Semidefinite Programming Relaxations of the Simplified Wasserstein Barycenter Problem: An ADMM Approach

by

Jeffrey Cheng

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2023

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of contribution included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of contribution

The research in this thesis was conducted at the University of Waterloo by Jeffrey Cheng under the supervision of Dr. Walaa M. Moursi and Dr. Henry Wolkowicz. Dr. Moursi and Dr. Wolkowicz provided Jeffrey Cheng with the research problem and its context. In addition, they provided Jeffrey Cheng with the main algorithm to solve this research problem. The research group worked together on: improving the algorithm and the corresponding code; and in developing the theory and specific examples in the thesis.

Abstract

The Simplified Wasserstein Barycenter problem, the problem of picking k points each chosen from a distinct set of n points as to minimize the sum of distances to their barycenter, finds applications in various areas of data science. Despite the simple formulation, it is a hard computational problem. The difficulty comes in the lack of efficient algorithms for approximating the solution. In this thesis, I propose a doubly non-negative relaxation to this problem and apply the alternating direction method of multipliers (**ADMM**) with intermediate update of multipliers, to efficiently compute tight lower and upper bounds on its optimal value for certain input data distributions. Our empirics show that generically the gap between upper and lower bounds is zero, though problems with symmetries exhibit positive gaps.

Acknowledgements

At first, I want to thank the examining committee who agree to read my thesis and provide me with valuable feedbacks for improvements.

Next, I want to thank my friends, Andrew Rambidis, Matthew Hough, and Rui Going, who I have met at Waterloo for providing me with both research and emotional supports.

Then, I want to thank Professor Giang Tran from the Applied Math department and Professor Levent Tuncel for introducing me to the world of convex optimization and its broad applications in various engineering disciplines.

Finally, I reserve my special appreciation to Professor Brian E. Forrest from the pure math department for introducing me to the world of mathematical analysis at my freshman year. His teaching eased my difficulty in grasping hard mathematical concepts whose "seeming" uselessness are deeply valued by me in my later studies. His guidance motivated me to enjoy the beauty of mathematical analysis and pursue more relevant applied areas.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Today's society has seen a growing popularity and use of data science, with applications ranging widely from medical science to business development. What lies at the heart of many data science problems are mathematical formulations that use tools from the areas of probability and optimization. What I intend to study in this thesis is one such problem.

Before we dive into the specific problem, it is important to understand the motivation behind it. When people talk about data, they often picture it as a set of points. However, only limited applications can be attempted with this form of representation. In order to broaden the scope of practically, one needs to generalize the representation from single points to probability distributions over candidate points. In fact, data collected in the modern world tends to be represented in this way. However, experienced mathematicians in e.g., functional analysis and other areas, may wonder why we stop here. Why not further generalize optimization problems to infinite dimensional space, e.g., optimization over functionals, or in a partial differential equation setting. A response is that probability distribution functions are more structured, and we intend to exploit certain structures in order to develop efficient algorithms for manipulating data. With this being said, it is fundamental to have tools that manipulate data in the form of probability distributions rather than just discrete single points.

The ability of processing data over probability distributions finds applications in numerous scientific fields [4]. For example, in reinforcement learning and game theory, probability distributions are used to represent mixed strategies and/or policies for maximizing utility. In statistical inference, posterior probability distribution is used to characterize fidelity to observations. In generative modelling, deep fakes[1] for maximizing plausibility are defined using probability distributions. In machine learning, a point cloud[2] is often represented by a probability distribution. In document clustering, word embeddings are characterized by probability distributions. In computer vision and computer graphics, a probability distribution is used to represent an image or an object mesh. In signal processing, sensor measurements are often represented by probability distributions. In neuroscience, a probability distribution over Functional magnetic resonance imaging (fMRI) scans is often used in various applications. In geometric data analysis, transport plans are often represented by probability distributions.

In order to analyze data, it is expected that various operations can be performed on it, such as de-noising, searching, interpolation, summarizing, and clustering. Since the object here is proba-

[1] Deep fakes are fake images or videos in which a person's countenance is replaced by someone else's likeliness.
[2] A point cloud is a discrete set of data points in space, which could represent certain shapes.

bility distribution, a proper distance measure is required to characterize similarity and difference among the data. One choice is to integrate vertical mass difference between two probability distributions. Such measures include L_p norms and Kullback- Leibler divergence. However, this measure captures only the magnitude of mass difference between two probability distributions instead of the locations in which the differences lie, hence lacks geometric meaning. Another choice that remedies this pitfall is to integrate horizontal mass difference. Such measure includes optimal transport distance, i.e: Wasserstein distance.[3] Even though this measure contains information about locations of mass difference, as for geometrically oriented applications, efficient computation of Wasserstein distance between probability distributions is the bottleneck.

After selecting an appropriate distance measure, an additional primitive for manipulating data as aforementioned is to be able to average probability distributions. A canonical way of geometrically averaging data in metric space is to compute their Wasserstein barycenter, the closest probability distribution to all given probability distributions. Efficient computation of Wasserstein barycenter finds applications in numerous fields, such as shape interpolation in computer graphics [28], improving Bayesian learning in statistics [26], unsupervised representation learning in natural language processing [14], sensor improvement [17], and clustering of documents [30, 31] and multilevel clustering of datasets [13, 23].

The problem I study in this thesis is called the Simplified Wasserstein Barycenter problem (3.2.1). Instead of averaging over probability distributions, the problem concerns averaging a group of k points each uniquely selected from each of the given n data sets, such that the aggregate pairwise distances are minimized. Results in [2] show that the standard Wasserstein barycenter problem can be (polynomially) reduced to the Simplified Wasserstein Barycenter problem. Hence, once an efficient algorithm for the Simplified Wasserstein Barycenter problem is discovered, it can be modified to construct an efficient algorithm to solve the standard Wasserstein Barycenter problem.

Despite the simple formulation, the Simplified Wasserstein Barycenter problem has proven to be **NP**-hard [2], i.e., a type of very hard computational problem. The difficulty of this problem comes from the lack of efficient numerical algorithms when the size of the input data grows to large scale. All state of the art algorithms have either inefficient running time or inaccurate approximate solutions(Section 4.3).

In this thesis, we develop an algorithm and study under what circumstances, specifically under what input data distributions, the algorithm becomes efficient in approximating the optimal solution of the Simplified Wasserstein Barycenter problem. Our approach is to introduce and apply a doubly non-negative relaxation to the Simplified Wasserstein Barycenter problem. We split its primal variables using a technique called facial reduction, and then apply the Peachman-Rachford algorithm (**rPRSM**), a variant of the well-known alternating direction methods of multipliers (**ADMM**), to compute tight lower and upper bounds on the optimal value of this **NP**-hard problem. Our empirical experiments illustrate that for input data sampled from the standard normal distribution, we get a zero duality gap between the bounds and thus exactly solve the original hard problem. However, we also show that problems with symmetries result in positive duality gaps.

[3] Wasserstein distance (Kantorovich-Rubinstein metric) is named after Russian-American mathematician Leonid Vaserstein. It intuitively measures the minimum cost for transforming one unit pile of sand into another unit pile of sand.

1.1 Outline

In Chapter 2, we introduce some of the background knowledge in convex analysis, complexity theory, and Euclidean distance matrices that we need in the thesis.

In Chapter 3, we survey some historical motivations for the Wasserstein barycenters problem and its **NP**-hardness result. Then, we introduce the Simplified Wasserstein Barycenter problem, the main problem of interest in this thesis. We apply a doubly non-negative (**DNN**) relaxation to the problem.

In Chapter 4, we survey some historical developments of the alternating method of multipliers (**ADMM**). Then, we apply the Peaceman-Rachford (**rPRSM**) algorithm, An **ADMM** with intermediate update of multipliers, to the **DNN** relaxation formulation of the Simplified Wasserstein Barycenter problem. In addition, we investigate certain input data distributions for which the **rPRSM** algorithm runs efficiently in approximating the optimal solutions of the **DNN** relaxation formulation of the Simplified Wasserstein Barycenter problem, and some techniques for speeding up the algorithm. At last, we reference some past algorithmic approaches to the Wasserstein barycenters problem for historical interest and present our own numerical experiments.

In Chapter 5, we conclude this thesis and suggest further research directions for approximating the optimal solutions of the Simplified Wasserstein Barycenter problem using the **ADMM** algorithm.

Chapter 2

Preliminaries

This chapter is devoted to the background required for the latter chapters. This includes details in convex analysis, complexity of problems, and Euclidean distance matrices. Well versed readers in these areas can skip this chapter for convenience. We have included both an Index (page **??**) and an Appendix (page 53) for the readers' convenience.

2.1 Notation

In this thesis, we use vectors and matrices to formulate our problem of interest. Given vector $x \in \mathbb{R}^n$, we use x_i to denote its i^{th} coordinate entry. Given matrix $X \in \mathbb{R}^{m\times n}$, we use X_{ij} to denote its entry at i^{th} row and j^{th} column. In addition, we use X_S to denote the principal submatrix of X formed by deleting rows and columns with indices not in S, for any $S \subseteq [n]$. In addition, we use the following operations on vectors and matrices. The inner product is the map $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ such that for any pair of vectors $v, g \in \mathbb{R}^n$,

$$\langle v, g \rangle := \sum_{i=1}^{n} v_i g_i.$$

The Hadamard product is the map $\circ : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ such that for any pair of vectors $v, g \in \mathbb{R}^n$,

$$v \circ g := \begin{bmatrix} v_1 g_1 \\ ... \\ v_n g_n \end{bmatrix}.$$

For certain applications, we want to be able to measure the length of a vector. For example, given any vector $v \in \mathbb{R}^n$, its 2-norm is $\|v\|_2 := \sqrt{\sum_{i=1}^{n} |v_i|^2}$.

As for operations on matrices, the inner product is the map $\langle \cdot, \cdot \rangle : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}$ such that for any pair of matrices $M, N \in \mathbb{R}^{m\times n}$,

$$\langle M, N \rangle := \text{trace}(M^T N).$$

The Hadamard product is the map $\circ : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ such that for any pair of matrices

$M, N \in \mathbb{R}^{m\times n}$,

$$M \circ N := \begin{bmatrix} M_{11}N_{11} & ... & M_{1n}N_{1n} \\ ... & ... & ... \\ M_{n1}N_{n1} & ... & M_{nn}N_{nn} \end{bmatrix}.$$

The tensor product is the map $\otimes : \mathbb{R}^{m\times n} \times \mathbb{R}^{p\times q} \to \mathbb{R}^{mp\times nq}$ such that for any pair of matrices $M \in \mathbb{R}^{m\times n}, N \in \mathbb{R}^{p\times q}$,

$$M \otimes N := \begin{bmatrix} M_{11}N & ... & M_{1n}N \\ ... & ... & ... \\ M_{m1}N & ... & M_{mn}N \end{bmatrix} \in \mathbb{R}^{mp\times nq}.$$

In order to measure a matrix, there are two perspectives. One perspective is to treat a matrix as a generalized vector and measure its norm by the 2-norm of the generalized vector. We use the Frobenius norm $\|\cdot\|_F := \sqrt{\langle \cdot, \cdot \rangle}$ to denote such a measure. Another perspective exploits the fact that any matrix is a linear transformation on vectors, and its measure should be the maximum scaling effect of the linear operation. We use the matrix 2-norm $\|\cdot\|_2 := \sigma_{max}(\cdot)$ to denote such a measure.

A generalization of a matrix in high dimensions is called a tensor. We use $\otimes_{i=1}^{m} \mathbb{R}^{n_i}$ to denote a tensor which is a tensor product of real vector spaces. One example is a probability tensor $P \in (\mathbb{R}_+^n)^{\otimes k}$ where each of its entries signals a probability. We use $m_i(P)$ to denote its i^{th} marginal, i.e: $[m_i(P)]_j := \sum_{j_1,...,j_{i-1},j_{i+1},...,j_k} P_{j_1,...,j_k}$. In addition, for probability distributions $\{\mu_1, ..., \mu_k\}$, we use the polytope $\mathcal{M}(\mu_1, ..., \mu_k) := \{P \in (\mathbb{R}_+^n)^{\otimes k} : m_i(P) = \mu_i, \forall i \in [k]\}$ to denote the set of probability tensors with each of its marginals matching the respective probability distribution. We call such polytope a transportation polytope. This object is our primary concern when we introduce the Multimarginal Optimal Transport problem in Chapter 4.

In addition, we are also interested in vectors and matrices with certain structures that aid in formulating our problem of interest. For instances, we use $\mathbb{S}^n$ to denote the space of symmetric matrices of dimension $n \times n$ equipped with the trace inner product, and $\mathbb{S}_+^n \subset \mathbb{S}^n$ to denote the cone of positive semidefinite matrices of dimension $n \times n$. A comprehensive list of mathematical objects is presented in Appendix A.

We now begin to present the background information required for the latter chapters.

Throughout the thesis, we use $\mathbb{E}$ to denote a general Euclidean space and $\mathbb{E}^n$ to infer the dimension of the Euclidean space.

Definition 2.1.1 (*Minkowski sum*)**.** *Let $C_1, C_2 \subseteq \mathbb{E}, \alpha_1, \alpha_2 \in \mathbb{R}$. Then*

$$\alpha_1 C_1 + \alpha_2 C_2 := \{\alpha_1 c_1 + \alpha_2 c_2 : c_1 \in C_1, c_2 \in C_2\}.$$

Definition 2.1.2 (*Orthogonal complement of a set*)**.** *Given $S \subseteq \mathbb{E}$, its orthogonal complement*

$$S^\perp := \{x \in \mathbb{E} : \langle s, x \rangle = 0, \forall s \in S\}.$$

2.2 Background of linear algebra

In this section, we review some basic definitions and theorems from linear algebra. A comprehensive list of linear operators used in this thesis is presented in Appendix B.

Definition 2.2.1 (*Null space of linear map,* $\text{null}(\cdot)$)**.** *Given a linear map* $\mathcal{A} : \mathbb{E}^n \to \mathbb{E}^m$*, its null space is*

$$\text{null}(\mathcal{A}) := \{x \in \mathbb{E}^n : \mathcal{A}(x) = 0\}.$$

Definition 2.2.2 (*Range of linear map,* $\text{range}(\cdot)$)**.** *Given a linear map* $\mathcal{A} : \mathbb{E}^n \to \mathbb{E}^m$*, its range space is*

$$\text{range}(\mathcal{A}) := \{y \in \mathbb{E}^m : \mathcal{A}(x) = y \text{ for some } x \in \mathbb{E}^n\}.$$

Definition 2.2.3 (*Adjoint of linear map*)**.** *Given a linear map* $\mathcal{A} : \mathbb{E}^n \to \mathbb{E}^m$*, its adjoint is the unique linear map* $\mathcal{A}^* : \mathbb{E}^m \to \mathbb{E}^n$ *such that*

$$\langle \mathcal{A}(x), y \rangle = \langle x, \mathcal{A}^*(y) \rangle, \forall x \in \mathbb{E}^n, \forall y \in \mathbb{E}^m.$$

Here is a fact that brings the concepts of null space, range, adjoint, and orthogonal complement all together.

Fact 2.2.4. *(Proposition 7.7, [8]) Given linear map* $\mathcal{A} : \mathbb{E}^n \to \mathbb{E}^m$*,* $\text{null}(\mathcal{A}) = \text{range}(\mathcal{A}^*)^\perp$*.*

The next fact characterizes any symmetric matrix as a linear transformation consisting of scaling and rotation.

Fact 2.2.5. *(Spectral Decomposition Theorem, Proposition 7.29, [8]): For every* $X \in \mathbb{S}^n$*, let* $\lambda(X) = \begin{bmatrix} \lambda_1 \\ ... \\ \lambda_n \end{bmatrix}$ *where* $\{\lambda_1, ..., \lambda_n\}$ *is the set of eigenvalues. Then, there is an orthogonal matrix* $Q = \begin{bmatrix} q_1 & ... & q_n \end{bmatrix} \in \mathbb{R}^{n \times n}$ *composed of orthonormal eigenvectors for* X *such that*

$$X = Q \,\text{Diag}(\lambda(X)) Q^T = \sum_{i=1}^{n} \lambda_i q_i q_i^T.$$

2.3 Positive (semi)definite matrices

For our problem of interest and more broad real world applications, the *convex cone* of positive (semi)definite matrices is an important set to understand. Hence, we review some of its properties.

Definition 2.3.1 (*Positive (semi)definite (**P.S.D.**) matrix*)**.** *A matrix* $X \in \mathbb{S}^n$ *is positive semidefinite, denoted by* $X \succeq 0$*, if*

$$(\forall z \in \mathbb{R}^n) \quad z^T X z \geq 0,$$

*and is positive definite (**P.D.**), denoted by* $X \succ 0$*, if*

$$(\forall 0 \neq z \in \mathbb{R}^n) \quad z^T X z > 0.$$

The following two facts are characterization of positive semidefinite matrices and positive definite matrices.

Fact 2.3.2. *(Characterization of **P.S.D.** matrix, Proposition 1.10, [27]) Let* $X \in \mathbb{S}^n$*. Then the following are equivalent.*

1. *X is **P.S.D.***
2. *For every nonsingular $L \in \mathbb{R}^{n\times n}$, $LXL^T \succeq 0$.*
3. $(\forall i \in [n]) \quad \lambda_i(X) \geq 0$.
4. $(\forall S \subseteq [n]) \quad \det(X_S) \geq 0$.
5. $(\forall Y \in \mathbb{S}^n_+) \quad \langle X, Y\rangle \geq 0$.

Fact 2.3.3. *(Characterization of **P.D.** matrix, Proposition 1.11, [27]) Let $X \in \mathbb{S}^n$. Then the following are equivalent.*

1. *X is **P.D.***
2. *For every nonsingular $L \in \mathbb{R}^{n\times n}$, $LXL^T \succ 0$.*
3. $(\forall i \in [n]) \quad \lambda_i(X) > 0$.
4. $(\forall S \subseteq [n]) \quad \det(X_S) > 0$.
5. $(\forall Y \in \mathbb{S}^n_+\backslash\{0\}) \quad \langle X, Y\rangle > 0$.

Here is a way to test whether a symmetric matrix with positive definite leading block is positive (semi)definite.

Fact 2.3.4. *(Schur's complement lemma, Proposition 1.22, [27]) Let $X \in \mathbb{S}^m, U \in \mathbb{R}^{m\times n}, T \in \mathbb{S}^n_{++}$. Then,*

$$\begin{bmatrix} T & U^T \\ U & X \end{bmatrix} \succeq 0 \iff X - UT^{-1}U^T \succeq 0.$$

In addition,

$$\begin{bmatrix} T & U^T \\ U & X \end{bmatrix} \succ 0 \iff X - UT^{-1}U^T \succ 0.$$

Relationship between faces of $\mathbb{S}^n_+$ and subspaces of $\mathbb{R}^n$

There is a nice relationship between faces of $\mathbb{S}^n_+$ and subspaces of $\mathbb{R}^n$. To understand this relationship, we need some terminology.

Definition 2.3.5 (*Interior*)**.** *Given a set $C \subseteq \mathbb{E}$,*

$$\text{int}(C) := \{x \in C : \exists \epsilon > 0, B_\epsilon(x) \subseteq C\}.$$

Sometimes, we are only interested in the "interior" of a set within its dimension.

Definition 2.3.6 (*Relative interior*)**.** *Given a set $C \subseteq \mathbb{E}$,*

$$\text{relint}(C) := \{x \in C : \exists \epsilon > 0, B_\epsilon(x) \cap \text{aff}(C) \subseteq C\}.$$

Definition 2.3.7 (*Face of a convex set*)**.** *Given convex set $C \subseteq \mathbb{E}$, $\mathcal{F} \subseteq C$ is defined to be a face of C if*

1. $\mathcal{F}$ *is convex.*

2. $(\forall x \in \mathcal{F})\ (\forall y, z \in C)$ *such that* $x \in \text{lineseg}(y, z)$, *we have* $y, z \in \mathcal{F}$.

Definition 2.3.8. *A face* $\mathcal{F}$ *of a convex set* $C \subseteq \mathbb{E}$ *is proper if* $\emptyset \neq \mathcal{F} \neq C$.

Fact 2.3.9 (*Equivalence between faces of* $\mathbb{S}_+^n$ *and subspaces of* $\mathbb{R}^n$, Example 2.2.3, [16])**.** *For any face* $\mathcal{F}$ *of* $\mathbb{S}_+^n$, *there exists a linear subspace* $L \subseteq \mathbb{R}^n$ *such that*

$$\mathcal{F} = \{X \in \mathbb{S}_+^n : \text{range}(X) \subseteq L\}$$

and vice versa. In addition, for a face $\mathcal{F}$ *of* $\mathbb{S}_+^n$ *that corresponds to a linear subspace* L, *we have*

$$\text{relint}(\mathcal{F}) = \{X \in \mathbb{S}_+^n : \text{range}(X) = L\},$$

and for any $V \in \mathbb{R}^{n \times m}$ *such that* $\text{range}(V) = L$,

$$\mathcal{F} = V \mathbb{S}_+^m V^T.$$

2.4 Background of convex analysis and convex optimization

In this section, we review some basic definitions and results of convex analysis and convex optimization. See e.g., the classical book [25].

Definition 2.4.1 (*Affine subspace*)**.** *A subspace* $A \subseteq \mathbb{E}$ *is defined to be affine if*

$$(\forall x, y \in A)(\forall \alpha \in \mathbb{R}) \quad \alpha x + (1 - \alpha) y \in A.$$

Sometimes, we are interested in sets with the property that given any pair of points in the set, the line segment passing through them is also in the set.

Definition 2.4.2 (*Convex set*)**.** *A set* $C \subseteq \mathbb{E}$ *is defined to be convex if*

$$(\forall x, y \in C)(\forall \alpha \in [0, 1]) \quad \alpha x + (1 - \alpha) y \in C.$$

Then, we need a notion to describe the smallest affine set containing a subset of a Euclidean space.

Definition 2.4.3 (*Affine hull*)**.** *Given a set* $C \subseteq \mathbb{E}$,

$$\text{aff}(C) := \{\sum_{i=1}^{k} \alpha_i x_i : x_i \in C, \alpha_i \in \mathbb{R}, k > 0, \sum_{i=1}^{k} \alpha_i = 1\}$$

is the affine combinations of elements of C.

Analogously, the smallest convex set containing a subset of a Euclidean space is just its affine hull restricted onto non-negative coefficients.

Definition 2.4.4 (*Convex hull*)**.** *Given a set $C \subseteq \mathbb{E}$,*

$$\operatorname{conv}(C) := \left\{ \sum_{i=1}^{k} \alpha_i x_i : x_i \in C, \alpha_i \in \mathbb{R}, k > 0, \sum_{i=1}^{k} \alpha_i = 1, \alpha_i \geq 0 \right\}$$

is the convex combinations of elements of C.

2.4.1 Normal cone

The normal cone is an important tool to characterize optimality conditions of optimization problems. To understand it, we first define the concept of a cone.

Definition 2.4.5 (*Cone*)**.** *A set $C \subseteq \mathbb{E}$ is defined to be a cone if*

$$(\forall c \in C)(\forall \alpha \in \mathbb{R}_+) \quad \alpha c \in C.$$

A cone is a generalization of the ray of non-negative numbers from one dimension to arbitrary dimensions. One useful cone is normal cone.

Definition 2.4.6 (*Normal cone, $\mathcal{N}_C(\cdot)$*)**.** *Given non-empty and convex set $C \subseteq \mathbb{E}$, the normal cone to C at point $x \in \mathbb{E}$ is*

$$\mathcal{N}_C(x) := \begin{cases} \{d \in \mathbb{E} : \langle c - x, d \rangle \leq 0, \forall c \in C\}, x \in C; \\ \emptyset, x \notin C. \end{cases}$$

Example 2.4.7. *For any linear subspace C of $\mathbb{E}$, $\mathcal{N}_C(x) = \begin{cases} C^\perp, x \in C; \\ \{0\}, x \notin C. \end{cases}$*

Proof.

$$\begin{aligned} (\forall x \in C) \quad \mathcal{N}_C(x) &= \{d \in \mathbb{E} : \langle c - x, d \rangle \leq 0, \forall c \in C\} \\ &= \{d \in \mathbb{E} : \langle v, d \rangle \leq 0, \forall v \in C\} \\ &= \{d \in \mathbb{E} : \langle v, d \rangle = 0, \forall v \in C\} \\ &= C^\perp. \end{aligned}$$

□

Example 2.4.8 (*The fundamental theorem of linear algebra*)**.** *Given linear map $\mathcal{A} : \mathbb{E}^n \to \mathbb{E}^m$, $\mathcal{N}_{\text{null}(\mathcal{A})}(x) = \text{range}(\mathcal{A}^*), \forall x \in \mathbb{E}^n$.*

Proof. Since $\text{null}(\mathcal{A})$ is a subspace of $\mathbb{E}^n$, $(\forall x \in \text{null}(\mathcal{A})) \quad \mathcal{N}_{\text{null}(\mathcal{A})}(x) = \text{null}(\mathcal{A})^\perp = \text{range}(\mathcal{A}^*)$. □

2.4.2 Proper function

Definition 2.4.9 (*Proper function*)**.** *A function $f : \mathbb{E} \to [-\infty, \infty]$ is defined to be proper if it never attains the negative infinity value and its domain is non-empty, where*

$$\text{dom}(f) := \{x \in \mathbb{E} : f(x) \text{ is finite}\}.$$

Example 2.4.10. *Any continuous function is proper.*

Definition 2.4.11 (*Indicator function*)**.** *Given set $C \subseteq \mathbb{E}$, the indicator function with respect to C is defined to be*

$$i_C : \mathbb{E} \to \{0, \infty\} : x \mapsto \begin{cases} 0, x \in C; \\ \infty, x \notin C. \end{cases}$$

Example 2.4.12. *The indicator function with respect to any non-empty set is proper.*

2.4.3 Lower semicontinuous function

Definition 2.4.13 (*Lower semicontinuous function(l.s.c.)*)**.** *A function $f : \mathbb{E} \to [-\infty, \infty]$ is defined to be l.s.c. at point $x \in \mathbb{E}$ if*

$$(\forall (x_n)_n \to x) \quad f(x) \leq \liminf_{n \to \infty} f(x_n).$$

f is defined to be l.s.c. if it is l.s.c. at all points in $\mathbb{E}$.

Definition 2.4.14 (*Lower level set of a function*)**.** *The lower level set of a function $f : \mathbb{E} \to [-\infty, \infty]$ at height $\alpha \in \mathbb{R}$ is defined to be*

$$lev_\alpha(f) := \{x \in \mathbb{E} : f(x) \leq \alpha\}.$$

Fact 2.4.15 (*Characterization of l.s.c. function*, Thm 3.17, [20])**.** *Given a function $f : \mathbb{E} \to [-\infty, \infty]$, the following are equivalent:*

1. *f is l.s.c.*
2. *$epi(f)$ is closed.*
3. *$(\forall \alpha \in \mathbb{R}) \quad lev_\alpha(f)$ is closed.*

2.4.4 Convex function

Definition 2.4.16 (*Epigraph of a function*)**.** *The epigraph of a function $f : \mathbb{E} \to [-\infty, \infty]$ is defined to be*

$$epi(f) := \{(x, \alpha) : f(x) \leq \alpha\}.$$

Definition 2.4.17 (*Convex function*)**.** *A function $f : \mathbb{E} \to [-\infty, \infty]$ is defined to be convex if $epi(f)$ is convex.*

Fact 2.4.18 (*Characterization of convex function(Jansen's inequality)*, Thm 3.6, [20])**.** *A function $f : \mathbb{E} \to [-\infty, \infty]$ is convex if and only if*

$$(\forall x, y \in \mathrm{dom}(f))(\forall \lambda \in (0, 1)) \quad f[\lambda x + (1 - \lambda)y] \leq \lambda f(x) + (1 - \lambda) f(y).$$

The mere convexity of the objective function of an optimization problem does not guarantee convergence of many algorithms applied on it. Hence, we require a stronger notion than convexity.

Definition 2.4.19 (*Strongly convex function*)**.** *A proper function $f : \mathbb{E} \to (-\infty, \infty]$ is defined to be β-strongly convex if*

$$(\forall x, y \in \mathbb{E})(\forall \lambda \in (0,1)) \quad f[\lambda x + (1-\lambda)y] \leq \lambda f(x) + (1-\lambda) f(y) - \frac{1}{2}\beta\lambda(1-\lambda)\|x-y\|^2 .$$

The parameter β is usually related to the convergence rate of algorithms applied on optimization problems with β-strongly convex objective functions.

Fact 2.4.20 (*First characterization of strongly convex functions*, Fact 24.4, [20])**.**

$$\textit{A proper function is } \beta\textit{-strongly convex} \iff f - \frac{\beta}{2}\|\cdot\|^2 \textit{ is convex.}$$

Local and global minimizers of convex functions

As the objective of most optimization problems is to search for a global minimizer of some function, we need a notion to define local and global minimizers of a function.

Definition 2.4.21 (*Local minimizers of a function*)**.** *Given a proper function $f : \mathbb{E} \to (-\infty, \infty]$, t is a local minimizer of f if*

$$\exists \delta > 0 \textit{ such that } (\forall x \in B_\delta(t)) \quad f(t) \leq f(x).$$

A nice property of proper convex function is that each local minimizer coincides with the global minimizer.

Fact 2.4.22 (Proposition 5.9, [20])**.** *Every local minimizer of a proper and convex function is a global minimizer.*

Fact 2.4.23 (Thm 24.8, [20])**.** *Given β-strongly convex and l.s.c. function $f : \mathbb{E} \to (-\infty, \infty]$, it has a unique minimizer x^* such that*

$$f(x) - f(x^*) \geq \frac{\beta}{2}\|x - x^*\|^2 , \forall x \in \mathrm{dom}(f).$$

Proper, l.s.c., and convex functions

Proper, lower semicontinuous, and convex functions admit many nice properties. For example, in the latter chapters, we will see that the **ADMM** algorithm converges only on optimization problems with proper, l.s.c., and convex objective functions. Hence, we want to characterize functions with these three properties.

Fact 2.4.24 (*Characterization of non-emptiness, closeness, and convexity of a set by its indicator function*, Example 3.19, [20])**.** *Given a set $C \subseteq \mathbb{E}$,*

1. *$C \neq \emptyset \iff i_C$ is proper.*
2. *C is closed $\iff i_C$ is l.s.c.*
3. *C is convex $\iff i_C$ is convex.*

Here is one proper, l.s.c., and convex function.

Example 2.4.25. *Given non-empty $C \subseteq \mathbb{E}$, the support function of C:*

$$\sigma_C : \mathbb{E} \to [-\infty, \infty] : u \mapsto \sup_{c \in C} \langle c, u \rangle$$

is proper, l.s.c., and convex.

The support function can be used to characterize equivalent non-empty closed convex sets.

Fact 2.4.26 (Lemma 8.15, [20])**.** *Given non-empty closed convex sets C and D,*

$$C = D \iff \sigma_C = \sigma_D.$$

Fact 2.4.27 (*Second characterization of strongly convex functions*, Fact 24.4, [20])**.** *Given a proper, l.s.c., and convex function $f : \mathbb{E} \to (-\infty, \infty]$, the following are equivalent:*

1. *f is β-strongly convex.*
2. $(\forall x \in \operatorname{dom}(\partial f))(\forall y \in \operatorname{dom}(f))(\forall u \in \partial f(x)) \quad f(y) \geq f(x) + \langle u, y - x \rangle + \frac{\beta}{2} \|y - x\|^2$.
3. $(\forall x, y \in \operatorname{dom}(\partial f))(\forall u \in \partial f(x))(\forall v \in \partial f(y)) \quad \langle x - y, u - v \rangle \geq \beta \|y - x\|^2$.

2.4.5 Fenchel conjugacy and duality

Definition 2.4.28 (*Fenchel conjugate of a function*)**.** *Given function $f : \mathbb{E} \to [-\infty, \infty]$, its Fenchel-Legendre convex conjugate is*

$$f^* : \mathbb{E} \to (-\infty, \infty] : u \mapsto \sup_{x \in \mathbb{E}} [\langle x, u \rangle - f(x)].$$

Example 2.4.29. *Given any non-empty, closed, and convex set $C \subseteq \mathbb{E}$, $i_C^* = \sigma_C$.*

Proof.

$$i_C^*(u) = \sup_{x \in \mathbb{E}} [\langle x, u \rangle - i_C(x)] = \sup_{x \in C} \langle x, u \rangle = \sigma_C(u).$$

□

2.4.6 Differentiability and subgradient calculus

Definition 2.4.30 (*Subgradient of a function at a point*)**.** *Given function $f : \mathbb{E} \to (-\infty, \infty]$ and point $x \in \mathbb{E}$,*

$$u \in \mathbb{E} \text{ is a subgradient of } f \text{ at } x \iff f(y) \geq f(x) + \langle y - x, u \rangle, \forall y \in \mathbb{E}.$$

Definition 2.4.31 (*Subdifferential of a function at a point*)**.** *Given function $f : \mathbb{E} \to (-\infty, \infty]$ and a point $x \in \mathbb{E}$, the subdifferential of f at x is*

$$\partial f(x) := \{u \in \mathbb{E} : u \text{ is a subgradient of } f \text{ at } x\}.$$

Definition 2.4.32 (*Subdifferentiability*)**.** *A function $f : \mathbb{E} \to (-\infty, \infty]$ is defined to be subdifferentiable over a set C if* $(\forall x \in C) \quad \partial f(x) \neq \emptyset$.

The subdifferential of a proper functions admits many nice properties.

Fact 2.4.33 (Proposition 13.1, [20])**.** *The subdifferential of a proper function is positive homogenous, i.e: Given proper function $f : \mathbb{E} \to (-\infty, \infty]$,*

$$(\forall \alpha > 0) \quad \partial(\alpha f) = \alpha(\partial f).$$

Fact 2.4.34 (Proposition 13.2, Thm 13.4, [20])**.** *The subdifferential of a proper function is quasi-linear, i.e: Given proper convex functions f_1, f_2,*

$$\partial f_1(x) + \partial f_2(x) \subseteq \partial(f_1 + f_2)(x), \forall x \in \operatorname{dom}(f_1) \cap \operatorname{dom}(f_2).$$

and linear, i.e:

$$\partial f_1(x) + \partial f_2(x) = \partial(f_1 + f_2)(x), \forall x \in \operatorname{dom}(f_1) \cap \operatorname{dom}(f_2),$$

if

$$\operatorname{int}[\operatorname{dom}(f_1)] \cap \operatorname{int}[\operatorname{dom}(f_2)] \neq \emptyset \text{ or } \operatorname{relint}[\operatorname{dom}(f_1)] \cap \operatorname{relint}[\operatorname{dom}(f_2)] \neq \emptyset.$$

In addition, if f_1 and f_2 are also l.s.c., then the equality constraint holds if

$$\operatorname{int}[\operatorname{dom}(f_1)] \cap \operatorname{dom}(f_2) \neq \emptyset \text{ or } \operatorname{relint}[\operatorname{dom}(f_1)] \cap \operatorname{relint}[\operatorname{dom}(f_2)] \neq \emptyset.$$

Fact 2.4.35 (Proposition 9.7, Proposition 9.9, [20])**.** *The subdifferential of a proper function at any point is both closed and convex. In addition, any function f is l.s.c. on* $\operatorname{dom}(\partial f)$.

Here is a characterization of the subdifferential of proper convex functions.

Fact 2.4.36 (Proposition 10.12, [20])**.** *Given proper convex function $f : \mathbb{E} \to (-\infty, \infty]$ and point $x \in \mathbb{E}$,*

$$u \in \partial f(x) \iff (u, -1) \in \mathcal{N}_{epi(f)}[x, f(x)].$$

Furthermore, subdifferential can also be used to guarantee convexity of proper functions.

Fact 2.4.37 (Proposition 9.10, [20])**.** *Given proper function $f : \mathbb{E} \to (-\infty, \infty]$, if* $\operatorname{dom}(f)$ *is convex and f is subdifferentiable over* $\operatorname{dom}(f)$*, then f is convex.*

Here is a nice example relating normal cone, subdifferential, and indicator function.

Example 2.4.38. *Given non-empty, closed, and convex $C \subseteq \mathbb{E}, (\forall x \in \mathbb{E}) \quad \mathcal{N}_C(x) = \partial i_C(x)$.*

Proof. If $x \notin C, \mathcal{N}_C(x) = \emptyset = \partial i_C(x)$; otherwise $d \in \mathcal{N}_C(x) \iff (\forall c \in C)\langle c - x, d\rangle \leq 0 \iff \langle c - x, d\rangle + i_C(x) \leq i_C(c) \iff d \in \partial i_C(x)$. □

Directional derivative

When a function is defined on a high dimensional Euclidean space, it is useful to understand the rate of change of the function along some given direction.

Definition 2.4.39 (*Directional derivative of a function at a point*)**.** *The directional derivative of a function $f : \mathbb{E} \to (-\infty, \infty]$ at point x is defined to be*

$$f'(x, \cdot) : \mathbb{E} \to \mathbb{R} : d \mapsto \lim_{\alpha \to 0} \frac{f(x + \alpha d) - f(x)}{\alpha}.$$

Here is a nice characterization of the directional derivative of proper convex functions on the interior of their domains using the support function.

Fact 2.4.40 (Thm 11.7, [20])**.** *Given proper and convex function $f : \mathbb{E} \to (-\infty, \infty]$, point $x \in \text{int}[\text{dom}(f)]$, and direction $d \in \mathbb{E}$,*

$$f'(x, d) = \sigma_{\partial f(x)}(d).$$

The subgradient calculus

Definition 2.4.41 (*Differentiability of a proper function*)**.** *A proper function $f : \mathbb{E} \to (-\infty, \infty]$ is defined to be differentiable at point $x \in \text{int}[\text{dom}(f)]$ if there exists a unique subgradient of f at x: $\nabla f(x)$ such that*

$$\lim_{h \to 0} \frac{f(x + h) - f(x) - \langle \nabla f(x), h \rangle}{\|h\|} = 0.$$

Here is a characterization for a proper convex function to be differentiable at a point.

Fact 2.4.42 (Fact 11.9, [20])**.** *Given proper convex function $f : \mathbb{E} \to (-\infty, \infty]$ and point $x \in \text{int}[\text{dom}(f)]$,*

$$f \text{ is differentiable} \iff f \text{ has a unique subgradient at } x.$$

Unsurprisingly, there is a relationship between directional derivative and gradient of a differentiable function.

Fact 2.4.43 (Proposition 11.8, [20])**.** *Given differentiable, proper, and convex function $f : \mathbb{E} \to (-\infty, \infty]$, point $x \in \text{int}[\text{dom}(f)]$, and direction $d \in \mathbb{E}$,*

$$f'(x, d) = \langle \nabla f(x), d \rangle.$$

Differentiability of convex function

Differentiable convex functions admit many nice properties. One of them is the monotonicity of their gradients.

Definition 2.4.44. *A function $g : \mathbb{E} \to (-\infty, \infty]$ is defined to be monotone if*

$$\langle x - y, g(x) - g(y) \rangle \geq 0, \forall x, y \in \mathbb{E}.$$

Fact 2.4.45 (Fact 6.2(iii), [20])**.** *Given proper, differentiable, and convex function $f : \mathbb{E} \to (-\infty, \infty]$ whose domain is open and convex, ∇f is monotone.*

Example 2.4.46. *Given non-empty, closed and convex $C \subseteq \mathbb{E}$, $\frac{1}{2} d_C^2(\cdot)$ is differentiable and convex. Hence, $\nabla[\frac{1}{2} d_C^2(\cdot)] = id - \mathcal{P}_C$ is monotone.*

A condition that guarantees the convergence of many algorithms on optimization problems with differentiable objective functions is smoothness.

Definition 2.4.47 (**L-smooth function**). *A function $f : \mathbb{E} \to (-\infty, \infty]$ is defined to be L-smooth over $D \subseteq \mathbb{E}$ if f is differentiable over D and ∇f is L-Lipschitz continuous over D, i.e:*

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|, \forall x, y \in D.$$

Fact 2.4.48 (**The descent lemma**, Lemma 23.6 [20]). *Given L-smooth function $f : \mathbb{E} \to (-\infty, \infty]$ over $D \subseteq \mathbb{E}$,*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2.$$

Smoothness of convex differentiable functions can be algebraically characterized as follows:

Fact 2.4.49 (Fact 23.8, [20]). *Given convex and differentiable function $f : \mathbb{E} \to \mathbb{R}$, the following are equivalent:*

1. *f is L-smooth.*
2. *$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2, \forall x, y \in \mathbb{E}$.*
3. *$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2L} \|\nabla f(y) - \nabla f(x)\|^2, \forall x, y \in \mathbb{E}$.*
4. *$\langle x - y, \nabla f(x) - \nabla f(y) \rangle \geq \frac{1}{L} \|\nabla f(x) - \nabla f(y)\|^2, \forall x, y \in \mathbb{E}$.*

2.4.7 Constrained convex optimization

In this section, we review some basic definitions and results of constrained convex optimization.

A constrained convex optimization problem has the following form:

$$\begin{array}{ll} \min_{x \in \mathbb{E}} & f(x) \\ s.t. & x \in C \end{array}$$

where the objective function $f : \mathbb{E} \to (-\infty, \infty]$ is proper, l.s.c., and convex and the constraint set C is non-empty, closed, and convex.

Two typical examples are linear programs and semidefinite programs.

Definition 2.4.50 (*Linear program,* ***LP***). *Given linear map $A : \mathbb{R}^n \to \mathbb{R}^m, b \in \mathbb{R}^m, c \in \mathbb{R}^n$, an **LP** has the following form:*

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & \langle c, x \rangle \\ s.t. & Ax \leq b \\ & x \geq 0 \end{array}$$

Definition 2.4.51 (*Semidefinite program,* ***SDP***). *Given linear map $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m, b \in \mathbb{R}^m, C \in \mathbb{R}^{n \times n}$, a **SDP** has the following form:*

$$\begin{array}{ll} \min_{X \in \mathbb{S}^n} & \langle C, X \rangle \\ s.t. & \mathcal{A}(X) = b \\ & X \succeq 0 \end{array}$$

Here are two examples that are not constrained convex optimization problems.

Definition 2.4.52 (*Integer quadratic program,* ***IQP***). *Given linear map $A : \mathbb{R}^n \to \mathbb{R}^m, b \in \mathbb{R}^m, D \in \mathbb{S}^n$, an* ***IQP*** *has the following form:*

$$\begin{array}{cc} \min_{x\in\mathbb{R}^n} & x^T D x \\ s.t. & Ax = b \\ & x \in \{0,1\}^n \end{array}$$

Definition 2.4.53 (*Binary-constrained quadratic program,* ***BCQP***). *A* ***BCQP*** *is an instance of an* ***IQP*** *where A is binary and $b = e$.*

The following lemma characterizes the global minimizer of a constrained convex optimization problem.

Fact 2.4.54 (*Rockafellar-Pshenichnyi lemma*, Thm 2.1, [24]). *Given convex set $C \subseteq \mathbb{E}$ and convex function $f : C \subseteq \mathbb{E} \to \mathbb{R}$,*

$$\textit{a point } x \in C \textit{ is a minimizer of } f \textit{ if and only if } 0 \in \partial f(x) + N_C(x).$$

Sometimes, the constraint set can be formulated as the intersection of finitely many sets which can be described using non-linear inequalities and equalities. We call such convex constrained problems as nonlinear programs.

Definition 2.4.55. *Given convex functions $\{f : \mathbb{R}^n \to \mathbb{R}, g_i : \mathbb{R}^n \to \mathbb{R} : i \in [n]\}$ and affine functions $\{h_i : \mathbb{R}^n \to \mathbb{R} : i \in [m]\}$, a convex optimization problem has the following form:*

$$\begin{array}{cc} \inf & f(x) \\ s.t. & g_i(x) \leq 0, \forall i \in [n] \\ & h_j(x) = 0, \forall j \in [m] \end{array}$$

Karush, Kuhn and Tucker refined the Rockafellar-Pshenichnyi lemma to the following conditions in order to characterize optimal solutions of a convex optimization problem.

Definition 2.4.56 (*Slater point*). *$x \in \mathbb{R}^n$ is a Slater point if*

1. $(\forall i \in [n]) \quad g_i(x) < 0.$
2. $(\forall i \in [m]) \quad h_i(x) = 0.$

Fact 2.4.57 (*Karush-Kuhn-Tucker,* ***KKT*** *conditions*, Thm 16.1, Thm 16.2, [20]). *Given a convex optimization problem that has a Slater point, a primal-dual pair $(x^*, \lambda_1, ..., \lambda_n, \beta_1, .., \beta_m)$ is an optimal solution if and only if the following* ***KKT*** *conditions hold:*

1. *Primal feasibility:* $(\forall i \in [n]) g_i(x^*) \leq 0$ *and* $(\forall i \in [m]) h_i(x^*) = 0.$
2. *Dual feasibility:*
   (a) $0 \in \partial f(x^*) + \sum_{i=1}^n \lambda_i \partial g_i(x^*) + \sum_{i=1}^m \beta_i \partial h_i(x^*).$
   (b) $(\forall i \in [n]) \quad \lambda_i \geq 0.$
3. *Complementary slackness:* $(\forall i \in [n]) \quad \lambda_i g_i(x^*) = 0.$

Optimality conditions for split problems

As we use a splitting algorithm to solve our problem of interest, we are particularly interested in the optimality conditions of the following convex split problem:

$$\begin{array}{ll} \min_{X\in\mathbb{S}^n, Y\in\mathbb{S}^m} & f(X,Y) \\ \text{s.t.} & \mathcal{A}_1(X)+\mathcal{A}_2(Y)=0 \\ & X\in\mathcal{X} \\ & Y\in\mathcal{Y} \end{array} \tag{2.4.1}$$

where the objective function f is differentiable and convex, both $\mathcal{A}_1:\mathbb{S}^n\to\mathbb{R}^m$ and $\mathcal{A}_2:\mathbb{S}^m\to\mathbb{R}^m$ are linear, and both $\mathcal{X}$ and $\mathcal{Y}$ are nonempty, closed and convex sets.

Define the linear manifold

$$C:=\{(X,Y)\in\mathbb{S}^n\times\mathbb{S}^m:\mathcal{A}_1(X)+\mathcal{A}_2(Y)=0\}$$

corresponding to the linear constraint. Then, the feasible set of (2.4.1) is

$$K:=C\cap(\mathcal{X}\times\mathcal{Y}).$$

By Rockafellar-Pshenichnyi lemma,

$$(X^*,Y^*) \text{ is an optimal solution to (2.4.1)} \iff -\nabla f(X^*,Y^*)\in\mathcal{N}_K(X^*,Y^*).$$

This characterization has very limited use as $\mathcal{N}_K(X^*,Y^*)$ is very hard to describe. It's much nicer to be able to characterize the optimality conditions using $\mathcal{A}_1,\mathcal{A}_2,\mathcal{N}_{\mathcal{X}}(\cdot)$, and $\mathcal{N}_{\mathcal{Y}}(\cdot)$ directly as they usually admit more structures to exploit. This motivates the next result.

Theorem 2.4.58. *Under the above setting, assume that there exists $X^*\in\mathcal{X}$ and $Y^*\in\mathcal{Y}$ such that*

$$\mathcal{N}_K(X^*,Y^*)=\mathcal{N}_C(X^*,Y^*)+\mathcal{N}_{\mathcal{X}}(X^*)\times\mathcal{N}_{\mathcal{Y}}(Y^*).$$

Then,

$$(X^*,Y^*) \text{ is an optimal solution to (2.4.1)} \iff \begin{cases} -\nabla_X f(X^*,Y^*)\in\operatorname{range}(\mathcal{A}_1^*)+\mathcal{N}_{\mathcal{X}}(X^*); \\ -\nabla_Y f(X^*,Y^*)\in\operatorname{range}(\mathcal{A}_2^*)+\mathcal{N}_{\mathcal{Y}}(Y^*). \end{cases}$$

Proof. Define $\mathcal{A}(X,Y):=\mathcal{A}_1(X)+\mathcal{A}_2(Y)$. By (2.4.8), $\mathcal{N}_C(X^*,Y^*)=\operatorname{range}(\mathcal{A}^*)$.

We now show that $\operatorname{range}(\mathcal{A}^*)=\operatorname{range}(\mathcal{A}_1^*)\times\operatorname{range}(\mathcal{A}_2^*)$.

$(\forall w\in\mathbb{R}^m)$

$$\begin{aligned} \langle(X,Y),\mathcal{A}^*(w)\rangle=\langle\mathcal{A}(X,Y),w\rangle &= \langle\mathcal{A}_1(X)+\mathcal{A}_2(Y),w\rangle \\ &= \langle\mathcal{A}_1(X),w\rangle+\langle\mathcal{A}_2(Y),w\rangle \\ &= \langle X,\mathcal{A}_1^*(w)\rangle+\langle Y,\mathcal{A}_2^*(w)\rangle \\ &= \langle(X,Y),(\mathcal{A}_1^*(w),\mathcal{A}_2^*(w))\rangle. \end{aligned}$$

Hence, $\mathcal{N}_C(X^*, Y^*) = \text{range}(\mathcal{A}_1^*) \times \text{range}(\mathcal{A}_2^*)$. With the assumption, we conclude

$$\begin{aligned}\mathcal{N}_K(X^*, Y^*) &= \mathcal{N}_C(X^*, Y^*) + [\mathcal{N}_{\mathcal{X}}(X^*) \times \mathcal{N}_{\mathcal{Y}}(Y^*)] \\ &= [\text{range}(\mathcal{A}_1^*) \times \text{range}(\mathcal{A}_2^*)] + [\mathcal{N}_{\mathcal{X}}(X^*) \times \mathcal{N}_{\mathcal{Y}}(Y^*)] \\ &= [\text{range}(\mathcal{A}_1^*) + \mathcal{N}_{\mathcal{X}}(X^*)] \times [\text{range}(\mathcal{A}_2^*) \times \mathcal{N}_{\mathcal{Y}}(Y^*)].\end{aligned}$$

□

Note that the above nice characterization relies on the decomposability of the normal cone. Does this condition hold all the time? Rockafellar gives a sufficient condition.

Fact 2.4.59 (Corollary 23.8.1 [25])**.** *Given convex sets $C_1, .., C_m$ such that the intersection of their relative interiors is non-empty. Then,*

$$\mathcal{N}_{C_1 \cap ... \cap C_m}(x) = \mathcal{N}_{C_1}(x) + ... + \mathcal{N}_{C_m}(x).$$

If in addition $C_1, .., C_k$ are polyhedral, the above conclusion holds if the intersection of $C_1, .., C_k, \text{relint}(C_{k+1}), ..., \text{relint}(C_m)$ is non-empty.

2.4.8 Projection and proximal point mapping

While solving optimization problems, if the current iterate of someone's algorithm is infeasible, he naturally wants to know the closest feasible point to the current iterate point. We use the projection operator to denote such an operation.

Definition 2.4.60 (*Projection onto a set*)**.** *Given set $S \subseteq \mathbb{E}$, the projection operator onto S is*

$$\mathcal{P}_S : \mathbb{E} \to S : x \mapsto \text{argmin}_{s \in S} \|s - x\|.$$

For example, here is an algebraic characterization of projection onto hyperplanes.

Example 2.4.61 (*Projection onto hyperplane*)**.** *Let $H := \{x \in \mathbb{E} : a^T x = b, a \neq 0\}$ be a hyperplane in $\mathbb{E}$. The projection onto H is*

$$\mathcal{P}_H : \mathbb{E} \to H : x \mapsto x + \frac{b - \langle a, x \rangle}{\|a\|^2} a.$$

In order to understand the projection operator, we need to firstly understand the proximal point mapping of a function.

Definition 2.4.62 (*Proximal point mapping of a function*)**.** *The proximal point mapping of a function $f : \mathbb{E} \to (-\infty, \infty]$ is define to be*

$$Prox_f : \mathbb{E} \to \mathcal{P}(\mathbb{E}) : x \mapsto \text{argmin}_{u \in \mathbb{E}} \{f(u) + \frac{1}{2} \|u - x\|^2\}.$$

The connection between projection operator and proximal point mapping is through the indicator function.

Proposition 2.4.63. *Given non-empty, closed, and convex set $C \subseteq \mathbb{E}, Prox_{i_C} = \mathcal{P}_C$.*

Proof. $(\forall x \in \mathbb{E}) Prox_{i_C}(x) = \operatorname{argmin}_{u\in\mathbb{E}}\{i_C(u) + \frac{1}{2}\|u-x\|^2\} = \operatorname{argmin}_{u\in C}\{\|u-x\|^2\} = \mathcal{P}_C(x)$. □

Fact 2.4.64. *The proximal point mapping of a proper, l.s.c., and convex function is single-valued.*

Corollary 2.4.65. *The projection onto any non-empty, closed and convex subset of a Euclidean space is single-valued.*

Proof. $(\forall \emptyset \neq C \subseteq \mathbb{E})$ such that C is both closed and convex, i_C is proper, l.s.c., and convex. □

Fact 2.4.66 (*Characterization of the proximal point mapping of a proper, l.s.c., and convex function*, Thm 25.3, [20])**.** *Given proper, l.s.c., and convex function $f : \mathbb{E} \to (-\infty, \infty]$ and point $x \in \mathbb{E}$,*

$$p = Prox_f(x) \iff x - p \in \partial f(p) \iff f(y) \geq f(p) + \langle y - p, x - p\rangle, \forall y \in \mathbb{E}.$$

Corollary 2.4.67 (*Characterization of projection onto non-empty, closed, and convex set*, Corollary 25.4, [20])**.** *Given non-empty, closed, and convex set $C \subseteq \mathbb{E}$ and $x \in \mathbb{E}$,*

$$p = \mathcal{P}_C(x) \iff p \in C \text{ and } \langle c - p, x - p\rangle \leq 0, \forall c \in C.$$

Fact 2.4.68 (*Characterization of minimizers of a proper, l.s.c., and convex function*, Thm 25.9, [20])**.** *Given proper, l.s.c., and convex function $f : \mathbb{E} \to (-\infty, \infty]$,*

$$x \text{ is a minimizer of } f \iff x = Prox_f(x).$$

Fact 2.4.69 (*Projection translation theorem*, Example 25.15, [20])**.** *Given non-empty, closed, and convex $C \subseteq \mathbb{E}$, points $x, y \in \mathbb{E}$,*

$$\mathcal{P}_{y+C}(x) = y + \mathcal{P}_C(x - y).$$

Fact 2.4.70 (*Characterization of projection onto affine subspace*)**.** *Given affine subspace U of $\mathbb{E}$ and point $x \in \mathbb{E}$,*

$$p = \mathcal{P}_U(x) \iff p \in U \text{ and } \langle y - z, x - p\rangle = 0, \forall y, z \in U.$$

Fact 2.4.71 (*Characterization of projection onto linear subspace*)**.** *Given linear subspace U of $\mathbb{E}$ and point $x \in \mathbb{E}$,*

$$p = \mathcal{P}_U(x) \iff p \in U \text{ and } x - p \in U^\perp.$$

Example 2.4.72 (*Generalized (Moore-Penrose) inverse of linear operator*)**.** *Given linear operator $L : \mathbb{R}^n \to \mathbb{R}^m$, its pseudo-inverse operator is*

$$L^+ : \mathbb{R}^m \to \mathbb{R}^n : y \mapsto \mathcal{P}_{C_y}(0),$$

where $C_y := \{x \in \mathbb{R}^n : L^*Lx = L^*y\}$.

In order to prove that an algorithm does not diverge, it is often necessary to prove that the operator T for each iteration of the algorithm is 1-Lipschitz, i.e:

$$\|Tx - Ty\| \leq \|x - y\|, \forall x, y \in \mathbb{E}.$$

We call such an operator contractive.

If in addition, T satisfies:

$$\|Tx - Ty\|^2 \leq \langle x - y, Tx - Ty \rangle,$$

we can prove that the algorithm converges. We call such an operator firmly contractive.

Fact 2.4.73. *The proximal point mapping of a proper, l.s.c., and convex function is firmly contractive.*

Corollary 2.4.74. *The projection operator onto non-empty, closed, and convex set is firmly contractive.*

Fact 2.4.75. *Given smooth, proper, l.s.c., and convex finite function f with L-Lipschitz continuous gradient ∇f, both $\frac{\nabla f}{L}$ and $id - \frac{\nabla f}{L}$ are firmly contractive.*

2.4.9 Algorithms - Subgradient methods

In this section, we will investigate some historical first-order methods developed for solving optimization problems. All of them use a descent direction of the objective function for each iterate update.

Definition 2.4.76 (**descent direction of a function at a point**). *Given proper function $f : \mathbb{E} \to (-\infty, \infty]$ and point $x \in \text{int}[\text{dom}(f)]$, a vector d is called a descent direction of f at x if the directional derivative $f'(x, d)$ exists and is negative.*

Example 2.4.77. *If $\nabla f(x)$ exists and is non-zero, then $-\nabla f(x)$ is a descent direction of f at x.*

Gradient steepest descent method (GSD)

The gradient steepest descent method is a greedy approach that assumes that the objective function f is differentiable and simply takes the steepest descent direction at each iterate update:

$$x_{n+1} := x_n - t_n \nabla f(x_n), \text{ where } t_n \in \text{argmin}_{t>0} f[x_n - t\nabla f(x_n)].$$

Peressini, Sullivan, and Uhl proved that when f is strictly convex, coercive, i.e: $\lim_{x:\|x\|\to\infty} f(x) = \infty$, and the set of global minimizes of f is non-empty, the convergence of GSD is guaranteed.

Projected subgradient method (PSM)

The projected subgradient method solves the constrained convex optimization problem (2.4.7) with the additional assumptions that

1. ∂f on C is bounded by some constant $L > 0$, i.e: $(\forall c \in C)(\forall d \in \partial f(c)) \quad \|d\| \leq L$.
2. The set of global minimizers S is non-empty.

Each iterate update rule is

$$x_{n+1} := x_n - t_n f'(x_n).$$

where

1. $f'(x_n) \in \partial f(x_n)$.
2. The sequence of step sizes $\{t_n\}_n$ are such that $\frac{\sum_{n=0}^{\infty} t_n^2}{\sum_{n=0}^{\infty} t_n} = 0, e.g : \{\frac{1}{n+1}\}_{n\in\mathbb{N}}$.

For error tolerance ϵ, PSM is guaranteed to converge in $\frac{L^2 d_S^2(x_0)}{\epsilon^2} - 1$ iterations, where $d_S(x_0) := \|x_0 - s\|$ denotes the distance between the initial point to the closest global minimizer s.

Proximal gradient method (PGM)

The proximal gradient method solves optimization problems of the form:

$$\min_{x\in\mathbb{E}} \quad f(x) + g(x)$$

with the following assumptions:

1. The set of global minimizers is non-empty.
2. $f : \mathbb{E} \to (-\infty, \infty]$ is proper, l.s.c., convex, and L-smooth on $\text{int}[\text{dom}(f)]$.
3. $g : \mathbb{E} \to (-\infty, \infty]$ is proper, l.s.c. and convex such that $\text{dom}(g) \subseteq \text{int}[\text{dom}(f)]$.

Optimization problems of this form can be viewed as a generalization of the constrained optimization problem (2.4.7) with $g = i_C$.

The update rule of each iterate is

$$x_{n+1} := Prox_{\frac{1}{L}g}[(id - \frac{1}{L}\nabla f)(x_n)].$$

At iteration n, the gap between the current objective value and the optimal value is upper bounded by $\frac{Ld_S^2(x_0)}{2n} \in O(\frac{1}{n})$ with the asymptotic regularity rate $\|x_{n+1} - x_n\|$ upper bounded by $\frac{\sqrt{2}d_S(x_0)}{\sqrt{n}} \in O(\frac{1}{\sqrt{n}})$.

Fast iterative soft thresholding algorithm (FISTA)

The convergence rate of PGM can be improved by using an auxiliary sequence $\{y_n\}_{n\in\mathbb{N}}$ at each iterate. The pseudocode is as follows:

Algorithm 1 FISTA

Initialization: $x_0 \in \mathbb{E}, y_0 := x_0; t_0 := 1$.
while stopping criterion is not satisfied **do**
  $t_{n+1} := \frac{1+\sqrt{1+4t_n^2}}{2}$.
  $x_{n+1} := Prox_{\frac{1}{L}g}[(id - \frac{1}{L}\nabla f)(y_n)]$.
  $y_{n+1} := x_{n+1} + \frac{t_n - 1}{t_{n+1}}(x_{n+1} - x_n)$.
end while

At iteration n, the gap between the current objective value and the optimal value is upper bounded by $\frac{Ld_S^2(x_0)}{(n+1)^2} \in O(\frac{1}{n^2})$, a quadratic improvement compared to that of PGM.

2.5 Background of complexity theory

Complexity theory is the study of hardness of computational problems. To classify different computational problems into their respective echelons, we need some notions to characterize how hard these problems are.

Naturally, we want to start with easy problems.

Definition 2.5.1 (*Deterministic polynomial time,* ***P***). *Let* Γ *be an alphabet. A language* $L \subseteq \Gamma^*$ *is in the computation complexity class* ***P*** *if there exists a deterministic Turing Machine that decides whether any string is in* L *in polynomial time.*

It describes the class of decision problems solvable by a deterministic Turing machine in polynomial time, which we characterize as easy problems. Certainly, there are harder computational problems.

Definition 2.5.2 (*Non-deterministic polynomial time,* ***NP***). *Let* Γ *be an alphabet. A language* $L \subseteq \Gamma^*$ *is in the computation complexity class* ***NP*** *if there exists a nondeterministic Turing Machine that decides whether any string is in* L *in polynomial time.*

It describes the class of decision problems for which the correctness of a given answer can be verified by a deterministic Turing machine in polynomial time. Note that it is not required for a **NP**-problem to be solvable in polynomial time, but to simply verify the correctness of a given answer.

Definition 2.5.3 (***NP****-hard*). *Let* Γ *be an alphabet. A language* $L \subseteq \Gamma^*$ *is* ***NP****-hard if any language in* ***NP*** *can be reduced to* L *in polynomial time.*

Definition 2.5.4 (***NP****-complete*). *Let* Γ *be an alphabet. A language* $L \subseteq \Gamma^*$ *is* ***NP****-complete if it is both* ***NP****-hard and in* ***NP****.*

A natural approach to efficiently tackle a **NP**-hard problem is to approximate the solution instead. Hence, we need a notion to characterize easy approximation problems.

Definition 2.5.5 (*Bounded-error probabilistic polynomial time,* ***BPP***). *Let* Γ *be an alphabet. A language* $L \subseteq \Gamma^*$ *is in the computation complexity class* ***BPP*** *if there exists a randomized Turing Machine that decides whether any string is in* L *with error probability upper bounded by* $\frac{1}{3}$ *in polynomial time.*

It describes the class of computational decision problems solvable by a probabilistic Turing machine in polynomial time with an error probability upper bounded by $\frac{1}{3}$.

Here is a nice relationship between **NP** problems and **BPP** problems.

Fact 2.5.6 ([7]). *Under standard cryptographic assumptions,* $\boldsymbol{NP} \not\subset \boldsymbol{BPP}$.

2.6 Basic results about Euclidean distance matrix

Euclidean distance matrix plays a significant role in the reformulation of our problem of interest. In this section, we review some definitions and results about Euclidean distance matrices. See e.g., [1]

Definition 2.6.1 (*Euclidean Distance Matrix,* ***EDM***). *A matrix $D = (D_{ij}) \in \mathbb{S}^n$ is defined to be a Euclidean distance matrix (**EDM**) if there exists a matrix of associated points $P := \begin{bmatrix} p_1^T \\ ... \\ p_n^T \end{bmatrix} \in \mathbb{R}^{n \times r}$ such that $D_{ij} = \|p_i - p_j\|^2 \left(= \|p_i\|^2 + \|p_j\|^2 - 2p_i^T p_j\right)$.*

We are particularly interested in the relationship between Euclidean distance matrix and positive semidefinite matrix. To bring these two concepts together, we need a couple of definitions.

Definition 2.6.2 (*Hollow space,* $\mathbb{S}_H^n$). *The hollow space of $\mathbb{S}^n$ is $\mathbb{S}_H^n := \{D \in \mathbb{S}^n : \operatorname{diag}(D) = 0\}$.*

Definition 2.6.3 (*Centred space,* $\mathbb{S}_C^n$). *The centred space of $\mathbb{S}^n$ is $\mathbb{S}_C^n := \{Y \in \mathbb{S}^n : Ye = 0\}$.*

Definition 2.6.4 (*Lindenstrauss operator,* $\mathcal{K}$). *The Lindenstrauss operator between symmetric matrices is $\mathcal{K} : \mathbb{S}^n \to \mathbb{S}^n : G \mapsto \operatorname{diag}(G)e^T + e \operatorname{diag}(G)^T - 2G$.*

Fact 2.6.5. *The Moore-Penrose generalized inverse of Lindenstrauss operator $\mathcal{K}$ is*

$$\mathcal{K}^\dagger : \mathbb{S}^n \to \mathbb{S}^n : D \mapsto -\frac{1}{2} J \operatorname{offDiag}(D) J$$

where

1. $J := proj_{e^\perp} = I - \frac{1}{n} ee^T$;
2. $\operatorname{offDiag} : \mathbb{S}^n \to \mathbb{S}^n : D \mapsto D - \operatorname{Diag}[\operatorname{diag}(D)]$ *is the orthogonal projection onto* $\mathbb{S}_H^n$.

Now, we are ready to state the relationship between Euclidean distance matrix and positive semidefinite matrix.

Fact 2.6.6 ([22, Pg. 5], *Relationship between* ***EDM*** *and* ***P.S.D.****matrix*). *Given an* ***EDM*** *D with associated points P, $D = \mathcal{K}(PP^T)$. Conversely, $\mathcal{K}^\dagger(D) \in \mathbb{S}_+^n \cap \mathbb{S}_C^n$.*

Chapter 3

Wasserstein barycenter

In this chapter, we explore the Wasserstein barycenter problem. At first, we illustrate **NP**-hardness of its computation. Then, we seek to develop an efficient approximation algorithm for the simplified Wasserstein barycenter problem, a simplified version of the dual feasibility problem to an equivalent **NP**-hard problem called Multimarginal Optimal Transport (**MOT**) problem, in high dimensions. There exists a polynomial algorithm that reduces the standard Wasserstein barycenter problem to the simplified Wasserstein barycenter problem. Hence, in order to tackle the standard Wasserstein barycenter problem, it suffices to develop an efficient approximation algorithm for the simplified Wasserstein barycenter problem. Our approach is to invoke a doubly non-negative relaxation to the simplified Wasserstein barycenter problem and apply the Peaceman-Rachford algorithm (**rPRSM**), an **ADMM** with intermediate update of multipliers, to approximate the optimal value of the simplified Wasserstein barycenter problem.

3.1 NP-hardness of the Wasserstein barycenter problem

At first, we define Wasserstein distance. Let (M, d) be a metric space. Define $\Gamma(\mu, \nu)$ to be the set of joint probability distributions on $M \times M$ whose first and second marginals are μ and ν respectively, i.e: $\mu(x) = \int_M \gamma(x, y)dy$ and $\nu(y) = \int_M \gamma(x, y)dx$. For $p \in [1, \infty)$, the Wasserstein p-distance between probability distributions μ and ν on M is

$$\mathcal{W}_p(\mu, \nu) := [\inf_{\gamma \in \Gamma(\mu,\nu)} \mathbb{E}_{(x,y)\sim\gamma} d(x, y)^p]^{\frac{1}{p}}.$$

Next, we define the standard Wasserstein barycenter problem. Let d be the l_2 norm. Given probability distributions $\{\psi_1, ..., \psi_k\}$ over $\mathbb{R}^d$ and non-negative weights $\{\beta_1, ..., \beta_k\}$, the standard Wasserstein barycenter problem is to search for a probability distribution in

$$\text{argmin}_\nu \sum_{i=1}^{k} \beta_i \mathcal{W}_2^2(\psi_i, \nu). \tag{3.1.1}$$

The general Wasserstein barycenter problem extends the l_2 norm to a general l_q norm. Given $(p, q) \subseteq [1, \infty)$, we use $W_{p,q}$ to denote the p-Wasserstein distance on metric space $(\mathbb{R}^d, l_q)$. The

general (p, q)-Wasserstein barycenter problem is to search for a probability distribution in

$$\text{argmin}_{\nu} \sum_{i=1}^{k} \beta_i \mathcal{W}_{p,q}^{p}(\psi_i, \nu). \tag{3.1.2}$$

Observe that when $p = 2 = q$, the standard Wasserstein barycenter problem is recovered. The rational for generalizing the standard Wasserstein barycenter problem is simple. The parameter $p \in [1, \infty)$ controls the effect of outliers on the notion of average. For example, when $p = 1$, the problem computes the geometric medians of given probability distributions, which is very robust to outliers. However, when p approaches infinity, an outlier can affect the average magnitude greatly and deviates the true optimal distribution, since any optimal distribution is constrained to be close to all given distributions simultaneously. As many real world applications vary in significance of outliers, we need the flexibility to control p. The parameter $q \in [1, \infty)$ represents the underlying geometry structure. For instance, if $q = 2$, the geometry is Euclidean space. However, since we want to extend our notion to other geometry structures, such as Riemannian geometry, we need the flexibility to control q.

3.1.1 Wasserstein barycenter of discrete probability distributions

For most computational applications, the given probability distributions represent point clouds over a finite number of points. By this discrete structure, we can confine the support of each probability distribution to a number of points that is upper bounded by some number n.

A natural optimization question arises. Does there exist an efficient algorithm that computes the standard Wasserstein barycenter of discrete distributions? Specifically, does there exist an algorithm that solve (3.1.1) with time complexity polynomial in the number of distributions k, the number of supported points n, the point dimension d, and the bit complexity $\log U$ of each entry in the given distributions and weights?

Theorem 1.1 of [2] shows that the answer is no.

Fact 3.1.1 ([2, Thm 1.1]). *Assume* $\boldsymbol{P \neq NP}$. *There does not exist an algorithm that solves (3.1.1) with uniform weights* $\beta_1 = ... = \beta_k = \frac{1}{k}$ *in* $poly(n, k, d, \log U)$ *time.*

Does this problem become easier to solve when the goal is only to approximate the optimal solution up to some error bound ϵ. Unfortunately, Theorem 1.2 of [2] brings the bad news again.

Fact 3.1.2 ([2, Thm 1.2]). *Assume* $\boldsymbol{NP \not\subset BPP}$. *Let* R *be an upper bound on the squared diameter of the supports of given probability distributions. There does not exist a randomized algorithm that approximates (3.1.1) with uniform weights* $\beta_1 = ... = \beta_k = \frac{1}{k}$ *to an accuracy of* ϵ *with probability lower bounded by* $\frac{2}{3}$ *in* $poly(n, k, d, \log U, \frac{R}{\epsilon})$ *time.*

Unsurprisingly, the **NP**-hardness result extends to the general Wasserstein barycenter problem as well.

Fact 3.1.3 ([2, Thm 1.3]). *Assume* $\boldsymbol{NP \not\subset BPP}$. *Let* $R_{p,q}$ *be an upper bound on the* p^{th} *power of the* l_q *norm diameter of the supports of given probability distributions. There does not exist a randomized algorithm that approximates (3.1.2) with uniform weights* $\beta_1 = ... = \beta_k = \frac{1}{k}$ *to an accuracy of* ϵ *with probability lower bounded by* $\frac{2}{3}$ *in* $poly(n, k, d, \log U, \frac{R_{p,q}}{\epsilon})$ *time.*

Next, we outline the techniques the author employed in proving these **NP**-hardness results. At first, we consider an equivalent **NP**-hard problem called Multimarginal Optimal Transport (**MOT**). The **MOT** problem corresponding to probability distributions $\{\psi_1 = \{x_{1,1}, ..., x_{1,n}\}, ..., \psi_k = \{x_{k,1}, ..., x_{k,n}\}\}$ and cost tensor $C \in (\mathbb{R}^n)^{\otimes k}$ is defined to be

$$\min_{P \in \mathcal{M}(\psi_1, ..., \psi_k)} \langle C, P \rangle \tag{3.1.3}$$

which is a **LP** over n^k variables.

Proposition 2.1 of [2] proves that (3.1.2) is equivalent to (3.1.3) with cost tensor $C \in (\mathbb{R}^n)^{\otimes k}$ for entries $C_j = \min_{y \in \mathbb{R}^d} \sum_{i=1}^k \beta_i \|x_{i,j_i} - y\|_q^p, \forall j \in [n]^k$.

It is worth mentioning that certain **MOT** problems with special structures of their cost tensors can be solved efficiently. For examples, some applications in financial risk management concern cost tensors with low ranks, and some applications in network reliability testing concern cost tensors with certain sparsity patterns. For these types of **MOT** problems, there exists polynomial-time algorithms.

However, general **MOT** problems with no particular structures on their cost tensors remain **NP**-hard. [2] reduces CHEAPEST-HUB$_{p,q}$, a simplified version of the dual feasibility problem to (3.1.3), to (approximately) solving the **MOT** problem. The problem of CHEAPEST-HUB$_{p,q}$ is to compute

$$\min_{j \in [n]^k} F_{p,q}(x_{1,j_1}, ..., x_{k,j_k})$$

where the cheapest p-distances measured in l_q norm with respect to locations $\{z_1, ..., z_k\}$ is defined to be

$$F_{p,q}(z_1, .., z_k) := \min_{y \in \mathbb{R}^d} \sum_{i=1}^k \|z_i - y\|_q^p.$$

CHEAPEST-HUB$_{p,q}$ has an intuitive geometric interpretation: If we are given k sets each consisting of n points, how do we find one point from each set in order to minimize the average distance to their closest hub. The geometry of searching for k points that are close to each other mimics the k-CLIQUE problem which is **NP**-hard [21]. The k-CLIQUE problem is to search for k vertices in a graph such that all pairs of them are close in the sense of being adjacent to each other. Due to the similar structures of these two problems, [2] shows a reduction from k-CLIQUE to CHEAPEST-HUB$_{p,q}$.

Note that for $p = 2 = q$, we are considering simply the Euclidean geometry. We call this problem (CHEAPEST-HUB$_{2,2}$) the simplified Wasserstein barycenter problem, the main problem of interest in this thesis.

3.1.2 Wasserstein barycenter of continuous probability distributions

One may also wonder how to compute Wasserstein barycenter of continuous probability distributions. However, the continuous setting faces more challenges than the discrete setting. The first issue is how to concisely represent a continuous probability distribution. Another issue concerns the efficiency of computing the Wasserstein distance between two continuous probability distributions. One particular continuous probability distribution that eases these two issues is the Gaussian distri-

bution. Many algorithms proposed for computing Wasserstein barycenter of continuous probability distributions are restricted only to the Gaussian setting.

3.2 The simplified Wasserstein barycenter problem

In this section, we apply a doubly non-negative relaxation to the simplified Wasserstein barycenter problem (CHEAPEST-HUB$_{2,2}$).

3.2.1 A reformulation using Euclidean distance matrix

Proposition 3.2.1. *Let $S_1, \ldots, S_k \subset \mathbb{R}^n$ be the set of points initially given. The simplified Wasserstein barycenter problem*

$$p_W^* := \min_{p_1 \in S_1, \ldots, p_k \in S_k} F_{2,2}(p_1, ..., p_k)$$

is equivalent to the problem of finding exactly one point in each set $S_i, i = 1, \ldots, k$, that minimizes the sum of squared distances:

$$(WIQP) \qquad 2kp_W^* = p^* := \min_{p_1 \in S_1, \ldots, p_k \in S_k} \sum_{i,j \in [k]} \|p_i - p_j\|^2. \tag{3.2.1}$$

Proof. Assume $p_i, i \in [k]$ are optimal points with barycenter y. Without loss of generality, we may assume $y = 0$ by translating all the points by y, i.e: $p_j \leftarrow p_j - y, \forall j$. Note that this translation does not affect the objective function. Define matrix P with rows p_i. Then, P is centred, i.e., $P^T e = 0$. Hence, the Gram matrix of P, $G = PP^T$ admits the property $Ge = 0$. Define the Euclidean distance matrix corresponding to P by the Lindenstrauss operator $\mathcal{K}$ such that $D_{ij} = \|p_i - p_j\|^2$,

$$D := \mathcal{K}(G) = \operatorname{diag}(G)e^T + e \operatorname{diag}(G)^T - 2G.$$

The result follows by noting that the sum of squared norms is

$$\begin{aligned} \textstyle\sum_{i,j \in [k]} \|p_i - p_j\|^2 &= e^T D e \\ &= e^T \left(\operatorname{diag}(G)e^T + e \operatorname{diag}(G)^T - 2G \right) e \\ &= 2k \operatorname{trace} G \\ &= 2k \textstyle\sum_{i \in [k]} \|p_i\|_2^2 \\ &= 2kp_W^*. \end{aligned}$$

where

$$e^T e = k, e^T \operatorname{diag}(G) = \operatorname{trace}(G) = \sum_{i \in [k]} \|p_i\|_2^2.$$

We now show an alternative proof. Firstly note that

$$\begin{aligned} p_W^* &= \min_{\substack{y\in\mathbb{R}^d \\ p_i\in S_i}} \sum_{i\in[k]} \|p_i - y\|^2 \\ &= \min_{p_i\in S_i} \min_{y\in\mathbb{R}^d} \sum_{i\in[k]} \|p_i - y\|^2 \\ &= \min_{p_i\in S_i} \sum_i \|p_i - \frac{\sum_j p_j}{k}\|^2 \\ &= \min_{p_i\in S_i} \sum_i \|\frac{1}{k}\sum_{j\neq i} p_i - p_j\|^2 \\ &= \frac{1}{k^2}\min_{p_i\in S_i}[\sum_i \sum_{j\in[k]} \|p_i - p_j\|^2 + \sum_{i,j,l\in[k],i,j,l \text{ are all differenet}} \langle p_i - p_j, p_i - p_l\rangle]. \end{aligned}$$

We now show that $\sum_{i,j,l\in[k],i,j,l \text{ are all differenet}} \langle p_i - p_j, p_i - p_l\rangle = \frac{k-2}{2}\sum_{i,j\in[k]} \|p_i - p_j\|^2$.

Base case: $k = 3$: Note that $2[\langle p_i - p_j, p_i - p_l\rangle + \langle p_j - p_i, p_j - p_l\rangle + \langle p_l - p_i, p_l - p_j\rangle]$
$= [\langle p_i - p_j, p_i - p_l\rangle + \langle p_j - p_i, p_j - p_l\rangle] + [\langle p_i - p_j, p_i - p_l\rangle + \langle p_l - p_j, p_l - p_i\rangle] + [\langle p_j - p_i, p_j - p_l\rangle + \langle p_l - p_j, p_l - p_i\rangle]$
$= \|p_i - p_j\|^2 + \|p_i - p_l\|^2 + \|p_j - p_l\|^2 = \frac{1}{2}\sum_{i,j\in[k]} \|p_i - p_j\|^2$.

Strong inductive hypothesis: Assume the statement holds for 1,...,k-1.

For k: There are in total $\binom{k}{3}$ different (i, j, l) tuples for the sum, each tuple corresponds to $\|p_i - p_j\|^2 + \|p_i - p_l\|^2 + \|p_j - p_l\|^2$. Once (i, j) is fixed, there are only k-2 possible choices left for l, resulting in the factor $\frac{k-2}{2}$.

By rearranging we get

$$k^2 p_W^* = \frac{k}{2}\min_{p_i\in S_i} \sum_{i,j\in[k]} \|p_i - p_j\|^2 \iff 2kp_W^* = \min_{p_i\in S_i} \sum_{i,j\in[k]} \|p_i - p_j\|^2.$$

□

Define

$$x := [v_1^T, ..., v_k^T]^T \in \{0, 1\}^{nk}, \quad A := \text{blkdiag}[e^T, ..., e^T] \in \mathbb{R}^{k\times nk}.$$

Then, the constraints of picking exactly one point from each set are equivalent to

$$Ax = e.$$

Hence, an **BCQP** reformulation using Euclidean distance matrix is as follows:

$$\begin{array}{rlll} & p^* = & \min & x^T D x = \langle D, xx^T\rangle \\ (\textbf{BCQP}) & & \text{s.t.} & Ax = e \\ & & & x = [v_1^T, ..., v_k^T]^T \in \{0, 1\}^{nk} \end{array} \tag{3.2.2}$$

3.2.2 Difficulty of the simplified Wasserstein barycenter problem

We now look at the simplified Wasserstein barycenter problem from another angle that illustrates its **NP**-hardness. In essence, this problem can be formulated as a constrained minimization of a concave function.

Theorem 3.2.2. *Let $G = \mathcal{K}^\dagger(D)$ denote the centred Gram matrix, and let $S \in \mathbb{S}^{nk}_+$ be its positive semidefinite square root. Define $g := \frac{k}{2} S^{-1} \operatorname{diag}(G)$. Then the objective function in (3.2.1) is equivalent to*

$$x^T D x = -2\|Sx - g\|^2 + 2\|g\|^2.$$

Proof.

$$\begin{aligned} x^T D x &= x^T(\mathcal{K}(G))x \\ &= x^T[\operatorname{diag}(G)e^T + e\operatorname{diag}(G)^T - 2G]x \\ &= x^T \operatorname{diag}(G)(e^T x) + (x^T e)\operatorname{diag}(G)^T x - 2x^T G x \\ &= 2kx^T \operatorname{diag}(G) - 2x^T G x \\ &= 4\langle x, Sg\rangle - 2x^T S^2 x \\ &= -2[\langle Sx, Sx\rangle - 2\langle Sx, g\rangle] \\ &= -2\|Sx - g\|^2 + 2\|g\|^2. \end{aligned}$$

□

Remark 3.2.3. *Theorem 3.2.2 implies that the objective function can be reduced to $\max \|Sx-g\|^2 = \|Sx - \frac{k}{2} S^{-1}\operatorname{diag}(S^2)\|^2$, a convex maximization problem. Note that we started with an **EDM** D, but the equivalence using $G = S^2$ and the properties that $x^T e = k$ allowed for the reduction to S.*

A brute force approach is to partition columns of S into k sections each consisting of n columns, and then enumerate all possible k-sums. Pick the one that is farthest from g. The complexity is $O(n^k)$. This assumes that no particular structures for Sx and g can be exploited.

3.2.3 Semidefinite programming(SDP) relaxation

In this subsection, we present a **SDP** relaxation of (3.2.2). The idea is to append an extra 1 in front of a feasible vector x: $\begin{bmatrix} 1 \\ x \end{bmatrix}$, lift it into a rank-1 matrix $Y_x := \begin{bmatrix} 1 \\ x \end{bmatrix}\begin{bmatrix} 1 \\ x \end{bmatrix}^T$, and relax the rank-1 constraint. During the relaxation stage, we will impose additional redundant constraints, such as $\operatorname{arrow}(Y_x) = e_0$, in order to maintain certain properties of (3.2.2).

SDP reformulation via facial reduction

With respect to matrix variate Y_x, define $\hat{D} := \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix}$, then the objective function of (3.2.2) becomes $\langle D, xx^T\rangle = \langle \hat{D}, Y_x\rangle$.

For the "only-one-element-from-each-set" binary linear constraint, we observe that

$$\begin{aligned} Ax = e &\iff \begin{bmatrix} 1 \\ x \end{bmatrix}^T \begin{bmatrix} -e^T \\ A^T \end{bmatrix} = 0 \\ &\iff Y_x K := \begin{bmatrix} 1 \\ x \end{bmatrix}\begin{bmatrix} 1 \\ x \end{bmatrix}^T \begin{bmatrix} -e^T \\ A^T \end{bmatrix}\begin{bmatrix} -e^T \\ A^T \end{bmatrix}^T = 0 \\ &\iff \langle Y_x, K\rangle = 0 \\ &\iff KY_x = 0, \text{ i.e: } \operatorname{range}(Y_x) \subseteq \operatorname{null}(K) \end{aligned}$$

The last step follows since $K := \begin{bmatrix} -e^T \\ A^T \end{bmatrix} \begin{bmatrix} -e^T \\ A^T \end{bmatrix}^T \succeq 0$ and $Y_x \succeq 0$. This implies that the binary linear constraint on vector x is equivalent to the constraint on the lifted matrix $Y_x : KY_x = 0$.

Now, it remains to consider the structure of Y_x.

Proposition 3.2.4.

$$\left\{ Y \in \mathbb{S}^{nk+1} : \text{rank}(Y) = 1, \text{arrow}(Y) = e_0 \right\} = \left\{ Y = \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}^T : x \in \{0,1\}^{nk} \right\}.$$

Proof. ($\supseteq$): This is obvious.

($\subseteq$): Since Y is symmetric and has rank 1, there exists $\begin{bmatrix} x_0 \\ x \end{bmatrix} \in \mathbb{R}^{nk+1}$ such that $Y = \begin{bmatrix} x_0 \\ x \end{bmatrix} \begin{bmatrix} x_0 \\ x \end{bmatrix}^T$. Since $\text{arrow}(Y) = e_0$, $x_0^2 = 1$ and $x \circ x = x_0 x$. If $x_0 = 1, x \in \{0,1\}^{nk}$; otherwise $x_0 = -1$ and $x \in \{0,-1\}^n$ and it is easy to verify that

$$\left\{ \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 \\ x \end{bmatrix}^T : x \in \{0,1\}^{nk} \right\} = \left\{ \begin{bmatrix} -1 \\ x \end{bmatrix} \begin{bmatrix} -1 \\ x \end{bmatrix}^T : x \in \{0,-1\}^{n} \right\}.$$

□

Therefore, the **SDP** reformulation is

$$(\textbf{SDP}) \qquad \begin{array}{lll} p^* = & \min_{Y \in \mathbb{S}^{nk+1}} & \langle \hat{D}, Y \rangle \\ & & \text{arrow}(Y) = e_0 \\ & & \text{rank}(Y) = 1 \\ & & KY = 0 \end{array}$$

Relaxing the rank-1 constraint

Since the **NP**-hardness of the **SDP** formulation comes from the rank-1 constraint, we now remove this constraint. The **SDP** relaxation of the above model is

$$(\textbf{SDP relax}) \qquad \begin{array}{lll} p^* = & \min_{Y \in \mathbb{S}^{nk+1}} & \langle \hat{D}, Y \rangle \\ & & \text{arrow}(Y) = e_0 \\ & & KY = 0 \end{array}$$

However, the improved processing efficiency of the relaxation model trades off the accuracy of the original model. The rank of an optimal Y now can be greater than 1. The idea now is to impose a "right" amount of redundant constraints in the **SDP** model that reduces the rank of an optimal solution as much as possible, without hurting the processing efficiency of the model too much.

Imposing the Gangster constraint

The Gangster constraint with respect to a Gangster index on a matrix zeros out some of its entries corresponding to the Gangster index. The Gangster constraint in our case comes from the binary linear constraint $Ax = e$. Specifically, for feasible x of (3.2.2), define $D_A := \text{Diag}[\text{diag}(A^T A)] = I$.

Proposition 3.2.5.

$$[A^TA - D_A] \circ xx^T = 0.$$

Proof.

$$\begin{aligned}
Ax = e &\iff A^TAx = A^Te = \mathrm{diag}(A^TA) \\
&\iff A^TAx - D_Ax = A^Te - D_Ax = \mathrm{diag}(A^TA) - \mathrm{Diag}[\mathrm{diag}(A^TA)]x \\
&\iff (A^TA - D_A)x = \mathrm{diag}(A^TA) \circ (e - x) = e - x \\
&\iff (A^TA - D_A)xx^T = (e - x)x^T = ex^T - xx^T \\
&\iff \mathrm{trace}[(A^TA - D_A)xx^T] = \mathrm{trace}[ex^T - xx^T] = \textstyle\sum_{i=1}^{nk} x_i - x_i^2 = 0 \\
&\iff (A^TA - D_A) \circ xx^T = 0.
\end{aligned}$$

□

Following the proposition, we define the unlifted Gangster index J to be $A^TA - I$. Geometrically, J represents the set of off-diagonal indices of the n-by-n diagonal blocks of $Y_x(2:\mathrm{end}, 2:\mathrm{end})$, i.e: the set of star indices of the following matrix:

$$\begin{bmatrix}
\begin{bmatrix} \times & \star & \star \\ \star & \ddots & \star \\ \star & \star & \times \end{bmatrix} & & & \\
& \begin{bmatrix} \times & \star & \star \\ \star & \ddots & \star \\ \star & \star & \times \end{bmatrix} & & \\
& & \ddots & \\
& & & \begin{bmatrix} \times & \star & \star \\ \star & \ddots & \star \\ \star & \star & \times \end{bmatrix}
\end{bmatrix}.$$

To align with the extra dimension of the lifted Y_x, we define our lifted Gangster index $\hat{\mathcal{J}} := \{(0,0)\} \cup \mathcal{J}$. Then, the binary linear constraint $Ax = e$ is equivalent to

$$\begin{bmatrix} 0 \\ A^T \end{bmatrix} \begin{bmatrix} 0 \\ A^T \end{bmatrix}^T - \begin{bmatrix} -1 & 0 \\ 0 & I_{nk} \end{bmatrix} = \mathcal{G}_{\hat{J}}(Y_x) = D_{00} := e_0 e_0^T.$$

Now, the **SDP** relaxation model becomes

$$\begin{aligned}
p^* = \quad \min_{Y \in \mathbb{S}^{nk+1}} \quad & \langle \hat{D}, Y \rangle \\
& \mathrm{arrow}(Y) = e_0 \\
& \mathcal{G}_{\hat{J}}(Y) = D_{00} \\
& KY = 0
\end{aligned} \tag{3.2.3}$$

3.2.4 Doubly non-negative(DNN) relaxation

In this subsection, we split the primal variable Y into two variables $\{Y, R\}$ and apply a doubly non-negative relaxation to (3.2.3).

Recall that a feasible Y_x has the form $\begin{bmatrix}1\\x\end{bmatrix}\begin{bmatrix}1\\x\end{bmatrix}^T$, where $x \in \{0,1\}^{nk}$. Hence, we can impose the redundant element-wise $[0,1]$-bound constraint on Y, i.e: $0 \leq Y \leq 1$.

For the constraint $KY = 0$, we apply the following facial reduction technique. Since $\text{null}(K)$ is a linear subspace of $\mathbb{R}^{nk+1}$, $\{Y_x \in \mathbb{S}_+^{nk+1} : \text{range}(Y_x) \subseteq \text{null}(K)\}$ is a face of $\mathbb{S}_+^{nk+1}$. Hence, for any $V \in \mathbb{R}^{(nk+1)\times(nk+1-k)}$ with full column rank such that $\text{range}(V) = \text{null}(K) = \text{null}(\begin{bmatrix}-e^T\\A^T\end{bmatrix}^T)$,

$$\{Y_x \in \mathbb{S}_+^{nk+1} : \text{range}(Y_x) \subseteq \text{null}(K)\} = V\mathbb{S}_+^{nk+1-k}V^T.$$

We call such V a facial reducer. The facial reduction naturally brings a second primal variable $R \in \mathbb{S}_+^{nk+1-k}$. With this additional variable, we can easily see that

$$KY = 0 \iff Y = VRV^T, R \in \mathbb{S}_+^{nk+1-k}.$$

Next, we derive a redundant trace constraint on Y and transform it onto R.

Proposition 3.2.6.

$$\{Y \in \mathbb{S}^{nk+1} : KY = 0, \text{arrow}(Y) = e_0\} \subseteq \{Y \in \mathbb{S}^{nk+1} : \text{trace}(Y) = k+1\}.$$

Proof. Recall that $K := \begin{bmatrix}-e^T\\A^T\end{bmatrix}\begin{bmatrix}-e^T\\A^T\end{bmatrix}^T$. Since $\text{null}(K) = \text{null}[\begin{bmatrix}-e^T\\A^T\end{bmatrix}^T]$,

$$KY = 0 \iff 0 = DY = \begin{bmatrix}-1 & e^T & ... & 0^T\\ ... & ... & ... & ...\\ -1 & 0^T & ... & e^T\end{bmatrix}\begin{bmatrix}Y_{0,0} & ... & Y_{0,nk}\\ ... & ... & ...\\ Y_{nk,0} & ... & Y_{nk,nk}\end{bmatrix}.$$

By expanding the first column of DY, we get $\sum_{i=1}^n Y_{jn+i,0} = 1, \forall j \in \{0,...,k-1\}$. Since $\text{arrow}(Y) = e_0$, this implies that $\text{trace}(Y) = Y_{0,0} + \sum_{j=1}^k \sum_{i=1}^n Y_{jn+i,0} = 1+k$. □

Now, the facial constraint says that $1 + k = \text{trace}(Y) = \text{trace}(VRV^T) = \text{trace}(RV^TV) = \text{trace}(R)$.

Next, we incorporate all these constraints into the **SDP** relaxation model to form the **DNN** relaxation model. Define

$$\mathcal{Y} := \{Y \in \mathbb{S}^{nk+1} : G_{\hat{j}}(Y) = D_{00}, \text{arrow}(Y) = e_0, 0 \leq Y \leq 1\},\ \mathcal{R} := \{R \in \mathbb{S}_+^{nk+1-k} : \text{trace}(R) = k+1\}.$$

Thus, the **DNN** relaxation model is:

$$(\textbf{DNN}) \qquad \begin{array}{ll}\min_{R,Y} & \langle \hat{D}, Y\rangle\\ \text{s.t.} & Y = VRV^T\\ & Y \in \mathcal{Y}\\ & R \in \mathcal{R}\end{array} \tag{3.2.4}$$

Observe that every feasible Y is non-negative element-wise and every feasible R is **P.S.D.**. Hence, this relaxation model admits the nomenclature doubly non-negative relaxation.

Optimality conditions

Define

$$\mathcal{A}_1 : \mathbb{S}^{nk+1} \to \mathbb{S}^{nk+1} : Y \mapsto Y.$$

and

$$\mathcal{A}_2 : \mathbb{S}^{nk+1-k} \to \mathbb{S}^{nk+1} : R \mapsto -VRV^T.$$

Then,

$$\mathcal{A}_1^* : \mathbb{S}^{nk+1} \to \mathbb{S}^{nk+1} : Z \mapsto Z.$$

and

$$\mathcal{A}_2^* : \mathbb{S}^{nk+1} \to \mathbb{S}^{nk+1-k} : Z \mapsto -V^T ZV.$$

In addition, as the objective function $f(Y, R) = \langle \hat{D}, Y \rangle$,

$$-\nabla_Y f(Y, R) = -\hat{D}.$$

Applying the results in section 2.4.7 gives the following optimality characterization conditions.

A primal-dual pair (Y, R, Z) is optimal if and only if

$$Y = VRV^T, \quad R \in \mathcal{R},\, Y \in \mathcal{Y}, \quad \text{(primal feasibility)} \tag{3.2.5a}$$

$$0 \in -V^T ZV + \mathcal{N}_{\mathcal{R}}(R), \quad \text{(dual } R \text{ feasibility)} \tag{3.2.5b}$$

$$0 \in \hat{D} + Z + \mathcal{N}_{\mathcal{Y}}(Y), \quad \text{(dual } Y \text{ feasibility)} \tag{3.2.5c}$$

By the definition of the normal cone, we can easily obtain the following (3.2.7).

Proposition 3.2.7 (characterization of optimality for **DNN** in (3.2.4))**.** *The primal-dual pair (R, Y, Z) is optimal for (3.2.4) if, and only if, (3.2.5) holds if, and only if,*

$$R = \mathcal{P}_{\mathcal{R}}(R + V^T ZV) \tag{3.2.6a}$$

$$Y = \mathcal{P}_{\mathcal{Y}}(Y - \hat{D} - Z) \tag{3.2.6b}$$

$$Y = VRV^T \tag{3.2.6c}$$

Chapter 4

ADMM algorithm

In this chapter we begin with a survey of some historical methods that motivate the development of **ADMM** algorithms. Then, we apply the Peaceman-Rachford version, i.e., an **ADMM** with intermediate update of multipliers, to solve the model (3.2.4) and obtain tight upper and lower bounds for the simplified Wasserstein barycenter problem. This approach in this thesis follows closely upon the work in [12]. We conclude in Section 4.3 with a review of some historical algorithmic approaches to the general Wasserstein barycenter problem.

4.1 Development of the ADMM algorithm

The alternating direction method of multipliers was first introduced by Gabay, Glowinski, Mercier, and Marrocco in the mid 1970s. The motivation originated from two optimization algorithms: the dual ascent method and the method of multipliers. We next survey these two algorithms and the rationale for the development of **ADMM**.

4.1.1 Dual ascent

Consider the equality constrained convex optimization problem

$$\min f(x) \text{ subject to } Ax = b, \tag{4.1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex, $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$. Its Lagrangian is

$$L(x, y) := f(x) + \langle y, Ax - b\rangle,$$

and the dual function is

$$g(y) := \inf_x L(x, y) = -f^*(-A^T y) - b^T y,$$

and hence the (unconstrained) dual problem is

$$\max_{y\in\mathbb{R}^m} g(y).$$

Problem (4.1.1) admits the following optimality conditions:

1. Primal feasibility: $Ax^* = b$;
2. Dual feasibility: $\nabla f(x^*) + A^T y^* = 0$.

The dual ascent method solves the dual problem using gradient ascent. Assume that an optimal primal-dual pair (x^*, y^*) exists. Then $x^* \in \operatorname{argmin}_x L(x, y^*)$. If in addition $L(x, y^*)$ has a unique minimizer, e.g: f is strictly convex, then $x^* = \operatorname{argmin}_x L(x, y^*)$. Furthermore, assume the dual function g is differentiable so that $\nabla g(y)$ can be evaluated. The dual ascent update is defined as follows:

1. Primal update: $x^{k+1} := \operatorname{argmin}_x L(x, y^k)$;
2. Dual update: $y^{k+1} := y^k + \alpha^k \nabla g(y^k) = y^k + \alpha^k (Ax^{k+1} - b)$.

Here (x^k, y^k) denotes the current iterate and $\alpha^k > 0$ denotes the current step size.

If the step size at each iterate is selected appropriately and some other assumptions hold, i.e: f is proper. Then, the primal-dual iterates converge to an optimal pair. One main benefit of the dual ascent method is dual decomposability, i.e: if the objective function is separable, then its Lagrangian is also separable and the primal updates can be performed in parallel instead of in sequence, which boosts the processing speed. However, as mentioned above, one major disadvantage is that it imposes many restrictions on f, which can fail to hold for many applications.

4.1.2 The method of multipliers

Define the Augmented Lagrangian for (4.1.1) with respect to penalty parameter $\rho > 0$ as

$$\mathcal{L}_\rho(x, y) := f(x) + \langle y, Ax - b \rangle + \frac{\rho}{2} \|Ax - b\|_2^2 .$$

It can be treated as the (unaugmented) Lagrangian associated with the following equality constrained convex optimization problem

$$\min f(x) + \frac{\rho}{2} \|Ax - b\|_2^2 \text{ subject to } Ax = b.$$

Its dual function is

$$g_\rho(y) := \inf_x \mathcal{L}_\rho(x, y),$$

which is differentiable under mild assumptions on (4.1.1).

The method of multipliers update is defined as follows:

1. Primal update: $x^{k+1} := \operatorname{argmin}_x \mathcal{L}_\rho(x, y^k)$;
2. Dual update: $y^{k+1} := y^k + \rho \nabla g_\rho(y^k) = y^k + \rho(Ax^{k+1} - b)$.

Note that the step size at each dual update is fixed at ρ. The intuition is that we want the primal update to resemble the dual feasibility condition (2). As the method progresses, the primal residual $\left\|Ax^k - b\right\|$ approaches 0, yielding primal feasibility and hence optimality.

One advantage of the method of multipliers over dual ascent method is that it imposes less constraints on f. For example, the iterates converge even if f is not strictly convex or takes on the value of positive infinity. However, as $\mathcal{L}_\rho$ is not separable because of the quadratic term, it foregoes the dual decomposability as a trade-off.

4.1.3 The ADMM algorithm

The rationale of the alternating direction methods of multipliers is to combine the decomposability of the dual ascent method with the advanced convergence property of the method of multipliers. It is designed to solve convex optimization problems with two variables:

$$\min f(x) + g(z) \text{ subject to } Ax + Bz = c, \tag{4.1.2}$$

where $x \in \mathbb{R}^n, z \in \mathbb{R}^m, A \in \mathbb{R}^{d\times n}, B \in \mathbb{R}^{d\times m}, c \in \mathbb{R}^d$, both f and g are convex. The augmented Lagrangian of (4.1.2) is

$$\mathcal{L}_\rho(x, z, y) := f(x) + g(z) + \langle y, Ax + Bz - c\rangle + \frac{\rho}{2} \|Ax + Bz - c\|_2^2 .$$

The **ADMM** updates contain two primal variable updates and one dual update:

1. $x^{k+1} := \operatorname{argmin}_x \mathcal{L}_\rho(x, z^k, y^k)$;
2. $z^{k+1} := \operatorname{argmin}_z \mathcal{L}_\rho(x^{k+1}, z, y^k)$;
3. $y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$.

Peaceman-Rachford algorithm

An **ADMM** with intermediate update of multipliers is called the Peaceman-Rachford algorithm. It updates the dual variable twice, one after the x-update and the other after the z-update. Hence, both the x-update and the z-update take into account of the newly informed dual variable.

1. $x^{k+1} := \operatorname{argmin}_x \mathcal{L}_\rho(x, z^k, y^k)$;
2. $y^{k+\frac{1}{2}} := y^k + \rho(Ax^{k+1} + Bz^k - c)$;
3. $z^{k+1} := \operatorname{argmin}_z \mathcal{L}_\rho(x^{k+1}, z, y^{k+\frac{1}{2}})$;
4. $y^{k+1} := y^{k+\frac{1}{2}} + \rho(Ax^{k+1} + Bz^{k+1} - c)$.

Convergence of the ADMM algorithm

The convergence of the **ADMM** algorithm was first proved by Gabay [18]. (In the appendix of [11], Boyd et al. also presents a proof.) Assume:

1. both f and g are proper, l.s.c., and convex;

2. the (unaugmented) Lagrangian of (4.1.2) has a saddle point, i.e., there exists a feasible primal-dual pair (x^*, y^*, z^*) such that

$$\mathcal{L}_0(x^*, y^*, z) \leq \mathcal{L}_0(x^*, y^*, z^*) \leq \mathcal{L}_0(x, y, z^*), \ \forall \text{ feasible primal-dual pair } (x, y, z).$$

Remark 4.1.1. *An equivalent characterization of saddle point (x^*, y^*, z^*) is obtained from first order conditions, i.e.:*

$$\begin{cases} 0 \in \partial_x \mathcal{L}_0(x, y^*, z^*); \\ 0 \in \partial_y \mathcal{L}_0(x^*, y, z^*); \\ 0 \in \partial_z \mathcal{L}_0(x^*, y^*, z). \end{cases}$$

Then,

- The primal residual $r^k := Ax^k + Bz^k - c \to 0$ as $k \to \infty$.
- The objective value converges to the optimal value.
- The dual variable $y^k \to y^*$ as $k \to \infty$.

Optimality conditions and stopping criterion

The optimality conditions of (4.1.2) are

- Primal feasibility: $r^* := Ax^* + Bz^* - c = 0$.
- Dual feasibility:

$$\begin{aligned} &\text{x} \ : \ 0 \in \partial f(x^*) + A^T y^*; \\ &\text{z} \ : \ 0 \in \partial g(z^*) + B^T y^*. \end{aligned}$$

Now, we analyze the primal updates in the **ADMM** step.

1. z^{k+1} minimizes $\mathcal{L}_\rho(x^{k+1}, z, y^k) \iff 0 \in \partial g(z^{k+1}) + B^T y^k + \rho B^T r^{k+1} = \partial g(z^{k+1}) + B^T[y^k + \rho r^{k+1}] = \partial g(z^{k+1}) + B^T y^{k+1} \iff$ The dual z feasibility is satisfied.
2. x^{k+1} minimizes $\mathcal{L}_\rho(x, z^{k+1}, y^k) \iff 0 \in \partial f(x^{k+1}) + A^T y^k + \rho A^T(Ax^{k+1} + Bz^k - c) = \partial f(x^{k+1}) + A^T[y^k + \rho r^{k+1} + \rho B(z^k - z^{k+1})] = \partial f(x^{k+1}) + A^T y^{k+1} + \rho A^T B(z^k - z^{k+1}) \iff \rho A^T B(z^{k+1} - z^k) \in \partial f(x^{k+1}) + A^T y^{k+1}$.

Therefore, we define the dual residual at step k to be $s^k := \rho A^T B(z^k - z^{k-1})$. Clearly, as $(r^k, s^k) \xrightarrow{k \to \infty} (0, 0)$, the optimality conditions tend to be satisfied. Hence, we stop the algorithm when

1. $\left\|r^k\right\|_2 < \epsilon^{primal} := \sqrt{d}\epsilon^{abs} + \epsilon^{rel} \max\{\left\|Ax^k\right\|_2, \left\|Bz^k\right\|_2, \|c\|_2\}$;
2. $\left\|s^k\right\|_2 < \epsilon^{dual} := \sqrt{n}\epsilon^{abs} + \epsilon^{rel} \left\|A^T y^k\right\|_2$.

Empiric results [11] suggest that $\epsilon^{rel} \in \{10^{-3}, 10^{-4}\}$ and ϵ^{abs} depends on the size of the primal variable.

Tuning the step size

One heuristic is to keep the primal and dual residual norms close to each other as they converge to 0, e.g: keeping them within a factor of μ . A large penalty ρ prioritizes primal feasibility over dual feasibility and a small penalty ρ prioritizes dual feasibility over primal feasibility. Hence, we want to scale ρ up if primal residual overshoots dual residual and scale ρ down if dual residual overshoots primal residual. Let τ^{incr} and τ^{decr} be scaling factors. We define the ρ-update as follows:

$$\rho^{k+1} := \begin{cases} \tau^{incr}\rho^k, & \left\|r^k\right\|_2 > \mu \left\|s^k\right\|_2; \\ \frac{\rho^k}{\tau^{decr}}, & \left\|s^k\right\|_2 > \mu \left\|r^k\right\|_2; \\ \rho^k, & \text{otherwise.} \end{cases}$$

4.2 The simplified Wasserstein barycenter problem

In this section, we apply the Peaceman-Rachford algorithm to our problem of interest.

4.2.1 Convergence of the ADMM algorithm

At first, we prove that applying the **ADMM** algorithm to the **DNN** model (3.2.4) results in convergence.

Recall the convergence conditions of the **ADMM** algorithm in section 4.1.3. Since our objective function is linear and any linear function is proper, l.s.c., and convex, the first condition is satisfied. It suffices to show that the second condition is also satisfied.

In our **DNN** model (3.2.4), the objective function is continuous and the feasible set is compact. By the extreme value theorem, an optimal primal pair (Y^*, R^*) always exists. By the strong duality theorem, a corresponding optimal dual variable Z^* exists.

Proposition 4.2.1. *The optimal primal-dual pair (Y^*, R^*, Z^*) is a saddle point of the (unaugmented) Lagrangian*

$$\mathcal{L}_0(Y, R, Z) = \langle \hat{D}, Y\rangle + \langle Z, Y - VRV^T\rangle + \mathbb{1}_{\mathcal{Y}}(Y) + \mathbb{1}_{\mathcal{R}}(R).$$

Proof. Note that

$$\begin{cases} \partial_Y \mathcal{L}_0(Y, R^*, Z^*) = \hat{D} + Z^* + \mathcal{N}_{\mathcal{Y}}(Y); \\ \partial_R \mathcal{L}_0(Y^*, R, Z^*) = -V^T Z^* V + \mathcal{N}_{\mathcal{R}}(R). \end{cases}$$

By the optimality conditions of the **DNN** model (3.2.5), we have

$$\begin{cases} 0 \in \partial_Y \mathcal{L}_0(Y, R^*, Z^*); \\ 0 \in \partial_R \mathcal{L}_0(Y^*, R, Z^*). \end{cases}$$

In addition,

$$0 = Y^* - VR^*V^T = \nabla_Z \mathcal{L}_0(Y^*, R^*, Z).$$

Hence, (Y^*, R^*, Z^*) is indeed a saddle point of the (unaugmented) Lagrangian. □

4.2.2 Peaceman-Rachford splitting method(PRSM) updates

Recall the **DNN** model (3.2.4). Its augmented Lagrangian is

$$\mathcal{L}_\beta(Y,R,Z) := \langle \hat{D}, Y\rangle + \langle Z, Y - VRV^T\rangle + \frac{\beta}{2}\|Y - VRV^T\|_F^2 + \mathbb{1}_{\mathcal{Y}}(Y) + \mathbb{1}_{\mathcal{R}}(R).$$

Hence, its **PRSM** updates are

1. $R^{k+1} := \operatorname{argmin}_{R\in\mathcal{R}} \mathcal{L}_\beta(R, Y^k, Z^k)$;
2. $Z^{k+\frac{1}{2}} := Z^k + \beta(Y^k - VR^{k+1}V^T)$;
3. $Y^{k+1} := \operatorname{argmin}_{Y\in\mathcal{Y}} \mathcal{L}_\beta(R^{k+1}, Y, Z^{k+\frac{1}{2}})$;
4. $Z^{k+1} := Z^{k+\frac{1}{2}} + \beta(Y^{k+1} - VR^{k+1}V^T)$.

Primal updates

As for the R-update,

$$\begin{array}{lll}
\operatorname{argmin}_{R\in\mathcal{R}} \mathcal{L}_\beta(R, Y^k, Z^k) & = \operatorname{argmin}_{R\in\mathcal{R}} \|Y^k - VRV^T + \frac{1}{\beta}Z^k\|_F^2 & \text{by completing the square} \\
& = \operatorname{argmin}_{R\in\mathcal{R}} \|V^TY^kV - R + \frac{1}{\beta}V^TZ^kV\|_F^2 & \text{since } V^TV = I \\
& = \operatorname{argmin}_{R\in\mathcal{R}} \|R - V^T(Y^k + \frac{1}{\beta}Z^k)V\|_F^2 & \\
& = \mathcal{P}_{\mathcal{R}}[V^T(Y^k + \frac{1}{\beta}Z^k)V] & =: \mathcal{P}_{\mathcal{R}}(M) \\
& = U\operatorname{Diag}[\mathcal{P}_{\Delta_{k+1}}(d)]U^T & M = U\operatorname{Diag}(d)U^T
\end{array}$$

where $\mathcal{P}_{\Delta_{k+1}}$ denotes the projection onto the simplex $\Delta_{k+1} := \{x \in \mathbb{R}^n_+ : \langle e, x\rangle = 1 + k\}$.

As for the Y-update,

$$\begin{array}{lll}
\operatorname{argmin}_{Y\in\mathcal{Y}} \mathcal{L}_\beta(R^{k+1}, Y, Z^{k+\frac{1}{2}}) & = \operatorname{argmin}_{Y\in\mathcal{Y}} \|Y - [VR^{k+1}V^T - \frac{1}{\beta}(\hat{D} + Z^{k+\frac{1}{2}})]\|_F^2 & \text{by completing the square} \\
& = \mathcal{P}_{\mathcal{Y}}[VR^{k+1}V^T - \frac{1}{\beta}(\hat{D} + Z^{k+\frac{1}{2}})] & \\
& = \mathcal{P}_{box}[\mathcal{G}_{\hat{\mathcal{J}}}[VR^{k+1}V^T - \frac{1}{\beta}(\hat{D} + Z^{k+\frac{1}{2}})]] &
\end{array}$$

where $\mathcal{G}_{\hat{\mathcal{J}}}$ shoots the Gangster entries to 0 and $\mathcal{P}_{box}$ projects onto the polyhedral set $\{Y \in \mathbb{S}^{nk+1} : Y_{ij} \in [0,1]\}$.

Dual updates

The correct choice of the Lagrange dual multiplier Z is important in obtaining strong lower bound. In addition, if the set of dual multipliers for all iterations is compact, then it indicates the stability of the primal problem. If an optimal Z^* for (3.2.4) is known in advance, then there is no need to impose the primal feasibility constraint $Y = VRV^T$. Hence, following the idea of exploiting redundant constraints, we aim to identify certain properties of an optimal dual multiplier and impose that property at each iteration of our algorithm.

Fact 4.2.2. *Define* $\mathcal{Z}_A := \{Z \in \mathbb{S}^{nk+1} : (Z+\hat{D})_{i,i} = 0, (Z+\hat{D})_{0,i} = 0, (Z+\hat{D})_{i,0} = 0, i = 1, ..., nk\}$. *Then, for every optimal primal-dual pair* (Y^*, R^*, Z^*) *to (3.2.4),* $Z^* \in \mathcal{Z}_A$.

The proof of this fact uses the dual Y feasibility condition (3.2.5c) and a reformulation of the Y-feasible set. The details are in [19, Thm 2.14] and [12]. This fact suggested that instead of updating Z as above, we should project the dual variable onto $\mathcal{Z}_A$ at each iterate, i.e:

- $Z^{k+\frac{1}{2}} := Z^k + \beta \mathcal{P}_{\mathcal{Z}_A}(Y^k - VR^{k+1}V^T)$;
- $Z^{k+1} := Z^{k+\frac{1}{2}} + \beta \mathcal{P}_{\mathcal{Z}_A}(Y^{k+1} - VR^{k+1}V^T)$.

4.2.3 Relaxed Peaceman-Rachford splitting method (rPRSM)

In this subsection, we present a relaxed version of the **PRSM** algorithm called **rPRSM**. The relaxation parameter is denoted by γ.

Algorithm 2 rPRSM

Initialization: $Y^0 = 0 \in S^{nk+1}, Z^0 = P_{Z_A}(0), \beta = max(\lfloor \frac{nk+1}{k} \rfloor, 1), \gamma = 0.9$
while termination criteria are not met **do**
$R^{k+1} = U\,\mathrm{Diag}[P_{\Delta_{k+1}}(d)]U^T$ where $U\,\mathrm{Diag}(d)U^T = eig(V^T(Y^k + \frac{1}{\beta}Z^k)V)$
$Z^{k+\frac{1}{2}} = Z^k + \gamma\beta P_{Z_A}(Y^k - VR^{k+1}V^T)$
$Y^{k+1} = P_{box}[G_j(VR^{k+1}V^T - \frac{1}{\beta}(\hat{D} + Z^{k+\frac{1}{2}}))]$
$Z^{k+1} = Z^{k+\frac{1}{2}} + \gamma\beta P_{Z_A}(Y^{k+1} - VR^{k+1}V^T)$
end while

4.2.4 Bounding and duality gaps

The Lagrangian dual function to the **DNN** model is

$$\begin{aligned} g : \mathbb{S}^{nk+1} \to \mathbb{R} : Z &\mapsto \min_{R\in\mathcal{R}, Y\in\mathcal{Y}} \langle \hat{D}, Y\rangle + \langle Z, Y - VRV^T\rangle \\ &= \min_{Y\in\mathcal{Y}, R\in\mathcal{R}} \langle \hat{D} + Z, Y\rangle - \langle Z, VRV^T\rangle \\ &= \min_{Y\in\mathcal{Y}} \langle \hat{D} + Z, Y\rangle + \min_{R\in\mathcal{R}} (-\langle V^TZV, R\rangle) \\ &= \min_{Y\in\mathcal{Y}} \langle \hat{D} + Z, Y\rangle - \max_{R\in\mathcal{R}} \langle V^TZV, R\rangle \\ &= \min_{Y\in\mathcal{Y}} \langle \hat{D} + Z, Y\rangle - \max_{\|v\|^2=(k+1)} v^TV^TZVv \\ &= \min_{Y\in\mathcal{Y}} \langle \hat{D} + Z, Y\rangle - (k+1)\lambda_{max}(V^TZV). \end{aligned}$$

Hence, at iteration k, a lower bound to the optimal value of the **DNN** model is

$$g(Z^k) = \min_{Y\in\mathcal{Y}} \langle \hat{D} + Z^k, Y\rangle - (k+1)\lambda_{max}(V^TZ^kV).$$

As for the upper bound, we consider two strategies for finding feasible solutions to the **BCQP**. Let $Y(2:end, 2:end)$ denote the unlifted part of the output matrix Y for the algorithm.

The first column approach is to take the first column of $Y(2:end, 2:end)$ and compute its nearest feasible solution to **BCQP**. It is equivalent to signal only the maximum weight index for each consecutive block of length n. The proof is in [12, section 3.2.2].

The dominant eigenvector approach is to take the dominant eigenvector of $Y(2:end, 2:end)$ and compute its nearest feasible solution to **BCQP**. It is again equivalent to signal only the maximum weight index for each consecutive block of length n.

Then, we compare the objective values for both approaches and select the upper bound with smaller magnitude.

The relative duality gap at the current iterate k is defined to be $\frac{UB_k - LB_k}{|UB_k| + |LB_k| + 1}$ where UB_k denotes upper bound at the current iterate and LB_k denotes lower bound at the current iterate.

4.2.5 Stopping criterion

By Proposition 3.2.7, we can define the primal and dual residuals of the **rPRSM** algorithm at iterate k as follows:

- Primal residual $r^k := Y^k - VR^kV^T$;
- Dual-R residual $s_R^k := R^k - \mathcal{P}_{\mathcal{R}}[R^k + V^T Z^k V]$;
- Dual-Y residual $s_Y^k := Y^k - \mathcal{P}_{\mathcal{Y}}[Y^k - \hat{D} - Z^{k+\frac{1}{2}}]$.

We terminate the algorithm once one of the following conditions is satisfied:

- The maximum number of iterations$(maxiter) := 10^4 + k(nk+1)$ is reached;
- The relative duality gap is upper bounded by $\epsilon := 10^{-5}$;
- $KKTres := \max\{r^k, s_R^k, s_Y^k\} < \eta := 10^{-5}$;
- Both the least upper bound and the greatest lower bound have not changed for boundCounterMax:=200 times.

4.2.6 Speed-up

Adaptive step size

We apply the heuristic idea presented in Section 4.1.3, namely we bound the gap between the primal and dual residual norms within a factor of $\mu := 2$ as they converge to 0. This guarantees that they converge to 0 at about the same rate and one residual will not overshoot the other residual by too much. Since a large penalty β prioritizes primal feasibility over dual feasibility and a small penalty β prioritizes dual feasibility over primal feasibility, we scale β by a factor of $\tau_{inc} := 2$ if the primal residual overshoots the dual residual by a factor of μ and scale β down by a factor of $\tau_{dec} := 2$ if the dual residual overshoots the primal residual by a factor of μ. Otherwise, we keep β unchanged. Specifically,

$$\beta^{k+1} := \begin{cases} \tau^{incr}\beta^k, & \left\|r^k\right\|_2 > \mu \left\|s^k\right\|_2; \\ \frac{\beta^k}{\tau^{decr}}, & \left\|s^k\right\|_2 > \mu \left\|r^k\right\|_2; \\ \beta^k, & \text{otherwise.} \end{cases}$$

Transformation of EDM

In this subsection, we explore techniques that can be applied to $\hat{D}$ without contaminating the objective function, such as scaling by a factor $\delta > 0$, or translation by $\alpha \in \mathbb{R}$. Define the orthogonal projection map $P_V := VV^T$. Then,

$$\begin{aligned} \langle \hat{D}, Y \rangle &:= \langle \hat{D} + \alpha I, Y \rangle - (n+1)\alpha \\ &= \langle \hat{D} + \alpha I, P_V Y P_V \rangle - (n+1)\alpha \\ &= \langle (P_V \hat{D} P_V + \alpha I), Y \rangle - (n+1)\alpha. \end{aligned}$$

Hence,

$$\begin{aligned} \langle \hat{D}, Y \rangle \text{ is minimized} &\iff \delta\langle \hat{D}, Y \rangle = \langle \delta(P_V \hat{D} P_V + \alpha I), Y \rangle - (n+1)\delta\alpha \text{ is minimized} \\ &\iff \langle \delta(P_V \hat{D} P_V + \alpha I), Y \rangle \text{ is minimized.} \end{aligned}$$

This lets us transform $\hat{D}$ into $\delta(P_V \hat{D} P_V + \alpha I)$ without contaminating the objective function.

Scaling EDM by $\delta < 0$

Numerical experiments show that once we scale $\hat{D}$ by some $\delta < 0$, the convergence becomes faster for the aforementioned input data distributions. There seems to be an optimal δ that minimizes the number of iterations for convergence.

4.2.7 Input data distributions

In this subsection, we investigate some input data distributions for which the proposed **rPRSM** algorithm achieves efficient convergence. The MATLAB command *randn* returns a random number sampled from the normal distribution with mean 0 and variance 1. We found that when we group each cluster of n points together following the standard normal distribution, the **rPRSM** algorithm converges very efficiently.

The following table provides running time and relative gap comparisons for a sample of problems.

Table 4.2.1 Performance comparison: **rPRSM** and CVX solvers

Specifications			Time (s)		Rel. Dist. to Sol.	
d	n	k	**rPRSM**	Mosek	**rPRSM**	Mosek
2	7	5	2.33e-01	3.66e-01	9.80e-08	2.41e-09
2	8	6	3.90e-01	6.94e-01	2.76e-10	5.91e-11
2	9	7	3.53e-01	1.30e+00	6.59e-07	1.55e-11
2	10	8	3.75e-01	3.92e+00	4.82e-08	4.96e-12
2	11	9	4.63e-01	1.30e+01	1.92e-09	2.21e-12
2	12	10	5.41e-01	3.09e+01	9.32e-10	8.41e-10
2	13	11	7.22e-01	7.31e+01	1.83e-08	2.94e-11

This table shows the scalability of the **rPRSM** algorithm for data of large size.

Table 4.2.2 Scalability of **rPRSM** algorithm

d	n	k	Time(s)	**KKT** residual	Relative duality gap
3	3	3	2.36e-02	2.20e-07	7.52e-15
4	4	4	1.38e-01	3.10e-08	9.95e-17
5	5	5	1.80e-01	7.02e-09	3.42e-16
6	6	6	3.06e-01	1.89e-08	9.09e-15
7	7	7	4.79e-01	1.19e-06	1.65e-14
8	8	8	3.16e-01	1.51e-06	5.83e-15
9	9	9	5.11e-01	1.43e-07	1.42e-14
10	10	10	5.46e-01	1.51e-07	1.46e-14
11	11	11	2.71e-01	7.38e-09	3.01e-14
12	12	12	1.01e+00	2.34e-08	2.02e-14
13	13	13	1.48e+00	4.76e-09	1.64e-14
14	14	14	2.98e+00	1.21e-06	2.75e-14
15	15	15	1.54e+00	9.83e-08	1.10e-14
16	16	16	1.27e+00	6.76e-08	1.70e-14
17	17	17	1.80e+00	1.36e-08	-2.46e-14
18	18	18	2.44e+00	2.93e-06	3.17e-15
19	19	19	3.19e+00	9.19e-10	1.15e-14
20	20	20	5.53e+00	1.56e-09	-4.15e-15
21	21	21	6.25e+00	1.53e-08	-3.86e-14
22	22	22	1.38e+01	2.67e-06	-1.32e-14
23	23	23	1.35e+01	4.16e-09	-1.42e-14
24	24	24	1.64e+01	8.28e-07	3.56e-14
25	25	25	2.72e+01	1.73e-09	-8.10e-16

If the proposed **rPRSM** algorithm outputs both tight lower and upper bounds for all input data distributions, doesn't this imply that $\mathbf{P} = \mathbf{NP}$? Note that the above result is achieved when each group of input points are clustered to each other according to a normal distribution. It is suspicious to assume that the same efficient convergence result still holds for different input data distributions. In fact, I will propose two particular input data distributions for which the duality gap between the optimal value of the **BCQP** formulation and the lower bound is non-trivial. Both of them share the same characteristic that more than one optimal solutions of the the simplified Wasserstein barycenter problem problem exist. In this circumstance, the **rPRSM** algorithm fails to break ties among them, resulting in a non-trivial duality gap.

A simple example

At first, we consider the simplest case where $n = k = 2$. Define $S_1 := \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 10 \\ 0 \end{bmatrix} \right\}$ and $S_2 := \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right\}$. Clearly, the optimal solution of the simplified Wasserstein barycenter problem with respect to this data distribution is to pick the first point of S_1 and either the first or the

second point of S_2. The former selection matches the solution vector $x = \begin{bmatrix}1\\0\\1\\0\end{bmatrix}$ corresponding to the lifted matrix $\begin{bmatrix}1&1&0&1&0\\1&1&0&1&0\\0&0&0&0&0\\1&1&0&1&0\\0&0&0&0&0\end{bmatrix}$ of the **DNN** formulation. The latter selection matches the solution vector $x = \begin{bmatrix}1\\0\\0\\1\end{bmatrix}$ corresponding to the lifted matrix $\begin{bmatrix}1&1&0&0&1\\1&1&0&0&1\\0&0&0&0&0\\0&0&0&0&0\\1&1&0&0&1\end{bmatrix}$ of the **DNN** formulation. Observe that the convex combination of these two matrices with coefficients $\{0.5, 0.5\}$ is $\tilde{Y} = \begin{bmatrix}1&1&0&0.5&0.5\\1&1&0&0.5&0.5\\0&0&0&0&0\\0.5&0.5&0&0.5&0\\0.5&0.5&0&0&0.5\end{bmatrix}$ whose facially reduced component $\tilde{R} = \begin{bmatrix}2&1&0\\1&0.5&0\\0&0&0.5\end{bmatrix}$ has rank 2.

Recall the Lagrangian dual function that we used in section 4.2.4 for computing the lower bound:

$$g(Z) = \min_{Y \in \mathcal{Y}} \langle \hat{D} + Z, Y \rangle - \max_{R \in \mathcal{R}} \langle V^T Z V, R \rangle.$$

With $\tilde{Z} := \begin{bmatrix}-0.3619&0&0&0&0\\0&0&1.3699&-1&-1\\0&1.3699&0&-1.5826&-1.5826\\0&-1&-1.5826&0&0.7873\\0&-1&-1.5826&0.7873&0\end{bmatrix}$, the **rPRSM** algorithm terminates with a **KKT** residual of 8.9157e-11.

With $\hat{D} = \begin{bmatrix}0&0&0&0&0\\0&0&0&1&1\\0&0&0&101&101\\0&1&101&0&0\\0&1&101&0&0\end{bmatrix}$, we have $g(\hat{Z}) = 1.6381 < 2 = \langle \hat{D}, \tilde{Y} \rangle$, admitting a strictly positive duality gap.

Odd wheels

We next present another input data distribution for which the duality gap between the optimal value of the **BCQP** formulation and the Lagrangian dual value is non-trivial. The issue is again the non-uniqueness of the optimal solutions and the **rPRSM** algorithm fails to break ties among them.

The data distributions compose of a wheel with an odd number of sets, hence we call it an odd wheel. Given problem size parameters (k, n, d), define

- $\theta_k := \frac{2\pi}{k}$.
- a set of k centroids encoded by a matrix $C \in \mathbb{R}^{k\times 2}$ such that

$$C(i,:) = \begin{bmatrix}\cos(i-1)\theta_k & \sin(i-1)\theta_k\end{bmatrix}, i = 1, ..., k.$$

- the radius of each cluster $r_k := \frac{\sqrt{\cos(\theta_k-1)^2+\sin\theta_k^2}}{4}$.
- the set of input points encoded by a matrix $P := (C \otimes e) + r_k(e \otimes C) \in \mathbb{R}^{k^2,2}$.

When k is odd, there exists more than one optimal solution. A simple example with $k = 3 = n$ is as follows:

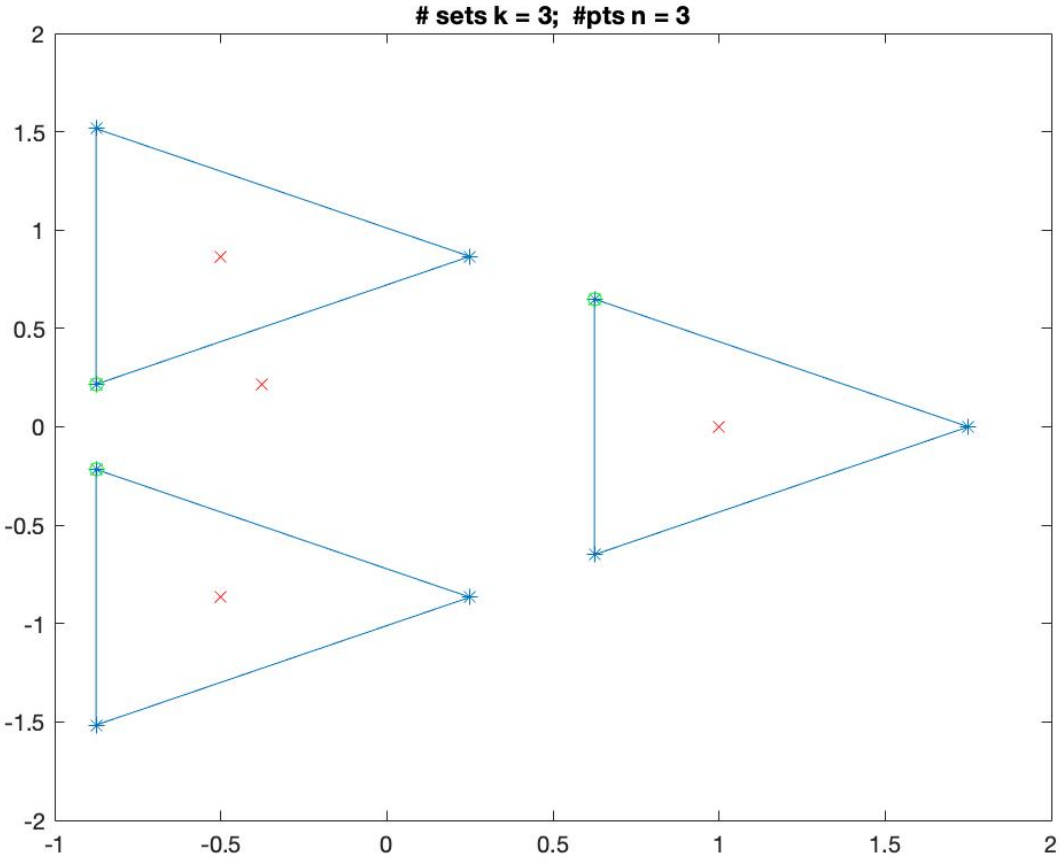

Figure 4.2.1: k=3=n

A simple inspection of the picture shows that reflecting the selected green points along the x-axis gives another optimal solution. In fact, for this example, six different optimal solutions exist.

However, when k is even, only one optimal solution exists and the duality gap becomes trivial. An example with $k = 6 = n$ is as follows:

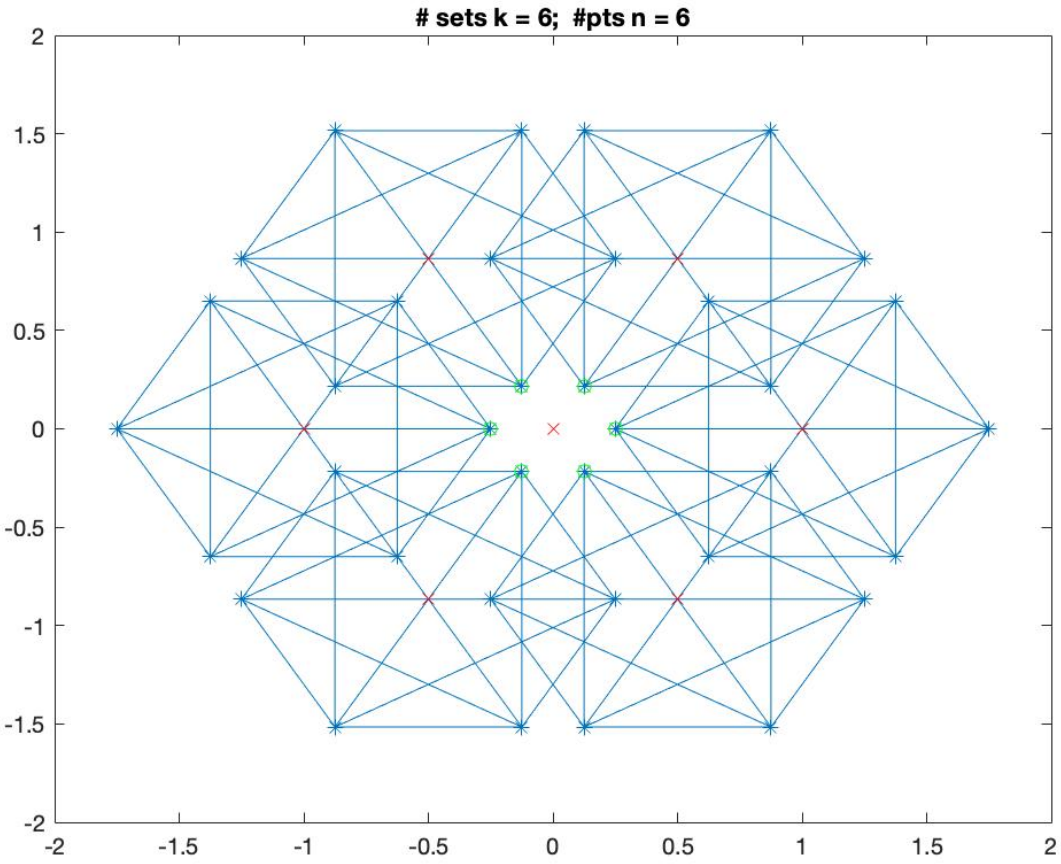

Figure 4.2.2: k=6=n

A heuristic strategy for breaking ties

We have demonstrated that the existence of more than one optimal solutions gives a non-trivial duality gap. However, this does not mean that we are unable to find an optimal solution of the simplified Wasserstein barycenter problem from the outputs of the proposed **rPRSM** algorithm.

Let's consider the odd wheel case (4.2.1). The unlifted part of the output matrix Y is

$$\begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0.375 & 0 & 0.125 \\ 0.125 & 0 & 0.375 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0.125 & 0.375 & 0 \\ 0.375 & 0.125 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0.375 & 0.125 \\ 0 & 0 & 0 \\ 0 & 0.125 & 0.375 \end{bmatrix} & \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} & \begin{bmatrix} 0.375 & 0.125 & 0 \\ 0 & 0 & 0 \\ 0.125 & 0.375 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0.125 & 0.375 \\ 0 & 0.375 & 0.125 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0.375 & 0 & 0.125 \\ 0.125 & 0 & 0.375 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{bmatrix}.$$

We can treat it as a probability matrix where the entry at index (ij) denotes the weight for selecting the j^{th} point in the i^{th} set. For example, in order to break ties for the first group, we can either select the second point or the third point, as they are equally weighted. However, once the second point in the first group is selected, we must select the first point in the second group and the second point in the third group, each attaining an equal weight of $0.375 > 0.125$. Similarly, once the third point in the first group is selected, we must select the third point in the second group and the first point in the third group by the same logic. This strategy guarantees the sum of probabilities for the selected point at each group is maximal.

This approach also suggests a better upper bound approximation method for the **BCQP** when more than one optimal solutions of the simplified Wasserstein barycenter problem exist, compared to the first column approach and the dominant eigenvector approach.

4.3 Historical algorithmic approaches to the Wasserstein barycenter problem

In this section, we survey some past algorithms proposed for solving the general Wasserstein barycenter problem. Unsurprisingly, either their running time depend exponentially on one of the input parameters or they fail to approximate the optimal value well.

4.3.1 Algorithms with time complexity exponential in d

The "fixed-support approximations" approach assumes that the barycenter is supported on a polynomial-sized set, hence the optimization problem becomes an efficiently solvable **LP**. However, the pitfall is that it implicitly requires S to be an ϵ-cover of the space $\mathbb{R}^d$, meaning any point in $\mathbb{R}^d$ must be approximated within error ϵ by some point in S. This implicit requirement costs all fixed-support approximation algorithms running time $\Omega[(\frac{R}{\epsilon})^d]$, which is exponential in d. Another issue is that the accuracy suffers due to the running time's dependence to $\frac{1}{\epsilon}$. Similar approaches,

such as the Frank-Wolfe algorithm and the Functional Gradient Descent algorithm, have the same pitfalls.

It's interesting to investigate whether the efficiency improves when d is fixed. In fact, [5] shows that the answer is yes. Specifically, (3.1.2) can be computed in $\text{poly}(n, k, \log U)$ time.

4.3.2 Algorithms with time complexity exponential in k

Since (3.1.2) is equivalent to (3.1.3), some algorithms just solve the **MOT** problem by brute force and transform the solution into an optimal solution of (3.1.2). Since (3.1.3) is a **LP** with n^k variables, it takes $\Omega(n^k)$ running time [6, 9].

4.3.3 Polynomial-time approximation algorithms with a factor of 2

An approximation algorithm called 2-approximation restricts the support of the candidate probability distribution onto the union of the supports of given probability distributions, hence reduces from n^k weight variables to only nk weight variables. [10] proved that the optimal value at most doubles the optimal value of the standard Wasserstein barycenter problem.

4.3.4 Algorithms based on entropic regularization

Some algorithms used entropic regularization for large-scale optimal transport problems. The idea is to penalize the objective function by an entropy cost, which makes it strongly convex and easier to optimize. However, [2] shows that this approach becomes inefficient for computing the general Wasserstein barycenter in high dimensions.

4.3.5 Developing efficient algorithms by exploiting structures of input distributions

Despite these theoretical hardness results, the broad applicability of the Wasserstein barycenter problem over discrete probability distributions motivated researchers to understand properties of input data and probability distributions under which efficient approximation algorithms can be developed.

One candidate is the uniform probability distribution. Unfortunately, Theorem 5.1 of [2] shows that this does not help in improving the efficiency of existing algorithms.

However, for probability distributions with certain structures, efficient computation of Wasserstein barycenter is possible. [3] shows that the Gaussian distributions, or more generally location-scatter families, assist in constructing polynomial-time algorithms for the Wasserstein barycenter problem. In addition, [15] shows that probability distributions represented by convolution neural network generative models and data distributions supported on low-dimensional manifolds assist in obtaining accurate empirical results.

In this thesis, as we focus on the simplified Wasserstein barycenter problem, we concern only the input data distributions. In particular, we investigated two types of data distributions, with one drawn from a Gaussian process and the other represents a graph structure for which tie of distances exists.

Chapter 5

Conclusion

In this thesis, we identified a **NP**-hard computational problem called the Simplified Wasserstein Barycenter problem that has applications in various fields of data science. For the sake of efficiently approximating the solution of this problem, we formulated the problem as a binary constrained quadratic program and applied doubly non-negative relaxations to it. In order to solve this relaxed optimization problem, we applied a relaxed Peaceman-Rachford (**rPRSM**) algorithm, an **ADMM** with intermediate update of multipliers, to compute tight lower and upper bounds on the optimal values of the Simplified Wasserstein Barycenter problem. The empirical numerical results suggest that both the efficiency and accuracy of our algorithm depend on input data distributions. For examples, as for input data sampled from a standard normal distribution, the accuracy of our algorithm is comparable to the state of the art **SDP** solvers such as Mosek, and the rate of convergence of our algorithm even outperforms those **SDP** solvers. However, as for input data with multiple optimal solutions, the algorithm has difficulty breaking ties among them, which results in a loose lower bound. Some heuristic approaches which address this issue include treating the output matrix as a probability matrix and select points based on their index weights.

As for future research, one direction is to identify more types of input data distributions for which the proposed algorithm either achieves great efficiency and accuracy, or has difficulty achieving either good efficiency or accuracy. Another direction is to explore more speeding-up techniques for the proposed algorithm. The idea of adaptive penalty parameter surveyed in [11] seems to work well for the standard normal data distribution. However, [29] suggests a more advanced technique for adapting the penalty parameter.

References

[1] A. Alfakih, *Euclidean distance matrices and their applications in rigidity theory*, Springer, Cham, 2018.

[2] J. Altschuler and E. Boix-Adserà, *Wasserstein barycenters are NP-hard to compute*, SIAM J. Math. Data Sci., 4 (2022), pp. 179–203.

[3] J. Altschuler, S. Chewi, P. R. Gerber, and A. Stromme, *Averaging on the bures-wasserstein manifold: dimension-free convergence of gradient descent*, in Advances in Neural Information Processing Systems, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds., vol. 34, Curran Associates, Inc., 2021, pp. 22132–22145.

[4] J. M. Altschuler, *Transport and Beyond: Efficient Optimization over Probability Distributions*, PhD thesis, Massachusetts Institute of Technology, 2018.

[5] J. M. Altschuler and E. Boix-Adserà, *Wasserstein barycenters can be computed in polynomial time in fixed dimension*, J. Mach. Learn. Res., 22 (2021), pp. Paper No. 44, 19.

[6] E. Anderes, S. Borgwardt, and J. Miller, *Discrete wasserstein barycenters: Optimal transport for discrete data*, 2015.

[7] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press, 2006.

[8] S. J. Axler, *Linear Algebra Done Right*, Undergraduate Texts in Mathematics, Springer, New York, 1997.

[9] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, *Iterative Bregman Projections for Regularized Transportation Problems*, SIAM Journal on Scientific Computing, 2 (2015), pp. A1111–A1138.

[10] S. Borgwardt, *An LP-based, strongly-polynomial* 2*-approximation algorithm for sparse Wasserstein barycenters*, tech. rep., 2017.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Machine Learning, 3 (2011), pp. 1–122.

[12] F. Burkowski, H. Im, and H. Wolkowicz, *A Peaceman-Rachford splitting method for the protein side-chain positioning problem*, tech. rep., University of Waterloo, Waterloo, Ontario, 2022. arxiv.org/abs/2009.01450,21.

[13] K. Chaudhuri and M. Sugiyama, eds., *Probabilistic Multilevel Clustering via Composite Transportation Distance*, vol. 89 of Proceedings of Machine Learning Research, PMLR, 16–18 Apr 2019.

[14] S. Chiappa and R. Calandra, eds., *Context Mover's Distance & Barycenters: Optimal Transport of Contexts for Building Representations*, vol. 108 of Proceedings of Machine Learning Research, PMLR, 26–28 Aug 2020.

[15] S. Cohen, M. Arbel, and M. P. Deisenroth, *Estimating barycenters of measures in high dimensions*, 2020.

[16] D. Drusvyatskiy and H. Wolkowicz, *The many faces of degeneracy in conic optimization*, 2017.

[17] F. Elvander, I. Haasler, A. Jakobsson, and J. Karlsson, *Multi-marginal optimal transport using partial information with applications in robust localization and sensor fusion*, Signal Processing, 171 (2020).

[18] D. Gabay, *Chapter ix applications of the method of multipliers to variational inequalities*, in Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems, M. Fortin and R. Glowinski, eds., vol. 15 of Studies in Mathematics and Its Applications, Elsevier, 1983, pp. 299–331.

[19] N. Graham, H. Hu, H. Im, X. Li, and H. Wolkowicz, *A restricted dual Peaceman-Rachford splitting method for a strengthened DNN relaxation for QAP*, INFORMS J. Comput., 34 (2022), pp. 2125–2143.

[20] W. M. M. Heinz H. Bauschke, *An introduction to convexity, optimization, and algorithms*, in An Introduction to Convexity, Optimization, and Algorithms, 2022.

[21] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department, R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, eds., Boston, MA, 1972, Springer US, pp. 85–103.

[22] N. Krislock and H. Wolkowicz, *Euclidean distance matrices and applications*, in Handbook on semidefinite, conic and polynomial optimization, vol. 166 of Internat. Ser. Oper. Res. Management Sci., Springer, New York, 2012, pp. 879–914.

[23] D. Precup and Y. W. Teh, eds., *Multilevel Clustering via Wasserstein Means*, vol. 70 of Proceedings of Machine Learning Research, PMLR, 06–11 Aug 2017.

[24] B. N. B. N. Pshenichnyi, *Necessary conditions for an extremum by B. N. Pshenichnyi. Translation edited by Lucien W. Neustadt. Translated by Karol Makowski.*, Pure and applied mathematics, 4, M. Dekker, New York, 1971.

[25] R. Rockafellar, *Convex analysis*, Princeton Landmarks in Mathematics, Princeton University Press, Princeton, NJ, 1997. Reprint of the 1970 original, Princeton Paperbacks.

[26] S. Srivastava, C. Li, and D. B. Dunson, *Scalable bayes via barycenter in Wasserstein space*, J. Mach. Learn. Res., 19 (2018), pp. 312–346.

[27] L. Tuncel, *Polyhedral and semidefinite programming methods in combinatorial optimization*, in Polyhedral and Semidefinite Programming Methods in Combinatorial Optimization, 2010.

[28] E. P. Xing and T. Jebara, eds., *Wasserstein Propagation for Semi-Supervised Learning*, vol. 32 of Proceedings of Machine Learning Research, Bejing, China, 22–24 Jun 2014, PMLR.

[29] Z. Xu, M. A. T. Figueiredo, and T. Goldstein, *Adaptive ADMM with spectral penalty parameter selection*, CoRR, abs/1605.07246 (2016).

[30] J. Ye, Y. Li, Z. Wu, J. Z. Wang, W. Li, and J. Li, *Determining Gains Acquired from Word Embedding Quantitatively Using Discrete Distribution Clustering*, Association for Computational Linguistics, Vancouver, Canada, July 2017, pp. 1847–1856.

[31] J. Ye, P. Wu, J. Z. Wang, and J. Li, *Fast discrete distribution clustering using Wasserstein barycenter with sparse support*, IEEE Transactions on Signal Processing, 65 (2017), pp. 2317–2332.

Appendix A

List of math symbols

$[n]$	{1,....,n}
$[n]^k$	The k-fold product space $[n] \otimes ... \otimes [n]$
$\mathbb{E}$	Euclidean space
$\mathbb{E}^n$	Euclidean space of dimension n
$\mathbb{R}_+$	The set of non-negative real numbers
$\mathbb{R}^n$	The set of real vectors of dimension n
$(\mathbb{R}^n)^{\otimes k}$	The k-fold product space $\mathbb{R}^n \otimes ... \otimes \mathbb{R}^n$
$(\mathbb{R}^n_+)^{\otimes k}$	The k-fold product space $\mathbb{R}^n_+ \otimes ... \otimes \mathbb{R}^n_+$
$\mathbb{R}^{m \times n}$	The set of real matrices of dimension $m \times n$
$\mathbb{S}^n$	The set of symmetric matrices of dimension $n \times n$
$\mathbb{S}^n_+$	The set of positive semidefinite matrices of dimension $n \times n$
e_n	The standard basis vector with 1 in the n^{th} index
e	The all-ones vector
E_S	The indicator matrix with respect to index S
I_n	The identity matrix in $\mathbb{R}^{n \times n}$
$B_\epsilon(x)$	The open ball of radius ϵ centred at x
Δ_{k+1}	The k-simplex
$\mathbb{S}^n_H$	The hollow space of $\mathbb{S}^n$
$\mathbb{S}^n_C$	The centred space of $\mathbb{S}^n$
$\text{lineseg}(y, z)$	The line segment defined by points y and z
$\inf, \min$	Infimum and minimum
$\sup, \max$	Supremum and maximum
$\liminf$	Limit inferior
$\limsup$	Limit superior
$\text{argmin}(\cdot)$	The set of global minimizers of a function

Appendix B

List of linear maps

- $\text{vec} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n^2} : X \mapsto \begin{bmatrix} X_{11} \\ ... \\ X_{1n} \\ ... \\ X_{n1} \\ ... \\ X_{nn} \end{bmatrix}$.
- $\text{Mat} : \mathbb{R}^{mn} \to \mathbb{R}^{m\times n} : x \mapsto \begin{bmatrix} x_1 & x_{m+1} & ... & x_{m(n-1)} \\ x_2 & x_{m+2} & ... & x_{m(n-1)+1} \\ ... & ... & ... & ... \\ x_m & x_{2m} & ... & x_{mn} \end{bmatrix}$.
- $\text{diag} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n} : X \mapsto \begin{bmatrix} X_{11} \\ X_{22} \\ ... \\ X_{nn} \end{bmatrix}$.
- $\text{Diag} : \mathbb{R}^{n} \to \mathbb{S}^{n} : x \mapsto \begin{bmatrix} x_1 & 0 & ... & 0 \\ 0 & x_2 & ... & ... \\ ... & ... & ... & ... \\ 0 & ... & ... & x_n \end{bmatrix}$.
- $\text{trace} : \mathbb{R}^{n\times n} \to \mathbb{R} : M \mapsto \sum_{i=1}^{n} M_{ii}$.
- $\text{blkdiag} : \mathbb{R}^{m_1\times n_1} \times ... \times \mathbb{R}^{m_t\times n_t} \to \mathbb{R}^{\sum_{i=1}^{t} m_i \times \sum_{i=1}^{t} n_i} : (A_1, ..., A_t) \mapsto \begin{bmatrix} A_1 & 0 & 0 & 0 \\ 0 & A_2 & 0 & 0 \\ ... & ... & ... & ... \\ 0 & 0 & 0 & A_t \end{bmatrix}$.
- $\text{BlkDiag} : \mathbb{R}^{\sum_{i=1}^{t} m_i \times \sum_{i=1}^{t} n_i} \mapsto \mathbb{R}^{m_1\times n_1} \times ... \times \mathbb{R}^{m_t\times n_t} : \begin{bmatrix} A_1 & \times & \times & \times \\ \times & A_2 & \times & \times \\ ... & ... & ... & ... \\ \times & \times & \times & A_t \end{bmatrix} \mapsto (A_1, ..., A_t)$.

- $\text{arrow} : \mathbb{S}^{n+1} \to \mathbb{R}^{n+1} : \begin{bmatrix} s_0 & s^T \\ s & \bar{S} \end{bmatrix} \mapsto \begin{bmatrix} s_0 \\ \text{diag}(\bar{S}) - s \end{bmatrix}$.
- $\text{Arrow} : \mathbb{R}^n \to \mathbb{S}^{n+1} : w \mapsto \begin{bmatrix} 0 & -\frac{w^T}{2} \\ -\frac{w}{2} & \text{Diag}(w) \end{bmatrix}$.
- $\text{Qarrow} : \mathbb{S}^{n+1} \to \mathbb{R}^n : \begin{bmatrix} s_0 & s^T \\ s & \bar{S} \end{bmatrix} \mapsto \text{arrow}(Q_0^T \begin{bmatrix} s_0 & s^T \\ s & \bar{S} \end{bmatrix} Q_0) = \text{diag}(Q\bar{S}Q^T) - Qs$, where $Q_0 := \begin{bmatrix} 1 & 0^T \\ 0 & Q \end{bmatrix}$.
- $\text{QArrow} : \mathbb{R}^n \to \mathbb{S}^{n+1} : w \mapsto Q_0^T \,\text{Arrow}(w) Q_0 = \begin{bmatrix} 0 & -\frac{w^T Q}{2} \\ -\frac{Q^T w}{2} & Q^T \,\text{Diag}(w) Q \end{bmatrix}$.
- Gangster operator with respect to gangster index set $J \in \{0, ..., n\}^2$:

$$\mathcal{G}_J : \mathbb{S}^{n+1} \to \mathbb{S}^{n+1} : Y \mapsto \mathcal{G}_J(Y)_{ij} = \begin{cases} Y_{ij}, (i,j) \in J \text{ or } (j,i) \in J; \\ 0, \text{ o.w.} \end{cases}$$

- Lindenstrauss operator $\mathcal{K} : \mathbb{S}^n \to \mathbb{S}^n : G \mapsto \text{diag}(G)e^T + e\,\text{diag}(G)^T - 2G$.
- $\mathcal{K}^* : \mathbb{S}^n \to \mathbb{S}^n : D \mapsto 2[\text{Diag}(De) - D]$.

Index