MACHINE LEARNING IN SPACE FORMS: EMBEDDINGS, CLASSIFICATION, AND SIMILARITY COMPARISONS

BY

PUOYA TABAGHI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Associate Professor Ivan Dokmanić, Chair
Professor Olgica Milenković
Professor Bruce Hajek
Associate Professor Maxim Raginsky

ABSTRACT

We take a non-Euclidean view at three classical machine learning subjects: low-dimensional embedding, classification, and similarity comparisons.

We first introduce *kinetic Euclidean distance matrices* to solve kinetic distance geometry problems. In distance geometry problems (DGPs), the task is to find a geometric representation, that is, an embedding, for a collection of entities consistent with pairwise distance (metric) or similarity (nonmetric) measurements. In kinetic DGPs, the twist is that the points are dynamic. And our goal is to localize them by exploiting the information about their trajectory class. We show that a semidefinite relaxation can reconstruct trajectories from incomplete, noisy, time-varying distance observations. We then introduce another distance-geometric object: *hyperbolic distance matrices*. Recent works have focused on hyperbolic embedding methods for low-distortion embedding of distance measurements associated with hierarchical data. We derive a semidefinite relaxation to estimate the missing distance measurements and denoise them. Further, we formalize the hyperbolic Procrustes analysis, which uses extraneous information in the form of anchor points, to uniquely identify the embedded points.

Next, we address the design of learning algorithms in mixed-curvature spaces. Learning algorithms in low-dimensional mixed-curvature spaces have been limited to certain non-Euclidean neural networks. Here, we study the problem of learning a linear classifier (a perceptron) in product of Euclidean, spherical, and hyperbolic spaces, i.e., space forms. We introduce a notion of linear separation surfaces in Riemannian manifolds and use a metric that renders distances in different space forms compatible with each other and integrates them into one classifier.

Lastly, we show how similarity comparisons carry information about the underlying space of geometric graphs. We introduce the *ordinal spread* of a distance list and relate it to the *ordinal capacity* of their underlying

space, a notion that quantifies the space's ability to host extreme patterns in nonmetric measurements. Then, we use the distribution of random ordinal spread variables as a practical tool to identify the underlying space form.

To my parents, Mehri and Javad.

ACKNOWLEDGMENTS

I have been extremely fortunate to study at UIUC and have the opportunity to be inspired by many talented people during my PhD studies. From day one, Professor Dokmanić was a positive, energetic, and supportive force as he taught me how to be a researcher. He encouraged me to branch out, utilize experimental techniques, develop theories, and present our findings in the most palatable ways. I have been lucky to work with such a brilliant and caring advisor. I will continue to admire him as a role model throughout my scientific career.

I joined Professor Milenkovic's research group in 2020, shortly after Professor Dokmanić moved to Europe. Professor Milenkovic's effective leadership, breadth of knowledge, creative mindset, and supportive role for students never stop to inspire me. I simply cannot thank her enough for believing in me and my abilities even when, at times, I did not.

I would like to thank the members of my doctoral committee: Professor Raginsky and Professor Hajek. They have given me excellent suggestions regarding the greater scope and the presentation of my work. Finally, special thanks to my friends and colleagues who made these years a great learning experience: Sidharth, Eli, Chao, Konik, and Jianhao.

TABLE OF CONTENTS

LIST OF SYMBOLS

$[N]$	Short for $\{1, \dots, N\}$
$[N]^2_{\text{as}}$	Asymmetric pairs $\{(i,j) : i < j, i, j \in [N]\}$
$x = (x_0, \dots, x_{d-1})^\top$	A vector in $\mathbb{R}^d$
$X = (x_{i,j})_{i \in [d_1], j \in [d_2]}$	A matrix in $\mathbb{R}^{d_1 \times d_2}$
$X \succeq 0$	A positive semidefinite (square) matrix
$\|X\|_F$	Frobenius norm of X
$\|X\|_2$	Operator norm of X
$\|X\|_{1,2}$	The ℓ_2 norm of columns' ℓ_1 norms, $\left\| [\|x_1\|_1, \dots, \|x_{d_2}\|_1]^\top \right\|_2$
$\mathbb{E}_N[x]$	Empirical expectation of a random variable, $N^{-1} \sum_{n=1}^N x_n$
$e_i \in \mathbb{R}^d$	The i-th standard basis vector in $\mathbb{R}^d$
$P_r(X)$	The projection of $X \succeq 0$ onto the span of its top r eigenvectors
1	All-one vector of appropriate dimension
0	All-zero vector of appropriate dimension
$a \vee b$	The maximum value of $\{a, b\}$
$a \wedge b$	The minimum value of $\{a, b\}$
$\langle x, y \rangle$	The dot product of vectors x and y
$[x, y]$	Lorentzian product, i.e., $[x, y] = -x_0 y_0 + \sum_{i=1}^d x_i y_i$
card C	The cardinality of a discrete set C
x_1	Either the first element of vector x or an indexed vector
$\mathbb{O}(d)$	The set of d-dimensional orthonormal matrices

CHAPTER 1

INTRODUCTION

The study of distance geometry problems (DGPs) began with the work of Menger [1], Schoenberg [2], Blumenthal [3], and Young and Householder [4]. DGPs are generally concerned with finding a geometric representation of a set of entities from a set of measured Euclidean distances [5]. Euclidean DGPs have a rich history of applications in robotics [6, 7], wireless sensor networks [8], molecular conformation analysis [9] and dimensionality reduction [10]. A class of approaches to solve DGPs relies on semidefinite characterization of Euclidean Distance Matrices (EDMs) [11, 12] in which the localization problem is reparameterized in terms of the Gram matrix of the point set. This leads to a rank-constrained semidefinite program in which the rank constraint is often relaxed to arrive at a semidefinite relaxation. Solvable DGPs have solution orbits, as opposed to having one unique solution, due to the invariance of distances to rigid motions. In order to obtain a unique solution in applications, we may be given absolute positions of a set of anchor points. We can then use Procrustes analysis to find the best match between the anchors and their corresponding points in the orbit. This is a common technique used in localization problems [5, 7]. In this thesis, we formalize two variations of the classical DGPs: kinetic and hyperbolic DGPs.

Kinetic DGPs are a time-varying version of the classical DGPs in which our goal is to localize *dynamic* point sets from a few snapshots of interpoint distances. These problems find applications in autonomous localization of robot swarms [13], especially in remote situations such as extraterrestrial exploration [14] or deep-water missions [15], and are further related to simultaneous localization and mapping [16, 17].

On the other hand, non-Euclidean spaces have recently been shown to provide significantly improved representations for various data structures [18] and measurement modalities [19, 20]. For instance, *hyperbolic spaces* are suitable for representing hierarchical data associated with trees [21, 22, 23, 24],

human-interpretable images [25], and olfactory data [26]. Further, *spherical spaces* are well-suited for capturing cycle-structures in graphs [27, 28], distance problems on Earth [29], and texture mapping [30]. Euclidean, spherical and hyperbolic geometries are categorical examples of constant curvature spaces, or space forms.

It is thus opportune to match the embedding space to the properties of data at hand. For example, in developmental biology and cancer genomics, single-cell RNA sequencing is used to differentiate cell types and cycles. The classification results have important implications for lineage identification and monitoring cell trajectories and dynamic cellular processes [31]. Klimovskaia *et al.* [32] use hyperbolic rather than Euclidean spaces for low-distortion embedding of complex cell trajectories (hierarchical structures). For embedding hierarchical structures, Ganea *et al.* [33] model order relations as a family of nested geodesically convex cones in a hyperbolic space. Zhou *et al.* [26] show that odors can be efficiently embedded in hyperbolic space provided that the similarity between odors is based on the statistics of their co-occurrences within natural mixtures.

Commonly in hyperbolic embedding applications, there is a tree-like data structure which encodes similarity between a number of entities. In most works that leverage hyperbolic geometry, e.g., hyperbolic multidimensional scaling [34], the embedding technique is not the primary focus and the related computations are often ad hoc. There exist Riemann gradient-based approaches [35, 36, 21, 37] which can be used to directly estimate such embeddings from metric measurements [38]. These methods are iterative and only guaranteed to return a locally optimal solution.

Among important developments in non-Euclidean representation learning are methods for finding "good" mixed-curvature representations for complex heterogeneous datasets [28]. However, despite these recent advances in nontraditional data spaces, almost all accompanying learning approaches have focused on (heuristic) designs of neural networks in constant curvature spaces [39, 40, 41, 42, 43, 44, 45]. The fundamental building block of these neural networks, the perceptron, has received little attention outside the domain of learning in Euclidean spaces. In this thesis we propose a principled design of perceptrons in product space forms with provable convergence guarantees.

Finally, in most practical embedding problems, we seek a representation

for a group of entities based on their pairwise dissimilarities, because the exact magnitudes of the distances may be unavailable. Relevant applications are found in neural coding [46], developmental biology [32], learning from perceptual data [47], and cognitive psychology [48]. Nonmetric embedding problems date back to the works of Shepard [49, 50] and Kruskal [51] in 1970s. Agarwal *et al.* [52] introduce generalized nonmetric multidimensional scaling, which is based on a semidefinite relaxation. Related to nonmetric embedding problems are techniques that study topological properties of graphs independently of the metric and geometric properties such as curvature [53]. An important problem in this domain is to detect intrinsic structure in neural firing patterns, invariant under nonlinear monotone transformations of measurements. Giusti *et al.* [46] propose to use a method based on statistical behavior of Betti curves method based on clique topology of the graph of correlations between pairs of neurons. In this thesis, we consider nonmetric embedding problems in which similarity measurements are sampled from a space form. Then, we propose novel tools to reveal the underlying geometry (i.e., curvature sign) of the embedded entities.

1.1 Overview of Contributions

Chapter 2: Kinetic Euclidean Distance Matrices

Kinetic distance geometry problems: a set of points moves according to an unknown trajectory that belongs to a known class of trajectories. At given time instants we measure a subset of pairwise distances; the subset changes in time and is too small to allow localization at any time alone. We ask: *Can we localize the points and reconstruct trajectories by exploiting the trajectory class?* To tackle kinetic DGPs, we introduce kinetic Euclidean distance matrices — time-dependent distance matrices that incorporate a class of trajectories. We show that polynomial and bandlimitted trajectories can be reconstructed from incomplete, noisy temporal distance measurements. Our proposed solution is based on semidefinite relaxation and gives us distance trajectories. To convert them to point trajectories, we utilize known and new results on spectral factorization of polynomial matrices.

Chapter 3: Hyperbolic Distance Matrices

We study DGPs in hyperbolic spaces: we aim to find a realization for a set of entities in a hyperbolic space given their incomplete and noisy pairwise non-Euclidean distances (e.g., distances on a weighted tree). Analogous to the Euclidean DGPs, we introduce hyperbolic distance matrices (HDMs). Then, we propose a semidefinite characterization of HDMs by studying the properties of hyperbolic Gram matrices—matrices of pairwise Lorentzian inner products for the point set. Together with a spectral factorization method to directly estimate the hyperbolic points, our proposed semidefinite characterization gives rise to flexible embedding algorithms that can handle diverse constraints and mixed metric and nonmetric data.

Chapter 4: Hyperbolic Procrustes Analysis

In order to uniquely identify the correct point locations, from the orbit of possible solutions generated by distance-preserving bijections, we may be given the exact position of a subset of points, called *anchors*. The Procrustes analysis picks the correct solution by finding the best match between the anchors with their corresponding points in the solution orbit. We formalize and use hyperbolic Procrustes analysis to find a joint estimate for hyperbolic translation and rotation maps that best aligns two sets of points. This joint estimation problem is then decoupled in the following steps: (1) translate the center mass of each point set to the coordinate origin, and (2) estimate the unknown rotation factor. We prove that this approach gives the theoretically optimal isometry if the point sets match perfectly.

Chapter 5: Linear Classification in Product Space Forms

We address the problem of linear classification in product space forms, i.e., product of Euclidean, hyperbolic and spherical spaces. An important property of such spaces is that they are endowed with logarithmic and exponential maps which help us establish rigorous performance results. We describe the "point-line" formulation for linear classifiers in d-dimensional space forms. Then, we prove that linear classifiers in d-dimensional space forms of any curvature can shatter exactly $d + 1$ points regardless of the curvature of

the underlying space form. The key idea behind our analysis is to define separation surfaces in space forms directly through the use of geodesics on Riemannian manifolds. We then introduce metrics that make distances in different space forms compatible with each other and integrate them in linear classifiers (in a product space form) with *majority* signed distance criteria. We propose the corresponding perceptron and SVM classifiers and establish convergence results for the former.

Chapter 6: Geometry of Similarity Comparisons

We argue that nonmetric information such as *distance comparisons* carries valuable information about the space the data points originated from. We introduce the notion of *ordinal spread* of a distance list which describes a pattern in which entities appear in the list. This notion is related to the *ordinal capacity* of their underlying space. The ordinal capacity of a metric space quantifies the space's ability to host extreme patterns of ordinal spreads (computed from similarity measurements). We show that the ordinal capacity of Euclidean and spherical spaces are equal and grow exponentially with their dimensions, while the ordinal capacity of a hyperbolic space is infinite — regardless of its dimension. We also associate an *ordinal spread random variable* with sets of random points in a space form and show how the distribution of this random variable serves as a practical tool to identify the underlying space form given nonmetric measurements. In numerical experiments, we correctly uncover the hyperbolicity of weighted trees, detect Euclidean and spherical geometries for ordinal measurements derived from local and global cartographic data, and uncover heterogeneous cell populations from noisy single-cell RNA sequencing data.

Appendices A to E

We delegate the proofs of theorems, propositions, and lemmas, additional numerical experiments, further discussions and analysis to their corresponding appendix sections.

CHAPTER 2

KINETIC EUCLIDEAN DISTANCE MATRICES

2.1 Introduction[1]

The famous distance geometry problem (DGP) [54] asks to reconstruct the geometry of a point set from a subset of interpoint distances. It models a wide gamut of practical problems, from sensor network localization [55, 56, 57] and microphone positioning [58, 59, 60, 61] to clock synchronization [62, 63], to molecular geometry reconstruction from nuclear magnetic resonance data [64, 65]. Among the most successful vehicles for the design of DGP algorithms are the Euclidean distance matrices (EDM) [5].

EDMs model static objects. When things move, they characterize a snapshot of the interpoint distances and the point set geometry. It seems intuitive that with a good model for the trajectories, we should be able to leverage the motion and improve trajectory estimation.

In this chapter, we review distance matrices for moving points, which we call Kinetic EDMs (KEDMs) inspired by the notion of kinetic data structures [66] for moving points. KEDMs are a generalization of EDMs whose entries now become functions of time. We show how by using KEDMs we can neatly address the kinetic distance geometry problem (KDGP), a natural generalization of the classical, static distance geometry problem (DGP) defined in Section 2.2. Unlike with the static DGP, in order to make the kinetic version well posed, we must constrain the point trajectories to belong to a class of functions, for example polynomial trajectories of a bounded degree. Informally, we ask the following question: *Suppose a set of points move according to a known trajectory model. At given time instants we measure*

[1] © 2019 IEEE. Reprinted, with permission, from P. Tabaghi, I. Dokmanic, and M. Vetterli, *Kinetic Euclidean Distance Matrices*, IEEE Transactions on Signal Processing (Volume: 68), December 2019. The published manuscript is available at `https://doi.org/10.1109/TSP.2019.2959260`

a subset of pairwise distances; the subset can change between measurements and it may be too small to allow localization at any time alone. Can we systematically localize the points and reconstruct trajectories by exploiting the trajectory model?

Localization of dynamic point sets from distance measurements finds applications whenever objects move. Robot swarms, for example, often must localize autonomously [13], especially in remote situations such as extraterrestrial exploration [14] or deep-water missions [15]. Related applications exist in environmental monitoring, for example for dynamic sensor networks composed of river-borne sensing nodes [67]. An important application of localization of moving objects is in global positioning with satellites where both the satellites and the users move. Applications are emerging where sensing is opportunistic and the positions of reference objects are not known [68]; in Section 2.6.4, we present a simulated example of global positioning with unknown satellite trajectories. This problem is further related to simultaneous localization and mapping (SLAM) [16, 17]. Kinetic distance geometry problems are common in computer vision. Examples are action recognition from dynamic interjoint distance skeleton data [69] and more generally data structures for describing kinetic point sets [66]. Applications in multi-robot coordination, crowd simulations, and motion retargeting are explored in [70, 71], where the authors introduce the *dynamical distance geometry problem* (dynDGP).[2] Even in applications to proteins and molecules, the atoms move (for example, proteins fold) in specific ways [72].

The study of distance geometry and EDMs began with the work of Menger [1], Schoenberg [2], Blumenthal [3], and Young and Householder [4]. Gower derived numerous results on EDMs [73, 74] including a complete rank characterization [74]. An extensive treatise on EDMs with many original results and an elegant characterization of the EDM cone was written by Dattorro [75]; a tutorial-style introduction to EDMs is presented in [5].

A large class of approaches to point set localization from distance measurements relies on semidefinite programming [11, 12]. Namely, the localization problem is written in terms of the Gram matrix of the point set which leads to a rank-constrained semidefinite program. The rank constraint is often relaxed to arrive at a semidefinite relaxation which is a convex optimization

[2]Though related, the dynDGP is rather different from our KDGP.

problem and can be solved using standard tools.

We take inspiration from these approaches and show how trajectory localization can also be formulated as a semidefinite program, thus answering the above question in the affirmative. Concretely, we show that the parameters of chosen trajectory models can be recovered by a semidefinite program and a tailor-made alignment procedure akin to Procrustes analysis. The latter can be interpreted as spectral factorization of semidefinite polynomial matrices with side information, and our developments rely on the related spectral factorization results [76].

2.1.1 Contributions and Outline

In Section 2.2, we extend the definition of the distance geometry problem (DGP) to its kinetic version and review the essential facts about Euclidean distance matrices and associated semidefinite programs. In Section 2.3, we introduce Kinetic Euclidean Distance Matrices (KEDMs)—a new kind of time-dependent distance matrices that incorporate motion. The entries of KEDMs become functions of time, the squared time-varying distances. Then, we study two smooth trajectory models—polynomial and bandlimited trajectories. In Section 2.4, we present our main contribution which is a semidefinite relaxation, inspired by similar strategies for static EDMs. We show that polynomial and bandlimitted trajectories can be reconstructed from incomplete, noisy distance observations, scattered over multiple time instants. The solution to the SDP, however, only gives us *distance* trajectories. To convert them to point trajectories, we need known and new results on spectral factorization of polynomial matrices developed in Section 2.5. Finally, we show through computational experiments that our semidefinite relaxation can indeed reconstruct model trajectories from sparse and noisy measurements. We can also reduce the number of measurements *per time instant* well below that minimally required for localization in the static case. The proofs of all propositions are delegated to Appendix A.[3]

[3]Documented code and data to reproduce all experiments is available online at `https://github.com/swing-research/kedm/`

2.2 Static and Kinetic Distance Geometry Problems

We begin by introducing the classical distance geometry problem (DGP) and then formulate its generalization to moving points. We also discuss an EDM-based approach to the DGP with noisy and incomplete distances.

The DGP can be informally stated as follows: find the d-dimensional locations $\left\{x_n \in \mathbb{R}^d\right\}_{n=1}^N$ of a set of points, given a subset of possibly noisy pairwise distances $\{d_{mn} : 1 \leq m < n \leq N\}$. We will work only with Euclidean distances so that $d_{mn} = \|x_m - x_n\|$.

An elegant formalization can be made in graph-theoretic terms. Consider a graph $G = (V, E)$ whose vertex set V corresponds to the points $\{x_n\}_{n=1}^N$. The edge set E tells us which distances are measured and which are not. Given two vertices $u, v \in V$ and the corresponding undirected edge $e = \{u, v\}$, we have $e \in E$ if and only if the distance between u and v is known. Let $f : E \to \mathbb{R}^+$ be the weight function that assigns those known, measured distances to edges. Then we can pose the following problem [54].

Problem 1 (Distance Geometry Problem)**.** *Given an integer $d > 0$ (the ambient dimension) and an undirected graph $G = (V, E)$ whose edges are weighted by a non-negative function $f : E \to \mathbb{R}^+$ (distance), determine whether there is a function $x : V \to \mathbb{R}^d$ such that*

$$\|x(u) - x(v)\| = f(\{u, v\}) \text{ for all } \{u, v\} \in E.$$

The function x which assigns coordinates to vertices is called an embedding or a realization of the graph in $\mathbb{R}^d$. Of course, in practice the measurements are corrupted by measurement errors, and the goal is to minimize some notion of discrepancy between the measured distances and the distances induced by our estimate; for example:

$$\underset{x:V\to\mathbb{R}^d}{\text{minimize}} \sum_{\{u,v\}\in E} \left(\|x(u) - x(v)\| - f(\{u, v\})\right)^2. \tag{2.1}$$

Section 2.2.1 explains how to use EDMs to proceed in this case. Figure 2.1 illustrates the DGP with an intermediate step of constructing an EDM. The EDM can be interpreted as a weighted adjacency matrix in which weights are squared distances.

In this chapter, we want to formalize distance geometry problems when the

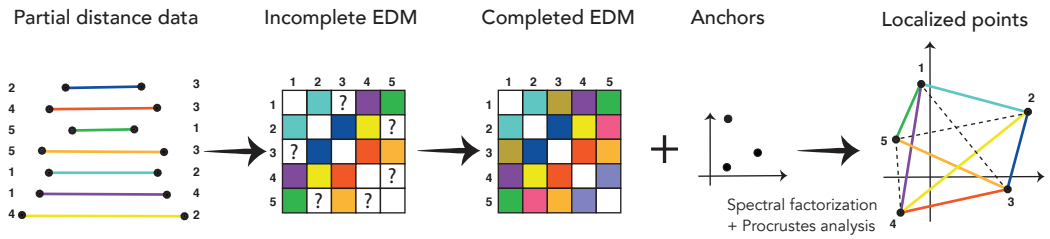


Figure 2.1: The objective of DGP is to find an embedding for a given partial pairwise distance data. This can be done in two steps: (a) Completing EDM associated with the measurements, i.e. estimating the missing measurements and (b) Estimating an embedding and using anchor points to resolve the rigid transformation ambiguity, discussed in Section 2.2.1.

points move and the set of measured distance changes over time. Instead of localizing the points only at the measurement times, our goal is to estimate entire trajectories for all times. To make this problem well posed we must introduce a class of admissible continuous trajectories $\mathcal{X}$. Then, we can formulate the following kinetic version of Problem 1.

Problem 2 (Kinetic Distance Geometry Problem)**.** *Given an embedding dimension $d > 0$, a set of T sampling times $\mathcal{T} = \{t_1, \dots, t_T\} \subset \mathbb{R}$, and a sequence of undirected graphs $G_i = (V, E_i)$ whose edges are weighted by non-negative functions $f_i : E_i \to \mathbb{R}^+$, for $i \in \{1, \dots, T\}$, determine whether there is a function $x : V \times \mathbb{R} \to \mathbb{R}^d$, where $x \in \mathcal{X}$ and for all $t_i \in \mathcal{T}$ we have:*

$$\|x(u, t_i) - x(v, t_i)\| = f_i(\{u, v\}) \text{ for all } \{u, v\} \in E_i,$$

where $\mathcal{X}$ is the set of admissible trajectories.

Figure 2.2 illustrates the KDGP for four trajectories. One way to interpret KDGP is as a sequence of static DGPs with additional information about sampling times and the trajectory model. Indeed, the KDGP can be seen as a nonlinear spatio–temporal sampling problem, with the nonlinear samples (distances) spread in space in time. A natural question is whether we can compensate for a reduction in the number of spatial samples by oversampling in time. We answer this question in affirmative in Section 2.6.

The first step is to estimate the continuous KEDM from samples distributed in space and time; this is discussed in Section 2.4. The second step is to use

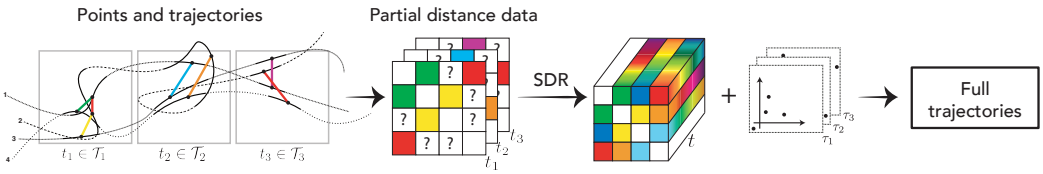


Figure 2.2: KDGP: Estimate an embedded trajectory for a given sequence of partial pairwise distances at different times, $t_1, \cdots, t_T$. We estimate the corresponding KEDM with a semidefinite relaxation Algorithm 1, and then use anchors to estimate the trajectories.

information about the absolute positions of a set of anchor points in order to assign absolute locations to trajectories; this is discussed in Section 2.5.

2.2.1 Solving the Distance Geometry Problem by EDMs

It is useful to recall the EDM-based approach to the DGP. Ascribe the coordinates of N points in a d-dimensional space to the columns of matrix $X \in \mathbb{R}^{d\times N}$, $X = [x_1,\ x_2,\ \cdots,\ x_N]$. The squared distance between x_i and x_j is

$$d_{ij}^2 = \|x_i - x_j\|^2 = \|x_i\|^2 - 2x_i^\top x_j + \|x_j\|^2,$$

from which we can read out the equation for the EDM $D = (d_{ij}^2)$ as

$$D = \mathcal{K}(G) \stackrel{\text{def}}{=} \operatorname{diag}(G)1^\top - 2G + 1\operatorname{diag}(G)^\top, \tag{2.2}$$

where 1 denotes the column vector of all ones, $G = X^\top X$ is the Gram matrix and $\operatorname{diag}(G)$ is a column vector of the diagonal entries of G. We see that the EDM of a point set is a linear function of its Gram matrix. Reformulating the problem in terms of the Gram matrix is beneficial because it will lead to a semidefinite program. If we can find the Gram matrix, the point set can be obtained by an eigenvalue decomposition.

To see how, let $G = U\Lambda U^\top$, where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_N)$ with all eigenvalues λ_i non-negative, and U orthonormal, as G is a symmetric positive semidefinite matrix. Assume that the eigenvalues are sorted in decreasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$. Then we can estimate the point set as $\widehat{X} \stackrel{\text{def}}{=} \left[\operatorname{diag}\left(\sqrt{\lambda_1},\ \ldots, \sqrt{\lambda_d}\right),\ 0_{d\times(N-d)}\right]U^\top$. Since the EDM only specifies the points up to a rigid transform, $\widehat{X}$ will be a rotated, reflected and translated version of X.

One way to estimate D from noisy, incomplete distance data is by semidefinite programming. This hinges on the one-to-one equivalence between EDMs with embedding dimension d and centered Gram matrices of rank d. Define the geometric centering matrix J as

$$J_N \stackrel{\text{def}}{=} I - \frac{1}{N} 1 1^\top.$$

Then $\mathcal{K}(G)$ is an invertible map on the set of Gram matrices which correspond to centered point sets (implying $G1 = 0$) with the inverse given by

$$-\frac{1}{2} J_N \mathcal{K}(G) J_N = G.$$

In particular, we have the following equivalence that holds for matrices D with a zero diagonal:

$$\left.\begin{array}{l} D = \mathcal{D}(X) \\ \text{affdim}(X) \leq d \end{array}\right\} \iff \begin{cases} -\frac{1}{2} J_N D J_N \succeq 0 \\ \text{rank}(J_N D J_N) \leq d, \end{cases}$$

where $\mathcal{D}(X) = \mathcal{K}(X^\top X)$ and affdim denotes the dimension of the smallest affine space that can hold X. In other words, instead of directly searching for the points X given distance data, we can search for the suitable Gram matrix.

Let $\widetilde{D}$ be the noisy, incomplete EDM from which we want to estimate the point locations, with unknown entries replaced by zeroes. Define the mask matrix $W = (w_{ij})$ as

$$w_{ij} \stackrel{\text{def}}{=} \begin{cases} 1, & (i,j) \in E \\ 0, & \text{otherwise.} \end{cases}$$

This mask matrix will let us compute the loss only on those entries that were actually measured. Note that W is precisely the adjacency matrix of the undirected graph from Problem 1.

Then the above discussion is summarized in the following rank-constrained

semidefinite program:

$$\begin{aligned} \underset{G}{\text{minimize}} \quad & \| \widetilde{D} - W \circ \mathcal{K}(G) \|_F^2 \\ \text{subject to} \quad & G \succeq 0 \\ & G1 = 0 \\ & \operatorname{rank}(G) \leq d. \end{aligned} \tag{2.3}$$

Since the Gram matrix (Gramian) is linearly related to the EDM, the objective function is convex. However, the rank constraint, $\operatorname{rank}(G) \leq d$, makes the feasible set in (2.3) non-convex. Note that (2.1) is also a non-convex program.

The value of reformulation in terms of G is that it admits a simple convexification strategy—a semidefinite relaxation—which was repeatedly shown to perform well in the context of distance geometry [5, 11]. An intuitive explanation is that while the rank condition ensures the correct embedding dimension, the constraint that G be positive semidefinite (in other words, that it be a Gramian) enforces a number of geometric constraints. For instance, it ensures that the entries of the EDM verify triangle inequalities, as well as other subtle properties (see, for example, the Cayley-Menger conditions [77]).

We should add that a semidefinite relaxation is by no means the only way to convexify (2.1) or (2.3). The mathematical optimization literature knows a number of others, many of which could also be applied in the X-domain (2.1). One may, for example, replace nonlinear terms in (2.3) by suitable convex over- and under-estimators. A well-known example is McCormick's convexification for bilinear terms [78], and similar strategies for quadratic and higher-order terms [79]. In this work we limit ourselves to semidefinite relaxation.

Once the rank constraint is discarded, the embedding dimension of the reconstructed point set is dictated by the measurements. One often looks for a solution with the lowest possible embedding dimension via various rank-minimization heuristics [5]. In general, especially with noisy measurements, our best expectation is that the reconstructed points will lie close to a linear variety of the desired dimension.[4] To ensure the right embedding dimension, a suboptimal solution can be computed by replacing the estimated Gramian

[4] An empirical study of the number of required measurements is available in [5].

with its best rank-d approximation; see Section 2.5 for the kinetic case.

The constraint $G1 = 0$ serves to set the centroid of the recovered point set at the origin of the coordinate system as it implies $X1 = 0$. This resolves the translational invariance of the problem. The remaining rotational (and reflection) invariance must be resolved once the points are estimated from the Gramian. The Gramian itself is invariant to the rotations of the point set since $G = X^\top X = (UX)^\top UX$ for any orthonormal matrix $U \in \mathbb{O}(d)$.

2.2.2 Orthogonal Procrustes Problem

As mentioned before, the EDM only specifies the point set up to a rigid transformation (rotation, translation, and reflection). If the task requires determining absolute positions of points, the standard method is to designate a subset of points as *anchors* whose positions are known, and use anchors to align the reconstructed point set.

Let $X_a \in \mathbb{R}^{d\times N_a}$ be the submatrix (a selection of columns) of X that should be aligned with the anchors listed as columns of $Y \in \mathbb{R}^{d\times N_a}$. We note that the number of anchors—columns in X_a—is typically small compared with the total number of points—columns in X.

We first center the columns of Y and X_a by subtracting the corresponding column means $y_c = Y J_{N_a}$ and $x_{a,c} = X_a J_{N_a}$, obtaining matrices $\overline{Y}$ and $\overline{X}_a$. Next, we perform the orthogonal Procrustes analysis—we search for the rotation and reflection that best maps $\overline{X}_a$ onto $\overline{Y}$:

$$R = \underset{Q:QQ^\top=I}{\arg\min} \left\| Q\overline{X}_a - \overline{Y} \right\|_F^2 . \tag{2.4}$$

The solution to (2.4) is given by the singular value decomposition (SVD) [80] as follows. Let $U\Sigma V^\top$ be the SVD of $\overline{X}_a \overline{Y}^\top$; then $R = VU^\top$. The best alignment is applied to the reconstructed point set as

$$X_{\text{aligned}} = R(X - x_{a,c}1^\top) + y_c 1^\top .$$

2.3 Trajectory Models and Basis Gramians

In order to extend the EDM-based tools to the KDGP, we must define the class of trajectories $\mathcal{X}$. We introduce two trajectory models—polynomial and bandlimited—and show how they can be parameterized in terms of the so-called basis Gramians.

The chosen trajectory models are standard; they model many interesting trajectories. The polynomial model is common in simultaneous localization and mapping as well as tracking, where it appears as constant velocity or constant acceleration model [81, 82]. The bandlimited model describes periodic trajectories of varying degrees of smoothness which are locally well-approximated by polynomials.

We use similar notation as in the static case. Let $X(t) = [x_1(t), \ldots, x_N(t)]$ be the trajectory matrix of N points in $\mathbb{R}^d$ where $x_n(t)$ is the position of n-th point at time t. We define the KEDM in a natural way.

Definition 1 (KEDM)**.** *Given a set of trajectories $X(t) \in \mathbb{R}^{d\times N}$, the corresponding KEDM is the time-dependent matrix $D(t) \in \mathbb{R}^{N\times N}[t]$ of time-varying squared distances between the points:*

$$D(t) \overset{def}{=} \mathcal{D}(X(t)).$$

2.3.1 Polynomial Trajectories

For a set of N points in $\mathbb{R}^d$, we define the set of polynomial trajectories of degree P as

$$\mathcal{X}_{\text{poly}} = \left\{ \sum_{p=0}^{P} t^p A_p \;\middle|\; A_p \in \mathbb{R}^{d\times N},\ p \in \{0, \ldots, P\} \right\}. \tag{2.5}$$

For $X(t) \in \mathcal{X}_{\text{poly}}$, the Gramian at time t can be written as

$$G(t) = \sum_{k=0}^{K} B_k t^k, \tag{2.6}$$

where $B_k = \sum_{i=\max\{0,k-p\}}^{\min\{k,p\}} A_i^\top A_{k-i}$ and $K = 2P$.[5] Similar to the static case, our goal is to cast the trajectory retrieval problem as a semidefinite program; we do so via the time-dependent Gramian in Section 2.4.

The key step is to parameterize the problem entirely in terms of (constant) positive semidefinite matrices, instead of the parameterization in terms of A_p or B_k. To do so, we fix $K+1$ time instants $\tau_0, \ldots, \tau_K$ and define $G_k \stackrel{\text{def}}{=} G(\tau_k)$. The matrices G_k should be interpreted as *elementary*, or *basis* Gramians in the sense that the Gramian $G(t)$ can be written as a linear combination of $G_0, \ldots, G_K$ as elaborated in the following proposition.

Proposition 1. *Consider the polynomial trajectory in* (2.5). *Let* $G_k,\ k \in \{0, 1, \ldots, K\},\ K = 2P$ *be given as above with* τ_k *all distinct. Then we have*

$$G(t) = \sum_{k=0}^{K} w_k(t) G_k,$$

with the weights $w(t) = [w_0(t), \cdots, w_K(t)]^\top$ *given as*

$$w(t) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ \tau_0^K & \tau_1^K & \cdots & \tau_K^K \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ t \\ \vdots \\ t^K \end{pmatrix}.$$

This result is a matrix version of Lagrange interpolation. Since entries of $G(t)$ are polynomials of degree $2P$ in t, they are completely determined by their values at $2P+1$ points. However, in this Gram matrix version it gives us something rather useful: a way to write a positive semidefinite $G(t)$ as a linear combination of positive semidefinite G_k, which lends itself nicely to convex optimization. We note that the question of how to choose the sampling times τ_k is beyond the scope of this article, though we give empirical results in Section 2.6.

2.3.2 Bandlimited Trajectories

Our second model is the set of periodic bandlimited trajectories. For a set of N points in $\mathbb{R}^d$, the set of periodic bandlimited trajectory of degree P can be

[5] Simplified from $G(t) = \big(\sum_{p=0}^{P} t^p A_p\big)^\top \big(\sum_{p=0}^{P} t^p A_p\big)$.

written as

$$\mathcal{X}_{\mathrm{BL}} = \Bigg\{ B_0 + \sum_{p=1}^{P} \{A_p \sin(p\omega t) + B_p \cos(p\omega t)\} \;\Bigg|\; A_p, B_0, B_p \in \mathbb{R}^{d\times N}, p \in \{1,\dots,P\}, \omega \in \mathbb{R}^{+} \Bigg\}. \tag{2.7}$$

Similar to the polynomial case, we represent the Gramian $G(t)$ as a linear combination of some Gramian basis.

Proposition 2. *Consider the bandlimited trajectory in* (2.7). *Let* $G_k,\ k \in \{0,1,\dots,K\},\ K = 4P$ *be given as above with* τ_k *all distinct (modulo* $\frac{2\pi}{\omega}$*). We have*

$$G(t) = \sum_{k=0}^{K} w_k(t) G_k,$$

with the weights $w(t) = [w_0(t), \cdots, w_K(t)]^\top$ *given as*

$$w(t) = \begin{pmatrix} 1 & \cdots & 1 \\ \sin(\omega\tau_0) & \cdots & \sin(\omega\tau_K) \\ \cos(\omega\tau_0) & \cdots & \cos(\omega\tau_K) \\ \vdots & \ddots & \vdots \\ \cos(2P\omega\tau_0) & \cdots & \cos(2P\omega\tau_K) \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ \sin(\omega t) \\ \cos(\omega t) \\ \vdots \\ \sin(2P\omega t) \\ \cos(2P\omega t) \end{pmatrix}.$$

We have thus developed a way to write a time-dependent Gramian in terms of a linear combination of positive semidefinite (constant) basis Gramians.

2.3.3 Ambiguities Beyond Rigid Transformations in KDGP

Same as the static DGP, the KDGP suffers from rigid transformation ambiguity. Namely, the rotated and translated trajectory sets cannot be distinguished from pairwise distance data. However, since at every time instant we can apply a different rigid transform, the set of ambiguities that arise in the KDGP is much larger than just the rigid transforms.

In particular, trajectory sets which look rather differently (nothing like rotations and translations of each other) could generate the same KEDM.

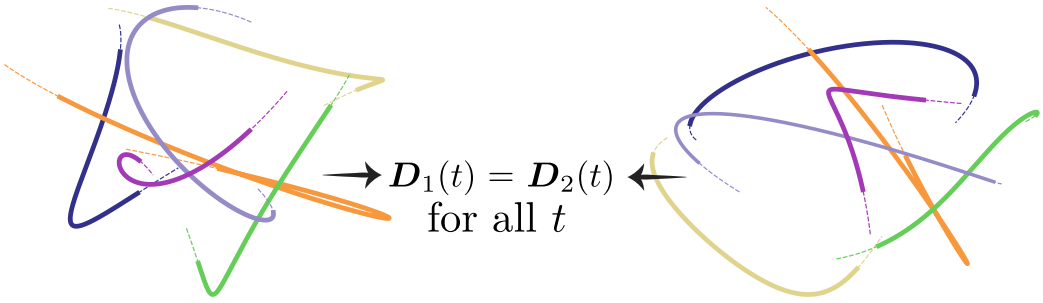


Figure 2.3: Two trajectory sets which are not rigid transforms of each other, but which generate the same KEDM. Corresponding points have the same color.

We give an example in Figure 2.3. The following straightforward result characterizes trajectories that lead to the same KEDM.

Definition 2. *We say that the two trajectories $X(t)$, $Y(t) \in \mathbb{R}^d[t]$ defined over some time interval T are distance-equivalent and write $X \overset{\mathcal{D}}{\sim} Y$ if and only if $\mathcal{D}(X(t)) = \mathcal{D}(Y(t))$ for all $t \in T$.*

Proposition 3. *Let $X(t)$, $Y(t)$ be arbitrary trajectories in $\mathbb{R}^d[t]$. Then, the following statements are equivalent:*

1. *$X \overset{\mathcal{D}}{\sim} Y$.*
2. *$Y(t) = U(t)X(t) + c(t)1^\top$ where $U(t)^\top U(t) = I$ and $c(t)$ is a d-dimensional time-varying vector.*

Requiring that the trajectories follow a particular model (for example polynomial or bandlimited) limits possible choices of the time-varying rigid transform parameters $U(t)$ and $c(t)$. In particular, known results on spectral factorization of polynomial matrices imply that the orthogonal $U(t)$ must be a constant matrix [83, 84]. On the other hand, as long as $c(t)$ is polynomial (or bandlimited) of the same degree as $X(t)$, it is a legal choice in the sense that the trajectories remain polynomial or bandlimited. But even with a fixed $U(t) = U$, varying $c(t)$ can produce trajectories of rather different shapes which are indistinguishable from their time-varying distances.

In Section 2.5, we propose a method for spectral factorization of kinetic Gramians based on anchor points and show how it resolves the described ambiguities. In our algorithms we will choose $c(t)$ so that the centroid of the

point set is kept fixed at the origin at all times, and then recover the correct centroid using anchor points. The following proposition will be useful.

Proposition 4. *For trajectories of the form* (2.5) *or* (2.7)*, the following statements are equivalent:*

1. *All coefficient matrices* $\{A_p\}_{p=0}^P$ *have zero-mean columns.*
2. $X(t)1 = 0,\ \forall t \in \mathbb{R}$.
3. $G_k 1 = 0,\ \forall k \in \{0, \cdots, K\}$.

2.4 Computing the KEDM from Noisy, Incomplete Data by Semidefinite Programming

In this section we use the basis Gramian representation to derive an algorithm that solves the KDGP. Just as in the static case, we can either search directly for the set of trajectories $X(t)$ which reproduces the measured distances, or we can search for the time-varying Gramian $G(t)$ and use spectral factorization to estimate $X(t)$. In the static case, the two formulations are equivalent (they produce the same solution up to a rigid transform), but the formulation in terms of the Gram matrix led to a convenient semidefinite relaxation. In the time-varying case, we again state both formulations, and argue that the difference is now more significant.

To treat polynomial and bandlimited trajectories at once, we define the symbol Θ to mean either $\Theta = \{A_p\}_{p=0}^P$ for the polynomial model or $\Theta = \{A_p, B_p\}_{p=1}^P \cup \{B_0\}$ for the bandlimited model, and similarly let $X_\Theta(t) = \sum_{p=0}^P A_p t^p$ (resp. $X_\Theta(t) = B_0 + \sum_{p=1}^P A_p \cos(p\omega t) + B_p \sin(p\omega t)$).

Formalizing in X domain In this case, trajectory retrieval is written as

$$\begin{aligned} \underset{\Theta \in \mathcal{A}}{\text{minimize}} \quad & \sum_{i=1}^{T} \alpha_i \left\| \widetilde{D}_{t_i} - W_i \circ \mathcal{D}\left(X_\Theta(t_i)\right) \right\|_F^2 \\ \text{subject to} \quad & X_\Theta(t)1 = 0, \forall\, t \in \mathbb{R}, \end{aligned} \tag{2.8}$$

where $\mathcal{D}(X) = \mathcal{K}(X^\top X)$, $\widetilde{D}_{t_i}$ is the matrix of partial measured distances at time t_i, W_i is the adjacency matrix corresponding to measurements, $\alpha_i \geq 0$ are non-negative weights, and $\mathcal{A}$ is the set of all feasible parameters. It is not hard to see that the objective in (2.8) is nonconvex in Θ (even though the constraint set is convex by Proposition 4). Hence, this problem involves minimizing a nonconvex functional which is in general difficult.

Formalizing in G Domain Next, we derive a semidefinite program inspired by (2.3) for the trajectory recovery problem. The key ingredient is the basis Gramian representation of $G(t)$ from Section 2.3. Since the actual kinetic Gramian is linear in basis Gramians, the overall objective will be convex as long as the data fidelity metric is convex. The latter holds true since we use the squared Frobenius norm:

$$\begin{aligned} \underset{(G_k : G_k \succeq 0)_{k=0}^K}{\text{minimize}} \quad & \sum_{i=1}^{T} \alpha_i \left\| \widetilde{D}_{t_i} - W_i \circ \mathcal{K}\left(\sum_{k=0}^{K} w_k(t_i) G_k \right) \right\|_F^2 \\ \text{subject to} \quad & G(t)1 = 0, \forall t \in \mathbb{R} \\ & G(t) \succeq 0, \forall t \in \mathbb{R} \\ & \max_{t \in \mathbb{R}} \ \operatorname{rank} G(t) = d. \end{aligned} \tag{2.9}$$

The constraints ensure that the solution corresponds to a time-varying Gramian $G(t)$ with correct rank.

Recall that any trajectory generates a Gramian with the form $G(t) = \sum_{k=0}^{K} w_k(t) G_k$. Hence, it is clear that the set

$$\mathcal{G} = \left\{ (G_k : G_k \succeq 0)_{k=0}^K : \ \sum_{k=0}^{K} w_k(t) G_k \succeq 0 \text{ for all } t \right\}$$

is non-empty. Further, $\mathcal{G}$ is convex as an (infinite) intersection of convex sets,

$$\mathcal{G} = \bigcap_{t \in \mathbb{R}} \{ (G_k : G_k \succeq 0)_{k=0}^K : \sum_{k=0}^{K} w_k(t) G_k \succeq 0 \}.$$

Let us emphasize that even though the objective is convex, the problem (2.9) is not easy to solve: it is still non-convex (due to the rank constraint) and in fact uncountably infinite (due to the continuous-time constraints).

There is no rotation ambiguity associated with this formulation because

the Gramian is invariant to rotation and reflection of the points. Translation ambiguity has been resolved by requiring that $G(t)1 = 0$ which implies that the recovered point set shall be centered at all times.

2.4.1 Equivalence Between (2.8) and (2.9)

The two formulations are equivalent if for every possible set of measurements, the solution sets produce the same KEDM. Denoting the optimizers (which could be sets) by Θ^* and $(G_k^*)_{k=0}^K$, it should hold that

$$\mathcal{D}\left(X_{\Theta^*}(t)\right) = \mathcal{K}\left(\sum_{k=0}^{K} w_k(t) G_k^*\right), \ t \in \mathbb{R}.$$

By Propositions 1 and 2, for any optimizer Θ^* of (2.8) and the corresponding trajectory $X_{\Theta^*}(t)$, we can find a Gramian basis $(\widetilde{G}_k)_{k=0}^K$ such that $\mathcal{D}\left(X_{\Theta^*}(t)\right) = \mathcal{K}\left(\sum_{k=0}^K w_k(t)\widetilde{G}_k\right)$. Therefore, we have

$$J_1(\Theta^*) = J_2((\widetilde{G}_k)_{k=0}^K) \geq J_2((G_k^*)_{k=0}^K),$$

where J_1 denotes the loss in (2.8), and J_2 denotes the loss in (2.9). The question is whether this inequality can be made strict. Could the solution to (2.9) lead to a Gramian $G(t)$ with no corresponding trajectory in $\mathcal{A}$? An in-depth study of this question is beyond the scope of this work, but to see that this is indeed possible consider a contrived case of no measurements at all, that is to say, a feasibility search.

Trivially, any $\Theta \in \mathcal{A}$ is a solution to (2.8) and any set of basis Gramians $(G_k : G_k \succeq 0)_{k=0}^K$ is a solution to (2.9). By Lemma 1 every Gramian $G(t)$ produced by its basis $(G_k)_{k=0}^K$ has a polynomial spectral factor, that is, it corresponds to a polynomial trajectory $X(t)$ such that $G(t) = X(t)^\top X(t)$. Even though $G(t)$ is real, its spectral factor, however, need not be; see [76] for a characterization of rank-deficient polynomial Gramians $G(t)$ without real spectral factors. This situation is fundamentally different from what we had in the static case. Hence, we can construct feasible "complex trajectories" which are outside of $\mathcal{A}$. Consequently, the constraints in (2.8) are necessary but not sufficient for the two formulations to be equivalent. Nonetheless, they become equivalent with sufficient measurements.

Proposition 5. *Suppose that* $\widetilde{D}_i = W_i \circ \mathcal{D}(X_\Theta(t_i))$ *and* (2.9) *has a unique optimizer* $G^*(t) = \sum_{k=0}^{K} w_k(t) G_k^*$. *Then*

$$G^*(t) = X_{\Theta^*}(t)^\top X_{\Theta^*}(t), \tag{2.10}$$

where $X_{\Theta^*}(t) = U X_\Theta(t) J_N$ *for some orthogonal matrix* $U \in \mathbb{R}^{d\times d}$ *(that is, it is a centered, rotated version of the true geometry).*

It is useful to interpret the two approaches in (2.8) and (2.9) in terms of graph-based definition of the KDGP (Problem 2). The sequence of incomplete and noisy distances, $\widetilde{D}_{t_1}, \cdots, \widetilde{D}_{t_T}$ is modeled as a series of incomplete graphs whose edge weights correspond to the measured distances. The goal of KDGP is to find a node function $x(u,t)$ that maps vertices of measurement graphs to points in $\mathbb{R}^d$ whose pairwise distances match the measured distances at sampling times $t_k \in \mathcal{T}$. From this perspective, the formulation (2.8) aims to directly estimate the node function $x(u,t)$ from distance measurements, while in formulation (2.9), we break the KDGP into two subproblems:

1. **Completing the measurement graphs:** This amounts to estimating the edge function, $f(e,t)$ for every $e \in E_t$ instead of only for $e \in E_i$, with E_i being the edges measured at time $t_i \in \mathcal{T}$.

2. **Estimating the node function,** $x(u,t)$: This is equivalent to spectral factorization of the time-dependent Gramian.

The formulation (2.9) solves the first subproblem since it outputs a time-varying Gramian $G(t)$ from which we easily get the KEDM as $\mathcal{K}(G(t))$. The second problem is addressed in Section 2.5.

Finally, we note that the KEDM formulation in (2.9) is a generalization of the static EDM formulation in (2.3). To see the equivalence, note that static points are modeled by a polynomial of degree zero, $P = 0$, in which case the Gramian becomes $G(t) = G_0$ since $w_0(t) = 1$.

2.4.2 Practical Considerations: Relax and Sample

To get a practical algorithm for (2.9), we sample the continuous-time semidefiniteness constraint, $G(t) \succeq 0$ for all $t \in \mathbb{R}$, and relax the non-convex rank

constraint. In Algorithm 1, we denote the set of sampling times for this constraint by $\mathcal{T}_{\text{psd}}$.

In relaxations for static EDMs, instead of simply removing the rank constraint, it is often replaced by a regularizer. Perhaps counterintuitively (see [5] for a longer discussion), a strategy that works well is to *maximize* the rank of the Gram matrix, as this corresponds to pushing the points apart and minimizing the embedding dimension. We use a similar strategy in our KEDM semidefinite relaxation (Algorithm 1).

One issue with the semidefinite relaxation for the standard DGP is that there are often no strictly feasible points; the feasible Gram matrices lie on the low-rank faces of the positive semidefinite cone. This is troublesome for the primal–dual interior point solvers since it precludes strong duality (Slater's constraint qualification fails). On the other hand, Krislock and Wolkowicz skillfully exploit it by noting that the degeneracy is due to the existence of cliques in the DGP graph. They characterize faces of the positive semidefinite cone associated with individual cliques, and design fast, accurate solvers for noiseless instances [85].

Whether their ideas can be applied to the KDGP remains an open question. At a glance, it seems challenging: not only does the connectivity graph in the KDGP change between the sampling instants, but we work with a non-unique decomposition of the time-varying Gramian into basis Gramians.

Thus, even with low-rank-promoting regularization, the recovered Gram matrices will rarely be *exactly* rank-d due noise and numerical issues of the off-the-shelf semidefinite solvers. To address this, we apply a standard rank projection to the retrieved Gramians by setting the least significant $N - d$ singular values to 0.

2.5 Spectral Factorization of the Gramian

Algorithm 1 produces a time-varying Gramian whose KEDM best represents the measured distance sequence. In this section, we show how to estimate the corresponding trajectory by factoring the Gramian as $G(t) = X(t)^\top X(t)$, where $X(t)$ is the set of point trajectories. We know that the trajectory can only be estimated up to a time-invariant rotation (and possibly reflection) [83] and a time-varying translation. To resolve this uncertainty, we introduce a

Algorithm 1 Semidefinite relaxation to solve KDGPs — $\texttt{SDR}\big(\{t_i\}_{i=1}^T, \{\widetilde{D}_{t_i}\}_{i=1}^T, \{W_i\}_{i=1}^T\big)$.

1: Solve for $\{G_k\}_{k=0}^K$:

$$\begin{aligned}
&\text{minimize} && \sum_{i=1}^{T} \alpha_i \left\| \tilde{D}_{t_i} - W_i \circ \mathcal{K}\left(\sum_{k=0}^{K} w_k(t_i) G_k\right)\right\|_F^2 - \lambda \sum_{k=0}^{K} \operatorname{Tr}(G_k) \\
&\text{w.r.t} && G_0, \cdots, G_K \succeq 0 \\
&\text{such that} && G_k 1 = 0 \qquad k \in \{0, \cdots, K\}, \\
& && \sum_{k=0}^{K} w_k(t) G_k \succeq 0 \qquad t \in \mathcal{T}_{\text{psd}}.
\end{aligned}$$

2: $G_k \leftarrow \texttt{RankProjection}(G_k, d)$, $k \in \{0, \cdots, K\}$

3: **return** $\widehat{D}(t) = \mathcal{K}\left(\sum_{k=0}^{K} w_k(t) G_k\right)$

set of anchors—points whose absolute positions are known.

In practice, anchors might correspond to nodes whose position is fixed such as buoys and beacons, or nodes equipped with a positioning technology such as GPS. Because in the KDGP the anchors can move (unlike in the usual DGP), we have more possibilities for anchor measurements than in the static case. For our trajectory models, we only need to know the positions of the anchor points at some fixed, finite set of times, but we could measure positions of different sets of points at different times.

Given a spectral factor[6] $\overline{X}(t)$ of the time-varying Gramian, the true trajectory $X(t)$ can be found as

$$X(t) = U\overline{X}(t) + x(t)1^\top + N(t),$$

where U is a $d \times d$ orthogonal matrix, $x(t)$ is a $d \times 1$ time-varying vector and $N(t)$ represents the net effect of model mismatch and measurement noise. The matrix U is constant (by the spectral factorization theorem) whereas the translation factor $x(t)$ is a function of time. On the other hand, the translation factor $x(t)$ must belong to the same trajectory model as $X(t)$ (polynomial or bandlimited). Hence, $x(t)$ can be written as

$$x(t) = Mz(t),$$

[6]One out of infinitely many possible.

where for the polynomial model $\mathcal{X}_{\text{poly}}$, we have

$$z(t) = [1,\ t,\ \cdots,\ t^P]^\top \text{ and } M \in \mathbb{R}^{d\times(P+1)}.$$

For the bandlimited model $\mathcal{X}_{\text{BL}}$, we have

$$z(t) = [1, \sin(\omega t), \cos(\omega t), \cdots, \sin(P\omega t), \cos(P\omega t)]^\top \text{ and } M \in \mathbb{R}^{d\times(2P+1)}.$$

A difficulty compared to the static case is that spectral factorization of polynomial Gram matrices is not straightforward and becomes brittle in the presence of noise. It is thus desirable to develop trajectory estimation methods that do not require full polynomial factorization. We show that this is possible at the expense of additional anchor measurements.

2.5.1 Known Spectral Factor

We start by assuming that we have access to *some* spectral factor $\overline{X}(t)$ such that $G(t) = \overline{X}(t)^\top \overline{X}(t)$. In this case, to estimate the unknown rotation and translation, we assume that at L distinct times $\tau_1, \ldots, \tau_L$ we measure positions of points $\mathcal{I}_1, \ldots, \mathcal{I}_L$, with $\mathcal{I}_\ell$ being the index set of points whose positions are measured at τ_ℓ. We let $X_{\mathcal{I}_\ell}$ denote the column selection of $X(\tau_\ell)$ corresponding to indices in $\mathcal{I}_\ell$.

An estimate for U and M can be computed by solving

$$\underset{U\in M_d(\mathbb{R}), M\in\mathbb{R}^{d\times L}}{\arg\min} \sum_{\ell=1}^{L} \left\| X_{\mathcal{I}_\ell} - U\overline{X}(\tau_\ell) - Mz(\tau_\ell)1^\top \right\|_F^2,$$

where $M_d(\mathbb{R})$ is the set of $d \times d$ orthonormal matrices and $L = P + 1$ (resp. $2P + 1$) for polynomial (resp. bandlimited) trajectories. This is a non-convex problem because $M_d(\mathbb{R})$ is a non-convex set.

The above optimization can be decoupled as in standard Procrustes analysis provided that there exists a time $\widetilde{\tau}_\ell \in \{\tau_1, \ldots, \tau_L\}$ at which we know the positions of at least $d + 1$ anchors. In this case, U can be estimated at this time alone using the technique described in Section 2.2.2. Once the rotation $\widehat{U}$ is found, we can estimate the matrix M by solving the following convex

problem:

$$\widehat{M} = \underset{M \in \mathbb{R}^{d \times L}}{\arg\min} \sum_{\ell=1}^{L} \left\| M z(\tau_\ell) - \frac{1}{N_{\tau_\ell}} \big(X_{\mathcal{I}_\ell}(\tau_\ell) - \widehat{U} \overline{X}(\tau_\ell) \big) 1 \right\|_2^2,$$

where $N_{\tau_l} \geq 1$ for $\ell \geq 2$. Finally, we note that matching d points (instead of $d+1$ points) leaves us with a flip ambiguity. So $d+1$ is indeed the smallest number of anchors that lets us to properly use the Procrustes analysis.

2.5.2 Unknown Spectral Factor (Practical Algorithm)

The previous section implies that $L+d$ anchor points are necessary to estimate the rotation U and translation M provided that a spectral factor $\overline{X}(t)$ of $G(t)$ is given. Unfortunately, algorithms for spectral factorization rely on unstable computations involving determinants and are often computationally demanding, which makes them unsuitable for our application where noise can be significant [86]. To avoid this step, we propose a method which relies on additional anchor measurements.

Assume that at each of L distinct times we measure positions of at least $d+1$ anchors; as before, denote the anchor indices at time τ_ℓ by $\mathcal{I}_\ell$, and the corresponding positions by $X_{\mathcal{I}_\ell}$. Now we can use Procrustes analysis at each time individually (that is, applied to constant matrices that are evaluations of time-varying matrices at these particular times) to estimate rotation and translation, $\widehat{U}_{\tau_l}$ and $\widehat{x}(\tau_l)$ at time τ_l. Denote by $\overline{X}(\tau_\ell)$ any matrix such that $\overline{X}(\tau_\ell)^\top \overline{X}(\tau_\ell) = G(\tau_\ell)$; since this involves only constant matrices, we can use the eigendecomposition method described in Section 2.2.1 to compute $\overline{X}(\tau_\ell)$.

Note that in doing so, there is no guarantee that these "marginal" estimates for the rotation correspond to the unique global U we are looking for, because we do not exploit any temporal model in computing the spectral factors $\overline{X}(\tau_\ell)$. In other words, all $\widehat{U}_{\tau_l}$ could be distinct, and in principle they will. Nevertheless, we can use them to estimate the trajectory by solving the following problem:

$$\Theta^* = \underset{\Theta \in \mathcal{A}}{\arg\min} \sum_{l=1}^{L} \left\| X_\Theta(\tau_l) - \big(\widehat{U}_{\tau_l} \overline{X}(\tau_l) + \widehat{x}(\tau_l) 1^\top \big) \right\|_F^2. \tag{2.11}$$

Algorithm 2 Spectral Factorization — $\texttt{SF}(\widehat{D}(t),\, \{X_{\mathcal{I}_\ell}\}_{l=1}^{L})$.

for $l \in \{1, \cdots, L\}$
  $\widehat{G}(\tau_l) \leftarrow -\frac{1}{2} J_N \widehat{D}(\tau_l) J_N$
  $\overline{X}(\tau_l) \leftarrow \widehat{G}(\tau_l)^{1/2}$
  Solve for $\widehat{U}_{\tau_l}$ using Procrustes analysis
  Estimate the translation at time τ_ℓ:

$$\widehat{x}(\tau_l) \leftarrow \frac{1}{N_{\tau_l}} (X_{\mathcal{I}_\ell} - \widehat{U}\overline{X}_{\mathcal{I}_\ell}(\tau_l)_l) 1$$

  Estimate point positions at time τ_ℓ:

$$\widehat{X}(\tau_l) \leftarrow \widehat{U}_{\tau_l} \overline{X}(\tau_l) + \widehat{x}(\tau_l) 1^\top$$

end for
Find the trajectory:

$$\Theta \leftarrow \arg\min_{\Theta \in \mathcal{A}} \sum_{\ell=1}^{L} \| X_\Theta(\tau_\ell) - \left(\widehat{U}_{\tau_\ell} \overline{X}(\tau_\ell) + \widehat{x}(\tau_\ell) 1^\top \right) \|_F^2$$

return $X_\Theta(t)$

The logic behind (2.11) is that even though the matrices $\widehat{U}_{\tau_\ell}$ are "wrong", the product $\widehat{U}_{\tau_l} \overline{X}(\tau_l)$ is correct thanks to the anchors. With sufficiently many marginal estimates, there is a unique set of polynomial trajectories passing through them. The described procedure is summarized in Algorithm 2 and the complete KDGP trajectory localization algorithm with anchors in Algorithm 3.

2.6 Simulation Results

We empirically evaluate different aspects of the proposed algorithm. We first study the influence of sampling time distribution in Section 2.6.1 as this choice

Algorithm 3 Overall KDGP algorithm — $\texttt{KDGP}(\{X_{\mathcal{I}_\ell}\}, \{t_i\}, \{\widetilde{D}_{t_i}\}, \{W_i\})$.

$\widehat{D}(t) = \texttt{SDR}(\{t_i\}, \{\widetilde{D}_{t_i}\}, \{W_i\})$
$X_\Theta(t) = \texttt{SF}(\widehat{D}(t),\, \{X_{\mathcal{I}_\ell}\})$
return $X_\Theta(t)$

affects the other experiments. In Section 2.6.2, we look at the maximum achievable measurement sparsity:[7] KDGP measurements are a sequence of incomplete EDMs and it is interesting to understand what proportion of missing entries we can tolerate.[8]

- For polynomial model, we uniformly generate samples t_i from $[T^-, T^+]$ for some $T^- \ll 0 \ll T^+$. Then, $\mathcal{T}_{\text{psd}}^+ = \{e^{t_i}\}$. Similarly $\mathcal{T}_{\text{psd}}^- = \{-e^{t_i}\}$ and $\mathcal{T}_{\text{psd}} = \mathcal{T}_{\text{psd}}^+ \cup \mathcal{T}_{\text{psd}}^-$.
- For bandlimited, $\mathcal{T}_{\text{psd}}$ is comprised of uniformly generated samples in $[0, \frac{2\pi}{\omega}]$.

Finally, in Section 2.6.3 we study the effect of measurement noise on the quality of the estimated trajectories. We conclude this section by applying our algorithms to a synthetic problem of satellite localization from noisy and very sparse distance measurements.

2.6.1 Distribution of Sampling Times

The measurements in Algorithm 3 are a sequence of (incomplete) snapshots of KEDM at different times, $\{W_i \circ \mathcal{D}(X(t_i))\}_i$. We experiment with different choices of sampling times $\{t_i\}_i$. To exclude the influence of other factors, we assume having access to all pairwise distances, and we contaminate the measurements by noise. Note that without noise, we can compute the Gramian basis simply by solving a linear system of equations so that any sampling strategy with sufficiently many samples gives the perfect estimation.

Let the true, noiseless distances be $d_{ij}(t) = \|x_i(t) - x_j(t)\|$ and noisy measurements given as

$$\widetilde{d}_{ij}(t) = d_{ij}(t) + n_{ij}(t), \tag{2.12}$$

where $n_{ij}(t) \sim \mathcal{N}(0, \sigma^2)$ is iid measurement noise. The corresponding KEDMs are $D(t) = [d_{ij}^2(t)]_{ij}$ and $\widetilde{D}(t) = [\widetilde{d}_{ij}^2(t)]_{ij}$.

[7] We use the term "sparsity" to refer to sparse or subsampled measured data, as is common in the inverse problems theory.

[8] In all experiments we sample the positive semidefinite constraint at random times. We have found empirically that this choice does not matter much, unlike the choice of measurement times. The exact details can be found in the reproducible code at `https://github.com/swing-research/kedm/`

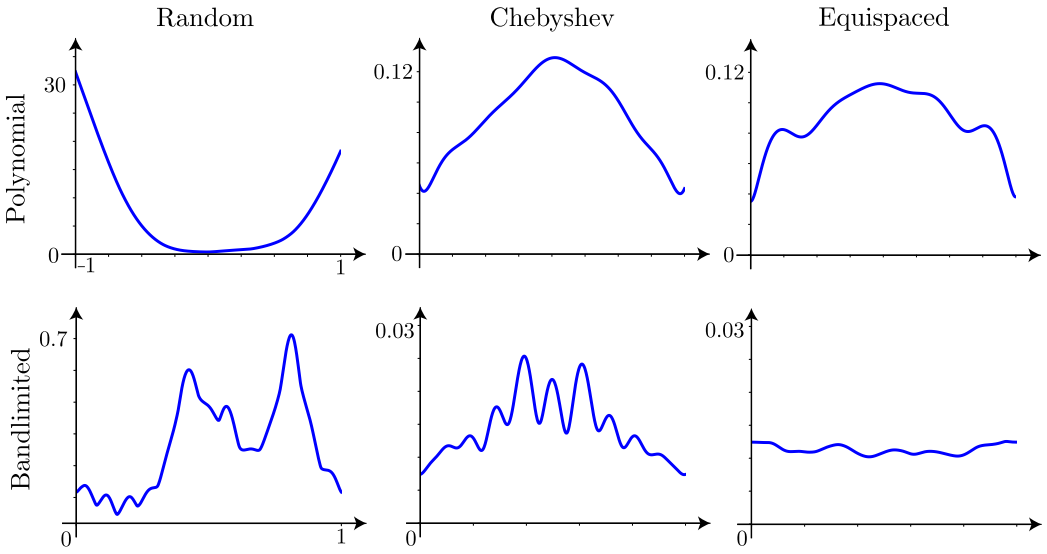


Figure 2.4: Relative reconstruction error $e_D(t)$ averaged over $M = 200$ realizations. The number of points is $N = 10$, ambient dimension $d = 2$, trajectory degree $P = 3$ and noise variance $\sigma^2 = 1$ for both models. The trajectory parameters, A_p, are drawn iid Gaussian—real valued for polynomial and complex for bandlimited with complex exponential basis. The sampled interval of interest is $[-1, 1]$ for the polynomial and $[0, 1]$ for the bandlimited model.

To compare the different sampling protocols, we average the reconstruction error over many trajectory and noise realizations. The reconstruction error is defined as

$$e_D(t) = \frac{\|D(t) - \widehat{D}(t)\|_F}{\|D(t)\|_F},$$

where $\widehat{D}(t) = \texttt{SDR}((\{t_i\}, \{\widetilde{D}_{t_i}\}, \{W_i\}))_{i=1}^T)$ is the KEDM estimated by Algorithm 3. The goal is to determine which sampling pattern minimizes $e_D(t)$ for all t in the interval of interest $[T_1, T_2]$. In Figure 2.4, we show the average errors for the following sampling patterns:

- random: $t_i \sim \text{Unif}([T_1, T_2])$,
- Chebyshev: $t_i = \frac{1}{2}(T_1 + T_2) + \frac{1}{2}(T_2 - T_1)\cos(\frac{2i-1}{2T}\pi)$,
- equispaced: $t_i = T_1 + (T_2 - T_1)\frac{i}{T}$,

where $i = 1, \ldots, T$. We can see that random sampling performs poorly for both the polynomial and the bandlimited model. Chebyshev and equispaced nodes give a similar relative error, with equispaced nodes performing slightly

better for the bandlimited model. Studying individual realizations shows that the worst-case error for Chebyshev and equispaced sampling is on the same order as the average error, but it is much worse for random sampling: large reconstruction errors occur when two consecutive measurement times are far apart. In the following experiments, we use equispaced measurement times.

All experiments were run on a laptop with a 2.9 GHz Core i5 processor and 16 GB of memory, using the `cvxpy` package [87, 88]. The interior point methods used by solvers in `cvxpy` tend to become slow as the number of points and the polynomial degree grow (e.g., for $N \geq 20, P \geq 5$), and should be replaced by faster, tailor-made optimizers.

2.6.2 Measurement Sparsity

Trajectory estimation from distances is a nonlinear sampling problem, with trajectory models allowing us to trade spatial for temporal samples. Here we empirically study the maximum sparsity level for spatial measurements. Given a sequence of measurement masks $W_1, \cdots, W_T \in \{0,1\}^{N \times N}$, the sparsity level, $0 \leq S \leq 1$, is defined as the ratio of average to total number of pairwise distances:

$$S = \frac{1}{\binom{N}{2}} \frac{1}{T} \sum_{i=1}^{T} \#\text{ of missing measurements at time } t_i.$$

We can expect the maximum sparsity level to vary with factors such as the trajectory model, temporal sampling pattern, measurement masks, and noise. To evaluate it, we fix parameters the trajectory class, degree, number of points, and ambient dimension. We declare a localization experiment successful if the relative trajectory mismatch

$$e_X = \int_{\mathcal{T}} \|X(t) - \widehat{X}(t)\|_F / \|X(t)\|_F \, \mathrm{d}t,$$

which we approximate by discretizing $\mathcal{T}$, is below some prescribe threshold δ. We are interested in numerically evaluating the probability that the localization succeeds (within tolerance δ) if on average over sampling times, m pairwise distances are missing. Denote this probability by $p(\delta, m)$. We would like to find conditions on m such that $p(\delta, m)$ is large. In particular,

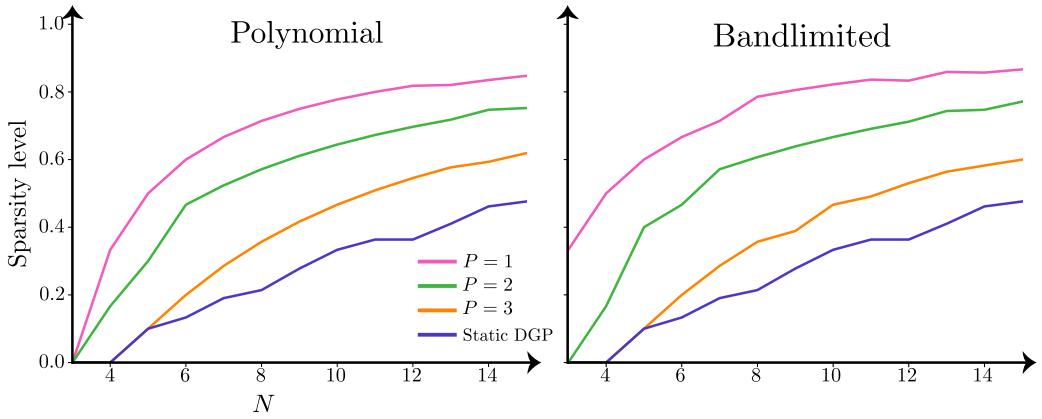


Figure 2.5: The estimated sparsity level $\widehat{S}$ for polynomial degrees P and numbers of points N. The success threshold δ is set to 0.99 and the target fraction of successful reconstructions q to 0.9.

for $0 \leq q < 1$, let $\overline{m}(\delta, q)$ be the largest m such that $p(\delta, m) \geq q$.

We run M localization trials for different realizations of random trajectories, and denote the number of succesful trials by M_1. For a given average number of missing pairwise distances m, the probability of correct localization is estimated as $\widehat{p}_M(\delta, m) = \frac{M_1}{M}$. The estimate of $\overline{m}(\delta, q)$ is then simply

$$\widehat{\overline{m}}(\delta, q) := \max \left\{ m \ : \ \widehat{p}_M(\delta, m) \geq q \right\}.$$

To compute $\widehat{\overline{m}}(\delta, q)$, we increase the number of missing measurements per sampling time, m, and count the number of δ-accurate estimates to compute $\widehat{\overline{m}}(\delta, q)$ and the corresponding $\widehat{S}(\delta, q) = \widehat{\overline{m}}(\delta, q) / \binom{N}{2}$.

In the first experiment, we fix the number of sampling times, T, and vary the number of points N and polynomial (or bandlimited) degree P. Specifically, in Figure 3.3 we choose $T = 7$ for polynomial and $T = 13$ for bandlimited models.

As expected, we observe that for a fixed N, as P grows (and consequently the number of parameters) the allowable sparsity level decreases, meaning that more complicated trajectories require more spatial samples. This is due to fact that ratio of number of measurements, which is fixed in this case, to number of parameters decreases. Importantly, compared to the static DGP, we see that KEDMs and the proposed semidefinite relaxation allow us to measure fewer distances at any given time, and compensate for this by sampling at multiple times.

In the second experiment we attempt to better characterize the observed spatio–temporal sampling tradeoff. To this end, we fix the parameters so that the ratio of the number of measurements to the number of the degrees of freedom is constant. That is, we keep the number of sampling times proportional to the number of basis Gramians, $T = K + 1$ for the polynomial and $T = 2K + 1$ for the bandlimited model.

As Tables 2.1 and 2.2 show, with this scaling the sparsity level is approximately constant as the polynomial degree P grows. In other words, even though the trajectories become more and more complicated, we can keep the number of spatial measurements fixed as long as we adjust the number of temporal sampling instants. The empirical observation that the required number of measurements scales linearly with the number of the degrees of freedom suggests that the proposed algorithms require an order-optimal number of samples.

However, there is a meaningful difference between sparsity levels for $P = 0$, i.e. static DGP, and $P \neq 0$ models. For simplicity, let us compare the polynomial models with $P = 0$ and $P = 1$. In the static model of $P = 0$, we sample the distance matrix one time and estimate point positions at that time, i.e. estimate A_0. On the other hand, for $P = 1$ model, we sample KEDM at $K + 1 = 3$ time instants to estimate A_0, A_1. This redundancy in parameterization of Gramian $G(t)$, which is due to convolution operator in (2.6), lets us achieve sparser measurements in non-trivial, $P \neq 0$, trajectory models.

Table 2.1: Maximal sparsity for the polynomial model and $d = 2$.

$P\backslash N$	5	6	7	8	9	10	11	12	13	14	15
$P = 1$	0.1	0.2	0.28	0.39	0.44	0.46	0.52	0.56	0.57	0.60	0.62
$P = 2$	0.1	0.2	0.33	0.35	0.41	0.46	0.51	0.54	0.57	0.60	0.62
$P = 3$	0.1	0.2	0.28	0.35	0.41	0.46	0.51	0.54	0.57	0.59	0.62

Table 2.2: Maximal sparsity for the bandlimited model and $d = 2$.

$P\backslash N$	5	6	7	8	9	10	11	12	13	14	15
$P = 1$	0.1	0.26	0.33	0.39	0.44	0.48	0.51	0.56	0.57	0.60	0.63
$P = 2$	0.1	0.2	0.28	0.35	0.41	0.44	0.49	0.53	0.56	0.60	0.63
$P = 3$	0.1	0.2	0.28	0.35	0.38	0.46	0.49	0.53	0.56	0.58	0.60

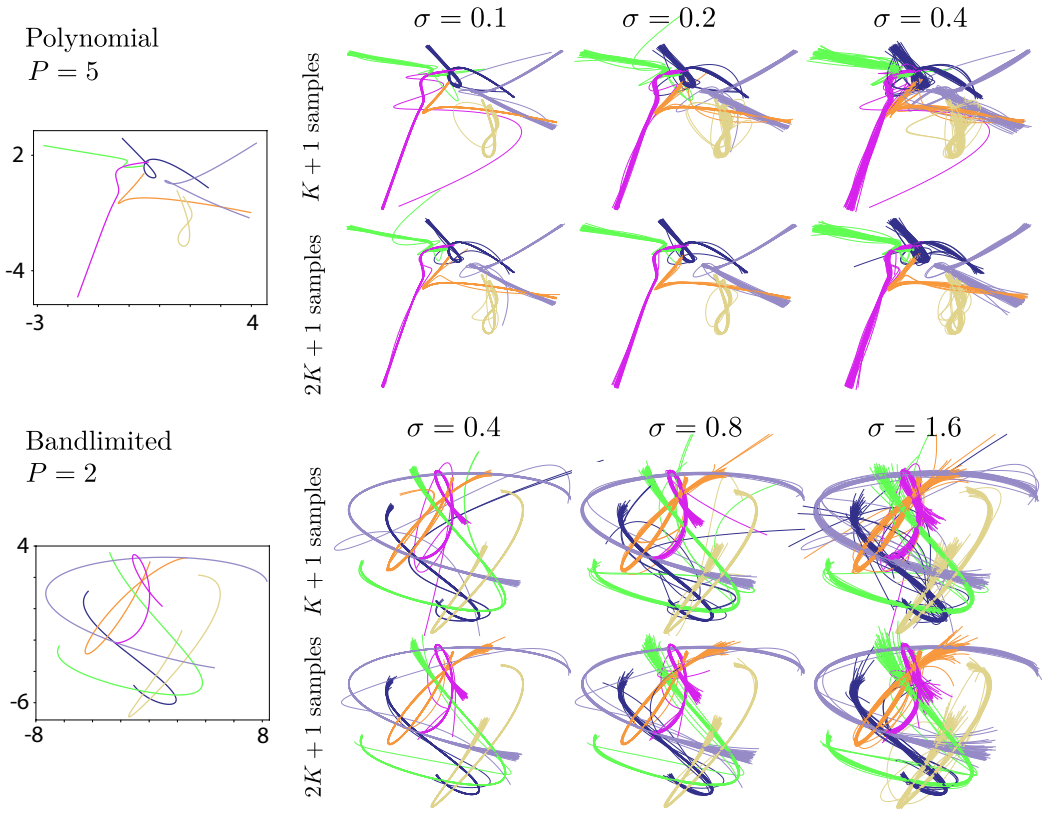


Figure 2.6: Estimated trajectories, $\widehat{X}(t)$, for $N = 6$ points in $\mathbb{R}^2$ at different levels of measurement noise and number of temporal measurements. The time interval of interest is $t \in [-1, 1]$ for polynomial and $t \in [0, 1]$ for bandlimited trajectories.

2.6.3 Noisy Measurements

We again quantify the influence of noise by the relative trajectory mismatch. We fix a trajectory, shown in Figure 2.6, and a set of distance sampling times $\{t_k\}_{k=0}^{K}$, and generate many realizations of noisy measurement sequences $\widetilde{\mathcal{D}}_{t_0}, \cdots, \widetilde{\mathcal{D}}_{t_K}$ with the same noise variance σ^2. The i.i.d. noise is added to the non-squared distances. The empirical trajectory mismatch is an average of relative trajectory mismatches over realizations, $\frac{1}{M}\sum_m e_X^{(m)}$.

In Figure 2.6, we show many estimated trajectories $\widehat{X}(t)$. As expected, the mismatch increases with measurement noise σ^2 and decreases with the number of measurements. In all cases, the estimated trajectories concentrate around the true ones.

2.6.4 A Stylized Application: Satellite Positioning

In this section we apply KEDMs in a stylized satellite positioning scenario where measurements are both very sparse and noisy. We consider a set of satellites moving with constant angular velocity, with angular frequency being an integer multiple of the fundamental frequency ω_0. Such trajectories have the form

$$x(t) = R \begin{pmatrix} a\cos(\omega t) \\ b\sin(\omega t) \\ 0 \end{pmatrix},$$

where R is a 3×3 rotation matrix.

The set of all satellite trajectories

$$X(t) = [x_1(t, p_1), \cdots, x_N(t, p_N)]$$

follows the bandlimited trajectory model. Concretely,

$$x(t, p) = a_1 \cos(p\omega_0 t) + a_2 \sin(p\omega_0 t)$$

is the trajectory of a satellite whose angular frequency is p times the fundamental frequency ω_0 and $a_1, a_2 \in \mathbb{R}^3$. The ensemble trajectory, $X(t)$, is a bandlimited trajectory of degree $P = \max_n p_n$.

We apply Algorithm 3 in two experiments. In Figure 2.7, we show trajectories of $N = 8$ satellites with the same orbiting frequency ω_0. Since the ellipses are of different sizes, the inner points can also be interpreted as vehicles on the earth. We measure three noisy pairwise distances, out of 28 available, per sampling time instant. This could model, for instance, occlusions by the Earth and other adversarial effects. We compensate for undersampling in space by oversampling in time, taking samples at $T = 30$ different times. Similarly, in Figure 2.8 we show $N = 5$ satellites with angular frequencies ω_0 and $2\omega_0$, that is, with $P = 2$; we measure only two pairwise distances per sampling time instant (these are extremely sparse measurements with which static localization is hopeless), at $T = 30$ sampling times. As figures show, in both experiments, we successfully reconstruct trajectories of the satellites.

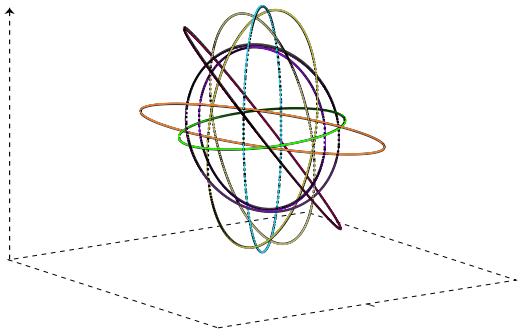


Figure 2.7: Reconstructing the trajectories of eight orbiting satellites. Colored and dashed lines represent actual and estimated trajectories. All satellites have the same angular frequency with $P = 1$. The measurement matrices are missing about 9/10 measurements, and noise level is set to $\sigma = 0.05$. The average reconstruction error is $\frac{1}{M}\sum_i e_D(t_i) = 0.01$.

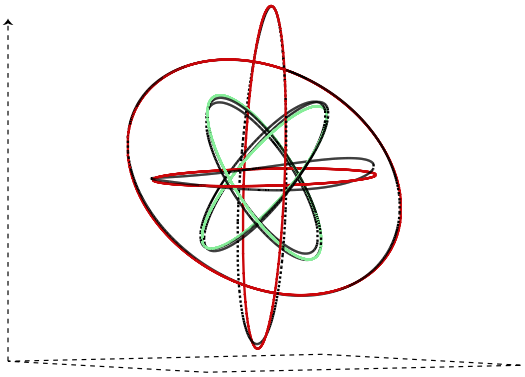


Figure 2.8: Reconstructing the trajectories of five orbiting satellites with angular frequencies of ω_0 and $2\omega_0$. The measurement matrices are 80% sparse, and average reconstruction error is $\frac{1}{M}\sum_i e_D(t_i) = 0.03$.

2.7 Conclusion

We extended the algebraic tools for localization from distances to the case when points are moving. We defined kinetic Euclidean distance matrices for polynomial and bandlimited trajectories, and we derived algorithms based on semidefinite programming to solve the associated trajectory localization problem. The chosen trajectory models are expressive and can approximate continuous trajectories commonly used in localization and tracking. The key step in our method is to represent the time-varying Gram matrices as time-varying linear combinations of certain constant matrices. This allowed us to rewrite the localization problem as a semidefinite program. Same as in the static case, the actual localization involves an additional spectral

factorization step. However, for polynomial matrices, this is much harder than a simple SVD, and especially from noisy data like those that we get. We circumvent the related difficulties by deriving a spectral factorization method that directly uses anchor measurements. We demonstrated through numerical experiments that the proposed algorithms can indeed reconstruct model trajectories from sparse and noisy measurements, and that they can explore the tradeoff between the number of distances measured at any given time, and the number of sampling times.

CHAPTER 3

HYPERBOLIC DISTANCE MATRICES

3.1 Introduction[1]

Hyperbolic spaces can embed hierarchical structures uniformly and with arbitrarily low distortion [23, 24]. In comparison, Euclidean spaces cannot achieve comparably low distortion even using an unbounded number of dimensions [89]. Embedding objects in hyperbolic spaces has found a myriad applications in exploratory science, from visualizing hierarchical structures such as social networks and link prediction for symbolic data [90, 21] to natural language processing [91, 37], brain networks [92], gene ontologies [93] and recommender systems [94, 95].

Commonly in these applications, there is a tree-like data structure which encodes *similarity* between a number of entities. We experimentally observe some relational information about the structure and the data mining task is to find a geometric representation of the entities consistent with the experimental information. In other words, the task is to compute an embedding. This concept is closely related to the classical distance geometry problems and multidimensional scaling (MDS) [96] in Euclidean spaces [10, 5].

The observations can be metric or nonmetric. Metric observations convey (inexact) distances; for example, in internet distance embedding a small subset of nodes with complete distance information are used to estimate the remaining distances [97]. Nonmetric observations tell us which pairs of entities are closer and which are further apart. The measure of closeness is typically derived from domain knowledge; for example, word embedding algorithms aim to relate semantically close words and their topics [98, 99].

[1]Reprinted, with permission, from P. Tabaghi and I. Dokmanic, *Hyperbolic Distance Matrices*, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020 [19]. The published manuscript is available at `https://doi.org/10.1145/3394486.3403224`

In scientific applications it is desirable to compute good low-dimensional hyperbolic embeddings. Insisting on low dimension not only facilitates visualization, but also promotes simple explanations of the phenomenon under study. However, in most works that leverage hyperbolic geometry the embedding technique is not the primary focus and the related computations are often ad hoc. The situation is different in the Euclidean case, where the notions of MDS, Euclidean distance matrices (EDMs) and their characterization in terms of positive semidefinite Gram matrices play a central role in the design and analysis of algorithms [10, 100].

In this chapter, we focus on computing low-dimensional hyperbolic embeddings. While there exists a strong link between Euclidean geometry and positive (semi)definiteness, we prove that what we call *hyperbolic distance matrices* (HDMs) can also be characterized via semidefinite constraints. Unlike in the Euclidean case, the hyperbolic analogy of the Euclidean Gram matrix is a linear combination of two rank-constrained semidefinite variables. Together with a spectral factorization method to directly estimate the hyperbolic points, this characterization gives rise to flexible embedding algorithms which can handle diverse constraints and mixed metric and nonmetric data.

3.1.1 Related Work

The usefulness of hyperbolic space stems from its ability to efficiently represent the geometry of complex networks [101, 102]. Embedding metric graphs with underlying hyperbolic geometry has applications in word embedding [98, 99], geographic routing [103], routing in dynamical graphs [104], odor embedding [26], internet network embedding for delay estimation and server selection [97, 105], to name a few. In the literature such problems are known as hyperbolic multidimensional scaling [34].

There exist Riemann gradient-based approaches [35, 36, 21, 37] which can be used to directly estimate such embeddings from metric measurements [38]. We emphasize that these methods are iterative and only guaranteed to return a locally optimal solution. On the other hand, there exist one-shot methods to estimate hyperbolic embeddings from a *complete* set of measured distances. The method of Wilson *et al.* [106] is based on spectral factorization of an inner product matrix (we refer to it as hyperbolic Gramian) that directly

minimizes a suitable *stress*. In this chapter, we derive a semidefinite relaxation to estimate the *missing* measurements and denoise the distance matrix, and then follow it with the spectral embedding algorithm to find the embeddings.

Nonmetric (or order) embedding has been proposed to learn visual-semantic hierarchies from ordered input pairs by embedding symbolic objects into a low-dimensional space [107]. In the Euclidean case, stochastic triplet embeddings [108], crowd kernels [109], and generalized nonmetric MDS [52] are some well-known order embedding algorithms. For embedding hierarchical structures, Ganea *et al.* [33] model order relations as a family of nested geodesically convex cones. Zhou *et al.* [26] show that odors can be efficiently embedded in hyperbolic space provided that the similarity between odors is based on the statistics of their co-occurrences within natural mixtures.

3.1.2 Contributions

We summarize our main contributions as follows:

- **Semidefinite characterization of HDMs:** We introduce HDMs as an elegant tool to formalize distance problems in hyperbolic space; this is analogous to Euclidean distance matrices (EDM). We derive a semidefinite characterization of HDMs by studying the properties of hyperbolic Gram matrices—matrices of Lorentzian (indefinite) inner products of points in a hyperbolic space.

- **A flexible algorithm for hyperbolic distance geometry problems (HDGPs):** We use the semidefinite characterization to propose a flexible embedding algorithm based on semidefinite programming. It allows us to seamlessly combine metric and nonmetric problems in one framework and to handle a diverse set of constraints. The nonmetric and metric measurements are imputed as linear and quadratic constraints.

- **Estimate the embedded points:** We propose a suboptimal method to find a low-rank approximation of the hyperbolic Gramian in the desired dimension. This method relies on a spectral factorization technique that was proposed at least as early as in [106], and as a result, gives the points' positions in the hyperbolic space.

Table 3.1: Essential elements in semidefinite approach for distance problems, Euclidean versus hyperbolic space.

Euclidean	Hyperbolic
Euclidean Distance Matrix	Hyperbolic Distance Matrix
Gramian	H-Gramian
Semidefinite relaxation to complete an EDM	Semidefinite relaxation to complete an HDM
Spectral factorization of a Gramian to estimate the points	Spectral factorization of an H-Gramian to estimate the points

3.1.3 Outline

We first briefly review the analytical models of hyperbolic space and formalize hyperbolic distance geometry problems (HDGPs) in Section 3.2.3. Our framework is parallel with semidefinite approaches for Euclidean distance problems as per Table 3.1. In the hyperboloid ('Loid) model, we define hyperbolic distance matrices to compactly encode hyperbolic distance measurements. We show that an HDM can be characterized in terms of the matrix of *indefinite* inner products, the hyperbolic Gramian. In Section 3.3, we propose a semidefinite representation of hyperbolic Gramians, and in turn HDMs. We cast HDGPs as rank-constrained semidefinite programs, which are then convexified by relaxing the rank constraints. We use a spectral method to find a sub-optimal low-rank approximation of the hyperbolic Gramian, to the correct embedding dimension. Lastly, we use closed-form factorization and rank correction methods to estimate the embedded points. Our proposed framework lets us tackle a variety of embedding problems, as shown in Section 3.4, with real (odors) and synthetic (random trees) data. Finally, in Appendix B, we present the derivations of proposed algorithms and the proofs of all propositions.

3.2 Hyperbolic Distance Geometry Problems

Hyperbolic space is a simply connected Riemannian manifold with constant negative curvature [110, 111]. In comparison, Euclidean and elliptic geometries are spaces with zero (flat) and constant positive curvatures. There are five isometric models for hyperbolic space: half-space ($\mathbb{H}^d$), Poincaré (interior of

the disk) ($\mathbb{I}^d$), jemisphere ($\mathbb{J}^d$), Klein ($\mathbb{K}^d$), and 'Loid ($\mathbb{L}^d$) [110] (Figure 3.1). Each provides unique insights into the properties of hyperbolic geometry.

In the machine learning community the most popular models of hyperbolic geometry are Poincaré and 'Loid. We work in the 'Loid model as it has a simple, tractable distance function. It lets us cast the HDGP (formally defined in Section 3.2.3) as a rank-constrained semidefinite program. Importantly, it also leads to a closed-form embedding by a spectral method. For better visualization, however, we map the final embedded points to the Poincaré model via the stereographic projection, see Sections 3.2.2 and 3.4.

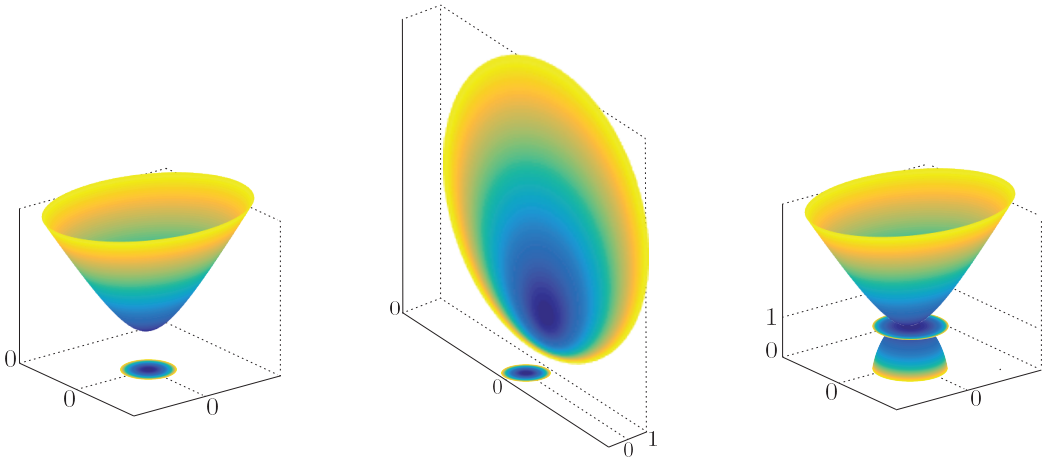
3.2.1 'Loid Model

Let x and y be vectors in $\mathbb{R}^{d+1}$ with $d \geq 1$. The Lorentzian inner product of x and y is defined as

$$[x, y] = x^\top H y, \tag{3.1}$$

where

$$H = \begin{pmatrix} -1 & 0^\top \\ 0 & I \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}. \tag{3.2}$$



$(\mathbb{L}^2, \mathbb{I}^2)$ $(\mathbb{H}^2, \mathbb{I}^2)$ $(\mathbb{L}^2, \mathbb{K}^2, \mathbb{J}^2)$

Figure 3.1: Models of hyperbolic space with level sets (colors) illustrating isometries.

This is an indefinite inner product on $\mathbb{R}^{d+1}$. The Lorentzian inner product has almost all the properties of ordinary inner products, except that

$$\|x\|_H^2 \stackrel{\text{def}}{=} [x, x]$$

can be positive, zero, or negative. The vector space $\mathbb{R}^{d+1}$ equipped with the Lorentzian inner product (5.5) is called a Lorentzian $(d+1)$-space, and is denoted by $\mathbb{R}^{1,d}$. In a Lorentzian space we can define notions similar to the Gram matrix, adjoint, and unitary matrices known from Euclidean spaces as follows.

Definition 3 (H-adjoint [112])**.** *The H-adjoint $R^{[*]}$ of an arbitrary matrix $R \in \mathbb{R}^{(d+1)\times(d+1)}$ is characterized by*

$$[Rx, y] = [x, R^{[*]}y], \ \forall x, y \in \mathbb{R}^{d+1}.$$

Equivalently,

$$R^{[*]} = H^{-1}R^\top H.$$

Definition 4 (H-unitary matrix [112])**.** *An invertible matrix R is called H-unitary if $R^{[*]} = R^{-1}$.*

The 'Loid model of d-dimensional hyperbolic space is a Riemannian manifold $\mathcal{L}^d = (\mathbb{L}^d, (g_x)_x)$, where

$$\mathbb{L}^d = \left\{x \in \mathbb{R}^{d+1} : \|x\|_H^2 = -1, x_0 > 0\right\}$$

and $g_x = H$ is the Riemannian metric.

Definition 5 (Lorentz Gramian, H-Gramian)**.** *Let the columns of $X = [x_1, x_2, \cdots, x_N]$ be the positions of N points in $\mathbb{R}^{d+1}$ (resp. $\mathbb{L}^d$). We define their corresponding Lorentz Gramian (resp. H-Gramian) as*

$$\begin{aligned} G &= ([x_i, x_j])_{i,j\in[N]} \\ &= X^\top H X, \end{aligned}$$

where H is the indefinite matrix given by (3.2)*.*

The subtle difference between the Lorentz Gramian (defined for points in

$\mathbb{R}^{d+1}$) and the H-Gramian (defined only on $\mathbb{L}^d \subset \mathbb{R}^{d+1}$) will be important for the low-rank projection and the spectral factorization steps in Section 3.3. The indefinite inner product (5.5) also determines the distance between $x, y \in \mathbb{L}^d$ as

$$d(x, y) = \operatorname{acosh}(-[x, y]).$$

3.2.2 Poincaré Model

In the Poincaré model ($\mathbb{I}^d$), the points reside in the unit d-dimensional Euclidean ball. The distance between $x, y \in \mathbb{I}^d$ is given by

$$d(x, y) = \operatorname{acosh}\Big(1 + 2\frac{\|x - y\|^2}{(1 - \|x\|^2)(1 - \|y\|^2)}\Big).$$

The isometry between the 'Loid and the Poincaré models, $h : \mathbb{L}^d \to \mathbb{I}^d$ is called the *stereographic projection*. For $y = h(x)$, we have

$$y_i = \frac{x_{i+1}}{x_0 + 1}. \tag{3.3}$$

The inverse of stereographic projection is given by

$$x = h^{-1}(y) = \frac{1}{1 - \|y\|^2} \begin{bmatrix} 1 + \|y\|^2 \\ 2y \end{bmatrix}.$$

The isometry between the 'Loid and Poincaré models makes them equivalent in their embedding capabilities. However, the Poincaré model facilitates visualization of the embedded points in a bounded disk, whereas the 'Loid model is an unbounded space.

3.2.3 Hyperbolic Distance Problems

In a metric hyperbolic distance problem, we want to find a point set $x_1, \ldots, x_N \in \mathbb{L}^d$ such that

$$\text{for all } (m, n) \in \mathcal{C} : d_{mn} = \operatorname{acosh}\left(-[x_m, x_n]\right),$$

for a subset of measured distances $\mathcal{D} = \{d_{mn} : (m, n) \in \mathcal{C} \subseteq [N]^2_{\text{as}}\}$.

In many applications we have access to the *true* distances only through an unknown nonlinear map $\tilde{d}_{mn} = \phi(d_{mn})$; examples are connectivity strength of neurons [46] or odor co-ocurrence statistics [26]. If all we know is that $\phi(\cdot)$ is a monotonically increasing function, then only the ordinal information has remained intact, i.e.,

$$d_{kl} \leq d_{mn} \Leftrightarrow \tilde{d}_{kl} \leq \tilde{d}_{mn}.$$

This leads to nonmetric problems in which the measurements are in the form of binary comparisons [52].

Definition 6. *For a set of binary distance comparisons of the form $d_{kl} \leq d_{mn}$, we define the set of ordinal distance measurements as*

$$\mathcal{O} = \left\{(k,l,m,n) : d_{kl} \leq d_{mn}, (k,l), (m,n) \in [N]^2_{\text{as}}\right\}.$$

We are now in a position to give a unified definition of metric and nonmetric embedding problems in a hyperbolic space.

Problem 3. *A hyperbolic distance geometry problem aims to find $x_1, \ldots, x_N \in \mathbb{L}^d$, given*

- *a subset of pairwise distances $\mathcal{D}$ such that*

$$d_{mn} = d(x_m, x_n), \quad \text{for all } d_{mn} \in \mathcal{D}$$

- *and/or a subset of ordinal distances measurements $\mathcal{O}$ such that*

$$d(x_{i_1}, x_{i_2}) \leq d(x_{i_3}, x_{i_4}), \quad \text{for all } i \in \mathcal{O},$$

where $d(x,y) = \operatorname{acosh}(-[x,y])$ and $i = (i_1, i_2, i_3, i_4)$.

We denote the complete sets of metric and nonmetric measurements by $\mathcal{D}_c$ and $\mathcal{O}_c$.

3.3 Hyperbolic Distance Matrices

We now introduce hyperbolic distance matrices in analogy with Euclidean distance matrices to compactly encode interpoint distances of a set of points

$x_1, \ldots, x_N \in \mathbb{L}^d$.

Definition 7. *The hyperbolic distance matrix (HDM) corresponding to the list of points $X = [x_1, \ldots, x_N] \in (\mathbb{L}^d)^N$ is defined as*

$$D = \mathcal{D}(X) = (d(x_i, x_j))_{i,j\in[N]}\,.$$

The ij-th element of $\mathcal{D}(X)$ is hyperbolic distance between x_i and x_j, given by $d(x_i, x_j) = \operatorname{acosh}(-[x_i, x_j])$ and for all $i, j \in [N]$.

HDMs are characterized by Lorentzian inner products which allows us to leverage the definition of an H-Gramian (Definition 5). Given points $x_1, \ldots, x_N \in \mathbb{L}^d$, we compactly write the HDM corresponding to G as

$$D = \operatorname{acosh}[-G],$$

where $\operatorname{acosh}[\cdot]$ is an elementwise $\operatorname{acosh}(\cdot)$ operator.

We now state our first main result: a semidefinite characterization of H-Gramians. This is a key step in casting HDGPs as rank-constrained semidefinite programs.

Proposition 6 (Semidefinite characterization of H-Gramian)**.** *Let G be the hyperbolic Gram matrix for a set of points $x_1, \cdots, x_N \in \mathbb{L}^d$. Then,*

$$\begin{aligned} &G = G^+ - G^- \\ &\textit{where} \qquad\qquad G^+, G^- \succeq 0 \\ &\qquad\qquad \operatorname{rank} G^+ \leq d \\ &\qquad\qquad \operatorname{rank} G^- \leq 1 \\ &\qquad\qquad \operatorname{diag} G = -1 \\ &\qquad\qquad e_i^\top G e_j \leq -1, \quad \forall i, j \in [N]. \end{aligned}$$

Conversely, any matrix $G \in \mathbb{R}^{N\times N}$ that satisfies the above conditions is a hyperbolic Gramian for a set of N points in $\mathbb{L}^d$.

3.3.1 Solving for H-Gramians

While Problem 3 could be formalized directly in X domain, this approach is unfavorable as the optimization domain, $\mathbb{L}^d$, is a non-convex set. What is more, the hyperbolic distances

$$d(x_m, x_n) = \operatorname{acosh}\left(-e_m^\top X^\top H X e_n\right)$$

are nonlinear functions of X with an unbounded gradient [34]. Similar issues arise when computing embeddings in other spaces such as Euclidean [5] or the space of polynomial trajectories [7]. A particularly effective strategy in the Euclidean case is the semidefinite relaxation which relies on the simple fact that the Euclidean Gramian is positive semidefinite. We thus proceed by formulating a semidefinite relaxation for hyperbolic embeddings based on Proposition 6.

Solving the HDGP involves two steps, summarized in Algorithm 4:

1. Complete and denoise the HDM via a semidefinite program.

2. Compute an embedding of the clean HDM: we propose a closed-form spectral factorization method.

Note that step (2) is independent of step (1): given accurate hyperbolic distances, spectral factorization will give the points that reproduce them. However, since the semidefinite relaxation might give a Gramian with a higher rank than desired, eigenvalue thresholding in step (2) might move the points off of $\mathbb{L}^d$. That is because eigenvalue thresholding can violate the necessary condition for the hyperbolic norm, $\|x\|_H^2 = -1$, or $\operatorname{diag} G = -1$ in Proposition 6. We fix this by projecting each individual point to $\mathbb{L}^d$. The spectral factorization and the projection are given in Algorithms 6 and 9.

Let $\widetilde{D}$ be the measured noisy and incomplete HDM, with unknown entries replaced by zeroes. We define the mask matrix $W = (w_{ij})$ as

$$w_{ij} \overset{\text{def}}{=} \begin{cases} 1, \text{ for } (i,j) \in \mathcal{C} \vee (j,i) \in \mathcal{C} \\ 0, \text{ otherwise.} \end{cases}.$$

This mask matrix lets us compute the loss only at those entries that were actually measured. We use the semidefinite characterization of hyperbolic

Algorithm 4 HDGP algorithm — $\texttt{HDGP}(\widetilde{D}, \widetilde{\mathcal{O}}, d)$.

Input: Incomplete and noisy distance matrix, $\widetilde{D}$, and ordinal measurements, $\widetilde{\mathcal{O}}$, and embedding dimension, d.
$G = \texttt{SDR}(\widetilde{D}, \widetilde{\mathcal{O}}, d) \Longrightarrow$ *Complete & denoise HDM*
$X = \texttt{Embed}(G, d) \Longrightarrow$ *Embed points in* $\mathbb{L}^d$

$$y_n = h(x_n), \ \forall n \in [N]$$

where $h(\cdot)$ is given by (3.3) $\Longrightarrow$ *Map the points to* $\mathbb{I}^d$
return $Y = [y_1, \cdots, y_N] \in \left(\mathbb{I}^d\right)^N$.

Gramians in Proposition 6 to complete and denoise the measured HDM, and eventually solve HDGP.

Although the set of hyperbolic Gramians for a given embedding dimension is non-convex due to the rank constraints, discarding the rank constraints results in a straightforward semidefinite relaxation.

However, if we convexify the problem by simply discarding the rank constraints, then all pairs $(G_1, G_2) \in \{(G^+ + P, G^- + P) : P \succeq 0\}$ become a valid solution. On the other hand, since

$$\text{rank } G + P \geq \text{rank } G \ \text{ for } \ G, P \succeq 0,$$

we can eliminate this ambiguity by promoting low-rank solutions for G^+ and G^-. While directly minimizing

$$\text{rank } G^+ + \text{rank } G^- \tag{3.4}$$

is NP-hard [113], there exist many approaches to make (3.4) computationally tractable, such as trace norm minimization [114], iteratively reweighted least squares minimization [115], or the log-det heuristic [116] that minimizes the following smooth surrogate for (3.4):

$$\log\det(G^+ + \delta I) + \log\det(G^- + \delta I),$$

where $\delta > 0$ is a small regularization constant. This objective function is linearized as $C + \text{Tr } W_k^+ G^+ + \text{Tr } W_k^- G^-$ for $W_k^+ = (G_k^+ + \delta I)^{-1}$ and $W_k^- = (G_k^- + \delta I)^{-1}$, which can be iteratively minimized.[2] In our numeri-

[2] In practice, we choose a diminishing sequence of δ_k.

Algorithm 5 Semidefinite relaxation for HDGP — $\texttt{SDR}(\widetilde{D}, \widetilde{\mathcal{O}}, d)$.

Input: Incomplete and noisy distance matrix, $\widetilde{D}$, and ordinal measurements, $\widetilde{\mathcal{O}}$, and embedding dimension, d.
Let W be the measurement mask. For small $\epsilon_1, \epsilon_2 > 0$, solve for G:

$$
\begin{array}{lll}
\text{minimize} & \operatorname{Tr} G^+ + \operatorname{Tr} G^- & \\
\text{w.r.t} & G^+, G^- \succeq 0 & \\
\text{subject to} & G = G^+ - G^-, & \\
& \operatorname{diag} G = -1, & \\
& e_i^\top G e_j \leq -1, & \forall i, j \in [N] \\
& \left\| W \circ \left(\cosh[\widetilde{D}] + G \right) \right\|_F^2 \leq \epsilon_1, & \\
& \mathcal{L}_k(G) \geq \epsilon_2, & \forall k \in \widetilde{\mathcal{O}}.
\end{array}
$$

return G.

cal experiments we will uset he trace norm minimization unless otherwise stated. Then, we enforce the data fidelity objectives and the properties of the embeddings space (Proposition 6) in the form of a variety of constraints.

Metric embedding: The quadratic constraint

$$\left\| W \circ \left(\cosh[\widetilde{D}] + G \right) \right\|_F^2 \leq \epsilon_1$$

makes sure the hyperbolic Gramian G accurately reproduces the given distance data.

Nonmetric embedding: The ordinal measurement constraint of

$$d(x_{i_1}, x_{i_2}) \leq d(x_{i_3}, x_{i_4}),$$

is simply a linear constraint in form of

$$\mathcal{L}_i(G) = e_{i_1}^\top G e_{i_2} - e_{i_3}^\top G e_{i_4} \geq 0,$$

where $i \in \mathcal{O}$ and $i = (i_1, i_2, i_3, i_4)$. In practice, we replace this constraint by $\mathcal{L}_i(G) \geq \epsilon_2 > 0$ to avoid trivial solutions.

'Loid model: The unit hyperbolic norm appears as a linear constraint, $\operatorname{diag} G = -1$,, which guarantees that the embedded points reside in sheets $\mathbb{L}^d \cup -\mathbb{L}^d$. Finally, $e_i^\top G e_j \leq -1$ enforces all embedded points to belong to

the same hyperbolic sheet, i.e. $x_n \in \mathbb{L}^d$ for all $n \in [N]$.

This framework can serve as a bedrock for multitude of other data fidelity objectives. We can seamlessly incorporate *outlier removal* schemes by introducing slack variables into the objective function and constraints [117, 118, 119]. For example, the modified objective function

$$\operatorname{Tr} G^+ + \operatorname{Tr} G^- + \sum_k \epsilon_k$$

can be minimized subject to $\mathcal{L}_k(G) + \epsilon_k \geq 0$ and $\epsilon_k \geq 0$ as a means of removing outlier comparisons (we allow some comparisons to be violated; see Section 3.4.3 for an example).

We can similarly implement outlier detection in metric embedding problems. As an example, we can adapt the outlier pursuit algorithm [120]. Consider the measured H-Gramian of a point set with a few outliers

$$\hat{G} = G + C + N,$$

where G is outlier-free hyperbolic Gramian, C is a matrix with only few nonzero columns and N represents the measurement noise. Outlier pursuit aims to minimize a convex surrogate for

$$\operatorname{rank} G + \lambda \left\|C\right\|_{0,c} \text{ s.t. } \left\|\hat{G} - G - C\right\|_F^2 \leq \epsilon,$$

where $\left\|C\right\|_{0,c}$ is the number of nonzero columns of C.

We note that scalability of semidefinite programs has been studied in a number of recent works [121], for example based on matrix sketching [122, 123].

3.3.2 Low-rank Approximation of H-Gramians

From Proposition 6, it is clear that the rank of a hyperbolic Gramian of points in $\mathbb{L}^d$ is at most $d+1$. However, the H-Gramian estimated by the semidefinite relaxation in Algorithm 5 does not necessarily have the correct rank. Therefore, we want to find its best rank-$(d+1)$ approximation, namely $\hat{G}$, such that

$$\left\|G - \hat{G}\right\|_F^2 = \inf_{X \in \left(\mathbb{L}^d\right)^N} \left\|G - X^\top H X\right\|_F^2. \tag{3.5}$$

In Algorithm 6 we propose a simple but suboptimal procedure to solve this low-rank approximation problem. Unlike iterative refinement algorithms based on optimization on manifolds [124], our proposed method is one-shot. It is based on the spectral factorization of the the estimated hyperbolic Gramian and involves the following steps:

Step 1: We find a set of points $\{z_n\}$ in $\mathbb{R}^{d+1}$, whose Lorentz Gramian best approximates G; See Definition 5 and lines 2 to 5 of Algorithm 6. In other words, we relax the optimization domain of (3.5) from $\mathbb{L}^d$ to $\mathbb{R}^{d+1}$,

$$Z = \underset{X \in \mathbb{R}^{(d+1)\times N}}{\arg\min} \left\| G - X^\top H X \right\|^2.$$

Step 2: We project each point z_n onto $\mathbb{L}^d$, i.e.,

$$\hat{X} = \underset{X \in \left(\mathbb{L}^d\right)^N}{\arg\min} \|X - Z\|_F^2.$$

This gives us an approximate rank-$(d+1)$ hyperbolic Gramian, $\hat{G} = \hat{X}^\top H \hat{X}$; see Figure 3.2 and Algorithm 6.

The first step of low-rank approximation of a hyperbolic Gramian G can

---

Algorithm 6 Low-rank approximation and spectral factorization of hyperbolic Gramian — $\texttt{Embed}(G, d)$.

Input: Hyperbolic Gramian G, and embedding dimension d.
Let $U^\top \Lambda U$ be eigenvalue decomposition of G, where $\Lambda = \operatorname{diag}(\lambda_0, \cdots, \lambda_{N-1})$ such that

- $\lambda_0 = \min_i \lambda_i$,
- λ_i is the top i-th element of $\{\lambda_i\}$ for $i \in [N] - 1$.

Let $G_{d+1} = U_d^\top \Lambda_d U_d$, where

$$\Lambda_d = \operatorname{diag}\left(\lambda_0, u(\lambda_1), \cdots, u(\lambda_d)\right),$$

$u(x) = \max\{x, 0\}$, and U_d be the corresponding sliced eigenvalue matrix.
$Z = R|\Lambda_d|^{1/2} U_d^\top$, for an arbitrary H-unitary matrix R.
For $Z = [z_1, \ldots, z_N]$, let

$$x_n = \texttt{Project}(z_n), \ \forall n \in [N].$$

return $X = [x_1, \ldots, x_N] \in \left(\mathbb{L}^d\right)^N$.

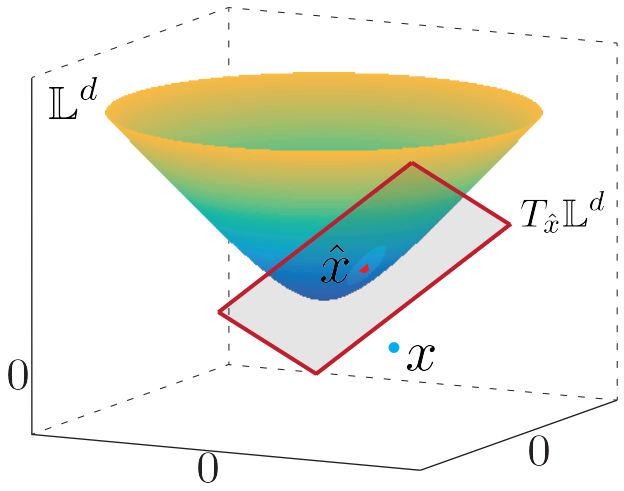


Figure 3.2: Projecting a point in $\mathbb{R}^{d+1}$ (blue) to $\mathbb{L}^d$ (red).

be interpreted as finding the positions of points in $\mathbb{R}^{d+1}$ (not necessarily on $\mathbb{L}^d$) whose Lorentz Gramian best approximates G.

3.3.3 Spectral Factorization of H-Gramians

To finally compute the point locations, we describe a spectral factorization method, proposed in [106], to estimate point positions from their Lorentz Gramian. This method exploits the fact that Lorentz Gramians have only one non-positive eigenvalue (see Lemma 2 in Appendix B) as detailed in the following proposition.

Proposition 7. *Let G be a hyperbolic Gramian for $X \in \left(\mathbb{L}^d\right)^N$, with eigenvalue decomposition $G = U\Lambda U^\top$, and eigenvalues $\lambda_0 \leq 0 \leq \lambda_1 \leq \ldots \leq \lambda_d$. Then, there exists an H-unitary matrix R such that $X = R|\Lambda|^{1/2}U$.*

Note that regardless of the choice of R, $X = R|\Lambda|^{1/2}U$ will reproduce G and thus the corresponding distances. This is the rigid motion ambiguity familiar from the Euclidean case [110]. If we start with an H-Gramian with a wrong rank, we need to follow the spectral factorization by Step 2 where we project each point $z_n \in \mathbb{R}^{d+1}$ onto $\mathbb{L}^d$. This heuristic is suboptimal, but it is nevertheless appealing since it only requires a single one-shot calculation as detailed in Proposition 7.

3.4 Experimental Results

In this section we numerically demonstrate different properties of Algorithm 4 in solving HDGPs. In a general hyperbolic embedding problem, we have a mix of metric and nonmetric distance measurements which can be noisy and incomplete. Code, data and documentation to reproduce the experimental results are available at `https://github.com/puoya/hyperbolic-distance-matrices`.

3.4.1 Missing Measurements

Missing measurements are a common problem in hyperbolic embeddings of concept hierarchies. For example, hyperbolic embeddings of words based on Hearst-like patterns rely on co-occurrence probabilities of word pairs in a corpus such as WordNet [125]. These patterns are sparse since word pairs must be detected in the right configuration [37]. In perceptual embedding problems, we ask individuals to rate pairwise similarities for a set of objects. It may be difficult to collect and embed all pairwise comparisons in applications with large number of objects [52].

The proposed semidefinite relaxation gives a simple way to handle missing measurements. The *metric sampling density* $0 \leq S \leq 1$ of a measured HDM is the ratio of the number of missing measurements to total number of pairwise distances, $S = 1 - \frac{|\mathcal{D}|}{|\mathcal{D}_c|}$. We want to find the probability $p(S)$ of successful estimation given a sampling density S. In practice, we fix the embedding dimension, d, and the number of points, N, and randomly generate a point set, $X \in \left(\mathbb{L}^d\right)^N$. A trial is successful if we can solve the HDGP for noise-free

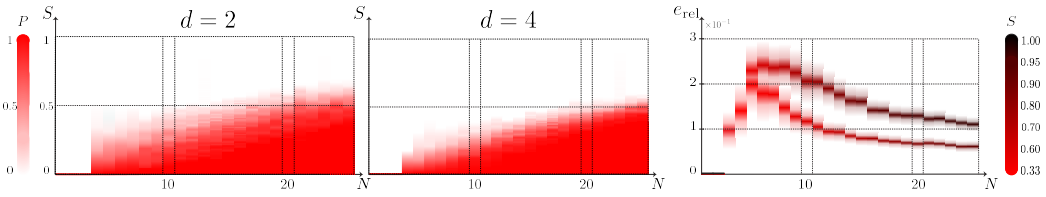


Figure 3.3: Left and middle: The probability of δ-accurate estimation for metric sampling density S, $M = 100$, and $\delta = 10^{-2}$. Right: The empirical error $e_{\text{rel}} = \mathbb{E}_K[e_{\text{rel}}(X)]$ for ordinal sampling density S, $d = 2$, $M = 50$, and $K = 10$. In each bar, shading width represents the empirical standard deviation of $e_{\text{rel}}(X)$.

measurements and a random mask W of a fixed size so that the estimated hyperbolic Gramian has a small relative error, $e_{\text{rel}}(\hat{G}) = \frac{\left\|\mathcal{D}(X) - \operatorname{acosh}[-\hat{G}]\right\|_F}{\|\mathcal{D}(X)\|_F} \leq \delta$. We repeat for M trials, and empirically estimate the success probability as $\hat{p}(S) = \frac{M_s}{M}$ where M_s is the number of successful trials. We repeat the experiment for different values of N and d, see Figure 3.3.

For nonmetric embedding applications, we want to have *consistent* embedding for missing ordinal measurements. The *ordinal sampling density* $0 \leq S \leq 1$ of a randomly selected set of ordinal measurements is defined as $S = 1 - \frac{|\mathcal{O}|}{|\mathcal{O}_c|}$. For a point set $X \in \left(\mathbb{L}^d\right)^N$, we define the average relative error of estimated HDMs as $e_{\text{rel}}(X) = \mathbb{E}_M \frac{\|D_{\mathcal{O}} - \mathbb{E}_M[D_{\mathcal{O}}]\|_F}{\|\mathbb{E}_M[D_{\mathcal{O}}]\|_F}$ where $D_{\mathcal{O}}$ is the estimated HDM for ordinal measurements $\mathcal{O}$, and empirical expectation is with respect to the random ordinal set $\mathcal{O}$. We repeat the experiment for K different realizations of $X \in \left(\mathbb{L}^d\right)^N$ (Figure 3.3). We can observe that across different embedding dimensions, the maximum allowed fraction of missing measurements for a consistent and accurate estimation increases with the number of points.

3.4.2 Weighted Tree Embedding

Tree-like hierarchical data occurs commonly in natural scenarios. In this section, we want to compare the embedding quality of weighted trees in hyperbolic and the baseline in Euclidean space.

We generate a random tree T with N nodes, maximum degree of $\Delta(T) = 3$, and i.i.d. edge weights from $\text{unif}(0, 1)$[3]. Let D_T be the distance matrix for T, where the distance between each two nodes is defined as the weight of the path joining them.

For the hyperbolic embedding, we apply Algorithm 5 with log-det heuristic objective function to acquire a low-rank embedding. On the other hand, Euclidean embedding of T is the solution to the following semidefinite relaxation

$$\begin{array}{ll} \text{minimize} & \left\|D_T^{\circ 2} - \mathcal{K}(G)\right\|_F^2 \\ \text{w.r.t} & G \succeq 0 \\ \text{subject to} & G\mathbf{1} = 0 \end{array} \tag{3.6}$$

[3]The most likely maximum degree for trees with $N \leq 25$ [126].

where $\mathcal{K}(G) = -2G + \mathrm{diag}(G)1^\top + 1\mathrm{diag}(G)^\top$ and $D_T^{\circ 2}$ is the entrywise square of D_T. This semidefinite relaxation (SDR) yields a *minimum error* embedding of T, since the embedded points can reside in an arbitrary dimensional Euclidean space.

The embedding methods based on semidefinite relaxation are generally accompanied by a projection step to account for the potentially incorrect embedding dimension. For hyperbolic embedding problems, this step is summarized in Algorithm 6, whereas it is simply a singular value thresholding of the Gramian for Euclidean problems. Note that the SDRs always find a $(N-1)$-dimensional embedding for a set of N points; see Algorithm 5 and (3.6). In this experiment, we define the optimal embedding dimension as

$$d_0 = \min\left\{ d \in \mathbb{N} : \frac{\|D_{N-1} - D_d\|_F}{\|D_{N-1} - D_{d+1}\|_F} \geq 1 - \delta \right\},$$

where D_n is the distance matrix for embedded points in $\mathbb{L}^n$ (or $\mathbb{R}^n$), and $\delta = 10^{-3}$. This way, we accurately represent the estimated distance matrix in a low-dimensional space. Finally, we define the relative (or normalized) error

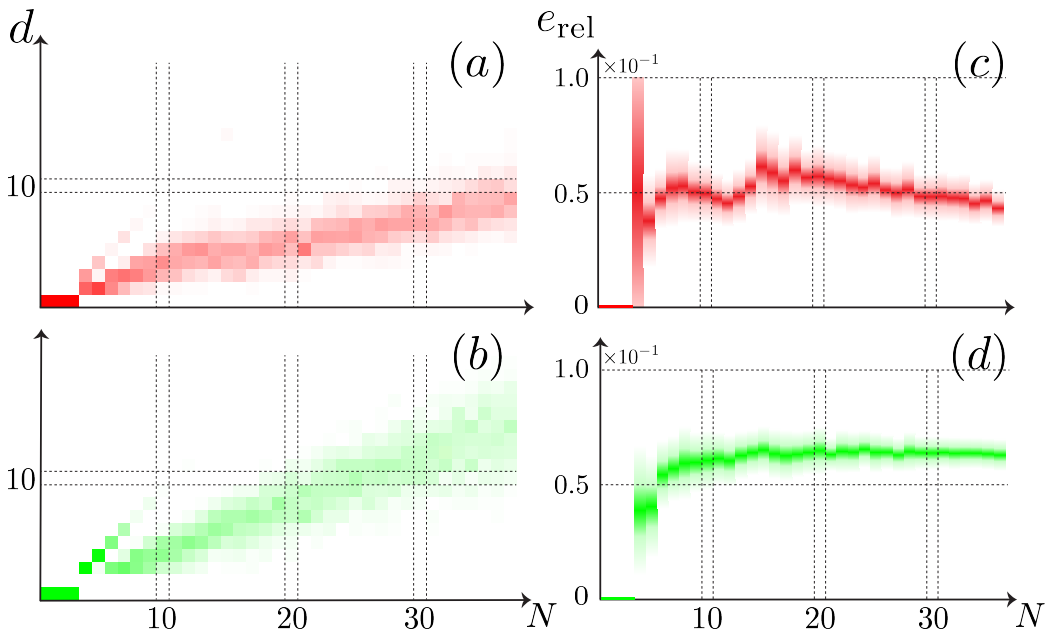


Figure 3.4: Tree embedding in hyperbolic (red) and Euclidean (green) space. Discrete distribution of optimal embedding dimension for $M = 100$, (a) and (b). Average, $\mathbb{E}_M[e_{\mathrm{rel}}(T)]$, and standard deviation of embedding error, (c) and (d).

of embedding T in d_0-dimensional space as $e_{\text{rel}}(T) = \frac{\|D_T - D_{d_0}\|_F}{\|D_T\|_F}$. We repeat the experiment for M randomly generated trees T with a varying number of vertices N. The hyperbolic embedding yields smaller average relative error $\mathbb{E}_M[e_{\text{rel}}(T)]$ compared to Euclidean embedding, see Figure 3.4. It should also noted that the hyperbolic embedding has a lower optimal embedding dimension, even though the low-rank hyperbolic Gramian approximation is sub-optimal.

3.4.3 Odor Embedding

In this section, we want to compare hyperbolic and Euclidean nonmetric embeddings of olfactory data following the work of Zhou *et al.* [26]. We conduct identical experiments in each space, and compare embedding quality of points from Algorithm 5 in hyperbolic space to its semidefinite relaxation counterpart in Euclidean space, namely generalized nonmetric MDS [52].

We use an olfactory dataset comprising mono-molecular odor concentrations measured from blueberries [127]. In this dataset, there are $N = 52$ odors across the total of $M = 164$ fruit samples. Like Zhou *et al.* [26], we begin by computing correlations between odor concentrations across samples [26]. The correlation coefficient between two odors x_i and x_j is defined as

$$C(i,j) = \frac{(x_i - \mu_{x_i}1)^\top (x_j - \mu_{x_j}1)}{\|x_i - \mu_{x_i}1\| \, \|x_j - \mu_{x_j}1\|},$$

where $x_n = (x_n^{(1)}, \ldots, x_n^{(M)})^\top$, $x_i^{(m)}$ is the concentration of i-th odor in m-th fruit sample, M is total number of fruit samples and $\mu_{x_n} = \frac{1}{M}\sum_{m=1}^{M} x_n^{(m)}$. The goal is to find an embedding for odors $y_1, \ldots, y_N \in \mathbb{I}^d$ (or $\mathbb{R}^d$) such that

$$d(y_{i_1}, y_{i_2}) \leq d(y_{i_3}, y_{i_4}), \quad (i_1, i_2, i_3, i_4) \in \mathcal{O},$$

where $\mathcal{O} \subseteq \mathcal{O}_c = \Big\{(i_1, i_2, i_3, i_4) \in ([N]^2_{\text{as}})^2 : C(i_1, i_2) \geq C(i_3, i_4)\Big\}$. The total number of distinct comparisons grows rapidly with the number of points, namely $|\mathcal{O}_c| = 0.87$ million. In this experiment, we choose a random set of size $|\mathcal{O}| = 2K\binom{N}{2}$ for $K = 4$ to have the sampling density of $S = 98.79\%$, which brings the size of ordinal measurements to $|\mathcal{O}| \approx 10^4$. In hyperbolic embedding, the sampling density is the ratio of number of ordinal measurements to number

of variables, i.e., $K = \frac{|\mathcal{O}|}{2\binom{N}{2}}$.

We ensure the embedded points do not collapse by imposing the following minimum distance constraint $d(x_i, x_j) \geq 1$ for all $(i, j) \in [N]^2_{\text{as}}$; this corresponds to a simple linear constraint in the proposed formulation. An ideal order embedding accurately reconstructs the missing comparisons. We calculate the percentage of correctly reconstructed distance comparisons as $\gamma_d = |\widehat{\mathcal{O}}_{c,d} \cap \mathcal{O}_c|/|\mathcal{O}_c|$, where $\widehat{\mathcal{O}}_{c,d}$ is the complete ordinal set corresponding to a d-dimensional embedding.

A simple regularization technique helps to remove outlier measurements and improve the generalized accuracy of embedding algorithms. We introduce the parameter ζ_p to permit SDR algorithms to dismiss at most p-percent of measurements, namely

$$\mathcal{L}_k(G) + \epsilon_k \geq \epsilon_2 \text{ and } \epsilon_k \geq 0, \ \forall k \in \mathcal{O} \text{ and } \sum_k \epsilon_k \leq \zeta_p,$$

where $\zeta_p = \frac{p}{100}|O|\epsilon_2$.

In Figure 3.5, we show the embedded points in $\mathbb{I}^2$ and $\mathbb{R}^2$ with different levels of allowable violated measurements. We can observe in Table 3.2 that hyperbolic space better represents the structure of olfactory data compared to Euclidean space of the same dimension. This is despite the fact that the

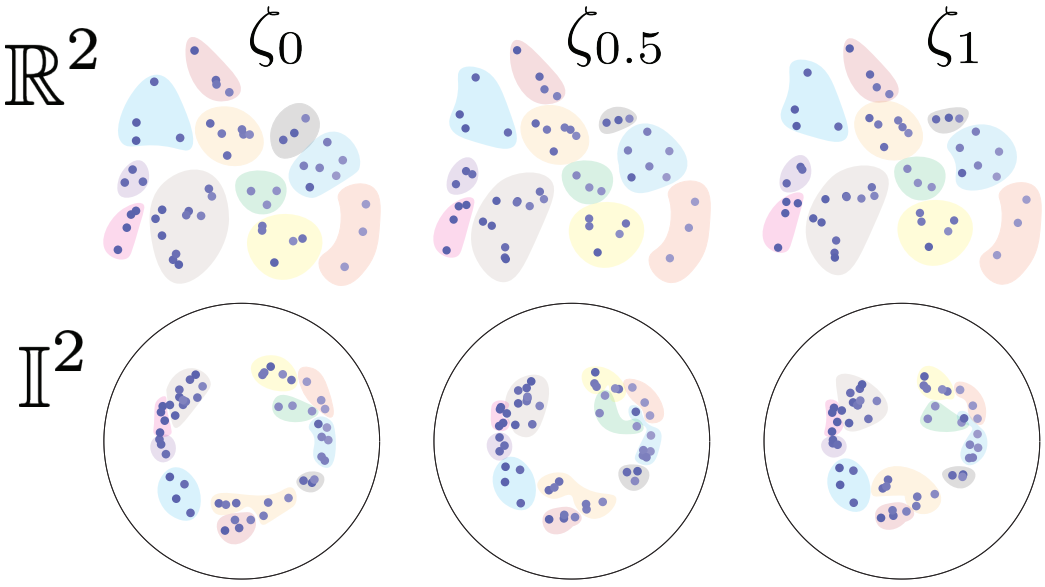


Figure 3.5: Embedding of odors for different levels of allowable violated measurements ζ_p. Clusters with the matching colors contain the same odors.

Table 3.2: Reconstruction accuracy of ordinal measurements γ_d for different levels of allowable violation ζ_p.

Space		$d=2$	$d=4$	$d=6$	$d=8$	$d=10$
Hyperbolic	ζ_0	76.06	83.60	86.87	89.48	91.03
	$\zeta_{0.5}$	76.52	83.71	86.94	89.68	91.16
	ζ_1	76.43	83.71	86.92	89.76	91.21
Euclidean	ζ_0	73.44	78.86	82.23	85.06	88.67
	$\zeta_{0.5}$	73.27	79.03	82.65	86.24	88.98
	ζ_1	73.12	78.92	82.51	86.01	89.02

number of measurements per variable is in favor of Euclidean embedding, and that the low-rank approximation of hyperbolic Gramians is suboptimal. Moreover, if we remove a small number of outliers we can produce more accurate embeddings. These results corroborate the statistical analysis of Zhou *et al.* [26] that aims to identify the geometry of the olfactory space.[4]

3.5 Conclusion

We introduced hyperbolic distance matrices, an analogy to Euclidean distance matrices, to encode pairwise distances in the 'Loid model of hyperbolic geometry. Same as in the Euclidean case, although the definition of hyperbolic distance matrices is trivial, analyzing their properties gives rise to powerful algorithms based on semidefinite programming. We proposed a semidefinite relaxation which is essentially plug-and-play: it easily handles a variety of metric and nonmetric constraints, outlier removal, and missing information and can serve as a template for different applications. Finally, we proposed a closed-form spectral factorization algorithm to estimate the point position from hyperbolic Gramians. In the next chapter, we study the role of the isometries in the 'Loid model and the related concepts such as Procrustes analysis.

[4]Statistical analysis of Betti curve behavior of underlying clique topology [46].

CHAPTER 4

HYPERBOLIC PROCRUSTES ANALYSIS

4.1 Introduction[1]

In 1962, Hurley and Catell introduced a point set matching problem known as Procrustes analysis [128].

Problem 4. *Let $\{z_n\}_{n=1}^N$ and $\{z'_n\}_{n=1}^N$ be two point sets in $\mathbb{R}^d$. Procrustes problem aims to find a map $\widehat{T}$ that minimizes the mismatch norm, i.e.,*

$$\widehat{T} = \underset{T \in \mathcal{T}}{\arg\min} \sum_{n=1}^{N} \|z_n - T(z'_n)\|_2^2 ,$$

where $\mathcal{T}$ is the set of "valid" maps, e.g., rotation, reflection, translation, and uniform scaling [129].

In computer vision, Procrustes analysis is relevant in *point cloud registering* problems. The task of rigid registration is to find an isometry between two (or more) sets of points sampled from a two- or three-dimensional object. Point registration has applications in object recognition [130], medical application [131] and localization of mobile robotics [132].

In signal processing, Procrustes analysis often refers to aligning shapes or point sets by a *distance preserving* bijection. Naturally, Procrustes analysis finds applications in distance geometry problems (DGPs) where we want to find the location of a point set that best matches with a set of given incomplete distances, i.e.,

$$z_1, \ldots, z_N \in \mathbb{R}^d : \|z_n - z_m\| = d_{mn},\ \forall (m,n) \in \mathcal{M},$$

[1] © 2021 IEEE. Reprinted, with permission, from P. Tabaghi and I. Dokmanic, *On Procrustes Analysis in Hyperbolic Space*, IEEE Signal Processing Letters, May 2021.

where $\{d_{m,n} : (m,n) \in \mathcal{M}\}$ is the set of measured distances [10]. If any, a distance geometry problem has a solution orbit of the form

$$O_{\mathcal{Z}} = \left\{ \{T(z_n)\}_{n=1}^{N} : T : \mathbb{R}^d \to \mathbb{R}^d \text{ is an isometry} \right\},$$

where $\mathcal{Z} = \{z_n\}_{n=1}^{N}$ is a particular solution. In order to uniquely identify the correct solution, from all the possible solutions in the orbit $O_{\mathcal{Z}}$, we may be given the exact position of a subset of points, called *anchors*. We use Procrustes analysis to pick the correct solution by finding the best match between the anchors with their corresponding points in the orbit. This technique is commonly used in localization problems [5, 7].

Procrustes analysis can be defined in any metric space. Hyperbolic Procrustes analysis is relevant since, in recent years, hyperbolic embedding problems are gaining attention in the machine learning community [19, 34]. Hierarchical or tree-like data structure is at the heart of hyperbolic embedding applications. Therefore, we can use hyperbolic Procrustes analysis to align hierarchical data, e.g., ontologies [133, 134]. In these problems, we want to find a (distance preserving) map between a fixed number of entities in two tree-like structures that best superimpose them on each other; see Figure 4.1.

4.1.1 Related Work

In unsupervised matching problems, an important first step is to find the correspondence between two point clouds, e.g., by iterative closest point algorithm [135]. A related line of research is *ontology matching* that aims to

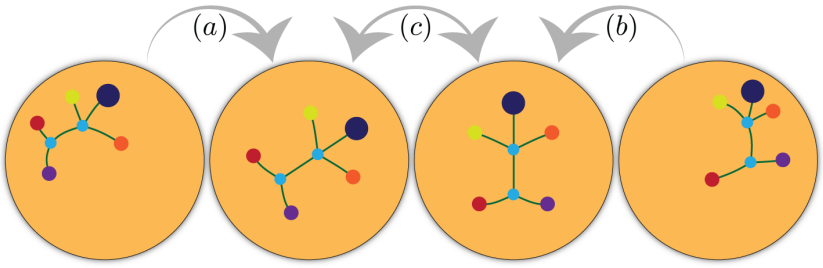


Figure 4.1: Tree alignment in Poincaré disk. Hyperbolic Procrustes analysis aims to align two trees — far left and far right figures. (a) and (b): we center each point set and (c) estimate the unknown rotation map.

find correspondences between semantically related entities in heterogeneous ontologies and has applications in ontology merging, query answering, or data translation [133]. Recently, Alvarez-Melis *et al.* [136]cast unsupervised hierarchy matching problem in hyperbolic space. Their proposed method jointly learns the "soft" correspondence and the aligning map characterized by a hyperbolic neural network.

4.1.2 Contributions and Outline

We review parametric isometries in the 'Loid model of hyperbolic spaces. We show how one can decompose any isometry into *elementary* isometries, e.g., hyperbolic translation, rotation, and reflection. The goal of Procrustes analysis is to find a joint estimate for hyperbolic translation and rotation maps that best aligns two point sets. In Section 4.3, we review the definition of center mass, or *centroid*, for a set of points in hyperbolic space. This enables us to "center" each set, and decouple the aforementioned joint estimation problems into the following steps: (1) translate the center mass of each point set to the coordinate origin (of the Poincaré model), and (2) estimate the unknown rotation factor. While hyperbolic centering have been studied in the literature [137], we present here a framework of Procrustes analysis similar to Euclidean counterpart, and give an optimal estimate for the unknown rotation factor — based on the weighted mean of pairwise inner products. More over, we prove that our proposed approach gives the theoretically optimal isometry if the point sets match perfectly. Finally, in Section 4.4, we give numerical performance bounds for matching noisy point sets. All proofs are delegated to Appendix C.

Summary: Let $\{x_n\}_{n\in[N]}$, and $\{x'_n\}_{n\in[N]}$ be two sets of points in a hyperbolic space, related through an isometric map, i.e., $x'_n = T(x_n), \forall n \in [N]$. Then,

$$T = T_{m_{x'}} \circ T_U \circ T_{-m_x},$$

where $m_x, m_y \in \mathbb{R}^d$ are points' centeroids, T_b is the translation operator by vector $b \in \mathbb{R}^d$, and T_U is a rotation operator by unitary matrix $U \in \mathbb{O}(d)$; see Section 4.3. For noisy points, i.e., $x'_n = T(x_n) + \epsilon_n, \forall n \in [N]$, this isometry is suboptimal (in ℓ_2 sense) and can be fine-tuned via a gradient-based algorithm.

4.2 Isometries in the 'Loid Model

Notations. Depending on the context, x_1 can either be the first element of vector x or an indexed vector. We denote the set of orthogonal matrix as $\mathbb{O}(d) = \{R \in \mathbb{R}^{d\times d} : R^\top R = I\}$. For any function f and its inputs $x_1, \ldots, x_N$, we define $\overline{f(x_n)} = \frac{1}{N}\sum_{n\in[N]} f(x_n)$.

The 'Loid model of d-dimensional hyperbolic space is a Riemannian manifold $\mathcal{L}^d = (\mathbb{L}^d, (g_x)_x)$, where

$$\mathbb{L}^d = \{x \in \mathbb{R}^{d+1} : [x, x] = -1, x_1 > 0\}$$

and $g_x = H$ is the Riemannian metric. Finally, 'Loid model's metric function is characterized by Lorentzian inner product, viz.

$$d(x, x') = \operatorname{acosh}(-[x, x']), \forall x, x' \in \mathbb{L}^d.$$

The map $T : \mathbb{L}^d \to \mathbb{L}^d$ is an isometry if it is bijective and preserves distances, i.e.,

$$d(x, x') = d\big(T(x), T(x')\big), \ \forall x, x' \in \mathbb{L}^d.$$

We can represent any hyperbolic isometry as a composition of two *elementary* maps that are parameterized by a d-dimensional vector and a $d \times d$ unitary matrix.

Fact 1. *[138] The function $T : \mathbb{L}^d \to \mathbb{L}^d$ is an isometry if and only if it can be written as $T(x) = R_U R_b x$, where*

$$R_U = \begin{bmatrix} 1 & 0^\top \\ 0 & U \end{bmatrix}, \; R_b = \begin{bmatrix} \sqrt{1 + \|b\|^2} & b^\top \\ b & (I + bb^\top)^{\frac{1}{2}} \end{bmatrix}$$

for a unitary matrix $U \in \mathbb{O}(d)$ and a vector $b \in \mathbb{R}^d$.

Fact 1 can be directly verified by finding the conditions for a real matrix R to be H-unitary, i.e., $R^\top H R = H$ or simply $R = H^{-\frac{1}{2}} C H^{\frac{1}{2}} \in \mathbb{R}^{(d+1)\times(d+1)}$, where $C^\top C = I$ and $C \in \mathbb{C}^{(d+1)\times(d+1)}$. We use this parametric decomposition of rigid transformations to solve the Procrustes problem in $\mathbb{L}^d$; see Section 4.3.

Fact 2. *$T_b^{-1} = T_{-b}$ and $T_U^{-1} = T_{U^\top}$, where $b \in \mathbb{R}^d$ and $U \in \mathbb{O}(d)$.*

The hyperbolic translation map $T_b : \mathbb{L}^d \to \mathbb{L}^d$ and hyperbolic rotation map $T_U : \mathbb{L}^d \to \mathbb{L}^d$ are defined as

$$\begin{aligned} T_b(x) &= R_b x, && \text{for } b \in \mathbb{R}^d \\ T_U(x) &= R_U x, && \text{for } U \in \mathbb{O}(d). \end{aligned}$$

4.3 Procrustes Analysis

Euclidean (orthogonal) Procrustes analysis has two main steps:

- Centering: moving the center mass of each points set to the origin of Cartesian coordinates.
- Finding the optimal rotation/reflection.

In this section, we review and visualize the textbook definition of center mass of point sets in hyperbolic space [137, Chapter 13].

We begin by projecting each point $x \in \mathbb{L}^d$ to the following sub-space

$$H_d = \left\{ x \in \mathbb{R}^{d+1} : x_1 = 0 \right\}.$$

Then, we can simply neglect the first element of the projected point (which

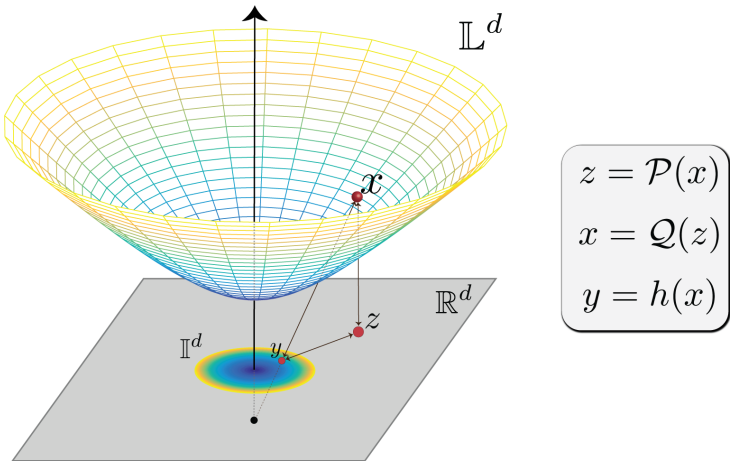


Figure 4.2: Geometric illustration of $\mathcal{P}$, $\mathcal{Q}$, and stereographic projection h.

is always zero), and define a bijection $\mathcal{P}$ between $\mathbb{L}^d$ and $\mathbb{R}^d$; see Figure 4.2. In Definition 8, we formalize this projection and its inverse function.

Definition 8. *We define the projection operator* $\mathcal{P} : \mathbb{L}^d \to \mathbb{R}^d$ *and its inverse function* $\mathcal{Q}$ *as*

$$\mathcal{P}\left(\begin{bmatrix} \sqrt{1 + \|z\|^2} \\ z \end{bmatrix}\right) = z, \;\; \mathcal{Q}(z) = \begin{bmatrix} \sqrt{1 + \|z\|^2} \\ z \end{bmatrix}.$$

For brevity, we define $\mathcal{P}(X) \stackrel{\text{def}}{=} [\mathcal{P}(x_1), \ldots, \mathcal{P}(x_N)]$ where $X = [x_1, \ldots, x_N] \in (\mathbb{L}^d)^N$. Similarly, we consider this extension for $\mathcal{Q}$ as well.

In Section 4.3.1, we review the hyperbolic centering process [137]. In other words, we find a map T_b to move the center mass of projected point sets to $0 \in \mathbb{R}^d$, i.e., $\overline{\mathcal{P}\big(T_b(x_n)\big)} = 0$. Then, we show how this centering method gives point sets whose locations are *invariant* with respect to arbitrary translations.

4.3.1 Hyperbolic Centering

In Euclidean Procrustes analysis, we have two point sets $z_1, \ldots, z_N$ and $z'_1, \ldots z'_N$ that are related via a composition of rotation, reflection, and translation maps, i.e.,

$$z_n = U z'_n + b,$$

where $U \in \mathbb{O}(d)$ and $b \in \mathbb{R}^d$. We extract translation invariant features by moving their point mass to $0 \in \mathbb{R}^d$, i.e.,

$$z_n - \overline{z_n} = U(z'_n - \overline{z'_n}).$$

The main purpose of centering is to map each point set to new locations, $z_n - \overline{z_n}$ and $z'_n - \overline{z'_n}$, that are invariant to the unknown translation b. This way, we can estimate the unknown unitary matrix $\hat{U}$, and the estimated translation term would be $\widehat{b} = \overline{z_n} - \widehat{U}\overline{z'_n}$.

In hyperbolic Procrustes analysis, we have

$$x_n = R_b R_U x'_n, \forall n \in [N], \tag{4.1}$$

where $U \in \mathbb{O}(d)$ and $b \in \mathbb{R}^d$. In a similar way, we pre-process a point set to extract (hyperbolic) translation invariant locations, i.e., centered point sets. In Proposition 8, we show that T_{-m_x} is the canonical translation map to center the point set $X \in \left(\mathbb{L}^d\right)^N$.

Proposition 8. *Let $x_1, \ldots, x_N$ and $x'_1, \ldots x'_N$ in $\mathbb{L}^d$ such that*

$$x_n = R_b R_U x'_n, \ \forall n \in [N],$$

for $b \in \mathbb{R}^d$ and $U \in \mathbb{O}(d)$. Then, $R_{-m_x} x_n = R_V R_{-m_{x'}} x'_n$ where R_V is a hyperbolic rotation matrix.

The map T_{-m_x} not only centers a set of points, but also rotates them. This phenomenon is rooted in non-commutative property of hyperbolic translation or *gyration*. More clearly, for any two vectors $b_1, b_2 \in \mathbb{R}^d$, we have

$$R_{b_1} R_{b_2} = R_V R_{b_2} R_{b_1}$$

for a specific unitary matrix $V \in \mathbb{O}(d)$ that accounts for the *gyration* factor; see the example in Figure 4.3 and further discussions in Section 4.3.3. This does not interfere with Procrustes analysis since any such rotation will be absorbed in U, and we estimate their *collective* unitary transformation.

Now, let us consider the following noisy case,

$$x_n = R_b R_U R_{\epsilon_n} x'_n, \ \forall n \in [N],$$

where $\epsilon_n \in \mathbb{R}^d$ is a translation noise for the point x'_n. Let $z_n = R_{\epsilon_n} x'_n$, then we have $R_{-m_x} x_n = R_V R_{-m_z} z_n$. The centroid m_z is a related to $m_{x'}$ and $\{\epsilon_n\}_{n\in[N]}$. Therefore, we can write $m_z = m_{x'} + \epsilon$ for a $\epsilon \in \mathbb{R}^d$. Therefore, we have

$$R_{-m_x} x_n = R_V R_{\epsilon'_n} R_{-m_{x'}} x'_n, \ \forall n \in [N],$$

where $R_{\epsilon'_n} = R_{-m_{x'}-\epsilon} R_{\epsilon_n} R_{m_{x'}}$. If translation noises are sufficiently small, then $R_V R_{\epsilon'_n} \approx R_{V'}$ for a $V' \in \mathbb{O}(d)$.

4.3.2 Hyperbolic Rotation & Reflection

To estimate the unknown hyperbolic rotation, we consider minimizing a weighted discrepancy between the centered point sets. More precisely, we have

$$\widehat{U} = \underset{V\in\mathbb{O}(d)}{\arg\min} \sum_{n\in[N]} w_n f\Big(d\big(R_{-m_x} x_n, R_V R_{-m_{x'}} x'_n\big)\Big), \tag{4.2}$$

where $d(x, x') = \operatorname{acosh}(-x^\top H x')$, $\{w_n\}_{n\in[N]}$ are positive weights, and $f(\cdot) = \cosh(\cdot)$ is a monotonic function.

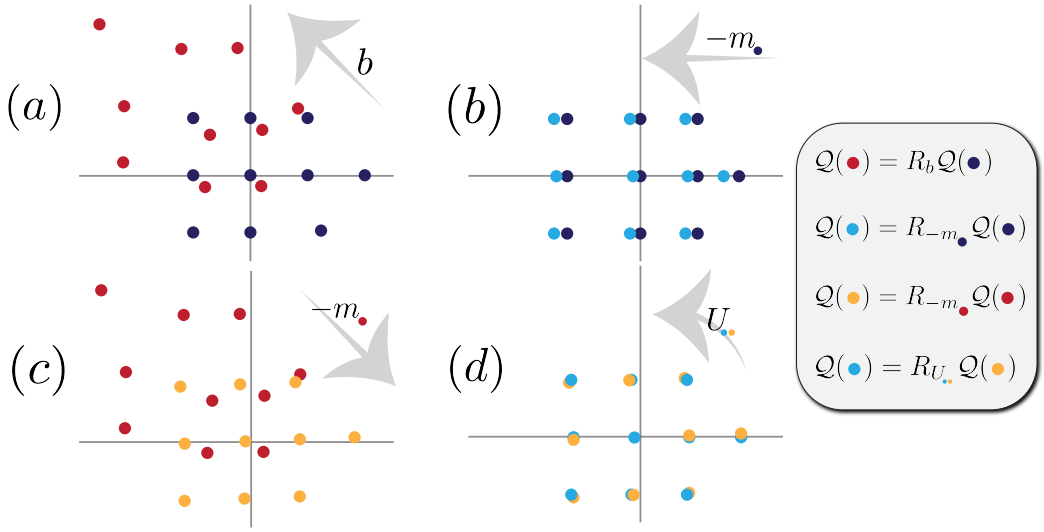


Figure 4.3: (a): Red and blue are projected points related by a translation, i.e., $X = R_b X'$. (b, c): Centering each point set. (d): Centered points are related via a rotation, i.e., $R_{m_x} R_b R_{-m_{x'}} \neq I_d$.

Proposition 9. *The optimal unitary matrix that solves* (4.2) *is* $\widehat{U} = U_l U_r^\top$, *where* $U_l \Sigma U_r^\top$ *is the singular value decomposition of* $\mathcal{P}(R_{-m_x} X) W \mathcal{P}(R_{-m_{x'}} X')^\top$ *and* $W = \text{diag}(w_1, \ldots, w_N)$.

4.3.3 Möbius Addition

In the Poincaré model, the points reside in the unit d-dimensional Euclidean ball. The isometry between the 'Loid and the Poincaré models $h : \mathbb{L}^d \to \mathbb{I}^d$ is called the *stereographic projection*. The distance between $y, y' \in \mathbb{I}^d$ is given by $d(y, y') = 2\text{tanh}^{-1}(\|-y \oplus y'\|)$ where $\oplus$ is Möbius addition — a non-commutative and non-associative operator. *Gyration* measures the deviation of Möbius addition from commutativity, i.e., $\text{gyr}[y, y'](y' \oplus y) = y \oplus y'$ [139].

Fact 3. *The maps* $h \circ R_U \circ h^{-1}$ *and* $h \circ T_U \circ h^{-1}$ *are the isometries in the Poincaré model, and can be written as*

$$h \circ T_U \circ h^{-1}(y) = Uy, \quad h \circ T_b \circ h^{-1}(y) = b' \oplus y,$$

where $b' = h \circ \mathcal{Q}(b)$.

The translation isometry is a result of Gyrotranslation theorem equality,

$$-(c \oplus y) \oplus c \oplus y^{'} = \text{gyr}[c, y](-y \oplus y'),$$

where $c \in \mathbb{I}^d$ [139]. Therefore, left Möbius addition preserves the distance of point sets in Poincaré model.[2] We can perform Procrustes analysis in Poincaré model by (1) centering each point set, i.e., subtracting their center mass from the left-hand side of Möbius addition, and (2) estimating the remaining rotation factor — a composition of gyrations and the unknown rotation between the two point sets.

[2]Möbius gyrations keep invariant the norm that they inherit from $\mathbb{R}^d$, i.e., $\|\text{gyr}[c, y](-y \oplus y')\| = \|-y \oplus y'\|$ [139].

4.4 Numerical Analysis

Let $x_n = R^* R_{\epsilon_n} x'_n$, $\forall n \in [N]$ where R^* is an H-unitary matrix and $\epsilon_1, \ldots, \epsilon_N$ is the set of translation noise samples. We consider the following three methods to compute an isometry that best matches the point sets X, X':

- Hyperbolic Procrustes (P): This method solves the hyperbolic Procrustes analysis using our proposed approach and returns the H-unitary matrix R_P.

- Gradient descent (GD): This method solves the hyperbolic Procrustes analysis using a gradient descent approach and returns the H-unitary matrix R_{GD}. To compute this matrix, we define the normalized discrepancy between X and $\overline{X}$ as $e(X, \overline{X}) \stackrel{\text{def}}{=} \frac{1}{Nd} \sum_{n\in[N]} d(x_n, \overline{x}_n)$. We then initialize $R_{\text{GD}} = I_{d+1}$ and iterate over the following steps:

  1. $\widehat{b} = -\alpha \frac{\partial}{\partial b} e(X, R_b R_{\text{GD}} X')|_{b=0}$ for a small $\alpha > 0$.
  2. $\widehat{U} = \arg\max_{U\in\mathbb{O}(d)} \sum_{n\in[N]} [x_n, R_U R_{\widehat{b}} R_{\text{GD}} x'_n]$.
  3. Update $R_{\text{GD}} \leftarrow R_{\widehat{U}} R_{\widehat{b}} R_{\text{GD}}$.

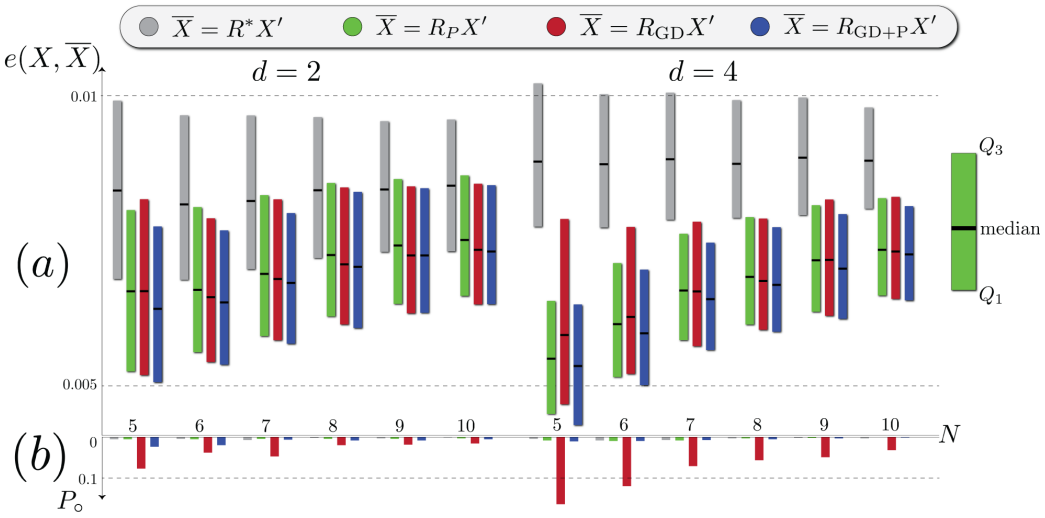


Figure 4.4: (a) Normalized discrepancy for random hyperbolic point sets of size $N \in \{5, \ldots, 10\}$ and dimensions $d \in \{2, 4\}$. For 10^3 trials, we report quartiles Q_1, Q_2 and Q_3 since they are robust to outliers. (b) The probability of an outlier event $P_\circ = 10^{-3} \times$ total number of outliers, e.g., failed to converge or outlier in the sense of (4.3).

- GD+P: This method iteratively applies both previous methods and returns the H-unitary matrix $R_{\text{GD+P}}$.

For a random H-unitary R^* and all $n \in [N]$, we sample d-dimensional $z_n \sim \mathcal{N}(0, I)$ and $\epsilon_n \sim 10^{-2}\mathcal{N}(0, I)$. Then, we let $x'_n = \mathcal{Q}(z_n)$ and $x_n = R^* R_{\epsilon_n} x'_n$. For 10^3 random (X, X') pairs, we compute their normalized discrepancy $e(X, RX')$, where $R \in \{R_P, R_{\text{GD}}, R_{\text{GD+P}}\}$. All methods successfully denoise the measurements, i.e., $e(X, R^*X') > e(X, RX')$; see Figure 4.4 (a).

However, the gradient descent method does not always converge to an acceptable solution. We define an outlier trial as follows:

$$(X, X') : |e(X, RX') - Q_2| > \frac{k}{2}|Q_3 - Q_1|, \tag{4.3}$$

where Q_1, Q_2, and Q_3 are the first, second, and third quartiles of the total reported discrepancies. We choose $k = 5$ for a conservative criterion to detect outliers (see Figure 4.4 (b)). The gradient descent method has the highest number of outliers (unstable solutions). On the opposite end, our proposed method has the minimum number of outliers — comparable to the number of outliers in the measurement noise. Therefore, the proposed close-form algorithm provides stable solutions for the hyperbolic Procrustes problem. Also, for noisy point sets — after removing the outlier trials — the accuracy of our proposed method is comparable to that of the gradient-based method and can be moderately improved with the post fine-tuning method.

4.5 Conclusion

Inspired by Euclidean counterpart, we posed the Procrustes problem in hyperbolic space. Using the parameterized decomposition of hyperbolic isometries in terms of hyperbolic rotation and translation, we showed that moving the center mass to the origin gives point sets that are invariant to hyperbolic translation (in cases with no measurement noise). This allows us to use the centered point sets to estimate the unknown rotation factor.

CHAPTER 5

LINEAR CLASSIFIERS IN PRODUCT SPACE FORMS

5.1 Introduction

Many practical datasets lie in Euclidean spaces and are thus naturally represented and processed using Euclidean geometry. Nevertheless, non-Euclidean spaces have recently been shown to provide significantly improved representations compared to Euclidean spaces for various data structures [18] and measurement modalities (e.g., metric and nonmetric) [19, 20]. Examples include *hyperbolic spaces,* suitable for representing hierarchical data associated with trees [21, 22], human-interpretable images [25], and olfactory data [26]; as well as *spherical spaces,* which are well-suited for capturing similarities in text embeddings and cycle-structures in graphs [27, 28]. Other important developments in non-Euclidean representation learning are methods for finding "good" mixed-curvature or hyperbolic representations for various types of complex heterogeneous datasets [28]. All three spaces considered — hyperbolic, Euclidean, and spherical — have *constant curvatures* but differ in their curvature sign (negative, zero and positive, respectively).

Despite these recent advances in nontraditional data spaces, almost all accompanying learning approaches have focused on (heuristic) designs of neural networks in constant curvature spaces [39, 40, 41, 42, 43, 44, 45]. The fundamental building block of these neural networks, the perceptron, has received little attention outside the domain of learning in Euclidean spaces.

Here, we address for the first time the problem of designing linear classifiers for product space forms (and generally, for geodesically complete Riemannian manifolds) with provable performance guarantees. Product space forms arise in a variety of applications in which graph-structured data captures both cycles and tree-like entities; examples of particular interest include social networks, such as the Facebook network for which product spaces reduce

the embedding distortion by more than 30% when compared to Euclidean or hyperbolic spaces alone [28]. An important property of such spaces is that they are endowed with logarithmic and exponential maps which play a crucial role in establishing rigorous performance results.

5.1.1 Related Work

Linear classifiers in spherical spaces have been studied in a number of works [140, 141]. More recent work has discussed linear classifiers in the Poincaré model of hyperbolic spaces, in the context of hyperbolic neural networks [40]. Specifically, linear classifiers (perceptrons and SVMs) in purely hyperbolic spaces has been studied in [142, 143]. However, simulation evidence and straightforward counterexamples show that the pure hyperbolic perceptron algorithm in [143] does not converge (see Appendix D for details). And although discussed within a limited context in [144, 39], classification in product spaces remains largely unexplored, especially from the theoretical aspect.

5.1.2 Contributions and Outline

We address the problem of linear classification in product space forms. In Sections 5.2 and 5.3, we describe the "point-line" formulation for linear classifiers in d-dimensional constant curvature spaces, e.g., Euclidean, hyperbolic and spherical spaces, using geodesics and Riemannian metrics which generalize straight lines and inner products in vector spaces. Also, we prove that linear classifiers in d-dimensional space forms of any curvature have the same expressive power, i.e., they can shatter exactly $d + 1$ points regardless of the curvature of the underlying space form.

Section 5.4 contains our main results, a description of our approach for generalizing linear classifiers in space forms to product spaces. The key idea behind our analysis is defining separation surfaces in constant curvature spaces directly through the use of geodesics on Riemannian manifolds (this definition matches the one proposed in [40, 44] for implementing hyperbolic neural networks); and, introducing metrics that render distances in different spaces compatible with each other and integrate them in linear classifiers with

majority signed distance criteria. We propose the corresponding perceptron and SVM classification algorithms and establish convergence results for the former. The proof techniques allow for generalizations to SVMs, discussed in Section 5.4.2.

In Section 5.5 and Appendix D, we demonstrate that our product space perceptron offers excellent performance on real-world datasets, such as the simple MNIST [145] and Omniglot [146] datasets, but also more complex structures such as CIFAR-100 [147] and single-cell expression measurements [148, 149, 150] which are paramount in computational biology, outperforming methods in Euclidean, spherical, or hyperbolic spaces which ignore the *hybrid* geometry of the data. The experimental results for synthetic and practical datasets such MNIST and Omniglot are delegated to Appendix D.

5.2 Linear Classifiers in Euclidean Space

Finite-dimensional Euclidean spaces are inner product vector spaces over the reals. In contrast, hyperbolic and (hyper)spherical spaces do not have the structure of a vector space. Therefore, we first have to clarify what linear classification means in spaces with nonzero curvatures. To introduce our approach, we begin by recasting the definition of Euclidean linear classifiers in terms of commonly used concepts in differential geometry such as geodesics and Riemannian metrics [151]. This will allow us to (1) present a unified view of the classification procedure in metric spaces that are not necessarily vector spaces; (2) formalize *distance-based* linear classifiers in space forms, i.e., classifiers that label data points based on their *signed distances* to the separation surface (Section 5.3); and (3) use the aforementioned classifiers as canonical building blocks for linear classifiers in product space forms (Section 5.4).

In a linear (more precisely, affine) binary classification problem we are given a set of N points in a Euclidean space and their binary labels, i.e., $(x_n, y_n) \in \mathbb{R}^d \times \{-1, 1\}$ for $n \in [N] \stackrel{\text{def}}{=} \{1, \dots, N\}$. The goal is to learn a linear classifier that produces the most accurate estimate of the labels. We

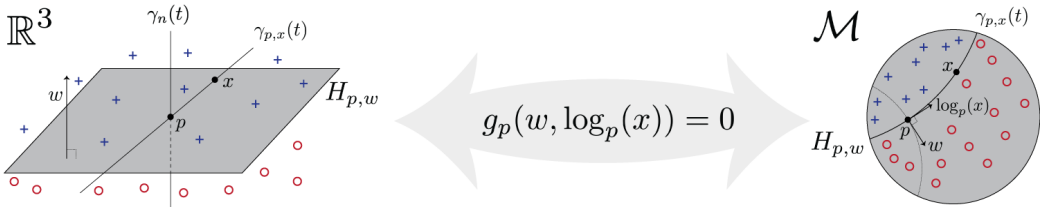


Figure 5.1: Linear classifiers in a three-dimensional Euclidean space (left) and on a manifold $\mathcal{M}$ (right). The location-varying metric in $\mathcal{M}$ causes the geodesics (shortest paths) to appear curved.

define a linear classifier with weight $w \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$ as

$$l^{\mathbb{E}}_{b,w}(x) = \operatorname{sgn}(w^\top x + b), \tag{5.1}$$

where $\|w\|_2 = 1$, and $l^{\mathbb{E}}_{b,w}(x)$ denotes the estimated label of $x \in \mathbb{R}^d$ for the given classifier parameters b, w. The expression (5.1) may be reformulated in terms of a "point-line" pair as follows: Let p be any point on the decision boundary and w a corresponding normal vector. Then, we have

$$l^{\mathbb{E}}_{b,w}(x) = \operatorname{sgn}(\langle w, x - p \rangle), \tag{5.2}$$

where $b = -p^\top w$ and $\langle \cdot, \cdot \rangle$ stands for the dot product. To see how this definition may be generalized, note that the linear classifier returns the sign of the inner product of tangent vectors of two straight lines, namely

$$\gamma_{p,x}(t) = (1-t)p + tx \quad \text{and } \gamma_n(t) = p + tw, \tag{5.3}$$

at their point of intersection $p \in \mathbb{R}^d$ (see Figure 5.1). Here, γ_n is the normal line and $\gamma_{p,x}$ is the line determined by p and the point x whose label we want to determine. These lines are smooth curves parameterized by $t \in [0, 1]$ (or an open interval in $\mathbb{R}$), which we interpret as *time*.

The linear classifier in (5.2) can be reformulated as

$$l^{\mathbb{E}}_{b,w}(x) = \operatorname{sgn}(\langle \frac{d}{dt}\gamma_{p,x}(t)|_{t=0}, \frac{d}{dt}\gamma_n(t)|_{t=0} \rangle),$$

where the derivative of a line $\gamma(t)$ at time 0 represents the *tangent vector* (or velocity) at the point $p = \gamma(0)$. This particular formulation leads to the following intuitive definition of linear classifiers in Euclidean spaces, which can be generalized for hyperbolic and spherical spaces.

Definition 9. *A linear classifier in Euclidean space returns the sign of the inner product between tangent vectors of two straight lines, described in* (5.3)*, at their unique meeting point.*

Often, we are interested in large-margin Euclidean linear classifiers for which we have $y_n\langle w, x_n - p\rangle \geq \varepsilon$, for all $n \in [N]$, and some margin $\varepsilon > 0$. For distance-based classifiers, we want ε to relate to the distance between the points x_n and the separation surface. For the classifier in (5.2), the distance between a point $x \in \mathbb{R}^d$ and the classification boundary, defined as $H_{p,w} = \big\{x \in \mathbb{R}^d : \langle w, x - p\rangle = 0\big\}$, can be computed as

$$\min_{y \in H_{p,w}} d(x, y) = |\langle w, x - p\rangle| = \big|w^\top x + b\big|.$$

Note that in the point-line definition (5.2), the point p can be anywhere on the decision boundary and it has d degrees of freedom whereas b from definition (5.1) is a scalar parameter. Therefore, we prefer definition (5.1) as it represents a distance-based Euclidean classifier with only $d + 1$ free parameters — w and b — and a norm constraint, $\langle w, w\rangle = 1$. In Section 5.3, we show that distance-based classifiers in d-dimensional space forms — of any constant curvature — can be defined with $d + 1$ free parameters and a norm constraint.

5.3 Linear Classifiers in Space Forms

A space form is a complete, simply connected Riemannian manifold of dimension $d \geq 2$ and constant curvature. Space forms are equivalent to spherical, Euclidean, or hyperbolic spaces up to an isomorphism [152]. To define linear classifiers in space forms, we first review basic concepts from differential geometry such as geodesics, tangent vectors and Riemannian metrics needed to generalize the key terms in Definition 9. For a detailed review, see [153, 151].

Let $\mathcal{M}$ be a Riemannian manifold and let $p \in \mathcal{M}$. The tangent space at the point p, denoted by $T_p\mathcal{M}$, is the collection of all tangent vectors at p. The Riemannian metric $g_p : T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$ is given by a positive-definite inner product in the tangent space $T_p\mathcal{M}$ which depends smoothly on the base point p. A Riemannian metric generalizes the notion of inner products for

Riemannian manifolds. The norm of a tangent vector $v \in T_p\mathcal{M}$ is given by $\|v\| = \sqrt{g_p(v,v)}$. The length of a smooth curve $\gamma : [0,1] \to \mathcal{M}$ (or path) can be computed as $L[\gamma] = \int_0^1 \|\gamma'(t)\|\, dt$. A geodesic γ_{p_1,p_2} on a manifold is the shortest-length smooth path between the points $p_1, p_2 \in \mathcal{M}$,

$$\gamma_{p_1,p_2} = \arg\min_{\gamma} L[\gamma] : \gamma(0) = p_1, \gamma(1) = p_2;$$

a geodesic generalizes the notion of a straight line in Euclidean space. Next, consider a geodesic $\gamma(t)$ starting at p and with initial velocity $v \in T_p\mathcal{M}$, e.g., $\gamma(0) = p$ and $\gamma'(0) = v$. The exponential map gives the position of this geodesic at $t = 1$, i.e., $\exp_p(v) = \gamma(1)$. Conversely, the logarithmic map is its inverse, i.e., $\log_p = \exp_p^{-1} : \mathcal{M} \to T_p\mathcal{M}$. In other words, for two points p and $x \in \mathcal{M}$, the logarithmic map $\log_p(x)$ gives the initial velocity (tangent vector) at which we can move — along the geodesic — from p to x in one time step.

In geodesically complete Riemannian manifolds, the exponential and logarithm maps are well-defined operators. Therefore, analogous to Definition 9, we can define a general notion of linear classifiers as described next.

Definition 10. *Let $(\mathcal{M}, g)$ be a geodesically complete Riemannian manifold, let $p \in \mathcal{M}$ and let $w \in T_p\mathcal{M}$ be a normal vector. A linear classifier $l_{p,w}$ over the manifold $\mathcal{M}$ is defined as*

$$l^{\mathcal{M}}_{p,w}(x) = \operatorname{sgn}\big(g_p(w, \log_p(x))\big), \text{ where } x \in \mathcal{M}.$$

Definition 10 is very general, but also has the following drawbacks: (1) The decision rule does not formalize a distance-based classifier since $\big|g_p(w, \log_p(x))\big|$ is not necessarily related to the distance of x to the decision boundary. (2) For a fixed $x \in \mathcal{M}$, the decision rule $g_p(w, \log_p(x))$ varies with the choice of p, which is an arbitrary point on the decision boundary. (3) Often, we can represent the decision boundary with other parameters that have a smaller number of degrees of freedom compared to that of w and p required by Definition 10 (see the Euclidean linear classifiers defined in (5.2) and (5.1)). We therefore next resolve these issues for linear classifiers in space forms.

5.3.1 Spherical Spaces

Let $p \in \mathbb{S}^d$ and $w \in T_p\mathbb{S}^d$ (see Table 5.1). The decision boundary is given by

$$H_{p,w} = \left\{ x \in \mathbb{S}^d : \langle w, \frac{\theta}{\sin(\theta)}(x - p\cos\theta)\rangle = 0, \theta = \operatorname{acos}(x^\top p) \right\}$$
$$\overset{(a)}{=} \left\{ x \in \mathbb{S}^d : w^\top x = 0 \right\} = \mathbb{S}^d \cap w^\perp, \tag{5.4}$$

where (a) is due to the fact that $w \in T_p\mathbb{S}^d = p^\perp$. This formulation uses two parameters $p \in \mathbb{S}^d$ and $w \in T_p\mathbb{S}^d$ to define the decision boundary (5.4). We note that one can actually characterize the *same* boundary with fewer parameters. Observe that for any $w \in \mathbb{R}^{d+1}$, we can pick an arbitrary base vector $p \in w^\perp \cap \mathbb{S}^d$ which ensures that $w \in T_p\mathbb{S}^d$. Therefore, without loss of generality, we can define the decision boundary using only one vector $w \in \mathbb{R}^{d+1}$, which has $d+1$ degrees of freedom. In Proposition 10, we identify a specific choice of $p \in w^\perp \cap \mathbb{S}^d$ that allows us to classify each data point based on its signed distance from the classification boundary.

Proposition 10. *Let $p \in \mathbb{S}^d$, $w \in T_p\mathbb{S}^d$, and $H_{p,w}$ be the decision boundary in (5.4). If $\langle w, w\rangle = 1$, then*

$$\forall x \in \mathbb{S}^d : \min_{y \in H_{p,w}} d(x,y) = \operatorname{asin}|w^\top x| = \left| g^{\mathbb{S}}_{p_\circ}(w, \log_{p_\circ}(x)) \right|,$$

where $g^{\mathbb{S}}$ is the Riemannian metric for a spherical space given in Table 5.1, and $p_\circ = \left\| P^\perp_w x \right\|^{-1} P^\perp_w x \in H_{p,w}$. Here, the projection operator is defined as $P^\perp_w x = x - \langle x, w\rangle w$.

It is important to point out that the classification boundary is invariant with respect to the choice of the base vectors, i.e., $H_{p,w} = H_{p_\circ,w}$. From

Table 5.1: Key properties of Euclidean ($\mathbb{R}^d$), spherical ($\mathbb{S}^d$), and hyperbolic ('Loid, $\mathbb{L}^d$) manifolds.

$\mathcal{M}$	$T_p\mathcal{M}$	$g_p(u,v)$	$\log_p(x) : \theta = d(x,p)$	$\exp_p(v)$	$d(x,p)$
$\mathbb{R}^d$	$\mathbb{R}^d$	$\langle u, v\rangle$	$x - p$	$p + v$	$\|x - p\|_2$
$\mathbb{S}^d$	$p^\perp$	$\langle u, v\rangle$	$\frac{\theta}{\sin(\theta)}(x - p\cos\theta)$	$\cos\|v\|\, p + \sin(\|v\|)\frac{v}{\|v\|}$	$\operatorname{acos}(\langle x, p\rangle)$
$\mathbb{L}^d$	$p^\perp$	$[u, v]$	$\frac{\theta}{\sinh(\theta)}(x - p\cosh\theta)$	$\cosh\|v\|\, p + \sinh(\|v\|)\frac{v}{\|v\|}$	$\operatorname{acosh}(-[x, p])$

Proposition 10, if we have

$$\forall n \in [N] : y_n \text{asin}\left(w^\top x_n\right) \geq \varepsilon,$$

then all data points are correctly classified and have the minimum distance of ε to the classification boundary. In summary, we can define distance-based linear classifiers in a spherical space as follows.

Definition 11. *Let $w \in \mathbb{R}^{d+1}$ with $\langle w, w\rangle = 1$. A spherical linear classifier is defined as*

$$l_w^{\mathbb{S}}(x) = \text{sgn}\big(\text{asin}(\langle w, x\rangle)\big).$$

5.3.2 Hyperbolic Spaces

The 'Loid model of a d-dimensional hyperbolic space [110] is a Riemannian manifold $\mathcal{L}^d = (\mathbb{L}^d, g^{\mathbb{H}})$ for which $\mathbb{L}^d = \big\{x \in \mathbb{R}^{d+1} : [x, x] = -1, x_1 > 0\big\}$, and $g_p^{\mathbb{H}}(u, v)$ corresponds to the Lorentzian inner product of u and $v \in T_p\mathbb{L}^d$, defined as

$$[u, v] = u^\top H v, \quad H = \begin{pmatrix} -1 & 0^\top \\ 0 & I_d \end{pmatrix}, \tag{5.5}$$

where I_d is the $d \times d$ identity matrix. Let $p \in \mathbb{L}^d$ and $w \in T_p\mathbb{L}^d$. The classification boundary of interest is given by

$$\begin{aligned} H_{p,w} &= \left\{ x \in \mathbb{L}^d : [w, \frac{\theta}{\sinh(\theta)}(x - p\cos\theta)] = 0 \right\} \\ &= \big\{x \in \mathbb{L}^d : [w, x] = 0\big\} = \mathbb{L}^d \cap w^\perp. \end{aligned} \tag{5.6}$$

Similar to the case of spherical spaces, we can simplify the formulation as follows. If w is a time-like vector — a vector that satisfies $w \in \{x : [x, x] > 0\}$ [151] — and $p \in \mathbb{L}^d \cap w^\perp$, then we have $w \in T_p\mathbb{L}^d$. In Proposition 11, we derive the expression for a special $p \in \mathbb{L}^d \cap w^\perp$ that allows us to formulate a distance-based hyperbolic linear classifier.

Proposition 11. *Let $p \in \mathbb{L}^d$, $w \in T_p\mathbb{L}^d$, and let $H_{p,w}$ be the decision boundary in* (5.6)*. If $[w, w] = 1$, then*

$$\min_{y \in H_{p,w}} d(x, y) = \operatorname{asinh}|[w, x]| = \left|g^{\mathbb{H}}_{p_\circ}(w, \log_{p_\circ}(x))\right|,$$

where $g^{\mathbb{H}}$ is the Riemannian metric for the hyperbolic space (given in Table 5.1), and $p_\circ = \left\|P^{\perp}_w x\right\|^{-1} P^{\perp}_w x \in H_{p,w}$. Note that $P^{\perp}_w x$ is the orthogonal projection of x onto $w^{\perp}$, i.e., $P^{\perp}_w x = x - [x, w]w$. Therefore, $p_\circ = \sqrt{\frac{1}{1+[x,w]^2}}(x - [x, w]w)$.

As a result, we have the following definition of distance-based linear classifiers in a hyperbolic spaces.

Definition 12. *Let $w \in \mathbb{R}^{d+1}$ with $[w, w] = 1$. A hyperbolic linear classifier is defined as*

$$l^{\mathbb{H}}_w(x) = \operatorname{sgn}\big(\operatorname{asinh}([w, x])\big).$$

From the previous discussion, we can deduce that *linear classifiers in d-dimensional space forms can be characterized with $d + 1$ free parameters and a norm constraint.* This supports the following result pertaining to the Vapnik-Chervonenkis (VC) dimension [154] of linear classifiers in space forms.

Theorem 1. *The VC dimension of a linear classifier in a d-dimensional space form is $d + 1$.*

Figure 5.2 illustrates linear classifiers in two-dimensional hyperbolic, Euclidean, spherical spaces and two product space forms. Next, we show how the first three classifiers — all of which have the same expressive power — can be "mixed" to define a linear classifier in product space forms.

5.4 Linear Classifiers in Product Space Forms

Definition 10 of linear classifiers applies to geodesically complete Riemannian manifolds. Our focus is linear classifiers in product space forms which are a special case of the aforementioned manifolds. We now describe a perceptron

Figure 5.2: Linear classifiers in Euclidean, spherical, hyperbolic spaces and product spaces. A hyperbolic space has dimension ≥ 2, but we reduced it to 1 for visualization purposes only.

algorithm for such spaces that provably learns an optimal classifier for linearly separable points in a finite number of iterations. Then, we extend this learning scheme to large-margin classifiers in product space forms.

Consider Euclidean, spherical, and hyperbolic manifolds, e.g., $(\mathbb{E}^{d_\mathbb{E}}, g^\mathbb{E})$, $(\mathbb{S}^{d_\mathbb{S}}, g^\mathbb{S})$, $(\mathbb{H}^{d_\mathbb{H}}, g^\mathbb{H})$ with sectional curvatures $0, C_\mathbb{S}, C_\mathbb{H}$, respectively (see Appendix D for detailed informations on space forms with arbitrary curvatures). The Euclidean manifold is simply $\mathbb{R}^{d_\mathbb{E}}$ while the hyperbolic space is the 'Loid model $\mathbb{L}^{d_\mathbb{H}}$.

The product manifold $\mathcal{M} = \mathbb{E}^{d_\mathbb{E}} \times \mathbb{S}^{d_\mathbb{S}} \times \mathbb{H}^{d_\mathbb{H}}$ admits a canonical Riemannian metric g, called the *product Riemannian metric*. The tangent space of $\mathcal{M}$ at a point $p = (p_\mathbb{E}, p_\mathbb{S}, p_\mathbb{H})$ can be decomposed as [155]

$$T_p\mathcal{M} = \bigoplus_{S \in \{\mathbb{E},\mathbb{S},\mathbb{H}\}} T_{p_S} S^{d_S}, \tag{5.7}$$

where the right-hand side expression is the direct sum $\bigoplus$ of individual tangent spaces $T_{p_\mathbb{E}}\mathbb{E}^{d_\mathbb{E}}$, $T_{p_\mathbb{S}}\mathbb{S}^{d_\mathbb{S}}$, and $T_{p_\mathbb{H}}\mathbb{H}^{d_\mathbb{H}}$. The *scaled* Riemannian metric used on $\mathcal{M}$ is

$$g_p(u, v) = \sum_{S \in \{\mathbb{E},\mathbb{S},\mathbb{H}\}} \alpha_S g^S_{p_S}(u_S, v_S), \tag{5.8}$$

where $u = (u_\mathbb{E}, u_\mathbb{S}, u_\mathbb{H}), v = (v_\mathbb{E}, v_\mathbb{S}, v_\mathbb{H}) \in T_p\mathcal{M}$, $p = (p_\mathbb{E}, p_\mathbb{S}, p_\mathbb{H})$, and $\alpha_\mathbb{E}$, $\alpha_\mathbb{S}$, $\alpha_\mathbb{H}$ are positive weights. The choice of the scaled Riemannian metric in Equation (5.8) — and hence the classification criteria — resolves the potential "distance compatibility" issues that arise from possibly vastly different ranges and variances of each component (e.g., $x_\mathbb{E}$, $x_\mathbb{S}$, and $x_\mathbb{H}$) which could lead to a classification criterion that is dominated by the component with the largest variance.

Based on our previous discussions, in order to describe linear classifiers on the above manifold $\mathcal{M}$, we first need to identify the logarithmic map (see Definition 10). For this purpose, we invoke the following known result

that formalizes geodesics, exponential and logarithmic maps on $\mathcal{M}$.

Fact 4. *[153] Let $\mathcal{M} = \mathbb{E}^{d_\mathbb{E}} \times \mathbb{S}^{d_\mathbb{S}} \times \mathbb{H}^{d_\mathbb{H}}$ with Riemannian metric given by (5.8). Then, the geodesics, exponential, and logarithmic maps on $\mathcal{M}$ are the concatenation of the corresponding maps of the individual space forms, i.e., $\gamma(t) = \big(\gamma_\mathbb{E}(t), \gamma_\mathbb{S}(t), \gamma_\mathbb{H}(t)\big)$, $\exp_p(v) = \big(\exp_{p_\mathbb{E}}(v_\mathbb{E}), \exp_{p_\mathbb{S}}(v_\mathbb{S}), \exp_{p_\mathbb{H}}(v_\mathbb{H})\big)$, and $\log_p(x) = \big(\log_{p_\mathbb{E}}(x_\mathbb{E}), \log_{p_\mathbb{S}}(x_\mathbb{S}), \log_{p_\mathbb{H}}(x_\mathbb{H})\big)$, where $p = (p_\mathbb{E}, p_\mathbb{S}, p_\mathbb{H})$, $x = (x_\mathbb{E}, x_\mathbb{S}, x_\mathbb{H}) \in \mathcal{M}$, $v = (v_\mathbb{E}, v_\mathbb{S}, v_\mathbb{H}) \in T_p\mathcal{M}$, and $\gamma_\mathbb{E}, \gamma_\mathbb{S}, \gamma_\mathbb{H}$ are geodesics in their corresponding space form.*[1]

Combining the results regarding distance-based linear classifiers in space forms (Section 5.3), the definition of tangent product spaces in terms of the product of tangent spaces in (5.7), and the choice of the Riemannian metrics given in Table 5.1, we arrive at the following formulation for a product space linear classifier. For detailed derivations, the reader is referred to Appendix D.

Proposition 12. *Let $\mathbb{S}^{d_\mathbb{S}}$ and $\mathbb{H}^{d_\mathbb{H}}$ be space forms with curvatures $C_\mathbb{S} > 0$, and $C_\mathbb{H} < 0$. Let $\mathcal{M} = \mathbb{E}^{d_\mathbb{E}} \times \mathbb{S}^{d_\mathbb{S}} \times \mathbb{H}^{d_\mathbb{H}}$ with the metric given by (5.8). The linear classifier on $\mathcal{M}$ is defined as*

$$l_w^\mathcal{M}(x) = \operatorname{sgn}\big(\langle w_\mathbb{E}, x_\mathbb{E}\rangle + \alpha_\mathbb{S}\operatorname{asin}(\langle w_\mathbb{S}, x_\mathbb{S}\rangle) + \alpha_\mathbb{H}\operatorname{asinh}([w_\mathbb{H}, x_\mathbb{H}]) + b\big),$$

where $w = (b, w_\mathbb{E}, w_\mathbb{S}, w_\mathbb{H})$, $w_\mathbb{E}, w_\mathbb{S}$, and $w_\mathbb{H}$ have norms of $\alpha_\mathbb{E}$, $\sqrt{C_\mathbb{S}}$, and $\sqrt{-C_\mathbb{H}}$, respectively.

This classifier can be associated with three linear classifiers, Euclidean, hyperbolic, and spherical space classifiers. For a point $x = (x_\mathbb{E}, x_\mathbb{S}, x_\mathbb{H}) \in \mathcal{M}$, the product space classifier takes a weighted vote based on the signed distances of each component (e.g, $x_\mathbb{E}$, $x_\mathbb{S}$, and $x_\mathbb{H}$) to its corresponding classifier's boundary. Figure 5.2 illustrates two classifiers in product space forms.

We now turn our attention to an algorithm for training linear classifiers in Proposition 12. To establish provable performance guarantees, we assume that the datasets satisfy the $\varepsilon > 0$ margin property, i.e.,

$$\forall (x, y) \in \mathcal{X} : y\big(w_\mathbb{E}^\top x_\mathbb{E} + b + \alpha_\mathbb{S}\operatorname{asin}(w_\mathbb{S}^\top x_\mathbb{S}) + \alpha_\mathbb{H}\operatorname{asinh}([w_\mathbb{H}, x_\mathbb{H}]\big) \geq \varepsilon, \quad (5.9)$$

[1]The distance between $x, y \in \mathcal{M}$ is given by $d(x, y) = \big(\sum_{S\in\{\mathbb{E},\mathbb{S},\mathbb{H}\}} \alpha_S^2 d_S(x_S, y_S)^2\big)^{\frac{1}{2}}$; see Table 5.1.

where $\mathcal{X}$ is the set of labeled training data, $\|w_{\mathbb{E}}\|_2 = \alpha_{\mathbb{E}}$, $\|w_{\mathbb{S}}\|_2 = \sqrt{C_{\mathbb{S}}}$, and $\sqrt{[w_{\mathbb{H}}, w_{\mathbb{H}}]} = \sqrt{-C_{\mathbb{H}}}$.

5.4.1 A Product Space Form Perceptron

The classification function is nonlinear in $w_{\mathbb{S}}$ and $w_{\mathbb{E}}$ and it requires equality constraints for all the weights involved. To analyze the classifier and allow for sequential updates of its parameters, we relax the norm constraints and propose perceptron updates in a Reproducing Kernel Hilbert Space (RKHS) which we denote by $\mathcal{H}$ [2]. We seek a map $\phi : \mathcal{M} \to \mathcal{H}$ to represent the classifier in (5.9) as an inner product of two vectors in $\mathcal{H}$, i.e., $l_w^{\mathcal{M}}(x) = \operatorname{sgn}\big(\langle \psi(w), \phi(x) \rangle_{\mathcal{H}}\big)$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product defined on $\mathcal{H}$.[3] The kernels $K_{\mathbb{E}}(w_{\mathbb{E}}, x_{\mathbb{E}}) = w_{\mathbb{E}}^\top x_{\mathbb{E}} + b$ and $K_{\mathbb{S}}(w_{\mathbb{S}}, x_{\mathbb{S}}) = \operatorname{asin}(w_{\mathbb{S}}^\top x_{\mathbb{S}})$ are symmetric and positive definite. Hence, they lend themselves to the construction of a valid RKHS. Unfortunately, $K_{\mathbb{H}}(w_{\mathbb{H}}, x_{\mathbb{H}}) = \operatorname{asinh}([w_{\mathbb{H}}, x_{\mathbb{H}}])$ is an indefinite kernel. To resolve this issue, we introduce an indefinite linear operator $M : \mathcal{H}' \to \mathcal{H}'$, where $\mathcal{H}' \supseteq \mathcal{H}$ is the set of functions $\mathcal{M} \to \mathbb{R}$ and $M^\top M = \mathrm{Id}$, with Id denoting the identity operator.[4] Therefore, we can write the classifier (5.9) as

$$l_w^{\mathcal{M}}(x) = \operatorname{sgn}\big(\langle \psi(w), M\phi(x) \rangle_{\mathcal{H}}\big), \tag{5.10}$$

where $\psi(w), \phi(x) \in \mathcal{H}'$, and $\langle \cdot, M\cdot \rangle_{\mathcal{H}}$ is also well-defined on $\mathcal{H}'$. This separable form allows us to formulate the update rule of the perceptron in $\mathcal{H}'$. Note that the decision rule (5.10) only depends on the inner products of vectors in $\mathcal{H}'$, i.e., kernel function evaluations for a point x and the misclassified training data points (see Algorithm 7). In Theorem 2, we prove that the product space perceptron in Algorithm 7 converges in a finite number of steps.

[2] This kernel approach is used to establish convergence results and is not a part of the algorithmic solution.

[3] We used $\psi(w)$ instead of $\phi(w)$ because the classification criteria might not be a symmetric function of x and w. In fact, we present the exact expression for $\psi(w)$, related to product space form classifiers, in Appendix D.

[4] The operator M can be obtained by analyzing the Taylor series of $\operatorname{asinh}(\cdot)$, as explained in Appendix D.

Theorem 2. *Let $\{x_n, y_n\}_{n=1}^N$ be points in a compact subset of $\mathcal{M}$ with labels in $\{-1, 1\}$, and $\|x_{\mathbb{H},n}\|_2 \leq R$ for all $n \in [N]$. If the point set is ε-margin linearly separable and $\|w_{\mathbb{H}}\|_2 \leq 1/R$, then Algorithm 7 converges in $O(\frac{1}{\varepsilon^2})$ steps.*

The norm constraint on $\|w_{\mathbb{H}}\|_2$ is imposed to ensure that the norm of $\psi(w)$ is bounded in $\mathcal{H}'$. This is necessary to establish the convergence bound for Algorithm 7.

Related Works and the Hyperbolic Perceptron

Linear classifiers in spherical spaces have been studied in a number of works [140, 141], while more recent work has focused on linear classifiers in the Poincaré model of hyperbolic spaces, in the context of hyperbolic neural networks [40]. A purely hyperbolic perceptron was described in [143]. Simulation evidence and some straightforward counterexamples show that the algorithm does not converge (see Appendix D for details). We therefore propose a modified update rule for a purely hyperbolic perceptron which is of independent interest given many emerging learning paradigms in hyperbolic spaces. Our hyperbolic perceptron uses a specialized update direction and provably converges, as described below (see more details in Appendix D).

Algorithm 7 A Product Space Form Perceptron.

Input: $\{x_n, y_n\}_{n=1}^N$: a set of pairs of point-labels in $\mathcal{M} \times \{-1, 1\}$.
Initialization: $k = 0$, $n = 1$, $x \stackrel{\text{def}}{=} (x_{\mathbb{E}}, x_{\mathbb{S}}, x_{\mathbb{H}})$, $f_1(x) = 0$.
repeat
  if $\operatorname{sgn}\big(f_k(x_n)\big) \neq y_n$ **then**
    $f_{k+1}(x) = f_k(x) + y_n\big(x_{\mathbb{E},n}^\top x_{\mathbb{E}} + 1 + \alpha_{\mathbb{S}}\operatorname{asin}(C_{\mathbb{S}} x_{\mathbb{S},n}^\top x_{\mathbb{S}}) + \alpha_{\mathbb{H}}\operatorname{asin}(\frac{\langle x_{\mathbb{H},n}, x_{\mathbb{H}}\rangle}{R^2})\big)$
    $k \leftarrow k + 1$
  end if
  $n \leftarrow \operatorname{mod}(n, N) + 1$
until A convergence criterion is met.

Theorem 3. *Let $\{x_n, y_n\}_{n=1}^N$ be a labeled point set from a bounded subset of $\mathbb{H}^{d_\mathbb{H}}$. Assume the point set is linearly separable by a margin ε. Then, the hyperbolic perceptron with the update rule $\operatorname{sgn}([w^k, x_n]) \neq y_n : w^{k+1} = w^k + y_n H x_n$, converges in $O\Big(\frac{1}{\sinh^2(\varepsilon)}\Big)$ steps.*

5.4.2 A Product Space Form SVM

In the previous section, we showed that the classification criterion for linear classifiers defined in Proposition 12 is a linear function of the *feature* vectors, or, more precisely, of $\{M\phi(x_n)\}_{n\in[N]}$. This fact and the subsequent performance guarantees are due to the update rule operating in the RKHS which, in effect, lifts a finite-dimensional point to a feature vector. Here, we use this analysis to formulate large-margin classifiers in product space forms. The idea behind our algorithm is to use the feature vector representation of linear classifiers, and maximize the distance between the points and the classification boundary. The closed-form expression of this distance is not available, but we can still provide an upper bound for this distance and perform maximum-margin classification. The described solution complements and extends the prior work on hyperbolic SVMs [142].

Let $x_1, \ldots, x_N \in \mathcal{M}$ be a fixed set of points. The *representer theorem* expresses the set of feasible parameters — in the space $\mathcal{H}'$ — as linear combinations of measured feature vectors.[5] In other words, any estimated parameter $\widehat{w}_N$ must satisfy the following condition:

$$\psi(\widehat{w}_N) \in \mathcal{L} = \{ \sum_{n\in[N]} \beta_n M\phi(x_n) : \sum_{n\in[N]} \beta_n^2 < \infty \}.$$

Let us assume that $\psi(w) = \sum_{n\in[N]} \beta_n M\phi(x_n)$. Then, the classification criterion is a linear function of $\beta = (\beta_1, \ldots, \beta_N)$, i.e.,

$$l_w^{\mathcal{M}}(x) = \operatorname{sgn}\Big(\sum_{n\in[N]} \beta_n \langle \phi(x), \phi(x_n) \rangle_{\mathcal{H}} \Big) = \operatorname{sgn}\Big(\sum_{n\in[N]} \beta_n k(x_n, x) \Big), \qquad (5.11)$$

where $k(x, x_n) = 1 + x_{\mathbb{E}}^\top x_{\mathbb{E},n} + \alpha_{\mathbb{S}} \operatorname{asin}(C_{\mathbb{S}} x_{\mathbb{S}}^\top x_{\mathbb{S},n}) + \alpha_{\mathbb{H}} \operatorname{asin}(R^{-2} x_{\mathbb{H}}^\top x_{\mathbb{H},n})$. In

[5] Feasible parameters are a subset of vectors in $\mathcal{H}'$ that can be used to define a proper distance-based linear classifiers in $\mathcal{M}$.

Algorithm 8 A Product Space Form SVM.

Input: $\{x_n, y_n\}_{n=1}^N$: a set of point-labels in $\mathcal{M} \times \{-1, 1\}$, and $r > 0$.
Let $\mathcal{B} = \big\{t : t^\top G_{\mathbb{E}} t < \alpha_{\mathbb{E}}^2,\ t^\top G_{\mathbb{S}} t < \frac{\pi}{2}, t^\top G_{\mathbb{H}}^- t \leq r, t^\top G_{\mathbb{H}}^+ t \leq r + \operatorname{asinh}(-R^2 C_{\mathbb{H}})\big\}$.
Solve for $\beta \in \mathcal{B}$:

$$\begin{aligned}
&\text{maximize} && \epsilon - \sum_{n\in[N]} \zeta_n \\
&\text{w.r.t} && \epsilon > 0, \{\zeta_n \geq 0\} \\
&\text{subject to} && \forall n \in [N]: \quad y_n \sum_{m\in[N]} \beta_m k(x_n, x_m) \geq \epsilon - \zeta_n
\end{aligned}$$

Algorithm 7, the weights are sequentially updated after each missclassification. Here, we directly optimize the weight vector β to ensure the maximum separability condition. In Proposition 13, we derive necessary conditions for the vector β that are conducive to a proper distance-based classifier.

Proposition 13. *For the classifier in* (5.11)*, we have the following equivalent conditions:*

- $\langle w_{\mathbb{E}}, w_{\mathbb{E}} \rangle = \alpha_{\mathbb{E}}^2 \Rightarrow \beta^\top G_{\mathbb{E}} \beta = \alpha_{\mathbb{E}}^2$
- $\langle w_{\mathbb{S}}, w_{\mathbb{S}} \rangle = C_{\mathbb{S}} \Rightarrow \beta^\top \operatorname{asin}\big[C_{\mathbb{S}} G_{\mathbb{S}}\big] \beta = \frac{\pi}{2}$
- $[w_{\mathbb{H}}, w_{\mathbb{H}}] = -C_{\mathbb{H}} \Rightarrow \beta^\top \operatorname{asinh}\big[R^{-2} G_{\mathbb{H}}\big] \beta = \operatorname{asinh}(-R^2 C_{\mathbb{H}})$,

where $\|x_{\mathbb{H},n}\| \leq R$ for all $n \in [N]$. The $N \times N$ matrices $G_{\mathbb{E}}$, $G_{\mathbb{S}}$, and $G_{\mathbb{H}}$ are Euclidean, spherical, and hyperbolic Gramians, respectively.

The constraints in Proposition 13, if they are met, define a product space form classifier in which the classification margin is *the sum (ℓ_1 norm) of the distances of the individual space components to the corresponding classifiers*, which is related to the weighted vote majority classification approach of Section 5.4.1. This distance is a proper upper bound (or a proxy) for the *true* distance of a point to the classification boundary which involves computing the ℓ_2 *norm* of the individual components' distances; see the footnote for Fact 4. To convexify the constraints in Proposition 13[6], we replace the Euclidean and spherical constraints with their convex hulls. The hyperbolic constraint, $[w_{\mathbb{H}}, w_{\mathbb{H}}] = -C_{\mathbb{H}}$, leads to a nonconvex second-order

[6]The set $\mathcal{A} = \big\{x : x^\top A x = 1\big\}$ is a non-convex set for any symmetric matrix A.

equality constraint on β. We let $G_{\mathbb{H}} = G_{\mathbb{H}}^{+} - G_{\mathbb{H}}^{-}$ for two positive semidefinite matrices $G_{\mathbb{H}}^{+}$ and $G_{\mathbb{H}}^{-}$. Then, we relax this second-order condition to $\beta^{\top} G_{\mathbb{H}}^{-} \beta \leq r$ and $\beta^{\top} G_{\mathbb{H}}^{+} \beta \leq r + \operatorname{asinh}(-R^2 C_{\mathbb{H}})$ for a small $r > 0$, i.e., we have $-r < \beta^{\top} G_{\mathbb{H}} \beta < r + \operatorname{asinh}(-R^2 C_{\mathbb{H}})$. The Algorithm 8 is our proposed soft-margin SVM classifier, for points with noisy labels, in product space forms.

5.5 Numerical Experiments: Real-world Datasets

We illustrate the practical performance of our product space form classifiers – Algorithms 7 and 8 – on (1) CIFAR-100 [147] (100 classes of size 600 each) and two scRNA datasets, (2) Lymphoma/Healthy donor with a targeted set of genes (two classes with $13,410$ samples total), and (3) blood cells with 965 landmark genes [156] only (landmark genes can be used to infer the activities of all other genes, and in this case we had 10 classes with $94,655$ samples total) [148, 149, 150]. In Appendix D, we present the convergence of our classifiers on synthetic datasets, Omniglot, and MNIST datasets.

5.5.1 CIFAR-100 Dataset

We use the mixed-curvature VAEs algorithm [144] to embed the dataset into chosen (product) space forms. We perform binary classification for 100 randomly selected pairs of classes. To enable K-class classification, we use K binary classifiers that are independently trained on the same training set to separate each single class from the remaining classes. For each classifier, we transform the resulting prediction scores into probabilities via Platt's scaling technique [157]. The predicted labels are decided by maximum a posteriori criteria, using the probability of each class.

Perceptron: We split the data into 80% training and 20% test points. We allow all perceptron algorithms to go over the whole dataset only once. In Table 5.2, we report the mean accuracy results with confidence intervals derived from repeated trials (over 10 repeated trials).

In Figure 5.3, we show the performance of the ternary classifiers. This is obtained by randomly selecting 100 sets of three classes; each point in the figure corresponds to one such combination, and its coordinate value equals

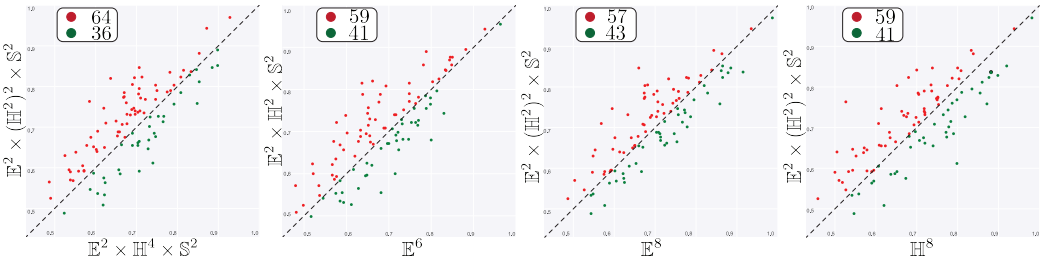


Figure 5.3: Classification accuracy of different product space form perceptrons on CIFAR-100 datasets. The labels on the x and y axes indicate the embedding spaces, and the counts in the top-left-corner indicate how often a binary classifier in one space outperforms that in another.

the averaged Macro F1 score of three independent runs. Red-colored points indicate better performance of the product space form perceptron specified on the y-axis, while green-colored points indicate better performance for the spaces specified on the x-axis.

SVM: To train a product space form SVM, we relax the optimization problem by removing the hyperbolic constraints to improve the run time of the method; see Algorithm 8.[7] We only use 100 training samples, and reserve the remaining samples for testing. Then, we compute the mean accuracy and the confidence levels for all selected class pairs and repeated trials; see Table 5.2. For simplicity, we let $\alpha_{\mathbb{E}} = \alpha_{\mathbb{S}} = \alpha_{\mathbb{H}} = 1$ in our implementation. Hyperbolic SVM is adopted from previous work [142].

Table 5.2: Classification mean accuracy (%) $\pm$ 95% confidence interval for perceptron (P) and SVM (S) algorithms in a product space $(d_{\mathbb{E}}, d_{\mathbb{H}}, d_{\mathbb{S}})$ on CIFAR-100.

$(d_{\mathbb{E}}, d_{\mathbb{H}}, d_{\mathbb{S}})$	$(2,2,2)$	$(6,0,0)$	$(2,2^2,2)$	$(2,4,2)$	$(8,0,0)$	$(0,8,0)$
(P)	70.26 ± 1.34	68.58 ± 1.29	$\mathbf{71.23 \pm 1.28}$	69.93 ± 1.25	69.96 ± 1.29	69.90 ± 1.43
(S)	69.51 ± 0.66	$\mathbf{75.69 \pm 0.58}$	73.05 ± 0.62	62.02 ± 0.10	74.53 ± 0.56	70.65 ± 0.93

[7] In Algorithm 8, the hyperbolic constraints are $t^\top G_{\mathbb{H}}^- t \leq r$ and $t^\top G_{\mathbb{H}}^+ t \leq r + \operatorname{asinh}(-R^2 C_{\mathbb{H}})$.

5.5.2 Single-cell RNA Datasets

Our first dataset, Lymphoma/Healthy donor with targeted set of genes, contains binary labeled samples. The second dataset, blood cells with landmark genes, contains 10 different labels. We use the standard experimental setups, similar to the ones for CIFAR-100 dataset, to conduct various classification experiments. In Table 5.3, we report the mean accuracy results, of our product space form perceptron and SVM algorithms, with confidence intervals derived from repeated trials (over 10 repeated trials).

The results across different datasets, learning methods, and embedding signatures (i.e., choices of dimensions of the components in the product spaces) suggest that product spaces can offer better low-dimensional representations for complex data structures, especially for scRNA sequencing data; see Figure 5.3 and Table 5.3. Generally, a higher-dimensional signature should lead to a better classification accuracy. However, the performance of the classification method dependents on the quality of discriminative features extracted from the mixed-curvature VAEs. This algorithm is not guaranteed to improve the embedding quality with increased dimensions when hyperparameters are fixed. Furthermore, finding a signature that allows for near-optimal embedding distortion is a hard problem that requires a sophisticated analysis of the geometry of datasets, and is thus beyond the scope of this work. The improvements in classification accuracy appear modest $\sim 2\%$.

5.6 Conclusion

We proposed linear classifiers for product space forms with provable performance guarantees. We showed how the "point-line" formulation for a

Table 5.3: Classification mean accuracy (%) $\pm$ 95% confidence interval for perceptron (P) and SVM (S) algorithms in a product space $(d_{\mathbb{E}}, d_{\mathbb{H}}, d_{\mathbb{S}})$. Datasets are Lymphoma (LMPH), and Blood-cells-landmark (BCL).

$(d_{\mathbb{E}}, d_{\mathbb{H}}, d_{\mathbb{S}})$	$(2,2,2)$	$(6,0,0)$	$(2,2^2,2)$	$(2,4,2)$	$(8,0,0)$	$(0,8,0)$
(P)-LMPH	$\mathbf{94.33 \pm 1.89}$	59.16 ± 11.56	46.66 ± 9.1	44.69 ± 9.01	75.42 ± 11.97	59.16 ± 8.51
(P)- BCL	70.79 ± 6.26	66.03 ± 6.83	65.01 ± 3.84	62.41 ± 3.89	$\mathbf{72.33 \pm 5.79}$	66.85 ± 6.30
(S)-LMPH	$\mathbf{94.48 \pm 1.31}$	70.61 ± 1.59	50.68 ± 7.03	43.9 ± 0.004	91.44 ± 2.38	65.83 ± 3.42
(S)-BCL	83.17 ± 5.42	74.61 ± 5.39	57.62 ± 8.15	74.18 ± 5.19	$\mathbf{89.89 \pm 7.09}$	77.75 ± 8.15

linear classifier in d-dimensional space forms can be simplified to a distance-based classifier. For linear classifiers in product space forms, we used an additive Riemannian metric that renders distances in different space forms compatible with each other. This formulation let us develop product space form perceptron and SVM. The perceptron algorithm comes with provable performance guarantees established via the use of indefinite kernels and their Taylor series. Our theoretical findings are supported with experimental results on several datasets, including synthetic data, CIFAR-100, MNIST, Omniglot, and single-cell RNA sequencing data. The results show that learning methods applied to low-dimensional embeddings in product space forms outperform their algorithmic counterparts in each space form.

CHAPTER 6

GEOMETRY OF SIMILARITY MEASUREMENTS

6.1 Introduction

Distances reveal the geometry of their underlying space. They are at the core of many machine learning algorithms. In particular, finding a geometrical representation for point sets based on pairwise distances is the subject of distance geometry problems (DGPs). Euclidean DGPs have a rich history of applications in robotics [6, 7], wireless sensor networks [8], molecular conformation analysis [9] and dimensionality reduction [10]. One is typically concerned with finding a geometric representation for a set of measured Euclidean distances [5]. Beyond Euclidean DGPs, recent works have focused of hyperbolic geometry methods in data analysis, most notably when dealing with hierarchical data. Social and FoodWeb networks [90, 158], gene ontologies [93], and Hearst graphs of hypernyms [37] are interesting examples of hierarchical datasets. Spherical embeddings represent sets of points on a (hyper)sphere [159], and have found applications in astronomy [160], distance problems on Earth [29], and texture mapping [30]. Euclidean, spherical and hyperbolic geometries are categorical examples of constant curvature spaces, or space forms, which are characterized by their curvature and dimension. The above examples represent instances of metric embeddings in space forms, as opposed to what is termed *nonmetric embeddings*. In the latter setting, one is provided with nonmetric information about data points, such as quantized distances or ordinal measurements such as comparisons or rankings.

We argue that nonmetric information such as *distance comparisons* carries valuable information about the space the data points originated from. To formally state our claims, assume that we are given a set of points $x_1, \ldots, x_N$ in an unknown metric space S. In nonmetric embedding problems [51, 52],

we work with dissimilarity (similarity) measurements of the form

$$\forall m, n \in [N] : y_{m,n} = \phi\big(d(x_m, x_n)\big),$$

where $d(x_m, x_n)$ is the distance between x_m and x_n in S, and $\phi(\cdot)$ is an unknown monotonically increasing (or decreasing) function. Since ϕ is unknown, we can *only* interpret the measurements as distance comparisons or ordinal measurements, i.e., if the entities indexed by n_1, n_2 are more similar than those indexed by n_3, n_4, then

$$y_{n_1,n_2} \leq y_{n_3,n_4} \Leftrightarrow d(x_{n_1}, x_{n_2}) \leq d(x_{n_3}, x_{n_4}).$$

We hence ask: *What do distance/similarity comparisons as those described above reveal about the space S?* Our work shows that one can use ordinal measurements to deduce the sign of the curvature and a lower-bound for the dimension of the underlying space form (in Euclidean and spherical spaces).

6.1.1 Related Work

In many applications we seek a representation for a group of entities based on their distances, but the exact magnitudes of the distances may be unavailable. What often *is* available (and prevalent) in applied sciences are nonmetric – dissimilarity or similarity – measurements: In neural coding [46], developmental biology [32], learning from perceptual data [47], and cognitive psychology [48]. Unfortunately, the datasets used in most of these studies are small (often involving fewer than 100 entities) and have limited utility for learning tasks that require sufficiently large sample complexity.

Nonmetric embedding problems originate from the works of Shepard [49, 50] and Kruskal [51]. Inspired by the Shepard-Kruskal scaling problem, Agarwal *et al.* [52] introduce generalized nonmetric multidimensional scaling, a semidefinite relaxation used to embed dissimilarity (or similarity) ratings of a set of entities in Euclidean space. Stochastic triplet embeddings [108] and crowd kernel learning [109] are used to embed triadic comparisons using probabilistic information. Tabaghi and Dokmanić [19] propose a semidefinite relaxation for metric and nonmetric embedding problems in hyperbolic space. In all these scenarios, the embedding space has to properly represent the measured data.

For example, in developmental biology and cancer genomics, single-cell RNA sequencing (scRNAseq) is used to differentiate cell types and cycles. The classification results have important implications for lineage identification and monitoring cell trajectories and dynamic cellular processes [31]. Klimovskaia *et al.* [32] use hyperbolic rather than Euclidean spaces for low-distortion embedding of complex cell trajectories (hierarchical structures).

Learning from distance comparisons is an active area of research. Among the relevant research topics are ranking objects from pairwise comparisons [161, 162], theoretical analysis of necessary number of distance comparisons to uniquely determine the embedding [163], nearest neighbor search [164], random forests [165], and classification based on triplet comparisons [166]. Understanding the underlying geometry of ordinal measurements is important in designing relevant algorithms.

Related to nonmetric embedding problems are the various techniques that study topological properties of point clouds independently of the choice of metric and of the geometric properties such as curvature [53]. An important problem in this domain is to detect intrinsic structure in neural firing patterns, invariant under nonlinear monotone transformations of measurements. Giusti *et al.* [46] propose a method based on clique topology of the graph of correlations between pairs of neurons. The clique topology of a weighted graph describes the behavior of cycles in its order complex [46] as a function of edge densities; these entities are also known as *Betti curves*. The statistical behavior of Betti curves is used to distinguish random and geometric structures of moderate sizes in Euclidean space. The more recent work of Zhou *et al.* [26] generalizes this statistical approach to hyperbolic spaces. These two works are the most closely related contributions to our proposed problem area. Nevertheless, the technical approaches used in there and in our work are fundamentally different. First, we provide a theoretical foundation for the study of geometric properties of space forms using similarity comparisons and derive the first known rigorous results related to their dimensions and curvatures. Second, we propose a computationally efficient method for inferring the sign of the curvature. The proposed statistical method can operate on large datasets as it uses subsampling techniques. Furthermore, we introduce new application areas in outlier identification, heterogeneity detection and imputation analysis for single-cell data measurements. To the best of our knowledge, we report the first study regarding the effect of

different imputation degrees on the geometry of similarity measurements in these datasets.

6.1.2 Contributions and Outline

The main results of our analysis are as follows:

1. We introduce the notion of *ordinal spread* of the sorted distance list, which is of fundamental importance in the study of the geometry of distance comparisons. The spread of ordinal measurements describes a pattern in which entities appear in the sorted list of distances, i.e., the ordinal spread gives the ranking of the first appearance of a data point in the list. This notion is related to another important geometric entity termed the *ordinal capacity*.

2. We define the notion of *ordinal capacity* of a space form to characterize the space's ability to host extreme patterns of ordinal spreads (computed from similarity measurements). We show that the ordinal capacity of a space form is related to its dimension and curvature sign. The ordinal capacity of Euclidean and spherical spaces are equal and grow exponentially with their dimensions, while the ordinal capacity of a hyperbolic space is infinite for any possible dimension of the space.

3. We derive a deterministic lower bound for Euclidean and spherical embedding dimensions using ordinal spreads and the (finite) ordinal capacity. We also associate an *ordinal spread random variable* with (1) a set of random points in a space form, and (2) a set of random vertex subsets from a similarity graph – a complete graph with edge weights corresponding to similarity scores of their defining nodes. The distributions of these random variables serves as a practical tool to identify the underlying space form given a similarity graph.

4. We illustrate the utility of our theoretical analysis by using them to correctly uncover the hyperbolicity of weighted trees. Moreover, we use them to detect Euclidean and spherical geometries for ordinal measurements derived from local and global cartographic data. Finally, we use the ordinal spread variables to determine the degree of heterogeneity of cell popula-

tions based on noisy scRNAseq data and how data imputation influences the geometry of the cell space trajectories.

6.2 The Ordinal Spread

Preliminaries. A space form is a complete, simply connected Riemannian manifold of dimension $d \geq 2$ and constant sectional curvature. Up to an isomorphism, space forms are equivalent to spherical ($\mathbb{S}^d$), Euclidean ($\mathbb{E}^d$), or hyperbolic spaces ($\mathbb{H}^d$) [167]. Distance geometry problems (DGPs) are concerned with finding an embedding for a set of pairwise measurements in a space form. DGP problems can be categorized as metric [19], nonmetric [52], or unlabeled [168, 169, 170], depending on the data modality and application domain. A nonmetric DGP aims to find $x_1, \ldots, x_N$ in a space form S, given a set of ordinal distance measurements $\mathcal{O} \subseteq [N]^4$ such that

$$\forall (n_1, n_2, n_3, n_4) \in \mathcal{O} : d(x_{n_1}, x_{n_2}) \leq d(x_{n_3}, x_{n_4}). \tag{6.1}$$

Although there exist theoretical results on the uniqueness of Euclidean embeddings [171] (up to an ordinal invariant transformation, i.e., an isotony), most often the underlying geometry of ordinal measurements is not known a priori [32, 172, 173].

We consider the problem of identifying the underlying space form from a given set of pairwise distance comparisons. For sufficiently many comparisons, this problem is equivalent to inferring geometrical information through the *sorted distance list* associated with ordinal measurements (6.1). A deterministic or a randomized binary sort algorithm needs at least $\Theta(\binom{N}{2} \log \binom{N}{2})$ pairwise comparisons to uniquely find the sorted distance list, if such a list exists [174]. Hence, we can define the *sorted index list* $(i_r, j_r)_{r \in \binom{N}{2}}$ according to

$$d(x_{i_1}, x_{j_1}) \geq \cdots \geq d(x_{i_{\binom{N}{2}}}, x_{j_{\binom{N}{2}}}), \tag{6.2}$$

where $i_r < j_r$ for all $r \in \left[\binom{N}{2}\right]$ and all pairs of indices are distinct. Any geometry-related inference problem must be *invariant* with respect to arbitrary permutations of the point indices. In particular, the pattern of the newly added indices in the sorted index list 6.2 is invariant to the permutations of point indices and has important geometrical implications. We formalize this

notion in Definition 13.

Definition 13. *The n-th ordinal spread of N points with a sorted index list is defined as*

$$\forall n \in [N] : \alpha_n = \min \left\{ m \in \mathbb{N} : \text{card} \bigcup_{r=1}^{m} \{i_r, j_r\} \geq n \right\}.$$

Alternatively, the ordinal spread α_n is the rank of the first appearance of the n-th point in the sorted index list, i.e.,

$$\text{card} \bigcup_{r=1}^{\alpha_n - 1} \{i_r, j_r\} < n, \ \text{card} \bigcup_{r=1}^{\alpha_n} \{i_r, j_r\} \geq n.$$

As an example, for $d(x_1, x_2) \geq d(x_1, x_3) \geq \cdots$, we have $\alpha_3 = 2$. From Definition 13, we observe that one can compute the ordinal spread α_n without knowing the point set positions, the distance and $\phi(\cdot)$ function or even the type of underlying space. For example, let $\{s_{m,n} = \phi(d(x_m, x_n)\}_{m,n \in [N]}$ be a set of pairwise similarities for a set of N points and $\phi(\cdot)$ be a strictly decreasing function. If $s_{n_1,n_2} \geq s_{n_3,n_4}$, then $(n_1, n_2, n_3, n_4) \in \mathcal{O}$. Nevertheless, in Section 6.2.1 and later on, we use $\alpha_n(\{x_n\}_{n=1}^N)$ to denote the n-th ordinal spread computed for the points $\{x_n\}_{n=1}^N$ in a metric space.

In general, the ordinal spreads $\{\alpha_n\}_{n \in [N]}$ depend on the configuration of the underlying point set, up to a similarity preserving map [171]. In Proposition 14, we make the first step in studying ordinal spread variables by computing their range of possible values.

Proposition 14. *For a set of $N \geq 4$ points with a given sorted index list, we have the following:*

- $\alpha_1 = \alpha_2 = 1$, $\alpha_3 = 2$.
- $4 \leq n \leq N : [\frac{n}{2}] \leq \alpha_n \leq \binom{n-1}{2} + 1$.

Clearly, the N-th ordinal spread variable, α_N, is the largest ordinal spread value which makes it a good choice for inferring geometry-related properties. In comparison, α_1, α_2, and α_3 are fixed and independent on the space and hence noninformative (see Appendix E for more details). We next provide two illustrative examples that show how the ordinal spread α_N may be used to reveal the hyperbolic, Euclidean, and spherical geometry of the measurements. These results motivate the study of ordinal capacity.

6.2.1 Hyperbolicity of Trees

Hyperbolic spaces are space forms that offer small distortion when embedding trees [24, 33]. Here, we describe how to verify this hyperbolicity by using *ordinal spread random variables*. We generate a random tree T with the vertex set $V = [10^4]$. The maximum node degree is 3 and edge weights are i.i.d. realizations of a unif$(0, 1)$-distributed random variable. Let $d_{m,n}$ be the distance between nodes m and n in V, defined as the sum of the weights on the unique path connecting the vertices. Then, we randomly subsample 10^6 different node subsets, or sub-cliques, of size $N \in \{10, 20\}$ ($N \ll |V| = 10^4$) from T as shown in Figure 6.1 (a). For each randomly selected sub-clique, we compute its N-th ordinal spread, α_N. Due to the inherently random nature of the clique selection process, α_N is a random variable which we term the ordinal spread random variable for the tree T. We can then compute the empirical distribution of the random variable, as illustrated in Figure 6.1(b_1, c_1). This motivates the following definition.

Definition 14. *Let S be a metric space, and P be a probability distribution on S. With a slight abuse of notation, we define the ordinal spread random variable α_N as*

$$\forall N \in \mathbb{N} : \alpha_N = \alpha_N(X),\ X \sim P^{\otimes N}.$$

An ordinal spread random variable is defined with respect to the distribution P. Let us assume an *oracle* picks a set of distributions for embedded points in each space form, e.g., (projected) normal for hyperbolic and Euclidean spaces, and uniform distribution in the spherical space. The distribution of the corresponding ordinal spread random variable α_N is invariant with respect to scaling. More precisely, it is invariant to strongly isotonic point transformations (more information in Appendix E). Then, we can compute the distribution of the random ordinal spread variable α_N for points generated in hyperbolic, Euclidean, and spherical spaces. As the results in Figure 6.1(b_1, c_1) indicate, the empirical distribution of α_N derived from a weighted tree T best matches (in the sense of total variation distance between the probability measures) with that of a random hyperbolic point set. For further verification, we repeated the same experiment for a random tree T with (1) additive measurements noise, e.g., $\tilde{d}_{m,n} = d_{m,n} + \eta$ where η is a sample of a zero-

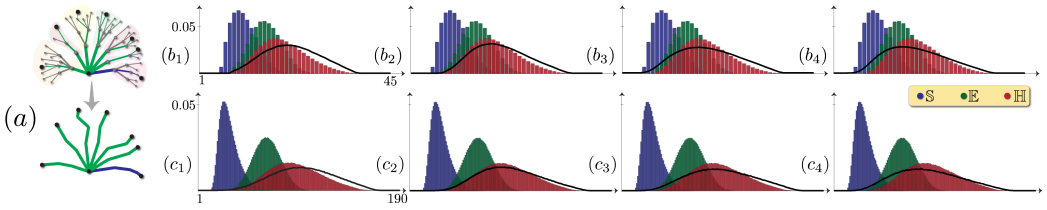


Figure 6.1: (a) Random selection of a sub-tree of size N. PMFs of α_{10} (top row) and α_{20} (bottom row) for random points in $\mathbb{H}^2$ (red), $\mathbb{E}^2$ (green), and $\mathbb{S}^2$ (blue). The black plots are empirical PMFs of α_N derived from (b_1, c_1) the noise-less tree T, (b_2, c_2) the additive noise contaminated tree, (b_3, c_3) the tree with permutation noise, and (b_4, c_4) a tree with both previous forms of noise.

mean Gaussian noise (with 20 decibel signal-to-noise ratio), (2) random permutation noise for the sorted index lists, e.g., $\tilde{i} = \pi(i)$ and $\tilde{j} = \pi(j)$ where π is a permutation with average displacement of $|V| = 10^4$, and (3) both additive and permutation noise; see Figure 6.1$(b_2, c_2), (b_3, c_3)$, and (b_4, c_4). The results clearly show that the distribution of the ordinal spread variable α_N is robust to noise and that it closely matches with that of a random hyperbolic point cloud. An important implication of this example is that ordinal spread variables can be used to determine the curvature sign of the underlying space. A more rigorous justification is provided in the subsequent exposition in Sections 6.3 and 6.4, where we formally connect the support of ordinal spread variables to a specific property of their underlying space forms, i.e., their *ordinal capacity*. In Appendix E, we show how to use ordinal capacity to compute a deterministic lower bound for the Euclidean embedding dimension of this tree.

6.2.2 Euclidean and Spherical Geometries of Cartographic Data

We describe next an experiment pertaining to the ordinal spread (random) variables of a similarity graph for geospatial data. The main idea is to use the distribution of these variables to show that the intrinsic geometry of small regions on the globe, which are "flat," is close to Euclidean, whereas that of large regions, which are spread across the globe, are close to spherical.

We use three datasets: (1) $1,627$ counties in the state of Illinois, (2) $11,954$ counties in Midwestern states, and (3) 10^4 (subsampled) cities and towns across the world; refer to Appendix E for details on data sources. We construct

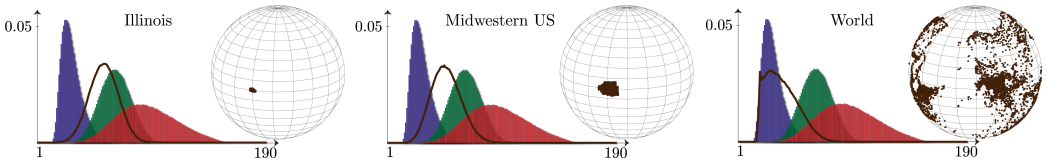


Figure 6.2: The empirical PMFs of α_{20} derived from subsampling the dissimilarity (distance) graph associated with points in the state of Illinois, across the Midwestern USA, and the world. Colored plots are PMFs of random points in $\mathbb{H}^2$ (red), $\mathbb{E}^2$ (green), and $\mathbb{S}^2$ (blue).

the dissimilarity graph by computing the pairwise distances between the points using the Haversine formula, which determines the great-circle distance between two points on the globe given their longitudes and latitudes [175]. For each dataset, we compute the empirical PMF of α_{20} from 10^6 randomly selected cliques of size 20 each; the results are shown in Figure 6.2. Comparing the PMFs for α_{20} and for random hyperbolic, Euclidean, and spherical points, we clearly observe the shift from an (approximately) Euclidean to a spherical geometry as the area spanned by the sampled points increases. We emphasize that these results are derived from distance comparisons only, since we discard the metric information in the distances.

6.3 The Ordinal Capacity

In the numerical experiment of Section 6.2.1, we discovered a distinguishing statistical behavior for the ordinal spread of randomly generated points in each possible space form. We show in what follows that this distinguishing pattern is related to the *capacity* of each space form to accommodate ordinal spread random variables with their underlying distributions. We define *ordinally dense sets* and show how they can help determine the support (the range of possible values) of the ordinal spread random variables in a space form.[1]

[1]We adopt Mirsky's notation $\{m, n\}_{\neq}$ for a set with two distinct elements m and n [176].

Definition 15. *Let $\{x_1, \ldots, x_N\}$ be a set of distinct points in a metric space S. If*

$$\exists n_0 \in [N] : \sup_{n \in [N] \setminus \{n_0\}} d(x_n, x_{n_0}) \leq \inf_{\{m,n\}_{\neq} \subseteq [N] \setminus \{n_0\}} d(x_m, x_n),$$

then we say that $\{x_n\}_{n=1}^N$ is an ordinally dense set in S, or in short $\{x_n\}_{n=1}^N \sqsubseteq S$.

In a nutshell, Definition 15 identifies point configurations that have a maximum possible ordinal spread. Intuitively, a set of N points is ordinally dense in S if and only if it has a subset of $N-1$ points whose pairwise distances are **all** larger than (or equal to) their distances to the N-th point, i.e.,

$$\{x_n\}_{n=1}^N \sqsubseteq S \iff \alpha_N\left(\{x_n\}_{n=1}^N\right) = \binom{N-1}{2} + 1. \tag{6.3}$$

The existence of an ordinally dense set of size N depends on the geometry of the underlying metric space, and is closely tied to what we term the *ordinal capacity* of the space (see Figure 6.3).

Definition 16. *The ordinal capacity of a metric space S is defined as*

$$K(S) = \sup\left\{\operatorname{card}\{x_n\} : \{x_n\} \sqsubseteq S\right\}.$$

The ordinal capacity is an indicator of the capability of a metric space to realize an extremal pattern of point indices in the sorted index list (6.3). In the next theorem, we show that the ordinal capacity of a space form is intimately related to a spherical cap packing problem [177], which is concerned with the maximum number of non-overlapping spherical caps (or domes with a certain polar angle) in a hypersphere.

Theorem 4. *Let N_d be the spherical $\frac{\pi}{6}$-cap packing number of $\mathbb{S}^d$. The ordinal capacity of a space form S is given by*

$$K(S) = \begin{cases} +\infty, & \text{if } S \cong \mathbb{H}^d \\ N_d + 1, & \text{if } S \cong \mathbb{E}^d, S \cong \mathbb{S}^d. \end{cases}$$

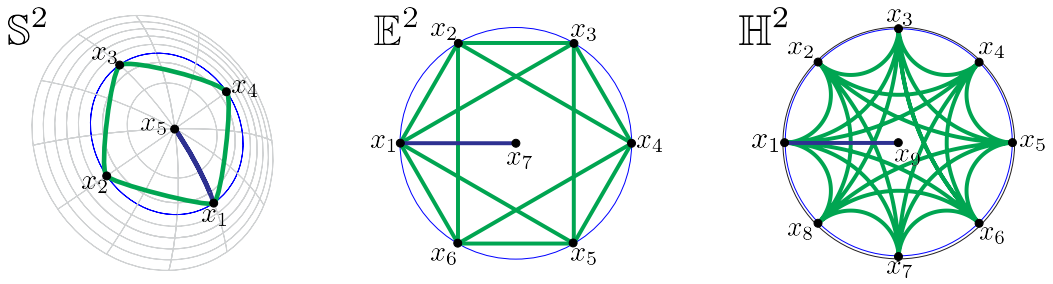


Figure 6.3: Ordinally dense point sets in two-dimensional space forms. As all distances in the (Euclidean) hexagon are greater than or equal to their distances to the center, the point set achieves the capacity $K(\mathbb{E}^2) = 7$.

Theorem 4 shows that the ordinal capacity of space forms depends on their curvature sign and dimension. The ordinal capacity of a hyperbolic space is infinite, regardless of its dimension. This implies that for any $N \in \mathbb{N}$, there exists an ordinally dense hyperbolic point set $\{x_n\}_{n=1}^N$. In the Poincaré model, a centered regular $(N-1)$-gon with an additional point in the "center" is an ordinally dense set (see Figure 6.3). In contrast, Euclidean and spherical spaces have *equal and finite* ordinal capacities. This finding is intuitively clear because any tangent space of $\mathbb{S}^d$ is a linear space of dimension d, and the spherical distance converges to the ℓ_2 distance as the distance between the points diminishes. In Appendix E, we propose a refinement for the ordinal capacity of spherical spaces by imposing a minimum distance constraint for the point sets. We note that the current notion of ordinal capacity does not distinguish between hyperbolic spaces of different dimensions. Therefore, one may need to develop a more refined notion of ordinal capacity for hyperbolic spaces, e.g., based on extremal appearance patterns of *multiple* indices in the distance lists.

Using the previous result, we can numerically compute an upper bound on of N_d, ρ_d, as a function of d, e.g. $\rho_1 = 2, \rho_2 = 6, \rho_3 = 15, \rho_4 = 31, \rho_5 = 59, \rho_6 = 106$ [177]. Note that the packing number N_d grows exponentially with the dimension d [178]. Hence, we have the following assymptotic bound for the ordinal capacity of a d-dimensional Euclidean (or spherical) space:

$$-\log\left(\frac{\sqrt{3}}{2}\right) + o(d) \leq \frac{1}{d}\log K(\mathbb{E}^d) \leq -\log\left(\frac{\sqrt{2}}{2}\right) + o(d).$$

Table 6.1: Numerical values for ρ_d.

d	1	2	3	4	5	6	7	8	9
ρ_d	2	6	15	31	59	106	183	308	507

6.4 The Support of Ordinal Spread Random Variables

In Section 6.2, we showed numerical evidence that ordinal spread random variables in Euclidean, spherical, and hyperbolic geometries have different supports. We therefore ask: *What is the maximum achievable ordinal spread, α_N, for a point set of size $N > K(S)$?* The answer to this question determines the support of ordinal spread random variables in Euclidean and spherical spaces, regardless of their underlying distribution P (see Definition 14). Note that since the ordinal capacity of a hyperbolic space is infinite, there always exists a point set of size N with maximum ordinal spread of $\binom{N-1}{2} + 1$ (see Proposition 14). For our subsequent analysis, we define the *N-point ordinal spread of a space form S* to be the maximum attainable ordinal spread α_N for the points in S. In Theorem 5, we express this quantity in terms of the ordinal capacity of S.

Theorem 5. *The N-point ordinal spread of a space form S is given by*

$$A_N(S) \overset{def}{=} \sup_{X \in S^N} \alpha_N(X) = E\big(T(N-1, K(S)-1)\big) + 1,$$

where $E\big(T(N,K)\big)$ is the number of edges of $T(N,K)$, the K-partite Turán graph [179] with N vertices.

As a conclusion, the N-point ordinal spread of a space form, i.e., the support of its ordinal spread random variable α_N, depends on its ordinal capacity and the number of points N. For a space S with finite ordinal capacity, there exists a point set $X \in S^N$ such that $\alpha_N(X) < \binom{N-1}{2} + 1$. This holds if $N > K(S)$. With this result, we can revise the ordinal spread bound in Proposition 14.

Proposition 15. *For a set of $N \geq 4$ points in a space form S, we have the following:*

- $\alpha_1 = \alpha_2 = 1$, $\alpha_3 = 2$.
- $4 \leq n \leq N : [\frac{n}{2}] \leq \alpha_n \leq A_n(S)$.

Theorem 5 and Proposition 15 explain in part the discriminatory ability of the support of ordinal spread random variables across different space forms. The N-point ordinal spread of a hyperbolic space $\mathbb{H}^d$ is the maximum value possible, i.e., $A_N(\mathbb{H}^d) = \binom{N-1}{2} + 1$, regardless of its dimension. Even though the N-point ordinal spread of Euclidean and spherical spaces, $\mathbb{E}^d$ and $\mathbb{S}^d$, varies with their dimension, they are equal to each other. This is evident from our distribution-free analysis of the ordinal capacity of these spaces (see Theorem 4 and its subsequent discussion). However, we can extend our distribution-free results to the following coarse lower bound for Euclidean (or spherical) embedding dimension of a similarity graph,

$$\min\left\{d : \sup_{X\subseteq V:|X|=N} \alpha_N(X) \leq A_N(\mathbb{E}^d), \forall N \in [|V|]\right\} \leq d,$$

where V is the vertex set of the graph. We may relax an exhaustive search over all 2^N vertex subsets, to a search over a random subselection of vertices. In Appendix E, we use such a relaxation to compute a lower bound for the embedding dimension of the tree discussed in Section 6.2.1.

6.4.1 Visualizing Point Sets with Maximum Ordinal Spread

Here, we aim to gain geometrical intuition about the point sets with maximum ordinal spread in different space forms. To this end, we generate independent and identically distributed point sets from a (projected) normal distribution in two-dimensional hyperbolic and Euclidean spaces. For each realization $\{x_n\}_{n=1}^N$, we compute the corresponding ordinal spread α_N. The maximum ordinal spread of the generated point sets, $\widehat{A}_N$, gives an estimate for $A_N(\mathbb{E}^2)$ and $A_N(\mathbb{H}^2)$ (see Theorem 5). We repeat this experiment for varying size of the point sets $N \in \{4, 5, \ldots, 13\}$.

For 5×10^5 realizations, we pick the point configurations with maximum ordinal spread; see Figure 6.4 (a, b). Recall that the point set with the theoretical maximum ordinal spread must have $N-1$ points sampled from a sphere centered at the N-th point. So, we repeat this experiment by fixing a point at 0, and projecting the remaining points to their circumscribed circle, i.e., $\forall n \in [N-1] : y_n = \frac{r}{\|x_n\|}x_n$, and $y_N = 0$, where $r = \max_{n\in[N-1]} \|x_n\|$. The randomly selected points $\{y_n\}_{n=1}^N$ produce a more accurate estimate for

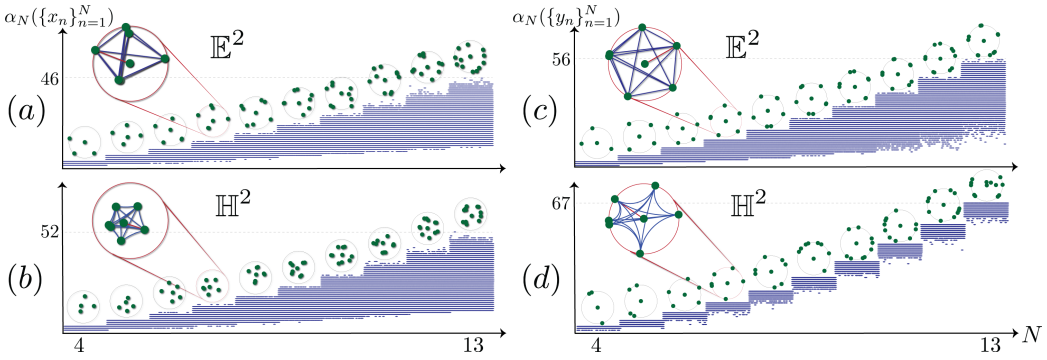


Figure 6.4: Ordinal spread of 5×10^5 i.i.d. point sets in $\mathbb{E}^2$ and $\mathbb{H}^2$. For fixed N, we mark the set with the maximum ordinal spread: $\{x_n\}_{n=1}^N$ in (a) and (b) and $\{y_n\}_{n=1}^N$ in (c) and (d).

$A_N(\mathbb{H}^2)$ and $A_N(\mathbb{E}^2)$; see Figure 6.4 (c, d). For example, we have $\widehat{A}_{13}(\mathbb{E}^2) = 56$, compared to the theoretical bound $A_{13}(\mathbb{E}^2) \leq 58$. Also, the estimated N-point ordinal spread of a hyperbolic space perfectly matches with the theoretical bound $A_N(\mathbb{H}^2) = \binom{N-1}{2} + 1$. The latter result is due to the capacity of hyperbolic spaces to host infinitely many ordinally dense point sets. Hence, the probability of randomly selecting an ordinally dense hyperbolic point set, of size N, is greater than its Euclidean counterpart.

Perhaps the most important observation is that the individual points in the extremal sets aggregate on nonoverlapping spherical caps of a circle, as seen in Figure 6.4 (c). The ordinal capacity of a space form equals the total number of such caps plus one (for the center point), i.e., $N_d + 1$. For example, there are five strictly non-overlapping spherical caps for two-dimensional Euclidean space, whereas this number is infinite for hyperbolic spaces. Finally, these results illustrate that the N-th ordinal spread of each space form, $A_N(S)$, is the total number of edges in Turán graphs (see Theorems 4 and 5).

6.5 Numerical Experiments: Single-cell RNA Sequencing Data

Here, we focus on results pertaining to an important new data format omnipresent in computational molecular biology: *single-cell RNA sequencing* (scRNAseq) data. By using recently developed single-cell isolation and barcoding techniques, and by trading individual cell coverage for the number of

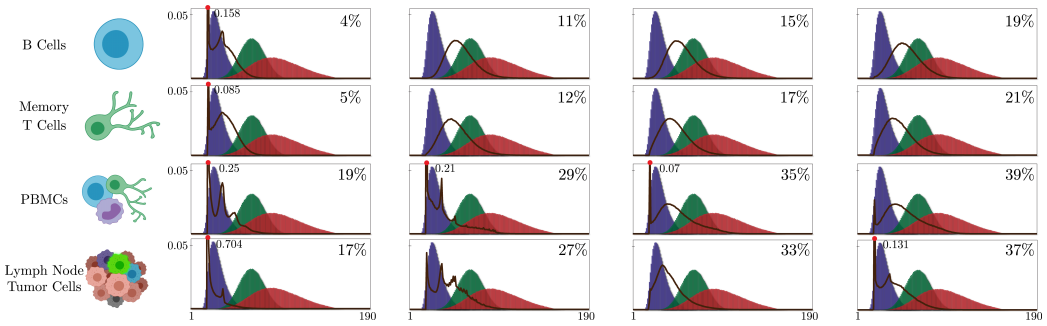


Figure 6.5: The empirical PMFs of α_{20} derived from subsampling the RFA similarity graph associated with scRNAseq data from homogeneous B cells ($\approx 10,000$ cells) and memory T cells ($\approx 10,000$), and heterogeneous PBMCs ($\approx 10,000$) and lymph node tumor cells ($\approx 3,000$). The left column shows the results for the raw data. From left to right, we increase the percentage of imputed data (densities are shown in the top-right corner).

cells captured, scRNA-seq data for the first time enables studying the activity patterns of millions of individual cells. This is in stark contrast to traditional bulk sequencing techniques that only provide *averaged snapshots* of cellular activity; scRNAseq measurements are also of special importance in cancer biology, as cancer cells are known to contain highly heterogeneous cell populations and the degree of heterogeneity carries significant information about disease progressions and the effectiveness of treatments [180]. Important for our study is the fact that due to the large number of different cells sequenced, cell measurements are extremely sparse and *imputed* in practice [181, 182, 183].

Further, it has been pointed out [183, 184] that scRNA-seq data is very noisy due to biological stochasticity as well as dropouts and systemic noise. Existing methods for denoising and imputation of raw scRNA-seq data often involve building connection graphs among cells [182, 181] using the distance between cells to diffuse the expression profiles among neighboring cells and smooth out possible outliers. Thus, *relative expression differences (comparisons)*, rather than *absolute expression values*, enable more accurate biological data mining via clustering, lineage detection, or inference of pseudotemporal orderings of cells [185]. As an example, [32] constructs similarity probabilities from a relative forest accessibility (RFA) matrix [186] and uses the obtained values to suggest that hyperbolic spaces are more suitable than Euclidean spaces for scRNAseq data embedding. We illustrate next that identifying the geometric properties of scRNAseq data using comparisons also provides

unique information about the diversity of cellular populations [180], outliers and the properties of imputation methods. Furthermore, since scRNA captures temporal hierarchical information about cells, as well as the cyclic nature of cell cycles, we expect spherical space forms to be equally useful as hyperbolic space forms in the process of embedding. To this end, we compute the empirical distribution of ordinal spread random variables associated with scRNA lymphoma (cancer) cells and cell *families* known as mononuclear cells (PBMCs), comprising T cells, B cells, and monocytes, which are often targeted in cancer immunotherapy. In this case, as illustrated by our numerical findings in Figure 6.5, these distributions contain peaks for small values that indicate that the data is sparse and contains outliers or highly heterogeneous cellular populations. Intuitively, probability peaks for small values of α_N arise when newly added indices in the ordered distance list appear in quick succession which can be attributed to one or multiple points at large distance from the remaining points (outliers); for more details see Appendix E. As imputation adds new data points by using averaged and smoothed information of observed measurements, it is expected to remove peaks in the aforementioned distributions, which is clearly the case for homogeneous cellular populations, but not for cancer cells and PBMCs. The reason why imputation does not remove peaks for the latter two categories can be attributed to the fact that the peaks arise due to the presence of many different cell types (e.g., recall that PBMCs contain B,T and monocytes and consequently, multiple peaks are observed in the ordinal spread distributions of raw data) which cannot and should not be smoothed out to form one class as this defeats the purpose of using single-cell measurements. Equally importantly, the results show that the Magic imputation software we used [181] imputes information into the noisy measurements without changing the geometry of the data, which is an important indicator of the quality of the procedure.

6.5.1 Nonmetric Embedding

Our next results pertain to the actual embedding quality of the measured similarities. We consider nonmetric embeddings [19, 52] of RFA scores of scRNAseq data from adult planarians [185]. The dataset contains $N \approx 26,000$ cells with gene expression vectors of dimension $d \approx 21,000$. In Figure 6.6

(a), we report the empirical probability of incorrect comparison $\mathbb{E}_N p_e$ for embedding RFA similarities in different space forms of varying dimensions. The results thus confirm that a *spherical* geometry is actually better suited for accurate nonmetric embeddings, which supports the frequently ignored understanding that cells are measured at various stages of the same cell cycle. For our analysis, we compute $\widehat{P}_{\alpha_{20}}$ from the similarity graph G. Due to the heavy-tailed nature of the original data distribution, we choose (oracle) log-normal distributions for the points in each space form. Then, we repeat the experiments for various dimensions and each space form/distribution parameters to find the closest ordinal spread variable to $\widehat{P}_{\alpha_{20}}$. From Figure 6.6 (b), we conclude that an ordinal spread variable from a high-dimensional ($\approx 1,000$) *spherical space* best matches $\widehat{P}_{\alpha_{20}}$.

6.6 Conclusion

This work offers a discussion about inferring the geometry of space forms from similarity comparisons between a set of entities. We introduce novel notions such as ordinal capacity and spread for metric spaces, as well as ordinally dense discrete sets. We provide theoretical and statistical analysis of ordinal spread variables. The proposed analysis, along with reasonable priors for the distribution of entities in a set of target spaces, can be used to identify the curvature sign of similarity graphs. This geometry driven approach for studying embedding spaces brings new perspective in designing algorithms related to similarity measurements.

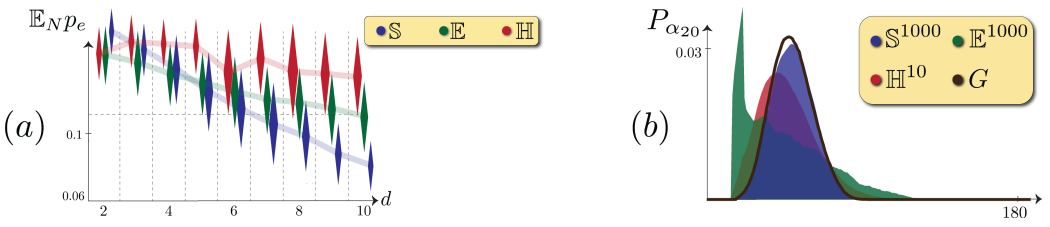


Figure 6.6: (a) PMFs of α_{20} from the RFA similarity graph (G) vs. random points in space forms of optimal dimensions. (b) $\mathbb{E}_N p_e$ for embedded points in d-dimensional space forms.

APPENDIX A

KINETIC EUCLIDEAN DISTANCE MATRICES

A.1 Spectral Factorization of the Time-varying Gramians

Let q stand for t for the polynomial model, or $e^{j\omega t}$ for the bandlimited model. Similarly, let $\mathcal{P} = \{0, \ldots, P\}$ for polynomial or $\mathcal{P} = \{-P, \ldots, P\}$ for bandlimited, and $\mathcal{P} + \mathcal{P} \stackrel{\text{def}}{=} \{p_1 + p_2 : p_1, p_2 \in \mathcal{P}\}$.[1]

Lemma 1. *Let $G(q) = \sum_{p \in \mathcal{P}+\mathcal{P}} B_p q^p$ (with $B_p \in \mathbb{C}^{N \times N}$) be rank-$d$ and positive semidefinite. Then there exists a unique (up to a $d \times d$ left unitary factor) $d \times N$ matrix $X(q) = \sum_{p \in \mathcal{P}} A_k q^p$ such that $G(q) = X(q)^H X(q)$.*

The statement has been proved for Laurent matrix polynomials in [84]. For $q = t$ it is equivalent to spectral factorization of polynomial matrices on the real line. Ephremidze [83] proved the full rank version; an entirely parallel construction to those in [83, 84] implies that it holds of rank-deficient matrices.

[1] A Laurent polynomial with coefficients in a field $\mathbb{F}$, is expressed as $x(z) = \sum_p c_p z^p$ where z is a formal variable and can have negative powers. Bandlimited trajectories are a special case of Laurent polynomials where $\mathbb{F} = \mathbb{C}^{d \times N}$ and $z = e^{j\omega}$.

A.2 Proof of Proposition 1

The Gramians can be written as linear combinations of a set of monomial terms (cf. (2.6)), which gives

$$\begin{aligned} G_0 &= B_0 + \tau_0 B_1 + \cdots + \tau_0^K B_K \\ &\vdots \\ G_K &= B_0 + \tau_K B_1 + \cdots + \tau_K^K B_K. \end{aligned} \tag{A.1}$$

Each matrix equation in (A.1) consists of $N \times N$ scalar equations for entries of G_k. Focusing on a particular entry (i, j) gives a linear system $g = Mb$ with column vector $g = [g_0, \cdots, g_K]^\top$ where g_k is the (i, j)-th element of G_k, the matrix $M \stackrel{\text{def}}{=} [\tau_k^{k'}]_{0 \le k, k' \le K}$, and $b = [b_0, \cdots, b_K]^\top$ where b_k is the (i, j)-th element of B_k. We also have from (2.6) that $[G(t)]_{ij} = (1, t, t^2, \ldots, t^K)b \stackrel{\text{def}}{=} t^\top b$. Since τ_k are distinct, the square Vandermonde matrix M is invertible. We have $b = M^{-1}g$ which gives $[G(t)]_{ij} = t^\top M^{-1} g$. Denoting $w(t) = (M^\top)^{-1} t$ we have that $[G(t)]_{ij} = w(t)^\top g = \sum_{k=0}^K w_k(t)[G_k]_{ij}$ which proves the claim.

A.3 Proof of Proposition 2

The proof is analogous to the polynomial case. We only need to show that the system matrix is full rank which is a standard result [187].

A.4 Proof of Proposition 3

(2) $\Rightarrow$ (1) is trivial. (1) $\Rightarrow$ (2): From $X \stackrel{\mathcal{D}}{\sim} Y$, by definition we have

$$\mathcal{K}(X(t)^\top X(t)) = \mathcal{K}(Y(t)^\top Y(t)), \forall t \in T.$$

Then, from (2.2) there is an orthogonal matrix $U(t)$ such that

$$JY(t) = U(t) JX(t)$$

for all $t \in T$. On the other hand, $X(t) = JX(t) + x(t)1^\top$, and $Y(t) = JY(t) + y(t)1^\top$ for $x(t), y(t) \in \mathbb{R}^d$. Finally, we have

$$\begin{aligned} Y(t) &= JY(t) + y(t)1^\top \\ &= U(t)JX(t) + y(t)1^\top \\ &= U(t)(X(t) - x(t)1^\top) + y(t)1^\top \\ &= U(t)X(t) + (y(t) - U(t)x(t))1^\top. \end{aligned}$$

A.5 Proof of Proposition 4

For polynomial trajectories, we prove that (1) is equivalent to (2) and (2) is equivalent to (3). We leave the straightforward extension to bandlimited trajectories to the reader.

It is obvious that (1) implies (2) and (2) implies (3). **(2) implies (1):** We have $X(t)1 = \sum_{p=0}^{P}(A_p 1)t^p = 0$. Since the monomials $\{t \mapsto t^p\}_{p=0}^{P}$ form a linearly independent set, the coefficients $A_p 1$ must all be zero. In other words, the column centroid of all A_p must be at the origin.
(3) implies (2): Since $G(t)1 = \sum_{k=0}^{K} w_k(t)G_k 1 = 0$, we have

$$\|X(t)1\|_2^2 = 1^\top X(t)^\top X(t)1 = 1^\top G(t)1 = 0.$$

Hence $X(t)1 = 0$ for all $t \in \mathbb{R}$.

A.6 Proof of Proposition 5

We prove this proposition by construction. Let us define

$$G_k^* = J_N X_\Theta(\tau_k)^\top X_\Theta(\tau_k) J_N$$

for $k \in \{0, \cdots, K\}$ and $G^*(t) = \sum_{k=0}^{K} w_k(t)G_k^*$. From Propositions 1 and 2, we deduce that $G^*(t) = J_N X_\Theta(t)^\top X_\Theta(t) J_N$. Hence, $G^*(t)$ belongs to the feasible set of (2.9) as $J_N X_\Theta(t)^\top X_\Theta(t) J_N$ is a zero-mean positive semidefinite matrix for all $t \in \mathbb{R}$, with rank at most d. On the other hand, since $\mathcal{D}(X_\Theta(t)) = \mathcal{K}(G^*(t))$, we have $J_2(G^*) = 0$. Finally, since (2.9) has a unique solution, the minimizer of (2.9) must have the form (2.10).

APPENDIX B

HYPERBOLIC DISTANCE MATRICES

B.1 Proof of Proposition 6

A hyperbolic Gramian can be written as $G = X^\top H X$ for a $X = [x_1, \dots, x_N] \in \left(\mathbb{L}^d\right)^N$. Let us rewrite it as

$$\begin{aligned} G &= \sum_{i=1}^{d} g_i g_i^\top - g_0 g_0^\top \\ &= G^+ - G^-, \end{aligned}$$

where $g_i^\top$ is the $(i+1)$-th row of X, $G^- = g_0 g_0^\top$ and $G^+ = \sum_{i=1}^d g_i g_i^\top$ are positive semidefinite matrices. We have $\operatorname{rank} G^- \leq 1$ and $\operatorname{rank} G^+ \leq d$. On the other hand, we have

$$\begin{aligned} e_i^\top G e_j &\stackrel{\text{def}}{=} [x_i, x_j] \\ &= -x_{0,i} x_{0,j} + \sum_{k=1}^{d} x_{k,i} x_{k,j} \\ &\stackrel{\text{(a)}}{=} -\sqrt{1 + \|\bar{x}_i\|^2}\sqrt{1 + \|\bar{x}_j\|^2} + \bar{x}_i^\top \bar{x}_j \\ &\stackrel{\text{(b)}}{\leq} -(1 + \bar{x}_i^\top \bar{x}_j) + \bar{x}_i^\top \bar{x}_j = -1, \end{aligned}$$

where $x_{k,i}$ is the $(k+1)$-th element of x_i, $\bar{x}_i = (x_{1,i}, \dots, x_{d,i})^\top$, and (a) is due to $\|x_i\|_H^2 = \|x_j\|_H^2 = -1$, and (b) results from Cauchy-Shwartz inequality. The equality holds for $i = j$, which yields the $\operatorname{diag} G = -1$ condition.

Conversely, let $G = G^+ - G^-$, where $G^+, G^- \succeq 0$, $\operatorname{rank} G^- \leq 1$, and $\operatorname{rank} G^+ \leq d$. Let us write $G^- = g_0 g_0^\top$ and $G^+ = \sum_{i=1}^d g_i g_i^\top$ for $g_0, \dots, g_d \in$

$\mathbb{R}^N$. Then, we define

$$X \stackrel{\text{def}}{=} \begin{bmatrix} g_0^\top \\ \vdots \\ g_d^\top \end{bmatrix} = [x_1, \cdots, x_N] \in \mathbb{R}^{(d+1)\times N},$$

where $x_n \in \mathbb{R}^{d+1}$ for all $n \in [N]$. By construction, we have $X^\top H X = G$ and

$$\operatorname{diag} G = -1 \Rightarrow \|x_n\|_H^2 = -1, \ \forall n \in [N].$$

Finally, the inequality $e_i^\top G e_j \leq -1$ guarantees that $x_n \in \mathbb{L}^d$ for all $n \in [N]$. We prove the contrapositive statement. Let x_i and x_j belong to different the hyperbolic sheets, e.g., $x_i \in \mathbb{L}^d, x_j \in -\mathbb{L}^d$. Then,

$$\begin{aligned} e_i^\top G e_j &\stackrel{\text{def}}{=} [x_i, x_j] \\ &= -x_{0,i}x_{0,j} + \sum_{k=1}^{d} x_{k,i}x_{k,j} \\ &\stackrel{\text{(a)}}{\geq} \sqrt{1 + \|\bar{x}_i\|^2}\sqrt{1 + \|\bar{x}_j\|^2} - \|\bar{x}_i\| \, \|\bar{x}_j\| \geq 0, \end{aligned}$$

where (a) is due to Cauchy-Shwartz inequality. This is in contradiction with $e_i^\top G e_j \leq -1$ condition. Therefore, $\{x_n\}$ belong to the same hyperbolic sheet, namely $\mathbb{L}^d$.

B.2 Derivations for Algorithm 6

Theorem 6. *Let $G \in \mathbb{R}^{N\times N}$ be a hyperbolic Gramian, with eigenvalue decomposition*

$$G = U^\top \Lambda U, \tag{B.1}$$

where $\Lambda = \operatorname{diag}(\lambda_0, \cdots, \lambda_{N-1})$ such that

- $\lambda_0 = \min_i \lambda_i$.
- *λ_i is the i-th top element of $\{\lambda_i\}$ for $i \in \{1, \cdots, d\}$.*

The best rank-$(d+1)$ Lorentz Gramian approximation of G, in ℓ_2 sense, is given by

$$G_{d+1} = U_d^\top \Lambda_d U_d,$$

where $\Lambda_d = \operatorname{diag}[\lambda_0, u(\lambda_1), \cdots, u(\lambda_d)]$, $u(x) = \max\{x, 0\}$, *and* $U_d \in \mathbb{R}^{(d+1)\times N}$ *is the corresponding sliced eigenvalue matrix.*

Proof. We begin by characterizing the eigenvalues of a Lorentz Gramian.

Lemma 2. *Let* $G \in \mathbb{R}^{N\times N}$ *be a Lorentz Gramian of rank* $d+1$ *with eigenvalues* $\psi_0 \leq \cdots \leq \psi_d$. *Then,* $\psi_0 < 0$, *and* $\psi_i > 0$, *for* $i \in \{1, \cdots, d\}$.

Proof. We write Lorentzian Gramian, $G = (g_{i,j})$, as $G = X^\top H X$ where

$$X = [x_1, \cdots, x_N] \stackrel{\text{def}}{=} \begin{bmatrix} g_0^\top \\ \vdots \\ g_d^\top \end{bmatrix} \in \mathbb{R}^{(d+1)\times N}.$$

Then, $G = G^+ - G^-$ where $G^+ \stackrel{\text{def}}{=} \sum_{i=1}^d g_i g_i^\top$ is a positive semi-definite matrix of rank d and with eigenvalues $0 < \gamma_1 \leq \cdots \leq \gamma_d$, and $-G^- \stackrel{\text{def}}{=} -g_0 g_0^\top$ is a negative definite matrix of rank 1, with eigenvalue $\mu \leq 0$. From Weyl's inequality [188], we have

$$\mu + \gamma_1 \leq \psi_0 \leq \mu + \gamma_d,$$

where ψ_0 is the smallest eigenvalue of G. Therefore, ψ_0 can be non-positive (negative if $\mu + \gamma_d < 0$). For other eigenvalues of G, we have

$$0 + \gamma_1 \leq \psi_i \leq \gamma_d, \text{ for } 1 \leq i \leq d.$$

Hence, $\psi_i > 0$ for $i \in \{1, \cdots, d\}$. This is result is irrespective to the order of eigenvalues.

Now, let us prove $\psi_0 < 0$. Suppose $g_0 \in S = \operatorname{span}\{g_i : i \in \{1, \cdots, d\}\}$. Then,

$$\operatorname{rank} G = \operatorname{rank} \begin{bmatrix} g_0^\top \\ \vdots \\ g_d^\top \end{bmatrix} < d+1,$$

which is a contradiction. Therefore, we write $g_0 = \alpha t + \beta s$ where $s \in S$,

$t \in S^\perp$ with $\|t\| = 1$, $\alpha, \beta \in \mathbb{R}$ and $\alpha \neq 0$. Then, we have

$$\begin{aligned}\psi_0 &\leq t^\top G t \\ &\overset{\text{(a)}}{=} -t^\top g_0 g_0^\top t \\ &= -\alpha^2 < 0,\end{aligned}$$

where (a) is due to $G = -g_0 g_0^\top + \sum_{i=1}^d g_i g_i^\top$ and $t \in S^\perp$. □

Consider eigenvalue decomposition of G in (B.1). Without loss of generality, we assume

- $\lambda_0 = \min_i \lambda_i < 0$.
- λ_i is the i-th top element of $\{\lambda_i\}$ for $i \in \{1, \cdots, d\}$.

By construction $G = X^\top H X$ and from $\operatorname{diag} G = -1$ condition, we have

$$\sum \lambda_i = -N.$$

Therefore, $\lambda_0 < 0$. From Lemma 2, one eigenvalue of a Lorentz Gramian is negative and the rest must be positive. Therefore, $\hat{G} = U_d^\top \Lambda_d U_d$ with eigenvalues $\Lambda_d = \operatorname{diag}\{\lambda_0, u(\lambda_1), \cdots, u(\lambda_d)\}$ and eigenvectors $U_d = [u_0, \cdots, u_d]$, is the best rank-$(d+1)$ Lorentz Gramian approximation to G, i.e.,

$$\left\|\hat{G} - G\right\|_2^2 = \inf_{H\text{: Lorentz Gram. of rank } \leq d+1} \|H - G\|_2^2 .$$

□

Finally, a rank-$(d+1)$ Lorentz Gramian with eigenvalue decomposition

$$G_{d+1} = U_d \Lambda_d U_d^\top$$

can be decomposed as $X = R|\Lambda|^{1/2} U_d^\top \in \mathbb{R}^{(d+1)\times N}$ where R is an arbitrary H-unitary matrix and $G_{d+1} = X^\top H X$.

B.3 The Projection Map — `Project` : $\mathbb{R}^d \to \mathbb{L}^d$

Algorithm 9 The projection map — `Project`(x) $\mathbb{R}^{d+1}$ to $\mathbb{L}^d$.

1: For $x \in \mathbb{R}^{d+1}$, let

$$\hat{x} = \begin{cases} (1, 0^\top)^\top & x \in \left\{(x_0, 0^\top)^\top : x_0 \leq 2\right\}. \\ (\frac{1}{2}x_0, \hat{x}_1, \cdots, \hat{x}_d)^\top & x \in \left\{(x_0, 0^\top)^\top : x_0 > 2\right\}. \\ x(\lambda^*) & \lambda^* : x(\lambda^*) \in S, \text{or } \|x(\lambda^*)\|_H^2 = -1. \end{cases}$$

Definitions: $x(\lambda) = (I + \lambda H)^{-1}x$ and

$$S = \left\{(x_0, x_1, \cdots, x_d) : x_1^2 + \cdots + x_d^2 = -1 + \frac{1}{4}x_0^2\right\}.$$

2: **return** $\hat{x}$.

Proof. Let us reformulate the following projection problem

$$\hat{x} \in \arg\min_{y \in \mathbb{L}^d} \|y - x\|^2 \tag{B.2}$$

as unconstrained augmented Lagrangian minimization, i.e.,

$$L(y, \lambda) = \|y - x\|^2 + \lambda(y^\top H y + 1).$$

The first-order necessary condition for $\hat{x}$ to be a (local) minimum of (B.2) is

$$(I + \lambda^* H)\hat{x} = x \tag{B.3}$$

for a $\lambda^* \in \mathbb{R}$ such that $\hat{x} \in \mathbb{L}^d$.

$\lambda^* = -1$: This happens when $x = (x_0, 0^\top)^\top$ and $x_0 \geq 2$. Following from optimality condition of (B.3) and $\|\hat{x}\|_H^2 = -1$, we have $\hat{x} = (\frac{1}{2}x_0, \hat{x}_1, \cdots, \hat{x}_d)^\top$, where

$$\hat{x}_1^2 + \cdots + \hat{x}_d^2 = -1 + \frac{1}{4}x_0^2.$$

Therefore, $\hat{x}$ could be any point on a $(d-1)$-dimensional sphere on $\mathbb{L}^d$. For $x = (x_0, 0^\top)^\top$ and $x_0 \leq 2$, we have $\hat{x} = (1, 0^\top)^\top$.

$\lambda^* = 1$: This happens for $x = (0, x_1, \cdots, x_d)^\top$. From optimality condition of (B.3), we have $\hat{x} = (\hat{x}_0, \frac{1}{2}x_1, \cdots, \frac{1}{2}x_d)$, where $\hat{x}_0 = \frac{1}{2}\sqrt{x_1^2 + \cdots + x_d^2 + 4}$.

For non-degenerate cases of $\lambda^* \neq \pm 1$, we have

$$\hat{x} = (I + \lambda^* H)^{-1} x, \tag{B.4}$$

where $\lambda^* \in \left\{\lambda : \|(I + \lambda H)^{-1} x\|_H^2 = -1, \hat{x}_0 \geq 0\right\}$.

(1) $\lambda^* \in (-1, 1)$: First, we define

$$f(\lambda) = \left\|(I + \lambda H)^{-1} x\right\|_H^2 .$$

This is a monotonous function on $(-1, 1)$, with $\lim_{\lambda \to 1^-} f(\lambda) = -\infty$, and $\lim_{\lambda \to -1^+} f(\lambda) = +\infty$. Hence, $f(\lambda) = -1$ has a unique solution $\lambda^* \in (-1, 1)$. Finally, $\hat{x}$ is a local minima since the second-order sufficient condition

$$I + \lambda^* H \succ 0$$

is satisfied for $\lambda^* \in (-1, 1)$. Lastly, from (B.4), we have $\hat{x}_0 x_0 \geq 0$. In other words, $\lambda^* \in [-1, 1]$ if and only if x is in the same half-space as $\mathbb{L}^d$, i.e., $x_0 \geq 0$.

(2) $\lambda^* \in (-\infty, -1)$: Similarly, $f(\lambda)$ is a continuous function in this interval with $\lim_{\lambda \to -1^-} f(\lambda) = +\infty$, $\lim_{\lambda \to -\infty} f(\lambda) = 0$, and its first-order derivative

$$\frac{d}{d\lambda} f(\lambda) = -\frac{2}{(1-\lambda)^3} x_0^2 - \frac{2}{(1+\lambda)^3} \sum_{i=1}^{d} x_i^2$$

has at most one zero. Therefore, $f(\lambda) = -1$ has at most two solutions. The second-order necessary condition for local minima is $v^\top (I + \lambda^* H) v \geq 0$ for all $v \in T_{\hat{x}} \mathbb{L}^d$, where

$$T_{\hat{x}} \mathbb{L}^d = \left\{v \in \mathbb{R}^{d+1} : x^\top (I + \lambda^* H)^{-1} H v = 0\right\}.$$

However, there exists a $v \in T_{\hat{x}} \mathbb{L}^d$ where $v = (0, \bar{v}^\top)^\top$ which violates the second-order necessary condition, $v^\top (I + \lambda^* H) v < 0$. Therefore, $\hat{x}$ – even if it exists – is not a local minima.

(3) $\lambda^* \in (1, \infty)$: We can easily see that $\lim_{\lambda \to 1^+} f(\lambda) = -\infty$, $\lim_{\lambda \to +\infty} f(\lambda) = 0$, and $\frac{d}{d\lambda} f(\lambda) = 0$ has at most one solution in this interval. Therefore, $f(\lambda) = -1$ has exactly one solution. However, we have $\hat{x}_0 x_0 \leq 0$ from (B.4). In other words, $\lambda^* \in (1, \infty)$ if and only if x is in the opposite half-space of $\mathbb{L}^d$, i.e., $x_0 \leq 0$. Finally, $\hat{x}$ is the unique minima, since the projection of $x \notin S$ to

the closed and convex set of

$$S = \left\{x : x_0 \geq 0, \|x\|_H^2 \leq -1\right\}$$

always exits and is unique. □

B.4 Proof Outline of Proposition 7

Let $X = R|\Lambda|^{1/2}U^\top$. Then,

$$\begin{aligned} X^\top H X &= U|\Lambda|^{1/2}R^\top H R|\Lambda|^{1/2}U^\top \\ &\overset{(a)}{=} U|\Lambda|^{1/2}H|\Lambda|^{1/2}U^\top \\ &\overset{(b)}{=} G, \end{aligned}$$

where (a) is due to properties of H-unitary matrices, (b) from $|\lambda_0|^{1/2}(-1)|\lambda_0|^{1/2} = \lambda_0$ for $\lambda_0 \leq 0$. Therefore $X = R|\Lambda|^{1/2}U^\top$ is a hyperbolic spectral factor of G. Finally, the uniqueness of these factors is due to fact that H-unitary operators fully characterize isometries in the 'Loid model [110].

APPENDIX C

HYPERBOLIC PROCRUSTES ANALYSIS

C.1 Proof of Proposition 8

Lemma 3 gives a simple method to center a projected point set.

Lemma 3. *[137] Let $x_1, x_2, \ldots, x_N \in \mathbb{L}^d$. Then, we have*

$$\overline{\mathcal{P}\big(R_{-m_x} x_n\big)} = 0,$$

where $m_x \overset{def}{=} \frac{1}{\sqrt{-[\overline{x_n}, \overline{x_n}]}}\ \overline{\mathcal{P}(x_n)}$.

From Lemma 3, we have

$$\overline{R_{-m_x} x_n} = \begin{bmatrix} a_1 \\ 0 \end{bmatrix}, \overline{R_{-m_{x'}} x'_n} = \begin{bmatrix} a_2 \\ 0 \end{bmatrix}$$

for $a_1, a_2 \in \mathbb{R}$. On the other hand, we can rewrite (4.1) in the following form

$$R_{-m_x} x_n = R' R_{-m_{x'}} x'_n, \forall n \in [N],$$

where $R' = R_{-m_x} R_b R_U R_{m_{x'}}$. Since R' is an H-unitary matrix, we can decompose it as $R' = R_c R_V$ for some $c \in \mathbb{R}^d$ and $V \in \mathbb{O}(d)$. Therefore, we have

$$\begin{bmatrix} a_1 \\ 0 \end{bmatrix} = R_c R_V \begin{bmatrix} a_2 \\ 0 \end{bmatrix}.$$

This gives $c = 0$.

C.2 Proof of Proposition 9

We can simplify (4.2) as follows:

$$\widehat{U} = \underset{V \in \mathbb{O}(d)}{\arg\max} \sum_{n \in [N]} \operatorname{Tr} R_{-m_{x'}} x'_n w_n (R_{-m_x} x_n)^\top H R_V.$$

From Fact 1, R_V is only parameterized on its lower-right block. The proof follows from matrix representation of the summation and von Neumann trace inequality [189].

APPENDIX D

LINEAR CLASSIFIERS IN PRODUCT SPACE FORMS

Space forms are Riemannian manifolds of dimension $d \geq 2$ that are isomorphic to spherical, Euclidean or hyperbolic spaces [152]. A d-dimensional spherical space with curvature $C > 0$ is a collection of points

$$\mathbb{S}^d = \left\{ x \in \mathbb{R}^{d+1} : \langle x, x \rangle = C^{-1} \right\},$$

where $\langle \cdot, \cdot \rangle$ is defined in the main text. Similarly, a d-dimensional hyperbolic space (i.e., the 'Loid model) with curvature $C < 0$ is a collection of points of the form

$$\mathbb{H}^d = \left\{ x \in \mathbb{R}^{d+1} : [x, x] = C^{-1} \right\},$$

where $[\cdot, \cdot]$ is defined in the main text. In Table D.1, we list the Riemannian metric, exponential and logarithmic maps for each of these spaces.

D.1 Proof of Proposition 10

Let $p \in \mathbb{S}^d$ and $w \in T_p\mathbb{S}^d = p^\perp$ such that $\langle w, w \rangle = C$. The separation surface $H_{p,w}$ is defined as

$$\begin{aligned} H_{p,w} &= \left\{ x \in \mathbb{S}^d : g_p(\log_p(x), w) = 0 \right\} \\ &= \left\{ x \in \mathbb{S}^d : \langle x, w \rangle = 0 \right\}. \end{aligned}$$

Table D.1: Summary of relevant operators in Euclidean, spherical, and hyperbolic ('Loid model) spaces with arbitrary curvatures.

$\mathcal{M}$	$T_p\mathcal{M}$	$g_p(u, v)$	$\log_p(x) : \theta = \sqrt{\lvert C \rvert} d(x, p)$	$\exp_p(v)$	$d(x, p)$
$\mathbb{R}^d$	$\mathbb{R}^d$	$\langle u, v \rangle$	$x - p$	$p + v$	$\lVert x - p \rVert_2$
$\mathbb{S}^d$	$p^\perp$	$\langle u, v \rangle$	$\frac{\theta}{\sin(\theta)}(x - p\cos\theta)$	$\cos(\sqrt{C}\lVert v \rVert)p + \sin(\sqrt{C}\lVert v \rVert)\frac{v}{\sqrt{C}\lVert v \rVert}$	$\frac{1}{\sqrt{C}}\operatorname{acos}(C\langle x, p \rangle)$
$\mathbb{L}^d$	$p^\perp$	$[u, v]$	$\frac{\theta}{\sinh(\theta)}(x - p\cosh\theta)$	$\cosh(\sqrt{-C}\lVert v \rVert)p + \sinh(\sqrt{-C}\lVert v \rVert)\frac{v}{\sqrt{-C}\lVert v \rVert}$	$\frac{1}{\sqrt{-C}}\operatorname{acosh}(C[x, p])$

We can compute the distance between $x \in \mathbb{S}^d$ and $H_{p,w}$ as

$$d(x, H_{p,w}) = \min_{y \in H_{p,w}} \frac{1}{\sqrt{C}} \text{acos}(Cy^\top x).$$

The projection of a point onto $H_{p,w}$ can be computed by solving the following constrained optimization problem

$$\max_{y \in \mathbb{R}^{d+1}} x^\top y \;\; \text{such that} \;\; w^\top y = 0, \; \langle y, y \rangle = C^{-1}.$$

From the first-order optimality condition for the Lagrangian, the projected point takes the form $\mathcal{P}(x) = \alpha x + \beta w$, where $\alpha, \beta \in \mathbb{R}$. Now, we impose the following subspace constraint,

$$\begin{aligned} w^\top \mathcal{P}(x) &= w^\top (\alpha x + \beta w) \\ &= \alpha w^\top x + \beta w^\top w \\ &= \alpha w^\top x + \beta C \\ &= 0, \end{aligned}$$

which gives $\beta = -\alpha C^{-1} x^\top w$. Subsequently, we have $\mathcal{P}(x) = \alpha(x - C^{-1} x^\top w w)$. On the other hand, from the norm constraint, we have

$$\begin{aligned} \|\mathcal{P}(x)\|^2 &= \alpha^2 (C^{-1} + C^{-1}(x^\top w)^2 - 2C^{-1}(x^\top w)^2) \\ &= \alpha^2 C^{-1}(1 - (x^\top w)^2) \\ &= C^{-1}, \end{aligned}$$

which gives $\alpha = (1 - (x^\top w)^2)^{-\frac{1}{2}}$. Then,

$$\mathcal{P}(x) = \sqrt{\frac{1}{1 - (x^\top w)^2}} (x - \frac{x^\top w}{w^\top w} w) = \frac{C^{-\frac{1}{2}}}{\|P_w^\perp x\|_2} P_w^\perp x, \tag{D.1}$$

where $P_w^\perp x = x - \frac{1}{\langle w, w \rangle} \langle x, w \rangle w$.

Next, let us define $\psi = \text{acos}(x^\top w)$, where $\psi \in [0, \pi]$. Then, the minimum

distance is given by

$$\begin{aligned} d(x, \mathcal{P}(x)) &= \frac{1}{\sqrt{C}} \operatorname{acos}\big(C x^\top \mathcal{P}(x)\big) \\ &= \frac{1}{\sqrt{C}} \operatorname{acos}\Big(\sqrt{\frac{1}{1-\cos^2\psi}}(1-\cos^2\psi)\Big) \\ &= \frac{1}{\sqrt{C}} \operatorname{acos}\big(|\sin\psi|\big) \\ &\overset{(a)}{=} \frac{1}{\sqrt{C}} \operatorname{asin}\big(|\cos\psi|\big) \\ &= \frac{1}{\sqrt{C}} \operatorname{asin}|x^\top w|, \end{aligned}$$

where (a) follows due to $\operatorname{acos}(|\sin(\psi)|) = \operatorname{asin}(|\cos(\psi)|)$, or

$$\begin{aligned} \cos\big(\operatorname{asin}(|\cos\psi|)\big) &= \cos\Big(\operatorname{asin}\Big|(\sin(\frac{\pi}{2}-\psi)\Big|\Big) \\ &= \cos\Big(\Big|\operatorname{asin}(\sin(\frac{\pi}{2}-\psi))\Big|\Big) \\ &= \cos(\Big|\frac{\pi}{2}-\psi\Big|) \\ &= |\sin\psi|, \end{aligned}$$

for $\psi \in [0, \pi]$. Now, let $x \in \mathbb{S}^d$, and let $\mathcal{P}(x)$ be as given in (D.1). We readily have $\mathcal{P}(x) \perp w$. Therefore,

$$\begin{aligned} w^\top \log_{\mathcal{P}(x)}(x) &= \frac{\operatorname{acos}(C\mathcal{P}(x)^\top x)}{\sin(\operatorname{acos}(C\mathcal{P}(x)^\top x))} x^\top w \\ &\overset{(a)}{=} \frac{\operatorname{acos}(C\mathcal{P}(x)^\top x)}{|x^\top w|} x^\top w \\ &= \operatorname{asin}(|x^\top w|)\operatorname{sgn}(x^\top w) \\ &= \operatorname{asin}(x^\top w) = \operatorname{sgn}(x^\top w)\sqrt{C} d(x, \mathcal{P}(x)), \end{aligned}$$

where (a) follows from

$$\begin{aligned} \sin\big(\operatorname{acos}(C\mathcal{P}(x)^\top x)\big) &= \sin\big(\operatorname{acos}(\sqrt{1-(x^\top w)^2})\big) \\ &= \sin\big(\operatorname{asin}(|x^\top w|)\big) \\ &= |x^\top w|. \end{aligned}$$

This completes the proof.

D.2 Proof of Proposition 11

Let $\mathbb{H}^d$ be the 'Loid model with curvature $C < 0$ (usually set to $C = -1$ for simplicity). The projection of $x \in \mathbb{H}^d$ onto $H_{p,w}$ is a point $\mathcal{P}(x) \in H_{p,w}$ that has the smallest distance to x. In other words, $\mathcal{P}(x)$ is the solution to the following constrained optimization problem

$$\max_y [y, x] \;\text{ such that } [y, y] = C^{-1}, [w, y] = 0,$$

where $[w, w] = -C$. The solution to this problem takes the form of $\mathcal{P}(x) = \alpha x + \beta w$, where $\alpha, \beta \in \mathbb{R}$. We can enforce the subspace condition as follows:

$$\begin{aligned}[\mathcal{P}(x), w] &= \alpha[x, w] + \beta[w, w] \\ &= \alpha[x, w] + \beta(-C) \\ &= 0,\end{aligned}$$

which gives $\beta = \alpha C^{-1}[x, w]$, or $\mathcal{P}(x) = \alpha(x + C^{-1}[x, w]w)$. On the other hand, we also have

$$\begin{aligned}[\mathcal{P}(x), \mathcal{P}(x)] &= \alpha^2(C^{-1} - C^{-1}[x, w]^2 + 2C^{-1}[x, w]^2) \\ &= \alpha^2 C^{-1}(1 + [x, w]^2) \\ &= C^{-1}.\end{aligned}$$

Then, we have

$$\mathcal{P}(x) = \sqrt{\frac{1}{1 + [x, w]^2}}(x + C^{-1}[x, w]w) = \frac{(-C)^{-\frac{1}{2}}}{\|P_w^{\perp} x\|} P_w^{\perp} x, \tag{D.2}$$

where $P_w^{\perp} x = x - \frac{1}{[w,w]}[x, w]w$ and $\left\|P_w^{\perp} x\right\| = \sqrt{-[P_w^{\perp} x, P_w^{\perp} x]}$. The minimum distance can be computed as

$$\begin{aligned}d(x, \mathcal{P}(x)) &= \frac{1}{\sqrt{-C}} \text{acosh}(C[\mathcal{P}(x), x]) \\ &= \frac{1}{\sqrt{-C}} \text{acosh}(C\sqrt{\frac{1}{1 + [x, w]^2}} C^{-1}(1 + [x, w]^2)) \\ &= \frac{1}{\sqrt{-C}} \text{acosh}(\sqrt{1 + [x, w]^2}).\end{aligned}$$

We can further simplify this expression to:[1]

$$d(x, \mathcal{P}(x)) = \frac{1}{\sqrt{-C}} \operatorname{asinh}|[x, w]|.$$

Now, let $x \in \mathbb{L}^d$ and let $\mathcal{P}(x)$ be given in (D.2). We can easily see that $[\mathcal{P}(x), w] = 0$. Therefore, we have

$$\begin{aligned} g_{\mathcal{P}(x)}(w, \log_{\mathcal{P}(x)}(x)) &= \frac{\operatorname{acosh}(C[\mathcal{P}(x), x])}{\sinh(\operatorname{acosh}(C[\mathcal{P}(x), x]))}[x, w] \\ &= \frac{\operatorname{asinh}(|[x, w]|)}{|[x, w]|}[x, w] \\ &= \operatorname{asinh}(|[x, w]|)\operatorname{sgn}([x, w]) \\ &= \operatorname{asinh}([x, w]) = \operatorname{sgn}([x, w])\sqrt{-C}d(x, \mathcal{P}(x)). \end{aligned}$$

This completes the proof.

D.3 Proof of Theorem 1

The Vapnik-Chervonenkis (VC) dimension [154] of a linear classifier is equal to the maximum size of a point set that a set of linear classifiers can *shatter*, i.e., completely partition into classes independent on how the point in the set are labelled. We establish the VC dimension for all three space forms $\mathcal{M} = \mathbb{R}^d, \mathbb{S}^d$, and $\mathbb{L}^d$ (clearly, the VC dimension of Euclidean space forms is well-known, as described below).

The Vapnik-Chervonenkis (VC) dimension of affine classifiers in $\mathbb{R}^d$ is $d + 1$ (see the treatment of VC dimensions of Dudley classes described in [190]). Therefore, there exists a set of $d + 1$ points that affine classifiers in $\mathbb{R}^d$ can shatter. Note again the distinction between affine and linear classifiers in Euclidean spaces.

Next, let $x_1, \ldots, x_N \in \mathbb{S}^d$ be a set of point in spherical space $\mathbb{S}^d$, which can be shattered by linear classifiers. In other words, we have

$$y_n = \operatorname{sgn}\big(\operatorname{asin}(w_{\mathbb{S}}^\top x_n)\big), \ \forall n \in [N],$$

and for any set of binary labels $(y_n)_{n\in[N]}$. The linear classifiers in spherical

[1] Since $\cosh(x)^2 - \sinh(x)^2 = 1$.

space are a subset of linear classifiers in a $(d+1)-$dimensional Euclidean space. Hence, their VC dimension must be less than or equal to $d+1$. On the other hand, if we project a set of $d+1$ points in $\mathbb{R}^{d+1}$ onto $\mathbb{S}^d$, that can be shattered by linear classifiers in Euclidean space (by a simple normalization). This way, we can find a set of (exactly) $d+1$ points that can be shattered by linear classifiers in $\mathbb{S}^d$. Hence, the VC dimension of linear classifiers in $\mathbb{S}^d$ is exactly $d+1$.

Next, let us turn our attention to d-dimensional hyperbolic spaces, namely the 'Loid model. Let $\mathcal{X} = \{x_n\}_{n\in[d+1]}$ be a set of $d+1$ points in d-dimensional 'Loid model of hyperbolic space such that

$$x_n = \begin{bmatrix} \sqrt{1 + \|z_n\|^2} \\ z_n, \end{bmatrix}$$

for $z_n \in \mathbb{R}^d$ and all $n \in [d+1]$. Furthermore, we assume that $z_1 = 0$, and $z_n = e_{n-1}$ for $n \in \{2, \ldots, d+1\}$, where e_n is the n-th standard basis vector of $\mathbb{R}^d$.

We claim that this point set can be shattered by the set of linear classifiers in hyperbolic spaces, i.e.,

$$l_w^{\mathbb{H}}(x) = \operatorname{sgn}\big(\operatorname{asinh}([w, x])\big) \tag{D.3}$$

where $w \in \big\{x \in \mathbb{R}^{d+1} : [x, x] > 0\big\}$. Let $(y_1, \ldots, y_{d+1})$ be an arbitrary set of labels in $\{-1, 1\}$. Then, we define

$$\forall n \in \{2, \ldots, d+1\} : t_1 = y_1,\ t_n = k y_n, \tag{D.4}$$

where $k > \sqrt{2} + 1$. Therefore, if we can show that there exists a $w \in \big\{x \in \mathbb{R}^{d+1} : [x, x] > 0\big\}$ such that

$$\forall n \in [d+1] : t_n = [w, x_n],$$

then we have $y_n = l_w^{\mathbb{H}}(x_n)$ for all $n \in [d+1]$. This is equivalent to showing that the following equation has a solution $w \in \{x : [x, x] > 0\}$,

$$t = X^\top H w,$$

where $t = (t_1, \ldots, t_{d+1})$ and

$$X^\top = \begin{bmatrix} \sqrt{1 + \|z_1\|^2} & z_1^\top \\ \sqrt{1 + \|z_2\|^2} & z_2^\top \\ \vdots & \vdots \\ \sqrt{1 + \|z_{d+1}\|^2} & z_{d+1}^\top \end{bmatrix} = \begin{bmatrix} 1 & 0^\top \\ \sqrt{2} & e_1^\top \\ \vdots & \vdots \\ \sqrt{2} & e_d^\top \end{bmatrix}.$$

The solution is $w = H(X^\top)^{-1}t$, described below,

$$w = H \begin{bmatrix} 1 & 0^\top \\ -\sqrt{2} & e_1^\top \\ \vdots & \vdots \\ -\sqrt{2} & e_d^\top \end{bmatrix} t = \begin{bmatrix} -t_1 \\ -\sqrt{2}t_1 + t_2 \\ \vdots \\ -\sqrt{2}t_1 + t_{d+1} \end{bmatrix}.$$

As the final step, we show that $w \in \{x : [x, x] > 0\}$. To this end we observe that

$$\begin{aligned} [w, w] &= -t_1^2 + \sum_{n=2}^{d+1} (-\sqrt{2}t_1 + t_n)^2 \\ &\overset{\text{(a)}}{=} y_1^2 \big(-1 + \sum_{n=2}^{d+1} (-\sqrt{2} + k\frac{y_n}{y_1})^2 \big) \\ &\overset{\text{(b}}{=} -1 + \sum_{n=2}^{d+1} (-\sqrt{2} + k\frac{y_n}{y_1})^2 \\ &\overset{\text{(c)}}{>} 0, \end{aligned}$$

where (a) is due to (D.4), (b) follows from $y_n \in \{-1, 1\}$, and (c) is obvious if $k > \sqrt{2} + 1$. Therefore, linear hyperbolic classifiers can generate any set of labels for the point set $\{x_n\}_{n \in [d+1]}$. Furthermore, hyperbolic classifiers in (D.3) can be seen as linear classifiers in $(d+1)$-dimensional Euclidean space. Hence, the VC dimension of linear classifiers in hyperbolic space is exactly $d + 1$.

From Theorem 1 and the fundamental theorem of concept learning [191], the set of linear product space form classifiers $\mathcal{L}$ is probably accurately correctly (PAC) learnable. More precisely, let Δ be a family of probability distributions on $\mathcal{M} \times \{-1, 1\}$, and let $\{(x_n, y_n)\}_{n \in [N]}$ be a set of i.i.d. samples from $P \in \Delta$.

Then, we have

$$\inf_{l\in\mathcal{L}} P(\widehat{l}_N(X)\neq Y) \leq \inf_{l\in\mathcal{L}} P(l(X)\neq Y) + C\sqrt{\frac{d+1}{n}} + \sqrt{\frac{2\log(\frac{1}{\delta})}{n}},$$

where $\widehat{l}_N = \arg\min_{l\in\mathcal{L}} \frac{1}{N}\sum_{n\in[N]} 1(l(x_n)\neq y_n)$ is the empirical risk minimizer. Therefore, spherical, hyperbolic, and Euclidean linear classifiers have the same learning complexity.

D.4 Proof of Proposition 12

Let $\mathcal{M} = \mathbb{E}^{d_\mathbb{E}} \times \mathbb{S}^{d_\mathbb{S}} \times \mathbb{H}^{d_\mathbb{H}}$ be a product space with the Riemannian metric $g = \alpha_\mathbb{E} g^\mathbb{E} + \alpha_\mathbb{S} g^\mathbb{S} + \alpha_\mathbb{H} g^\mathbb{H}$. Fact 1 gives us the logarithm map and tangent space at a point $p = (p_\mathbb{E}, p_\mathbb{S}, p_\mathbb{H}) \in \mathcal{M}$. A tangent vector $w \in T_p\mathcal{M}$ can be expressed as $w = (w_\mathbb{E}, w_\mathbb{S}, w_\mathbb{H})$ where $w_\mathbb{E} \in T_{p_\mathbb{E}}\mathbb{E}^{d_\mathbb{E}}$, $w_\mathbb{S} \in T_{p_\mathbb{S}}\mathbb{S}^{d_\mathbb{S}}$, and $w_\mathbb{H} \in T_{p_\mathbb{H}}\mathbb{H}^{d_\mathbb{H}}$. From the point-line definition of linear classifiers (Definition 2), we have

$$\begin{aligned} l^\mathcal{M}_{p,w}(x) &= \operatorname{sgn}\big(g_p(\log_p(x), w)\big) \\ &= \operatorname{sgn}\Big(\sum_{S\in\{\mathbb{E},\mathbb{S},\mathbb{H}\}} \alpha_S g^S_{p_S}(\log_{p_S}(x_S), w_S)\Big). \end{aligned}$$

In Propositions 10 and 11, we derived specific spherical and hyperbolic base points to formalize distance-based classifiers. From these results, we may define a linear classifier in $\mathcal{M}$ that is parameterized only with a tangent vector w, i.e.,

$$l^\mathcal{M}_w(x) = \operatorname{sgn}\big((\alpha_\mathbb{E} w_\mathbb{E})^\top x_\mathbb{E} + b + \alpha_\mathbb{S}\operatorname{asin}(w_\mathbb{S}^\top x_\mathbb{S}) + \alpha_\mathbb{H}\operatorname{asinh}([w_\mathbb{H}, x_\mathbb{H}])\big), \quad \text{(D.5)}$$

where $\|w_\mathbb{E}\| = 1$, $\|w_\mathbb{S}\| = C_\mathbb{S}$, and $[w_\mathbb{H}, w_\mathbb{H}] = -C_\mathbb{H}$. This completes the proof.

Remark. The linear classifier of (D.5) is not a distance-based classifier with respect to our choice of the Riemannian metric g. The distance between a point x and the classification boundary $H_{p,w}$ can be computed as

$$\begin{aligned} d(x, H_{p,x}) &= \min_{y\in H_{p,w}} d(x,y) \\ &= \Big(\alpha^2_\mathbb{E}\,\|x_\mathbb{E} - y^*_\mathbb{E}\|^2 + \alpha^2_\mathbb{S}\frac{1}{C_\mathbb{S}}\operatorname{acos}^2(C_\mathbb{S} x_\mathbb{S}^\top y^*_\mathbb{S}) + \alpha^2_\mathbb{H}\frac{1}{-C_\mathbb{H}}\operatorname{acosh}^2(C_\mathbb{H}[x_\mathbb{H}, y^*_\mathbb{H}])\Big)^{1/2}, \end{aligned}$$

where y^* is the projection of x onto the separation plane $H_{p,w}$. It is easy to verify that this distance is not related to the decision criteria, i.e., $(\alpha_{\mathbb{E}} w_{\mathbb{E}})^\top x_{\mathbb{E}} + b + \alpha_{\mathbb{S}} \text{asin}(w_{\mathbb{S}}^\top x_{\mathbb{S}}) + \alpha_{\mathbb{H}} \text{asinh}([w_{\mathbb{H}}, x_{\mathbb{H}}])$, which only takes the weighted sum of (signed) distances between x_S and H_{p_S,w_S} for $S \in \{\mathbb{E}, \mathbb{S}, \mathbb{H}\}$.

D.5 Proof of Theorem 2

Lemma 4. *Let $K(x_1, x_2) = \text{asin}(x_1^\top x_2)$, where $x_1, x_2 \in B_\circ = \left\{x \in \mathbb{R}^d : \|x\|_2 \leq 1\right\}$. Then, there exists a Hilbert space $\mathcal{H}_\circ$, and a mapping $\phi_\circ : B_\circ \to \mathcal{H}_\circ$ such that*

$$K(x_1, x_2) = \langle \phi_\circ(x_1), \phi_\circ(x_2) \rangle_{\mathcal{H}_\circ},$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}_\circ}$ is the inner product on $\mathcal{H}_\circ$. Moreover, we can construct a space $\mathcal{H}'_\circ \supset \mathcal{H}_\circ$ such that indefinite inner products of the form $\langle \cdot, M_\circ \cdot \rangle_{\mathcal{H}_\circ}$ are well-defined on $\mathcal{H}'_\circ$. The indefinite operator $M_\circ : \mathcal{H}'_\circ \to \mathcal{H}'_\circ$ admits the following representation

$$K_H(x_1, x_2) = \text{asinh}(x_1^\top x_2) = \langle \phi_\circ(x_1), M_\circ \phi_\circ(x_2) \rangle_{\mathcal{H}_\circ},$$

for all x_1, x_2 in a compact subset of $\mathbb{R}^d$, and it satisfies $M_\circ^\top M_\circ = \text{Id}$, where Id *denotes the identity operator.*

Proof. The Taylor series expansion of asin can be used to establish that

$$\text{asin}(x_1^\top x_2) = \sum_{n=0}^{\infty} \frac{(2n)!}{2^{2n}(n!)^2(2n+1)} (x_1^\top x_2)^{2n+1}, \tag{D.6}$$

where $\left|x_1^\top x_2\right| \leq 1$. All the coefficients of this Taylor series are non-negative. Hence, from Theorem 2.1 in [192], this is a valid positive-definite kernel. Therefore, there is a Hilbert space $\mathcal{H}_\circ$ endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_\circ}$ such that

$$\text{asin}(x_1^\top x_2) = \langle \phi_\circ(x_1), \phi_\circ(x_2) \rangle_{\mathcal{H}_\circ},$$

for $x_1, x_2 \in B_\circ$ and vectors $\phi_\circ(x_1)$ and $\phi_\circ(x_2) \in \mathcal{H}_\circ$.

On the other hand, we have

$$\operatorname{asinh}(x_1^\top x_2) = \sum_{n=0}^{\infty} (-1)^n \frac{(2n)!}{2^{2n}(n!)^2(2n+1)} (x_1^\top x_2)^{2n+1},$$

where $x_1, x_2 \in B \subseteq \mathbb{R}^d$ — a compact subset of $\mathbb{R}^d$. This Taylor series is the same as the one given in (D.6) except for the alternating signs of the coefficients. The analytical construction of the vector $\phi_\circ(x)$ in [192] gives a straightforward way to define an indefinite operator $M_\circ : \mathcal{H}'_\circ \to \mathcal{H}'_\circ$ such that $M_\circ^\top M_\circ = \mathrm{Id}$ and

$$\operatorname{asinh}(x_1^\top x_2) = \langle \phi_\circ(x_1), M_\circ \phi_\circ(x_2) \rangle_{\mathcal{H}'_\circ}.$$

Note that $M_\circ$ is a finite-dimensional diagonal matrix with elements ± 1 that represent the signs of the Taylor series coefficients.

The space $\mathcal{H}'_\circ$ contains $\mathcal{H}_\circ$ with the same definite inner product, i.e., if $\phi_\circ(x), \phi_\circ(y) \in \mathcal{H}'_\circ \cap \mathcal{H}$, then $\langle \phi_\circ(x), \phi_\circ(y) \rangle_{\mathcal{H}_\circ} = \langle \phi_\circ(x), \phi_\circ(y) \rangle_{\mathcal{H}'_\circ}$. However, a point $\phi_\circ(x) \in \mathcal{H}'_\circ \setminus \mathcal{H}_\circ$ may have an unbounded norm, i.e., $\langle \phi_\circ(x), \phi_\circ(x) \rangle_{\mathcal{H}'_\circ} = \infty$. Nevertheless, the indefinite inner products of the form $\langle \phi_\circ(x), M_\circ \phi_\circ(x) \rangle_{\mathcal{H}'_\circ}$ are always well-defined so long as $x \in B$, a compact subset of $\mathbb{R}^d$. This is due to the fact that the convergence domain for the Taylor series of $\operatorname{asinh}(\cdot)$ is any compact subset of $\mathbb{R}$. Hence, we can simply define $\mathcal{H}'_\circ = \left\{ \phi(x) : x \in B \subset \mathbb{R}^d \right\}$, where B is a compact subset of $\mathbb{R}^d$. □

Let $\{x_1, \ldots, x_N\}$ be a set of N points in the product space $\mathcal{M}$. For any point $x = [x_\mathbb{E}, x_\mathbb{S}, x_\mathbb{H}] \in \mathcal{M}$, we define

$$\phi(x) = \big(1, x_\mathbb{E}, \sqrt{\alpha_\mathbb{S}} \phi_\circ(\sqrt{C_\mathbb{S}} x_\mathbb{S}), \sqrt{\alpha_\mathbb{H}} \phi_\circ(H \frac{1}{R} x_\mathbb{H})\big),$$

where $\phi(x)$ is defined as in the proof of Lemma 4, and R is an upper bound for the norm of the hyperbolic component of x, i.e., $\|x_\mathbb{H}\|_2 \leq R$. Note that in order to distinguish the curvatures of different space forms, we added appropriate subscripts.

The linear classifier in product space form can be written as

$$\begin{aligned} l_w^\mathcal{M}(x) &= \operatorname{sgn}\big(w_\mathbb{E}^\top x_\mathbb{E} + b + \alpha_\mathbb{S} \operatorname{asin}(w_\mathbb{S}^\top x_\mathbb{S}) + \alpha_\mathbb{H} \operatorname{asinh}((R w_\mathbb{H})^\top \frac{1}{R} H x_\mathbb{H})\big) \\ &= \operatorname{sgn}\big(\langle \psi(w), M \phi(x) \rangle_\mathcal{H}\big), \end{aligned}$$

where $\mathcal{H}$ is a simple product of $\mathbb{R}^{d_{\mathbb{E}}+1}$, $\mathcal{H}_\circ$ and $\mathcal{H}'_\circ$ accompanied by their corresponding inner products, $\psi(w) = \big(b, w_{\mathbb{E}}, \sqrt{\alpha_{\mathbb{S}}}\phi_\circ(\frac{1}{\sqrt{C_{\mathbb{S}}}}w_{\mathbb{S}}), \sqrt{\alpha_{\mathbb{H}}}\phi_\circ(Rw_{\mathbb{H}})\big) \in \mathcal{H}$, and $M = \operatorname{diag}\{I, I, M_\circ\}$ is a product operator on $\mathcal{H}$ such that

$$\begin{aligned}\langle \psi(w), M\phi(x)\rangle_{\mathcal{H}} = w_{\mathbb{E}}^\top x_{\mathbb{E}} + b + \alpha_{\mathbb{S}}\langle \phi_\circ(\frac{1}{\sqrt{C_{\mathbb{S}}}}w_{\mathbb{S}}), \phi_\circ(\sqrt{C_{\mathbb{S}}}x_{\mathbb{S}})\rangle_{\mathcal{H}_\circ} \\ + \alpha_{\mathbb{H}}\langle \phi_\circ(Rw_{\mathbb{H}}), M_\circ\phi_\circ(\frac{1}{R}Hx_{\mathbb{H}})\rangle_{\mathcal{H}'_\circ}.\end{aligned}$$

From the problem assumptions, we assume the data points are linearly separable, i.e.,

$$\forall n \in [N] : y_n\langle w^*, M\phi(x_n)\rangle_{\mathcal{H}} \geq \varepsilon,$$

for a specific w^* in $\mathcal{H}$. Similar to the hyperbolic perceptron setting, we use the following update rule in RKHS

$$w^{k+1} = w^k + y_n M\phi(x_n) \;\text{ if }\; y_n\langle w^k, M\phi(x_n)\rangle_{\mathcal{H}} \leq 0.$$

If we initialize $w^0 = 0 \in \mathcal{H}$, we have

$$\begin{aligned}\langle w^*, w^{k+1}\rangle_{\mathcal{H}} &= \langle w^*, w^k\rangle_{\mathcal{H}} + \langle w^*, My_n\phi(x_n)\rangle_{\mathcal{H}} \\ &\geq \langle w^*, w^k\rangle_{\mathcal{H}} + \varepsilon \\ &\geq k\varepsilon.\end{aligned}$$

On the other hand, we can bound the norm as

$$\begin{aligned}&\langle w^{k+1}, w^{k+1}\rangle_{\mathcal{H}} \\ &= \langle w^k, w^k\rangle_{\mathcal{H}} + \langle y_n M\phi(x_n), y_n M\phi(x_n)\rangle_{\mathcal{H}} + 2\langle w^k, y_n M\phi(x_n)\rangle_{\mathcal{H}} \\ &\leq \langle w^k, w^k\rangle_{\mathcal{H}} + \langle \phi(x_n), \phi(x_n)\rangle_{\mathcal{H}} \\ &\leq \langle w^k, w^k\rangle_{\mathcal{H}} + 1 + \|x_{\mathbb{E},n}\|_2^2 + \alpha_{\mathbb{S}}\langle \phi_\circ(\sqrt{C_{\mathbb{S}}}x_{\mathbb{S},n}), \phi_\circ(\sqrt{C_{\mathbb{S}}}x_{\mathbb{S},n})\rangle_{\mathcal{H}_\circ} \\ &+ \alpha_{\mathbb{H}}\langle \phi_\circ(\frac{1}{R}Hx_{\mathbb{H},n}), \phi_\circ(\frac{1}{R}Hx_{\mathbb{H},n})\rangle_{\mathcal{H}'_\circ} \\ &\overset{\text{(a)}}{\leq} k(1 + R_{\mathbb{E}}^2 + (\alpha_{\mathbb{S}} + \alpha_{\mathbb{H}})\frac{\pi}{2}),\end{aligned}$$

where $R_{\mathbb{E}}$ is an upper bound for the norm of the Euclidean components of the vectors, and (a) is due to

$$\langle \phi_\circ(\sqrt{C_{\mathbb{S}}}x_{\mathbb{S},n}), \phi_\circ(\sqrt{C_{\mathbb{S}}}x_{\mathbb{S},n})\rangle_{\mathcal{H}_\circ} = \operatorname{asin}(C_{\mathbb{S}}x_{\mathbb{S},n}^\top x_{\mathbb{S},n}) = \frac{\pi}{2},$$

and

$$\langle \phi_\circ(\frac{1}{R} H x_{\mathbb{H},n}), \phi_\circ(\frac{1}{R} H x_{\mathbb{H},n}) \rangle_{\mathcal{H}'_\circ} = \operatorname{asin}(\frac{1}{R^2} x_{\mathbb{H},n}^\top x_{\mathbb{H},n}) \leq \frac{\pi}{2}.$$

Hence, we have

$$\begin{aligned} \frac{(\langle w^{k+1}, w^* \rangle_{\mathcal{H}})^2}{\langle w^{k+1}, w^{k+1} \rangle_{\mathcal{H}} \langle w^*, w^* \rangle_{\mathcal{H}}} &\geq \frac{k^2 \varepsilon^2}{k B_T \langle w^*, w^* \rangle_{\mathcal{H}}} \\ &= k \frac{\varepsilon^2}{B_T \langle w^*, w^* \rangle_{\mathcal{H}}}, \end{aligned}$$

where $B_T = 1 + R_{\mathbb{E}}^2 + (\alpha_{\mathbb{S}} + \alpha_{\mathbb{H}}) \frac{\pi}{2}$. Therefore, convergence is guaranteed in $k \leq \frac{B_T \langle w^*, w^* \rangle_{\mathcal{H}}}{\varepsilon^2}$ steps. Finally, the upper bound for the ℓ_2 norm of $w_{\mathbb{H}}$ guarantees the boundedness of $\langle w^*, w^* \rangle_{\mathcal{H}}$.

D.6 Proof of Theorem 3

Let $w^0 = 0 \in \mathbb{R}^{d+1}$ and let $w^k \in \mathbb{R}^{d+1}$ be the estimated normal vector at the k-th iteration of the perceptron algorithm (see Algorithm 10). If the point $x_n \in \mathbb{L}^d$ ($y_n[w^k, x_n] < 0$) is misclassified, the perceptron algorithm produces the $(k+1)$-th estimate of the normal vector according to

$$w^{k+1} = w^k + y_n H x_n.$$

Let w^* be the normal vector that classifies all the points with margin of at least ε, i.e., $y_n \operatorname{asinh}([w^*, x_n]) \geq \varepsilon$, $\forall\, n \in [N]$, and $[w^*, w^*] = 1$. Then, we have

$$\begin{aligned} (w^*)^\top w_{k+1} &= (w^*)^\top w^k + y_n [w^*, x_n] \\ &\geq (w^*)^\top w^k + \sinh(\varepsilon) \\ &\geq k \sinh(\varepsilon). \end{aligned}$$

Algorithm 10 The hyperbolic perceptron.

Input: $\{x_n, y_n\}_{n=1}^N$: a set of point-labels in $\mathbb{H}^{d_{\mathbb{H}}} \times \{-1, 1\}$.
Initialization: $w^0 = 0 \in \mathbb{R}^{d_{\mathbb{H}}+1}$, $k = 0$, $n = 1$.
repeat
  if $\text{sgn}([w^k, x_n]) \neq y_n$ **then**
    $w^{k+1} = w^k + y_n H x_n$;
    $k = k + 1$;
  end if
  $n = \text{mod}(n, N) + 1$;
until Convergence criteria is met.

In what follows, we provide an upper bound on the term[2] $\left\|w^{k+1}\right\|$,

$$\begin{aligned}
\left\|w^{k+1}\right\|^2 &= \left\|w^k + y_n H x_n\right\|^2 \\
&= \left\|w^k\right\|^2 + \|x_n\|^2 + 2y_n[w^k, x_n] \\
&\overset{(a)}{\leq} \left\|w^k\right\|^2 + R^2 \\
&= kR^2,
\end{aligned}$$

where (a) is due to $\|x_n\|^2 \leq R^2$ and $y_n[w^k, x_n] \leq 0$, due to the error in classifying the point x_n. Hence,

$$\left\|w^{k+1}\right\| \leq \sqrt{k}R \quad \text{and} \quad (w^*)^\top w^{k+1} \geq k\text{sinh}(\varepsilon). \tag{D.7}$$

To complete the proof, define $\theta_k = \text{acos}(\frac{(w^k)^\top w^*}{\left\|w^k\right\| \|w^*\|})$. Then,

$$\begin{aligned}
\frac{(w^{k+1})^\top w^*}{\|w^{k+1}\| \|w^*\|} &\overset{(a)}{\geq} \frac{k\text{sinh}(\varepsilon)}{\sqrt{k}R \|w^*\|} \\
&= \sqrt{k}\frac{\text{sinh}(\varepsilon)}{R \|w^*\|},
\end{aligned}$$

where (a) follows from (D.7). For $k \geq \left(\frac{R\|w^*\|}{\text{sinh}(\varepsilon)}\right)^2$, we have $w^{k+1} = \alpha_{k+1} w^*$ for a positive scalar α_{k+1}. Hence, $\frac{1}{\sqrt{[w^{k+1}, w^{k+1}]}} w^{k+1} = w^*$.

[2]Here, the norm is taken in the Euclidean sense, i.e., $\|w\| = \sqrt{w^\top w}$.

D.6.1 Discussion

Linear classifiers in spherical spaces have been studied in a number of works [140, 141], while more recent work has focused on linear classifiers in the Poincaré model of hyperbolic spaces, in the context of hyperbolic neural networks [33]. A purely hyperbolic perceptron (in the same 'Loid model used in this work) was described in [143]. The proposed update rule reads as

$$u^k = w^k + y_n x_n \quad \text{if} \quad -y_n[w^k, x_n] < 0 \tag{D.8}$$

$$w^{k+1} = u^k / \min\{1, \sqrt{[u^k, u^k]}\}, \tag{D.9}$$

where (D.9) is a "normalization step". Unfortunately, the above update rule does not allow the hyperbolic perceptron algorithm (equations (D.8) and (D.9)) to converge, which is due to the choice of the update direction. The convergence issue is also illustrated by the following two examples.

Let $x_1 = [\sqrt{2}, 1, 0]^\top \in \mathbb{L}^2$ with label $y_1 = 1$. We choose the initial vector in the update rule to be $w^0 = [\frac{-\sqrt{2}+3}{4}, \frac{-1+3\sqrt{2}}{4}, 0]^\top$ (in contrast to $w^0 = e_2$, which was chosen in the proof [143]). This is a valid choice because $[w^0, w^0] = \frac{1}{2} > 0$. In the first iteration, we must hence update w^0 since $-y_1[w^0, x_1] = -\frac{1}{4} < 0$. From (D.8), we have $u^0 = w^0 + y_1 x_1 = [\frac{3\sqrt{2}+3}{4}, \frac{3+3\sqrt{2}}{4}, 0]^\top$, and $[u^0, u^0] = 0$. This means that $w^1 = \frac{1}{\sqrt{[u^0, u^0]}} u^0$ is clearly ill-defined.

As another example, let $w^\star$ be the optimal vector with which we can classify all data points with margin ε. If we simply choose $w^0 = 0$, then we can satisfy the required condition $[w^0, w^\star] \geq 0$ postulated for the hyperbolic perceptron. This leads to $u^0 = x_1$. Then, for any $x_1 \in \mathbb{L}^2$, we have $[x_1, x_1] = -1$, which leads to a normalization factor $\sqrt{[u^0 u^0]}$ that is a complex number.

D.7 Proof of Proposition 13

Let $\psi(w) = [b, w_\mathbb{E}, \sqrt{\alpha_\mathbb{S}}\phi_\circ(\frac{1}{\sqrt{C_\mathbb{S}}} w_\mathbb{S}), \sqrt{\alpha_\mathbb{H}}\phi_\circ(R w_\mathbb{H})] = \sum_{n\in[N]} \beta_n M \phi(x_n)$. We now consider the norm constraint for each component separately.

The parameters for Euclidean component can be written as

$$b = \sum_{n\in[N]} \beta_n, \;\; w_\mathbb{E} = \sum_{n\in[N]} \beta_n x_{\mathbb{E},n}.$$

The distance-based Euclidean classifier asks for a vector such that $\|w_{\mathbb{E}}^*\|_2 = \alpha_{\mathbb{E}}$. We can impose this condition as a quadratic equality constraint on the vector $\beta = (\beta_1, \ldots, \beta_N)$ as follows

$$\|w_{\mathbb{E}}\|^2 = \beta^\top K_{\mathbb{E}} \beta = \alpha_{\mathbb{E}}^2,$$

where $K_{\mathbb{E}} = \left(x_{\mathbb{E},i}^\top x_{\mathbb{E},j}\right)_{i,j\in[N]}$. The parameter of the spherical component can be written as

$$\phi_\circ(\frac{1}{\sqrt{C_{\mathbb{S}}}} w_{\mathbb{S}}) = \sum_{n\in[N]} \beta_n \phi_\circ(\sqrt{C_{\mathbb{S}}} x_{\mathbb{S},n}).$$

A distance-based spherical classifier requires $w_{\mathbb{S}} : \|w_{\mathbb{S}}\|_2 = \sqrt{C_{\mathbb{S}}}$. Therefore, we must have $\phi_\circ(\frac{1}{\sqrt{C_{\mathbb{S}}}} w_{\mathbb{S}})^\top \phi_\circ(\frac{1}{\sqrt{C_{\mathbb{S}}}} w_{\mathbb{S}}) = \text{asin}(1)$, which can be imposed by the following quadratic constraint

$$\left\| \phi_\circ(\frac{1}{\sqrt{C_{\mathbb{S}}}} w_{\mathbb{S}}) \right\|^2 = \beta^\top K_{\mathbb{S}} \beta = \frac{\pi}{2},$$

where $K_{\mathbb{S}} = \left(\text{asin}(C_{\mathbb{S}} x_{\mathbb{S},i}^\top x_{\mathbb{S},j})\right)_{i,j\in[N]}$. Finally, we can write the hyperbolic component as follows

$$\phi_\circ(R w_{\mathbb{H}}) = \sum_{n\in[N]} \beta_n M_\circ \phi_\circ(\frac{1}{R} H x_{\mathbb{H},n}).$$

The distance-based hyperbolic classifier $w_{\mathbb{H}}$ must satisfy the norm constraint of $[Rw_{\mathbb{H}}, Rw_{\mathbb{H}}] = -R^2 C_{\mathbb{H}}$. Consequently, we must have

$$\phi_\circ(R w_{\mathbb{H}})^\top M_\circ \phi_\circ(RH w_{\mathbb{H}}) = \text{asinh}(-R^2 C_{\mathbb{H}}).$$

Lemma 5. $\phi_\circ(RH w_{\mathbb{H}}) = \sum_{i\in[N]} \beta_n M_\circ \phi_\circ(\frac{1}{R} x_{\mathbb{H},n})$.

Proof.

$$\begin{aligned}
\operatorname{asinh}(x_{\mathbb{H}}^{\top} w_{\mathbb{H}}) &= \phi_{\circ}(\frac{1}{R} x_{\mathbb{H}})^{\top} M_{\circ} \phi_{\circ}(R w_{\mathbb{H}}) \\
&\overset{\text{(a)}}{=} \sum_{n \in [N]} \beta_n \operatorname{asin}(\frac{1}{R^2}[x_{\mathbb{H}}, x_{\mathbb{H},n}]) \\
&= \sum_{n \in [N]} \beta_n \phi_{\circ}(\frac{1}{R} H x_{\mathbb{H}})^{\top} M_{\circ} M_{\circ} \phi_{\circ}(\frac{1}{R} x_{\mathbb{H},n}) \\
&= \phi_{\circ}(\frac{1}{R} H x_{\mathbb{H}})^{\top} M_{\circ} \sum_{n \in [N]} \beta_n M_{\circ} \phi_{\circ}(\frac{1}{R} x_{\mathbb{H},n}),
\end{aligned}$$

where (a) is due to $\phi_{\circ}(R w_{\mathbb{H}}) = \sum_{n \in [N]} \beta_n M_{\circ} \phi_{\circ}(\frac{1}{R} H x_{\mathbb{H},n})$. From $\operatorname{asinh}(x_{\mathbb{H}}^{\top} w_{\mathbb{H}}) = \phi_{\circ}(\frac{1}{R} H x_{\mathbb{H}})^{\top} M_{\circ} \phi_{\circ}(RH w_{\mathbb{H}})$, we have $\phi_{\circ}(RH w_{\mathbb{H}}) = \sum_{n \in [N]} \beta_n M \phi_{\circ}(\frac{1}{R} x_{\mathbb{H},n})$. □

From the lemma, we have

$$\begin{aligned}
\operatorname{asinh}([R w_{\mathbb{H}}, R w_{\mathbb{H}}]) &= \phi_{\circ}(R w_{\mathbb{H}})^{\top} M \phi_{\circ}(RH w_{\mathbb{H}}) \\
&= \sum_{i,j} \beta_i \beta_j \operatorname{asinh}(\frac{1}{R^2}[x_{\mathbb{H},i}, x_{\mathbb{H},j}]) \\
&= \operatorname{asinh}(-R^2 C_{\mathbb{H}}).
\end{aligned}$$

The kernel matrix $K_{\mathbb{H}} = \big(\operatorname{asinh}(\frac{1}{R^2}[x_{\mathbb{H},i}, x_{\mathbb{H},j}])\big)_{i,j \in [N]}$ is an indefinite matrix. Therefore, we have the following non-convex second-order equality constraint

$$\phi_{\circ}^{\top}(R w_{\mathbb{H}}) M_{\circ} \phi_{\circ}(RH w_{\mathbb{H}}) = \beta^{\top} K_{\mathbb{H}} \beta = \operatorname{asinh}(-R^2 C_{\mathbb{H}}).$$

D.8 Experiments

We present additional experimental results that are not covered in the main text. All experiments were conducted on a Linux machine with 48 cores, 376GB of system memory.

D.8.1 Datasets

MNIST[3] [145], Omniglot[4] [146], CIFAR-100[5] [147], and single-cell expressions [148, 149, 150][6,7,8] are publicly available datasets. Specific details of the three single-cell expressions datasets are as follows:

1. *Lymphoma patient.* Human dissociated lymph node tumor cells of a 19-year-old male Hodgkins Lymphoma patient were obtained by 10x Genomics from Discovery Life Sciences. Whole transcriptome libraries were generated with Chromium Next GEM Single Cell 3' Reagent Kits v3.1 (Dual Index) User Guide (CG000315) and sequenced on an Illumina NovaSeq 6000. The targeted libraries were generated using the Targeted Gene Expression Reagent Kits User Guide (CG000293) and Human Immunology Panel reagent (PN-1000246) and sequenced on an Illumina NovaSeq 6000.

2. *Lymphoma-healthy donor.* Human peripheral blood mononuclear cells (PBMCs) of a healthy female donor aged 25 were obtained by 10x Genomics from AllCells. Whole transcriptome libraries were generated with Chromium Next GEM Single Cell 3' Reagent Kits v3.1 (Dual Index) User Guide (CG000315) and sequenced on an Illumina NovaSeq 6000. The aforementioned two datasets have 13410 samples (combined) and each for a class (binary classification). The dimension of each cell expression vector is 1020 (both datasets).

3. *Blood cells landmark.* We use the dataset originally from this paper and extract the gene expression data for (1) B cells, (2) Cd14 monocytes, (3) Cd34 monocytes, (4) Cd4 t helper cells, (5) Cd56 natural killer cells, (6) Cytotoxic T cells, (7) Memory T cells, (8) Naive cytotoxic cells, (9) Native T cells, and (10) Regulatory T cells. This dataset has 94655 samples from the total 10 classes. The dimension of each cell expression vector is 965.

[3] `yann.lecun.com/exdb/mnist/`

[4] `github.com/brendenlake/omniglot`

[5] `www.cs.toronto.edu/~kriz/cifar.html`

[6] `10xgenomics.com/resources/datasets/hodgkins-lymphoma-dissociated-tumor-targeted-compare-immunology-panel-3-1-standard`

[7] `10xgenomics.com/resources/datasets/pbm-cs-from-a-healthy-donor-targeted-immunology-panel-3-1-standard`

[8] `nature.com/articles/ncomms14049`

D.8.2 Convergence Analysis of Hyperbolic Perceptron

As pointed out in Appendix D.6.1, the hyperbolic perceptron described in [143] does not converge, which can be shown both through counterexamples and simulation studies. We report the following experimental results to validate this point and in particular, demonstrate that a convergence rate of $O\left(\frac{1}{\sinh(\varepsilon)}\right)$ is not possible.

First, we randomly generate a valid w^* such that $[w^*, w^*] = 1$. Then, we generate a random set of $N = 5,000$ points $\{x_i\}_{i=1}^N$ in $\mathbb{L}^2$. For margin values $\varepsilon \in [0.1, 1]$, we remove points that violate the required distance to the classifier (parameterized with w^*), i.e., we decimate the points so that the condition $\forall n : |[w^*, x_n]| \geq \sinh(\varepsilon)$ is satisfied. Then, we assign binary labels to each data point according to the optimal classifier, so that $y_n = \operatorname{sgn}\big(\operatorname{asinh}([w^*, x_n])\big)$. We repeat this process for 100 different values of ε.

In the first experiment, we compare our proposed hyperbolic perceptron Algorithm 10 and the Algorithm 1 in [143] by running the methods until the number of updates achieved a predetermined upper bound (stated in Theorem 3) or until the classifier correctly classified all the data points. In Figure D.1 (a), we report the classification accuracy of each method on the training data. Note that our theoretically established convergence rate $O\left(\frac{1}{\sinh^2(\varepsilon)}\right)$ is larger than $O\left(\frac{1}{\sinh(\varepsilon)}\right)$, the rate derived in Theorem 3.1 in [143]. So, for the second experiment, we repeated the same process but terminate both algorithms after $O\left(\frac{1}{\sinh(\varepsilon)}\right)$ updates. The classification performance of the two algorithms under this setting is shown in Figure D.1 (b). From these results, we can easily conclude that (1) our algorithm always converge within the theoretical upper bound provided in Theorem 3, and (2) both methods violate the theoretical convergence rate upper bound of [143].

D.8.3 Synthetic Data

We illustrate the practical performance of our product space form perceptron Algorithm 1 on both synthetic and real-world datasets. In order to establish the benefits of product space form embeddings and learning, we compare our results with those obtained by using a Euclidean perceptron. As is a common approach for perceptron methods, we evaluate the classification accuracy on the training sets. To ensure a fair comparison, we restrict the latent dimension

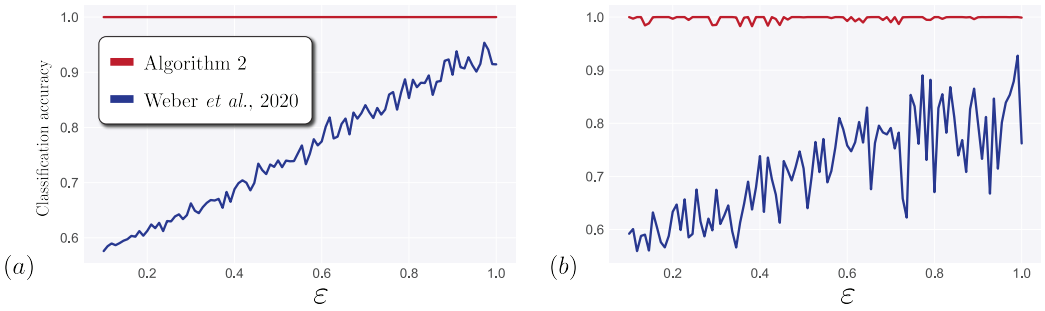


Figure D.1: A comparison between the classification accuracy of our hyperbolic perceptron Algorithm 10 and the algorithm in [143] for different values of the margin ε. The classification accuracy is the average of five independent random trials. The stopping criterion is either a 100% classification accuracy or the theoretical upper bound in Theorem 3 (Figure (a)), and Theorem 3.1 in [143] (Figure (b)).

of the embeddings of both methods to be the same, meaning that data points lie in $\mathbb{E}^{d_{\mathbb{E}}} \times \mathbb{S}^{d_{\mathbb{S}}} \times \mathbb{H}^{d_{\mathbb{H}}}$ for the product space form perceptron and in $\mathbb{E}^{d_{\mathbb{E}}+d_{\mathbb{S}}+d_{\mathbb{H}}}$ for the Euclidean perceptron.

We generate binary-labeled synthetic data satisfying a ε-margin assumption as follows. First, we randomly and independently sample N points from a Gaussian distribution in each of the three spaces $\mathbb{E}^2, \mathbb{E}^3, \mathbb{E}^3$; subsequently, we project the points in $\mathbb{E}^3$ and $\mathbb{E}^3$ onto $\mathbb{S}^2_1$ and $\mathbb{H}^2_{-1}$, respectively. Then, we concatenate the coordinates from the three space form components to obtain the product space form embeddings. Finally, we randomly generate the optimal decision hyperplane $w^* = (w^*_{\mathbb{E}}, 0, w^*_{\mathbb{S}}, w^*_{\mathbb{H}})$ under the constraints stated in Theorem 1 and assign binary labels to data points. To ensure that the ε-margin assumption is satisfied, we translate points that violate this assumption.

We use the same data for both the Euclidean and product space form

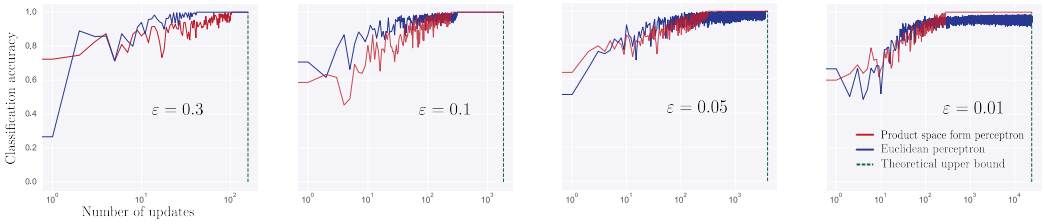


Figure D.2: Classification accuracy after each update of the Euclidean and product space form perceptron algorithms for $N = 300$ and different values of ε.

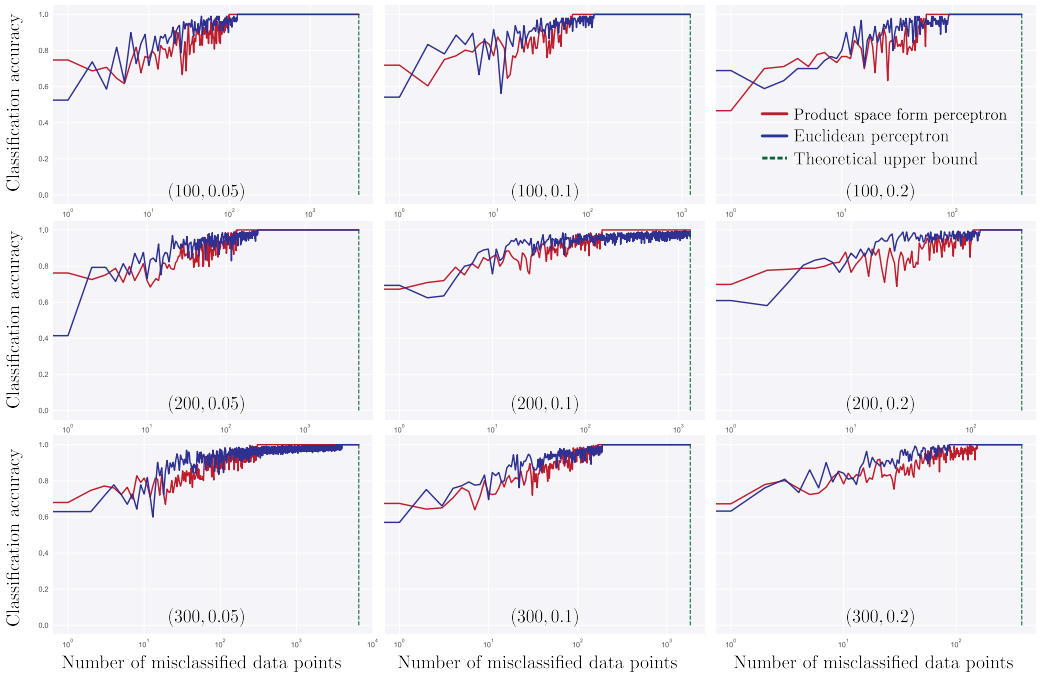


Figure D.3: Classification accuracy after each update of the Euclidean and product space form perceptron algorithms for nine different combinations of (N, ε).

perceptron to demonstrate the efficiency and performance gains of the former method. This is because our method respects the geometry of data, whereas the purely Euclidean setting assumes input data lies in $\mathbb{E}^8$. In Figure D.2, we show four different typical experimental convergence plots for $N = 300$ points with the same optimal decision hyperplane w^*, but with different separation margins, i.e., ε. We observe that the number of updates made by the product space form perceptron is always smaller than the theoretical upper bound provided in Theorem 1. When the margin is small, the data is not linearly separable in Euclidean space and the Euclidean perceptron does not converge to a 100% accurate solution. As ε increases, the data becomes easy to classify for both algorithms, and the number of updates made by the Euclidean perceptron decreases. The performance guarantees of the proposed methods is *independent* of the size of datasets.

We now fix the optimal decision hyperplane w^*. In Figure D.3, we show nine more experiments with different combinations of (N, ε). We observe that the number of updates made by the product space form perceptron is always smaller than the theoretical upper bound described in Theorem 2. And, in most cases, the product space form perceptron requires a smaller number of

updates to converge then the Euclidean perceptron due to the fact that it accounts for the geometry of the data.

APPENDIX E

GEOMETRY OF SIMILARITY MEASUREMENTS

Notation For any two numbers $a, b \in \mathbb{R}$, we let $a \vee b$ and $a \wedge b$ be their maximum and minimum. Let C be a subset of a metric space (S, d), and $x \in S$; We define

$$d_{\min}(C) = \inf \{d(x, y) : x, y \in C, x \neq y\},$$
$$d_{\max}(x, C) = \sup \{d(x, y) : y \in C\}.$$

The cardinality of a discrete set C is denoted by card C. The graph-theoretic notations simplifies our main results. For a graph G, we denote its edge set as $E(G)$. Let $G_{p_1,\ldots,p_K}$ be a complete K-partite graph with part sizes $p_1, \ldots p_K$. The Turán graph [179] $T(N, K)$ is a complete K-partite graph with N vertices, and part sizes[1]

$$p_k = \begin{cases} N_1 + 1, & \text{for } \ 1 \leq k \leq K_1 \\ N_1, & \text{for } \ K_1 + 1 \leq k \leq K. \end{cases}$$

Then, card $E\big(T(N, K)\big) = \binom{N}{2} - K_1\binom{N_1+1}{2} - (K - K_1)\binom{N_1}{2}$.[2]

E.1 Proof of Proposition 14

From Definition 1, the values for $\alpha_1(X), \alpha_2(X)$ and $\alpha_3(X)$ are trivial. The lower bound for $\alpha_N(X)$ simply follows from the uniqueness of pairwise dis-

[1] From $\sum_{k=1}^{K} p_k = N$, we have $N_1 = [\frac{N}{K}]$, $K_1 = N - KN_1$.

[2] This is simplified from card $E\big(G_{p_1,\ldots,p_K}\big) = \binom{N}{2} - \sum_{k=1}^{K}\binom{p_k}{2}$. For $K > N$, we assume the graph is complete and $E(T(N, K)) = \binom{N}{2}$.

tances. To put formally, we have

$$\alpha_N(X) = \min_{1 \le m \le \binom{N}{2}} \left\{ \text{card} \bigcup_{s=1}^{m} \{\lambda_{1,s}, \lambda_{2,s}\} = N \right\} \ge [\frac{N}{2}].$$

For the upper bound, $\alpha_N(X)$ is maximum when all $N-1$ smallest pairwise distances are incident to a unique point (see Figure 6.3). The total length of the distance list is $\binom{N}{2}$. Therefore, we have

$$\alpha_N(X) \le \binom{N}{2} - (N-1) + 1 = \binom{N-1}{2} + 1.$$

E.2 Proof of Theorem 4

Let us separately consider hyperbolic, Euclidean, and spherical spaces.

E.2.1 Hyperbolic Space

Let $r \in \mathbb{R}^+$, and $x_1(r), \ldots, x_N(r) \in \mathbb{L}^d$ be a set of parameterized points in 'Loid model of d-dimensional hyperbolic space with $C = -1$ (see Table D.1), such that

$$x_n(r) = \begin{bmatrix} \sqrt{1 + \|y_n(r)\|^2} \\ y_n(r) \end{bmatrix}, \forall n \in [N],$$

where $y_N(r) = 0$, and $y_i(r)^\top y_j(r) = r^2 \cos 2\pi \frac{|i-j|}{N-1}, \forall i, j \in [N-1]$. To see an example, see Figure E.1.

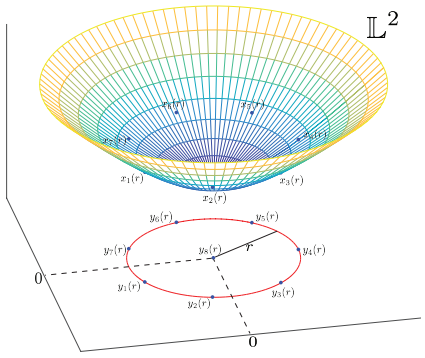


Figure E.1: An example of $N = 8$ parameterized points $\{x_n(r)\}_{n=1}^N$ in $\mathbb{L}^2$ and $\{y_n(r)\}_{n=1}^N$ in $\mathbb{R}^2$.

For these data points, we have

$$d_{\min}\left(\{x_n(r)\}_{n=1}^{N-1}\right) = \operatorname{acosh}\left(1 + r^2(1 - \cos\frac{2\pi}{N-1})\right)$$
$$d_{\max}\left(\{x_n(r)\}_{n=1}^{N-1}, x_N(r)\right) = \operatorname{acosh}\left(\sqrt{1+r^2}\right).$$

Therefore, for any $N \in \mathbb{N}$, there exists a $r \in \mathbb{R}^+$ such that $\{x_n(r)\}_{n=1}^{N} \sqsubseteq \mathbb{L}^d$. Hence,

$$\begin{aligned} K(\mathbb{L}^d) &= \sup\left\{N : \{x_n(r)\}_{n=1}^{N} \sqsubseteq \mathbb{L}^d\right\} \\ &= \infty. \end{aligned}$$

This result hold for all dimensions $d \geq 2$.

E.2.2 Euclidean Space

Lemma 6. *There is a set of points $x_1, \ldots, x_N$ in $\mathbb{R}^d$ such that*

$$\|x_n - x_N\| = 1, \forall n \in [N-1],$$

where $d_{\max}(x_N, \{x_n\}_{n=1}^{N-1}) \leq d_{\min}(\{x_n\}_{n=1}^{N-1})$ and $N = K(\mathbb{R}^d)$.

Proof. Let $\{y_n\}_{n=1}^{N}$ be a set of points in $\mathbb{R}^d$ such that

$$d_{\max}(y_N, \{y_n\}_{n=1}^{N-1}) \leq d_{\min}(\{y_n\}_{n=1}^{N-1}),$$

or $\alpha_N(\{y_n\}_{n=1}^{N}) = \binom{N-1}{2} + 1$. Without loss of generality, we assume $y_N = 0$ and $d_{\max}(y_N, \{y_n\}_{n=1}^{N-1}) = 1$. Let $x_n = \frac{1}{\|y_n\|} y_n,\ \forall n \in [N-1]$ and $x_N = y_N$. We want to show that $\alpha_N(\{x_n\}_{n=1}^{N}) \geq \alpha_N(\{y_n\}_{n=1}^{N})$. Following the definition

of ordinal spread, we have

$$\begin{aligned}
&\alpha_N\left(\{x_n\}_{n=1}^N\right)\\
&\overset{(a)}{\geq} \operatorname{card}\left\{(i,j) : d(x_i,x_j) \geq d_{\max}(x_N, \{x_n\}_{n=1}^{N-1}), i,j \in [N-1], i>j\right\} + 1\\
&\overset{(b)}{=} \operatorname{card}\left\{(i,j) : d(x_i,x_j) \geq 1, i,j \in [N-1], i>j\right\} + 1\\
&\overset{(c)}{\geq} \operatorname{card}\left\{(i,j) : d(y_i,y_j) \geq 1, i,j \in [N-1], i>j\right\} + 1\\
&= \alpha_N(\{y_n\}_{n=1}^N),
\end{aligned}$$

where (a) holds with equality if x_N appears last in the sorted distance list, i.e., if $x_N = x_{(N)}$, (b) is due to $d_{\max}(x_N, \{x_n\}_{n=1}^{N-1}) = 1 = d_{\max}(y_N, \{y_n\}_{n=1}^{N-1})$. To prove inequality (c), let $d(y_i, y_j) \geq 1$ for distinct $i, j \in [N-1]$. Then,

$$\begin{aligned}
d(y_i,y_j)^2 &= \frac{\|y_i\|-1}{\|y_i\|}\left(\|y_i-y_j\|^2 - \|y_j\|^2 + \|y_i\|\right) + \left\|\frac{1}{\|y_i\|}y_i - y_j\right\|^2\\
&= \frac{d(y_N,y_i)-1}{\|y_i\|}\left(d(y_i,y_j)^2 - d(y_N,y_j)^2 + d(y_N,y_i)\right) + \left\|\frac{1}{\|y_i\|}y_i - y_j\right\|^2\\
&\overset{(a)}{\leq} \left\|\frac{1}{\|y_i\|}y_i - y_j\right\|^2\\
&\overset{(b)}{\leq} \left\|\frac{1}{\|y_i\|}y_i - \frac{1}{\|y_j\|}y_j\right\|^2\\
&= d(x_i,x_j)^2,
\end{aligned}$$

where (a) follows from $d(y_N, y_i) \leq 1$, $d(y_N, y_j) \leq 1$, $d(y_i, y_j)^2 \geq 1$, and (b) follows from the symmetry in the argument. Therefore, we have

$$\left\{(i,j) \in [N-1]_{\text{as}}^2 : d(y_i,y_j) \geq 1\right\} \subseteq \left\{(i,j) \in [N-1]_{\text{as}}^2 : d(x_i,x_j) \geq 1\right\}.$$

Hence, $\{x_n\}_{n=1}^N$ is an ordinally dense subset of $\mathbb{R}^d$. □

From Lemma 6, we want find an ordinally dense set of points $x_1, \ldots, x_N$ in $\mathbb{R}^d$ such that

$$\|x_n\| = 1, n \in [N-1] \text{ and } x_N = 0.$$

From the definition of ordinal spread, we have

$$\begin{aligned}
&\alpha_N(\{x_n\}_{n=1}^N) \\
&= \text{card}\left\{(i,j) : d(x_i,x_j) \geq d_{\max}(x_N, \{x_n\}_{n=1}^{N-1}), i,j \in [N-1], i > j\right\} + 1 \\
&= \text{card}\left\{(i,j) : \|x_i\|^2 + \|x_j\|^2 - 2x_i^\top x_j \geq 1^2, i,j \in [N-1], i > j\right\} + 1 \\
&= \text{card}\left\{(i,j) : \text{acos}(x_i^\top x_j) \geq \frac{\pi}{3}, i,j \in [N-1], i > j\right\} + 1.
\end{aligned}$$

We can find a maximum number of ordinally dense points by solving a spherical cap packing problem; see Figure E.2.

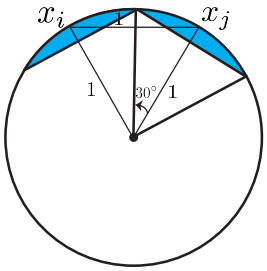


Figure E.2: Spherical $\frac{\pi}{6}$-cap packing on the surface of a unit sphere $\mathbb{S}^1$.

Definition 17. *Let $\mathbb{S}^{d-1}$ be the $(d-1)$-dimensional unit sphere in $\mathbb{R}^d$. We define the spherical α-cap $C_x(\alpha)$ as follows*

$$C_x(\alpha) = \left\{y \in \mathbb{S}^{d-1} : x^\top y < \cos(\alpha)\right\},$$

for any $x \in \mathbb{S}^{d-1}$.

Definition 18. *The maximum number of non-overlapping $C_x(\alpha)$ is defined as follows*

$$\begin{aligned}
N(\alpha) = \max_{N \in \mathbb{N}} \{N : \exists x_1, \ldots, x_N \in \mathbb{S}^{d-1} &\textit{such that} \\
&\bigcup_{j \in \mathcal{I}, j \neq i} C_{x_j}(\alpha) \cap C_{x_i}(\alpha) = \emptyset, \forall \mathcal{I} \subseteq [N], \forall i \in [N]\}.
\end{aligned}$$

Therefore, we have

$$
\begin{aligned}
K(\mathbb{R}^d) &= \sup\left\{\operatorname{card}\{x_n\} : \{x_n\} \sqsubseteq \mathbb{R}^d\right\} \\
&= \sup\left\{N : x_1, \ldots, x_N \in \mathbb{R}^d, \alpha_N\left(\{x_n\}_{n=1}^N\right) = \binom{N-1}{2} + 1\right\} \\
&= \sup\{N : x_1, \ldots, x_N \in \mathbb{R}^d \text{ such that} \\
&\qquad \operatorname{card}\left\{(i,j) \in [N-1]^2_{\text{as}} : \operatorname{acos}(x_i^\top x_j) \geq \frac{\pi}{3}\right\} = \binom{N-1}{2}\} \\
&= \sup\{N : x_1, \ldots, x_N \in \mathbb{R}^d \text{ such that} \\
&\qquad \operatorname{acos}(x_i^\top x_j) \geq \frac{\pi}{3}, i, j \in [N]^2_{\text{as}}\} + 1 \\
&\overset{\text{(a)}}{=} N(\frac{\pi}{6}) + 1 \\
&\overset{\text{(b)}}{\leq} [\sqrt{\frac{\pi}{8}} \frac{\Gamma\left(\frac{d-1}{2}\right)}{\Gamma\left(\frac{d}{2}\right) \int_0^{\frac{\pi}{4}} \sin^{d-2}\theta \left(\cos\theta - \frac{\sqrt{2}}{2}\right) d\theta}] + 1,
\end{aligned}
$$

where (a) follows from a simple illustration in Figure E.2, and (b) is given in [177]. For large d, Rankin provided the following approximation,

$$
N(\alpha) \sim \frac{(\frac{1}{2}\pi d^3 \cos 2\alpha)^{\frac{1}{2}}}{(\sqrt{2}\sin\alpha)^{d-1}}.
$$

Therefore, we have $N(\frac{\pi}{6}) \sim \sqrt{\pi} d^{\frac{3}{2}} 2^{\frac{d-3}{2}} = O(2^{\frac{d+3\log d}{2}})$. The maximum number of non-overlapping spherical caps of half angle θ which can be placed on the unit sphere in $\mathbb{R}^d$ is not less than $\exp(-d\log\sin 2\theta + o(d))$ [178]. Therefore, the lower bound on $N(\frac{\pi}{6})$ is given by $\exp(-d\log\frac{\sqrt{3}}{2} + o(d))$.

The centers of spherical caps in $\mathbb{R}^2$ form a regular hexagon; see Figure 6.3. Therefore, we have $K(\mathbb{R}^2) = 6 + 1 = 7$. However, these spherical caps overlap each other at exactly one point. Hence, the number of strictly non-overlapping spherical caps in $\mathbb{R}^2$ is 5. This leads to the pentagon configuration in Figure 6.4 (b).

E.2.3 Spherical Space

Lemma 7. *There is a set of points $x_1, \ldots, x_N$ in $\mathbb{S}^d$ such that*

$$d\left(x_n, x_N\right) = \operatorname{acos}\left(1-\epsilon\right), \forall n \in [N-1],$$

where $d_{\max}(x_N, \{x_n\}_{n=1}^{N-1}) \leq d_{\min}(\{x_n\}_{n=1}^{N-1})$, $N = K(\mathbb{S}^d)$, and for some $\epsilon \geq 0$.

Proof. Let $\{y_n\}_{n=1}^{N}$ be a set of points in $\mathbb{S}^d$ such that

$$d_{\max}(y_N, \{y_n\}_{n=1}^{N-1}) \leq d_{\min}(\{y_n\}_{n=1}^{N-1}),$$

or $\alpha_N(\{y_n\}_{n=1}^{N}) = \binom{N-1}{2}+1$. Without loss of generality, we assume $\alpha_N(\{y_n\}_{n=1}^{N}) = \binom{N-1}{2}+1$, $y_N = e_1$,[3] and $d_{\max}(y_N, \{y_n\}_{n=1}^{N-1}) = \operatorname{acos}\left(1-\epsilon\right)$. From the latter condition, we have

$$y_n \stackrel{\text{def}}{=} \begin{bmatrix} \sqrt{1-\|z_n\|^2} \\ z_n \end{bmatrix}, \text{ such that } \|z_n\| \leq \sqrt{1-(1-\epsilon)^2}.$$

Let us define

$$x_n = \begin{bmatrix} 1-\epsilon \\ \sqrt{1-(1-\epsilon)^2}\frac{1}{\|z_n\|} z_n \end{bmatrix}, \forall n \in [N-1]$$

and $x_N = e_1$. Then, we claim $\alpha_N(\{x_n\}_{n=1}^{N}) \geq \alpha_N(\{y_n\}_{n=1}^{N})$. Following the definition of ordinal spread, we have

$$\begin{aligned}
&\alpha_N(\{x_n\}_{n=1}^{N}) \\
&\stackrel{\text{(a)}}{=} \operatorname{card}\left\{(i,j) \in [N-1]_{\text{as}}^2 : d(x_i, x_j) \geq d_{\max}(x_N, \{x_n\}_{n=1}^{N-1})\right\} + 1 \\
&\stackrel{\text{(b)}}{=} \operatorname{card}\left\{(i,j) \in [N-1]_{\text{as}}^2 : d(x_i, x_j) \geq \operatorname{acos}\left(1-\epsilon\right), \right\} + 1 \\
&\stackrel{\text{(c)}}{\geq} \operatorname{card}\left\{(i,j) : d(y_i, y_j) \geq \operatorname{acos}\left(1-\epsilon\right), i,j \in [N-1]_{\text{as}}^2\right\} + 1 \\
&= \alpha_N(\{y_n\}_{n=1}^{N}),
\end{aligned}$$

where (a) holds with equality if x_N appears last in the sorted distance list,

[3] e_1 is the first standard base vector for $\mathbb{R}^{d+1}$.

(b) is due to $d_{\max}(x_N, \{x_n\}_{n=1}^{N-1}) = \operatorname{acos}(1-\epsilon) = d_{\max}(y_N, \{y_n\}_{n=1}^{N-1})$. For inequality (c), let $d(y_i, y_j) \geq \operatorname{acos}(1-\epsilon)$ for distinct $i, j \in [N-1]$ and $z_i^\top z_j = \|z_i\| \|z_j\| \cos\theta_{ij}$. Therefore, we have

$$\begin{aligned}
\cos\theta_{ij} &= \frac{1}{\|z_i\| \|z_j\|} z_i^\top z_j \\
&\overset{\text{(a)}}{\leq} \frac{1}{\|z_i\| \|z_j\|} \left(1 - \epsilon - \sqrt{1-\|z_i\|^2}\sqrt{1-\|z_j\|^2}\right) \\
&\overset{\text{(b)}}{\leq} 0,
\end{aligned}$$

where (a) is due to

$$y_i^\top y_j = \sqrt{1-\|z_i\|^2}\sqrt{1-\|z_j\|^2} + z_i^\top z_j \leq 1 - \epsilon$$

and inequality (b) is due $\sqrt{1-\|z_i\|^2} \geq \sqrt{1-\sqrt{1-(1-\epsilon)^2}^2} = \sqrt{1-\epsilon^2}$.[4] Then, since $(1-(1-\epsilon)^2)\cos\theta_{ij} \leq \|z_i\| \|z_j\| \cos\theta_{ij}$ if $\cos\theta_{ij} \leq 0$, we have

$$\begin{aligned}
d(x_i, x_j) &= \operatorname{acos}\left((1-\epsilon)^2 + (1-(1-\epsilon)^2)\cos\theta_{ij}\right) \\
&\geq \operatorname{acos}\left(\sqrt{1-\|z_i\|^2}\sqrt{1-\|z_j\|^2} + z_i^\top z_j\right) \\
&= d(y_i, y_j).
\end{aligned}$$

Therefore, we have

$$\left\{(i,j) \in [N-1]_{\text{as}}^2 : d(y_i, y_j) \geq \delta\right\} \subseteq \left\{(i,j) \in [N-1]_{\text{as}}^2 : d(x_i, x_j) \geq \delta\right\},$$

where $\delta = \operatorname{acos}(1-\epsilon)$. Hence, $\{x_n\}_{n=1}^N$ is an ordinally dense subset of $\mathbb{S}^d$. □

Now, let us find ordinally dense set of points $x_1, \ldots, x_N$ in $\mathbb{S}^d$ with

$$x_n = \begin{bmatrix} 1-\epsilon \\ z_n \end{bmatrix}, \forall n \in [N-1] \text{ and } x_N = e_1.$$

We have $\|z_n\|^2 = 1-(1-\epsilon)^2$ for all $\forall n \in [N-1]$. We begin from the definition

[4] Similarly, we have $\sqrt{1-\|z_j\|^2} \geq \sqrt{1-\epsilon^2}$.

of ordinal spread as follows

$$\begin{aligned}
&\alpha_N(\{x_n\}_{n=1}^N) \\
&= \operatorname{card}\left\{(i,j) \in [N-1]_{\text{as}}^2 : d(x_i,x_j) \geq d_{\max}(x_N, \{x_n\}_{n=1}^{N-1})\right\} + 1 \\
&= \operatorname{card}\left\{(i,j) \in [N-1]_{\text{as}}^2 : d(x_i,x_j) \geq \operatorname{acos}(1-\epsilon)\right\} + 1 \\
&= \operatorname{card}\left\{(i,j) \in [N-1]_{\text{as}}^2 : \frac{1}{\|z_i\|\,\|z_j\|} z_i^\top z_j \leq \frac{\epsilon(1-\epsilon)}{1-(1-\epsilon)^2}\right\} + 1 \\
&= \operatorname{card}\left\{(i,j) \in [N-1]_{\text{as}}^2 : \operatorname{acos}(\widehat{z}_i^\top \widehat{z}_j) \geq \frac{\pi}{3}\right\} + 1,
\end{aligned}$$

where $\widehat{z}_i = \frac{1}{\|z_i\|} z_i$, $\widehat{z}_j = \frac{1}{\|z_j\|} z_j$, and $\sup_\epsilon \frac{\epsilon(1-\epsilon)}{1-(1-\epsilon)^2} = \frac{1}{2}$. Similar to the Euclidean space, this problem is equivalent to spherical $\frac{\pi}{6}$-cap packing number in $\mathbb{R}^d$, since $\widehat{z}_n \in \mathbb{R}^d$. Finally, if we assume $\min_{i,j\in[N],i>j} d(x_i,x_j) = \delta$, we have $d_{\max}(x_N, \{x_n\}_{n=1}^{N-1}) \geq \delta$. Therefore, the cap angles can be computed as follows

$$\alpha = \min_{\epsilon \geq 1-\cos\delta} \frac{1}{2}\operatorname{acos}\frac{\epsilon(1-\epsilon)}{1-(1-\epsilon)^2} = \frac{1}{2}\operatorname{acos}\frac{\cos\delta}{1+\cos\delta} > \frac{\pi}{6}.$$

In this case, the ordinal capacity can be refined as spherical α-cap packing number.

E.3 Proof of Theorem 5

Let S be a d-dimensional space form, and $N \leq K(S)$. From Definition 4, we can find an ordinally dense subset $x_1, \ldots, x_N \in S$. Hence, we have

$$\begin{aligned}
A_N(S) &= \sup_{x_1,\ldots,x_N \in S} \alpha_N\left(\{x_n\}_{n=1}^N\right) \\
&\overset{\text{(a)}}{=} \binom{N-1}{2} + 1,
\end{aligned}$$

where (a) directly follows from Proposition 1. This is the number of edges of a complete graph with $N-1$ vertices plus one.

Now, let us consider $N > K(S)$. This could only happen in (d-dimensional) Euclidean and spherical spaces, since hyperbolic spaces have infinite ordinal capacity, i.e., $K(\mathbb{H}^d) = \infty$.

In Appendix E.2, we proved that there is a set of points $x_1, \ldots, x_{N-1} \in \mathbb{R}^d$ on the unit sphere and $x_N = 0$ such that

$$\begin{aligned} A_N(S) &= \alpha_N \left(\{x_n\}_{n=1}^N \right) \\ &= \text{card} \left\{ (i,j) : d(x_i, x_j) \geq 1, i, j \in [N-1], i > j \right\} + 1. \end{aligned}$$

Consider a pair of points $x_i, x_j \in \mathbb{R}^d$ with $d(x_i, x_j) < 1$. We can move the point x_i and place it on x_j if

$$\begin{aligned} \text{card} \left\{ (i,k) \in [N-1]_{\text{as}}^2 : d(x_i, x_k) \geq 1 \right\} \leq \\ \text{card} \left\{ (j,k) \in [N-1]_{\text{as}}^2 : d(x_j, x_k) \geq 1 \right\}. \end{aligned}$$

This condition is to ensure that we do not decrease $\alpha_N \left(\{x_n\}_{n=1}^N \right)$. We repeat this process and lump the set of $N-1$ point on $K < N-1$ positions, i.e., $p_1, \ldots, p_K$. At each position p_k, we place multiple vertices. Finally, $\alpha_N \left(\{x_n\}_{n=1}^N \right)$ is equal to the number of edges – with length greater than 1 – in this K-partite graph with $N-1$ vertices. This graph is K-partite because the distance between points in a partition have distances of zero. Hence, their edges do not contribute in calculating the ordinal spread of the point set. This graph becomes a complete K-partite graph if all distinct positions $\{p_k\}$ belong to the centers of spherical $\frac{\pi}{6}$-caps on the unit sphere. On the other hand, the number of edges in a complete K-partite graph is maximized when the size of the parts differs by at most one, i.e., Turán graph $T(N-1, K)$ [179]. Therefore, the N-point ordinal spread of S (Euclidean or spherical space) is given by

$$A_N(S) = \text{card } E(T(N-1, K(S)-1)) + 1.$$

The maximum number of possible partitions $(K(S)-1)$ gives the maximum number of edges, i.e.,

$$\text{card } E(T(N-1,1)) \leq \text{card } E(T(N-1,2)) \leq \cdots \leq \text{card } E(T(N-1, K(S)-1)).$$

This completes the proof.

E.4 Proof of Proposition 15

The proof follows from the definition of $A_N(S)$, the N-point ordinal spread of a space form S, in Theorem 5.

E.5 Numerical Experiments

All our experiments were conducted on a Dual-Core Intel Core i5 Mac machine, 16GB of system memory.

E.5.1 Datasets

We used cartographic data (counties in the state of Illinois, counties in Midwestern states,[5] and cities and towns across the world[6] and single-cell RNA expression data[7,8,9,10] [148, 149, 150, 185] which are publicly available datasets. Details of the single-cell expressions datasets are as follows:

1. *Lymphoma patient.* Human dissociated lymph node tumor cells of a 19-year-old male Hodgkins Lymphoma patient were obtained by 10x Genomics from Discovery Life Sciences. Whole transcriptome libraries were generated with Chromium Next GEM Single Cell 3' Reagent Kits v3.1 (Dual Index) User Guide (CG000315) and sequenced on an Illumina NovaSeq 6000. The targeted libraries were generated using the Targeted Gene Expression Reagent Kits User Guide (CG000293) and Human Immunology Panel reagent (PN-1000246) and sequenced on an Illumina NovaSeq 6000.

2. *Lymphoma-healthy donor.* Human peripheral blood mononuclear cells (PBMCs) of a healthy female donor aged 25 were obtained by 10x Genomics from AllCells. Whole transcriptome libraries were generated

[5] public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitude/.

[6] simplemaps.com/data/world-cities

[7] 10xgenomics.com/resources/datasets/hodgkins-lymphoma-dissociated-tumor-targeted-compare-immunology-panel-3-1-standard

[8] 10xgenomics.com/resources/datasets/pbm-cs-from-a-healthy-donor-targeted-immunology-panel-3-1-standard

[9] nature.com/articles/ncomms14049

[10] shiny.mdc-berlin.de/psca/

with Chromium Next GEM Single Cell 3' Reagent Kits v3.1 (Dual Index) User Guide (CG000315) and sequenced on an Illumina NovaSeq 6000. The aforementioned two datasets have 13410 samples (combined) and each for a class (binary classification). The dimension of each cell expression vector is 1020.

3. *Blood cells landmark.* We use the dataset originally from this paper and extract the gene expression data for (1) B cells, (2) Memory T cells, and (3) Native T cells. The complete dataset has 94655 samples, and the dimension of each cell expression vector is 965.

4. We use the single-cell RNA sequencing atlas provided in [185]. This atlas contains 26000 cell expression vectors for adult planarians. Each cell is a 21000-dimensional integer-valued vector representing read counts of gene expressions. Therefore, this raw data reside in a 21000-dimensional Euclidean space.

Imputations. Existing methods for denoising and imputation of raw scRNA-seq data often involve building connection graphs among cells [182, 181] using the distance between cells to diffuse the expression profiles among neighbor cells and smooth out possible outliers. In our experiment we used MAGIC [181] to impute our raw sequencing data with different number of neighbors and steps in the diffusion process to get different level of imputation results.

RFA score. For datasets $(1-3)$, we construct the five-nearest neighbbor graph, and set the kernel width (σ) to have an (soft) average of three neighbors; see Appendix E.5.3 for more detail on computing RFA scores.

E.5.2 Hyperbolicity of Trees

We generate random weighted trees with $N = 10^4$ nodes. The edge weights are drawn from i.i.d. uniform distribution in $[0, 1]$. The distance between each two nodes is the weight of the path joining them. We contaminate the corresponding distance matrix by an additive zero mean Gaussian noise with the signal to noise ratio of 40 dB. In this experiment, we consider three different trees with maximum degrees of $4, 5, 6$.[11] In Figure E.3, we show the distribution of node degrees for each tree.

[11] In the main manuscript, we only considered a binary tree with $\Delta(T) = 3$.

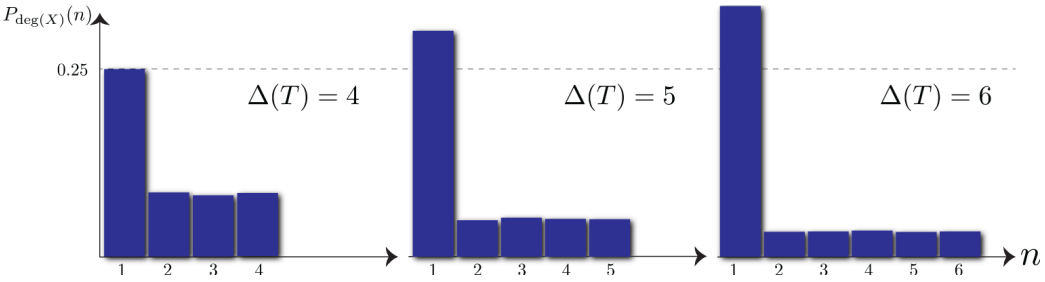


Figure E.3: The distribution of node degrees for each random tree.

We generate random points in space forms of dimension $d = 2, \ldots, 5$, from the following distributions

- Hyperbolic space: $x = [\sqrt{1 + \|z\|^2}, z^\top]^\top$, where $z \sim \mathcal{N}(0, \sigma^2 I)$ and $\sigma = 100$;
- Euclidean space: $x \sim \mathcal{N}(0, \sigma^2 I)$;
- Spherical space: $x = \frac{1}{\|z\|} z$, where $z \sim \mathcal{N}(0, I)$.[12]

Commonly, in embedding trees, the leaves concentrate near the boundary of the Poincaré disk. Hence, we choose a large variance σ to heavily sample the points closer to the boundary of Poincaré disk. Finally, we devise a hypothesis test based on the total variation distance of probability measures,[13] i.e.,

$$\delta(P, Q) = \|P - Q\|_1 .$$

For each tree T with $\Delta(T) = 4, 5$ and 6, we report the distances between the target (oracle) and empirical probability mass functions (PMF) of α_k for a set of N points generated in each space form. In Tables E.1 to E.3, we consider sub-cliques — randomly sampled from each tree — with $N = 20$ nodes. From the hypothesis tests for α_k, $k \in \{3, \ldots, 20\}$, we conclude that the ordinal spread variables of random trees better match with hyperbolic ordinal spread variables.

[12] Therefore, the points are distributed uniformly on $\mathbb{S}^d$.

[13] The total variation distance is $\delta(P, Q) = \frac{1}{2} \|P - Q\|_1$, but we can ignore the constant term.

Table E.1: $\delta(P_{\alpha_k}, \widehat{P}_{\alpha_k}) \times 10^{-3}$ for different space forms — $\Delta(T) = 4$.

k	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\mathbb{H}^2$	0	1.6	1.9	1.7	1.8	2.1	2.4	2.5	2.3	2.1	2.1	2.0	2.0	2.1	2.2	2.3	2.4	2.2
$\mathbb{H}^3$	0	2.3	2.8	2.5	2.4	2.5	2.9	3.0	2.9	2.6	2.5	2.3	2.2	2.0	1.9	1.8	1.8	1.7
$\mathbb{H}^4$	0	2.7	3.3	3.0	2.8	2.8	3.0	3.2	3.3	3.0	2.8	2.6	2.4	2.2	2.0	1.7	1.5	1.3
$\mathbb{H}^5$	0	3.0	3.6	3.4	3.0	3.1	3.2	3.4	3.5	3.3	3.0	2.8	2.6	2.3	2.1	1.8	1.5	1.1
$\mathbb{E}^2$	0	1.5	1.9	2.1	2.4	2.8	2.9	3.1	3.2	3.3	3.4	3.5	3.7	3.9	4.2	4.7	5.4	6.7
$\mathbb{E}^3$	0	2.2	2.7	2.6	2.8	3.4	3.5	3.7	3.7	3.8	3.9	4.0	4.1	4.1	4.3	4.5	4.9	5.9
$\mathbb{E}^4$	0	2.6	3.2	3.1	3.1	3.6	3.8	3.9	4.0	4.1	4.2	4.3	4.3	4.3	4.3	4.4	4.7	5.4
$\mathbb{E}^5$	0	2.8	3.5	3.5	3.3	3.8	3.9	4.1	4.1	4.2	4.4	4.4	4.4	4.3	4.3	4.4	4.6	5.1
$\mathbb{S}^2$	0	8.4	9.9	10.2	10.1	10	9.9	9.8	9.6	9.2	8.7	8.6	8.6	8.6	8.5	8.5	8.5	8.8
$\mathbb{S}^3$	0	8.4	9.8	10.2	10.1	10.1	10	9.9	9.6	9.3	8.8	8.7	8.7	8.7	8.7	8.7	8.7	8.8
$\mathbb{S}^4$	0	8.4	9.8	10.1	10.2	10.1	10	9.9	9.7	9.4	8.9	8.8	8.8	8.8	8.8	8.8	8.8	8.8
$\mathbb{S}^5$	0	8.5	9.8	10.1	10.2	10.1	10.1	9.9	9.7	9.4	8.9	8.9	8.9	8.9	8.8	8.8	8.8	8.7

Table E.2: $\delta(P_{\alpha_k}, \widehat{P}_{\alpha_k}) \times 10^{-3}$ for different space forms — $\Delta(T) = 5$.

k	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\mathbb{H}^2$	0	1.6	2.0	2.6	2.8	2, 8	2.8	2.6	2.6	2.5	2.4	2.2	2.1	1.9	1.7	1.4	1.1	0.7
$\mathbb{H}^3$	0	2.3	2.8	3.1	3.4	3.5	3.5	3.5	3.3	3.3	3.1	3.0	2.8	2.6	2.3	2.0	1.5	0.8
$\mathbb{H}^4$	0	2.7	3.3	3.4	3.7	3.9	3.9	3.9	3.8	3.7	3.6	3.5	3.3	3.1	2.8	2.5	2.0	1.2
$\mathbb{H}^5$	0	3.0	3.6	3.6	3.8	4.2	4.2	4.2	4.2	4.0	3.9	3.8	3.6	3.4	3.1	2.8	2.3	1.6
$\mathbb{E}^2$	0	1.5	2.3	3.4	3.6	4.0	4.2	4.4	4.6	4.8	5.0	5.3	5.6	5.9	6.3	6.9	7.5	8.4
$\mathbb{E}^3$	0	2.2	2.7	3.8	4.2	4.6	4.8	5.0	5.1	5.2	5.4	5.6	5.8	6.0	6.3	6.7	7.1	7.8
$\mathbb{E}^4$	0	2.6	3.2	3.9	4.5	4.9	5.0	5.2	5.4	5.5	5.6	5.8	5.9	6.1	6.3	6.6	6.9	7.3
$\mathbb{E}^5$	0	2.8	3.5	4.1	4.6	5.0	5.2	5.4	5.5	5.7	5.8	5.9	6.0	6.1	6.3	6.5	6.7	7.0
$\mathbb{S}^2$	0	8.4	9.9	10.2	10.1	10	9.9	9.8	9.6	9.5	9.5	9.5	9.5	9.5	9.5	9.5	9.6	9.7
$\mathbb{S}^3$	0	8.4	9.8	10.2	10.1	10.1	10	9.9	9.6	9.5	9.6	9.6	9.6	9.6	9.7	9.7	9.7	9.7
$\mathbb{S}^4$	0	8.4	9.8	10.1	10.2	10.1	10	9.9	9.7	9.6	9.6	9.7	9.7	9.7	9.7	9.8	9.8	9.7
$\mathbb{S}^5$	0	8.5	9.8	10.1	10.2	10.1	10.1	9.9	9.7	9.7	9.7	9.7	9.7	9.8	9.8	9.8	9.8	9.7

Note that we can also deign an aggregate hypothesis test based on $\boldsymbol{\alpha}_N$ by defining the following distance function between $P_{\boldsymbol{\alpha}_N}$ and $\widehat{P}_{\boldsymbol{\alpha}_N}$, e.g., $\delta\big(P_{\boldsymbol{\alpha}_N}, \widehat{P}_{\boldsymbol{\alpha}_N}\big) = \sum_{k=1}^{N} \delta(P_{\alpha_k}, \widehat{P}_{\alpha_k})$. This definition involves all ordinal spread variables related to sub-cliques of size N, i.e., α_k. Then, we can perform minimum-distance hypothesis tests for sub-cliques of sizes $N \in \{5, \dots 20\}$. For each experiment, hyperbolic spaces provide the best matches for ordinal

Table E.3: $\delta(P_{\alpha_k}, \widehat{P}_{\alpha_k}) \times 10^{-3}$ for different space forms — $\Delta(T) = 6$.

k	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\mathbb{H}^2$	0	1.6	2.5	3.0	3.2	3.2	3.1	2.9	2.9	2.7	2.5	2.3	2.1	1.9	1.6	1.3	1.1	1.4
$\mathbb{H}^3$	0	2.3	2.8	3.6	3.8	3.8	3.8	3.7	3.6	3.5	3.3	3.0	2.8	2.5	2.1	1.6	1.0	0.8
$\mathbb{H}^4$	0	2.7	3.3	3.9	4.0	4.2	4.2	4.2	4.0	3.9	3.7	3.5	3.2	2.9	2.5	1.9	1.2	0.4
$\mathbb{H}^5$	0	3.0	3.6	4.0	4.2	4.5	4.5	4.4	4.3	4.1	4.0	3.8	3.5	3.2	2.8	2.2	1.5	0.5
$\mathbb{E}^2$	0	1.5	2.8	3.8	4.0	4.3	4.5	4.6	4.7	4.9	5.0	5.2	5.4	5.6	5.8	6.2	6.7	7.5
$\mathbb{E}^3$	0	2.2	3.0	4.2	4.5	4.9	5.0	5.2	5.3	5.3	5.4	5.5	5.6	5.7	5.8	6.0	6.2	6.8
$\mathbb{E}^4$	0	2.6	3.2	4.4	4.8	5.2	5.3	5.4	5.5	5.6	5.6	5.7	5.7	5.8	5.8	5.9	6.0	6.3
$\mathbb{E}^5$	0	2.8	3.5	4.5	5.0	5.3	5.4	5.6	5.7	5.7	5.8	5.8	5.8	5.8	5.8	5.8	5.8	5.9
$\mathbb{S}^2$	0	8.4	9.9	10.2	10.1	10	9.9	9.8	9.6	9.4	9.4	9.4	9.4	9.3	9.3	9.3	9.2	9.3
$\mathbb{S}^3$	0	8.4	9.8	10.2	10.1	10.1	10	9.9	9.6	9.5	9.5	9.5	9.5	9.5	9.5	9.4	9.4	9.3
$\mathbb{S}^4$	0	8.4	9.8	10.1	10.2	10.1	10	9.9	9.7	9.6	9.6	9.6	9.6	9.5	9.5	9.5	9.4	9.3
$\mathbb{S}^5$	0	8.5	9.8	10.1	10.2	10.1	10.1	9.9	9.7	9.6	9.6	9.6	9.6	9.6	9.6	9.5	9.5	9.3

spread variables of each random tree; see Tables E.4 to E.6. This aggregate hypothesis test proves to more robustly reveal the hyperbolicity of weighted trees, compared to the individual tests based on ordinal spread variable α_N.

Table E.4: $\delta\big(P_{\boldsymbol{\alpha}_N}, \widehat{P}_{\boldsymbol{\alpha}_N}\big) \times 10^{-2}$ for different space forms — $\Delta(T) = 4$.

N	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\mathbb{H}^2$	3.3	3.7	3.9	4.1	4.2	4.2	4.1	4.1	4.0	4.0	3.9	3.8	3.8	3.7	3.6	3.6
$\mathbb{H}^3$	3.0	3.8	4.2	4.6	4.7	4.7	4.7	4.6	4.6	4.5	4.4	4.4	4.3	4.2	4.1	4.0
$\mathbb{H}^4$	3.4	4.1	4.7	5.0	5.2	5.2	5.1	5.1	5.0	4.9	4.8	4.8	4.7	4.5	4.4	4.4
$\mathbb{H}^5$	3.7	4.4	5.0	5.3	5.5	5.5	5.5	5.4	5.3	5.3	5.2	5.1	5.0	4.9	4.8	4.7
$\mathbb{E}^2$	7.0	8.0	8.3	8.4	8.2	8.0	7.8	7.6	7.3	7.1	6.8	6.6	6.4	6.2	6.0	5.9
$\mathbb{E}^3$	5.9	6.9	7.9	8.1	8.0	7.9	7.9	7.8	7.6	7.5	7.3	7.1	6.9	6.7	6.6	6.4
$\mathbb{E}^4$	5.2	6.6	7.6	7.9	7.9	7.9	8.0	8.0	7.8	7.7	7.5	7.3	7.2	7.0	6.9	6.7
$\mathbb{E}^5$	5.0	6.4	7.4	7.8	7.9	8.0	8.1	8.1	8.0	7.8	7.6	7.5	7.4	7.2	7.1	6.9
$\mathbb{S}^2$	12.0	15.5	17.8	19.4	20.2	20.3	20.2	19.8	19.3	18.8	18.3	17.7	17.1	16.6	16.1	15.6
$\mathbb{S}^3$	11.8	15.5	17.8	19.5	20.2	20.4	20.3	19.9	19.5	18.9	18.4	17.9	17.3	16.8	16.2	15.7
$\mathbb{S}^4$	11.7	15.4	17.8	19.5	20.1	20.4	20.3	20.0	19.6	19.0	18.5	17.9	17.4	16.9	16.3	15.8
$\mathbb{S}^5$	11.6	15.4	17.8	19.4	20.1	20.4	20.4	20.0	19.6	19.1	18.6	18.0	17.5	16.9	16.4	15.9

Table E.5: $\delta\big(P_{\boldsymbol{\alpha}_N}, \widehat{P}_{\boldsymbol{\alpha}_N}\big) \times 10^{-2}$ for different space forms — $\Delta(T) = 5$.

N	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\mathbb{H}^2$	5.4	5.5	5.3	5.4	5.2	5.1	4.9	4.8	4.5	4.4	4.2	4.1	3.9	3.8	3.7	3.6
$\mathbb{H}^3$	4.5	4.8	5.4	5.9	5.9	5.9	5.8	5.7	5.5	5.4	5.3	5.2	5.1	4.9	4.8	4.7
$\mathbb{H}^4$	4.1	5.0	5.9	6.3	6.4	6.4	6.4	6.3	6.2	6.2	6.1	5.9	5.8	5.7	5.5	5.4
$\mathbb{H}^5$	4.2	5.2	6.2	6.6	6.8	6.8	6.8	6.8	6.8	6.7	6.6	6.4	6.3	6.2	6.1	5.9
$\mathbb{E}^2$	9.3	10.8	11.5	11.5	11.3	11.1	10.9	10.7	10.3	10	9.7	9.4	9.1	8.9	8.6	8.4
$\mathbb{E}^3$	8.0	9.6	10.6	11.0	11.0	11.1	11.0	10.8	10.5	10.3	10.0	9.7	9.5	9.3	9.0	8.8
$\mathbb{E}^4$	7.4	9.0	10.3	10.8	10.9	11.0	11.0	10.8	10.6	10.4	10.1	9.9	9.7	9.5	9.3	9.1
$\mathbb{E}^5$	6.9	8.7	10.1	10.6	10.8	10.9	10.9	10.8	10.6	10.4	10.2	10.0	9.8	9.6	9.4	9.2
$\mathbb{S}^2$	13.6	17.2	19.6	21.0	21.7	21.8	21.6	21.2	20.6	20.0	19.4	18.8	18.1	17.5	16.9	16.4
$\mathbb{S}^3$	13.4	17.3	19.8	21.1	21.8	21.9	21.8	21.3	20.8	20.2	19.6	18.9	18.3	17.7	17.1	16.5
$\mathbb{S}^4$	13.3	17.3	19.8	21.1	21.8	22.0	21.8	21.4	20.9	20.3	19.7	19.0	18.4	17.8	17.1	16.6
$\mathbb{S}^5$	13.1	17.3	19.7	21.1	21.9	22.0	21.8	21.5	20.9	20.3	19.7	19.0	18.4	17.8	17.2	16.6

Table E.6: $\delta\big(P_{\boldsymbol{\alpha}_N}, \widehat{P}_{\boldsymbol{\alpha}_N}\big) \times 10^{-2}$ for different space forms — $\Delta(T) = 6$.

N	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\mathbb{H}^2$	5.8	5.8	5.9	5.8	5.7	5.5	5.2	5.0	4.9	4.7	4.6	4.4	4.3	4.2	4.1	3.9
$\mathbb{H}^3$	4.8	5.2	5.7	5.9	6.1	6.1	5.9	5.8	5.7	5.5	5.4	5.3	5.2	5.0	4.9	4.8
$\mathbb{H}^4$	4.5	5.0	5.9	6.4	6.6	6.5	6.4	6.3	6.3	6.2	6.0	5.9	5.8	5.6	5.5	5.4
$\mathbb{H}^5$	4.3	5.1	6.2	6.7	6.9	6.9	6.8	6.7	6.7	6.6	6.5	6.3	6.2	6.1	6.0	5.8
$\mathbb{E}^2$	9.6	11.1	11.7	11.7	11.4	11.1	10.8	10.5	10.2	9.9	9.6	9.3	9.0	8.7	8.5	8.3
$\mathbb{E}^3$	8.3	9.9	10.8	11.0	10.9	10.9	10.8	10.6	10.4	10.1	9.8	9.6	9.3	9.1	8.9	8.7
$\mathbb{E}^4$	7.7	9.2	10.3	10.7	10.8	10.8	10.8	10.6	10.4	10.2	10.0	9.7	9.5	9.3	9.1	8.9
$\mathbb{E}^5$	7.2	8.8	10.0	10.5	10.6	10.8	10.8	10.6	10.4	10.2	10.0	9.8	9.6	9.4	9.2	9.0
$\mathbb{S}^2$	13.3	17.0	19.3	20.6	21.3	21.4	21.2	20.8	20.3	19.7	19.1	18.5	17.9	17.3	16.7	16.2
$\mathbb{S}^3$	13.1	17.1	19.4	20.7	21.4	21.5	21.4	20.9	20.5	19.9	19.3	18.7	18.0	17.4	16.9	16.3
$\mathbb{S}^4$	13.0	17.1	19.4	20.7	21.4	21.6	21.4	21.0	20.5	20.0	19.4	18.7	18.1	17.5	16.9	16.4
$\mathbb{S}^5$	12.9	17.1	19.4	20.7	21.4	21.6	21.5	21.1	20.6	20.0	19.4	18.8	18.2	17.6	17.0	16.4

We generate a random tree T with $N = 10^4$ nodes, maximum degree of Δ, and i.i.d. edge weights from unif$(0, 1)$. Let $\widetilde{D}_\Delta = D_\Delta + n$, where n is a zero mean Gaussian noise with 40 decibel signal-to-noise ratio, be the noisy distance matrix for T. The embedding goal is to find a representation $x_1, \ldots, x_N$ for tree nodes in S, such that

$$d(x_i, x_j) \leq d(x_k, x_l) \Longleftrightarrow \widetilde{D}_\Delta(i, j) \leq \widetilde{D}_\Delta(k, l).$$

We randomly select 10^6 sub-cliques of sizes $N \in \{2, 4, \ldots, 20, 100\}$. In Table E.7, we give the empirical N-th ordinal spread based on nonmetric measurements associated with the sub-cliques, i.e., $\widehat{A}_N$. The distribution-free test gives a lower bound of $\hat{d} \geq 4$ for Euclidean embedding dimension.

On the other hand, consider a random weighted tree and a node x_n with degree Δ_n.[14] We can easily see that

$$\max_{i \in [\Delta_n]} d(x_n, x_{n_i}) \leq \min_{\substack{i,j \in [\Delta_n] \\ i \neq j}} d(x_{n_i}, x_{n_j}),$$

where $x_{n_1}, \ldots, x_{n_\Delta}$ are adjacent points to x_n. Hence, $\{x_n\} \cup \{x_{n_i}\}_{i=1}^{\Delta_n}$ is a set of $\Delta_n + 1$ points with maximum ordinal spread. Therefore, a lower bound for embedding dimension of a metric tree T (in Euclidean space) is given by

$$\hat{d} \geq \min \left\{ d : K(\mathbb{R}^d) \geq \Delta(T) + 1 \right\}.$$

The exponential growth of ρ_d gives $\hat{d} = \Omega(\log \Delta(T))$.

Remark. In absence of any prior information for proper distributions of data points, the estimate for the dimension of underlying space form is unreliable. The statistics of the ordinal spread variables are *invariant* with respect isotonic transformation of data points, e.g., rotation, translation, and uniform scaling in Euclidean space.

Fact 5. *Let $\{x_n\}_{n=1}^N$ be a set of points in (S, d). The ordinal spread vector is invariant with respect to strongly isotonic transformation [171] of points. In other words, let $\psi : S \to S$ be an arbitrary function such that for all*

[14] We assume the existence of a perfect embedding.

Table E.7: The N-point ordinal spread for $\mathbb{E}^2, \mathbb{E}^3, \mathbb{E}^4$ versus $\widehat{A}_N$ estimated from $\widetilde{D}_4, \widetilde{D}_5$ and $\widetilde{D}_6$.

N	6	8	10	12	14	16	18	20	100
$\widetilde{D}_4 : \widehat{A}_N$	11	22	37	56	79	106	137	169	4421
$\widetilde{D}_5 : \widehat{A}_N$	11	22	37	56	79	106	136	172	4412
$\widetilde{D}_6 : \widehat{A}_N$	11	22	37	56	79	106	137	170	4454
$A_M(\mathbb{E}^2)$	11	21	34	51	71	94	121	151	4048
$A_N(\mathbb{E}^3)$	11	22	37	56	79	106	135	168	4573
$A_N(\mathbb{E}^4)$	11	22	37	56	79	106	137	172	4741

$x, y, z, w \in S$ we have

$$d(x,y) < d(z,w) \Rightarrow d\big(\psi(x),\psi(y)\big) < d\big(\psi(z),\psi(w)\big)$$
$$d(x,y) = d(z,w) \Rightarrow d\big(\psi(x),\psi(y)\big) = d\big(\psi(z),\psi(w)\big)$$

then, $\boldsymbol{\alpha}\left(\{x_n\}_{n=1}^N\right) = \boldsymbol{\alpha}\left(\{\psi(x_n)\}_{n=1}^N\right)$.

Therefore, we can also use compact distributions, e.g., the multivariate uniform distribution. The arbitrary choices of Gaussian and uniform distributions do not significantly change the statistics of the ordinal spread variables — at least, it does not affect the key results in this experiment.

E.5.3 Single-cell RNA Expression Data

We use the single-cell RNA sequencing atlas provided in [185]. This atlas contains 26000 cell expression vectors for adult planarians. Each cell is an integer-valued vector representing read counts of gene expressions. The specific choices of pre-processing method and the comparison criteria *imply* a geometry — namely, geometry of similarity comparisons — that is not necessarily related to the domain of data vectors. The choice of comparison is the relative forest accessibility score:

Relative forest accessibility (RFA) index: For a set of points $x_1 \ldots, x_N$, we construct the local connectivity edge set E from a symmetric k-nearest neighbor method. The relative forest accessibility matrix is a $N \times N$ doubly stochastic matrix defined as $P = (I + L)^{-1}$ where $L = D - A$ is the Laplacian

matrix, $A = (A_{i,j})$ such that

$$A_{i,j} = \exp\big(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\big)[(i,j) \in E],$$

where the Iverson bracket $[(i,j) \in E] = 1$ if $(i,j) \in E$ and is 0 otherwise, and D is a diagonal matrix with $D_{ii} = \sum_{j \in [N]} A_{i,j}$. The ij-th element of P is the probability of a spanning forest includes a rooted tree at x_i and is connected to x_j — a measure of similarity between x_i and x_j [32]. In this experiment, we let $\sigma = \frac{1}{\sqrt{10}N^2} \sum_{i,j \in [N]} \|x_i - x_j\|$ and ignore the hard edge assignment since the conservative choice of kernel width performs a soft edge assignment. For a fast implementation of $P = (I + L)^{-1}$, we approximate the weighted adjacency matrix $A \in \mathbb{R}^{N \times N}$ with a rank-500 semidefinite matrix — via a simple eigenvalue thresholding — and use Woodbury matrix identity to compute P. The points x_i, x_j are more similar than x_k, x_l if the relative forest accessibility index $p_{i,j}$ is greater than $p_{k,l}$. The geometry of RFA comparisons is unknown.

For embedding ordinal measurements, we pick a random clique of size 200 and embed it in low-dimensional space forms of different dimensions. Then, we compute the empirical probability of erroneous comparison, i.e., error occurs if $d(x_i, x_j) \geq d(x_k, x_l)$ whereas the points x_i, x_j are more similar to each other compared to the points x_k, x_l. We repeat the experiment 200 times, and report the mean and standard deviations of the probability of error p_e. An important observation is that higher dimensional of space forms do not necessarily give better matches for the empirical PMF of ordinal spread variables.

E.6 Nonmetric Embedding Algorithms

We can use semidefinite programs to solve nonmetric embedding problems in hyperbolic and Euclidean spaces [19, 52]. The main objects in these problems are distance matrices, and the matrix of inner products, e.g., Gramian in Euclidean space and Lorentzian matrix in hyperbolic space. The traditional interior point method to solve semidefinite programs do not scale to large problems. This is especially the case for nonmetric embedding problems in which we have $\binom{\binom{N}{2}}{2} = O(N^4)$ distinct inequality constraints related to

pairwise distance comparisons. Therefore, we propose nonmetric embedding algorithms based on the method of alternative projections; see Algorithms 11 to 13.

E.6.1 Hyperbolic Embedding

We start with an arbitrary hyperbolic distance matrix (refer to [19]), and a sorted index list. The function $\texttt{IndexList}(D)$ computes the index list associated with the distance matrix D.

We begin with arranging the elements of D according to the target index list (i, j). In other words, we have

$$\texttt{sort}(D, (i, j)) = (d_{\pi(i_r, j_r)})_{i_r, j_r \in [N]}$$

where $\pi : [N]^2 \to [N^2]$ is a one-to-one map, such that $\pi(i_r, j_r) = \pi(j_r, i_r)$, $\pi(i_r, i_r) = (i_r, i_r)$, and $\texttt{IndexList}(\texttt{sort}(D, (i, j))) = (i, j)$. The resulting symmetric matrix is no longer a valid hyperbolic distance matrix. Therefore, we proceed with finding the best rank-$(d+1)$ Lorentzian matrix— the matrix of Lorentizan inner products. We compute the corresponding point set, in $\mathbb{R}^{d+1}$, by a simple spectral factorization of the Lorentizan matrix; see Algorithm 11 lines $6-8$ and refer to [19]. Finally, we use a simple method to map each point (columns of X) to $\mathbb{L}^d$, viz.,

$$P_{\mathbb{R}^{d+1} \to \mathbb{L}^d}(x) = \begin{bmatrix} \sqrt{1 + \|y\|^2} \\ y \end{bmatrix} \text{ where } y = (x_2, \ldots x_{d+1})^\top.$$

We compute the hyperbolic Gramian, $G = X^\top H X$, where $H = \text{diag}(-1, 1, \ldots, 1) \in \mathbb{R}^{(d+1)\times(d+1)}$. This gives us the update for hyperbolic distance matrix $D = \text{acosh}[-G]$; refer to [19]. We repeat this process till convergence.

Algorithm 11 Non-metric hyperbolic embedding.

input: Index list (i, j), and embedding dimension d.
initialize: a hyperbolic distance matrix D, and an arbitrary index list $(\tilde{i}, \tilde{j})$.
while $\left\|(\tilde{i}, \tilde{j}) - \texttt{IndexList}(D)\right\| > 0$ **do**
  $(\tilde{i}, \tilde{j}) \leftarrow \texttt{IndexList}(D)$. *The index list related to* D
  $D \leftarrow \texttt{sort}(D, (i, j))$. *Update* D *by sort distances according to* (i, j)
  Let $U\Sigma U^\top$ be the eigenvalue decomposition of $G = -\cosh[D]$ such that $\sigma_1 \geq \ldots \geq \sigma_N \in \mathbb{R}$.
  $X = |\Sigma_d|^{1/2} U_d^\top$, where $\Sigma_d = \text{diag}[(\sigma_1)_+, \ldots, (\sigma_d)_+, (\sigma_N)_-]$ and U_d is the sliced eigenvector matrix.
  $X \leftarrow P_{\mathbb{R}^{d+1} \to \mathbb{L}^d}(X)$. *Map each column of* $X \in \mathbb{R}^{(d+1)\times N}$ *to* $\mathbb{L}^d$
  $G = X^\top H X$. *Hyperbolic Gramian*
  $D \leftarrow \text{acosh}[-G]$. *Hyperbolic distance matrix*
end while
return X

E.6.2 Spherical Embedding

We propose a similar method for spherical embedding. The matrix of inner products G and the spherical distance matrix D are related via $D = \text{acos}[G]$. For points in d-dimensional spherical space, the matrix G is a positive semidefinite matrix of rank $(d + 1)$, and with diagonal elements of 1. The spectral factorization of G gives us the point positions.

Algorithm 12 Non-metric spherical Embedding.

input: Index list (i, j), and embedding dimension d.
initialize: A spherical distance matrix D, and an arbitrary index list $(\tilde{i}, \tilde{j})$.
while $\left\|(\tilde{i}, \tilde{j}) - \texttt{IndexList}(D)\right\| > 0$ **do**
  $(\tilde{i}, \tilde{j}) = \texttt{IndexList}(D)$. *The index list related to* D
  $D \leftarrow \texttt{sort}(D, (i, j))$. *Update* D *by sort distances according to* (i, j)
  Let $U\Sigma U^\top$ be eigenvalue decomposition of $G = \cos[D]$ such that $\sigma_1 \geq \ldots \geq \sigma_N \in \mathbb{R}$.
  Let $\Sigma_d = \text{diag}[(\sigma_1)_+, \ldots, (\sigma_{d+1})_+]$, and U_d be corresponding eigenvector matrix.
  $X = \Sigma_d^{1/2} U_d^\top$.
  $X \leftarrow P_{\mathbb{R}^{d+1} \to \mathbb{S}^d}(X)$. *Map each column of* $X \in \mathbb{R}^{(d+1)\times N}$ *to* $\mathbb{S}^d$
  $G = X^\top X$. *Gram matrix*
  $D \leftarrow \text{acos}[G]$.
end while
return X

E.6.3 Euclidean Embedding

Unlike hyperbolic and spherical counterparts, Euclidean distance matrix $D \in \mathbb{R}^{N\times N}$ is the matrix of squared distances between a set of N points $X \in \mathbb{R}^{d\times N}$. This definition lets us to express it as a linear function of the Gram matrix $G = X^\top X$, i.e., $D = \mathcal{K}(G) = -2G + \text{diag}(G)1^\top + 1\text{diag}(G)^\top$, where $\text{diag}(G)$ is a vector of diagonal elements of G, and $1 \in \mathbb{R}^N$ is the vector of all ones. The Gram matrix G is positive semidefinite of rank at most d. We can find the centered Gramian from a given distance matrix as $G = -\frac{1}{2}JDJ$, where $J = I - \frac{1}{N}11^\top$. At each iteration of Algorithm 13, we find the best rank-d positive semidefinite matrix via a simple eigenvalue thresholding of $G = -\frac{1}{2}JDJ$; see lines $7 - 8$ of Algorithm 13. The spectral factorization of G gives the point set in $\mathbb{R}^d$. We repeat this process until convergence.

Algorithm 13 Non-metric Euclidean embedding.

input: Index list (i, j), and embedding dimension d.
initialize: A Euclidean distance matrix D, and an arbitrary index list $(\tilde{i}, \tilde{j})$.
while $\left\|(\tilde{i}, \tilde{j}) - \texttt{IndexList}(D)\right\| > 0$ **do**
  $(\tilde{i}, \tilde{j}) \leftarrow \texttt{IndexList}(D)$. *The index list related to D*
  $D = \texttt{sort}(D, (i, j))$.
  Let $U\Sigma U^\top$ be the eigenvalue decomposition of $G = -\frac{1}{2}JDJ$ such that $\sigma_1 \geq \ldots \geq \sigma_N \in \mathbb{R}$.
  $G = U_d \Sigma_d U_d^\top$, where $\Sigma_d = \text{diag}[(\sigma_1)_+, \ldots, (\sigma_d)_+]$ and U_d is the sliced eigenvector matrix.
  $D \leftarrow \mathcal{K}(G)$. *Euclidean distance matrix*
end while
return $X = \Sigma_d^{1/2} U_d^\top$.

REFERENCES

[1] K. Menger, "Untersuchungen über allgemeine Metrik," *Mathematische Annalen*, vol. 100, no. 1, pp. 75–163, 1928.

[2] I. J. Schoenberg, "Remarks to Maurice Frechet's Article 'Sur La Definition Axiomatique D'Une Classe D'Espace Distances Vectoriellement Applicable Sur L'Espace De Hilbert'," *Annals of Mathematics*, pp. 724–732, 1935.

[3] L. M. Blumenthal, *Theory and Applications of Distance Geometry*. Clarendon Press, 1953.

[4] M. Browne, "The Young-Householder algorithm and the least squares multidimensional scaling of squared distances," *Journal Classification*, vol. 4, no. 2, pp. 175–190, 1987.

[5] I. Dokmanić, R. Parhizkar, J. Ranieri, and M. Vetterli, "Euclidean distance matrices: Essential theory, algorithms, and applications," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 12–30, 2015.

[6] J. M. Porta, L. Ros, F. Thomas, and C. Torras, "A branch-and-prune solver for distance constraints," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 176–187, 2005.

[7] P. Tabaghi, I. Dokmanić, and M. Vetterli, "Kinetic Euclidean distance matrices," *IEEE Transactions on Signal Processing*, vol. 68, pp. 452–465, 2019.

[8] A. M.-C. So and Y. Ye, "Theory of semidefinite programming for sensor network localization," *Mathematical Programming*, vol. 109, no. 2-3, pp. 367–384, 2007.

[9] G. M. Crippen and T. F. Havel, *Distance Geometry and Molecular Conformation*, vol. 74. Taunton: Research Studies Press, 1988.

[10] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, "Euclidean distance geometry and applications," *SIAM Review*, vol. 56, no. 1, pp. 3–69, 2014.

[11] N. Krislock and H. Wolkowicz, "Euclidean distance matrices and applications," in *Handbook on Semidefinite, Conic and Polynomial Optimization*, pp. 879–914, Boston, MA: Springer, Jan. 2012.

[12] A. M.-C. So and Y. Ye, "Theory of semidefinite programming for sensor network localization," *Mathematical Programming*, vol. 109, pp. 367–384, Mar. 2007.

[13] A. Cornejo and R. Nagpal, "Distributed range-based relative localization of robot swarms," in *Algorithmic Foundations of Robotics XI*, pp. 91–107, Springer, 2015.

[14] J. Matthaei, T. Krüger, S. Nowak, and U. Bestmann, "Swarm exploration of unknown areas on Mars using SLAM," in *International Micro Air Vehicle Conference and Flight Competition*, 2013.

[15] J. S. Jaffe, P. J. Franks, P. L. Roberts, D. Mirza, C. Schurgers, R. Kastner, and A. Boch, "A swarm of autonomous miniature underwater robot drifters for exploring submesoscale ocean dynamics," *Nature Communications*, vol. 8, p. 14189, 2017.

[16] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[17] M. Kreković, I. Dokmanić, and M. Vetterli, "EchoSLAM: Simultaneous localization and mapping with acoustic echoes," in *Proceedings of the International Conference on Acoustics, Speech, & Signal Processing*, pp. 11–15, IEEE, 2016.

[18] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[19] P. Tabaghi and I. Dokmanić, "Hyperbolic distance matrices," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1728–1738, 2020.

[20] P. Tabaghi and I. Dokmanić, "Geometry of comparisons," *arXiv preprint arXiv:2006.09858*, 2020.

[21] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Advances in Neural Information Processing Systems*, pp. 6338–6347, 2017.

[22] F. Sala, C. De Sa, A. Gu, and C. Ré, "Representation tradeoffs for hyperbolic embeddings," in *Proceedings of the International Conference on Machine Learning*, pp. 4460–4469, PMLR, 2018.

[23] J. Lamping and R. Rao, "Laying out and visualizing large trees using a hyperbolic space," in *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, pp. 13–14, ACM, 1994.

[24] R. Sarkar, "Low distortion delaunay embedding of trees in hyperbolic plane," in *International Symposium on Graph Drawing*, pp. 355–366, Springer, 2011.

[25] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky, "Hyperbolic image embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6418–6428, 2020.

[26] Y. Zhou, B. H. Smith, and T. O. Sharpee, "Hyperbolic geometry of the olfactory space," *Science Advances*, vol. 4, no. 8, p. eaaq1458, 2018.

[27] Y. Meng, J. Huang, G. Wang, C. Zhang, H. Zhuang, L. Kaplan, and J. Han, "Spherical text embedding," in *Advances in Neural Information Processing Systems*, pp. 8208–8217, 2019.

[28] A. Gu, F. Sala, B. Gunel, and C. Ré, "Learning mixed-curvature representations in product spaces," in *Proceedings of the International Conference on Learning Representations*, 2018.

[29] S. Bai, H.-D. Qi, and N. Xiu, "Constrained best Euclidean distance embedding on a sphere: A matrix optimization approach," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 439–467, 2015.

[30] A. Elad, Y. Keller, and R. Kimmel, "Texture mapping via spherical multi-dimensional scaling," in *International Conference on Scale-Space Theories in Computer Vision*, pp. 443–455, Springer, 2005.

[31] A. Tanay and A. Regev, "Scaling single-cell genomics from phenomenology to mechanism," *Nature*, vol. 541, no. 7637, pp. 331–338, 2017.

[32] A. Klimovskaia, D. Lopez-Paz, L. Bottou, and M. Nickel, "Poincaré maps for analyzing complex hierarchies in single-cell data," *Nature Communications*, vol. 11, no. 1, pp. 1–9, 2020.

[33] O. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic entailment cones for learning hierarchical embeddings," in *Proceedings of the International Conference on Machine Learning*, pp. 1646–1655, PMLR, 2018.

[34] C. De Sa, A. Gu, C. Ré, and F. Sala, "Representation tradeoffs for hyperbolic embeddings," vol. 80, p. 4460, NIH Public Access, 2018.

[35] K. Chowdhary and T. G. Kolda, "An improved hyperbolic embedding algorithm," *Journal of Complex Networks*, vol. 6, no. 3, pp. 321–341, 2018.

[36] M. Nickel and D. Kiela, "Learning continuous hierarchies in the Lorentz model of hyperbolic geometry," *arXiv preprint arXiv:1806.03417*, 2018.

[37] M. Le, S. Roller, L. Papaxanthos, D. Kiela, and M. Nickel, "Inferring concept hierarchies from text corpora via hyperbolic embeddings," *arXiv preprint arXiv:1902.00913*, 2019.

[38] S. Roller, D. Kiela, and M. Nickel, "Hearst patterns revisited: Automatic hypernym detection from large text corpora," *arXiv preprint arXiv:1806.03191*, 2018.

[39] G. Bachmann, G. Bécigneul, and O. Ganea, "Constant curvature graph convolutional networks," in *Proceedings of the International Conference on Machine Learning*, pp. 486–496, PMLR, 2020.

[40] O.-E. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," *arXiv preprint arXiv:1805.09112*, 2018.

[41] I. Chami, Z. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 4868–4879, 2019.

[42] A. Tifrea, G. Bécigneul, and O.-E. Ganea, "Poincaré GloVe: Hyperbolic word embeddings," *arXiv preprint arXiv:1810.06546*, 2018.

[43] Q. Liu, M. Nickel, and D. Kiela, "Hyperbolic graph neural networks," in *Advances in Neural Information Processing Systems*, pp. 8230–8241, 2019.

[44] R. Shimizu, Y. Mukuta, and T. Harada, "Hyperbolic neural networks++," *arXiv preprint arXiv:2006.08210*, 2020.

[45] J. Dai, Y. Wu, Z. Gao, and Y. Jia, "A hyperbolic-to-hyperbolic graph convolutional network," *arXiv preprint arXiv:2104.06942*, 2021.

[46] C. Giusti, E. Pastalkova, C. Curto, and V. Itskov, "Clique topology reveals intrinsic geometric structure in neural correlations," *in Proceedings of the National Academy of Sciences*, vol. 112, no. 44, pp. 13455–13460, 2015.

[47] Ç. Demiralp, M. S. Bernstein, and J. Heer, "Learning perceptual kernels for visualization design," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1933–1942, 2014.

[48] D. J. Navarro and M. D. Lee, "Common and distinctive features in stimulus similarity: A modified version of the contrast model," *Psychonomic Bulletin & Review*, vol. 11, no. 6, pp. 961–974, 2004.

[49] R. N. Shepard, "The analysis of proximities: Multidimensional scaling with an unknown distance function. I.," *Psychometrika*, vol. 27, no. 2, pp. 125–140, 1962.

[50] R. N. Shepard, "The analysis of proximities: Multidimensional scaling with an unknown distance function. II.," *Psychometrika*, vol. 27, no. 3, pp. 219–246, 1962.

[51] J. B. Kruskal, "Nonmetric multidimensional scaling: A numerical method," *Psychometrika*, vol. 29, no. 2, pp. 115–129, 1964.

[52] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. Kriegman, and S. Belongie, "Generalized non-metric multidimensional scaling," in *Artificial Intelligence and Statistics*, pp. 11–18, 2007.

[53] G. Carlsson, "Topology and data," *Bulletin of the American Mathematical Society*, vol. 46, no. 2, pp. 255–308, 2009.

[54] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, "Euclidean distance geometry and applications," *SIAM Review*, vol. 56, no. 1, pp. 3–69, 2014.

[55] P. Biswas, T. C. Liang, K. C. Toh, Y. Ye, and T. C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 4, pp. 360–371, 2006.

[56] N. Krislock and H. Wolkowicz, "Explicit sensor network localization using semidefinite representations and facial reductions," *SIAM Journal on Optimization*, vol. 20, pp. 2679–2708, Jan. 2010.

[57] S. Bai and H. Qi, "Tackling the flip ambiguity in wireless sensor network localization and beyond," *Digital Signal Processing*, vol. 55, pp. 85–97, July 2016.

[58] I. Dokmanić, J. Ranieri, and M. Vetterli, "Relax and unfold: Microphone localization with Euclidean distance matrices," in *Proceedings of the European Signal Processing Conference*, pp. 265–269, IEEE, 2015.

[59] I. Dokmanić and M. Vetterli, "Room helps: Acoustic localization with finite elements," in *International Conference on Acoustics, Speech, & Signal Processing*, pp. 2617–2620, IEEE, 2012.

[60] I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, "Acoustic echoes reveal room shape," *in Proceedings of the National Academy of Sciences*, vol. 110, no. 30, pp. 12186–12191, 2013.

[61] R. Parhizkar, I. Dokmanić, and M. Vetterli, "Single-channel indoor microphone localization," in *Proceedings of the International Conference on Acoustics, Speech, & Signal Processing*, pp. 1434–1438, IEEE, 2014.

[62] A. Singer, "A remark on global positioning from local distances," *in Proceedings of the National Academy of Sciences*, vol. 105, no. 28, pp. 9507–9511, 2008.

[63] L. Liberti and C. Lavor, "Open research areas in distance geometry," in *Open Problems in Optimization and Data Analysis*, pp. 183–223, Springer, 2018.

[64] B. Hendrickson, "The molecule problem: Exploiting structure in global optimization," *SIAM Journal on Optimization*, vol. 5, no. 4, pp. 835–857, 1995.

[65] B. A. Hendrickson, "The molecule problem: Determining conformation from pairwise distances," tech. rep., Cornell University, 1990.

[66] L. J. Guibas, "Kinetic data structures—A state of the art report," tech. rep., Stanford University, 1998.

[67] Q. Wu, A. Tinka, K. Weekly, J. Beard, and A. M. Bayen, "Variational Lagrangian data assimilation in open channel networks," *Water Resources Research*, vol. 51, pp. 1916–1938, Apr. 2015.

[68] J. Morales, P. Roysdon, Z. M. Kassas, and 2016, "Signals of opportunity aided inertial navigation," in *Proceedings of the International Technical Meeting of the Satellite Division of The Institute of Navigation*, (Portland, OR), 2016.

[69] A. Hernandez Ruiz, L. Porzi, S. Rota Bulò, and F. Moreno-Noguer, "3D CNNs on distance matrices for human action recognition," in *Proceedings of the 2017 ACM on Multimedia Conference*, pp. 1087–1095, ACM, 2017.

[70] A. Mucherino and D. S. Gonçalves, "An approach to dynamical distance geometry," in *International Conference on Geometric Science of Information*, pp. 821–829, Springer, 2017.

[71] A. Mucherino, J. Omer, L. Hoyet, P. R. Giordano, and F. Multon, "An application-based characterization of dynamical distance geometry problems," *Optimization Letters*, pp. 1–15, 2018.

[72] "Protein folding—Wikipedia, the free encyclopedia," 2018. [Online; accessed 19-October-2018].

[73] J. C. Gower, "Euclidean distance geometry," *Mathematical Sciences*, vol. 7, no. 1, pp. 1–14, 1982.

[74] J. C. Gower, "Properties of Euclidean and non-Euclidean distance matrices," *Linear Algebra and Its Applications*, vol. 67, pp. 81–97, 1985.

[75] J. Dattorro, *Convex Optimization & Euclidean Distance Geometry*. Meboo, 2011.

[76] P. Tabaghi and I. Dokmanić, "Real polynomial gram matrices without real spectral factors," *arXiv preprint arXiv:1903.04085*, 2019.

[77] L. Blumenthal and B. Gillam, "Distribution of points in n-space," *The American Mathematical Monthly*, vol. 50, no. 3, pp. 181–185, 1943.

[78] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part I—convex underestimating problems," *Mathematical Programming*, vol. 10, no. 1, pp. 147–175, 1976.

[79] E. M. Smith and C. C. Pantelides, "A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex minlps," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 457–478, 1999.

[80] P. H. Schönemann, *A solution of the orthogonal Procrustes problem with applications to orthogonal and oblique rotation*. PhD thesis, University of Illinois at Urbana-Champaign, 1964.

[81] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas," in *The IEEE International Conference on Robotics and Automation*, pp. 842–849, IEEE, 2003.

[82] C.-C. Wang, C. Thorpe, and A. Suppe, "Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds," in *The 2003 IEEE Intelligent Vehicles Symposium*, pp. 416–421, IEEE, 2003.

[83] L. Ephremidze, "An elementary proof of the polynomial matrix spectral factorization theorem," *in Proceedings of the Royal Society of Edinburgh Section A: Mathematics*, vol. 144, no. 4, pp. 747–751, 2014.

[84] L. Ephremidze, I. Spitkovsky, and E. Lagvilava, "Rank-deficient spectral factorization and wavelets completion problem," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 13, no. 03, p. 1550013, 2015.

[85] N. Krislock and H. Wolkowicz, "Explicit sensor network localization using semidefinite representations and facial reductions," *SIAM Journal on Optimization*, vol. 20, no. 5, pp. 2679–2708, 2010.

[86] L. Ephremidze, F. Saied, and I. M. Spitkovsky, "On the algorithmization of Janashia-Lagvilava matrix spectral factorization method," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 728–737, 2018.

[87] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[88] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.

[89] N. Linial, E. London, and Y. Rabinovich, "The geometry of graphs and some of its algorithmic applications," *Combinatorica*, vol. 15, no. 2, pp. 215–245, 1995.

[90] K. Verbeek and S. Suri, "Metric embedding, hyperbolic space, and social networks," in *Proceedings of the Thirtieth Annual Symposium on Computational geometry*, pp. 501–510, 2014.

[91] B. Dhingra, C. J. Shallue, M. Norouzi, A. M. Dai, and G. E. Dahl, "Embedding text in hyperbolic spaces," *arXiv preprint arXiv:1806.04313*, 2018.

[92] C. V. Cannistraci, G. Alanis-Lobato, and T. Ravasi, "From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks," *Scientific Reports*, vol. 3, p. 1613, 2013.

[93] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, *et al.*, "Gene ontology: Tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, p. 25, 2000.

[94] T. D. Q. Vinh, Y. Tay, S. Zhang, G. Cong, and X.-L. Li, "Hyperbolic recommender systems," *arXiv preprint arXiv:1809.01703*, 2018.

[95] B. P. Chamberlain, S. R. Hardwick, D. R. Wardrope, F. Dzogang, F. Daolio, and S. Vargas, "Scalable hyperbolic recommender systems," *arXiv preprint arXiv:1902.08648*, 2019.

[96] J. B. Kruskal and M. Wish, *Multidimensional Scaling*. No. 11, Sage, 1978.

[97] Y. Shavitt and T. Tankel, "Hyperbolic embedding of internet graph for distance estimation and overlay construction," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 25–36, 2008.

[98] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

[99] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[100] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, "Solving Euclidean distance matrix completion problems via semidefinite programming," *Computational Optimization and Applications*, vol. 12, no. 1-3, pp. 13–30, 1999.

[101] D. Asta and C. R. Shalizi, "Geometric network comparison," *arXiv preprint arXiv:1411.1350*, 2014.

[102] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguná, "Hyperbolic geometry of complex networks," *Physical Review E*, vol. 82, no. 3, p. 036106, 2010.

[103] R. Kleinberg, "Geographic routing using hyperbolic space," in *26th IEEE International Conference on Computer Communications*, pp. 1902–1909, IEEE, 2007.

[104] A. Cvetkovski and M. Crovella, "Hyperbolic embedding and routing for dynamic graphs," in *IEEE International Conference on Computer Communications*, pp. 1647–1655, IEEE, 2009.

[105] M. Boguná, F. Papadopoulos, and D. Krioukov, "Sustaining the internet with hyperbolic mapping," *Nature Communications*, vol. 1, p. 62, 2010.

[106] R. C. Wilson, E. R. Hancock, E. Pekalska, and R. P. Duin, "Spherical and hyperbolic embeddings of data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2255–2269, 2014.

[107] I. Vendrov, R. Kiros, S. Fidler, and R. Urtasun, "Order-embeddings of images and language," *arXiv preprint arXiv:1511.06361*, 2015.

[108] L. Van Der Maaten and K. Weinberger, "Stochastic triplet embedding," in *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, IEEE, 2012.

[109] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. T. Kalai, "Adaptively learning the crowd kernel," *arXiv preprint arXiv:1105.1033*, 2011.

[110] J. W. Cannon, W. J. Floyd, R. Kenyon, W. R. Parry, *et al.*, "Hyperbolic geometry," *Flavors of Geometry*, vol. 31, pp. 59–115, 1997.

[111] R. Benedetti and C. Petronio, *Lectures on Hyperbolic Geometry*. Springer Science & Business Media, 2012.

[112] I. Gohberg, P. Lancaster, and L. Rodman, "Matrices and indefinite scalar products," *Acta Applicandae Mathematica*, vol. 6, pp. 101–102, May 1986.

[113] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.

[114] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre, "Low-rank optimization with trace norm penalty," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2124–2149, 2013.

[115] M. Fornasier, H. Rauhut, and R. Ward, "Low-rank matrix recovery via iteratively reweighted least squares minimization," *SIAM Journal on Optimization*, vol. 21, no. 4, pp. 1614–1640, 2011.

[116] M. Fazel, H. Hindi, and S. P. Boyd, "Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 3, pp. 2156–2162, IEEE, 2003.

[117] C. Olsson, A. Eriksson, and R. Hartley, "Outlier removal using duality," in *Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1450–1457, IEEE, 2010.

[118] Y. Seo, H. Lee, and S. W. Lee, "Outlier removal by convex optimization for l-infinity approaches," in *Pacific-Rim Symposium on Image and Video Technology*, pp. 203–214, Springer, 2009.

[119] J. Yu, A. Eriksson, T.-J. Chin, and D. Suter, "An adversarial optimization approach to efficient outlier removal," *Journal of Mathematical Imaging and Vision*, vol. 48, no. 3, pp. 451–466, 2014.

[120] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Advances in Neural Information Processing Systems*, pp. 2496–2504, 2010.

[121] A. Majumdar, G. Hall, and A. A. Ahmadi, "Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, 2019.

[122] A. Yurtsever, J. A. Tropp, O. Fercoq, M. Udell, and V. Cevher, "Scalable semidefinite programming," *arXiv preprint arXiv:1912.02949*, 2019.

[123] A. Yurtsever, M. Udell, J. A. Tropp, and V. Cevher, "Sketchy decisions: Convex low-rank matrix optimization with optimal storage," *arXiv preprint arXiv:1702.06838*, 2017.

[124] P. Jawanpuria, M. Meghwanshi, and B. Mishra, "Low-rank approximations of hyperbolic embeddings," *arXiv preprint arXiv:1903.07307*, 2019.

[125] G. A. Miller, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[126] J. W. Moon *et al.*, "On the maximum degree in a random tree.," *The Michigan Mathematical Journal*, vol. 15, no. 4, pp. 429–432, 1968.

[127] J. L. Gilbert, M. J. Guthart, S. A. Gezan, M. P. de Carvalho, M. L. Schwieterman, T. A. Colquhoun, L. M. Bartoshuk, C. A. Sims, D. G. Clark, and J. W. Olmstead, "Identifying breeding priorities for blueberry flavor using biochemical, sensory, and genotype by environment analyses," *PLoS One*, vol. 10, no. 9, p. e0138494, 2015.

[128] J. R. Hurley and R. B. Cattell, "The Procrustes program: Producing direct rotation to test a hypothesized factor structure," *Behavioral Science*, vol. 7, no. 2, p. 258, 1962.

[129] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[130] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pp. 22–31, 2004.

[131] J. M. Fitzpatrick, J. B. West, and C. R. Maurer, "Predicting error in rigid-body point-based registration," *IEEE Transactions on Medical Imaging*, vol. 17, no. 5, pp. 694–702, 1998.

[132] F. Pomerleau, F. Colas, and R. Siegwart, "A review of point cloud registration algorithms for mobile robotics," *Foundations and Trends in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.

[133] P. Shvaiko and J. Euzenat, "Ontology matching: state of the art and future challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 158–176, 2011.

[134] J. Euzenat, P. Shvaiko, *et al.*, *Ontology Matching*, vol. 18. Springer, 2007.

[135] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, IEEE, 2001.

[136] D. Alvarez-Melis, Y. Mroueh, and T. Jaakkola, "Unsupervised hierarchy matching with optimal transport over hyperbolic spaces," in *International Conference on Artificial Intelligence and Statistics*, pp. 1606–1617, PMLR, 2020.

[137] K. V. Mardia and P. E. Jupp, *Directional Statistics*, vol. 494. John Wiley & Sons, 2009.

[138] J. G. Ratcliffe, S. Axler, and K. Ribet, *Foundations of Hyperbolic Manifolds*, vol. 149. Springer, 2006.

[139] A. A. Ungar, "A gyrovector space approach to hyperbolic geometry," *Synthesis Lectures on Mathematics and Statistics*, vol. 1, no. 1, pp. 1–194, 2008.

[140] A. B. Novikoff, "On convergence proofs for perceptrons," tech. rep., Stanford Research Institute, 1963.

[141] S. Dasgupta, A. T. Kalai, and A. Tauman, "Analysis of perceptron-based active learning," *Journal of Machine Learning Research*, vol. 10, no. 2, 2009.

[142] H. Cho, B. DeMeo, J. Peng, and B. Berger, "Large-margin classification in hyperbolic space," in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1832–1840, PMLR, 2019.

[143] M. Weber, M. Zaheer, A. S. Rawat, A. Menon, and S. Kumar, "Robust large-margin learning in hyperbolic space," *arXiv preprint arXiv:2004.05465*, 2020.

[144] O. Skopek, O.-E. Ganea, and G. Bcigneul, "Mixed-curvature variational autoencoders," in *International Conference on Learning Representations*, 2020.

[145] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," vol. 86, pp. 2278–2324, Ieee, 1998.

[146] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[147] A. Krizhevsky, G. Hinton, *et al.*, *Learning Multiple Layers of Features From Tiny Images*. Citeseer, 2009.

[148] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, *et al.*, "Massively parallel digital transcriptional profiling of single cells," *Nature Communications*, vol. 8, no. 1, pp. 1–12, 2017.

[149] "Hodgkin's Lymphoma, Dissociated Tumor: Targeted-Compare, Immunology Panel by Cell Ranger 4.0.0," *10x Genomics*, July 7th, 2020.

[150] "PBMCs from a Healthy Donor: Targeted, Immunology Panel by Cell Ranger 4.0.0," *10x Genomics*, July 7th, 2020.

[151] J. G. Ratcliffe, S. Axler, and K. Ribet, *Foundations of Hyperbolic Manifolds*, vol. 149. Springer, 1994.

[152] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature*, vol. 176. Springer Science & Business Media, 2006.

[153] J. H. Gallier and J. Quaintance, *Differential Geometry and Lie Groups: A Computational Perspective*, vol. 12. Springer Nature, 2020.

[154] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.

[155] L. W. Tu, *An Introduction to Manifolds*. Springer, New York, 2011.

[156] A. Subramanian, R. Narayan, S. M. Corsello, D. D. Peck, T. E. Natoli, X. Lu, J. Gould, J. F. Davis, A. A. Tubelli, J. K. Asiedu, *et al.*, "A next generation connectivity map: L1000 platform and the first 1,000,000 profiles," *Cell*, vol. 171, no. 6, pp. 1437–1452, 2017.

[157] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[158] P. Li and O. Milenkovic, "Inhomogoenous hypergraph clustering with applications," in *Advances in Neural Information Processing Systems*, pp. 2305–2315, 2017.

[159] R. C. Wilson, E. R. Hancock, E. Pkekalska, and R. P. Duin, "Spherical embeddings for non-Euclidean dissimilarities," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1903–1910, IEEE, 2010.

[160] R. M. Green and R. M. Green, *Spherical Astronomy*. Cambridge University Press, 1985.

[161] F. Wauthier, M. Jordan, and N. Jojic, "Efficient ranking from pairwise comparisons," in *Proceedings of the International Conference on Machine Learning*, pp. 109–117, PMLR, 2013.

[162] K. G. Jamieson and R. Nowak, "Active ranking using pairwise comparisons," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2240–2248, 2011.

[163] K. G. Jamieson and R. D. Nowak, "Low-dimensional embedding using adaptively selected ordinal data," in *2011 49th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1077–1084, IEEE, 2011.

[164] S. Haghiri, D. Ghoshdastidar, and U. von Luxburg, "Comparison-based nearest neighbor search," in *Artificial Intelligence and Statistics*, pp. 851–859, PMLR, 2017.

[165] S. Haghiri, D. Garreau, and U. Luxburg, "Comparison-based random forests," in *Proceedings of the International Conference on Machine Learning*, pp. 1871–1880, PMLR, 2018.

[166] Z. Cui, N. Charoenphakdee, I. Sato, and M. Sugiyama, "Classification from triplet comparison data," *Neural Computation*, vol. 32, no. 3, pp. 659–681, 2020.

[167] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature*. Springer New York, 2007.

[168] S. S. Skiena, W. D. Smith, and P. Lemke, "Reconstructing sets from interpoint distances," in *Proceedings of the sixth annual symposium on Computational geometry*, pp. 332–339, 1990.

[169] N. C. Jones, P. A. Pevzner, and P. Pevzner, *An Introduction to Bioinformatics Algorithms*. MIT Press, 2004.

[170] S. Huang and I. Dokmanić, "Reconstructing point sets from distance distributions," *arXiv preprint arXiv:1804.02465*, 2018.

[171] M. Kleindessner and U. Luxburg, "Uniqueness of ordinal embedding," in *Proceedings the Conference on Learning Theory*, pp. 40–67, PMLR, 2014.

[172] Q. Cao, Y. Ying, and P. Li, "Similarity metric learning for face recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2408–2415, 2013.

[173] B. McFee and G. Lanckriet, "Learning multi-modal similarity," *Journal of Machine Learning Research*, vol. 12, no. Feb, pp. 491–523, 2011.

[174] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 2009.

[175] G. Van Brummelen and E. A. Hamm, "Heavenly mathematics: The forgotten art of spherical trigonometry," *Aestimatio: Critical Reviews in the History of Science*, vol. 11, pp. 127–130, 2014.

[176] L. Mirsky, *Transversal Theory: An Account of Some Aspects of Combinatorial Mathematics*. Academic Press, 1971.

[177] R. A. Rankin, "The closest packing of spherical caps in n dimensions," *Glasgow Mathematical Journal*, vol. 2, no. 3, pp. 139–144, 1955.

[178] A. D. Wyner, "Random packings and coverings of the unit n-sphere," *The Bell System Technical Journal*, vol. 46, no. 9, pp. 2111–2118, 1967.

[179] P. Turán, "On an external problem in graph theory," *Középiskolai Matematikai és Fizikai Lapok*, vol. 48, pp. 436–452, 1941.

[180] C. E. Meacham and S. J. Morrison, "Tumour heterogeneity and cancer cell plasticity," *Nature*, vol. 501, no. 7467, pp. 328–337, 2013.

[181] D. van Dijk, J. Nainys, R. Sharma, P. Kaithail, A. J. Carr, K. R. Moon, L. Mazutis, G. Wolf, S. Krishnaswamy, and D. Pe'er, "MAGIC: A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data," *BioRxiv*, p. 111591, 2017.

[182] W. V. Li and J. J. Li, "An accurate and robust imputation method scimpute for single-cell RNA-seq data," *Nature Communications*, vol. 9, no. 1, pp. 1–9, 2018.

[183] S. C. Hicks, F. W. Townes, M. Teng, and R. A. Irizarry, "Missing data and technical variability in single-cell RNA-sequencing experiments," *Biostatistics*, vol. 19, no. 4, pp. 562–578, 2018.

[184] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, "Single-cell RNA-seq denoising using a deep count autoencoder," *Nature Communications*, vol. 10, no. 1, pp. 1–14, 2019.

[185] M. Plass, J. Solana, F. A. Wolf, S. Ayoub, A. Misios, P. Glažar, B. Obermayer, F. J. Theis, C. Kocks, and N. Rajewsky, "Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics," *Science*, vol. 360, no. 6391, 2018.

[186] P. Chebotarev and E. Shamis, "The matrix-forest theorem and measuring relations in small social groups," *arXiv preprint math/0602070*, 2006.

[187] R. F. Bass and K. Gröchenig, "Random sampling of multivariate trigonometric polynomials," *SIAM Journal on Mathematical Analysis*, vol. 36, pp. 773–795, Jan. 2005.

[188] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 2012.

[189] L. Mirsky, "A trace inequality of John von Neumann," *Monatshefte für mathematik*, vol. 79, no. 4, pp. 303–306, 1975.

[190] R. M. Dudley, "Central limit theorems for empirical measures," *The Annals of Probability*, pp. 899–929, 1978.

[191] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[192] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *Journal of Machine Learning Research*, vol. 2, no. Nov, pp. 67–93, 2001.