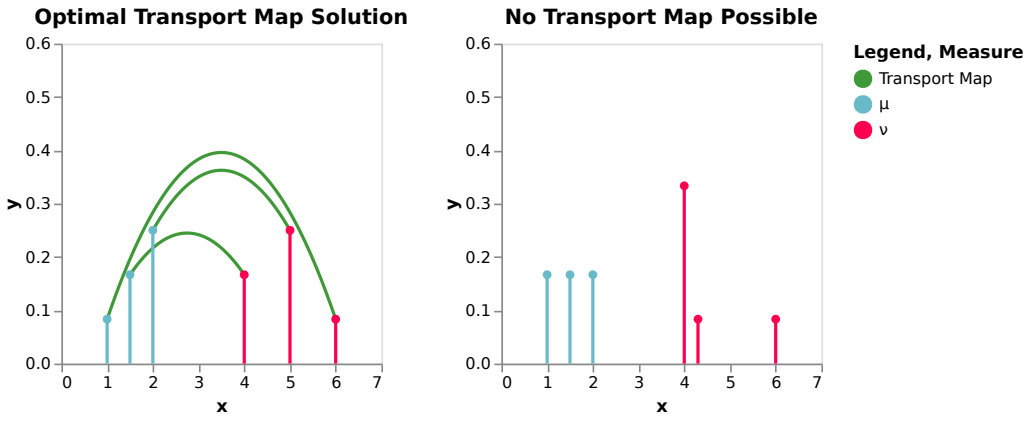

Figure 2.3: Example of two Optimal Transport Problems. On the left, there exists an optimal transport plan, while on the right there is no possible solution.

2.3 Kantorovich Problem

The Monge Problem is hard to solve due to its constraint on T which is not closed under weak convergence [72], and it might not have a solution even for "simple" probability distributions (e.g. Figure 2.3). Because of these limitations, we focus the relaxed version, the so called Kantorovich Problem. This relaxation consists of allowing mass to be split, thus making the set of possible solutions larger. Before stating the Kantorovich Problem, let's introduce some more definitions.

Definition 2.3.1. (Projection and Marginal) Let $\gamma \in \mathcal{P}(X \times Y)$ and $\pi_X : X \times Y \to X$ such that $\pi_X(x, y) = x, \forall (x, y) \in X \times Y$. Hence, we say that π_X is the projection operator on X. We then call $(\pi_X)_{\#}\gamma = \mu$ the marginal distribution of γ with respect to X.

Equivalently, if for every measurable set $A \subset X$, we have $\gamma(A \times Y) = \mu(A)$, then μ is the marginal of γ with respect to X.

Definition 2.3.2. (Coupling) Let (X, μ) and (Y, ν) be probability spaces. For $\gamma \in \mathcal{P}(X \times Y)$, we say that γ is a coupling of (μ, ν) if $(\pi_X)_{\#}\gamma = \mu$ and $(\pi_Y)_{\#}\gamma = \nu$. Also, we call $\Pi(\mu, \nu)$ the set of **Transport Plans**:

$$\Pi(\mu, \nu) := \{\gamma \in \mathcal{P}(X \times Y) \ : \ (\pi_X)_{\#}\gamma = \mu \quad \text{and} \quad (\pi_Y)_{\#}\gamma = \nu\} \tag{2.3}$$

Finally, we can state the Kantorovich Problem.

Definition 2.3.3. (Kantorovich Problem) Given two probability measures $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ and a cost function $c : X \times Y \to [0, +\infty]$, solve:

$$(KP) \qquad \inf \left\{ \int_{X \times Y} c(x, y) d\gamma \ : \ \gamma \in \Pi(\mu, \nu) \right\} \tag{2.4}$$

One can prove that indeed every time the Monge Problem has a solution, so will the Kantorovich Problem. More than that, the minimal cost of both problems will indeed coincide. Note that when the Monge Problem has a solution $T : X \to Y$, then $\gamma = (id, T)_{\#}\mu$ is a solution to the Kantorovich Problem.

We stated in the beginning of this section that (KP) was a relaxed version of (MP). Let's now formalize this concept.

Definition 2.3.4. (Lower Semi-Continuity) A function $f : X \to \mathbb{R}$ is lower semi-continuous (l.s.c) if

$$\forall x \in X, \ f(x) \leq \liminf_{x_n \to x} f(x_n) \tag{2.5}$$

Definition 2.3.5. (Relaxation) Given a metric space X and functional $F : X \to \mathbb{R} \cup \{+\infty\}$ bounded below. We call $\bar{F} : X \to \mathbb{R} \cup \{+\infty\}$ a of relaxation of F if:

$$\bar{F}(x) := \inf \left\{ \liminf_{n} F(x_n) \ : \ x_n \to x \right\} \tag{2.6}$$

Hence, $\bar{F}$ is the maximal functional G where G is lower semi-continuous and $G \leq F$.

Figure 2.4 illustrates an example of a relaxation. As a consequence of this definition, $\inf_x F = \inf_x \bar{F}$. Therefore, if we can prove that Kantorovich Problem is a relaxation of the Monge Problem, we would get that inf (KP) = inf (MP).

To prove that indeed (KP) is a relaxation of (MP) under some conditions, we use the following theorem, for which the complete proof can be found on Santambrogio [72].

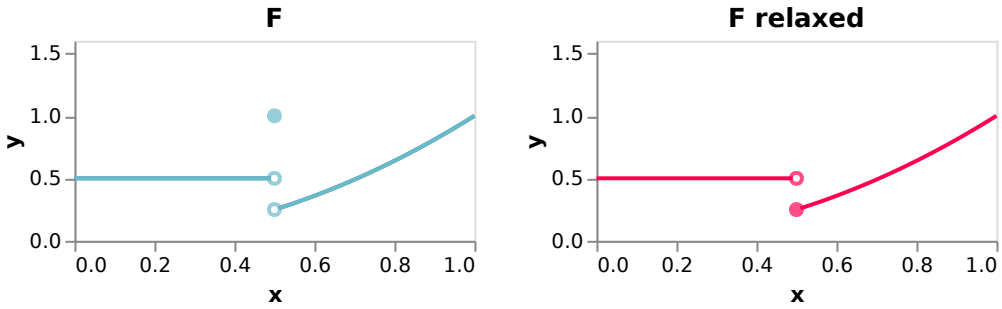

Figure 2.4: Example of a function F and it's relaxation.

Theorem 2.3.1. *(Santambrogio 1.32) Let $\Omega \subset \mathbb{R}^d$ compact, with $c : \Omega \times \Omega :\to [0, +\infty]$ continuous and $\mu \in \mathcal{P}(\Omega)$ atomless (i.e., for every $x \in \Omega$, we have $\mu(\{x\}) = 0$). Then, the set of plans $\gamma_T = (id, T)_{\#}\mu$ induced by the map T is dense in $\Pi(\mu, \nu)$.*

We can now prove the following:

Theorem 2.3.2. *For $\Omega \subset \mathbb{R}^d$ compact, $c : \Omega \times \Omega :\to [0, +\infty]$ continuous and $\mu \in \mathcal{P}(\Omega)$ atomless. Then, (KP) is a relaxation of (MP).*

Proof. First, let's restate the Monge Problem as

$$\inf\{J(\gamma) \ : \ \gamma \in \Pi(\mu, \nu)\}$$

Where, $\gamma_T := (id, T)_{\#}\mu$, and

$$J(\gamma) = \begin{cases} K(\gamma) = \int_\Omega c(x, T(x))\ d\mu = \int_{\Omega\times\Omega} c\ d\gamma_T, & \text{if } \gamma = \gamma_T \\ +\infty & \text{otherwise} \end{cases}$$

Note that indeed minimizing J is equal to minimizing the Monge Problem, since we only consider the transport plans γ_T that coincide with the cost when using a transport map T.

For $K(\gamma) = \int_{\Omega\times\Omega} c\ d\gamma$, we can show that K is continuous with respect to weak convergence (see 6.2.2), since

$$\gamma_n \rightharpoonup \gamma \iff \forall f \text{ continuous}, \int f d\gamma_n \to \int f d\gamma \implies$$

$$\implies K(\gamma_n) = \int_{\Omega\times\Omega} c\ d\gamma_n \to K(\gamma), \text{ for } c \text{ continuous.}$$

Also, by the definition of J, for any $\gamma \in \Pi(\mu, \nu)$, then $K(\gamma) \leq J(\gamma)$.

By Theorem 2.3.1, for any $\gamma \in \Pi(\mu, \nu)$ we can create a sequence of $\gamma_{T_n} \rightharpoonup \gamma$. And by the continuity of K with respect to weak convergence, we have that $J(\gamma_{T_n}) = K(\gamma_{T_n}) \to K(\gamma)$. Therefore:

$$\forall \gamma \in \Pi(\mu, \nu), \exists (\gamma_{T_n}) \ : \ \liminf_{n \to +\infty} J(\gamma_{T_n}) = K(\gamma)$$

Hence,

$$\inf\{\liminf_{n \to +\infty} J(\gamma_n) \ : \ \gamma_n \rightharpoonup \gamma\} \leq K(\gamma) \leq J(\gamma)$$

We can conclude that

$$\inf\{\liminf_{n \to +\infty} J(\gamma_n) \ : \ \gamma_n \rightharpoonup \gamma\} = K(\gamma)$$

□

2.4 On the Existence of Transport Plans

As stated before, it is not trivial to know when the Monge Problem indeed has a solution. It is easier to work with the Kantorovich Problem. In this section we present some results that relate to the existence of Optimal Transport Plans for the Kantorovich Problem. We start with more restrictive conditions and move to the more general case.

Theorem 2.4.1. *(Santambrogio 1.4) Let X and Y be compact metric spaces. Given $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ and $c : X \times Y \to [0, +\infty]$, if c is continuous, then (KP) admits a solution.*

Proof. We begin by using the notion of weak convergence to characterize continuity of functions defined on probability measures.

Note that since c is continuous and $(X \times Y)$ is compact, then c is continuous and bounded. Also, $K(\gamma) = \int_{X \times Y} c \, d\gamma$ is continuous with respect to weak convergence, since $\gamma_n \rightharpoonup \gamma$, if, and only if, for every f continuous and bounded function, $\int f \, d\gamma_n \to \int f \, d\gamma$.

Now, let's **show that $\Pi(\mu, \nu)$ is compact**. Take $\gamma_n \in \Pi(\mu, \nu)$. Note that γ_n is tight (6.2.3), because $(X \times Y)$ is compact. Then, by Prokhorov Theorem 6.2.3, $\exists \gamma_{n_k} \rightharpoonup \gamma$.

Take $\phi(x) \in C_b(X)$ and $\psi(y) \in C_b(Y)$. Therefore,

$$\int \phi(x)\ d\mu \underset{Cor.6.2.2}{=} \int \phi(x)\ d\gamma_{n_k} \to \int \phi(x)\ d\gamma$$
$$\int \psi(y)\ d\nu \underset{Cor.6.2.2}{=} \int \psi(y)\ d\gamma_{n_k} \to \int \psi(y)\ d\gamma$$

We conclude that $\gamma \in \Pi(\mu, \nu)$, which implies that $\Pi(\mu, \nu)$ is compact. Finally, since $K(\cdot)$ is continuous with respect to weak convergence and defined on a compact set, it attains a minimum. In other words, there exists a transport plan γ that minimizes the Kantorovich Problem.

□

Theorem 2.4.2. *(Santambrogio 1.5) Let X and Y be compact metric spaces. Given $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ and $c : X \times Y \to [0, +\infty]$, if c is lower semi-continuous, then (KP) admits a solution.*

Theorem 2.4.3. *(Santambrogio 1.7) Let X and Y be Polish (complete and separable) metric spaces. Given $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ and $c : X \times Y \to [0, +\infty]$, if c is lower semi-continuous then (KP) admits a solution.*

The proof for these two theorems can be found in the Appendix 6.1.

2.5 Duality of the Kantorovich Problem

We begin this section introducing the notion of the Dual formulation for the Kantorovich Problem, followed by the definition of c-transforms. Next, we state the main Duality Theorems without the proofs, which can be found in the Appendix 6.1. We close the section with the celebrated Kantorovich-Rubinstein Duality Theorem.

The Kantorovich Problem (2.3.3) is equivalent to:

$$\inf_{\gamma \in \mathcal{M}_+(X \times Y)} \int_{X \times Y} c(x, y) d\gamma + \sup_{(\phi, \psi) \in B} \left\{ \int_X \phi(x)\ d\mu + \int_Y \psi(y)\ d\nu - \int_{X \times Y} \phi(x) + \psi(y)\ d\gamma \right\} \quad (2.7)$$

Where $B := \{\phi \in C_b(X) \text{ and } \psi \in C_b(Y)\}$. The Dual Problem consists of exchanging the order of the inf and the sup, for the Kantorovich Problem

reformulated according: By Lemma 6.1.1, we can reformulate (KP) With (KP) reformulated, the Dual Problem consists of exchanging the order of the inf and the sup:

- **Primal**:

$$\inf_{\gamma\in\mathcal{M}_+(X\times Y)}\sup_{(\phi,\psi)\in B}\int_{X\times Y} c\,d\gamma+\int_X \phi\,d\mu+\int_Y \psi\,d\nu-\int_{X\times Y}\phi\oplus\psi\,d\gamma \quad (2.8)$$

- **Dual**:

$$\sup_{(\phi,\psi)\in B}\inf_{\gamma\in\mathcal{M}_+(X\times Y)}\int_{X\times Y} c\,d\gamma+\int_X \phi\,d\mu+\int_Y \psi\,d\nu-\int_{X\times Y}\phi\oplus\psi\,d\gamma \quad (2.9)$$

Note that in the Dual formulation, we can rewrite it as:

$$\sup_{(\phi,\psi)\in B}\int_X \phi\,d\mu+\int_Y \psi\,d\nu-\inf_{\gamma\in\mathcal{M}_+(X\times Y)}\int_{X\times Y} c-(\phi\oplus\psi)\,d\gamma \quad (2.10)$$

If there exists an A such that for all $\forall(x,y)\in A,\ \phi(x)+\psi(y)\geq c(x,y)$, then $\inf_\gamma \int c-(\phi\oplus\psi)\,d\gamma=-\infty$ since we can choose any $\gamma\in\mathcal{M}_+(X\times Y)$.

Therefore, we can formally state the Dual Problem as:

Definition 2.5.1. Given $\mu\in\mathcal{P}(X)$, $\nu\in\mathcal{P}(Y)$ and a cost $c:X\times Y\to\mathbb{R}_+$. The Dual Problem is given by

$$\text{(DP)}\quad \sup\left\{\int_X \phi\,d\mu+\int_Y \psi\,d\nu\ :\phi\in C_b(X)\ ,\psi\in C_b(Y)\ ,\ \phi\oplus\psi\leq c\right\} \quad (2.11)$$

We call **Weak Duality** if (DP) $\leq$ (KP), and we call **Strong Duality** if (DP) = (KP). When Strong Duality is true, the functions ϕ,ψ that maximize the Dual Problem are called the **Kantorovich Potentials**. One can easily prove that for (KP), the Weak Duality is always true (Lemma 6.1.2). The more interesting question is "When is Strong Duality true?".

Before stating the main theorems regarding Strong Duality, we must introduce the concept of c-transform.

Definition 2.5.2. (c-Transform) Given $f : X \to \overline{\mathbb{R}}$, and $c : X \times Y \to \overline{\mathbb{R}}$, the c-transform of f is:

$$f^c(y) := \inf_x c(x, y) - f(x) \tag{2.12}$$

Function f^c is also called the c-conjugate of f. Moreover, we say that f is c-concave if $\exists\ g : Y \to \overline{\mathbb{R}}$ such that $g^c(x) = f(x)$.

Note that the c-transform is a generalization of the Legendre-Fenchel transform, which is defined as:

$$f^*(y) := \sup_x x \cdot y - f(x) \tag{2.13}$$

Theorem 2.5.1. *For X and Y compact metric spaces, and $c : X \times Y \to \overline{\mathbb{R}}$ continuous. Then,* $\max(\mathrm{DP}) = \min(\mathrm{KP})$*, and DP admits a solution (ϕ, ϕ^c).*

Theorem 2.5.2. *For X and Y Polish spaces and $c : X \times Y \to \mathbb{R}$ uniformly continuous and bounded. Then, (DP) admits a solution (ϕ, ϕ^c) and* $\max(\mathrm{DP}) = \min(\mathrm{KP})$.

One cost that is of special interest is the quadratic cost $\frac{1}{2}|x - y|^2$. Note that this cost is neither bounded nor uniformly continuous for non-compact metric spaces. Hence, the previous theorems do not address it. But one can still prove that Strong Duality is true for such case.

Theorem 2.5.3. *(Santambrogio 1.40) Let $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$, with $c(x, y) = \frac{1}{2}|x - y|^2$. Suppose that $\int |x|^2 d\mu, \int |y|^2 d\nu < +\infty$[3]. Instead of the original Dual Problem, consider the following formulation:*

$$(\mathrm{DP}') \quad \sup \left\{ \int_{\mathbb{R}^d} \phi \ d\mu + \int_{\mathbb{R}^d} \psi \ d\nu \ : \phi \in L^1(\mu)\ , \psi \in L^1(\nu)\ ,\ \phi \oplus \psi \leq c \right\} \tag{2.14}$$

Therefore, (DP') admits a solution (ϕ, ψ) and $\max(\mathrm{DP}') = \min(\mathrm{KP})$.

The most general result regarding Strong Duality is the following:

[3]This is Theorem 1.40 in Santambrogio [72], but note that there is a small typo in the book, where it states $\int |x|^2 dx, \int |y|^2 dy < +\infty$ instead of the correct $\int |x|^2 d\mu, \int |y|^2 d\nu < +\infty$.

Theorem 2.5.4. *(Santambrogio 1.42) For X and Y Polish spaces and $c : X \times Y \to \mathbb{R} \cup \{+\infty\}$ l.s.c and bounded from below. Then,* $\sup(\text{DP}) = \min(\text{KP})$. *Note that in this theorem, one cannot guarantee the existence of the (ϕ, ψ) that maximize the Dual Problem.*

Note that under the conditions of Theorem 2.5.4, one cannot guarantee the existence of the optimal Kantorovich Potentials.

If the cost $c(x, y)$ is actually a distance metric (Def. 6.2.1), then we can prove the following result:

Lemma 2.5.1. *Let X be a metric space, and $c : X \times X \to \mathbb{R}$, where c is a distance metric. Therefore, a function $f : X \to \mathbb{R}$ is c-concave if and only if it is Lipschitz continuous with a constant less than 1 with respect to the distance c. We call $Lip_1^{(c)}$ this set of Lipschitz functions with constant less than 1. Moreover, $f^c = -f$.*

Lastly, using Theorem 2.5.4 and Lemma 2.5.1, one obtains the famous Kantorovich-Rubinstein Duality:

Theorem 2.5.5. *(Kantorovich-Rubinstein)*

Let (X, d) be a Polish space with metric d, and cost function $c(x, y) = d(x, y)$. Then, for $\mu, \nu \in \mathcal{P}(X)$, the Kantorovich Problem is equivalent to

$$\sup \left\{ \int_X \phi \, d\mu - \int_X \phi \, d\nu \ : \phi \in Lip_1(X) \right\} \tag{2.15}$$

2.6 Wasserstein Distance

In this section we focus on how the minimal transport cost can be used as a distance metric in the space of probability measures.

Definition 2.6.1. (Probability space with p-Moments) Let (X, d) be a metric space with $p \in [1, +\infty)$.

$$\mathcal{P}_p(X) := \{\mu \in \mathcal{P}(X) : \int_{X \times X} d(x, y)^p \, d\mu(x) d\mu(y) < +\infty\} \tag{2.16}$$

Note that this is equivalent to the set of probability measures such that $\int_X d(x, x_0) \, d\mu < +\infty$ for every $x_0 \in X$. The proof of this statement can be found in Garling [30] Proposition 21.1.1.

Definition 2.6.2. (Wasserstein Distance)

Let (X, d) be a Polish metric space, with $c : X \times X \to \mathbb{R}$ such that $c(x, y) = d(x, y)^p$, and $p \in [1, +\infty)$. For $\mu, \nu \in \mathcal{P}_p(X)$, the Wasserstein Distance is given by:

$$W_p(\mu, \nu) := \left(\inf_{\gamma \in \Pi(\mu,\nu)} \int_{X \times X} d(x, y)^p \, d\gamma \right)^{1/p} \tag{2.17}$$

Note that the restriction to $\mu, \nu \in \mathcal{P}_p(X)$ is necessary for W_p to be a distance metric. Moreover, for $p = 1$, then $c(x, y) = d(x, y)$ is a metric on X, therefore, for X Polish, one can use Kantorovich-Rubinstein's Duality Theorem 2.5.5 to obtain:

$$W_1(\mu, \nu) = \sup_{\phi \in Lip_1} \int_X f d(\mu - \nu) \tag{2.18}$$

Proposition 2.6.1. *$W_p(\cdot, \cdot)$ is a metric on $\mathcal{P}_p(X)$.*

Definition 2.6.3. (Wasserstein Space) For a Polish space X, we call $\mathcal{P}_p(X)$ a Wasserstein space if it is endowed with the p-Wasserstein metric. Note that is also common to see this space symbolized by $\mathcal{W}_p(X)$.

Proposition 2.6.2. *For a bounded Polish space X, $p \in [1, +\infty)$, $\mu, \nu \in \mathcal{P}_p(X)$ and $M \in \mathbb{R}_+$, then*

$$W_1(\mu, \nu) \leq W_p(\mu, \nu) \leq M W_1(\mu, \nu)^{1/p} \tag{2.19}$$

Next, let's present some of the topological properties of such space.

Theorem 2.6.1. *Let (X, d) be a Polish compact space with $\mu_n, \mu \in P_p(X)$ and $p \in [1, +\infty)$, then $W_p(\mu_n, \mu) \to 0 \iff \mu_n \rightharpoonup \mu$.*

Theorem 2.6.2. *For (X, d) a Polish metric space, $\mu_n, \mu \in \mathcal{P}_p(X)$ and $x_0 \in X$. Then*

$$W_p(\mu_n, \mu) \to 0 \iff \int_X d(x, x_0)^p d\mu_n \to \int_X d(x, x_0)^p d\mu \text{ and } \mu_n \rightharpoonup \mu. \tag{2.20}$$

Let's just put some words on these last results we introduced. We showed that the p-Wasserstein distance metrizes weak convergence of probability measures in the space $\mathcal{P}_p(X)$, with (X, d) a Polish space. Such property is very useful and is not present in many other commonly used distances such as Total Variation and the Kullback-Leibler Divergence.

Yet, there are many other ways to metrize weak convergence, such as Prokhorov's distance and bounded Lipschitz distance. So, besides this *metrization*, Villani [90] gives the following reasons that make W_p such an interesting metric:

(i) Its definition makes it a natural choice in OT problems;

(ii) The distance has a rich duality, especially for $p = 1$;

(iii) Since it's defined with an infimum, it is easy to bound from above;

(iv) Wasserstein distances incorporate information of the ground geometry.

For applications in Data Science, the equivalence with weak convergence and the incorporation of the ground geometry are probably the most attractive characteristics. Figure 2.5 highlights how W_p takes into account the underlying geometry compared to the Kullback-Leibler divergence, which does not.

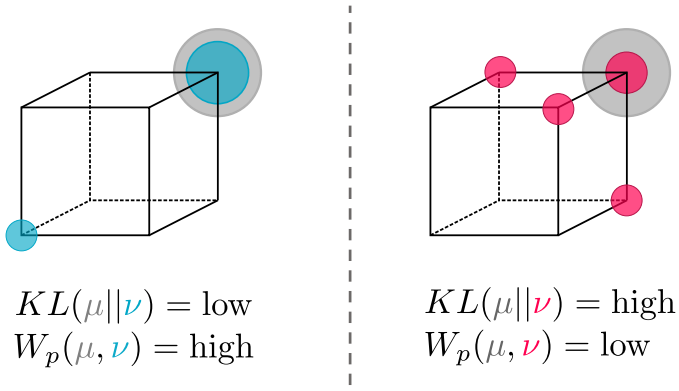

Figure 2.5: Comparison between Wasserstein distance and KL Divergence, based on Montavon et al. [56]. On the left, there is a large overlap between the two distributions, but a large geometrical distance for a portion. On the right, there is much less overlap, but the whole distribution is geometrically closer. These two cases clearly highlight how W_p incorporates geometrical information while KL does not.

Chapter 3

Optimal Transport Computational Aspects

In this section we briefly introduce some computational aspects of Optimal Transport. Most of the content of this section is based on Computational Optimal Transport: With Applications to Data Science by Peyré et al. [61]. Solving an Optimal Transport Problem can be quite challenging depending on the type of distributions and costs involved, hence, the search for efficient numerical methods is a quite active area of research.

3.1 Exact Solution of the Wasserstein Distance

3.1.1 One Dimensional Optimal Transport

For $\mu, \nu \in P_p(\mathbb{R})$, the Wasserstein has a closed form solution, which relies on the pseudoinverse of the cumulative distribution function.

Definition 3.1.1. Let $\mu \in \mathcal{P}(\mathbb{R})$. The cumulative distribution function (CDF) is

$$F_\mu(x) := \mu((-\infty, x]) \tag{3.1}$$

Note that F_μ is a nondecreasing and right-continuous function.

Definition 3.1.2. Given a nondecreasing and right-continuous function $F : \mathbb{R} \to [0, 1]$, its pseudoinverse is

$$F^{-1}(x) := \inf\{y \in \mathbb{R} \ : \ F(y) \geq x\} \tag{3.2}$$

After introducing these definitions, we can present the formula for computing the Wasserstein distance (Remark 2.30 on Peyré et al. [61]):

$$W_p(\mu,\nu)^p = \int_0^1 |F_\mu^{-1}(x) - F_\nu^{-1}(x)|^p dx \tag{3.3}$$

Note that for $p = 1$ and μ, ν both atomless, then there exists an optimal Monge map $T = F_\nu^{-1} \circ F_\mu$.

For the discrete 1-D distributions, an even simpler algorithm can be devised. Let $\mu = \sum_i^n = 1u_i\delta_{x_i}$ and $\nu = \sum_j^m = 1v_i\delta_{v_j}$, where $x_1 \leq x_2 \leq ... \leq x_n$ and $y_1 \leq y_2 \leq ... \leq y_m$. Consider that each position x_i has mass u_i and each position y_j has capacity v_j. The optimal transport plan consists of moving particle x_i to the closest position y_j, until capacity v_j is filled.

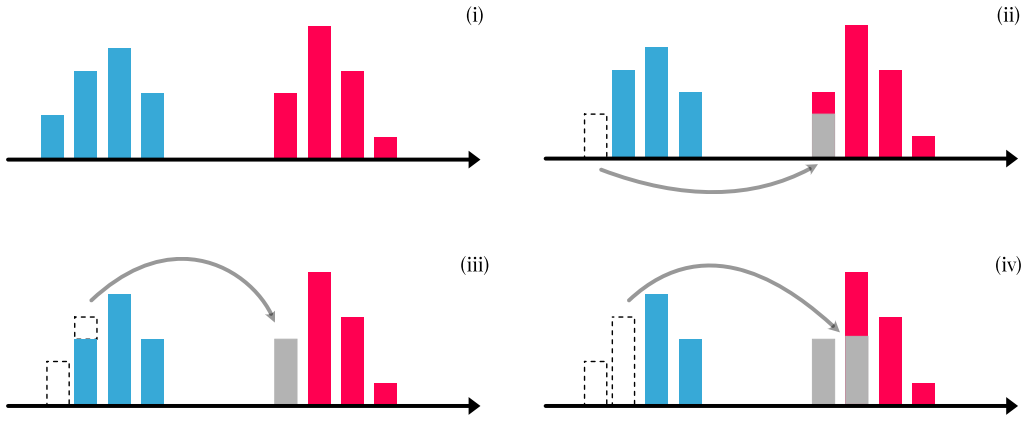

Figure 3.1: Illustration of the algorithm for optimally transporting distribution μ in blue to distribution ν in red.

3.1.2 Distance between Gaussians

The distance between Gaussian distributions also has a closed form. Let $\mu \in N(\boldsymbol{m}_\mu, \boldsymbol{\Sigma}_\mu)$ and $\nu \in N(\boldsymbol{m}_\nu, \boldsymbol{\Sigma}_\nu)$ defined on $\mathbb{R}^d$. Then, there exists an optimal map of the form:

$$T : x \to \boldsymbol{m}_\nu + A(x - \boldsymbol{m}_\mu) \tag{3.4}$$

with

$$A = \boldsymbol{\Sigma}_\mu^{-1/2} \left(\boldsymbol{\Sigma}_\mu^{1/2}\boldsymbol{\Sigma}_\nu\boldsymbol{\Sigma}_\mu^{1/2}\right)^{1/2} \boldsymbol{\Sigma}_\mu^{-1/2} \tag{3.5}$$

Using such map, one can then obtain a closed formula for the 2-Wasserstein distance:

$$W_2^2(\mu,\nu) = |\boldsymbol{m}_\mu - \boldsymbol{m}_\nu|^2 + tr\left(\boldsymbol{\Sigma}_\mu + \boldsymbol{\Sigma}_\nu - 2\left(\boldsymbol{\Sigma}_\mu^{1/2}\boldsymbol{\Sigma}_\nu\boldsymbol{\Sigma}_\mu^{1/2}\right)^{1/2}\right) \tag{3.6}$$

3.1.3 Transport Between Discrete Measures

Let μ be a finite discrete probability measure, hence

$$\mu := \sum_{i=1}^{n} u_i \delta_{x_i} \tag{3.7}$$

Where $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{R}^{n\times d}$ represent the location of each mass particle $i \in \{1, ..., n\}$. Vector $\mathbf{u} \in \mathbb{R}^{n\times 1}$, with components $u_i \in (0, 1]$, is the vector of weights, representing the amount of "mass" of each particle. Hence, discrete measures might be represented by a vector $\mathbf{x}$ of positions, and $\mathbf{u}$ of weights.

Now, suppose that both μ and ν are discrete measures. Let $\mathbf{u} \in \mathbb{R}^{n\times 1}$ and $\mathbf{v} \in \mathbb{R}^{m\times 1}$ represent the vector of weights, and $\mathbf{x} \in \mathbb{R}^{n\times d}, \mathbf{y} \in \mathbb{R}^{m\times d}$ represent the positions of each particle of μ and ν, respectively. In this scenario, the Optimal Transport Problem might be reformulated as the following. The cost function $c(x, y)$ can be substituted by a cost matrix $\mathbf{C} \in \mathbb{R}^{n\times m}$, where

$$\mathbf{C}_{i,j} := c(x_i, y_j), \quad i \in \{1, ..., n\},\ j \in \{1, ..., m\} \tag{3.8}$$

Any transport plan γ can be written as a matrix $\mathbf{P} \in \mathbb{R}_+^{n\times m}$, such that $\mathbf{P}_{i,j}$ represents the amount of mass flowing from particle i to particle j. Since $\gamma \in \Pi(\mu, \nu)$, the set of possible transport plans can be written as:

$$\mathbf{U}(\mathbf{u}, \mathbf{v}) := \left\{\mathbf{P} \in \mathbb{R}_+^{n\times m} \ :\ \mathbf{P}\mathbf{1}_m = \mathbf{u},\ \mathbf{P}^{\mathrm{T}}\mathbf{1}_n = \mathbf{v}\right\} \tag{3.9}$$

Where $\mathbf{1}_n$ is a vector with n components, each equal to 1. In words, the sum of each row of $\mathbf{P}$ must be equal to $\mathbf{u}$ and the sum of each column must be equal to $\mathbf{v}$.

The Kantorovich Problem can be written as:

$$\text{(KP-Disc.)} \quad \mathrm{L}_{\mathbf{C}}(\mathbf{u}, \mathbf{v}) := \min_{\mathbf{P}\in\mathbf{U}(\mathbf{u},\mathbf{v})} \langle \mathbf{C}, \mathbf{P}\rangle = \min_{\mathbf{P}\in\mathbf{U}(\mathbf{u},\mathbf{v})} \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{C}_{i,j}\mathbf{P}_{i,j} \tag{3.10}$$

The Dual Problem becomes:

$$\text{(DP-Disc.)} \qquad \mathrm{L}_{\mathbf{C}}(\mathbf{u},\mathbf{v}) := \max_{(\mathbf{f},\mathbf{g})\in\mathbf{R}(\mathbf{C})} \langle \mathbf{f},\mathbf{u}\rangle + \langle \mathbf{g},\mathbf{v}\rangle \tag{3.11}$$

Where

$$\mathbf{R}(\mathbf{C}) := \{(\mathbf{f},\mathbf{g}) \in \mathbb{R}^n \times \mathbf{R}^m \ : \ \forall (i,j) \in \{1,...,n\} \times \{1,...,m\}, \ \mathbf{f} \oplus \mathbf{g} \leq \mathbf{C}\} \tag{3.12}$$

The Discrete Optimal Transport Problem is actually a Linear Programming (LP) problem. Hence, one can rearrange Equation (3.10) to the standard form of LP.

Definition 3.1.3. (Optimal Transport as standard LP)

$$\begin{aligned} &\text{minimize} && \mathbf{c}^{\mathrm{T}}\mathbf{p} \\ &\text{subject to} && \mathbf{A}\mathbf{p} = \begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}, \\ & && \mathbf{p} \geq 0 \end{aligned}$$

Where

$$\mathbf{p} := \begin{bmatrix}\mathbf{P}_{1,1}\\ \mathbf{P}_{2,1}\\ \vdots \\ \mathbf{P}_{n,1}\\ \mathbf{P}_{2,1}\\ \vdots \\ \mathbf{P}_{n,m}\end{bmatrix}, \quad \mathbf{c} := \begin{bmatrix}\mathbf{C}_{1,1}\\ \mathbf{C}_{2,1}\\ \vdots \\ \mathbf{C}_{n,1}\\ \mathbf{C}_{2,1}\\ \vdots \\ \mathbf{C}_{n,m}\end{bmatrix}, \quad \mathbf{A} := \begin{bmatrix}\mathbf{1}_n^{\mathrm{T}} \otimes \mathbf{I}_m\\ \mathbf{I}_n \otimes \mathbf{1}_m^{\mathrm{T}}\end{bmatrix},$$

Note that $\mathbf{I}_n$ stands for the identity matrix, and $\otimes$ is the tensor product.

Definition 3.1.4. (Optimal Transport Dual Problem as LP)

$$\begin{aligned} &\text{minimize} && \begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}^{\mathrm{T}} \mathbf{h} \\ &\text{subject to} && \mathbf{A}^{\mathrm{T}}\mathbf{h} \leq \mathbf{c} \end{aligned}$$

Where $\mathbf{h} = [f_1, \ldots, f_n, g_1, \ldots, g_m]^{\mathrm{T}}$, with $\mathbf{c}$ and $\mathbf{A}$ the same as in the primal LP.

Since the Optimal Transport Problem is actually a Linear Programming Problem, therefore, one can use known methods for solving such problems, such as Simplex or Interior Point Method. An explanation on how to solve such LP problems in OT can be found in Chapter 3 of Peyré et al. [61].

3.1.4 Entropic Regularization

For measures $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$, one can approximately find the Wasserstein distance by discretizing the distributions, and solve the Linear Programming problem for discrete measures. The problem with such approach is that it can be quite costly, specially in Machine Learning scenarios, where high dimensionality is the norm. Hence, more efficient methods are necessary if one wants to use OT in Machine Learning.

One way of reducing the computational complexity of OT problems is to use the Entropic Regularization.

Definition 3.1.5. (Entropic Regularized KP - General Formulation)

Let $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$, $c : X \times Y \to \mathbb{R}$ and $\varepsilon > 0$. Then, the regularized problem is:

$$\text{(KP-ER)} \quad \overline{OT}_{c,\varepsilon}(\mu, \nu) := \inf_{\gamma \in \Pi(\mu,\nu)} \int_{X \times Y} c(x, y)\ d\gamma + \varepsilon \text{KL}(\gamma | \mu \otimes \nu) \tag{3.13}$$

Where $\mu \otimes \nu$ is the product measure and $\text{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence:

$$\text{KL}(\gamma || \mu \otimes \nu) := \int_{X \times Y} \log \left(\frac{d\gamma}{d(\mu \otimes \nu)} \right)\ d\gamma \tag{3.14}$$

Let γ^* be the optimal transport plan that solves the Entropic Regularized Kantorovich Problem, then

$$OT_{c,\varepsilon}(\mu, \nu) := \int_{X \times Y} c(x, y)\ d\gamma^* \tag{3.15}$$

is called Entropic Regularized OT distance (EROT), or Entropic Regularized Wasserstein distance, in case $c(x, y) = d(x, y)$. Note that, this distance is not an actual metric, since for $\varepsilon > 0$, it's possible that $OT_{c,\varepsilon}(\mu, \mu) > 0$. But by multiplying it by $\mathbb{1}_{\mu \neq \nu}$ one can show that it becomes a metric [51].

Before moving on, let's briefly clarify some point of confusion. In the literature, some authors use the use "Sinkhorn distance" to refer to function $OT_{c,\varepsilon}$, while others instead use "Sinkhorn distance" to refer to $\overline{OT}_{c,\varepsilon}$. Even more confusing, there is also another distance which is usually called Sinkhorn divergence, that actually refers to an unbiased version of the Entropic Regularization OT distance (3.22).

To keep things clear, we'll avoid using the term "Sinkhorn distance". Instead, we'll only use the term "Sinkhorn divergence" to refer to (3.22), Entropic Regularized OT *distance* for $OT_{c,\varepsilon}$ and Entropic Regularized OT *cost* for $\overline{OT}_{c,\varepsilon}$.

3.1.5 Entropic Regularization for Discrete OT

Although we just presented the general formulation, entropic regularization is most useful for the case of discrete measures. In such scenario, there exists efficient algorithms for solving the optimization problem.

Definition 3.1.6. (Entropic Regularized KP - Discrete Formulation)

Let μ and ν be discrete probability measures, with a cost matrix $\mathbf{C}$, transport plan matrix $\mathbf{P}$ and $\varepsilon > 0$.

$$\mathrm{L}_{\mathbf{C},\varepsilon}(\mathbf{u}, \mathbf{v}) := \min_{\mathbf{P} \in \mathbf{U}(\mathbf{u},\mathbf{v})} \langle \mathbf{C}, \mathbf{P} \rangle - \varepsilon \mathbf{H}(\mathbf{P}) \tag{3.16}$$

Where $\mathbf{H}(\cdot)$ is the discrete entropy [1]:

$$\mathbf{H}(\mathbf{P}) := -\sum_{i,j} \mathbf{P}_{i,j}(\log(\mathbf{P}_{i,j}) - 1) \tag{3.17}$$

Note that adding the regularization term to the problem makes it strongly ε-convex, smooth and less costly to solve. Also, as shown in Figure 3.2, the entropic regularization decreases the sparsity of the optimal plan.

Below, we present a proof that as the regularization term decreases, the minimum optimal transport cost approaches the real minimum cost.

Proposition 3.1.1. *(Peyré et al. [61] Proposition 4.1)*

[1] Note that this formula for entropy is slightly different from what is commonly used, since a 1 is subtracted. This formulation is taken from Peyré et al. [61].

Let $\mathbf{P}_\varepsilon$ be the solution to the regularized problem for discrete measures. Then,

$$\mathbf{P}_\varepsilon \xrightarrow{\varepsilon \to 0} \arg\min_{\mathbf{P}} \{-\mathbf{H}(\mathbf{P}) \ : \ \mathbf{P} \in \mathbf{U}(\mathbf{u}, \mathbf{v}), \ \langle \mathbf{C}, \mathbf{P} \rangle = \mathrm{L}_{\mathbf{C}}(\mathbf{u}, \mathbf{v})\} \tag{3.18}$$

This means that as ε goes to 0, then the solution of the regularized problem converges to the solution of the original Kantorovich Problem with the most entropy. Hence, $\mathrm{L}^{\varepsilon}_{\mathbf{C}}(\mathbf{u}, \mathbf{v}) \xrightarrow{\varepsilon \to 0} \mathrm{L}_{\mathbf{C}}(\mathbf{u}, \mathbf{v})$.

Proof. First, let $\mu \in \mathcal{P}(X)$ and $\nu \in \mathcal{P}(Y)$ be discrete measures. Moreover, the support of γ is compact, since $\gamma \in P(\{x_1, \ldots, x_n\} \times \{y_1, \ldots, y_m\})$. Therefore, $\mathbf{U}(\mathbf{u}, \mathbf{v})$ is compact. Now, take $\epsilon_n \to 0$, where $\mathbf{P}_n$ is the solution of the regularized problem. Since $(\mathbf{P}_n) \subset \mathbf{U}(\mathbf{u}, \mathbf{v})$, there there exists a subsequence $\mathbf{P}_{n_k} \to \mathbf{P}^*$

Now, let $\mathbf{P}$ be the optimal solution to the unregularized problem. Note that, for any ε_{n_k}

$$\langle \mathbf{C}, \mathbf{P} \rangle - \varepsilon_{n_k} \mathbf{H}(\mathbf{P}) \geq \langle \mathbf{C}, \mathbf{P}_{n_k} \rangle - \varepsilon_n \mathbf{H}(\mathbf{P}_{n_k})$$

$$\therefore$$

$$0 \leq \langle \mathbf{C}, \mathbf{P}_{n_k} \rangle - \langle \mathbf{C}, \mathbf{P} \rangle \leq \varepsilon_{n_k} (\mathbf{H}(\mathbf{P}_{n_k}) - \mathbf{H}(\mathbf{P}))$$

We know that $\mathbf{H}$ is continuous, which implies that $\mathbf{H}(\mathbf{P}_{n_k}) \to \mathbf{H}(\mathbf{P}^*)$, hence $\langle \mathbf{C}, \mathbf{P}^* \rangle = \langle \mathbf{C}, \mathbf{P} \rangle$ and $\mathbf{H}(\mathbf{P}) \leq \mathbf{H}(\mathbf{P}^*)$. Since $\mathbf{H}$ is strictly convex, then there is only one $\mathbf{P}^*$ that maximizes entropy, which means that any sequence $\varepsilon_n \to 0$ also converges to the same $\mathbf{P}^*$. □

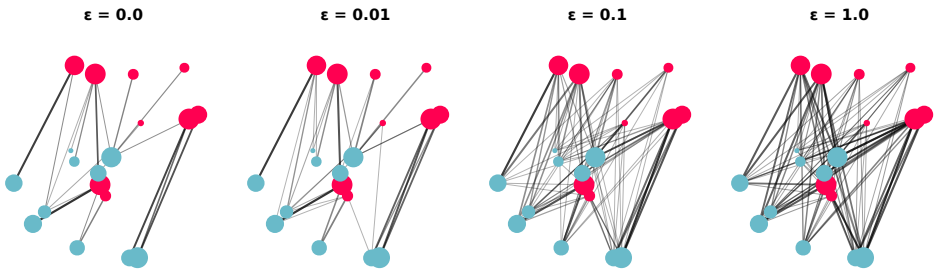

Figure 3.2: Impact of the regularization term ε in the optimal transport plan of discrete measures. The points in red and blue represent the discrete distributions, while the lines in black is the transport plan.

3.1.6 Sinkhorn Algorithm

One way to solve this discrete Entropic OT problem is with the Sinkhorn algorithm. This algorithm has become quite prominent after Cuturi [20] showed it's efficiency in solving OT problems, which is mainly due to its parallelization.

Before presenting the algorithm, we prove the following proposition:

Proposition 3.1.2. *(Peyré et al. [61] Proposition 4.3)*
The solution of (3.16) *is of the form*

$$\mathbf{P}_{i,j} = \mathbf{w}_i \mathbf{K}_{i,j} \mathbf{z}_j \tag{3.19}$$

Where $(\mathbf{w}, \mathbf{z}) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$ *are two scaling variables, and* $\mathbf{K}$ *is a Gibbs Kernel, i.e.* $\mathbf{K}_{i,j} := \exp(-\mathbf{C}_{i,j}/\varepsilon)$.

Proof. First, take the Lagrangian of the optimization problem, where $\mathbf{f}, \mathbf{g}$ are the dual variables:

$$\mathcal{L}(\mathbf{P}, \mathbf{f}, \mathbf{g}) = \langle \mathbf{P}, \mathbf{C} \rangle - \varepsilon \mathbf{H}(\mathbf{P}) - \langle \mathbf{f}, \mathbf{P}\mathbf{1}_m - \mathbf{u} \rangle - \langle \mathbf{g}, \mathbf{P}^{\mathrm{T}}\mathbf{1}_n - \mathbf{v} \rangle$$

Since this is a convex optimization problem, the optimal is where the gradient is null, then:

$$\frac{\partial \mathcal{L}(\mathbf{P}, \mathbf{f}, \mathbf{g})}{\partial \mathbf{P}_{i,j}} = \mathbf{C}_{i,j} + \varepsilon \log(\mathbf{P}_{i,j}) - \mathbf{f}_i - \mathbf{g}_j = 0 \iff \mathbf{P}_{i,j} = e^{\mathbf{f}_i/\varepsilon} e^{-\mathbf{C}_{i,j}/\varepsilon} e^{\mathbf{g}_j/\varepsilon}$$

Therefore, $\mathbf{w} = \mathbf{e}^{\mathbf{f}_i/\varepsilon}$ and $\mathbf{z} = \mathbf{e}^{\mathbf{g}_j/\varepsilon}$.

□

By knowing the form of the optimal solution, the Sinkhorn algorithm performs successive iterations in order to obtain the final $\mathbf{w}$ and $\mathbf{z}$. These iterations are based on the fact that $\mathbf{P} \in \mathbf{U}(\mathbf{u}, \mathbf{v})$, which means that:

$$\mathrm{diag}(\mathbf{w})\mathbf{K}\mathrm{diag}(\mathbf{z})\mathbf{1}_m = \mathbf{u} \quad \mathrm{diag}(\mathbf{z})\mathbf{K}^{\mathrm{T}}\mathrm{diag}(\mathbf{w})\mathbf{1}_n = \mathbf{v} \tag{3.20}$$

Hence, one obtains that:

$$\mathbf{w} = \frac{\mathbf{u}}{\mathbf{K}\mathbf{z}} \quad \text{and} \quad \mathbf{z} = \frac{\mathbf{v}}{\mathbf{K}\mathbf{w}} \tag{3.21}$$

Where the division in the equation above is element-wise. Below we present the actual Sinkhorn algorithm implementation.

Algorithm 1: Sinkhorn($\mathbf{u}, \mathbf{v}, \mathbf{C}, \varepsilon$,N)

$\mathbf{K} = \exp(-\mathbf{C}_{i,j}/\varepsilon)$; // applied element-wise

$\mathbf{z} = \mathbf{1}_m$; // initialize vector of 1's

for *i=1 to N* **do**

  $\mathbf{w} = \frac{\mathbf{u}}{\mathbf{Kz}}$; // division element-wise

  $\mathbf{z} = \frac{\mathbf{v}}{\mathbf{K}^{\mathrm{T}}\mathbf{w}}$; // division element-wise

end

return $\mathrm{diag}(\mathbf{w})\mathbf{K}\mathrm{diag}(\mathbf{z})$

In this version of Sinkhorn, the variable N, which represents the number of iterations, is freely defined, but this can be altered by using convergences criterions, such as the size of the variation in each iteration.

One of the issues with Sinkhorn's algorithm is numerical stability. As $\varepsilon \to 0$, the convergences of the algorithm deteriorates, due to the fact that the matrix product $\mathbf{Kw}$ and $\mathbf{K}^{\mathrm{T}}\mathbf{z}$ become too small, resulting in a division by 0. There are ways to improve such stability, such as doing the computations in the log domain. For a more comprehensive exposition of Sinkhorn and Entropic Regularization, the interest reader can read the work of Peyré et al. [61].

3.1.7 Sinkhorn Divergence

Using entropic regularization, one can obtain an optimal plan γ_ε and calculate the cost such that $OT_{c,\varepsilon} := \int_{\Omega\times\Omega} c(x,y)\ d\gamma_\varepsilon$. Thus, one can define the Sinkhorn Divergence as:

$$S_{c,\varepsilon}(\mu,\nu) := OT_{c,\varepsilon}(\mu,\nu) - \frac{1}{2}OT_{c,\varepsilon}(\mu,\mu) - \frac{1}{2}OT_{c,\varepsilon}(\nu,\nu) \tag{3.22}$$

The terms $OT_{c,\varepsilon}(\mu,\mu)$ and $OT_{c,\varepsilon}(\nu,\nu)$ are introduced due to the fact that $OT_{c,\varepsilon}(\mu,\mu)$ is not always equal to 0 when $\varepsilon > 0$.

Recently, Feydy et al. [26] proved the following :

Theorem 3.1.1. *(Properties of Sinkhorn Divergence)*

For Ω compact metric space, and $c(x,y)$ a Lipschitz function that induces a positive universal kernel $k_\varepsilon(x,y) := \exp(-c(x,y)/\varepsilon)$ and $\varepsilon > 0$. Then,

$S_{c,\varepsilon}$ is symmetric positive definite, smooth and convex in each input. Also, it metrizes weak convergence, i.e. $S_{c,\varepsilon}(\mu_n, \mu) \to 0 \iff \mu_n \rightharpoonup \mu$.

Note that for $\Omega \subset \mathbb{R}^d$ compact, and $c(x, y) = |x - y|$ or $|x - y|^2$, the conditions of the theorem are valid.

3.2 Sliced Wasserstein

Besides regularization, another popular distance used as a proxy of the Wasserstein distance is the Sliced Wasserstein (SW). Introduced by Bonneel et al. [10], this distance is equivalent to the Wasserstein distance of the Radon transformation of the original measures. First, let's define the Radon Transform and then define the Sliced Wasserstein.

Definition 3.2.1. (Definition from Kolouri et al. [42])
Let $\mathcal{R} : L^1(\mathbb{R}^d) \to L^1(\mathbb{R} \times \mathbb{S}^{d-1})$, where $\mathbb{S}^{d-1} := \{\theta \in \mathbb{R}^d : ||\theta|| = 1\}$ is the unit sphere [2]. $\mathcal{R}$ is a Radon transform if for any $f \in L^1(\mathbb{R}^d)$

$$\mathcal{R}f(t, \theta) = \int_{\mathbb{R}^d} f(x)\delta(t - \langle x, \theta \rangle)dx \tag{3.23}$$

For $(t, \theta) \in \mathbb{R} \times \mathbb{S}^{d-1}$.

This defines the Radon transform for functions. To define a transform for measures, we first need the adjoint.

Definition 3.2.2. (Definition adapted from Bonneel et al. [10])
$R^* : C_0(\mathbb{R} \times \mathbb{S}^{d-1}) \to C_0(\mathbb{R}^d)$ is the adjoint Radon (back projection) operator if for every $g \in C_0(\mathbb{R} \times \mathbb{S}^{d-1})$, then

$$R^*g(x) = \int_{\mathbb{S}^{d-1}} g(\langle x, \theta \rangle, \theta)d\theta \tag{3.24}$$

With this, we can use the duality of $\mathcal{P}(\mathbb{R}^d)$ with $C_0(\mathbb{R}^d)$ to define a transform on measures.

Definition 3.2.3. (Radon transform for probability measures)

[2]Note that Peyré et al. [61] uses a different notation, where $\mathbf{S}^d$ is equivalent to $\mathbb{S}^{d-1}$.

Let $\mu \in \mathcal{P}(\mathbb{R}^d)$. Hence, $\mathcal{R}\mu = \nu$ if for all $g \in C_0(\mathbb{R} \times \mathbb{S}^{d-1})$ then

$$\int_{\mathbb{R}\times\mathbb{S}^{d-1}} g(t,\theta)d\nu(t,\theta) = \int_{\mathbb{R}^d} (R^* g)(x)d\mu(x) \tag{3.25}$$

Thus, $R : \mathcal{P}(\mathbb{R}^d) \to \mathcal{P}(\mathbb{R} \times \mathbb{S}^{d-1})$.

We can now define the Sliced Wasserstein distance.

Definition 3.2.4. Let $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$, the p-Sliced Wasserstein distance is the p-Wasserstein distance of the Radon transformed measures, i.e.

$$SW_p(\mu,\nu) := W_p(\mathcal{R}\mu, \mathcal{R}\nu) \tag{3.26}$$

Bonneel et al. [10] proves that this is also equivalent to

$$SW_p(\mu,\nu) = \left(\int_{\mathbb{S}^{d-1}} [W_p(P_{\theta\#}\mu, P_{\theta\#}\nu)]^p d\theta\right)^{\frac{1}{p}} \tag{3.27}$$

Where $P_\theta : \mathbb{R}^d \to \mathbb{R}$ is the projection operator.

Note that to estimate SW, one only needs to calculate the 1-dimensional Wasserstein and integrate over the sphere. In this case, the Wasserstein distance has a closed form solution and for discrete measures consists of a sorting algorithm. While the integration can be performed via Monte Carlo. Figure 3.3 illustrates how the SW distance is calculated.

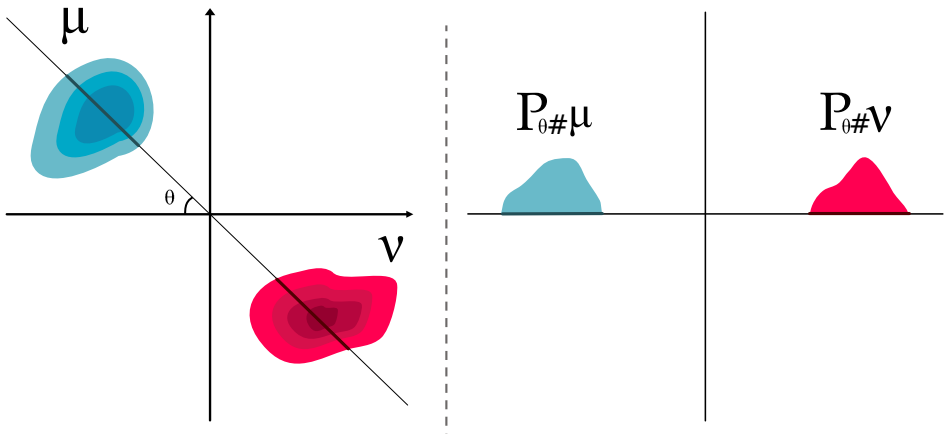

Figure 3.3: Schematic drawing of the calculation of the Sliced Wasserstein distance.

3.3 Gromov-Wasserstein

The Wasserstein metric and its variants presented until now are restricted to measures defined in the same space. Yet, there are many possible applications where one is interested in defining distances for measures across different spaces, for example, comparing images with different sizes which is akin to comparing measures defined on spaces with different dimensions. To address this issue, the Gromov-Wasserstein distance was introduced by Mémoli [54].

Definition 3.3.1. (Gromov-Wasserstein)

Let (X, d_x, μ_x) and (Y, d_y, ν_y) be two metric measure spaces[3] where d_x, d_y are the distance metrics and μ_x, ν_y probability measures defined in each respect space. Then, the Gromov-Wasserstein (GW) distance is given by:

$$GW_p(\mu_x, \nu_y) := \inf_{\gamma \in \Pi(\mu_x, \nu_y)} \left(\int_{X^2 \times Y^2} |d_x(x, x') - d_y(y, y')|^p d\gamma(x, y) d\gamma(x', y') \right)^{\frac{1}{p}} \tag{3.28}$$

The GW distance is a metric between measures in different spaces up to an isometry, where two metric measure spaces are said to be isometric if $\exists f : X \to Y$ such that f is a bijection satisfying $f_{\#}\mu_x = \nu_y$ and $d_y(f(x), f(x')) = d_x(x, x')$. Therefore, if the measure spaces are isometric, then the Gromov-Wasserstein distance is equal to zero. This can be shown by noting that the bijective function that defines the isometry can be used as optimal transport map.

Note that since GW is a distance up to an isometry, then such distance is invariant to transformations such as translations and rotations. Intuitively, what this distance does is take a pair of particles of measure μ_x and compare how much their distance changed after transportation. Figure 3.4 exemplifies how the Gromov-Wasserstein is calculated.

Similarly to the Wasserstein distance, the Gromov-Wasserstein also has an entropic regularized version, which can be used to approximate GW [61].

Finally, another variant of this metric is the so called Fused Gromov-Wasserstein (FGW), which was recently introduced by Titouan et al. [84]. This new distance is used on what the authors define as Structured Objects. One example of such objects are undirected labeled graphs.

[3] A metric measure space is a triple (X, d, m) consisting of a space X, a complete separable metric d on X and a Borel probability measure m on it.

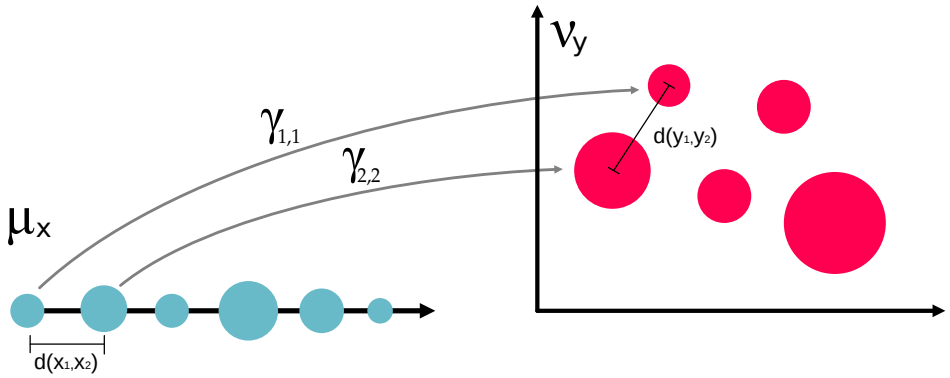

Figure 3.4: Example of Gromov-Wasserstein for discrete measures, with $X = \mathbb{R}^1$ and $Y = \mathbb{R}^2$.

Definition 3.3.2. (Structured Object)

Let (Ω, d) be a metric space, which is called *feature space*. A structured object is the triplet $(X \times A, d_x, \mu)$, where (X, dx) is a compact metric space called structure space, $A \subset \Omega$ is compact, and $\mu \in \mathcal{P}(X \times A)$. μ_A is the marginal measure on (A, d) and μ_x the marginal on (X, d_x).

From Vayer et al. [89], we formally define the FGW distance as the following[4]:

Definition 3.3.3. (Fused Gromov-Wasserstein)

Let $\mathcal{G}_1, \mathcal{G}_2$ be two structured objects on $(X \times A, d_x, \mu)$ and $(Y \times B, d_y, \nu)$, respectively, and with feature space (Ω, d). For $\alpha \in [0, 1]$ and $p, q \geq 1$, the Fused Gromov-Wasserstein (FGW) distance is given by:

$$\inf_{\gamma \in \Pi(\mu,\nu)} \left(\int_E D_\alpha^{(q)}(a, b, x, x', y, y')^p \; d\gamma((x, a), (y, b)) d\gamma((x', a), (y', b)) \right)^{\frac{1}{p}} \tag{3.29}$$

Where

$$D_\alpha^{(q)}(a, b, x, x', y, y') := ((1 - \alpha)d(a, b)^q + \alpha |d_x(x, x') - d_y(y, y')|^q) \tag{3.30}$$

$$E := X \times A \times Y \times B \tag{3.31}$$

[4]There seems to be a typo in the definition from the paper from Vayer et al. [89]. The authors integrate in terms of $\gamma((x', a'), (y', b'))$, but the correct should be $\gamma((x', a), (y', b))$

Note that when $\alpha = 0$, then the FGW is equal to the Wasserstein distance on the feature space, and when $\alpha = 1$ it becomes equal to the Gromov-Wasserstein in the structure space. Thus the reason for it's name. Figure 3.5 exemplifies the use of this metric.

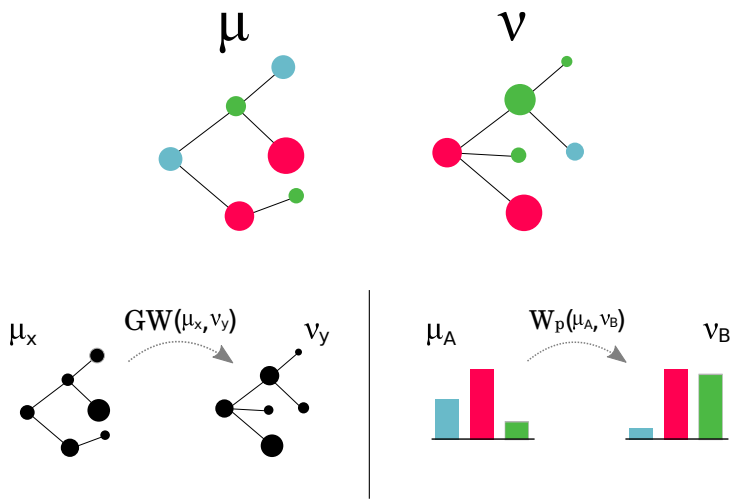

Figure 3.5: Example of Fused Gromov-Wasserstein for measures defined on undirected labeled graphs. The color represents the feature space where a Wasserstein distance is defined, while the graphs represent the structure space. Note that the FGW distance is a combination of both.

3.4 Barycenter in Optimal Transport

In Machine Learning and Statistics, there are several instances where one is interested in computing a "mean" or some sort of point estimate that represents a set of data points. The generalization of this notion of mean for metric spaces is known as **Fréchet mean**. In a similar manner, Agueh and Carlier [2] introduced the notion of a barycenter for Wasserstein spaces. As shown in Figure 3.6, the Wasserstein barycenter preserves the shape of the measures, while the Euclidean mean smooths out the distribution.

The generalized definition of barycenters for Optimal Transport is given below.

Definition 3.4.1. Let (X, d) be a metric space and $(\nu_s)_{s\in S} \subset \mathcal{P}(X)$. The barycenter of the set $(\nu_s)_{s\in S}$ is defined as:

$$\beta((\nu_s)) := \underset{\mu\in\mathcal{P}(X)}{\operatorname{argmin}} \sum_{s=1}^{S} \lambda_s OT_c(\mu, \nu_s) \tag{3.32}$$

Where $OT_c(\cdot,\cdot)$ is the minimum transport cost, $\lambda_s \geq 0$ and $\sum_{s=1}^{S} \lambda_s = 1$. If $c(x, y) = d(x, y)^p$, then this is called the p-Wasserstein barycenter, and is given by:

$$\beta((\nu_s)) := \underset{\mu\in\mathcal{P}_p(X)}{\operatorname{argmin}} \sum_{s=1}^{S} \lambda_s W_p^p(\mu, \nu_s) \tag{3.33}$$

Interestingly, the notion of p-Wasserstein barycenter is directly related to k-means. Let $(\nu_s) = \{\nu\}$, that is, the set containing only one measure. Also, let $(\mathbb{R}^n, d)$ be an Euclidean space with $d(x, y) = ||x - y||_2$ and $p = 2$. By reformulating the barycenter problem as:

$$\min_{\mu\in\Sigma_k(\mathbb{R}^n)} W_2^2(\mu, \nu) \tag{3.34}$$

Where Σ_k is the set of discrete measures supported on k points. Then, the solution to (3.34) is equivalent to k-means Peyré et al. [61].

For discrete measures, the barycenter problem might be formulated as a Linear Program, but it's complexity becomes prohibitively large even for medium scale problems [61]. Similarly to the Wasserstein distance, there are many variants to the Wasserstein barycenter, which use other distances such as entropic regularized Wasserstein [21] and Sliced Wasserstein [10].

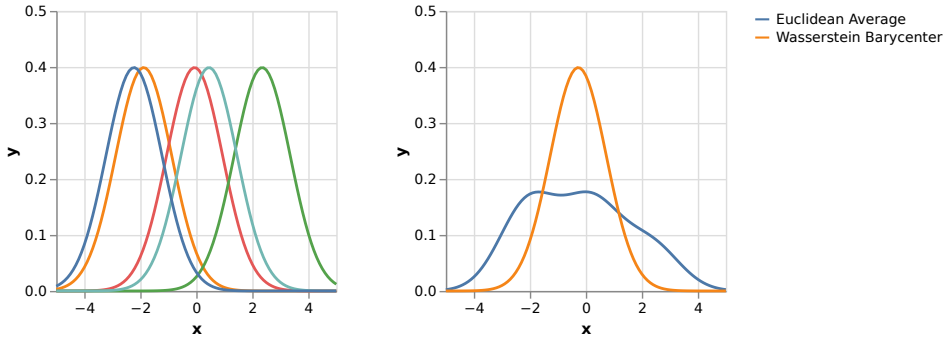

Figure 3.6: In the left you have different distributions, and in the right, there is a comparison between averaging these distribution versus finding the OT Barycenter.

3.5 Unbalanced Optimal Transport

All the methods presented until now deal with the case where measures μ, ν have the same total mass. Yet, in some ML problems, such as image segmentation, one has to deal with unnormalized measures. These cases are called Unbalanced Optimal Transport.

Definition 3.5.1. (Definition from Peyré et al. [61])

Let $\mu \in \mathcal{M}_+(X)$ and $\nu \in \mathcal{M}_+(Y)$, where $\int_X d\mu \neq \int_Y d\nu$. Hence, the Unbalanced Optimal Transport problem is defined as:

$$\begin{aligned} OT_c^{(\tau_1,\tau_2)}(\mu,\nu) := \inf_{\gamma \in \mathcal{M}_+(X\times Y)} & \int_{X\times Y} c(x,y)\ d\gamma \\ & + \tau_1 D_\phi((\pi_x)_\# \gamma, \mu) + \tau_2 D_\phi((\pi_y)_\# \gamma, \nu) \end{aligned} \tag{3.35}$$

Where $\tau_1, \tau_2 \in \mathbb{R}_+$ and D_ϕ is a divergence function, such as Kullback-Leibler.

Note that this formulation seeks to find a transport plan γ that is not necessarily a coupling of μ and ν anymore. Instead, the coupling condition is relaxed through the use of a penalization on the difference between the marginal distributions of γ and the measures μ and ν. We can see that this is indeed a relaxation of the original problem since when $\tau_1, \tau_2 \to +\infty$ and $\int_X d\mu = \int_Y d\nu$, the original OT problem is obtained.

Chapter 4

Optimal Transport for Machine Learning

Now that we've introduced both theoretical and the computational aspects of Optimal Transport, we can delve into its usage on Machine Learning. In this section, we start with an overview of the subject before going into detail of the applications themselves. This overview is aimed at providing an organized look at the whole landscape of applications. After that, we review the many methods developed in the literature. We divide the section according to each learning task shown in Figure 1.3. Note that there is no single taxonomy that is agreed upon on Machine Learning, hence, its use in this work is mostly aimed on providing a cohesive way to present the different applications.

4.1 Overview - The Landscape of Applications

The field of Machine Learning is very broad and tackles many different problems, thus, OT has also been applied in different ways. Yet, there are mainly two categories in terms of how OT used in ML (recall Figure 1.4).

Optimal Transport is most frequently used as a metric. In ML, we are regularly working with probability distributions, hence, metrics such as Wasserstein are very helpful, since it metrizes weak convergence and preserves the geometric properties of the underlying space, producing meaningful distances even when the distributions do not share the same support.

The usefulness of these properties comes in many ways. For example, in Generative Adversarial Network, the model seeks to learn the latent distribution of the dataset, thus, the loss function consists in comparing the generated distribution versus the real dataset distribution. The original GAN introduced by Goodfellow et al. [33] uses the Jensen-Shannon Divergence (JSD)[1] as loss function, which is a symmetric version of the Kullback-Leibler divergence.

A problem with such divergence is that when the supports of the distributions do not overlap, or the overlap is too small, the divergence converges to $\log 2$ producing a gradient of zero. Hence, the model becomes difficult to train, as one usually relies on gradient descent to update the weights in the network. This is illustrated in Figure 4.1[2].

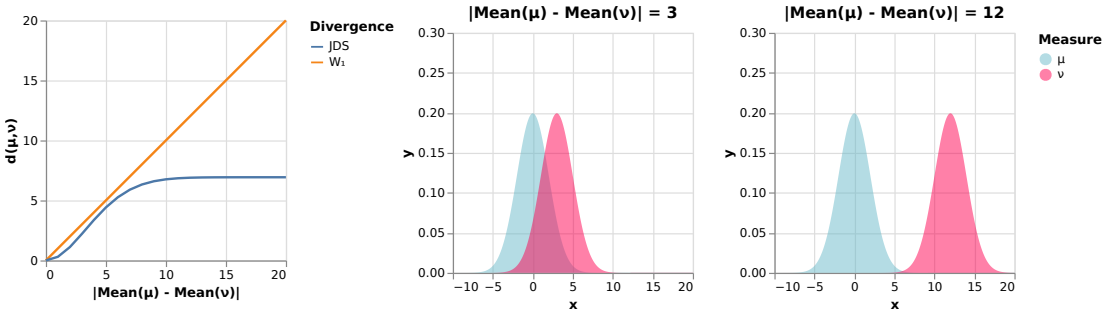

Figure 4.1: Comparison between the Jensen-Shannon divergence and the 1-Wasserstein distance for two Gaussian distributions μ and ν. The image on the left illustrates how each of these distances change as the distributions move apart. On the right, we show two examples of μ and ν, first when they are 3 units apart and then when they are 12 units apart.

In the scenario shown in Figure 4.1, while the JSD gradient quickly vanishes, the Wasserstein distance returns meaningful gradients in all cases. Thus, even when the distributions are very far apart, the network is still able to learn. This is one of the great advantages of models such as WGAN [5].

Besides been used as a loss functions, OT distances can also be used to measure the discrepancy between input datasets (e.g. Alvarez-Melis and Fusi [3]) and even between models themselves (e.g. Li et al. [49]).

[1] $JSD(\mu||\nu) := \frac{1}{2}KL(\mu||\frac{\mu+\nu}{2}) + \frac{1}{2}KL(\nu||\frac{\mu+\nu}{2})$

[2] The JSD is scaled by 10 to improve the visibility of the plot.

Yet, this flexibility comes at a cost. The use of distances such as Wasserstein requires that we solve an Optimal Transport problem, which, in the case of discrete metrics, corresponds to solving a Linear Program. This means that when training a model, we would need to solve a Linear Program before every gradient descent step, which is too costly for practical use.

Many alternative metrics have been developed with the aim of approximating the Wasserstein distance, but with increased computational efficiency. Some of these alternative metrics were already introduced in Section 3.

The other common application consists in using the actual optimal transport plan. The most prominent example is the use of the OT plan for dataset alignment for Transfer Learning (e.g. [17]). The idea consists in transporting the source dataset to the target dataset, in a way that a model trained on the transported source can be used in the target data. There are many variations to this idea, such as the addition of regularization to enforce that datasets with the same label are kept together.

Transport plans are also used to calculate the OT barycenter. As explained in Section 3.4, the OT barycenter can be intuitively understood as a kind of averaging that preserves the "shape" of the distributions. It has been naturally employed in model ensembling methods [25].

The least common are ML algorithms that reformulate its original problem turning it into an Optimal Transport problem. In these situations, the proposed algorithms are usually quite unique, and the learning process usually involves the solution of an Optimal Transport problem. An example of this is the work of Liutkus et al. [50], where the authors reformulate the generative modeling problem turning it into a gradient flow problem. Another example of this is the work of Cuturi et al. [22], where a the authors reformulate the problem of ranking as an Optimal Transport problem, and use entropic regularization to create a differentiable ranking operator.

Figure 4.2 summarizes on a single visualization how the 70 selected papers are distributed across the different taxonomies proposed (i.e. ML paradigm, OT Usage Type, OT Problem Formulation).

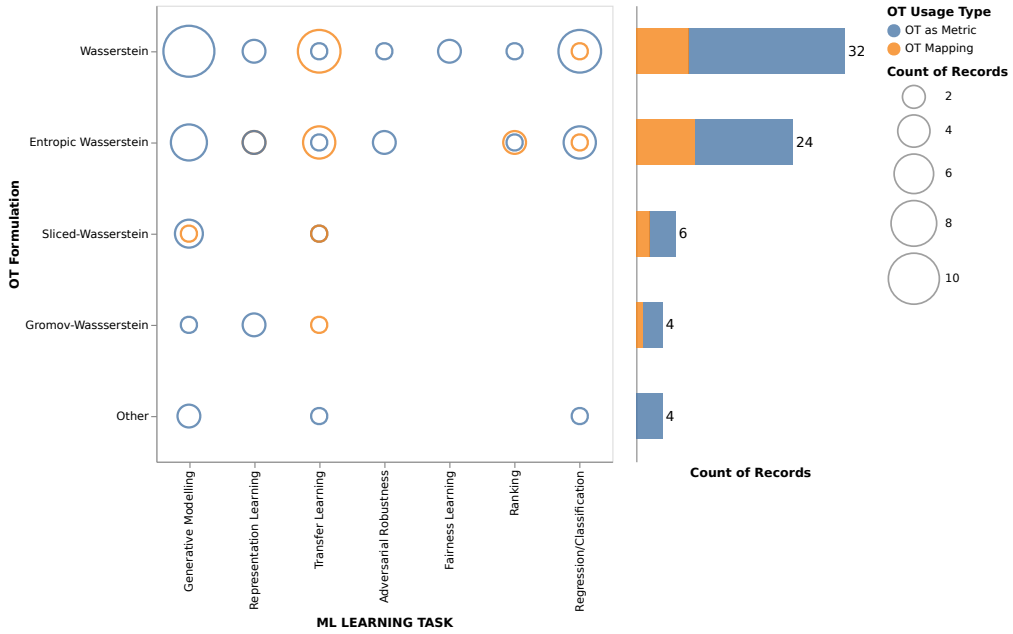

Figure 4.2: Distribution of the selected papers across ML paradigm, OT Usage Type and OT Problem Formulation.

4.2 Unsupervised Learning

Mohri et al. [55] defines Unsupervised Learning as:

> The learner exclusively receives unlabeled training data, and makes predictions for all unseen points. Since in general no labeled example is available in that setting, it can be difficult to quantitatively evaluate the performance of a learner. Clustering and dimensionality reduction are examples of unsupervised learning problems.

In our survey, the category Unsupervised Learning has the most applications of Optimal Transport, with almost all of them consisting in using OT as a metric.

4.2.1 Generative Modelling

Generative Modelling assumes that the dataset can be described by an parametric probability distribution and the goal is to learn it [77]. Hence, one would be able to generate samples from this distribution, which is the case of algorithms such as Generative Adversarial Networks (GAN).

Although classified as an Unsupervised Learning task, there are cases where Generative Models (GM) might be used for Supervised Learning. Yet, they are mostly used in unsupervised scenarios as shown in this recent survey by Harshvardhan et al. [37].

Restricted Boltzmann Machine (RBM) with Wasserstein Distance

One of the first uses of OT for generative modeling was Montavon et al. [56]. The authors proposed using Entropic Regularized Wasserstein cost for training a Restricted Boltzmann Machine. RBMs are graph based models with one layer for input variables and a hidden layer for latent variables, with the capacity to learn a representation of the data. RBMs usually try to the KL Divergence, but in this work, the authors propose the following objective function:

$$\min_{\theta \in \Theta} W_{1,\varepsilon}(\hat{p}, p_\theta) + \lambda\ \Omega(\theta), \tag{4.1}$$

where $\hat{p}$ is the empirical distribution, $W_{1,\varepsilon}$ is the Entropic Regularized 1-Wasserstein cost and $\lambda\ \Omega(\theta)$ is a regularization term added to confer stability

to the optimization. The Wasserstein-RBM model produced distributions with lower Wasserstein distance from the original and performed better than the regular RBM in tasks such as completion and denoising.

Wasserstein Generative Adversarial Networks

Generative Adversarial Networks were introduced by Goodfellow et al. [33], with the goal of generating realistic synthetic data based on real samples. The general idea behind GANs consists in using two neural networks to compete against one another, where one is responsible for generating the synthetic samples (Generator), and the other tries to identify which data is real and which is synthetic (Discriminator).

During training, the GAN seeks to solve the following minmax problem:

$$\min_G \max_D V(D,G) = \mathbb{E}_{P_{data}(\boldsymbol{x})}\left[\log\left(D(\boldsymbol{x})\right)\right] + \mathbb{E}_{P_z(\boldsymbol{z})}\left[\log(1 - D(G(\boldsymbol{z}))\right], \quad (4.2)$$

Where $P_{data}(\boldsymbol{x})$ is the probability distribution of the data, P_g is the distribution of the generated samples, $P_z(\boldsymbol{z})$ is the prior distribution of the input, D is the discriminator network and G is the generator. If the discriminator is optimal, the minimization problem for the generator becomes equivalent to minimizing the Jensen-Shannon Divergence (JSD).

The Wasserstein GAN (WGAN), introduced by Arjovsky et al. [5], modifies the original GAN by changing the loss function from Jensen-Shannon Divergence to a 1-Wasserstein Distance, as shown in Figure 4.3.

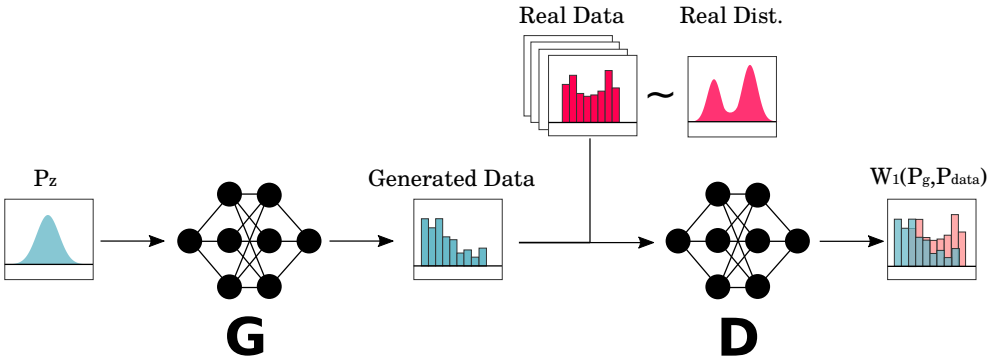

Figure 4.3: Schematic drawing of a WGAN.

Note that for the 1-Wasserstein Distance, the cost function is a metric, hence, one can make use of the of the Kantorovich-Rubinstein's Theorem

(2.5.5). Therefore, the Dual of the 1-Wasserstein distance can be written as the supremum of 1-Lipschitz functions. Thus, in the optimization problem for the WGAN the generator seeks to minimize the Wasserstein distance between the generated distribution and the real distribution:

$$\min_G W_1(P_{\text{data}}, P_g) = \min_G \max_{D \in Lip_1} \mathbb{E}_{P_{\text{data}}}[D(x)] - \mathbb{E}_{P_g}[D(x)] \tag{4.3}$$

The goal of network D now is to approximate the 1-Lipschitz function that solves the Optimal Transport between P_{data} and P_g. For WGAN, one usually calls D a critic instead of a discriminator, since the output of D is not restricted to $[0, 1]$ anymore.

Arjovsky et al. [5] also showed that one could differentiate $W_1(P_{\text{data}}, P_g)$, obtaining $\nabla_{\theta_g} W_1(P_{\text{data}}, P_g) = -\mathbb{E}_{P_z}[\nabla_\theta f(g(\mathbf{z}))]$, where θ_g are the parameters of the generator network. Hence, this allows the use of stochastic gradient descent to train the model.

The authors proposed that to reinforce the Lipschitz condition in (4.3) the weights of the neural network of D should be clipped between $[-a, a], a \in \mathbb{R}_+$. However, Arjovsky et al. [5] argue that clipping is not an appropriate method for guaranteeing the Lipschitz condition, as it becomes another hyperparameter to tune and makes D a very sparse and unstable networks.

Gulrajani et al. [35] suggested using a gradient penalization as a means to guarantee the Lipschitz condition, which the authors showed to work both in theory and empirically. The penalized loss function becomes:

$$\mathbb{E}_{P_{\text{data}}}[D(x)] - \mathbb{E}_{P_g}[D(x)] + \lambda \mathbb{E}_{P_{\hat{x}}}[(||\nabla \hat{x} D(\hat{x})||_2 - 1)^2] \tag{4.4}$$

Where $\hat{x} = \epsilon x + (1 - \epsilon)\tilde{x}$, $\epsilon \sim U[0, 1]$, $x \sim P_{data}$, $\tilde{x} \sim P_g$ and $\lambda > 0$.

Another method for enforcing this Lipschitz condition was proposed by Saito et al. [68], which they called Singular Value Clipping. The method consists in performing Singular Value Decomposition (SVD) in the matrix of weights of each layer, and then replacing all singular values larger then one with one, thus satisfying the 1-Lipschitz constraint. Since the neural network consists of a composition of functions, the authors argued that if every $f_i \in Lip_1$, then $f_N \circ f_{N-1} \circ ... \circ f_1 = D \in Lip_1$.

Motivated by the lack of stability of gradient descent for WGANs, Nagarajan and Kolter [57] suggested the addition of a regularization term to the gradient descent. Their proposal does not seek to enforce the Lipschitz condition, but only provide more stability to the gradient descent. The modified

gradient descent step proposed was

$$\theta_g^{(i+1)} := \theta_g^i - \alpha\nabla_{\theta_g}(W_1(P_{data}, P_g) + \eta||\nabla_{\theta_D}W_1(P_{data}, P_g)||^2) \quad , \eta > 0. \quad (4.5)$$

Finally, Cao et al. [14] extended the WGAN to the problem of multiple marginal matching, where one wants to generate samples for multiple target distributions instead of just one. While the original WGAN solves a regular Optimal Transport problem, this new model requires the solution of a multi-marginal Optimal Transport problem.

Generative Networks with Sinkhorn Divergence

Genevay et al. [32] introduced a method for training generative models using Sinkhorn Divergence[3]. With this, the objective function for the generator becomes $\min_\theta S_{c,\varepsilon}(\mu_\theta, \nu)$. To solve this optimization problem, the authors proposed to approximate $S_{c,\varepsilon}$ with a mini-batch sampling and use an L fixed number of steps in the Sinkhorn Algorithm 1, which allows the use automatic differentiation techniques to obtain the gradient. Note that since $S_{c,\varepsilon}$ is solved using Sinkhorn, this model does not actually need a discriminator (critic) network.

By solving the primal problem instead of the Kantorovich-Rubinstein dual, one can choose different cost functions besides using a distance metric. Hence, Genevay et al. [32] proposed to obtain the cost function c as:

$$c_\phi(x, y) := ||f_\phi(x) - f_\phi(y)|| \quad (4.6)$$

Where f_ϕ is modeled using neural networks similarly role to the discriminator in a GAN. Note that if this cost function estimation is not performed, the actual model if composed only of a generator network, since the Sinkhorn divergence is calculated via the Sinkhorn algorithm.

The use of Sinkhorn with a fixed amount of steps has the great advantages of been fully tractable. Nevertheless, it has the disadvantage of not been a valid metric and that fixed mini-batch sizes are not unbiased estimators of the gradients [69]. To address this issue, Salimans et al. [69] proposed a model called Optimal Transport GAN. The model also uses the Sinkhorn Divergence, introduces a new metric called Minibatch Energy Distance (MED) given by:

$$D^2_{MED}(\mu, \nu) := 2\mathbb{E}[S_{c,\varepsilon}(\hat{\mathbf{x}}_b, \hat{\mathbf{y}}_b)] - \mathbb{E}[S_{c,\varepsilon}(\hat{\mathbf{x}}_b, \hat{\mathbf{x}'}_b)] - \mathbb{E}[S_{c,\varepsilon}(\hat{\mathbf{y}}_b, \hat{\mathbf{y}'}_b)] \quad (4.7)$$

[3]The authors used a version slightly different from the one in Equation (3.22), by multiplying it by 2.

Where $\hat{\mathbf{x}}_b$, $\hat{\mathbf{x}'}_b$ are independently sampled mini-batches from distribution μ and $\hat{\mathbf{y}}_b$, $\hat{\mathbf{y}'}_b$ are independently sampled mini-batches from distribution ν. The authors also propose to learn the cost function adversarially.

In Sanjabi et al. [71], the authors also proposed the use of Sinkhorn Divergence, calling their model Smoothed WGAN. Differently from Genevay et al. [32], the Sinkhorn algorithm is not used to approximate the optimal transport plan, but to use a discriminator network to solve the regularized dual problem and then estimate the gradient of the Sinkhorn Divergence. The authors proved that the Entropic Wasserstein distance is smooth with respect to the generator parameters, which is not true under the original WGAN model. They also proved the convergence of stochastic gradient descent to a stationary solution. In the experiments performed, the Smoothed WGAN obtained faster convergence and better inception score when compared to the original WGAN, the model by Genevay et al. [32] and the model by Salimans et al. [69].

Generative Networks with Sliced-Wasserstein

Another interesting variation of generative network is the Sliced-Wasserstein Generator (SWG) proposed by Deshpande et al. [24]. Similar to Genevay et al. [32], the model does not have a discriminator network, only a generator which is trained using the Sliced-Wasserstein distance. To estimate the Sliced-Wasserstein distance, several random projections are sampled, hence, the actual optimization problem becomes

$$\min_{\theta} \frac{1}{|\hat{\Theta}|} \sum_{\theta \in \Theta} W_2^2(P_g^\theta, P_{data}^\theta), \tag{4.8}$$

where P_g^θ and P_{data}^θ are the 1-dimensional projections of the empirical distributions, and $|\hat{\Theta}|$ is the set of randomly sampled unit vectors in which the data is projected.

Wu et al. [93] proposed the use of Sliced-Wasserstein distance to Generative Adversarial Networks (SWGAN). The authors devised a method for approximating SW using a small number of parametrized orthogonal projection. Their model uses a discriminator network D to solve the dual formulation of the Sliced-Wasserstein distance via this method of orthogonal projections.

Generative Networks with Other OT Metrics

Guo et al. [36] proposed a method called Relaxed Wasserstein GAN, which used a new distance named Relaxed Wasserstein (RW). The main difference to the Wasserstein distance is that the cost function for RW is not necessarily a metric, but a Bregman cost function B_ϕ, defined as

$$B_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y \rangle, \tag{4.9}$$

where ϕ is strictly convex and continuously differentiable. Note that many distances are Bregman costs, such as the Euclidean distance (i.e. $||x - y||^2$) and the KL divergence. The authors focused on using the KL divergence as a cost, and devised a similar dual formulation to the W_1 distance. The RWGAN model is very similar to the WGAN, in which the critic is trained by clipping the weights in order to ensure the Lipschitz condition. The authors reported results similar to WGAN, but the model was able to produce real looking images with less training.

Chen et al. [15] developed a metric called Feature-Mover's distance (FMD) to be used on Generative Adversarial Networks for text generation. The model uses a learned word embedding to turn each word into a vector and each sentence into a matrix, and then a neural-network to extract a feature vector. The Feature-Mover's distance is the minimum transport cost between two set of feature, with the cost function equal to the cosine distance between features. The authors proposed the use of the Inexact Proximal point method for solving Optimal Transport problem (IPOT) [96] to calculate the FMD. The generative model with FMD is shown in Figure 4.4

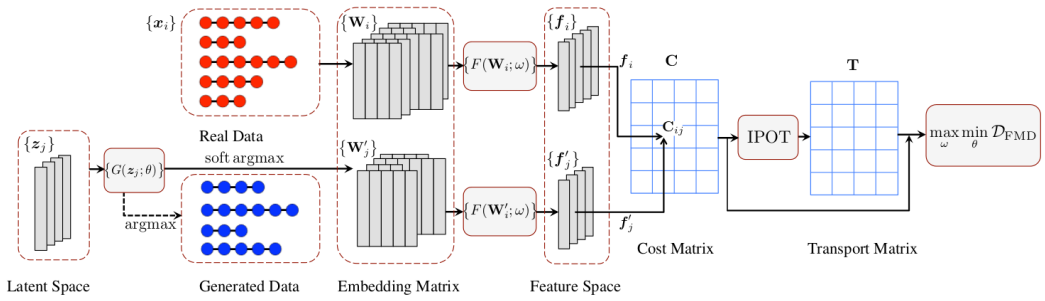

Figure 4.4: Illustration from Chen et al. [15] showing the GAN model using FMD for text generation.

In Berthelot et al. [9], instead of minimizing W_1, the authors suggested to minimize a lower bound of W_1 in order to train an auto-encoder based GAN.

Wasserstein Auto-Encoder

Besides Generative Adversarial Networks, another very popular method in Machine Learning for generative modelling are Variational Auto-Encoders (VAE) [41]. Variational Auto-Encoders seeks to minimize

$$D_{VAE}(P_{data}, P_g) = \inf_{Q(Z|X)\in\mathcal{Q}} \mathbb{E}_{P_X}\left[KL(Q(Z \mid X)||P_Z)\right] - \mathbb{E}_{Q(Z|X)}[\log p_G(X \mid Z)], \tag{4.10}$$

where $p_G(X \mid Z)$ is a density function and $\mathcal{Q}$ is a family of distributions. $\mathcal{Q}$ is typically chosen to be the set of Gaussian with $Q(Z \mid X) = \mathcal{N}(Z; m(X), \Sigma(X))$, such that $m(X)$ is the mean and $\Sigma(X)$ is the diagonal covariance, both parametrized by a neural network [12]. A VAE works by first learning an encoder that approximates a distribution $P(Z \mid X)$ via $Q(Z \mid X)$, where $X \sim P_{data}$, and $Z \sim P_z$, and then learning a decoder that reconstructs the real data base on samples from $Q(Z \mid X)$.

Building on the theoretical analysis performed by Bousquet et al. [12], Tolstikhin et al. [85] devised a new family of regularized Auto-Encoders, called Wasserstein Auto-Encoders (WAE). Bousquet et al. [12] proved that the OT distance was equivalent to

$$OT_c(P_X, P_G) = \inf_{P\in\Pi(P_X,P_G)} \mathbb{E}_P[c(X, G(Z))] = \inf_{Q:Q_Z=P_Z} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}[c(X, G(Z))] \tag{4.11}$$

Instead of solving an optimization problem with hard constraints, the authors suggest to enforce the condition $Q_Z = P_Z$ using a penalization, thus, the objective function becomes

$$D_{WAE}(P_X, P_G) := \inf_{Q:Q(Z|X)\in\mathcal{Q}} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}[c(X, G(Z))] + \lambda\mathcal{D}(Q_Z, P_Z), \tag{4.12}$$

where $\lambda > 0$ and $\mathcal{D}$ is some divergence function. Tolstikhin et al. [85] proposed two variations of WAE by varying the divergence function. The first, called WAE-GAN, uses the Jensen-Shannon divergence and approximates it via adversarial training. The second uses the Maxium Mean Discrepancy and is called WAE-MMD. The performance of the models were evaluated in the MNIST dataset, with both models outperforming a regular VAE, and the WAE-GAN obtaining the best results among the three models.

Following Tolstikhin et al. [85], the work of Kolouri et al. [42] proposes instead to use the Sliced-Wasserstein distance as $\mathcal{D}$, which avoids the need to perform adversarial training in the encoding space. To calculate SW, Kolouri et al. [42] used the same sampling method as Deshpande et al. [24]. The model, which the authors called Sliced-Wasserstein Auto-Encoder (SWAE), was shown to be competitive in the experiments performed.

Another variation of the Wasserstein Auto-Encoder was proposed by Patrini et al. [59]. The Sinkhorn Auto-Encoder used the Sinkhorn divergence as $\mathcal{D}$ and restricted the use of cost functions to the form $c(x, y) = d(x, y)^p$, such that $OT_c(\cdot, \cdot) = W_p(\cdot, \cdot)$. The reason for this restriction was due to a theoretical finding by the authors in which it was shown that for the optimization problem at hand, equation (4.12) would provably approximate (4.11).

Zhao et al. [101] and Zhang et al. [100] proposed two models based on Wasserstein Auto-Encoders. The former focused on learning generative models for discrete sequences such as text data, and the latter adapted the WAE for collaborative filtering.

Generative Modelling via Gradient Flow

Liutkus et al. [50] proposed to recast the generative modelling problem as a gradient flow problem. While models such as GAN and VAE rely on approximating the data distribution with a family of parametric functions modeled via neural networks, generative models based on gradient flow are non-parametric. The model proposed by Liutkus et al. [50], named Sliced-Wasserstein Flow (SWF), uses the Sliced-Wasserstein distance to measure the dissimilarity between the generated distribution and the real data distribution. The model seeks to minimize

$$\min_{\mu \in \mathcal{P}_2(\Omega)} \mathcal{F}_\lambda^\nu(\mu) := \min_{\mu \in \mathcal{P}_2(\Omega)} \frac{1}{2} SW_2^2(\mu, \nu) + \lambda \mathcal{H}(\mu), \tag{4.13}$$

where ν is the dataset distribution we are trying to generate, $\lambda > 0$ and $\mathcal{H}$ in the negative entropy[4]. Thus, the model tries to approximate ν with μ by modelling the gradient flow given by

$$\partial_t \mu_t = -\nabla_{W_2} \mathcal{F}_\lambda^\nu(\mu_t) \tag{4.14}$$

[4] $\mathcal{H}(\mu) := \int_\Omega \rho(x) \log \rho(x) dx$ if μ has a density ρ with respect to a Lebesgue measure and $\mathcal{H}(\mu) = +\infty$ otherwise

Liutkus et al. [50] developed a practical algorithm that approximates the solution to problem similarly to stochastic gradient Markov Chain Monte Carlo Methods (MCMC).

Generative Modelling Across Incomparable Spaces

Tackling a different kind of problem, Bunne et al. [13] used the Gromov-Wasserstein distance with a GAN (GWGAN) to learn generative models across incomparable spaces, e.g. spaces with different dimensions or different data types.

Due to the computational complexity to solve the Gromov-Wasserstein distance, it was proposed the use of Entropic Regularization together with a bias correction, similarly to the Sinkhorn Divergence. Experiments were performed first to guarantee that the model worked as a regular GAN. After that, other experiments were done showing that the model was indeed able to generate data in different dimensions and to recover the manifold structure on the generated distribution (Figure 4.5).

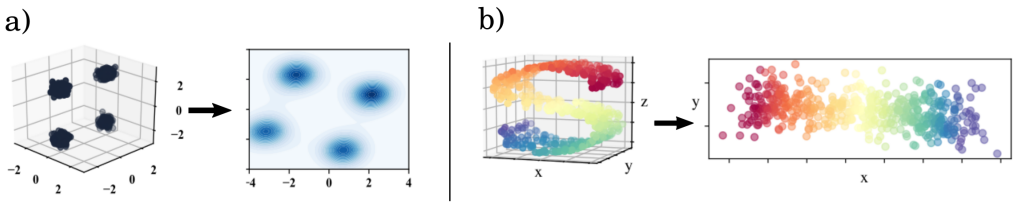

Figure 4.5: In (a) the model is receiving a 3-dimensional Gaussian mixture and generating a 2-dimensional gaussian mixture. While in image (b), one observe the preservation of the manifold. Figure adapted from Bunne et al. [13].

4.2.2 Representation Learning

Bengio et al. [8] define Representation Learning as:

> Learning representations of the data that make it easier to extract useful information when building classifiers or other predictors. In the case of probabilistic models, a good representation is often one that captures the posterior distribution of the underlying explanatory factors for the observed input.

Note that this encompass many different sub-tasks, such as Clustering, Dimensionality Reduction, Feature Engineering, and more. Sometimes, the term Representation Learning also refers to learning a neural network such that the architecture and weights might be reused for similar tasks.

Dictionary Learning with Wassersteins Distance

A branch of Representation Learning is called Dictionary Learning. In Dictionary Learning, for a collection of observations $\mathbf{X} = (x_1, ..., x_m)$ of m vectors with dimension n, we seek to find a dictionary $\mathbf{D} = (d_1, ..., d_k)$ of k elements and also dimension n, such that there exists a matrix of weights $\mathbf{H} = (h_1, ..., h_m)$ with $\mathbf{DH} \approx \mathbf{X}$. Note that for non-negative observations, this problem is also known as Non-negative Matrix Factorization (NMF) and is given by

$$\min_{\mathbf{D} \in \mathbb{R}_+^{n \times k}, \mathbf{H} \in \mathbb{R}_+^{k \times m}} \sum_{i=1}^{m} l(x_i, \mathbf{D}h_i) + R(\mathbf{D}, \mathbf{H}), \tag{4.15}$$

where l is a divergence measures such as KL divergence or the Euclidean distance, and R is regularization function. For observations consisting of discrete distributions, Sandler and Lindenbaum [70] were the first to propose the use of Wasserstein distance as loss function l. Their method did not use any regularization and required the solution of a Linear Program in order to estimate the Wasserstein distance. Hence, it was not applicable to large problems. Rolet et al. [65] solved this scalability problem by proposing the use of Entropic regularization. The use of Entropic regularization makes the problem convex in $\mathbf{D}$ and $\mathbf{H}$, individually. Therefore, the authors suggest to solve (4.15) using block-coordinate descent on $\mathbf{D}$ and $\mathbf{H}$. Figure 4.6 illustrates the different results obtained when using the Wasserstein distance instead of KL divergence.

Schmitz et al. [73] proposed a non-linear Dictionary Learning algorithm by using the Wasserstein Barycenter. The goal of the algorithm is also to find a dictionary $\mathbf{D}$ and a weight matrix $\mathbf{H}$, but, instead of using $\mathbf{DH}$ to approximate $\mathbf{X}$, each observation in $\mathbf{X}$ is approximated via the Entropic Regularized Wasserstein Barycenter. Hence, the minimization problem becomes

$$\min_{\mathbf{D}\in\mathbb{R}_+^{n\times k},\mathbf{H}\in\mathbb{R}_+^{k\times m}} \sum_{i=1}^{m} l(x_i, \beta_{h_i}^{\varepsilon}(\mathbf{D})), \tag{4.16}$$

where $\beta_{h_i}^{\varepsilon}(\mathbf{D})$ is the Entropic Wasserstein Barycenter of with respect to the histograms $d_1, ..., d_k$ with weights h_i.

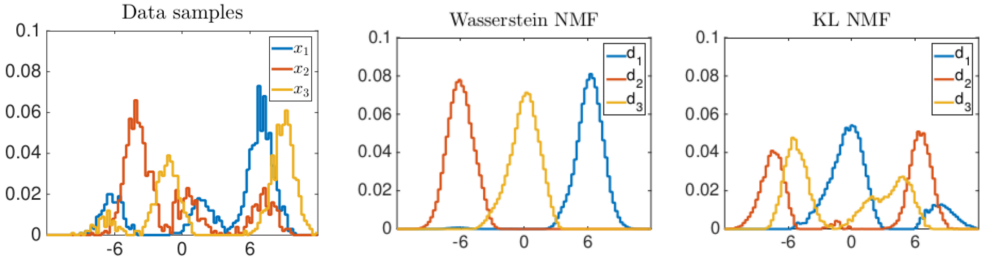

Figure 4.6: Figure from Rolet et al. [65]. The image on the right shows examples of histograms sampled from a mixture of randomly shifted Gaussians. In the middle is shown the dictionaries learned using Wasserstein NMF, while in the left is shown the dictionaries learned with KL NMF.

Wasserstein Principal Geodesic

A method commonly used in ML for Dimensionality Reduction is the Principal Component Analysis. The method consists in finding a set of orthogonal vectors that maximize the variance of the projected data. Seguy and Cuturi [74] proposed a new algorithm for obtaining what they call the Wasserstein Principal Geodesic (WPG). Similarly to PCA, the WPG is used for Dimensionality Reduction, but the goal in this case consists in finding curves in $\mathcal{P}(X)$ that summarize a family of probability measures on a Hilbert space X. Such curves are named geodesics and consist of functions describing the optimal flow from a set of measures (μ_i). Seguy and Cuturi [74] define the geodesic as $g_t(v_1, v_2) := (\mathrm{id} - v_1 + t(v_1 + v_2))_{\#}\gamma^*, t \in [0, 1]$, where $\bar{\mu}$ is 2-Wasserstein barycenter of the set of measures, and $\mathrm{id} - v_1, \mathrm{id} + v_2$ are

optimal mappings. The algorithm devised by Seguy and Cuturi [74] solves the following minimization problem for a family of empirical measures (μ_i):

$$\min_{v_1,v_2 \in L^2(\bar{\mu},X)} \lambda(\langle v_1, v_2 \rangle_{L^2} - ||v_1||_{L^2} \cdot ||v_2||_{L^2})^2 + \sum_{i=1}^{N} \min_{t \in [0,1]} W_2^2(g_t(v_1, v_2), \mu_i), \tag{4.17}$$

such that $\lambda > 0$, and $v_1 + v_2 \in \text{span}(\{v_1^{(i)} + v_2^{(i)}\})^{\perp}$. Figure 4.7 exemplifies the WPG for a toy model of empirical measures in 2 dimensions.

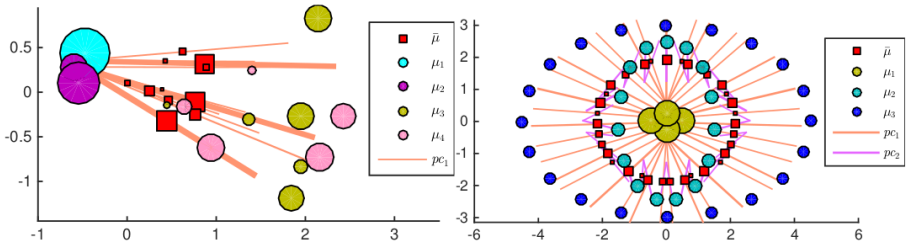

Figure 4.7: Figure from Seguy and Cuturi [74]. Two examples of empirical measures and the Principal Geodesic component. In the left, only the first component is shown, while in the right the first two are shown.

Self-labelling via Optimal Transport Clustering

Self-supervision is an increasingly popular technique where unlabeled data is labeled through clustering at the same time that a Neural Network learns to classify based on this synthetic label. Asano et al. [6] devised a method that uses a Deep Neural Network to perform classification, while OT is used to cluster the data. For the method to work, the number K of labels is defined by the user. The algorithm enforces that the data of size N to be divided equally among the K labels. The training method is the following. Let $\Phi(\cdot)$ represent the feed-forward of the Deep Neural Network, such that x_i represents a sampled data (e.g. an image) and $\Phi(x_i)$ returns a vector $\boldsymbol{w}^{(i)}$ where each entry $\boldsymbol{w}_j^{(i)}$ corresponds to the probability of x_i belonging to label y_j. In the first training step, each data sample is attributed a label randomly with equal probability. Then, the Neural Network is trained using cross-entropy to classify according to the given label. Thus, a matrix $\mathbf{W}$ is obtained, such that $\mathbf{W}_{i,j} = \Phi(x_i)_j = P(y_j \mid x_i)$. Such matrix is used as a

cost matrix in an Entropic Regularized OT problem

$$\min_{\mathbf{Q}\in\mathbf{U}(\mathbf{u},\mathbf{v})} \langle \mathbf{W}, \mathbf{Q} \rangle - \varepsilon \mathbf{H}(\mathbf{Q}), \tag{4.18}$$

where $\mathbf{u} = \frac{1}{K}\mathbf{1}$ and $\mathbf{v} = \frac{1}{N}$.

The Sinkhorn Algorithm is used to obtain an optimal plan $\mathbf{Q}$, that maps each data to the respective label. With these new labels, the deep neural network is trained, and the process is repeated.

Co-clustering via Optimal Transport

For a given dataset $X \subset \mathbb{R}^{n\times m}$, most clustering algorithms focus on performing the clustering only in the data samples, that is, in the rows of matrix X. Co-clustering consists in performing clustering of both data samples (rows) and features (columns) simultaneously. The output of these kinds of models consists of blocks of data which are called co-clusters, and are useful in different kinds of applications, such as recommendation systems and text mining [46].

In the work of Laclau et al. [46], the authors devised a co-clustering algorithm using Entropic Optimal Transport. The method defines a transportation problem between two discrete measures

$$\hat{\mu} := \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i} \quad \text{and} \quad \hat{\nu} := \frac{1}{m}\sum_{j=1}^{m} \delta_{y_j}, \tag{4.19}$$

where x_i is a data sample, and y_j is a feature sample. The cost matrix is calculated using the Euclidean norm, hence $C_{i,j} = ||x_i - y_j||$. In case $n > m$ or $n < m$, the larger dimension is sampled in order to guarantee that x_i and y_j have the same dimension. In order to avoid such sampling operation, the authors proposed another method in which the cost matrix was calculated using the Gromov-Wasserstein distance, which is appropriate, since the GW distance can deal with vectors in different dimensions.

As shown in Section 3.1.4, the optimal transport for the Entropic OT problem can be expressed as $\gamma^* = \text{diag}(\mathbf{w})\mathbf{K}\text{diag}(\mathbf{z})$. Laclau et al. [46] sort the vectors $\mathbf{w}$ and $\mathbf{z}$, and apply a non-parametric jump detection algorithm in order to find the number of clusters. Each "significant" jump in the vector is seen as a possible cluster. The jump locations define the matrix co-cluster. Finally, the authors showed through experimental results that their method

outperformed the other state-of-the-art methods used for co-clustering, while also being computationally efficient and capable of detecting the number of co-clusters.

Wasserstein Procrustes

The Procrustes analysis is a method for learning a linear transformation between two sets of observations given by $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{Y} \in \mathbb{R}^{n \times d}$. By restricting the linear transformations to orthogonal matrices, one obtains the following problem

$$\min_{\mathbf{Q} \in \mathcal{O}(n)} ||\mathbf{XQ} - \mathbf{Y}||^2, \tag{4.20}$$

where $\mathcal{O}(n) = \{\mathbf{Q} \in \mathbb{R}^n \times \ : \ \mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}\}$. Equation (4.20) is called the **orthogonal Procrustes problem**. Suppose that $\mathbf{X}$ and $\mathbf{Y}$ are word embeddings, such that, for example, the rows of $\mathbf{X}$ are phrases in portuguese and the rows of $\mathbf{Y}$ are phrases in english. By solving (4.20), matrix $\mathbf{Q}$ can be used to translate from portuguese to english.

Grave et al. [34] proposed to use OT on this cross-lingual by adapting the orthogonal Procrustes problem. Their algorithm uses the 2-Wasserstein distance to measure the discrepancy between $\mathbf{XQ}$ and $\mathbf{Y}$, thus, the minimization problem becomes

$$\min_{\mathbf{Q} \in \mathcal{O}(n)} W_2^2(\mathbf{XQ}, \mathbf{Y}) = \min_{\mathbf{Q} \in \mathcal{O}(n)} \min_{\mathbf{P} \in \mathcal{P}_n} ||\mathbf{XQ} - \mathbf{PY}||_2^2, \tag{4.21}$$

where $\mathcal{P}_n = \{\mathbf{P} \in \{0,1\}^{n \times n},\ \mathbf{P}\mathbf{1}_n = \mathbf{1}_n,\ \mathbf{P}^{\mathrm{T}}\mathbf{1}_n = \mathbf{1}_n\}$ is the set of permutation matrices, and $\mathbf{P}$ is the optimal transport map. The authors propose to solve this minimization problem by applying stochastic gradient descent alternating between $\mathbf{P}$ and $\mathbf{Q}$.

Alvarez-Melis and Jaakkola [4] used the Gromov-Wasserstein distance on word embeddings in the orthogonal Procrustes problem. While Grave et al. [34] solved the Procrustes problem and the Optimal Transport problem simultaneously, Alvarez-Melis and Jaakkola [4] first finds the Optimal Transport plan $\boldsymbol{\gamma}^*$ for the GW distance and then solves the Procrustes problem

$$\min_{\mathbf{Q} \in \mathcal{O}(n)} ||\mathbf{X}\boldsymbol{\gamma}^* - \mathbf{PY}||. \tag{4.22}$$

The authors use Entropic Regularization to the Gromov-Wasserstein distance in order to efficiently solve it.

4.3 Semisupervised Learning

According to Mohri et al. [55], Semisupervised Learning is defined by:

> The learner receives a training sample consisting of both labeled and unlabeled data, and makes predictions for all unseen points. Semisupervised learning is common in settings where unlabeled data is easily accessible but labels are expensive to obtain. Various types of problems arising in applications, including classification, regression, or ranking tasks, can be framed as instances of semi-supervised learning.

This category is less common in Machine Learning than Unsupervised or Supervised learning. For the case of Optimal Transport applications, although Reinforcement Learning is a popular branch of Machine Learning, not many works were obtained. Transfer Learning was the only subcategory with a significant amount of work to be revised.

4.3.1 Transfer Learning

Transfer Learning consists in adapting the knowledge gained from one domain to another. The domain where labeled data is available is called source domain, and the transfer of knowledge is done to a target domain. We can formally define it as the following:

Definition 4.3.1. (Transfer Learning)
Let $\mathcal{D}_s = \{\mathcal{X}_s, P_s(x_s)\}$ and $\mathcal{T}_s = \{\mathcal{Y}_s, P_s(y_s \mid x_s)\}$ define the domain source and task source, and $\mathcal{D}_t = \{\mathcal{X}_t, P_t(x_t)\}$ and $\mathcal{T}_t = \{\mathcal{Y}_t, P_t(y_t x_t)\}$ define the domain target and task target. Then Transfer Learning aims to improve the learning of the target predictive function $P_t(y_t \mid x_t)$, using knowledge gained from the source, where $(\mathcal{D}_s, \mathcal{T}_s) \neq (\mathcal{D}_t, \mathcal{T}_s)$.

The field of Transfer Learning can be split in many subcategories according to the assumptions made on the domain and tasks of both source and target. The most studied subcategory is Domain Adaptation, where the source and target are assumed to share the same task, but have different marginal distributions on the data. In the Appendix 6.4, we expand on the different ways Transfer Learning is usually categorized.

Besides the different types of Transfer Learning problems, there are also a vast number of methods for transferring knowledge. For example, sharing the model parameters, using re-weighting schemes to account for the difference in data distribution or finding a similar subspace where both source and target domains can be represented.

Optimal Transport for Label Propagation in Graphs

To our knowledge, Solomon et al. [82] was the first to use OT for semi-supervised learning. The authors tackled the problem known as label propagation in graphs. Suppose that only a portion of the vertices have known information, and this information consists in a distribution. The goal is to predict the distribution in the vertices of the graph where there is no information. This type problem is very common in Transfer Learning. Take for example traffic prediction, where one has the traffic distribution of 24-hours in some intersections, and the goal is to somehow predict the distribution in the intersections where there is no information. Other example is weather forcasting, where information there is information only in a subset of cities, and we wish to predict this information for where there is no data.

The model proposed by Solomon et al. [82] consisted in using the 2-Wasserstein distance to propagate the distributions across the vertices, where the weight in the edges symbolized the geometric distance. Given two distributions μ and ν in two vertices, and a vertice in the middle with unknown information. The model proposed by Solomon et al. [82] seeks to minimize the Dirichlet energy, in which the Wasserstein distance is used to measure the discrepancy between the distributions:

$$\mathcal{E}_D[f] := \sum_{(v,w)\in E} \omega_e W_2(\mu_v, \mu_w), \tag{4.23}$$

where E is the set of edges, v and w are two vertices connected by an edge, ω_e is the weight of the edge, μ_v and μ_w are the distributions on each vertex.

Optimal Transport Domain Adaptation

Similar to the work of Arjovsky et al. [5], the work of Courty et al. [17] is seminal in the use of Optimal Transport for Machine Learning. From it, many other works have been developed, either extending it or proposing new methods base on the main idea introduced in the paper.

Courty et al. [17] proposed to use OT to tackle the problem of Unsupervised Domain Adaptation, that corresponds to the case where there is no label at all in the target. The authors assume that the difference in the source and target distribution is due to an unknown transformation $T : \mathcal{X}_s \to \mathcal{X}_t$, but that this transformation still preserves the conditional distribution:

$$P_s(y \mid x^s) = P_t(y \mid T(x^s)). \tag{4.24}$$

The main idea behind the model, which we call OTDA, consists in finding an Optimal Transport map from the source data distribution $\mu_s = \sum_{i=1}^{n_s} u_i^s \delta_{x_i^s}$ to the target data distribution $\mu_t = \sum_{i=1}^{n_t} u_i^t \delta_{x_i^t}$. Then, use it to transport the source data to the target data. After this, the model trained on the transported source data can be immediately used on the target data. This process is illustrated in Figure 4.8.

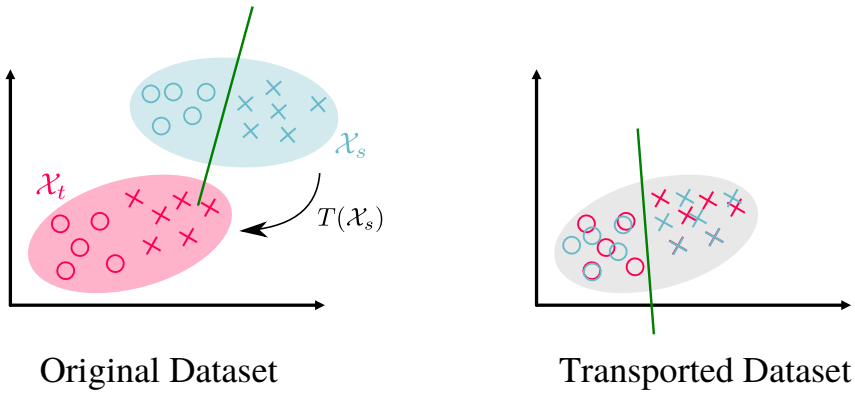

Figure 4.8: Schematic drawing of the OTDA algorithm. T corresponds to the unknown transformation between the source and the target. The shape of the markers represent the class label of each data sample, and the color indicates from which dataset they belong (blue for source and red for target). The line in green represents the learned classifier.

To find an Optimal Transport map, one would need to solve the Monge Problem, which would be quite computationally expensive. Instead, Courty et al. [17] suggest to solve the Kantorovich Problem with Entropic Regularization, thus, using the Sinkhorn algorithm to find the Optimal Transport plan. One cannot use the transport plan to transport the source data, since the transport plan may split the sample data. Instead, the authors propose

the use of **barycentric mapping**, which is

$$\hat{x_i^s} = \text{argmin}_x \sum_j \boldsymbol{\gamma}^*(i,j)c(x, x_j^t), \tag{4.25}$$

where γ^* is the OT plan with entropic regularization. Note that if the cost function is the squared L_2 distance, then this mapping is just the weighted average. The barycentric mapping is illustrated in Figure 4.9.

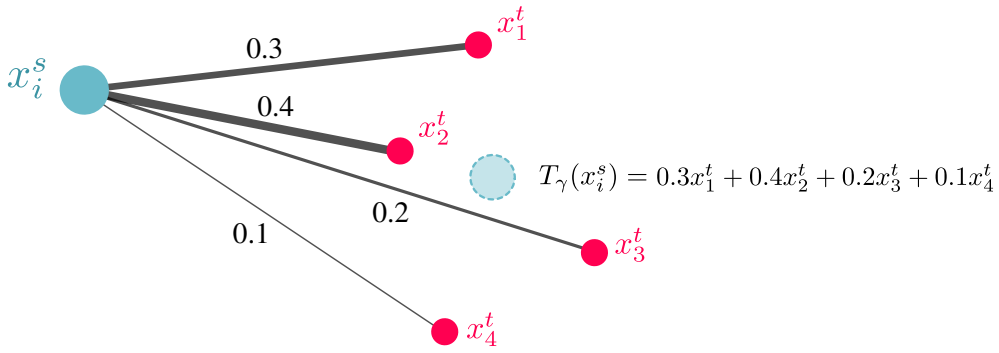

Figure 4.9: Example of barycentric coordinate mapping if the cost function is the squared L^2 distance in $\mathbb{R}^2$. The circle in blue represents one data sample from the source, the circles in red represent the target data, the circle in blue with dotted line represents the barycentric mapping of the source sample, and the lines in gray represent the optimal transport plan.

Courty et al. [17] also proposed the addition of an extra regularization term to the Entropic OT problem in order to enforce transport only inside the same class label. The idea is to avoid transport plans where, for example, a data sample x_j^t receives the same amount of mass from samples x_i^s and x_k^s, where sample i has a class label different than sample k. Therefore, the OT problem becomes:

$$\gamma^* = \underset{\gamma \in \Pi(\mu_s, \mu_t)}{\text{argmin}} \overline{OT}_{c,\varepsilon}(\mu_s, \mu_t) + \eta \sum_j \sum_l ||\boldsymbol{\gamma}(\mathcal{I}_l, j)||_q^p, \tag{4.26}$$

where $\eta > 0$, $\mathcal{I}_l$ is the set of indices of samples with class label l, and $\boldsymbol{\gamma}(\mathcal{I}_l, j)$ is a vector containing the coefficients of the jth column of the transport plan matrix $\boldsymbol{\gamma}$ associated to class l.

Courty et al. [17] used $p = 1/2$ and $q = 1$ for algorithmic reasons. In a follow up paper, Courty et al. [18] suggested using $p = 1$ and $q = 2$, also

known as group-lasso regularizer, and devised an efficient algorithm for solving (4.26). The authors also proposed to use Laplacian regularization, but the experimental results comparing the three types of regularization showed that the group-lasso outperformed the others.

Rousselle and Canu [66] proposed two new algorithms for the case where there is some known labels in the target data. In both algorithms the target data with known labels (X_k) is transported to the target data with unknown label (X_u) using the OT formulation from Courty et al. [17]. The first algorithm consisted in splitting the data according to class label, and then finding an OT plan for couple (X_s^l, X_k^l), where X_s^l is the portion of the source dataset with label l, and X_k^l is the portion of the target dataset with known label l. The second algorithm consisted in finding a plan between X_s to X_k without caring for the classes, and then performing a post processing on the optimal transport plan matrix γ such that each entry of the matrix with class-crossing is sent to zero, and the matrix is renormalized.

Joint Distribution Domain Adaptation

The OTDA algorithm assumed that the labels were transported along the features, which might not always be true. Courty et al. [19] devised the model named Joint Distribution Domain Adaptation (JDOT), where such restriction was relaxed. The main idea of the model is to align the feature/label space of the source and target, at the same time the classifier f is learned.

The JDOT algorithm seeks to solve the following optimization problem:

$$\begin{aligned} &\min_{f \in H} OT_c(\hat{\mathcal{P}}_s, \hat{\mathcal{P}}_{t,f}) \\ &\text{s.t.} \quad c((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)) = \alpha ||\mathbf{x}_1 - \mathbf{x}_2||^2 + \mathcal{L}(y_1, y_2), \end{aligned} \tag{4.27}$$

where $\alpha > 0$, $\mathcal{L}(y_1, y_2)$ is the loss function for a classification task, $\hat{\mathcal{P}}_s$ is the empirical distribution of (X_s, Y_s) and $\hat{\mathcal{P}}_{t,f}$ is the distribution of $(X_t, f(X_t))$. Note that since the target is unlabeled, we use the learned classifier to come up with the labels.

Problem (4.27) can be optimized in an alternative way. First, fixing f, we have to solve an Optimal Transport problem, which can be done using Linear Programming algorithms such as the Simplex, or, we can regularize the problem and solve with the Sinkhorn algorithm. Then, for a fixed transport plan γ^*, we minimize f. Courty et al. [19] suggested the use of either Block Gra-

dient Descent or Gauss-Sidel method. The authors showed that the JDOT method outperformed many state-of-the-art Transfer Learning algorithms, including OTDA.

Damodaran et al. [23] expanded the JDOT model with their DeepJDOT, which uses Deep Neural Networks to learn a representation g of the data at the same time it learns the classifier f. The new optimization problem then becomes

$$\min_{\mathbf{P}\in\mathcal{P},f,g} \frac{1}{n^{(s)}} \sum_i \mathcal{L}_s(y_i^{(s)}, f(g(x_i^{(s)}))) + \sum_{i,j} \mathbf{P}_{i,j}(\alpha|g(x_i^{(s)}) - g(x_j^{(t)})|^2 + \lambda_t \mathcal{L}_t(y_i^{(s)}, f(g(x_j^{(t)}))), \quad (4.28)$$

where $\mathcal{L}_s$ and $\mathcal{L}_t$ are the loss functions used in the source and target respectively. The authors proposed to solve this minimization problem with stochastic approximations using gradient descent in both the source and target. Experimental results showed that DeepJDOT consistently outperformed the other Deep Learning methods to which it was compared, such as DeepCORAL [83].

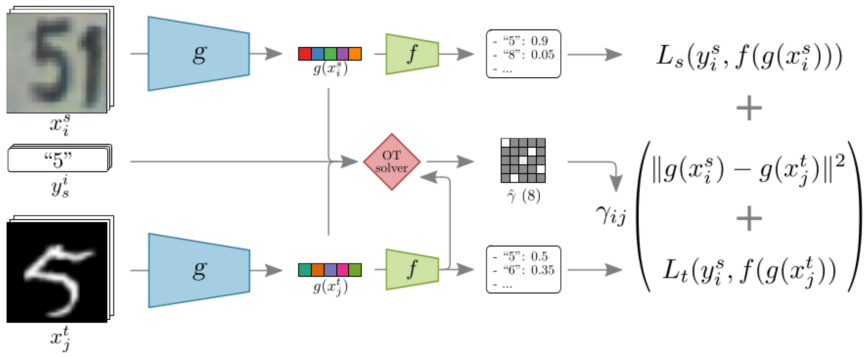

Figure 4.10: Figure from Damodaran et al. [23] illustrating the DeepJDOT model.

Another model based on JDOT was proposed by Redko et al. [62]. While JDOT focuses on covariate shift (Figure 6.3), the model proposed by Redko et al. [62], called Joint Class Proportion and Optimal Transport (JCPOT), focuses on the problem of target shift (Figure 6.2)[5]. The main idea of the

[5]Target shift is also known as prior shift.

model is to reweigh samples in the source to compensate the discrepancy in class proportion between the sources and the target. Both the Optimal Transport plans and the class proportion in the target are estimated jointly by solving a constrained Entropic Wasserstein Barycenter problem.

Let μ^t be the empirical distribution of the target dataset, to which we don't know the label, $\mu_1^s, ..., \mu_k^s$ the empirical distribution for the sources to which we do know the label, and n_i^s the number of samples in each source data. For each source, define $\mathbf{h}_i$ to be the vector containing the proportion of each class, and $\mathbf{D}_{h_i}$ a linear transformation such that $\mathbf{D}_{h_i}\mathbf{h}_i = [\frac{1}{n_i^s}, ..., \frac{1}{n_i^s}]^{\mathrm{T}}$ which is the vector of equal weights of the empirical distribution μ_i^s[6]. This allows us to write the source distribution as a function of the proportions of each label as $\mu_s = (\mathbf{D}_i\mathbf{h}_i)\boldsymbol{\delta}_{\mathbf{X}_i}$. Thus, the optimization for the Barycenter problem for JCPOT becomes

$$\underset{\mathbf{h}\in\Delta^l}{\operatorname{argmin}} \sum_{j=1}^{k} \lambda_j W_{1,\varepsilon}\left((\mathbf{D}_i\mathbf{h}_i)\boldsymbol{\delta}_{\mathbf{X}_i}, \mu^t\right), \tag{4.29}$$

where $\Delta^l := \{\alpha \in \mathbb{R}_+^l \ : \ \sum_{i=1}^l \alpha_i = 1\}$ and l is the number of classes. Solving (4.29) gives an estimate on the class proportions of the target, and one can reconstruct the sources distributions as $\mu_i^s = (\mathbf{D}_i\mathbf{h}^*)$ wher h^* is the argument that minimizes (4.29). The next step is to find the Entropic OT plan from each μ_i^s to μ. The labels in the target are estimated by calculating the proportion of mass coming from each label in the source, akin to a boosting technique. Figure 4.11 compares the JCPOT with OTDA by Courty et al. [17], which does not take into account the target shift.

Turrisi et al. [87] uses the same modified Wasserstein distance as JDOT, but apply it instead to the case of multiple sources. The authors propose to weight each source according to its proximity to the target. The optimization problem proposed is

$$\min_{\alpha\in\delta^S, f} W_{JDOT}\left(\hat{p}^f, \sum_{s=1}^{S} \alpha_s \hat{p}_s\right), \tag{4.30}$$

where S is the number of sources and δ^S is the simplex of dimension S.

[6]Remember that $\mu_i^s = \frac{1}{n_i^s}\sum_j^{n_i^s} \delta_{x_j^s}$, where each sample has equal weight.

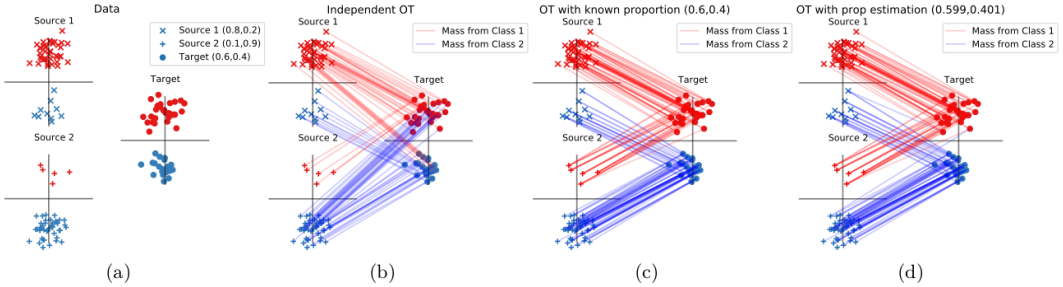

Figure 4.11: Figure from Redko et al. [62] illustrating the importance of proportion estimation for target shift: (a) 2 sources and 1 target with different class proportions; (b) Transport plan from OTDA [17]; (c) Transport plan using OTDA with reweigh based on true class proportions; (d) Transport plan obtained with JCPOT without knowing the true class proportion.

Transport plans obtained from the solution of discrete OT problems cannot be used for out-of-sample data, since they are matrices transporting each data point on which the model was trained. The work of Perrot et al. [60] is perhaps the first to address this issue by proposing a method to estimate the underlying transport map T that approximates the learned optimal transport plan. The authors propose a joint optimization problem, where one seeks to learn a transformation T regularized by a transport plan. The optimization problem is

$$\underset{T\in\mathcal{H},\gamma\in\Pi(\mu_s,\mu_t)}{\operatorname{argmin}} \frac{1}{n^{(s)}d_t}||T(\mathbf{X}^{(s)}) - n^{(s)}\gamma\mathbf{X}^{(t)}||^2 + \frac{\lambda_\gamma}{\max(\mathbf{C})}\langle\gamma,\mathbf{C}\rangle + \frac{\lambda_T}{d_s d_t}R(T), \tag{4.31}$$

where d_s, d_t are the dimensions of the source and target, λ_s and λ_t are hyperparameters, $R(\cdot)$ is a generic regularization term. For the space $\mathcal{H}$ of possible transport maps, the authors experimented with the space of linear maps, and the space of non-linear maps using the kernel trick.

Geometric Dataset Distance

Models such as JDOT assume that the label set from the source and the target are the exact same, but this is not always true. For example, one might want to transfer knowledge from a model trained on the MNIST

dataset to perform handwritten digits recognition, to instead perform handwritten letters recognition. The work of Alvarez-Melis and Fusi [3] proposes a very elegant way of measuring the distance between dataset (i.e. features and labels). This new metric is a hierarchical Optimal Transport distance, because it uses an OT metric inside another OT metric. The authors named this new metric as Optimal Transport Dataset Distance (OTDD) and is given by

$$d_z((x, y), (x', y')) := (d_x(x, x')^p + W_p(\alpha_y, \alpha_{y'}))^{1/p}, \tag{4.32}$$

where x is a sample in the feature set, y is a sample from the label set, d_x is a metric defined in the feature space, and $\alpha_y(x) := P(X = x \mid Y = y)$ is the conditional distribution of obtaining a sample x given a label y. Note that to calculate d_z one needs to solve an OT problem inside another OT problem, thus becoming quite computationally expensive. Alvarez-Melis and Fusi [3] proposed different methods for dealing with such computational complexity, such as using the Sinkhorn divergence, and considering α_y to be Gaussian which leads to a closed form solution. Then, the authors showed how OTDD correlated with transferability capacity of different datasets, such that it could be used as criterion for source dataset selection to perform transfer learning. Figure 4.12 exemplifies the capacity of OTDD to the use of label-agnostic OT distance.

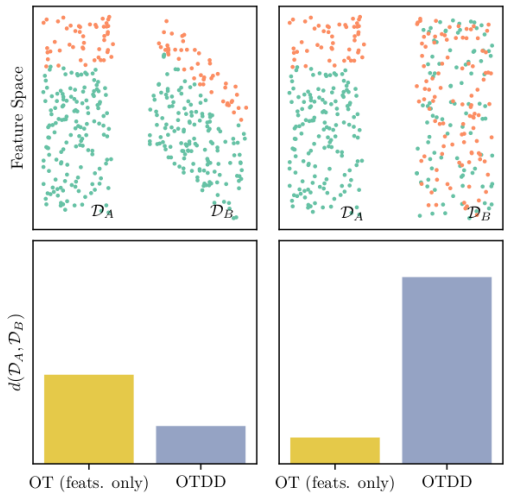

Figure 4.12: Figure from Alvarez-Melis and Fusi [3] illustrating the OTDD distance.

Transfer Learning with Adversarial Networks

In Transfer Learning, there are many methods that use adversarial learning to train a discriminator to distinguish source and target, while a generator tries to learn a representation of the data. Lee et al. [47] proposed the use of Sliced-Wasserstein distance with the task-specific adversarial learning model, introduced by Saito et al. [67]. They named their model Sliced-Wasserstein Discrepancy (SWD).

The model consists of a generator and two discriminators that are initialized with different weights. Hence, the model produces two different decision boundaries. The training then consists of freezing the parameters of the generator, and maximizing the discrepancy between the decision boundaries while keeping the classification accuracy of each discriminator. Then, freezing the discriminators, and training the generator to minimize the discrepancy. Lee et al. [47] contribution consists in using the Sliced-Wasserstein distance to measure such discrepancy. The method is summarized in Figure 4.13.

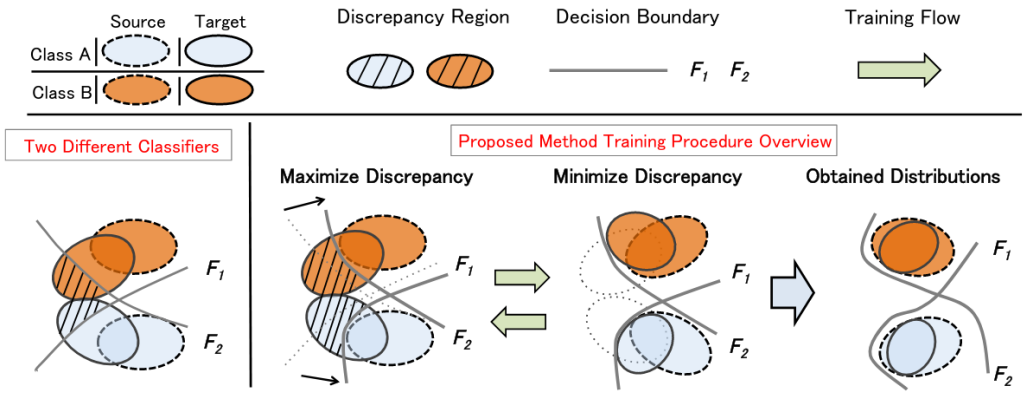

Figure 4.13: Figure from Saito et al. [67] illustrating the task-specfic adversarial learning algorihtm.

.

Model Ensemble and Feature Selection

Shen et al. [78] used a Neural Network with structure similar to WGAN, where a discriminator was used to estimate the Wasserstein distance between

the source and target samples, while another network optimizes a feature extractor. This model was named Wasserstein Distance Guided Representation Learning (WDGRL).

Another very unique approach was proposed by Singh and Jaggi [79], where it was proposed a method of ensembling neural networks, which successfully yielded "one-shot" knowledge transfer. The method consisted in using the Wasserstein barycenter as a way of averaging the weights in the neural networks, performing Optimal Transport layer by layer.

Similarly to Singh and Jaggi [79], the work of Li et al. [49] also uses OT on a Neural Network architecture, but instead of finding a barycenter for model parameters, this work proposes to use the discrepancy in the source network and the target network as a regularizer when training the target network. Again, the use of OT comes to metrize the discrepancy in the networks. Figure 4.14 sketches how the algorithm works.

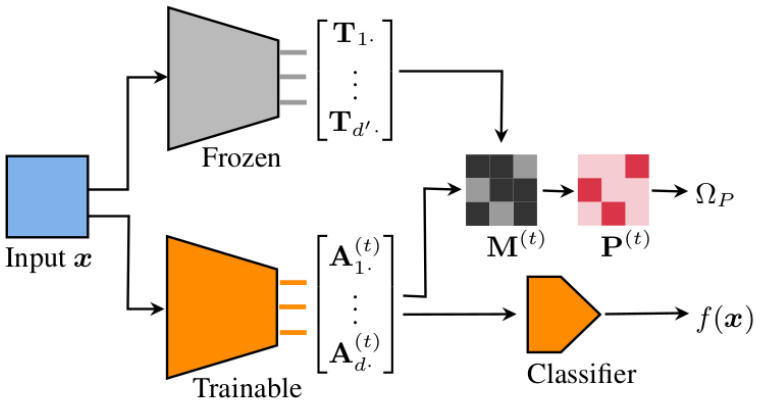

Figure 4.14: Figure from Li et al. [49] illustrates the process of training the target Neural Network by adding the regularization term based on the discrepancy with the parameters of the source Network. In the image, $\mathbf{M}$ is the transport cost matrix, $\mathbf{P}$ is the optimal transport plan matrix, and Ω_P represents the regularization term.

Gautheron et al. [31] proposed to use Entropic OT for feature selection for unsupervised Domain Adaptation. The authors argue that every dataset is composed of cartesian product of a feature space $\mathcal{F}$ (i.e. column space) and an instance space (i.e. row space). The main idea consists in finding

an Optimal Transport plan $\gamma^{\mathcal{F}}$ from the feature space of the source to the feature space of the target, such that the k features with higher transport rate from and to itself, i.e. $\gamma^{\mathcal{F}}_{i,i}$ are selected. Since the number of rows is usually different between source and target, a sampling method is proposed in order to select rows such that both end up with the same size, thus enabling the use of OT, where each feature (column) is a point mass δ_{f_i} with dimension equal to the number of rows.

Transfer Learning Across Incomparable Spaces

The work of Yan et al. [97] tackled the problem of Heterogeneous Domain Adaptation (HDA), in which the space of the source and the target is different (e.g. source is in $\mathbb{R}^3$ and target is in $\mathbb{R}^2$). Due to the differing spaces, the authors use the entropic Gromov-Wasserstein metric (EGW) to find the optimal transport between the source and the target. Their method, called Semi-supervised Gromov-Wasserstein (SGW), is similar to OTDA proposed by Courty et al. [17], and it also works by transporting the source to the target and then performing the training.

4.4 Supervised Learning

Supervised Learning is the most common scenario in ML, and consists in the learner receiving a batch of labeled data as training set, thus making prediction on the unlabeled set [55]. Although it was the category with the least amount of papers, it was the most diverse in terms of applications.

4.4.1 Classifcation and Regression

Classification and Regression are two of the main tasks of Supervised Learning. Classification accounts for the cases where the prediction set is finite, while regression is used when the prediction set is not finite.

Word Mover's Distance

Similar to the WGAN, the work of Kusner et al. [45] is one of the first successful cases of applying Optimal Transport to Machine Learning, with more than a thousand citations today. This work introduced the Word Mover's Distance (WMD), which is a distance function between documents using the 1-Wasserstein distance. Consider $\mathbf{X} \in \mathbb{R}^{d \times n}$ a *word2vec* matrix, where each column represents a word in the corpus. Each document is represented as a normalized *bag-of-words*, that is, it's a vectors of size n, where each element represents the percentage of times a word appeared in the document in terms of the total number of words in the document . For example, suppose that word "blue" is the 10th column of matrix $\mathbf{X}$, and in document $\mathbf{d}^{(j)}$ there are 100 words, and "blue" appeared 20 times, hence, $d_{10}^{(j)} = \frac{20}{100}$.

The authors define the cost function to be the Euclidean distance between words, that is $c(i, j) = ||\mathbf{x}_i - \mathbf{x}_j||_2$ where $\mathbf{x}_i, \mathbf{x}_j$ are the columns of $\mathbf{X}$. Since each document can be seen as an empirical probability measure, the WMD is defined as the 1-Wasserstein distance between each document.

Kusner et al. [45] evaluated the WMD using it on kNN classification for document categorization tasks. The WMD outperformed 7 state-of-the-art alternative document distances in 6 of the 8 classification tasks that were performed in the paper.

Huang et al. [38] extended the Word Mover's Distance by using the label of each document, creating a new metric named Supervised Word Mover's Distance (S-WMD). While the WMD only uses the feature space to measure the distance between each document, the S-WMD incorporates information

of the label of each document, enforcing that documents with the same label are closer and with different labels are distant. To do this, the authors trained a linear mapping A and an importance vector $\mathbf{w}$ to re-weight the word histograms of each document. Thus, the new distance becomes:

$$D_{\mathrm{A},\mathbf{w}}(\mathbf{d}^{(i)}, \mathbf{d}^{(j)}) := \min_{\mathbf{P}\geq 0} \sum_{i,j=1}^{n} \mathbf{P}_{i,j} ||\mathrm{A}(\mathbf{x}_i - \mathbf{x}_j)||_2^2 \tag{4.33}$$
$$\text{s.t.} \quad \mathbf{P}\mathbf{1}_n = \tilde{d}^{(i)}, \ \mathbf{P}^{\mathrm{T}}\mathbf{1}_n = \tilde{d}^{(j)}.$$

The authors proposed the use of entropic regularization to solve the Optimal Transport problem in order to make the training process less computationally costly.

Wasserstein Distance for Label Space Similarity

In classification problems, it can be the case that the underlying label space is comprised of some structure, which might be of interest when making predictions. For example, when doing digits image recognition, one might want to enforce actual numerical difference, such the fact that predicting a 2 instead of a 4 is more costly then predicting a 3 instead of a 4.

Frogner et al. [29] were the first to address this kind of problem by using Optimal Transport. The authors developed a loss function for multi-label learning using the Entropic Wasserstein cost. Note that when the output of a model is a probability distribution, the gradient for the entropic regularized Wasserstein has a closed formula given by:

$$\frac{\partial W_p^p}{\partial h(x)} = \frac{\log \mathbf{w}}{\lambda} - \frac{\log \mathbf{w}^{\mathrm{T}}\mathbf{1}}{\lambda \mathbf{K}}\mathbf{1}, \tag{4.34}$$

where $h(x)$ is the learned classifier (e.g. a neural network), $\mathbf{w}$ and $\mathbf{K}$ are the terms presented in the Sinkhorn Algorithm [1]. Hence, one can use stochastic gradient descent to train a Neural Network. The experimental results showed that the use of Entropic Wasserstein as loss function increased the smoothness in predictions on the output space such that the model misclassified, it tended to guess a category semantically closer to the real label.

While Frogner et al. [29] used the Entropic Wasserstein **cost** to approximate the Wasserstein distance as loss function, the work of Luise et al. [51]

used the Entropic Wasserstein **distance**[7], and obtained better experimental results. Although, the authors note that the use of Entropic Wasserstein distance increases the computational complexity of the algorithm.

Consider the problem of model stacking, where one is interested in combining multiple forcasts in order to improve the final prediction. There are simple methods, such as taking the arithmetic average, and there are more complex methods such as training a decision tree to perform the final prediction based on the output of each model. Under the assumption that output space has some sort of similarity structure, Dognin et al. [25] proposed the use of Wasserstein barycenter on the output predictions. Their experimental results showed that the barycenter ensemble outperformed the use of arithmetic and geometric average.

WGAN for Regression

Manchanda et al. [53] uses the WGAN for regression, but proposes a method for solving the primal problem, instead of the dual, thus avoiding the problems with enforcing the 1-Lipschitz condition. The model takes as input a sample $\mathbf{x} \in \mathbb{R}^n$ of the feature space and a noise vector $\mathbf{z} \in \mathbb{R}^d$, and tries to reproduce a sample $\mathbf{y} \in \mathbb{R}^m$ that looks like it was originated from $p(\mathbf{y} \mid \mathbf{x})$. Since the model assumes that both $\mathbf{x}$ and $\mathbf{y}$ are continuous, estimating $p(\mathbf{y} \mid \mathbf{x})$ becomes non-trivial since there will only be one sample for each distribution. Thus, the authors propose instead to sample from the joint distribution and constraint the transport plans to only map $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ to $(\mathbf{x}^{(j)}, \mathbf{y}^{(j)})$ if they are close in the feature space. Since the computation of the Wasserstein distance involves the solution of a LP, the authors sparsify the cost matrix by precomputing the $\mathbf{x}$-neighbors thus removing high-cost edges, and using an efficient method for sparse linear assignment. Experiments conducted on synthetic data with non-Gaussian noise showed that this method was able to outperform Deep Neural Networks, XGBoost, Regression with Gaussian Process, among others.

Wasserstein Multi-task Regression

Janati et al. [39] proposed the use of Entropic Wasserstein cost as a

[7]The Entropic Wasserstein cost incorporates the regularization term in the cost, while the Entropic Wasserstein distance uses the transport plan from the regularized OT problem to calculate the cost of transportation. Check Section 3.1.4.

regularizer for multi-task regression learning. The problem of multi-task regression can be posed as jointly estimating the coefficients $\theta^{(1)}, ..., \theta^{(T)} \in \mathbb{R}^p$, such that $Y^{(t)} = X^{(t)}\theta^{(t)} + \epsilon^{(t)}$, where each $t \in \{1, ..., T\}$ symbolizes a task and $\epsilon^{(t)}$ are Gaussian noise. Hence, the objective function can be written as:

$$\min_{\theta^{(1)},\cdots,\theta^{(T)}} \frac{1}{2n} \sum_{t=1}^{T} ||X^{(t)}\theta^{(t)} - Y^{(t)}||_2^2 + J(\theta^{(1)}, \cdots, \theta^{(T)}) \tag{4.35}$$

In this work, the authors proposed the following formulation for the regularization function J:

$$J(\theta^{(1)}, \cdots, \theta^{(T)}) := \frac{\xi_1}{T} \sum_{t=1}^{T} [W_{1,\varepsilon}(\theta_+^{(t)}, \bar{\theta}_+) + W_{1,\varepsilon}(\theta_-^{(t)}, \bar{\theta}_-)] + \frac{\xi_2}{T} \sum_{t=1}^{T} ||\theta^{(t)}||_1, \tag{4.36}$$

where $\xi_1, \xi_2 > 0$, $\theta_+ := \max\{0, \theta\}$, $\theta_- := \{0, -\theta\}$ and $\bar{\theta}$ is the Wasserstein barycenter of the parameters $\{\theta^{(1)}, ..., \theta^{(T)}\}$. This problem is actually an Unbalanced OT problem, since $\theta_+^{(t)}$ are not normalized. Janati et al. [39] showed using experiments with both real and synthetic data that the WMT model obtained better results than other multi-task regression methods such as Multi-Level Lasso.

Optimal Transport for Graph Classification

When the input data is a graph, one can use the so called Graph Neural Networks (GNN). Bécigneul et al. [7] developed a new GNN using Optimal Transport. Their model, which was named OT-GNN, used the Wasserstein distance to measure the discrepancy between the point cloud of embeddings from the Graph Neural Network and the prototype point clouds. Such Wasserstein distances then becomes the input to a final Multi-layer Perceptron which performs the regression/classification task. The authors also added contrastive regularization for improving the model's performance. Four computational experiments were conducted on molecular property prediction, with the OT-GNN model outperforming the state-of-the-art baseline models.

With a very different approach, Vayer et al. [89] introduced the Fused Gromov-Wasserstein distance (3.29) in order to measure the distance of structured objects such as graphs. They performed classification on these graphs using a Support Vector Machine algorithm with indefinite kernel

matrix $e^{-\xi FGW}$. The authors compared their SVM model with four other state-of-the-art graph kernel methods. Their model either reached or outperformed the other models in all the 12 benchmark datasets with which they experimented.

Barycentric Coordinate Regression

Bonneel et al. [11] devised what they called Barycentric Coordinate Regression (BRC). This new problem formulation is a histogram regression, where the input data is a discrete measure $q \in \Sigma_N$, and for a set of discrete measures (p_s), the goal is to find a probability vector $\lambda \in \Sigma_s$ such that the Wasserstein barycenter $\beta_\lambda((p_s)) \approx q$. This is an inverse problem in relation to the original barycenter problem, in which one already knows λ and is instead interested in finding the barycenter. The authors used the Entropic Regularized Wasserstein cost instead of pure Wasserstein, which enabled the creation of a faster algorithm for solving the regression. The work empirically showed that the BRC can be successfully used for tackling problem such as inferring missing data.

Wasserstein Discriminant Analysis

Flamary et al. [27] created a new method named Wasserstein Discriminant Analysis (WDA), which is a supervised linear dimensionality reduction algorithm. This method uses Entropic Regularized Wasserstein distance and searches for a projection linear map $\mathrm{A} : \mathbb{R}^d \to \mathbb{R}^p$ where p is much smaller then d. The algorithm optimizes

$$\max_{\mathrm{A} \in \Delta} \frac{\sum_{l,l'>l} W_\lambda(\mathrm{A}\mathbf{X}^{(l')}, \mathrm{A}\mathbf{X}^{(l)})}{\sum_l W_\lambda(\mathrm{A}\mathbf{X}^{(l)}, \mathrm{A}\mathbf{X}^{(l)})}, \tag{4.37}$$

where $\Delta := \{\mathrm{A} = [\mathbf{a}_1, ..., \mathbf{a}_p] \ : \ \mathbf{a}_i \in \mathbb{R}^d, ||\mathbf{a}_i||_2 = 1 \text{ and } \mathbf{a}_i^{\mathrm{T}}\mathbf{a}_j = 0 \ \forall i \neq j\}$, and $\mathbf{X}^{(l)}$ represents the data with label l. Hence, note that the optimization problem is maximizing the ratio between the distance of inter class and intra-class. Compared to other popular methods such as PCA, FDA (Fisher Discriminant Analysis), LFDA (Local Fisher Discriminant Analysis), the Wasserstein Discriminant Analysis obtained the best overall performance for kNN classification.

Wasserstein Market Basket

The market basket problem consists in trying to predict which products a client will purchase in the next visit to a store, using as input the purchase history of all the clients. Kraus and Feuerriegel [44] developed a new algorithm for such problem using Optimal Transport. The main idea behind the proposed algorithm is to find the most similar subsequence of market baskets that matches the history of the target client, and then use it to predict what the client will buy next. To achieve this, the algorithm does the following: first creates an embedding of the products; then uses the Wasserstein metric to measure the distance between market baskets; with these distances, the closest subsequence of market baskets is found using the k-Nearest Neighbors algorithm together with Subsequence Dynamic Time Wrapping (kNN-SDTW). Experimental results showed that the model outperformed the other baseline models.

4.4.2 Ranking

According to Shalev-Shwartz and Ben-David [77]:

> Ranking is the problem of ordering a set of instances according to their relevance." A typical application is ordering results of a search engine according to their relevance to the query. Another example is a system that monitors electronic transactions and should alert for possible fraudulent transactions. Such a system should order transactions according to how suspicious they are.

Wasserstein for Ranking Learning

The work of Yu et al. [99] tackled the *learning to rank* problem by creating a new method called WassRank. The goal of the algorithm is to predict the relevance of a list of documents given a query. The proposed method used the Entropic Regularized Wasserstein cost to measure the discrepancy between the predicted and the real relevance vectors. The model consists in training a Neural Network that receives a query and returns a vector of relevance score for each document, which are then normalized. Thus, the Entropic Wasserstein distance is calculated, using a custom cost function devised by the authors. Using two benchmark datasets, the empirical results showed that the WassRank outperformed other four listwise ranking methods (i.e., LambdaRank, ListNet, ListMLE and ApxNDCG).

Another ranking learning problem, but with a slightly different objective, is the top-K recommendation problem. Given a list of users $i \in \mathcal{D}$, its assumed that each user has an ordered list of preferred items $\mathcal{D}_i := \{I_1, ..., I_{|\mathcal{D}_i|}\}$, where each I_j is an item. Each training sample constitutes a list $\mathcal{S}_i$ of already known preferred items and $\mathcal{T}_i$ of unknown items (i.e. $\mathcal{S}_i \cap \mathcal{T}_i = \varnothing$, $\mathcal{S}_i \cup \mathcal{T}_i = \mathcal{D}_i$). The goal is to recommend an ordered list $\mathcal{X}_i$ that is closer to $\mathcal{T}_i$, such that $\mathcal{S}_i \cap \mathcal{X}_i = \varnothing$ and $|\mathcal{X}_i| \leq k$. Ma et al. [52] proposed a model for this problem that works by embedding the users and items in an Euclidean space with latent dimension h, and assumes that each one is represented by a Normal distribution with mean m and co-variance Σ which are both learned. Then, the recommendation is done for each user i by evaluating the Wasserstein distances between the user and each item, where the k items closer to the user and not in $\mathcal{S}_i$ are recommend. The use of Normal distribution allows for an easy computation of the Wasserstein distance (3.6). A comparison was done using five real-world datasets against five state-of-the-art methods, and the proposed model outperformed the others by 4-22% in terms of recall on the top-K recommendation.

Differentiable Ranking Operators with Optimal Transport

Many Machine Learning applications rely on sorting, for example the k-Nearest Neighbors. Yet, a common problem of such operators is that they are not differentiable, hence, one cannot use ranking as a loss function with gradient descent. To address such issues, Cuturi et al. [22] used Entropic Regularized Optimal Transport in order to define a differentiable sorting operator. The link between OT and sorting is quite clear in 1-D, where the Optimal Transport plan consists in moving the most left mass in μ to the most left mass in ν (check section 3.1.1). Hence, for a vector $\mathbf{x} \in \mathbf{R}^n$ unsorted and a sorted auxiliary vector $\mathbf{y} \in \mathbf{R}^m$, one can construct an OT problem where the μ is a discrete measure with mass proportional to the values of $\mathbf{x}$ and supported on the vector's indexes, and ν is the equivalent for $\mathbf{y}$. Cuturi et al. [22] then shows how to construct the sorting and ranking operators as functions of the optimal transport plan. Using regularization and the Sinkhorn algorithm with fixed number of iterations, the sorting algorithm becomes easily differentiable, thus allowing its use for supervised learning tasks.

While Cuturi et al. [22] proposed a differentiable sorting operator, Xie et al. [95] used Entropic OT to create a differentiable version of the top-

K operator. While a sorting operator returns the sorted array, the top-K operator returns only the first k largest or smallest elements in an array. To obtain the k largest elements of $\mathbf{x}$, make $\mu = \sum_{i=0}^{n} \delta_i 1/n$ where n is the number of elements of x, and $\nu = \delta_0 k/n + \delta_1(n-k)/n$. Thus, the $(n-k)$ smallest x_i will be transported to position 0 and the k largest to position 1. The top-K operator can be written as $A = n\gamma^* \cdot [0,1]^{\mathrm{T}}$ where γ^* is the optimal transport plan. With entropic regularization A becomes differentiable in terms of positions $\mathbf{x}$. Hence, one can, for example, use the top-K operator to train a Neural Network based on kNN.

4.4.3 Adversarial Learning and Robust Modelling

Adversarial Learning and Robust Modelling are two related subjects in Machine Learning. While Adversarial Learning focuses on devising methods for producing adversarial attacks to ML models, Robust Modelling consists in finding way to make the ML models robust to this type of attacks. A very common example in this regard are the adversarial attacks to Convolutional Neural Networks in which small noise is added to the images, and although the noise is almost imperceptible to the human eye, it is still able to fool the CNN.

Distributional Robustness in the Wasserstein Ball

Shafieezadeh Abadeh et al. [75] proposed a distributionally robust logistic regression, where the Wasserstein distance is used to define the ball radius of the space of acceptable probability distributions. While stochastic programming framework assumes that the uncertainty is governed by a known probability distribution, and robust optimization framework aims to minimize the worst-case scenario. The distributionally robust optimization framework (DRO) seeks to minimize the worst-case probability distribution given a family of distributions. In this work, the family of distributions used consists in all the probability measures that have a Wasserstein distance at most ϵ from the empirical distribution. Therefore, the new logistic regression model seeks to optimize the following problem:

$$\inf_{\beta} \sup_{\mu \in B_\epsilon(\nu_N)} \mathbb{E}_\mu[l_\beta(x,y)], \tag{4.38}$$

where $\nu_N := \frac{1}{N}\sum_{i=1}^{N} \delta_{(x_i,y_i)}$ is the empirical distribution of the training

dataset, $B_\epsilon(\hat{\mu}_N) := \{\mu \in \mathcal{P} \ : W_1(\mu, \nu_N) \leq \epsilon\}$ is the Wasserstein ball of radius ϵ, and l_β is the logloss function[8]. To solve this new problem, the authors devised a new convex tractable formulation. The model was tested in both synthetic and real world examples. In both cases, the model outperformed the vanilla logistic regression and the regularized logistic regression in terms of correct classification ratio in test sets and the conditional value-at-risk in out-of-sample logloss distribution.

Shafieezadeh-Abadeh et al. [76] extends the work of Shafieezadeh Abadeh et al. [75] by introducing many other distributionally robust models (e.g. robust linear regression, support vector regression, quantile regression, robust linear classification, ...) using the Wasserstein ball. This work also presents a new perspective on regularization theory by showing that the classical regularized learning models emerge as special cases of the Wasserstein distributionally robust framework as the ground cost function changes.

Sinha et al. [80] devised a method for training Neural Networks with small to moderate amounts of robustness by penalizing the perturbation of the data distribution inside the Wasserstein ball. For this, the authors showed the following equivalence with regards to the distributionally robust optimization problem with Lagragian relaxation and a fixed penalty $\lambda \geq 0$

$$\min_\theta \sup_P \mathbb{E}_P[\ell(\theta; Z)] - \lambda W_c(P, P_0) = \mathbb{E}_{P_0}[\phi_\lambda(\theta; z_0)], \tag{4.39}$$

where $\phi_\lambda(\theta; z_0) = \sup_{z \in \mathcal{Z}} \ell(\theta; z) - \lambda c(z, z_0)$, such that c represents the cost to adversarially perturb the data and ℓ is the loss function. The authors then propose an algorithm to solve such problem, which consists in solving ϕ_y via stochastic gradient ascent on z, and then perform stochastic gradient descent on θ.

Wasserstein Adversarial Attacks

While Sinha et al. [80] used the Wasserstein distance to perturb the underlying data distribution, Wong et al. [92] proposes to use the Wasserstein distance to perturb each example. The idea here is to devise a method for generating adversarial examples that can dissuade the original trained model, but that still resemble the original training dataset very closely. One of the most common methods to do this is the Projected Gradient Descent (PGD), which seeks to find projection of a sample x in a ball $B_\varepsilon(x)$ that maximize

[8] $l_\beta(x, y) = \log(1 + \exp(-y\langle\beta, x\rangle))$.

the loss function, by moving in the direction of gradient that increases the error. The PGD adversarial attack consists in the following iteration:

$$x^{(t+1)} = \text{proj}_{B_\varepsilon(x)} \left(x^{(t)} + \underset{||v|| \leq \alpha}{\text{argmax}} \; v^{\mathrm{T}} \nabla \ell(x^{(t)}, y) \right), \tag{4.40}$$

where α is the step size, ℓ is the loss function (e.g. cross-entropy), y is the label for this training sample x, and proj is the projection operator. The innovation proposed by Wong et al. [92] consists in using the projection in the Wasserstein ball, instead of the more common L^2 or L^∞. The authors devised an efficient method for calculating this projection with the use of Entropic Regularization, performing what they called Sinkhorn Projection.

Wu et al. [94] reformulates the Wasserstein attack proposed by Wong et al. [92], obtaining stronger adversarial examples with a faster algorithm. While the original formulation consisted in maximizing over the loss in the image space, this new proposal maximizes the loss in the transport space, thus searching for a transportation plan such that the cost is inside the Wasserstein ball. Using this new formulation, the authors developed two algorithms for generating the adversarial examples, one using PGD and another using the Frank-Wolfe algorithm. Examples of adversarial images are shown in Figure 4.15.

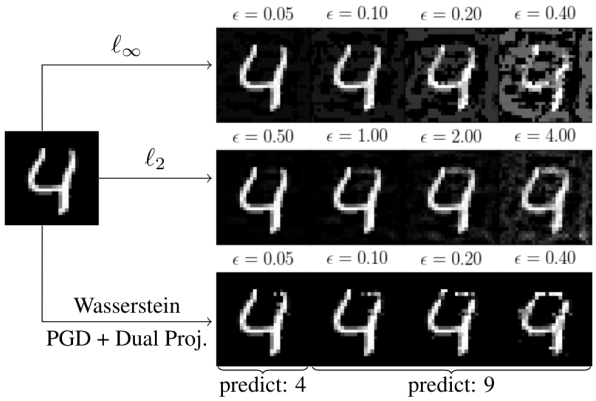

Figure 4.15: Example of adversarial images generated by Wu et al. [94]. The figure makes it clear how different the Wasserstein attack is compared to L^2 or L^∞ attacks.

4.4.4 Fairness Learning

In the learning process, the ML algorithm can implicitly take into account variables that are not considered "fair", such as gender, ethnicity, and others. The idea is that once a model is trained, one can check if it is biased against a set of chosen "sensitive" variables. Hence, Fairness Learning is concerned in trying to guarantee that the model in independent of such set of variables.

In order to enforce fairness in classifier models, Jiang et al. [40] used the 1-Wasserstein to achieve what they called *strong demographic parity* (SDP). Given a sensitive random variable $\mathbf{A}$, the SPD criterion consists in ensuring that the belief variable S, that corresponds to the output of a logistic regression model, is independent of $\mathbf{A}$. The authors introduced two methods to achieve the SPD criterion: by post-processing the model beliefs to match the Wasserstein barycenter; and by penalizing the logistic regression. The barycenter that achieves SPD is given by

$$p_{\bar{S}} = \underset{p^* \in \mathcal{P}(\Omega)}{\operatorname{argmin}} \sum_{\mathbf{a} \in \mathcal{A}} p_{\mathbf{a}} W_1(p_{S_{\mathbf{a}}}, p^*), \tag{4.41}$$

where $\mathcal{A}$ is the set of sensitive variables and $p_{S_{\mathbf{a}}} = P(S \mid \mathbf{A} = \mathbf{a})$. The penalized loss function for the logistic regression is

$$\mathcal{L}_{W_1}(\theta) := \alpha \ell(\theta) + (1 - \alpha)\beta \sum_{\mathbf{a} \in \mathcal{A}} W_1(\hat{p}_{S_{\mathbf{a}}}, \hat{p}_{\bar{S}}), \tag{4.42}$$

where $\hat{p}$ are the empirical distributions, ℓ is the logistic regression loss function, $\beta > 0$ and $\alpha \in [0, 1]$.

Risser et al. [64] proposed the use of 2-Wasserstein regularized logistic regression for addressing the same fairness problem. Although, their model assumes that the sensitive variable is binary, hence, the regularization term can be computed without the need to estimate the barycenter. The new penalized loss function thus becomes

$$\mathcal{L}_{W_1}(\theta) := \ell(\theta) + \beta W_2^2(\hat{p}_{S_0}, \hat{p}_{S_1}), \tag{4.43}$$

where $\hat{p}_{S_0}$ and $\hat{p}_{S_1}$ are the empirical distributions of the output variable S in case the sensitive variable is equal to 0 or equal to 1, respectively.

Chapter 5

Conclusion

Our main goal was to provide a detailed review of the applications of Optimal Transport to Machine learning, serving as a guide to new ML researchers interested in understanding how OT can be used to improve Machine Learning algorithms.

By searching for papers relating Optimal Transport to Machine Learning, a total of 444 papers were initially collected from the Microsoft Academic platform. From this initial collection, the papers were classified and 200 were considered to be on the topic of interest for this dissertation, i.e. "Algorithms for Machine Learning with Optimal Transport". A final selection of 70 papers were chosen to be part of the review. This accounted for roughly 94% of all citations for the 200 papers in the category of "Algorithms for Machine Learning".

Before going into the actual literature review, we presented both theoretical and computational aspects of Optimal Transport. We selected the main topics necessary for understanding the ML applications in the review.

Based on the literature in the subject, we proposed a taxonomy for classifying the applications according to the OT problem formulation and how is was used in the algorithm. Categorizing the papers based on this new taxonomy and on the Machine Learning task (e.g. Transfer Learning, Generative Modeling, Representation Learning), we were able to provide an overview of how these two fields have been combined. For example, it became clear that optimal transport plans are used mostly on Transfer Learning, while the other areas of Machine Learning tend to use the optimal transport cost, which defines a distance metric between probability distributions.

Finally, we concluded the main objective of this dissertation by reviewing

the 70 selected papers, which covered applications in all three sub-fields of Machine Learning, i.e. Supervised, Unsupervised and Semisupervised Learning.

Bibliography

[1] URL `https://academic.microsoft.com/home`.

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

[3] David Alvarez-Melis and Nicolò Fusi. Geometric dataset distances via optimal transport. *arXiv preprint arXiv:2002.02923*, 2020.

[4] David Alvarez-Melis and Tommi S Jaakkola. Gromov-wasserstein alignment of word embedding spaces. *arXiv preprint arXiv:1809.00013*, 2018.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[6] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

[7] Gary Bécigneul, Octavian-Eugen Ganea, Benson Chen, Regina Barzilay, and Tommi Jaakkola. Optimal transport graph neural networks. *arXiv preprint arXiv:2006.04804*, 2020.

[8] 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.

[9] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[10] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

[11] Nicolas Bonneel, Gabriel Peyré, and Marco Cuturi. Wasserstein barycentric coordinates: histogram regression using optimal transport. *ACM Trans. Graph.*, 35(4):71–1, 2016.

[12] Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, Carl-Johann Simon-Gabriel, and Bernhard Schoelkopf. From optimal transport to generative modeling: the vegan cookbook. *arXiv preprint arXiv:1705.07642*, 2017.

[13] Charlotte Bunne, David Alvarez-Melis, Andreas Krause, and Stefanie Jegelka. Learning generative models across incomparable spaces. In *International Conference on Machine Learning*, pages 851–861. PMLR, 2019.

[14] Jiezhang Cao, Langyuan Mo, Yifan Zhang, Kui Jia, Chunhua Shen, and Mingkui Tan. Multi-marginal wasserstein gan. *arXiv preprint arXiv:1911.00888*, 2019.

[15] Liqun Chen, Shuyang Dai, Chenyang Tao, Dinghan Shen, Zhe Gan, Haichao Zhang, Yizhe Zhang, and Lawrence Carin. Adversarial text generation via feature-mover's distance. *arXiv preprint arXiv:1809.06297*, 2018.

[16] Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046, 2018.

[17] Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 274–289. Springer, 2014.

[18] 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.

[19] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/0070d23b06b1486a538c0eaa45dd167a-Paper.pdf.

[20] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

[21] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International conference on machine learning*, pages 685–693. PMLR, 2014.

[22] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable ranking and sorting using optimal transport. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/d8c24ca8f23c562a5600876ca2a550ce-Paper.pdf.

[23] 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.

[24] Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative modeling using the sliced wasserstein distance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3483–3491, 2018.

[25] Pierre Dognin, Igor Melnyk, Youssef Mroueh, Jerret Ross, Cicero Dos Santos, and Tom Sercu. Wasserstein barycenter model ensembling. *arXiv preprint arXiv:1902.04999*, 2019.

[26] 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.

[27] Rémi Flamary, Marco Cuturi, Nicolas Courty, and Alain Rakotomamonjy. Wasserstein discriminant analysis. *Machine Learning*, 107(12): 1923–1945, 2018.

[28] Rémi Flamary. Optimal transport for machine learning. page 97, November 2019.

[29] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. Learning with a wasserstein loss. *arXiv preprint arXiv:1506.05439*, 2015.

[30] David JH Garling. *Analysis on Polish spaces and an introduction to optimal transportation*, volume 89. Cambridge University Press, 2018.

[31] Léo Gautheron, Ievgen Redko, and Carole Lartizien. Feature selection for unsupervised domain adaptation using optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 759–776. Springer, 2018.

[32] 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.

[33] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

[34] Edouard Grave, Armand Joulin, and Quentin Berthet. Unsupervised alignment of embeddings with wasserstein procrustes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1880–1890. PMLR, 2019.

[35] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[36] Xin Guo, Johnny Hong, Tianyi Lin, and Nan Yang. Relaxed wasserstein with applications to gans. *arXiv preprint arXiv:1705.07164*, 2017.

[37] GM Harshvardhan, Mahendra Kumar Gourisaria, Manjusha Pandey, and Siddharth Swarup Rautaray. A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, 38:100285, 2020.

[38] Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha, and Kilian Q Weinberger. Supervised word mover's distance. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/file/10c66082c124f8afe3df4886f5e516e0-Paper.pdf`.

[39] Hicham Janati, Marco Cuturi, and Alexandre Gramfort. Wasserstein regularization for sparse multi-task regression. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 1407–1416. PMLR, 16–18 Apr 2019. URL `http://proceedings.mlr.press/v89/janati19a.html`.

[40] Ray Jiang, Aldo Pacchiano, Tom Stepleton, Heinrich Jiang, and Silvia Chiappa. Wasserstein fair classification. In *Uncertainty in Artificial Intelligence*, pages 862–872. PMLR, 2020.

[41] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[42] Soheil Kolouri, Phillip E Pope, Charles E Martin, and Gustavo K Rohde. Sliced-wasserstein autoencoder: An embarrassingly simple generative model. *arXiv preprint arXiv:1804.01947*, 2018.

[43] Wouter M Kouw and Marco Loog. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*, 2018.

[44] Mathias Kraus and Stefan Feuerriegel. Personalized purchase prediction of market baskets with wasserstein-based sequence matching. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2643–2652, 2019.

[45] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR, 2015.

[46] Charlotte Laclau, Ievgen Redko, Basarab Matei, Younes Bennani, and Vincent Brault. Co-clustering through optimal transport. In *International Conference on Machine Learning*, pages 1955–1964. PMLR, 2017.

[47] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019.

[48] Na Lei, Kehua Su, Li Cui, Shing-Tung Yau, and Xianfeng David Gu. A geometric view of optimal transportation and generative model. *Computer Aided Geometric Design*, 68:1–21, 2019.

[49] Xuhong Li, Yves Grandvalet, Rémi Flamary, Nicolas Courty, and Dejing Dou. Representation transfer by optimal transport. *arXiv preprint arXiv:2007.06737*, 2020.

[50] Antoine Liutkus, Umut Simsekli, Szymon Majewski, Alain Durmus, and Fabian-Robert Stöter. Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *International Conference on Machine Learning*, pages 4104–4113. PMLR, 2019.

[51] Giulia Luise, Alessandro Rudi, Massimiliano Pontil, and Carlo Ciliberto. Differential properties of sinkhorn approximation for learning with wasserstein distance. *arXiv preprint arXiv:1805.11897*, 2018.

[52] Chen Ma, Liheng Ma, Yingxue Zhang, Ruiming Tang, Xue Liu, and Mark Coates. Probabilistic metric learning with adaptive margin for

top-k recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1036–1044, 2020.

[53] Saurav Manchanda, Khoa Doan, Pranjul Yadav, and S Sathiya Keerthi. Regression via implicit models and optimal transport cost minimization. *arXiv preprint arXiv:2003.01296*, 2020.

[54] Facundo Mémoli. A spectral notion of gromov–wasserstein distance and related methods. *Applied and Computational Harmonic Analysis*, 30(3):363–401, 2011.

[55] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[56] Grégoire Montavon, Klaus-Robert Müller, and Marco Cuturi. Wasserstein training of restricted boltzmann machines. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 3718–3726. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf`.

[57] Vaishnavh Nagarajan and J. Zico Kolter. Gradient descent gan optimization is locally stable. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5585–5595. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/7e0a0209b929d097bd3e8ef30567a5c1-Paper.pdf`.

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

[59] Giorgio Patrini, Rianne van den Berg, Patrick Forré, Marcello Carioni, Samarth Bhargav, Max Welling, Tim Genewein, and Frank Nielsen. Sinkhorn autoencoders. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning*

Research, pages 733–743, Tel Aviv, Israel, 22–25 Jul 2020. PMLR. URL http://proceedings.mlr.press/v115/patrini20a.html.

[60] Michaël Perrot, Nicolas Courty, Rémi Flamary, and Amaury Habrard. Mapping estimation for discrete optimal transport. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4204–4212, 2016.

[61] 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.

[62] 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.

[63] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. A survey on domain adaptation theory. *arXiv preprint arXiv:2004.11829*, 2020.

[64] Laurent Risser, Quentin Vincenot, and Jean-Michel Loubes. Tackling algorithmic bias in neural-network classifiers using wasserstein-2 regularization. *arXiv e-prints*, pages arXiv–1908, 2019.

[65] 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.

[66] Denis Rousselle and Stéphane Canu. Optimal transport for semi-supervised domain adaptation. In *Proceedings*, page 373. Presses universitaires de Louvain, 2015.

[67] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

[68] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the*

IEEE international conference on computer vision, pages 2830–2839, 2017.

[69] Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving gans using optimal transport. *arXiv preprint arXiv:1803.05573*, 2018.

[70] Roman Sandler and Michael Lindenbaum. Nonnegative matrix factorization with earth mover's distance metric. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1873–1880. IEEE, 2009.

[71] Maziar Sanjabi, Jimmy Ba, Meisam Razaviyayn, and Jason D Lee. On the convergence and robustness of training gans with regularized optimal transport. *arXiv preprint arXiv:1802.08249*, 2018.

[72] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[73] 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.

[74] Vivien Seguy and Marco Cuturi. Principal geodesic analysis for probability measures under the optimal transport metric. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper/2015/file/f26dab9bf6a137c3b6782e562794c2f2-Paper.pdf`.

[75] Soroosh Shafieezadeh Abadeh, Peyman Mohajerin Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper/2015/file/cc1aa436277138f61cda703991069eaf-Paper.pdf`.

[76] Soroosh Shafieezadeh-Abadeh, Daniel Kuhn, and Peyman Mohajerin Esfahani. Regularization via mass transportation. *Journal of Machine Learning Research*, 20(103):1–68, 2019. URL `http://jmlr.org/papers/v20/17-633.html`.

[77] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[78] 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.

[79] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22045–22055. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/fb2697869f56484404c8ceee2985b01d-Paper.pdf`.

[80] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

[81] Hannah Snyder. Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104:333–339, 2019.

[82] Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. Wasserstein propagation for semi-supervised learning. In *International Conference on Machine Learning*, pages 306–314. PMLR, 2014.

[83] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[84] Vayer Titouan, Nicolas Courty, Romain Tavenard, and Rémi Flamary. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pages 6275–6284. PMLR, 2019.

[85] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[86] Nilesh Tripuraneni, Michael I Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. *arXiv preprint arXiv:2006.11650*, 2020.

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

[88] user125646 (https://math.stackexchange.com/users/125646/user125646). How to show that the set of all lipschitz functions on a compact set x is dense in c(x)? Mathematics Stack Exchange. URL `https://math.stackexchange.com/q/665686`. URL:https://math.stackexchange.com/q/665686 (version: 2014-02-07).

[89] Titouan Vayer, Laetita Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Fused gromov-wasserstein distance for structured objects: theoretical foundations and mathematical properties. *arXiv preprint arXiv:1811.02834*, 2018.

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

[91] Jake Williams, Abel Tadesse, Tyler Sam, Huey Sun, and George D Montanez. Limits of transfer learning. *arXiv preprint arXiv:2006.12694*, 2020.

[92] Eric Wong, Frank Schmidt, and Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *International Conference on Machine Learning*, pages 6808–6817. PMLR, 2019.

[93] Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3713–3722, 2019.

[94] Kaiwen Wu, Allen Wang, and Yaoliang Yu. Stronger and faster wasserstein adversarial attacks. In *International Conference on Machine Learning*, pages 10377–10387. PMLR, 2020.

[95] Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k with optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20520–20531. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/ec24a54d62ce57ba93a531b460fa8d18-Paper.pdf`.

[96] Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in Artificial Intelligence*, pages 433–453. PMLR, 2020.

[97] Yuguang Yan, Wen Li, Hanrui Wu, Huaqing Min, Mingkui Tan, and Qingyao Wu. Semi-supervised optimal transport for heterogeneous domain adaptation. In *IJCAI*, volume 7, pages 2969–2975, 2018.

[98] Qingsong Yang, Pingkun Yan, Yanbo Zhang, Hengyong Yu, Yongyi Shi, Xuanqin Mou, Mannudeep K Kalra, Yi Zhang, Ling Sun, and Ge Wang. Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. *IEEE transactions on medical imaging*, 37(6):1348–1357, 2018.

[99] Hai-Tao Yu, Adam Jatowt, Hideo Joho, Joemon M Jose, Xiao Yang, and Long Chen. Wassrank: Listwise document ranking using optimal transport theory. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 24–32, 2019.

[100] Xiaofeng Zhang, Jingbin Zhong, and Kai Liu. Wasserstein autoencoders for collaborative filtering. *Neural Computing and Applications*, pages 1–10, 2020.

[101] Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. Adversarially regularized autoencoders. In *International conference on machine learning*, pages 5902–5911. PMLR, 2018.

[102] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 2020.

Chapter 6

Appendix

6.1 Optimal Transport theory - Extra

This section is focused on providing the proofs for the results presented in Chapter 2. The proofs are mainly based on the book "Optimal Transport for Applied Mathematicians" by Santambrogio [72]. We do not prove the measurability of the sets, functions and maps, although it can be indeed shown that the ones presented here are measurable.

6.1.1 On the Existence of Transport Plans

Theorem 2.4.2. *(Santambrogio 1.5) Let X and Y be compact metric spaces. Given $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ and $c : X \times Y \to [0, +\infty]$, if c is lower semi-continuous, then (KP) admits a solution.*

Proof.

This proof follows the same ideas from the proof of Theorem 2.4.1. The only thing we need to prove is that $K(\gamma)$ is l.s.c with respect to weak convergence.

For $c : X \to \mathbb{R} \cup \{+\infty\}$ bounded from below, then, c is l.s.c if and only if there exists a sequence of $k-$Lipschitz functions c_k such that $\forall x \in X$, $\sup_k c_k(x) = c(x)$ (see Lemma 6.2.1).

Since c is indeed l.s.c and bounded from below, we know that $c = \sup_k c_k$, and by the Monotone Convergence Theorem,

$$K(\gamma) = \int c \, d\gamma = \int \sup_k c_k \, d\gamma = \sup_k \int c_k \, d\gamma$$

Note that we also know that c_k are Lipschitz, hence, they are also all continuous and bounded. This implies that $K_k(\gamma) = \int c_k \, d\gamma$ is also bounded and continuous with respect to weak convergence. Therefore, $K(\gamma) = \sup_k K_k(\gamma)$, which implies that $K(\gamma)$ is l.s.c and bounded. By the Weierstrass's Theorem, we conclude that there exists a transport plan γ that minimizes the Kantorovich Problem. □

Theorem 2.4.3. *(Santambrogio 1.7) Let X and Y be Polish (complete and separable) metric spaces. Given $\mu \in \mathcal{P}(X)$, $\nu \in \mathcal{P}(Y)$ and $c : X \times Y \to [0, +\infty]$, if c is lower semi-continuous then (KP) admits a solution.*

Proof.

Let's prove that $\Pi(\mu, \nu)$ is compact. To do this, we prove that $\Pi(\mu, \nu)$ is tight (6.2.3), and therefore, by Prokhorov's Theorem (i) 6.2.3, it is pre-compact. Once this is done, the proof follows in the same manner as Theorem 2.4.1.

Note that since μ and ν are probability measures, then, the families $\{\mu\}$ and $\{\nu\}$ each containing only one element are pre-compact (actually, compact). Since X is Polish, we can use Prokhorov (ii) 6.2.3, to conclude that μ and ν are tight. Hence, for $\epsilon > 0, \exists K_X \subset X$ and $K_Y \subset Y$ both compacts, such that $\mu(X \setminus K_X), \nu(Y \setminus K_Y) < \epsilon/2$.

Next, note that

$$(X \times Y) \setminus (K_X \times K_Y) \subset (X \setminus K_X \times Y) \cup (X \times Y \setminus K_Y)$$

Therefore, for any $\gamma_n \in \Pi(\nu, \mu)$ we obtain

$$\gamma_n((X \times Y) \setminus (K_X \times K_Y)) \leq \gamma_n((X \setminus K_X) \times Y) + \gamma_n(X \times (Y \setminus K_Y))$$

Finally, note that $\gamma_n(A \times Y) = \mu(A)$. Hence,

$$\gamma_n((X \times Y) \setminus (K_X \times K_Y)) \leq \mu(X \setminus K_X) + \nu(Y \setminus K_Y) < \epsilon$$

Which shows that every sequence $\gamma_n \in \Pi(\mu, \nu)$ is tight, concluding our proof.

□

6.1.2 Duality of the Kantorovich Problem

Auxiliary Lemmas

Lemma 6.1.1. *The Kantorovich Problem (2.3.3) is equivalent to:*

$$\inf_{\gamma\in\mathcal{M}_+(X\times Y)}\int_{X\times Y}c(x,y)d\gamma+\sup_{(\phi,\psi)\in B}\left\{\int_X\phi(x)\;d\mu\right.$$
$$\left.+\int_Y\psi(y)\;d\nu-\int_{X\times Y}\phi(x)+\psi(y)\;d\gamma\right\}\quad(6.1)$$

Where $B:=\{\phi\in C_b(X)$ and $\psi\in C_b(Y)\}$.

Proof. Let's suppose that $\gamma\notin\Pi(\mu,\nu)$. Then, without lost of generality, $\exists A\;:\;\mu(A)\neq\gamma(A,Y)$. Hence, can make $\phi(x)=M$ in A and null elsewhere. So,

$$\int_A\phi\;d\mu-\int_A\phi\;d\gamma=M(\mu(A)-\gamma(A,Y))$$

Since we can make M arbitrarily large or small, we conclude that

$$\sup_{(\phi,\psi)\in B}\int_X\phi(x)\;d\mu+\int_Y\psi(y)\;d\nu-\int_{X\times Y}\phi(x)+\psi(y)\;d\gamma=+\infty$$

This implies that for $\gamma\notin\Pi(\mu,\nu)$, equation (2.7) is $+\infty$. If $\gamma\in\Pi(\mu,\nu)$, then we return to

$$\inf_{\gamma\in\Pi(\mu,\nu)}\int_{X\times Y}c\;d\gamma$$

With this, we proved that the argument that minimizes equation (2.7) must be inside $\{\gamma\in\Pi(\mu,\nu)\}$, which is the original Kantorovich Problem. □

Lemma 6.1.2. *The Dual Problem for the Kantorovich Problem always satisfies the Weak Duality, i.e.* (DP) ≤ (KP).

Proof. Since $\phi\oplus\psi\leq c$,

$$\int_X\phi\;d\mu+\int_Y\psi\;d\nu=\int_{X\times Y}\phi\oplus\psi\;d\gamma\leq\int_{X\times Y}c\;d\gamma$$

□

Lemma 6.1.3. *Let* $c:X\times Y\to\overline{\mathbb{R}}$ *be uniformly continuous. Define two functions* $\phi:X\to\mathbb{R}$ *and* $\psi:Y\to\mathbb{R}$ *Therefore,* ϕ^c *and* ψ^c *have the same modulus of continuity* [1] *as* c.

[1] Check Theorem 6.2.5 for the definition of modulus of continuity

Proof. By Theorem 6.2.5, there exists a modulus of continuity ω, such that

$$|c(x,y) - c(x',y')| \leq \omega(d(x,x') + d(y,y'))$$

Observe that for $g_x(y) = c(x,y) - \phi(x)$

$$|g_x(y) - g_x(y')| = |c(x,y) - c(x,y')| \leq \omega(d(x,x) + d(y,y')) = \omega(d(y,y'))$$

Hence, g_x has modulus of continuity ω. Now, using the Inf-Sup Inequality 6.3.1

$$\begin{aligned} |\inf_x g_x(y) - \inf_x g_x(y')| = |\phi^c(y) - \phi^c(y')| \leq \sup_x |g_x(y) - g_x(y')| = \\ = \sup_x |c(x,y) - c(x,y')| \leq \omega(d(y,y')) \end{aligned}$$

Using the same argument for ψ^c, we showed that both $c-$transforms have the same modulus of continuity.

□

Lemma 2.5.1. *Let X be a metric space, and $c : X \times X \to \mathbb{R}$, where c is a distance metric. Therefore, a function $f : X \to \mathbb{R}$ is c-concave if and only if it is Lipschitz continuous with a constant less than 1 with respect to the distance c. We call $Lip_1^{(c)}$ this set of Lipschitz functions with constant less than 1. Moreover, $f^c = -f$.*

Proof.

$\implies$) Let $f : X \to \mathbb{R}$ be a c-concave function. Hence, $\exists\ g : X \to \overline{\mathbb{R}}$ such that

$$f(x) := \inf_y c(x,y) - g(y)$$

Using the triangle inequality of the cost, we get:

$$\begin{aligned} c(x,y) \leq c(x,z) + c(z,y) &\implies \sup_y c(x,y) - c(y,z) \leq c(x,z) \\ c(y,z) \leq c(y,x) + c(x,z) &\implies \sup_y c(y,z) - c(x,y) \leq c(x,z) \\ &\therefore \\ \sup_y |c(y,z) - c(x,y)| &\leq c(x,z) \end{aligned}$$

Therefore,

$$|f(x) - f(z)| = |\inf_y\{c(x,y) - g(y)\} \; - \; \inf_y\{c(z,y) - g(y)\}| \leq$$
$$\underset{6.3.1}{\leq} \sup_y |c(x,y) - c(z,y)| \leq c(x,z)$$

$\Longleftarrow$) Let $f \in \text{Lip}_1^{(c)}$. Using the Lipschitz inequality,

$$f(x) - f(y) \leq c(x,y) \implies f(x) \leq \inf_y c(x,y) + f(y)$$

But note that $f(x) = c(x,x) + f(x) \geq \inf_y c(x,y) - f(y)$. This implies that $f(x) = \inf_y c(x,y) + f(y)$. Hence, $f(x) = g^c(x)$, where $g(y) = -f(y)$. Which proves that f is c-concave, and $f = (-f)^c$. Finally, note that $-f$ is also Lip_1, therefore, the same argumentation leads to $-f = f^c$.

□

Proving the Duality Theorems

This subsection focuses on proving the Duality Theorems that were stated without proof on Chapter 2. Before proving them, we'll need to some more definitions and results.

Theorem 6.1.1. *(Santambrogio 1.11)*
For X and Y compact metric spaces, and $c : X \times Y \to \overline{\mathbb{R}}$ continuous. Then, the Dual Problem has a solution (ϕ, ϕ^c) for ϕ $c-$concave. Hence

$$\max(\text{DP}) = \max_{\phi \in c-conc.(X)} \int_X \phi \; d\mu + \int_Y \phi^c \; d\nu \tag{6.2}$$

Proof. Let (ϕ_n, ψ_n) be a maximizing sequence of the Dual problem. Note that the c-transforms always improve the Dual Problem, since $\phi_n \oplus \psi_n \leq c$, which implies that

$$\phi_n^c(y) := \inf_x c(x,y) - \phi_n(x) \geq \psi_n(y)$$
$$\psi_n^c(x) := \inf_y c(x,y) - \psi_n(y) \geq \phi_n(x)$$
$$\int_X \phi_n \; d\mu + \int_Y \psi_n \; d\nu \leq \int_X \phi_n \; d\mu + \int_Y \phi_n^c \; d\nu$$

Hence, the sequence (ϕ_n, ϕ_n^c) is also maximizing.

Since $X \times Y$ is compact, the cost c is uniformly continuous. Therefore, by Lemma 6.1.3, the $c-$transforms of ϕ_n and ψ_n are bounded by the same modulus of continuity ω as the cost function c.

Instead of using

$$\psi_n^c(x) = \inf_y c(x,y) - \psi(y)$$

We will use

$$\psi_n^c(x) := \inf_y c(x,y) - \phi_n^c(y) = \phi_n^{cc}(x)$$

This sequence is still maximizing, since

$$\begin{aligned}\phi_n^c(y) = \inf_x c(x,y) - \phi_n(x) \geq \psi_n(y) &\implies \phi_n(x) + \phi_n^c(y) \leq c(x,y) \\ &\implies \psi_n^c(x) = \inf_y c(x,y) - \phi_n^c(y) \geq \phi_n(x)\end{aligned}$$

Therefore, for a maximizing sequence (ϕ_n, ψ_n), we can instead take the maximizing sequence $(\psi_n^c, \phi_n^c) = (\phi_n^{cc}, \phi_n^c)$.

Our goal now is to use the Àrzela-Ascoli Theorem (6.2.6), so we can take a subsequence converging uniformly. To use the theorem, we'll show that our sequence (ψ_n^c, ϕ_n^c) is Equicontinuous (see Definition 6.2.5) and Equibounded (see definition 6.2.6).

First, note that (ψ_n^c, ϕ_n^c) is in fact Equicontinuous, since for any $\epsilon > 0$, we can take $\delta > 0$ such that $d(y,y') < \delta \implies w(d(y,y')) < \epsilon$ and $|\phi_n^c(y) - \phi_n^c(y')| \leq w(d(y,y')) < \epsilon$, for every $n \in \mathbb{N}$.

Next, let's prove that the sequence is Equibounded. Taking the supremum of the inequality, we obtain

$$\sup_{y,y'} |\phi_n^c(y) - \phi_n^c(y')| \leq \sup_{y,y'} w(d(y,y')) = w(\text{diam}(Y))$$

The equality in the equation above is true because the function ω is increasing, and the set Y is compact. Again, the same argument works for ψ_n^c.

Next, realize that we can add and subtract constants from the Dual Problem without modifying the results:

$$\int_X \psi_n^c \, d\mu + \int_Y \phi_n^c \, d\nu = \int_X \psi_n^c + C_n \, d\mu + \int_Y \phi_n^c - C_n \, d\nu$$

Let's take $C_n = \min_y \phi_n^c(y)$. We now change the sequence of functions to $(\psi_n^c + C_n, \phi_n^c - C_n)$, which preserves the maximizing property. Note that $\min_y \phi_n^c - C_n = 0$. Hence,

$$\sup_{y,y'} |\phi_n^c(y) - \phi_n^c(y')| = \max_y \phi_n^c(y) - \min_y \phi_n^c(y) = \max_y \phi_n^c(y) \leq \omega(\text{diam}(Y))$$

Also, for any $x \in X$:

$$\psi_n^c(x) = \inf_y c(x,y) - \phi_n^c(y) \in [\min_y \ c(x,y) - \omega(\text{diam}(Y)), \max_y \ c(x,y) \ ]$$

With this, we showed that the sequence is Equibounded. Therefore, since we are on a compact set and the sequence (ψ_n^c, ϕ_n^c) is both Equicontinuous and Equibounded, we can apply the Àrzela-Ascoli Theorem 6.2.6. Thus, we can obtain a subsequence $(\psi_{n_k}^c, \phi_{n_k}^c)$ that converges uniformly to (ψ, ϕ). As a consequence of this uniform convergence

$$\int_X \psi_{n_k}^c \ d\mu + \int_Y \phi_{n_k}^c \ d\nu \to \int_X \phi \ d\mu + \int_Y \psi \ d\nu$$

With this, we proved that there exists a pair of functions (ϕ, ψ) that are the limits of a maximizing sequence and that satisfy the constraint (i.e. $\phi(x) + \psi(y) \leq c(x,y)$), hence, the Dual Problems has a solution. Also, since $\phi^c \geq \psi$, then (ϕ, ϕ^c) is also an optimal solution for the Dual, and this maximization problem can be restricted to searching in c-concave functions, i.e.:

$$\max(\text{DP}) = \max_{\phi \in c-conc.(X)} \int_X \phi \ d\mu + \int_Y \phi^c \ d\nu$$

□

Definition 6.1.1. (Cyclic Monotonicity) For $c : X \times Y \to \overline{\mathbb{R}}$, a set $\Gamma \subset X \times Y$ is called c-cyclical monotone (c-CM) if $\forall n \in \mathbb{N}$ and $(x_i, y_i) \in \Gamma$ for $i \in \{1, ..., n\}$

$$\sum_{i=1}^n c(x_i, y_i) \leq \sum_{i=1}^n c(x_i, y_{\sigma(i)}) \tag{6.3}$$

Where $\sigma(i)$ is a permutation of the indexes.

Note that this is a stronger property than monotonicity, since for $n = 2$ and $c(x,y) = \langle x, y \rangle$, if Γ is c-CM, then monotonicity is satisfied:

$$\langle x_1, y_1 \rangle + \langle x_2, y_2 \rangle \leq \langle x_1, y_2, \rangle + \langle x_2, y_1 \rangle \tag{6.4}$$

Definition 6.1.2. For X a separable metric space, we define the support of a a measure μ as

$$\text{spt } \mu := \bigcap \{A \ : \ A \text{ is closed and } \mu(X \setminus A) = 0\} \tag{6.5}$$

We can now give an overview of the proof of first Strong Duality Theorem. The proof consists of showing that for an optimal plan γ, its support $\text{spt}(\gamma)$ is c-CM and that for a c-CM set there exists a c-concave function $\phi(x)$ such that $\phi(x) + \phi^c(y) = c(x,y)$ for $(x,y) \in \text{spt}(\gamma)$. Hence, this would prove that

$$\int_{X\times Y} c(x,y)\ d\gamma = \int_X \phi(x)\ d\mu + \int_Y \phi^c(y) d\nu \tag{6.6}$$

Theorem 6.1.2. *(Santambrogio 1.37) Suppose $\Gamma \neq \varnothing$ and is c-CM with $c : X \times Y \to \mathbb{R}$. Then, there exists a c-concave function $\phi : X \to \mathbb{R} \cup \{-\infty\}$ (different than the constant value $-\infty$) such that*

$$\Gamma \subset \{(x,y) \ : \ \phi(x) + \phi^c(y) = c(x,y)\} \tag{6.7}$$

In other words, $\forall (x,y) \in \Gamma,\ c(x,y) = \phi(x) + \phi^c(y)$.

Proof. Fix a point $(x_0, y_0) \in \Gamma$. For $x \in X$, let

$$\begin{aligned}\phi(x) := \inf\{&c(x,y_n) - c(x_n,y_n) + c(x_n,y_{n-1}) - c(x_{n-1},y_{n-1}) + ...+ \\ &+ c(x_1,y_0) - c(x_0,y_0) \ : \ n \in \mathbb{N}, (x_i,y_i) \in \Gamma\ \forall i = 1,...,n\}\end{aligned}$$

$$\begin{aligned}\psi(y) := -\inf\{&-c(x_n,y) + c(x_n,y_{n-1}) - c(x_{n-1},y_{n-1}) + ...+ \\ &c(x_1,y_0) - c(x_0,y_0)) \ : \ n \in \mathbb{N}, (x_i,y_i) \in \Gamma\ \forall i = 1,...,n, y_n = y\}\end{aligned}$$

Note that if $y \notin (\pi_Y)(\Gamma)$, then there is no $(x_n, y) = (x_n, y_n) \in \Gamma$. Therefore,

$$\psi(y) = -\inf\{\varnothing\} = -\infty$$

This implies that $\psi(y) > -\infty \iff y \in (\pi_Y)(\Gamma)$. Note that:

$$\begin{aligned}\psi^c(x) &= \inf_y c(x,y) - \psi(y) = \inf_{y\in(\pi_Y)(\Gamma)} c(x,y) - \psi(y) \\ &= \inf_{y\in(\pi_Y)(\Gamma)} c(x,y) + \inf\{-c(x_n,y) + ... + +c(x_1,y_0) - c(x_0,y_0)) : \\ &\qquad\qquad n \in \mathbb{N}, (x_i,y_i) \in \Gamma\ \forall i = 1,...,n, y_n = y\} \\ &= \phi(x)\end{aligned}$$

Hence, $\phi(x)$ is c-concave, and $\phi(x)$ is not constantly equal to $-\infty$, since for $x = x_0$, we have

$$c(x_0, y_n) + (\sum_{i=0}^{n-1} c(x_{i+1}, y_i)) - \sum_{i=0}^{n} c(x_i, y_i) \geq 0$$
$$\implies \phi(x_0) = \inf\{c(x_0, y_n) + (\sum_{i=0}^{n-1} c(x_{i+1}, y_i)) - \sum_{i=0}^{n} c(x_i, y_i)\} \geq 0$$

Note that the inequality above is true due to the fact that Γ is c-CM.

Now, the only thing left to prove is that $\phi(x) + \phi^c(y) = c(x, y)$ for every $(x, y) \in \Gamma$. First, note that for $\epsilon > 0$ and $(x, y) \in \Gamma$, then:

$$\phi(x) = \psi^c(x) = \inf_y c(x, y) - \psi(y) = \inf_{y \in (\pi_Y)(\Gamma)} c(x, y) - \psi(y) \implies$$
$$\exists \bar{y} \in (\pi_Y)(\Gamma) \ : \ \phi(x) + \epsilon > c(x, \bar{y}) - \psi(\bar{y})$$

Also, note that from the definition of ψ, we have:

$$-\psi(y) \leq -c(x, y) + c(x, \bar{y}) - c(\bar{x}_n, \bar{y}) + ... - c(\bar{x}_0, \bar{y}_0) \ : \forall i, (\bar{x}_i, \bar{y}_i) \in \Gamma$$

Since this is true for any chain on Γ starting on $\bar{y}$, it's true for the infimum, therefore:

$$-\psi(y) \leq -c(x, y) + c(x, \bar{y}) - \psi(\bar{y}) \leq -c(x, y) + \phi(x) + \epsilon$$

Since the ϵ was arbitrary, we can conclude that $c(x, y) \leq \phi(x, y) + \psi(x)$. But, we also know that

$$\begin{aligned}
\phi^c(y) = \psi^{cc}(y) &= \inf_x c(x, y) - \phi(x) \\
&= \inf_x c(x, y) - \inf_y c(x, y) - \psi(y) \\
&\geq \inf_x c(x, y) - c(x, y) + \psi(y) \\
&= \psi(y)
\end{aligned}$$

Hence, $\phi(x) + \phi^c(y) \geq \phi(x) + \psi(y) \geq c(x, y)$.

Lastly, one would need to show that this ϕ is indeed measurable. The general proof is complicated, but, if we assume that c is uniformly continuous, then, we know that c-transforms are continuous (this was shown in Theorem 6.1.1). Since $\phi = \psi^c$, then, ϕ is continuous, therefore, it is measurable if we consider the Borel σ-algebra. $\square$

Theorem 6.1.3. *(Santambrogio 1.38) If γ is an optimal transport plan for cost c continuous, then spt γ is c-CM.*

Proof. The proof consists in supposing that spt γ is not c-CM. Then, we construct a $\tilde{\gamma} \in \Pi(\mu, \nu)$ such that $\int_{X\times Y} c(x,y)\ d\tilde{\gamma} < \int_{X\times Y} c(x,y)\ d\gamma$, which contradicts the optimality of γ.

Check Santambrogio [72] for the complete proof. □

With these results, we can prove the first Strong Duality theorem.

Theorem 2.5.1. *For X and Y compact metric spaces, and $c : X \times Y \to \overline{\mathbb{R}}$ continuous. Then,* $\max(\mathrm{DP}) = \min(\mathrm{KP})$*, and DP admits a solution (ϕ, ϕ^c).*

Proof. Using Theorem 2.4.1, we obtain that $\exists\gamma \in \Pi(\mu, \nu)$ such that it minimizes the Kantorovich Problem, therefore, by Theorem 6.1.3, sptγ is c-CM.

By Proposition 6.1.1, we know that a solution to the Dual Problem can be found in the set of c-concave functions. Using 6.1.2, we can assert that there is a set of c-concave functions such that $\phi(x)+\phi^c(y) = c(x,y)$ for every $(x,y) \in$ spt γ. Since $X \times Y$ is compact, then c is uniformly compact, which implies that ϕ and ϕ^c are continuous and bounded.

Hence, since we already know that $\max(\mathrm{DP}) \leq \min(\mathrm{KP})$, we conclude that $\max(\mathrm{DP}) = \min(\mathrm{KP})$. □

Theorem 2.5.2. *For X and Y Polish spaces and $c : X \times Y \to \mathbb{R}$ uniformly continuous and bounded. Then, (DP) admits a solution (ϕ, ϕ^c) and* $\max(\mathrm{DP}) = \min(\mathrm{KP})$.

Proof. First, note that since X and Y are Polish and c is continuous, one can use Theorem 2.4.3 and affirm that exists an optimal solution γ to (KP).

By the same arguments used on the proof of Theorem 2.5.1, we stablish that spt γ is c-CM, and that ϕ, ϕ^c are continuous functions such that $\forall(x,y) \in$ spt γ, $\phi(x) + \phi^c(y) = c(x,y)$.

In the Dual Problem, the admissible functions ϕ and ψ must be continuous and bounded. Hence, we just need to prove that the ϕ and ϕ^c are indeed bounded. Note that, since c is bounded, then, $|c| \leq M \in \mathbb{R}$ and

$$\phi^c(y) = \inf_x c(x,y) - \phi(x) \leq \inf_x M - \phi(x) = M - \sup_x \phi(x)$$

Note that in 6.1.2, we showed that ϕ is not constantly $-\infty$. Therefore,

$$-\infty < L < \sup_x \phi(x) \implies \phi^c(y) \leq M - \sup_x \phi(x) \leq M - L$$

Similarly, since $\phi = \psi^c$ and $\phi^c(y) \geq \psi(y)$ (shown in 6.1.3), then:

$$\begin{aligned}\phi(x) = \inf_y c(x,y) - \psi(y) \geq -M - \sup_y \psi(y) &\geq -M - \sup_y \phi^c(y) \\ &\geq -M - M + L\end{aligned}$$

Hence, we obtained an upper bound for ϕ^c and a lower bound for ϕ. Now, we obtain an upper bound for ϕ and a lower bound for ϕ^c using a similar argument and relying on the fact that $\sup \psi(y) > L > -\infty$:

$$\begin{aligned}\phi(x) &= \inf_y c(x,y) - \psi(y) \leq M - \sup_y \psi(y) \leq M - L \\ \phi^c(x) &= \inf_x c(x,y) - \phi(x) \geq -M - \sup_x \phi(x) \geq -M - M - L\end{aligned}$$

Finally, using the same arguments as Theorem 2.5.1, we conclude that $\max(\text{DP}) = \min(\text{KP})$ and that (ϕ, ϕ^c) are a solution for the Dual Problem. □

Theorem 2.5.3. *(Santambrogio 1.40) Let $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$, with $c(x,y) = \frac{1}{2}|x-y|^2$. Suppose that $\int |x|^2 d\mu, \int |y|^2 d\nu < +\infty$[2]. Instead of the original Dual Problem, consider the following formulation:*

$$(\text{DP}') \quad \sup\left\{ \int_{\mathbb{R}^d} \phi \, d\mu + \int_{\mathbb{R}^d} \psi \, d\nu \ : \phi \in L^1(\mu) \, , \psi \in L^1(\nu) \, , \ \phi \oplus \psi \leq c \right\} \tag{2.14}$$

Therefore, (DP') admits a solution (ϕ, ψ) and $\max(\text{DP}') = \min(\text{KP})$.

Proof. First, in the same way as the proof of Theorem 2.5.2, (KP) has an optimal solution γ with spt γ that is c-CM and $\forall (x,y) \in \text{spt } \gamma$ we have $\phi(x) + \psi(y) = c(x,y)$. We also have that $-\psi(y) = -\phi^c(y) = \sup_x -\frac{|x-y|^2}{2} +$

[2]This is Theorem 1.40 in Santambrogio [72], but note that there is a small typo in the book, where it states $\int |x|^2 dx, \int |y|^2 dy < +\infty$ instead of the correct $\int |x|^2 d\mu, \int |y|^2 d\nu < +\infty$.

$\phi(x)$. Note that, for $h(x) := \frac{|x|^2}{2} - \phi(x)$

$$h^*(y) := \sup_x \langle x, y\rangle - h(x) = \sup_x \langle x, y\rangle - \frac{|x|^2}{2} + \phi(x) = \frac{|y|^2}{2} + \sup_x -\frac{|x-y|^2}{2} + \phi(x) = \frac{|y|^2}{2} - \psi(y)$$

Therefore, $h(x)$ is equal to the Legendre-Fenchel transform of $\frac{|y|^2}{2} + \psi(y)$, which implies that h is convex l.s.c. The same argument can be used to show that $\frac{|y|^2}{2} - \psi(y)$ is also convex l.s.c.

Since $\frac{|x^2|}{2} - \phi(x)$ is convex, there exists a supporting hyperplane, hence, it is bounded from below by a linear function, which implies that

$$\frac{|x^2|}{2} - \phi(x) \geq \alpha\langle x, y\rangle + \beta \implies \phi(x) \leq \frac{|x^2|}{2} - \alpha\langle x, y\rangle - \beta$$
$$\implies \int_{\mathbb{R}^d} \phi(x)\, d\mu \leq \int_{\mathbb{R}^d} \frac{|x^2|}{2} - \alpha\langle x, y\rangle - \beta\, d\mu < +\infty$$

The same argument can be made for ψ, which means that $\phi_+ \in L^1(\mu)$ and $\psi_+ \in L^1(\nu)$. Due to the fact that $\phi(x) + \psi(y) = c(x, y)$ in the support of γ, then

$$\int_{\mathbb{R}^d \times \mathbb{R}^d} \phi \oplus \psi\, d\gamma = \int_{\mathbb{R}^d \times \mathbb{R}^d} c\, d\gamma \geq 0$$

Which implies that the negative portions of ϕ and ψ are also integrable, leading us to conclude that $\phi \in L^1(\mu)$ and $\psi \in L^1(\nu)$.

Finally, by the same arguments as the previous theorems, we prove that $\max(\text{DP}') = \min(\text{KP})$.

□

We restate the stronger Duality theorem, but we do not present a proof. We defer the interested reader again to Santambrogio [72].

Theorem 2.5.4. *(Santambrogio 1.42) For X and Y Polish spaces and $c : X \times Y \to \mathbb{R} \cup \{+\infty\}$ l.s.c and bounded from below. Then,* $\sup(\text{DP}) = \min(\text{KP})$*. Note that in this theorem, one cannot guarantee the existence of the (ϕ, ψ) that maximize the Dual Problem.*

6.1.3 Wasserstein Distance

Proposition 2.6.1. *$W_p(\cdot, \cdot)$ is a metric on $\mathcal{P}_p(X)$.*

Proof. Let's prove each of the three properties that categorize a metric.

i) $d(x,y) = 0 \iff x = y$.

If $\mu = \nu$, then $(id, id)_{\#}\mu = \gamma$, hence $\int_{X\times X} d(x,y)^p \, d\gamma = \int_{X\times X} d(x,x)^p \, d\mu = 0$.

If $W_p(\mu,\nu) = 0$, then $\int_{X\times X} d(x,y)^p \, d\gamma = 0$. Therefore, γ is concentrated on $\{x = y\}$, otherwise, there would exist a set $A \times B$ such that $\gamma(A \times B) > 0$ and $x \neq y$. Therefore $\int_X d(x,y)^p d\gamma > 0$.

Since γ is concentrated on $\{x = y\}$, then for any set Borel set $K \subset X$:

$$\gamma(K) = \int_{X\times X} \mathbb{1}_K(x,y) \, d\gamma = \int_{x=y} \mathbb{1}_K(x,y) \, d\gamma = \int_{x=y} \mathbb{1}_K(x) \, d\mu = \int_{x=y} \mathbb{1}_K(y) \, d\nu$$

We can conclude that $\mu(K) = \nu(K)$ for every Borel set K, therefore $\mu = \nu$ almost everywhere.

ii) $d(x,y) = d(y,x)$.

$$W_p(\mu,\nu) = \left(\int_{X\times X} d(x,y)^p d\gamma\right)^{1/p} = \left(\int_{X\times X} d(y,x)^p d\gamma\right)^{1/p} = W_p(\nu,\mu)$$

iii) $d(x,z) \leq d(x,y) + d(y,z)$.

Let $\mu, \nu, \rho \in \mathcal{P}_p(X)$, and $\gamma^+ \in \Pi(\mu,\rho)$, $\gamma^- \in \Pi(\rho,\nu)$ are the optimal transport plans for the respective measures. Using the Gluing Lemma 6.1.4, we know that there exists a measure $\sigma \in \mathcal{P}(X \times X \times X)$, where $(\pi_{X,Y})_{\#}\sigma = \gamma^+$

and $(\pi_{Y,Z})_{\#}\sigma = \gamma^{-}$. Also, let $\gamma := (\pi_{X,Z})_{\#}\sigma$. Hence,

$$\begin{aligned}
W_p(\mu,\nu) &\leq \left(\int_{X\times X} d(x,z)^p \; d\gamma\right)^{1/p} = \left(\int_{X\times X} d(x,z)^p \; d(\pi_{X,Z})_{\#}\sigma\right)^{1/p} \\
&\underset{Thm.6.2.8}{=} \left(\int_{X\times X\times X} d(x,z)^p \; d\sigma\right)^{1/p} \\
&\leq \left(\int_{X\times X\times X} (d(x,y)+d(y,z))^p \; d\sigma\right)^{1/p} \\
&= ||d\circ(\pi_{X,Y})(x,y,z) - d\circ(\pi_{Y,Z})(x,y,z)||_{L^p(\sigma)} \\
&\underset{6.3.2}{\leq} ||d\circ(\pi_{X,Y})(x,y,z)||_{L^p(\sigma)} + ||d\circ(\pi_{Y,Z})(x,y,z)||_{L^p(\sigma)} \\
&= \left(\int_{X\times X\times X} d(x,y)^p \; d\sigma\right)^{1/p} + \left(\int_{X\times X\times X} d(y,z)^p \; d\sigma\right)^{1/p} \\
&= \left(\int_{X\times X} d(x,y)^p \; d\gamma^{+}\right)^{1/p} + \left(\int_{X\times X} d(y,z)^p \; d\gamma^{-}\right)^{1/p} \\
&= W_p(\mu,\rho) + W_p(\rho,\nu)
\end{aligned}$$

Which proves the triangle inequality for the Wasserstein distance.

□

Proposition 2.6.2. *For a bounded Polish space X, $p \in [1,+\infty)$, $\mu,\nu \in \mathcal{P}_p(X)$ and $M \in \mathbb{R}_+$, then*

$$W_1(\mu,\nu) \leq W_p(\mu,\nu) \leq M W_1(\mu,\nu)^{1/p} \tag{2.19}$$

Proof. Let $p \leq q \in [1,+\infty)$ and $\gamma \in \Pi(\mu,\nu)$. Hence, $\phi(x) = x^{q/p}$ is a convex function, so by Jensen's inequality:

$$\begin{aligned}
\phi\left(\int d(x,y)^p d\gamma\right)^{1/q} = \left(\int d(x,y)^p d\gamma\right)^{1/p} &\leq \left(\int \phi(d(x,y)^p) d\gamma\right)^{1/q} \\
&= \left(\int (d(x,y)^q) d\gamma\right)^{1/q}
\end{aligned}$$

This implies that $W_p(\mu,\nu) \leq W_q(\mu,\nu)$, when $p \leq q$. In particular, $W_1(\mu,\nu) \leq W_p(\mu,\nu)$ for $p \geq 1$.

Now, since X is bounded, then $d(x,y) \leq \sup_{x,y\in X} d(x,y) = d(X)$. Hence,

$$d(x,y)^p \leq d(X)^{p-1} d(x,y)$$
$$\therefore$$
$$\left(\int d(x,y)^p d\gamma\right)^{1/p} \leq \left(\int d(x,y) d\gamma\right)^{1/p} d(X)^{\frac{p-1}{p}}$$

Therefore, we conclude that $W_p(\mu,\nu) \leq d(X)^{\frac{p-1}{p}} W_1(\mu,\nu)^{1/p}$

□

Theorem 2.6.1. *Let (X,d) be a Polish compact space with $\mu_n, \mu \in P_p(X)$ and $p \in [1,+\infty)$, then $W_p(\mu_n,\mu) \to 0 \iff \mu_n \rightharpoonup \mu$.*

Proof.

$\Longrightarrow$) Let $W_p(\mu_n,\mu) \to 0$. Since X is compact and c is a continuous function, by Theorem 2.4.1 the Kantorovich Problem has a solution. Also, by Theorem 2.5.1, we obtain that $\max(\text{DP}) = \min(\text{KP})$. First, we prove for $p = 1$. In this case, using the Lipschitz version of DP:

$$W_1(\mu,\nu) = \max\left\{\int_X \phi \, d\mu - \int_X \phi \, d\nu \ : \phi \in Lip_1(X)\right\} \to 0$$

This implies that for any $f \in \text{Lip}_1$, $\int f d\mu_n \to \int f d\mu$. Note that, by linearity, the same is true for any Lipschitz function. Since X is compact, then Lipschitz functions are dense on $C(X)$ (see Theorem 6.2.7), which leads us to conclude that $\mu_n \rightharpoonup \mu$ (by Portmanteau 6.2.1). Now, by Proposition 2.6.2, the same is valid for any $p \geq 1$.

$\Longleftarrow$) Let $\mu_n \rightharpoonup \mu$. Define a subsequence μ_{n_k} such that $\lim_k W_1(\mu_{n_k},\mu) = \limsup_n W_1(\mu_n,\mu)$. By the same arguments already used, we know that for each μ_{n_k} there is a $\phi_{n_k} \in \text{Lip}_1$ such that $W_1(\mu_{n_k},\mu) = \int_X \phi_{n_k} d(\mu_{n_k} - \mu)$.

For an arbitrary $\epsilon > 0$, make $\delta = \epsilon$. Since ϕ_{n_k} is 1-Lipschitz, if $d(x,y) < \delta$, then $|\phi_{n_k}(x) - \phi_{n_k}(y)| \leq d(x,y) < \epsilon$, $\forall k \in \mathbb{N}$. Therefore, the sequence is Equicontinuous.

Also, for $x_0 \in X$, we can make $\phi'_{n_k}(x) := \phi_{n_k}(x) - \phi_{n_k}(x_0)$. Note that these functions are 1-Lipschitz and still satisfy $W_1(\mu_{n_k},\mu) = \int_X \phi'_{n_k} d(\mu_{n_k} - \mu)$. Hence, let's use ϕ'_{n_k} as our new subsequence. In this case,

$$|\phi'_{n_k}(x)| = |\phi_{n_k}(x) - \phi_{n_k}(x_o)| \leq d(x,x_o) \leq d(X) < +\infty$$

This implies that this sequence of ϕ'_{n_k} is Equibounded. With this, we can use Arzelà-Ascoli Theorem (6.2.6) to obtain a sub-subsequence that converges uniformly to a $\phi \in \mathrm{Lip}_1(X)$. Replace and relabel the original subsequence, obtaining:

$$\begin{aligned}
W_1(\mu_{n_k}, \mu) &= \int_X \phi_{n_k} d(\mu_{n_k} - \mu) \\
&= \left| \int_X \phi_{n_k} d\mu_{n_k} + \int_X \phi d\mu_{n_k} - \int_X \phi d\mu_{n_k} + \int_X \phi d\mu - \int_X \phi d\mu - \int_X \phi_{n_k} d\mu \right| \\
&\leq \underbrace{\left| \int_X \phi_{n_k} d\mu_{n_k} - \int_X \phi d\mu_{n_k} \right|}_{\text{Goes to 0,due to } \phi_{n_k} \to_u \phi} + \underbrace{\left| \int_X \phi d\mu - \int_X \phi_{n_k} d\mu \right|}_{\text{Goes to 0,due to } \phi_{n_k} \to_u \phi} + \underbrace{\left| \int_X \phi d\mu_{n_k} - \int_X \phi d\mu \right|}_{\text{Goes to 0,due to } \mu_{n_k} \rightharpoonup \mu}
\end{aligned}$$

Therefore $\limsup_n W_1(\mu_n, \mu) \leq 0 \implies W_1(\mu_n \mu) \to 0$. To conclude the proof for any $p \in [1, +\infty)$, we use Proposition 2.6.2:

$$0 \leq W_p(\mu_n, \mu) \leq C W_1(\mu_n, \mu)^{1/p} \leq 0$$

□

Lemma 6.1.4. *(Gluing Lemma)*

Let (X, d) be a metric space. For $\mu, \nu, \rho \in \mathcal{P}(X)$ and $\gamma^+ \in \Pi(\mu, \rho)$, $\gamma^- \in \Pi(\rho, \nu)$. Then, $\exists\ \sigma \in \mathcal{P}(X \times X \times X)$ such that $(\pi_{X,Y})_\# \sigma = \gamma^+$, and $(\pi_{Y,Z})_\# \sigma = \gamma^-$.

Proof. First, use disintegration (Def. 6.2.4) with respect to $f = \pi_Y$ to obtain γ_y^+ and γ_y^-. We know that such disintegration exists and is essentially unique since X is Polish (see Theorem 6.2.4). Note that disintegrated measures are actually defined on $X \times \{y\} \subset X \times X$, but, by abuse of notation, we'll consider that they are measures on X, and y is only an index.

Therefore, make $\sigma = \gamma_y^+ \otimes \rho \otimes \gamma_y^-$, and let $\phi : X \times X \to \mathbb{R}$ be a measurable function. Hence:

$$\begin{aligned}
\int_{X \times X \times X} \phi(x, y)\ d\sigma &\overset{\text{Fubini}}{=} \int_X \int_X \int_X \phi(x, y)\ d\gamma_y^+(x) \otimes \rho(y) \otimes \gamma_y^-(z) \\
&\overset{\text{Indep.}}{=} \int_X d\gamma_y^-(z) \int_X \int_X \phi(x, y)\ d\gamma_y^+(x) \otimes \rho(y) \\
&\overset{\text{Disint.}}{=} \int_X d\gamma_y^-(z) \int_{X \times X} \phi(x, y)\ d\gamma^+(x, y) \\
&= \int_{X \times X} \phi(x, y)\ d\gamma^+(x, y)
\end{aligned}$$

Since $\phi(x, y)$ is arbitrary, then by Corollary 6.2.2, we can conclude that $(\pi_{X,Y})_{\#}\sigma = \gamma^{+}$. By the same argument, we obtain $(\pi_{Y,Z})_{\#}\sigma = \gamma^{-}$, which concludes our proof.

□

Lemma 6.1.5. *For a space of probability measures, we say that μ_n converges weakly to μ, i.e. $\mu_n \rightharpoonup \mu \iff \forall f \in C_c(X),\ \int f\ d\mu_n \to \int f\ d\mu$, where $C_c(X)$ is the space of continuous functions with compact support. Note that $C_c(X) \subset C_0(X) \subset C_b(X)$.*

Proof.

$\Longrightarrow$) If $\mu_n \rightharpoonup \mu$, then $f \in C_c(X) \subset C_b(X)$, hence $\int f d\mu_n \to \int f d\mu$.

$\Longleftarrow$) Suppose that $\forall f \in C_c(X),\ \int f d\mu_n \to \int f d\mu$. Hence, note that for any constant M, $\int f + M d\mu_n = \int f d\mu_n + M \to \int f d\mu + M$. Take $g \in C_b(X)$ and make $g' = g + M \geq 0$ and $g'\mathbb{1}_{[-k,k]} = f_k \in C_c(X)$. Which implies that $f_k \uparrow g'$. Now,

$$\begin{aligned}\left|\int g d\mu_n - \int g d\mu\right| &= \left|\int g' d\mu_n - \int g' d\mu\right| \\ &\leq \left|\int g' d\mu_n - \int f_k d\mu_n\right| + \left|\int f_k d\mu_n - \int f_k d\mu\right| + \left|\int f_k d\mu - \int g' d\mu\right|\end{aligned}$$

Since $f_k \in C_c(X)$, then for n big enough, $\left|\int f_k d\mu - \int f_k d\mu_n\right| < \epsilon$. Therefore,

$$\left|\int g d\mu_n - \int g d\mu\right| \leq \left|\int g' d\mu_n - \int f_k d\mu_n\right| + \epsilon + \left|\int f_k d\mu - \int g' d\mu\right|$$

Since $f_k \uparrow g'$, then, by the Monotone Convergence Theorem,

$$\begin{gathered}\lim_{k\to+\infty}\left|\int g' d\mu_n - \int f_k d\mu_n\right| = 0 \\ \lim_{k\to+\infty}\left|\int f_k d\mu - \int g' d\mu\right| = 0 \\ \therefore \\ \lim_{k\to+\infty}\left|\int g d\mu_n - \int g d\mu\right| = \left|\int g d\mu_n - \int g d\mu\right| \leq \epsilon\end{gathered}$$

□

Theorem 6.1.4. *For $X \subset \mathbb{R}^n$, $\mu_n, \mu \in \mathcal{P}_p(X)$, $x_0 \in X$ and d is metric on X, then*

$$W_p(\mu_n, \mu) \to 0 \iff \int_X d(x, x_0)^p d\mu_n \to \int_X d(x, x_0)^p d\mu \text{ and } \mu_n \rightharpoonup \mu \quad (6.8)$$

Proof.

$\Longrightarrow$) Let $W_p(\mu_n, \mu) \to 0$. Since X is Polish, and c is a continuous function, by Theorem 2.4.3 the Kantorovich Problem has a solution. Also, by Theorem 2.5.4, we obtain that sup(DP) = min(KP). We know that $W_p(\mu_n, \mu) \geq W_1(\mu_n, \nu) \geq 0$, hence, using the Lipschitz version of the Dual Problem for W_1:

$$\sup \left\{ \int_X \phi \, d\mu_n - \int_X \phi \, d\mu \ : \phi \in Lip_1(X) \right\} \to 0$$

This implies that for any $f \in \text{Lip}_1, \int f d\mu_n \to \int f d\mu$. Note that, by linearity, the same is true for any Lipschitz function, not only Lip_1. Finally, since Lipschitz functions are dense on $C_c(X)$ (see Theorem 6.2.7), we can use Lemma 6.1.5 to conclude that $\mu_n \rightharpoonup \mu$.

To prove the other condition (i.e. $\int_X d(x, x_0)^p d\mu_n \to \int_X d(x, x_0)^p d\mu$), define δ_{x_0} as a measure with mass on x_0. Which means that the optimal transport plan γ_n is in $\Pi(\mu_n, \delta_{x_0})$. This implies that $\gamma_n(x, y) = 0$ for any $y \neq x_0$. Therefore

$$\begin{aligned} W_p(\mu_n, \delta_{x_0})^p = \int_{X \times X} d(x, y)^p d\gamma_n &= \int_{X \times \{x_0\}} d(x, y)^p d\gamma_n \\ &= \int_X d(x, x_0)^p d\mu_n \to W_p(\mu, \delta_{x_0})^p = \int_X d(x, x_0)^p d\mu \end{aligned}$$

Where we used the fact that $W(\mu_n, \delta_{x_0}) \to W(\mu, \delta_{x_0})$, which is true since $W(\mu_n, \delta_{x_0}) - W(\mu, \delta_{x_0}) \leq W(\mu_n, \mu)$.

$\Longleftarrow$) Consider now that $\mu_n \rightharpoonup \mu$ and Define $\pi_R : X \to \overline{\text{B}(R)}$, which is the projection on the closed ball with radius R centered at x_0. Since $W_p(\cdot, \cdot)$ is a metric, we have:

$$W_p(\mu_n, \mu) \leq W_p(\mu_n, (\pi_R)_\# \mu_n) + W_p((\pi_R)_\# \mu_n, (\pi_R)_\# \mu) + W_p((\pi_R)_\# \mu_n, \mu)$$

For sake of clarity in the proof, let's define, without loss of generalization, that $d(x, x_0) = |x|$ and $d(x, y) = |x - y|$. Now, note that $|x - \pi_R(x)| = |x| - |x| \wedge R$ and that the plan $(id, \pi_R)_{\#}\mu$ is a possible solution to the OT Problem of transporting μ to $(\pi_R)_{\#}\mu$. Therefore:

$$\begin{aligned} W_p(\mu, (\pi_R)_{\#}\mu)^p &\leq \int_{X\times X} |x-y|^p d(id, \pi_R)_{\#}\mu = \int_{(id,\pi_R)^{-1}(X\times X)} |x - \pi_R(x)|^p d\mu \\ &= \int_X |x - (x \wedge R)|^p d\mu = \int_{B(R)^c} (|x| - R)^p d\mu \end{aligned}$$

And using the same arguments:

$$W_p(\mu_n, (\pi_R)_{\#}\mu_n)^p \leq \int_{B(R)^c} (|x| - R)^p d\mu_n$$

Now, note that

$$\int_X |x|^p - (|x| \wedge R)^p d\mu = \int_{B(R)} |x|^p - |x|^p d\mu + \int_{B(R)^c} |x|^p - R^p d\mu \leq \int_{B(R)^c} |x|^p d\mu$$

Since $\mu_n, \mu \in \mathcal{P}_p(X)$, we know that $\int_X |x|^p d\mu = C < +\infty$ and $\int_X |x|^p d\mu_n = C < +\infty$ then

$$\int_{B(R)^c} |x|^p d\mu = C - \int_{B(R)} |x|^p d\mu \quad \therefore \quad \lim_{R\to 0} \int_{B(R)^c} |x|^p = 0$$

Using that $(|x| - R)^p \leq |x|^p - (|x| \wedge R)^p$ for every $x \in B(R)^c$, we get

$$W_p(\mu_n, (\pi_R)_{\#}\mu)^p \leq \int_{B(R)^c} (|x| - R)^p d\mu_n \leq \int_{B(R)^c} |x|^p - R^p d\mu_n \leq \int_{B(R)^c} |x|^p$$

Now, note that since $\int |x|^p \mu_n \to \int |x|^p d\mu$ and that $(|x| \wedge R)$ is continuous and bounded,

$$\begin{aligned} \lim_n W_p(\mu_n, (\pi_R)_{\#}\mu_n) &\leq \lim_n \int_{B(R)^c} (|x| - R)^p d\mu_n \\ &\leq \lim_n \int_{B(R)^c} |x|^p - R^p d\mu_n = \int_{B(R)^c} |x|^p - R^p d\mu \leq \int_{B(R)^c} |x|^p d\mu \end{aligned}$$

Hence,

$$\lim_R \lim_n (W_p(\mu_n, (\pi_R)_\# \mu_n) \leq \lim_R \int_{B(R)^c} |x|^p d\mu = 0$$
$$\lim_R (W_p(\mu, (\pi_R)_\# \mu) \leq \lim_R \int_{B(R)^c} |x|^p d\mu = 0$$

Lastly, note that since $\overline{B(R)}$ is compact, then we can use Theorem 2.6.1 to stablish that

$$\lim_n W_p((\pi_R)_\# \mu_n, (\pi_R)_\# \mu) = 0$$

We can then conclude that

$$\begin{aligned} \limsup_n W_p(\mu_n, \mu) \leq & \lim_R \limsup_n (W_p(\mu_n, (\pi_R)_\# \mu_n) \\ & + W_p((\pi_R)_\# \mu_n, (\pi_R)_\# \mu) \\ & + W_p((\pi_R)_\# \mu_n, \mu)) \\ = & 0 \end{aligned}$$

□

The Theorem above was proved for $X \subset \mathbb{R}^d$. The proof more general result for Polish spaces (Thm. 2.6.2) can be found in Villani [90] under Theorem 6.9.

6.2 Auxiliary - Probability and Analysis

This section contains definitions and results in Probability and Analysis that are used throughout the text. These results are listed here mostly without proofs.

Definition 6.2.1. Let $d : X \times X \to \mathbb{R}_+$. We say that d is a metric on the set X if for all $x, y, z \in X$, the following three assertions are true:

i) $d(x, y) = 0 \iff x = y$

ii) $d(x, y) = d(y, x)$

iii) $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

Definition 6.2.2. (Weak convergence) We say that $\mu_n \rightharpoonup \mu$ if and only if $\forall f$ continuous and bounded, we have $\int f \, d\mu_n \to \int f \, d\mu$.

Note that this is equivalent to the notion of convergence in distribution, which is more commonly known in probability.

Theorem 6.2.1. *(Portmanteau) Given $\mu \in \mathcal{P}(X)$, where X is a metric space. Then, the following statements are equivalent:*

i) $\mu_n \rightharpoonup \mu$;

ii) $\forall f$ *bounded and uniformly continuous, we have* $\int f \, d\mu_n \to \int f \, d\mu$;

iii) $\forall F \subset X$ *closed,* $\mu(F) \geq \limsup_n \mu_n(F)$;

iv) $\forall F \subset X$ *open,* $\mu(A) \leq \liminf_n \mu_n(A)$;

v) $\forall B$ *such that* $\mu(\partial B) = 0$, *then* $\mu_n(B) \to \mu(B)$.

Note that every set B with $\mu(\partial B) = 0$is called a continuity set. And ∂B is the boundary set of B, hence $\partial B := \hat{B} \setminus \mathring{B}$.

Theorem 6.2.2. *Let X, Y be metric spaces and $\mu_n \rightharpoonup \mu$. Given a continuous map $h : X \to Y$, then $h_{\#}\mu_n = \mu_n \circ h^{-1} \rightharpoonup h_{\#}\mu$.*

Corollary 6.2.1. *If $\mu_n \rightharpoonup \mu$ with $h : X \to Y$ such that $\mu(D_h) = 0$ where D_h is the set of points of discontinuity. Then, $\mu_n \circ h^{-1} \rightharpoonup \mu \circ h^{-1}$.*

Proposition 6.2.1. *If X is Polish, and d is a lower semi-continuous metric on X. For $p \in [1, +\infty)$ and $x_0 \in X$, $\mu_n \rightharpoonup \mu$ and $\int_X d(x, x_0)^p d\mu_n \to \int_X d(x, x_0)^p d\mu$, if, and only if, $\mu_n \rightharpoonup \mu$ and $\lim_{R\to\infty} \int_{d(x,x_0)\geq R} d(x, x_0) d\mu_n \to 0$ (uniformly integrable).*

Definition 6.2.3. (Tight) A family of probability measures $\mathcal{A}$ is tight if for $\epsilon > 0$, $\exists K \subset X$ compact, such that for any $\mu_\alpha \in \mathcal{A}$, $\mu_\alpha(X \setminus K) < \epsilon$

Theorem 6.2.3. *(Prokhorov) This theorem consists in two separate results.*

i) If the family $\mathcal{G} = \{\mu_\alpha\}_{\alpha\in\Lambda}$ is tight, then $\mathcal{G}$ is sequentially pre-compact, i.e. for any $(\mu_n) \subset \mathcal{G}$, $\exists \mu_{n_k} \rightharpoonup \mu$, where $\mu \in \overline{\mathcal{G}}$;

ii) If X is Polish and $\mathcal{G} = \{\mu_\alpha\}_{\alpha\in\Lambda} \subset \mathcal{P}(X)$ is pre-compact. Then $\mathcal{G}$ is tight. In other words, for X polish, and $\mu_n \in \mathcal{P}(X)$ with $\mu_n \rightharpoonup \mu$, then the sequence (μ_n) is tight.

Definition 6.2.4. (Disintegration)

For a Borel measurable space X with a measure μ. Given a function $f : X \to Y$. We say that the family $(\mu_y)_{y\in Y}$ is a Disintegration of μ according to f if every measure μ_y is concentrated on $f^{-1}(\{y\})$, and for every $\phi \in C(X)$, the map $\phi \mapsto \int_X \phi d\mu_y$ is Borel measurable with

$$\int_X \phi \, d\mu = \int_Y \int_X \phi \, d\mu_y(x) \, d\nu(y), \quad \text{where } \nu = f_{\#}\mu \tag{6.9}$$

Note that the existence and uniqueness of disintegration families depend on the spaces where the probabilities are defined, to which we introduce the next theorem.

Theorem 6.2.4. *(Garling [30] 16.10.1) Suppose that X and Y are Polish spaces, that $\mu \in \mathcal{P}(X)$ and that f is a Borel measurable map from X to Y. Then, the f-disintegration of μ exists, and is essentially unique (i.e. $\mu(f^{-1}(B)) = 0$, with $B := \{y \in f(X) : \mu_y \neq \mu'_y\}$ where μ_y and μ'_y are two disintegrations).*

Theorem 6.2.5. *$f : X \to \mathbb{R}$ is uniformly continuous $\iff$ $\exists\ \omega : \mathbb{R}_+ \to \mathbb{R}_+$, such that ω is increasing and $\lim_{x\to 0} w(x) = 0$ with $|f(x) - f(y)| \leq \omega(d(x, y)),\ \forall x, y \in X$. We call ω the modulus of continuity.*

Definition 6.2.5. (Equicontinuous) For a metric space X, the sequence of functions $f_n : X \to \mathbb{R}$ is equicontinuous if $\forall\epsilon > 0,\ \exists\delta > 0 :\ d(x, y) < \delta \implies d(f_n(x), f_n(y)) < \epsilon$ for every $n \in \mathbb{N}$.

Definition 6.2.6. (Equibounded) We say that a sequence (or family) of functions (f_n) is equibounded, if $\exists M > 0\ :\ |f_n(x)| < M < +\infty\ \forall n \in \mathbb{N}$. In words, there is a value M that bounds all functions in the sequence.

Theorem 6.2.6. *(Arzelà-Ascoli) If X is a compact metric space with f_n equicontinuous and equibounded, then $\exists f_{n_k} \to_{unif.} f$, where f is continuous.*

Theorem 6.2.7. *Let (X, d) be metric space. Thus, if X is compact, then $\mathrm{Lip}(X)$ is dense in $C(X)$.*

Proof. (Proof from user125646 [88]) Let $g : X \to \mathbb{R}$ be a continuous function, then since X is compact, g is uniformly continuous. Therefore, for any $\varepsilon > 0$, one can take a $\delta > 0$ such that $d(x, y) < \delta$ implies $|g(x)-g(y)| < \varepsilon$. Now, let $M = \sup_x |g(x)|$ and define

$$f(x) := \sup_y g(y) - \frac{2Md(x, y)}{\delta}$$

Now, note that f is Lipschitz, since

$$\begin{aligned} f(x_1) - f(x_2) &= \sup_y \left(g(y) - \frac{2Md(x_1, y)}{\delta} \right) - \sup_y \left(g(y) - \frac{2Md(x_2, y)}{\delta} \right) \\ &\leq \sup_y \frac{2M(d(x_1, y) - d(x_2, y))}{\delta}) \end{aligned}$$

By the triangle inequality, $d(x_1, y) - d(x_2, y) \leq d(x_1, x_2)$, then

$$\sup_y \frac{2M(d(x_1, y) - d(x_2, y))}{\delta} \leq \sup_y \frac{2Md(x_1, x_2)}{\delta} = \frac{2Md(x_1, x_2)}{\delta}$$

The same argument is valid by exchanging x_1 and x_2, so f has Lipschitz constant $\frac{2M}{\delta}$. Next, let's prove that $\sup_x |g(x) - f(x)| < \varepsilon$.

A first point to notice is that $f(x) \geq g(x)$, since for $y = x$, we have $f(x) = g(x)$. For $d(x, y) \geq \delta$,

$$f(x) = \sup_y g(y) - \frac{2Md(x, y)}{\delta} \leq \sup_y -2M \leq -M \leq g(x)$$

Hence $f(x) \geq g(x) \geq f(x)$, so we obtain an equality.

For $d(x, y) < \delta$,

$$f(x) - g(x) = \sup_y g(y) - g(x) - \frac{2Md(x, y)}{\delta} \leq \varepsilon - \frac{2Md(x, y)}{\delta} < \varepsilon$$

We conclude that $0 < f(x) - g(x) < \varepsilon$, so $\sup_x |f(x) - g(x)| < \varepsilon$.

□

Theorem 6.2.8. *Let $T : X \to Y$ be a measurable map between $(X, \mathcal{F}, \mu)$ and $(Y, \mathcal{G})$. Then, $T_{\#}\mu$ is a measure on $(Y, \mathcal{G})$ and $\forall f$ measurable and integrable with respect to $T_{\#}\mu$ one has:*

$$\int_Y f dT_{\#}\mu = \int_X f \circ T d\mu \tag{6.10}$$

Proof. Let f_n be a simple positive measurable function. Hence

$$f_n(y) = \sum_{i=0}^{N} a_i \mathbb{1}_{A_i}(y) \;\; \therefore \int_Y f_n \; dT_{\#}\mu = \sum_{i=0}^{N} a_i T_{\#}\mu(A_i) = \sum_{i=0}^{N} a_i \mu(T^{-1}(A_i))$$

$$(f_n \circ T)(x) = \sum_{i=0}^{N} a_i \mathbb{1}_{A_i}(T(x)) = \sum_{i=0}^{N} a_i \mathbb{1}_{T^{-1}(A_i)}(x)$$

$$\therefore$$

$$\int_X f_n \circ T \; d\mu = \sum_{i=0}^{N} a_i \mu(T^{-1}(A_i))$$

Hence, $\int_X f_n \circ T \; d\mu = \int_Y f_n \; dT_{\#}\mu$.

Now, for a positive integrable measurable function f, there exists a sequence of positive simple functions such that $f_n \uparrow f$. Then, by the Monotone Convergence Theorem,

$$\int_Y f \; dT_{\#}\mu = \int_Y \lim_{n\to+\infty} f_n \; dT_{\#}\mu = \lim_{n\to+\infty} \int_Y f_n \; dT_{\#}\mu =$$

$$= \lim_{n\to+\infty} \int_X f_n \circ T \; d\mu = \int_Y f \; dT_{\#}\mu$$

If f is non-positive, just use the same argument by splitting the negative and positive portions of the function.

□

Corollary 6.2.2. *Given $\gamma \in \mathcal{P}(X \times Y)$, μ and ν are the marginals in X and Y, respectively $\iff$ for every f, g integrable measurable non-negative functions, we have*

$$\int_{X\times Y} f + g \; d\gamma = \int_X f d\mu + \int_Y g d\nu$$

Proof. $\implies$) Note that $(f \circ \pi_X)(x, Y) = f(\pi_X(x, Y)) = f(x)$, therefore,

$$\int_{X\times Y} f(x) \; d\gamma = \int_{X\times Y} f \circ \pi_X(x, y) \; d\gamma \underset{Theo.1}{=} \int_X f \; d(\pi_X)_{\#}\gamma = \int_X f \; d\mu$$

$\Longleftarrow$) If for all integrable measurable non-negative functions f, g we have

$$\int_{X\times Y} f+g\ d\gamma = \int_x f d\mu + \int_Y g d\nu$$

Then, for any $A \subset X$ measurable, make $f(x) = \mathbb{1}_A(x)$ and $g(y) = 0$. Hence,

$$\gamma(A\times Y) = \int_{X\times Y} \mathbb{1}_{A\times Y}(x,y)\ d\gamma = \int_{X\times Y} \mathbb{1}_A(x)\ d\gamma = \int_X \mathbb{1}_A(x)\ d\mu = \mu(A)$$

□

Lemma 6.2.1. *Let (X,d) be a metric space and $f_k : X \to \mathbb{R}$ be l.s.c and bounded from below for every $k \in \mathbb{N}$. Then, $f = \sup_k f_k$ is also l.s.c and bounded from below.*

Proof. Since $f_k > L$, then $\sup_k f_k > L$, thus f is bounded from below. Next, since f_k is l.s.c, therefore for $x_n \to x$:

$$f_k(x) \le \liminf_j \inf_{n\ge j} f_k(x_n) \implies \sup_k f_k(x) \le \sup_k \liminf_j \inf_{n\ge j} f_k(x_n)$$

Note that $\inf_{n\ge j} f_k(x_n) \le \sup_k \inf_{n\ge j} f_k(x_n)$, hence

$$\liminf_j \inf_{n\ge j} f_k(x_n) \le \liminf_j \sup_k \inf_{n\ge j} f_k(x_n) \implies \sup_k \liminf_j \inf_{n\ge j} f_k(x_n) \le \liminf_j \sup_k \inf_{n\ge j} f_k(x_n)$$

Also, note that $\inf_{n\ge j} f_k(x_n) \le \inf_{n\ge j} \sup_k f_k(x_n)$, hence

$$\sup_k \inf_{n\ge j} f_k(x_n) \le \inf_{n\ge j} \sup_k f_k(x_n) \implies \limsup_j \sup_k \inf_{n\ge j} f_k(x_n) \le \limsup_j \inf_{n\ge j} \sup_k f_k(x_n)$$

We conclude that $\sup_k f(x) \le \lim_j \inf_{n\ge j} \sup_k f_k(x_n)$. So f is l.s.c.

□

6.3 Auxiliary - Inequalities

Lemma 6.3.1. *(Inf-Sup Inequality)*

$$|\inf_{x\in A} f(x) - \inf_{x\in A} g(x)| \le \sup_{x\in A} |f(x) - g(x)| \tag{6.11}$$

Proof. Let's write $\sup_{x\in A} f(x)$ as $\sup_A f$ for simplicity. Note that $f = f - g + g$, hence,

$$\sup_A f = \sup_A f - g + g \leq \sup_A (f-g) + \sup_A g \implies$$
$$\sup_A f - \sup_A g \leq \sup_A f - g \leq \sup_A |f-g|$$

Using the same argument for g, we obtain that

$$|\sup_A f - \sup_A g| \leq \sup_A |f-g| \tag{6.12}$$

Finally, note that

$$|\sup_A f - \sup_A g| = |\inf_A(-f) - \inf_A(-g)| = |-\inf_A f + \inf_A g| =$$
$$= |\inf_A f - \inf_A g| \leq \sup_A |f-g|$$

□

Lemma 6.3.2. *(Minkowski's Inequality) Let X be a measurable space, for $p \in [1, +\infty)$ and $f, g \in L^p(X)$. Therefore,*

$$||f+g||_{L^p(X)} \leq ||f||_{L^p(X)} + ||g||_{L^p(X)} \tag{6.13}$$

Where $||f||^p_{L^p(X)} = \int_X |f|^p d\mu$.

6.4 Transfer Learning Categories

In the literature, there are several ways of categorizing the field of Transfer Learning. A first way is in terms of the differences in the target and source.

1. Inductive transfer learning - Same domain and different tasks (i.e $\mathcal{D}_s = \mathcal{D}_t$, $\mathcal{T}_t \neq \mathcal{T}_t$).

2. Transductive transfer learning - Different domains and the same task (i.e $\mathcal{D}_s \neq \mathcal{D}_t$, $\mathcal{T}_t = \mathcal{T}_t$).

3. Unsupervised transfer learning - Different domains and different tasks (i.e $\mathcal{D}_s \neq \mathcal{D}_t$, $\mathcal{T}_t \neq \mathcal{T}_t$).

A word of caution. This categorization appears in the literature with different meanings. This first classification was introduced by Pan and Yang [58]. Their categorization was also based on the presence or lack of labeled data. Redko et al. [63] use the same categories, but in their case, they consider that the feature spaces are all equal (i.e $\mathcal{X}_s = \mathcal{X}_t$ and $\mathcal{Y}_s = \mathcal{Y}_t$), with the only difference being in the probability distributions.

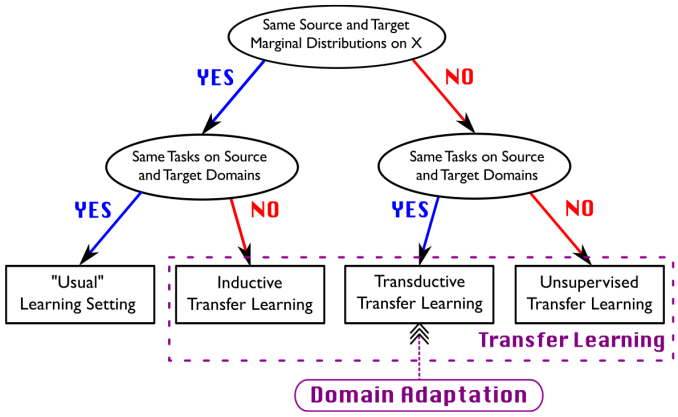

Figure 6.1: Categorizations defined by Redko et al. [63].

Besides this initial categorization, there are many others present in the literature, which classify based on the different modeling techniques, different data perspectives, and others. A good overview is presented by Zhuang et al. [102].

Another source of confusion is the term **Domain Adaptation**. This term is also very present in the literature. Arguably, this term is almost as ubiquitous as the term "Transfer Learning", so much so, that some authors, such as Kouw and Loog [43],use these two terms interchangeably. Other authors, such as Zhuang et al. [102] and Pan and Yang [58] consider Domain Adaptation to be a modeling technique, while Redko et al. [63] considers Domain Adaptation to be equivalent to the transductive transfer learning for the same feature and label space and different probability distributions (i.e $\mathcal{X}_s = \mathcal{X}_t$, but $P(X_s) \neq P(X_t)$), hence, it is a subcategory of Transfer Learning. In this dissertation, we adopt the same definition as Redko et al. [63].

Although Transfer Learning has become quite popular, advances in the theory are still lagging behind. Most of the surveys in the field focus on the different modeling methodologies, which are mostly developed without theoretical guarantees. As shown in the previous section, the field of Transfer Learning is quite large, and there are many scenarios to explore. Each category of Transfer Learning is a world of its own, with different assumptions necessary to develop theory. More than that, each different modeling solution (e.g boosting methods, parameter transfers,...) also entails a different theoretical perspective of tackling the problem.

The main question that the current theory of Transfer Learning tries to address is "when and how can a classifier generalize from a source to a target domain?" [43]. Therefore, researchers have tried to both establish statistical guarantees for transfer learning (e.g [86]), as well as to prove in which scenarios learning is not possible (e.g [91]).

In terms of theory, the most advanced branch of Transfer Learning is Domain Adaptation. By assuming that the only differences are in the probability distributions, one can more easily find error bounds given some assumption of similarity between the source and the target.

Usually, one is interested in minimizing the target risk R_t:

$$\begin{aligned} R_t(h) &= \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}} \mathcal{L}(h(x) \mid y) p_t(x, y) dx \\ &= \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}} \mathcal{L}(h(x) \mid y) p_t(x, y) \frac{p_s(x, y)}{p_s(x, y)} dx \\ &= \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}} \mathcal{L}(h(x) \mid y) p_s(x, y) \frac{p_t(x, y)}{p_s(x, y)} dx \end{aligned}$$

Note that h is the learned hypothesis function, and the risk can be estimated by the empirical sample average:

$$\hat{R}_t(h) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(h(x_i), y_i) \frac{p_t(x_i, y_i)}{p_s(x_i, y_i)}$$

Note that the samples are drawn from the source, and not the target. By decomposing the joint distribution (i.e $p(x, y) = p(x \mid y)p(y) = p(y \mid x)p(x)$), some assumptions can be made regarding the different probability distributions.

- **Prior Shift or Target Shift** - The prior distribution of the labels are different, $p_s(y) \neq p_t(y)$, but the conditional distributions are equivalent, $p_s(x \mid y) = p_t(x \mid y)$;

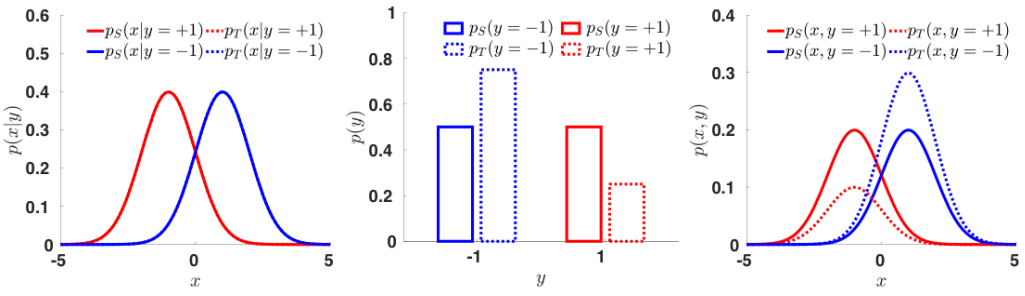

Figure 6.2: Example of Prior Shift [43].

- **Covariate Shift** - The prior for the feature space are different, $p_s(x) \neq p_t(x)$, but the posteriors are equal, $p_s(y \mid x) = p_t(y \mid x)$;

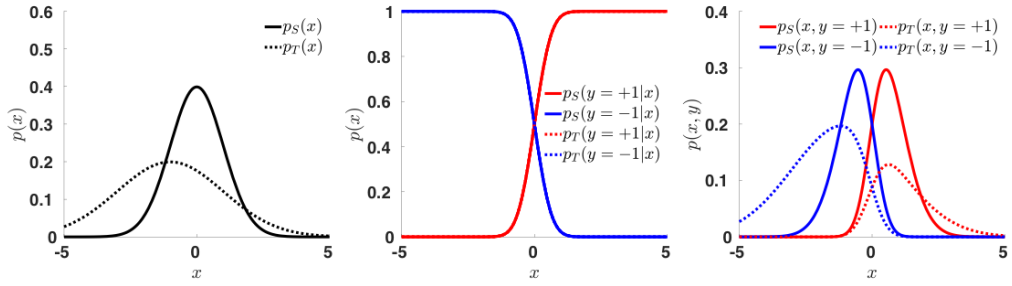

Figure 6.3: Example of Covariate Shift [43].

- **Concept Shift** - The posterior distributions are different, $p_s(y \mid x) \neq p_t(y \mid x)$, but the prior distributions of the feature space are equal, $p_s(x) = p_t(x)$.

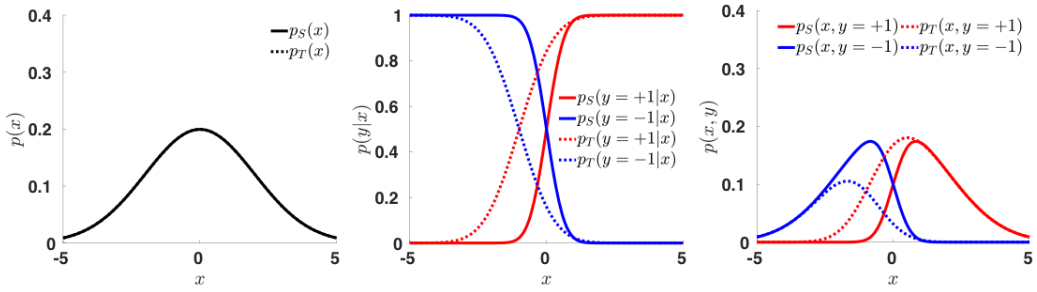

Figure 6.4: Example of Concept Shift [43].

Other than probability shifts, there are many other assumptions necessary for obtaining interesting theoretical results. Redko et al. [63] provides a thorough review of the theory developed for Domain Adaptation, listing many of the different assumptions and results obtained in the literature.