博士論文 (要約)

Machine Learning over Space Forms

(空間形上の機械学習)

鈴木　惇

Atsushi Suzuki

Abstract

Machine learning can be interpreted as dissimilarity learning using a program. The class of dissimilarity measures given by space forms is considered an appropriate class because it describes an adequate range of dissimilarity measures and provides simple optimization and implementation. Limited research has been conducted on the applications of space forms in terms of fundamental elements of machine learning that include modelling, optimization, and evaluation. Specifically, (i) the models for learning dissimilarity measures in the data domain using ordinal data or multi-relational graph data have not been studied; (ii) the two major optimization methods (i.e., the natural gradient and the Riemannian gradient methods) have not been compared; (iii) evaluation methods for machine learning over space forms have not been discussed. This study is aimed at solving these issues by (a) proposing methods for dissimilarity learning over (non-Euclidean) space forms applicable for ordinal data or multi-relational graph data, utilizing the distance function and the exponential map, (b) drawing theoretical comparisons among first-order stochastic optimization methods using the traditional Euclidean metric, the natural gradient method, and the Riemannian gradient method in the machine-learning setting, and (c) proposing a model evaluation method based on the minimax regret principle that is a primitive evaluation principle that does not require any statistical data-distribution assumptions. These three solutions provide the foundations for machine learning over space forms.

Contents

Chapter 1

Introduction

This chapter formulates the issues that we have considered in this thesis. First, the interpretation of machine learning as dissimilarity learning is discussed, which is followed by an introduction to space forms and their advantages in machine-learning applications. Finally, the issues in machine learning over space forms are discussed, which is followed by an introduction to the fundamental elements of machine learning; based on the discussion, we list the research questions and propose their solutions.

1.1 Machine Learning and Dissimilarity

Learning can be defined as a procedure implemented using a program to improve the measures of performance P of tasks T through experience E. (Mitchell, 1997). In reality, a T can be a prediction, clustering, generation, or description. Correspondingly, P can be the accuracy of a prediction, effectiveness of clustering, reality of generation, or conciseness of a description. E can be data in the past. This definition of learning does not require an explicit description of what is being learnt through machine learning. However, to discuss the validity or improvement of machine learning tasks, no explicit description of what is learnt by machine learning is inconvenient. Although there can be different ways of describing how a model learns through machine learning, the interpretation of machine learning as dissimilarity learning over a data space is preferred. Based on this interpretation, we reformulate machine-learning tasks in terms of dissimilarity learning. Examples of such a reformulation are presented below.

Example 1.1.1 (Machine learning tasks as dissimilarity learning)**.**

Classification and Regression Classification and regression are tasks aimed at finding an appropriate map f from feature domain $\mathcal{X}$ to domain $\mathcal{Y}$. Domain $\mathcal{Y}$ is a discrete set in the case of the classification task, called label domain, and a continuous space in the case of the regression task, called the objective domain. Here, f is called a classifier or regressor, and $f(x)$ indicates the class label or estimator of feature x. Classification and regression are considered tasks aimed at finding an appropriate set (c, d), where $c : \mathcal{Y} \to \mathcal{X}$ indicates the "center" of the class or value $y \in \mathcal{Y}$ by $c(y)$ for each class or value y, and $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a dissimilarity measure that indicates the dissimilarity between $x_1 \in \mathcal{X}$ and $x_2 \in \mathcal{X}$ by $d(x_1, x_2)$. Here, the classifier or regressor map f is given by

$$f(x) \coloneqq \operatorname*{argmin}_{y \in \mathcal{Y}} \{d(c(y), x)\}. \tag{1.1}$$

Clustering Clustering is a task aimed at finding an appropriate map $i : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$ over data domain $\mathcal{X}$ that is called a cluster identifier, where $i(x_1, x_2) = 0$ indicates that data x_1 and x_2 belong to the same cluster, and $i(x_1, x_2) = 1$ indicates that data

x_1 and x_2 belong to different clusters. Clustering is a task aimed at finding an appropriate set (K, c, d), where $K \in \mathbb{Z}_{\geq 0}$ is the number of clusters, $c : \{1, 2, \cdots, K\} \to \mathcal{X}$ indicates the center of cluster k by $c(k)$, and $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a dissimilarity measure that indicates the dissimilarity between $x_1 \in \mathcal{X}$ and $x_2 \in \mathcal{X}$ by $d(x_1, x_2)$. Here, cluster identifier i is given by

$$i(x_0, x_1) \coloneqq \begin{cases} 1 & \text{if } c(x_0) = c(x_1) \\ 0 & \text{if } c(x_0) \neq c(x_1) \end{cases} \tag{1.2}$$

Generation Generation is a task aimed at finding an appropriate distribution in data domain $\mathcal{X}$. $p : \mathcal{X} \to \mathbb{R}$ denotes the probability density function p on some base measure $\lambda_{\mathcal{X}}$ of the appropriate distribution. Generation is considered a task aimed at finding an appropriate set (c_0, d), where $c_0 \in \mathcal{X}$ indicates the center of the data domain, and $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a dissimilarity measure that indicates the dissimilarity between $x_1 \in \mathcal{X}$ and $x_2 \in \mathcal{X}$ by $d(x_1, x_2)$. Here, the probability density function $p : \mathcal{X} \to \mathbb{R}$ is given by

$$p(x) \coloneqq \frac{\exp(-d(c_0, x))}{\int_{\mathcal{X}} \exp(-d(c_0, x)) \mathrm{d}\lambda_{\mathcal{X}}(x)}. \tag{1.3}$$

Description Description is a task aimed at finding an appropriate coding scheme on data domain $\mathcal{X}$. According to the Kraft-McMillan inequality (Kraft, 1949; McMillan, 1956) for code length function, a necessary and sufficient condition for the existence of prefix code with the code length, and the existence of coding schemes such as the Huffman code (Huffman, 1952) that yields a sufficient condition for coding scheme determining code length function is essential to the coding scheme because if we have a code length function that satisfies the Kraft-McMillan inequality given by

$$\int_{\mathcal{X}} \exp(-l(x)) \mathrm{d}\lambda(x) \leq 1, \tag{1.4}$$

we can immediately obtain a coding scheme such as the Huffman code based on the code length function. Here, $l : \mathcal{X} \to \mathbb{R}$ denotes the code length function p on some base measure $\lambda_{\mathcal{X}}$ of the appropriate distribution. Generation is considered a task aimed at finding an appropriate set (c_0, d), where $c_0 \in \mathcal{X}$ indicates the center of the data domain, and $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a dissimilarity measure that indicates the dissimilarity between $x_1 \in \mathcal{X}$ and $x_2 \in \mathcal{X}$ by $d(x_1, x_2)$. Here the code length function $l : \mathcal{X} \to \mathbb{R}$ is given by

$$l(x) = \ln \frac{1}{p(x)}, \tag{1.5}$$

where

$$p(x) \coloneqq \frac{\exp(-d(c_0, x))}{\int_{\mathcal{X}} \exp(-d(c_0, x)) \mathrm{d}\lambda_{\mathcal{X}}(x)}. \tag{1.6}$$

It can be inferred from the aforementioned examples that learning is aimed at finding a dissimilarity measure, and machine learning can be interpreted as a procedure implemented using a program to find a dissimilarity measure over the data domain in some class of tasks T with respect to some class of performance measures P through experience E. The following section discusses methods of finding an appropriate dissimilarity measure.

1.2 Space forms in Machine Learning

1.2.1 Requirements for dissimilarity measure

As discussed in the previous section, machine learning can be considered a procedure implemented using a program to find an appropriate dissimilarity measure. Because this procedure is automatic, the fundamental problem encountered when using machine learning is the determination of an appropriate class of dissimilarity measures. This problem can be directly reduced to the problem of defining the appropriateness of a dissimilarity measure class. In machine learning, a dissimilarity measure class must contain a sufficient number of dissimilarity measures such that there exists a dissimilarity measure in the class that performs well for the objective task. Although a large dissimilarity measure class can be considered, it may include an unnecessary hypothesis and cause over fitting. In addition, if we use a dissimilarity measure class wherein we cannot consider the differential of a function, optimization shall be very difficult. Therefore, it is necessary to find a good dissimilarity measure class that is (i) machine-friendly in terms of model implementation, optimization, and evaluation, and (ii) able to express a wide range of dissimilarity measures for modelling; these measures must appear in reality but the range must not be so wide that it leads to over fitting. The most implementation-friendly method involves mapping the points in the data domain to points in the Euclidean space and using the distance function as a dissimilarity measure. Using the distance function as a dissimilarity measure offers simple implementation and an intuitive interpretation. This is because it satisfies the axioms of a metric: non-negativity, identity of indiscernibles, symmetry, and triangle inequality. However, the dissimilarity measure class of the Euclidean space is limited, which may cause a problem in reality. For example, even if the dissimilarity measure given by the graph distance of a tree is required for good performance, it is not included in the dissimilarity measure class of Euclidean space because the surface area of the Euclidean space grows polynomially with respect to its radius, whereas the number of nodes grows exponentially with respect to the distance from the root of the tree. Hence, the dissimilarity measure classes based on a metric space, together with those given by the Euclidean space, are required to retain the intuitive nature.

1.2.2 Riemannian manifolds and space forms

Hence, this study focuses on Riemannian manifolds that have appropriate properties that satisfy the above requirements. A Riemannian manifold is a space locally isomorphic to a metric vector space. Riemannian manifolds satisfy the above requirements for machine learning owing to the following reasons:

- They can describe a variety of dissimilarity measures depending on local metrics. Particularly through local metrics, the distance between two points and the angle between two lines can be described. These notions of distance and angle can be used to describe a dissimilarity measure. Despite its capability of describing a variety of dissimilarity measures, it is simple to limit its class to avoid the over fitting problem. The class of constant curvature manifolds is a possible solution which is discussed later.
- They can be easily implemented as they are locally isomorphic to a vector space that is naturally auto-implemented. In addition, many Riemannian manifolds can be described as a subspace of a vector space, and upon differentiation, an objective function defined on a Riemannian manifold can be efficiently optimized.

In addition, if we obtain a dissimilarity measure as a map to Riemannian manifolds, the Riemannian manifolds must have some homogeneity in their metric structure such that all the relations among points are reduced to distances between two points. The metric structure of a Riemannian manifold is determined by the sectional curvature that indicates the growing speed of space in an infinitesimal plane on a point. Hence, to ensure homogeneity, the sectional curvature must be constant in every infinitesimal plane at every point. Thus, we focus on machine learning over space forms, which are Riemannian manifolds with a constant sectional curvature. Determining the kinds of study required for realizing machine learning over space forms is the next concern. To clarify this problem, we divided the machine-learning procedure into fundamental elements and discussed the lack of studies on the use of space forms.

1.3 Fundamental Elements of Machine Learning

The machine-learning procedure can be divided into three elements that are described below on the basis of interpretation of machine learning as dissimilarity learning.

1.3.1 Modelling

Interpreting machine learning as dissimilarity learning, modelling refers to establishing subjective reliabilities among dissimilarity measures using a criterion to quantify how dissimilarity measures reflect past experiences. However, to avoid the over fitting problem, putting no reliability on some dissimilarity measure set is inevitable; thus, the model must be limited to some dissimilarity measure set. Moreover, depending on situations, we put different levels of reliability among the limited dissimilarity measures even though we put the same amount of reliability among the set. The dissimilarity measure set with a specific reliability is often formulated as a parameter domain, and the criterion to quantify how a dissimilarity measure reflects a past experience is often formulated as a loss function. The reliability among dissimilarity measures is often formulated as a prior distribution or a regularization term.

1.3.2 Optimization

Optimization is a procedure to select the best dissimilarity measure in the model. If a loss function is given as a formulation of the model, this procedure involves the optimization of the loss function and the reliability term such as a regularization term. Depending on the difficulty of optimization, the original objective function can be replaced with an alternative relaxed objective function.

1.3.3 Evaluation

At the end of the machine-learning procedure, the performance must be evaluated. However, it is not possible to evaluate the performance directly from past experiences. Many principles have been proposed to solve this problem. In most cases, the original loss function is not directly used as a function that yields the model performance evaluation based on the principles. We can categorize these principles into two types according to (Miyaguchi, 2018). The first category of principles is based on some statistical assumptions such as independent identical distribution property, and the second category of principles does not make any assumptions. Examples of the first category include validation, the Akaike information criterion (AIC) (Akaike, 1974), the Bayesian information criterion (BIC) (Schwarz, 1978), probably approximately correct (PAC) risk bounds

(Valiant, 1984), minimax risk principle (Wald, 1949; Lehmann and Casella, 2006), and Bayes principle (Gelman et al., 2013). Examples of the second category include the prequential principle (Dawid, 1984), minimax regret principle (Savage, 1951), and minimum description length principle (Rissanen, 1978) that can be considered an integration of the two (Miyaguchi, 2018). The first category tends to be suitable for evaluation of some tasks such as prediction in an independent identical distribution setting. Principles belonging to the second category can be considered more primitive and provide applications as they can be applied even when there are no details of the data source (Miyaguchi, 2018). Hence, this study focuses on the latter category, i.e., the no assumption category. Specifically, the normalized maximum likelihood code length is discussed as it can be justified by both the prequential principle and minimax regret principle.

1.4 Research Question

In the following section, these three elements for machine learning over space forms are analyzed.

1.4.1 Model over space forms

A simple and strong method for a Riemannian space to yield a dissimilarity measure involves identifying the data domain with the Riemannian space. Specifically, embedding objects into a Riemannian space, identifying a discrete space with a Riemannian space, has many applications and has been studied extensively (Sarkar, 2011; Nickel and Kiela, 2017; Ganea et al., 2018a). For object embeddings, some additional relational data are required because objects in a mere discrete space do not contain any information for embedding. The simplest example is a graph, which contains objects connected by edges. In a graph, two objects are connected by an edge if and only if they are similar to each other and are identified as two points close to each other. However, limiting the class of relational data to edge sets may be too conservative in reality, and we have to consider other types of relational data. Relational data can be categorized into the following two types: "weaker" and "stronger" than edge sets. Weaker data only contains partial information of the edge set, whereas stronger data contains more information than that of a single edge set. A typical example of the former are human ratings, wherein the absolute value of a rating is not reliable, but relative ranking among one rater's ratings. Here, we have ordinal information of dissimilarity measures instead of an edge set. A typical example of the latter is a knowledge base, wherein many types of relations exist among different entities. Here, multiple edge sets must be considered in the embedding of objects. The former data set is considered an object set with ordinal information among dissimilarity, called ordinal data. The latter data set is considered an object set with multiple edge sets, called a multi-relational graph. Although existing methods using a non-Euclidean Riemannian manifold have focused on a graph, no study has focused on other kinds of relational data such as ordinal data or a multi-relational graph. The above discussion gives rise to the following research questions.

Q1 How can we establish a machine-learning model over Riemannian manifolds that is applicable for ordinal data or multi-relational graph embedding?

1.4.2 Optimization over space forms

Optimization methods for a function on Riemannian spaces have been extensively studied in this decade. In the machine-learning setting, the stochastic first-order methods are the most important because of full-batch optimization. For example, the Newton methods or conjugate gradient methods are intractable in many machine-learning cases owing to the large number of data and parameters. To meet this demand for optimization over space forms, Zhang and Sra (Zhang and Sra, 2016) have proved the significant inequality that derives a relation between the gradient and curvature; this has led many researches to represent the theoretical behavior of many first-order methods on Riemannian spaces (Becigneul and Ganea, 2019). The Riemannian gradient methods update the parameters using the exponential map with some gradient vector. On the contrary, in the context of the information geometric, the natural gradient method (Amari, 1998) has been proposed as a method based on the metric structure of a manifold. Unlike the Riemannian gradient method, the update in the natural gradient method adds the coefficients of a Riemannian gradient vector to the coordinate of the current parameter without using the exponential map. Although the original study (Amari, 1998) has proposed the natural gradient method for optimization over statistic manifolds, the natural gradient method and its stochastic variants have been used in general manifolds for machine learning (Nickel and Kiela, 2017)and Riemannian gradient methods owing to the computational simplicity. However, the performance of the natural gradient method over general manifolds has not been studied, and the natural gradient methods and Riemannian gradient methods have not been compared. The above discussion gives rise to the following research questions:

Q2 Is the Riemannian gradient method better than the natural gradient method in the setting of machine learning over space forms? Under what situation does a difference arise and what is the magnitude of this difference?

1.4.3 Evaluation over space forms

Evaluation methods for space forms have not been developed well. Some evaluation principles based on the asymptotic theory, such as validation, AIC, or BIC, can be directly applied to space forms because their computation is not model-specific owing to the asymptotic theory. However, the principles that provide non-asymptotic theoretical guarantee require model-specific calculations, and their applications in machine learning over space forms have not been studied. Specifically, the normalized maximum likelihood code length, which is a loss function based on minimum description length principle, has not been studied for machine learning over space forms.

Q3 How can we non-asymptotically calculate the normalized maximum likelihood code length for machine learning over space forms?

The following section describes the solutions for these research questions.

1.5 Our contributions

Our solutions to these research questions are listed below:

A1 We proposed methods for dissimilarity learning over (non-Euclidean) space forms that are applicable for ordinal data or multi-relational graphs. The framework of the general Riemannian manifold was constructed before proposing a model of a

specific manifold. By exploiting the metric structure of space forms, our model can extract dissimilarity structures from the data that cannot be extracted by a model over Euclidean space.

A2 We provided the theoretical comparisons among first-order stochastic optimization methods based on the traditional Euclidean metric, natural gradient method, and Riemannian gradient method, in the machine-learning setting. In particular, the results indicate that the Euclidean-metric-based and natural-gradient-method-based methods can fail even in the case of a simple problem for which the Riemannian-gradient-based method guarantees convergence.

A3 We proposed a model-evaluation method based on the minimum description principle. First, we proposed a general calculation method for normalized maximum likelihood code length, which is a basic evaluation method based on the minimum description principle.

Each of the above statements systematically contributes to the fundamental elements of machine learning over space forms. Combining them together, this thesis presents theoretical foundations and specific methods for realizing machine learning over space forms. The rest of the thesis is organized as follows. In Chapter 2, we present basic notions of Riemannian manifolds and space forms as a preliminary. In Chapter 3 and Chapter 4, we explain our contribution to models for machine learning over space forms. In Chapter 3, we propose a dissimilarity learning method for ordinal data. In Chapter 4, we propose a dissimilarity learning method for multi-relational graphs. In Chapter 5, we discuss our contributions in the optimization area for machine learning over space forms, theoretically comparing first-order stochastic optimization methods based on the traditional Euclidean metric, natural gradient method, and Riemannian gradient method. In Chapter 6, we discuss our contributions in the model evaluation area for machine learning over space forms, providing a general calculation method for normalized maximum likelihood code length. These results are closely related to each other. The results of Chapter 3 and 4 share a similar motivation—mapping data spaces to space forms—but they deal with different kinds of data sets, whereas both of them often appear in reality. The theoretical analysis in Chapter 5 motivated us to adopt the Riemannian-gradient-method-based optimization in the setting of Chapter 3 and 4. The methodology in Chapter 6 aims at evaluating the probabilistic model for data points on space forms such as the ones given by the models in Chapter 3 and 4. A visual summary of the overview is depicted in Figure 1.1.

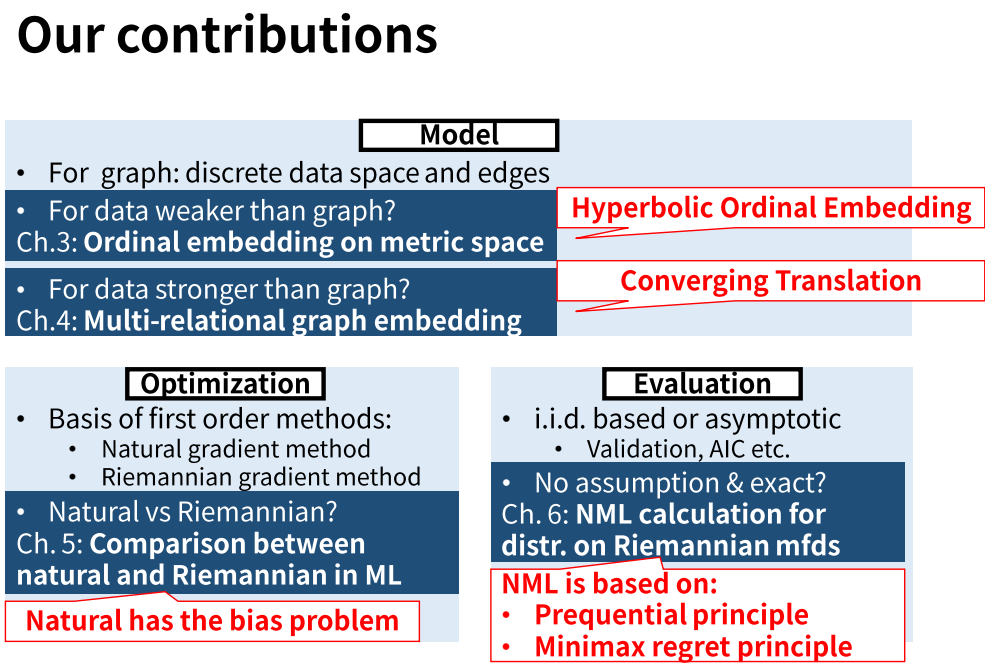

Fig. 1.1. An overview of the thesis. This thesis contributes to the fundamental elements of machine learning over space forms: the modelling, optimization, and evaluation

Chapter 2

Preliminaries: Differential Geometry for Machine Learning

In this chapter, we introduce the Riemannian manifold along with its space form and discuss about their properties, which motivate us to focus on the use of the space forms in machine learning applications.

2.1 C^∞ Manifolds

In this section, we introduce the topological manifold. Topological manifolds are locally Euclidean; therefore, we can implement them in computer algorithms. Because of this property, we are motivated us to use a topological manifold in machine learning. In this thesis, we only consider the C^∞ case. For further details and proofs, see e.g., Loring (2011); John (2018).

Definition 2.1.1. A topological space $\mathcal{M}$ is *locally Euclidean of dimension* D if every point p in the $\mathcal{M}$ has a neighborhood U such that there is a homeomorphism[*1] ϕ from U onto an open subset of $\mathbb{R}^D$. We call the pair $(U, \phi : U \to \mathbb{R}^D)$ a chart, U a coordinate neighborhood, or a coordinate open set, and ϕ a coordinate map or a coordinate system on U. We say that a chart (U, ϕ) is *centered* at $p \in \mathcal{M}$ if $\phi(p) = \mathbf{0}$.

Definition 2.1.2. A *topological manifold* is a Hausdorff[*2],second countable[*3],and locally Euclidean space. It is said to be of dimension D if it is locally Euclidean.

Note that the dimension of a manifold $\mathcal{M}$ is well defined because if $D \neq D'$, $\mathbb{R}^D$, and $\mathbb{R}^{D'}$ are not homeomorphic to each other and if points $p, p' \in \mathcal{M}$ have neighborhoods U, U' that are homeomorphic to open subsets $\mathbb{R}^D, \mathbb{R}^{D'}$, it holds that $D = D'$.

Definition 2.1.3. Two charts: $(U, \phi : U \to \mathbb{R}^D)$and $(U', \phi' : U' \to \mathbb{R}^D)$ of a topological manifold are C^∞-*compatible* if the two maps

$$\begin{aligned} \phi' \circ \left[(\phi)^{-1} \Big|_{\phi(U \cap U')} \right] &: \phi(U \cap U') \to \phi'(U \cap U'), \\ \phi \circ \left[(\phi')^{-1} \Big|_{\phi'(U \cap U')} \right] &: \phi'(U \cap U') \to \phi(U \cap U'), \end{aligned} \tag{2.1}$$

[*1] For topological spaces $\mathcal{M}$ and $\mathcal{M}'$, a map $\phi : \mathcal{M} \to \mathcal{M}'$ is called a homeomorphism if ϕ is a bijection and continuous, and its inverse function ϕ^{-1} is also continuous.

[*2] A topological space $\mathcal{M}$ is called a Hausdorff space if for all point pairs $p, p' \in \mathcal{M}$, there exist neighborhoods U and U' of p and p', respectively, which are disjoint to each other.

[*3] A topological space $\mathcal{M}$ is called second countable if it has a countable open basis.

are C^∞.

Definition 2.1.4. A C^∞ *atlas* on a manifold $\mathcal{M}$ is a collection $\mathfrak{U} = \{(U_i)\}_{i\in I}$ of pairwise C^∞-compatible charts that cover $\mathcal{M}$, i.e., , such that $\mathcal{M} = \bigcup_{i\in I} U_i$.

Definition 2.1.5. A topological space $\mathcal{M}$ is a C^∞ *manifold* if

1. $\mathcal{M}$ is Haussdorff and second countable,
2. $\mathcal{M}$ has a C^∞ atlas.

We can say that a C^∞ manifold is a space such that we can implement a neighborhood of any point in the space in a computer, including the operations defined by the differentiation.

Definition 2.1.6. Let $\mathcal{M}$ and $\mathcal{M}'$ be manifolds of dimension D and D', respectively. A continuous map $F : \mathcal{M} \to \mathcal{M}'$ is C^∞ at a point p in $\mathcal{M}$ if there are charts (U, ϕ) about p and (U', ϕ') about $F(p)$ such that the composition $\phi \circ F \circ \left[(\phi')^{-1}\Big|_{\phi'(U\cap U')} \right]$, the map from the open subset $\phi\big(F^{-1}(U) \cap U'\big)$ of $\mathbb{R}^D$ to $\mathbb{R}^{D'}$, is C^∞ at $\phi(p)$. The continuous map $F : \mathcal{M} \to \mathcal{M}'$ is said to be C^∞ if it is C^∞ at every point at $\mathcal{M}$.

A map $F : \mathcal{M} \to \mathbb{R}$ is called a function on the manifold $\mathcal{M}$ and called a C^∞ function on $\mathcal{M}$ if it is C^∞ as a map. We denote the set of all C^∞ function on $\mathcal{M}$ by $C^\infty(\mathcal{M})$.

2.2 Curves and tangent spaces

In machine learning applications, which can be regarded as a dissimilarity measure learning, the relation among points in the manifold is important. In this section, we introduce curves in a manifold; curves are an important concept used to describe the relation among points. In addition, we introduce tangent vectors, which can be regarded as an infinitesimal part of a curve. The tangent vector is an important tool that can be used to reduce the calculation of curve properties, such as the length of and the angles between curves, to notions in the metric vector space.

Definition 2.2.1. A C^∞ *curve* in a manifold $\mathcal{M}$ is a C^∞ map $c :]a, b[\to \mathcal{M}$, where $a, b \in \mathbb{R}$ satisfy $a < b$, and $]a, b[$ is an open interval from a to b. The parameter of a curve in a manifold is called *time*. We say that c is a *curve starting at* p if $0 \in]a, b[$ and $c(0) = p$.

Definition 2.2.2. The velocity vector $\frac{\mathrm{d}}{\mathrm{d}t}\big|_{t_0} c$ of a curve $c :]a, b[\to \mathcal{M}$ at time $t_0 \in]a, b[$ is a function of $C^\infty(\mathcal{M})$ given by the following:

$$\left(\frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{t_0} \right) c(f) \coloneqq \frac{\mathrm{d}}{\mathrm{d}t}(f \circ c)(t_0), \tag{2.2}$$

for $f \in C^\infty(\mathcal{M})$.

Definition 2.2.3. The *tangent space* $T_p\mathcal{M}$ at p on $\mathcal{M}$ is the set of velocity vectors at time 0 of all the C^∞ curves starting at p, i.e., , the ones that satisfy $c(0) = p$. An element in the tangent space $T_p\mathcal{M}$ is called a *tangent vector* at p.

As implied by the definition, a tangent vector can be regarded as an infinitesimal curve. The following proposition claims that tangent vectors are elements of a linear space.

Proposition 2.2.4. *Let p be a point on $\mathcal{M}$. The tangent space $T_p\mathcal{M}$ is closed under the*

scalar multiplication and addition defined as follows:

$$\begin{aligned}\left(a \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_0 c\right)(f) &\coloneqq a \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_0 c(f),\\ \left(\left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_0 c + \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_0 c'\right)(f) &\coloneqq \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_0 c(f) + \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_0 c'(f),\end{aligned} \tag{2.3}$$

for $f \in C^\infty(\mathcal{M})$ and $a \in \mathbb{R}$. Thus, the tangent space $T_p\mathcal{M}$ is a linear space equipped with the above scalar multiplication and addition. The tangent space $T_p\mathcal{M}$ is the D-dimensional linear space. Let (U, ϕ) be a chart centered at p, i.e., , $\phi(p) = \mathbf{0}$, and $c^{(d)}$ denote the curve given by

$$c^{(d)}(t) \coloneqq \phi^{-1}(t\mathbf{e}_d), \tag{2.4}$$

where $\mathbf{e}_d$ is the D-dimensional one-hot vector such that d-th element is one and the other elements are zero. Then the set

$$\left\{ \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_0 c^{(1)}, \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_0 c^{(2)}, \ldots, \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_0 c^{(D)} \right\} \tag{2.5}$$

of tangent vectors is a basis set of $T_p\mathcal{M}$.

Let $\left(x^1, x^2, \ldots, x^D\right)$ be a coordinate system of $\mathbb{R}^D$. The velocity vector of the curve $c^{(d)}$ is given by the partial derivative operator $\frac{\partial}{\partial x^d}$ as follows:

$$\left(\left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_0 c^{(d)}\right)(f) = \frac{\partial}{\partial x^d}\left(f \circ \phi^{-1}\right)(\phi(p)). \tag{2.6}$$

Therefore, in the following, we denote the tangent vector $\left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_0 c^{(d)}$ by $\frac{\partial}{\partial x^d}_p$. Thus, a basis of $T_p\mathcal{M}$ is given by the set

$$\left\{ \frac{\partial}{\partial x^1}_p, \frac{\partial}{\partial x^2}_p, \ldots, \frac{\partial}{\partial x^d}_p \right\}. \tag{2.7}$$

Whereas a tangent vector is a function on $C^\infty(\mathcal{M})$, an element in $C^\infty(\mathcal{M})$, which is a function on $\mathcal{M}$, is also regarded as a function on $T_p\mathcal{M}$. The function on $T_p\mathcal{M}$ induced by $f \in C^\infty(\mathcal{M})$ is called the differential of f.

Definition 2.2.5. The *differential* of f is the function $\mathrm{d}f_p : T_p\mathcal{M} \to \mathbb{R}$ given by the following

$$\mathrm{d}f_p v \coloneqq v(f). \tag{2.8}$$

By definition, the differential of f is a linear function on $T_p\mathcal{M}$; in other words, $\mathrm{d}f_p$ is an element of $T_p^*\mathcal{M} \coloneqq (T_p\mathcal{M})^*$, the dual space of $T_p\mathcal{M}$ as a linear space.

Definition 2.2.6. The dual space of the tangent space $T_p\mathcal{M}$ at p in $\mathcal{M}$ is called the *cotangent space* and is denoted by $T_p^*\mathcal{M}$. An element of a cotangent space is called a *cotangent vector.*

The differential $\mathrm{d}f_p$ of $f \in C^\infty(\mathcal{M})$ is a cotangent vector. We consider a basis of the cotangent space. Let U, ϕ be a chart centered at $p \in U$, that is $\phi(p) = \mathbf{0}$. We denote a C^∞ function on $(\mathcal{M})$ by x^d such that in an open ball centered at p,

$$x^d(p) = (\mathbf{e}_d)^\top \phi(p), \tag{2.9}$$

that is, a function that is equal to the coordinate function along d-th axis around the point p.

Remark 2.2.7. The differential $\mathrm{d}x^1, \mathrm{d}x^2, \ldots, \mathrm{d}x^D$ of $x^1, x^2, \ldots, x^D$ are well defined; it only depends on the behavior of $x^1, x^2, \ldots, x^D$ in a neighborhood of p.

Proposition 2.2.8. *The set of cotangent vectors*

$$\left\{\mathrm{d}x^1{}_p, \mathrm{d}x^2{}_p, \ldots, \mathrm{d}x^d{}_p\right\}. \tag{2.10}$$

is the basis of $T_p^*\mathcal{M}$. *In addition, it is the dual basis of*

$$\left\{\frac{\partial}{\partial x^1}_p, \frac{\partial}{\partial x^2}_p, \ldots, \frac{\partial}{\partial x^d}_p\right\}, \tag{2.11}$$

that is

$$\mathrm{d}x^{d'}{}_p\left(\frac{\partial}{\partial x^d}_p\right) = \delta^d_{d'}, \tag{2.12}$$

where $\delta^d_{d'}$ *is Kronecker's delta.*

2.3 Metrics and Riemannian manifolds

To consider the length of and angles between curves, we first introduce tangent vectors for the infinitesimal parts of curves. This motivates us to introduce a metric in the tangent space. A manifold with tangent spaces containing a metric is called a Riemannian manifold.

As a preliminary, we introduce a tensor space.

Definition 2.3.1. A (r, s) tensor relative to a vector space $T_p\mathcal{M}$ is a multilinear function

$$T : \underbrace{T_p^*\mathcal{M} \times \cdots \times T_p^*\mathcal{M}}_{r \text{ copies}} \times \underbrace{T_p\mathcal{M} \times \cdots \times T_p\mathcal{M}}_{s \text{ copies}} \to \mathbb{R}. \tag{2.13}$$

The set of the (r, s) tensors are denoted by $\left(\bigotimes^r T_p\mathcal{M}\right) \otimes \left(\bigotimes^s T_p^*\mathcal{M}\right)$. We consider a basis of $\left(\bigotimes^r T_p\mathcal{M}\right) \otimes \left(\bigotimes^s T_p^*\mathcal{M}\right)$. A basis of $\left(\bigotimes^r T_p\mathcal{M}\right) \otimes \left(\bigotimes^s T_p^*\mathcal{M}\right)$ is constructed by the tensor product on the basis of $T_p\mathcal{M}$ and $T_p^*\mathcal{M}$. We define the tensor product of tensors.

Definition 2.3.2. The tensor product of a (r, s) tensor T and (r', s') tensor T' is the $(r + r', s + s')$ tensor $T \otimes T'$ given by

$$\begin{aligned}&(T \otimes T')\left(v^{(1)}, v^{(2)}, \ldots, v^{(r)}, v'^{(1)}, v'^{(2)}, \ldots, v'^{(r')}, v_{(1)}, v_{(2)}, \ldots, v_{(s)}, v'_{(1)}, v'_{(2)}, \ldots, v'_{(s')}\right) \\ &:= T\left(v^{(1)}, v^{(2)}, \ldots, v^{(r)}, v_{(1)}, v_{(2)}, \ldots, v_{(s)}\right) T'\left(v'^{(1)}, v'^{(2)}, \ldots, v'^{(r')}, v'_{(1)}, v'_{(2)}, \ldots, v'_{(s')}\right).\end{aligned} \tag{2.14}$$

Proposition 2.3.3. *The following tensor set is a basis of* $\left(\bigotimes^r T_p\mathcal{M}\right) \otimes \left(\bigotimes^s T_p^*\mathcal{M}\right)$.

$$\left\{\frac{\partial}{\partial x^{d^{(1)}}} \otimes \cdots \otimes \frac{\partial}{\partial x^{d^{(r)}}} \otimes \mathrm{d}x^{d_{(1)}} \otimes \cdots \otimes \mathrm{d}x^{d_{(s)}} \;\middle|\; d^{(1)}, \ldots, d^{(r)}, d_{(1)}, \ldots, d_{(s)} = 1, \ldots, D\right\}, \tag{2.15}$$

In particular, $\left(\bigotimes^r T_p\mathcal{M}\right) \otimes \left(\bigotimes^s T_p^*\mathcal{M}\right)$ *is a* D^{r+s}*-dimensional linear space. The coordinate of tensor* T *with respect to the vector* $\underbrace{\frac{\partial}{\partial x^{d^{(1)}}} \otimes \cdots \otimes \frac{\partial}{\partial x^{d^{(r)}}}}_{r \text{ copies}} \otimes \underbrace{\mathrm{d}x^{d_{(1)}} \otimes \cdots \otimes \mathrm{d}x^{d_{(s)}}}_{s \text{ copies}}$ *in the*

basis is given by

$$T\left(\mathrm{d}x^{d^{(1)}},\dots,\mathrm{d}x^{d^{(r)}},\frac{\partial}{\partial x^{d_{(1)}}},\dots,\frac{\partial}{\partial x^{d_{(s)}}}\right). \tag{2.16}$$

Definition 2.3.4. A (r,s) tensor field is a map $\mathcal{T}:\mathcal{M}\to\bigcup_{p\in\mathcal{M}}\left(\bigotimes^r T_p\mathcal{M}\right)\otimes\left(\bigotimes^s T_p^*\mathcal{M}\right)$ that satisfies $T(p)\in\left(\bigotimes^r T_p\mathcal{M}\right)\otimes\left(\bigotimes^s T_p^*\mathcal{M}\right)$. A (r,s) tensor field is C^∞ at $p\in\mathcal{M}$ if for a chart (U,ϕ) centered at p; the map $\mathcal{M}\to\mathbb{R}^{D^{r+s}}$ given by

$$\left((\mathcal{T}(p))\left(\mathrm{d}x^{d^{(1)}},\dots,\mathrm{d}x^{d^{(r)}},\frac{\partial}{\partial x^{d_{(1)}}},\dots,\frac{\partial}{\partial x^{d_{(s)}}}\right)\right)_{d^{(1)},\dots,d^{(r)},d_{(1)},\dots,d_{(s)}=1,\dots,D}\in\mathbb{R}^{D^{r+s}}, \tag{2.17}$$

is continuous at p. A (r,s) tensor field is said to be a C^∞ tensor field if it is C^∞ at every point at $\mathcal{M}$.

Definition 2.3.5. A metric tensor field is a symmetric positive definite $(0,2)$ tensor field g, i.e., , for every point p at $\mathcal{M}$ and $v,v'\in T_p\mathcal{M}$,

- $g(p)(v,v)>0$ for all non-zero vectors $v\in T_p\mathcal{M}$.
- $g(p)(v,v')=g(p)(v,v')$ for all two vectors $v,v'\in T_p\mathcal{M}$.

Definition 2.3.6. A Riemannian manifold is a pair $(\mathcal{M},g)$ of a C^∞ manifold $\mathcal{M}$ and C^∞ metric tensor g.

A Riemannian manifold has a metric for the tangent space at every point. Based ob the metric tensor, we can define the norm of a tangent vector and inner-product between tangent vectors, which gives the length of an infinitesimal curve and the angle between the curves. Thus, a metric tensor makes a manifold applicable in an engineering setting. In particular, a metric tensor equips a manifold with a structure such as a metric space.

Definition 2.3.7. Let $(\mathcal{M},g)$ be a Riemannian manifold. The distance function $d_{(\mathcal{M},g)}:\mathcal{M}\times\mathcal{M}\to\mathbb{R}_{\geq 0}$ of $(\mathcal{M},g)$ is a function given by

$$d_{(\mathcal{M},g)}(p,p')\coloneqq\inf\left\{\int_{t_0}^{t_1}\sqrt{g\left(\left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_\tau c,\left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_\tau c\right)}\mathrm{d}\tau\;\middle|\;c:[t_0,t_1]\to\mathcal{M}\text{ is a }C^\infty\text{ curve.}\right\}. \tag{2.18}$$

Proposition 2.3.8. *A Riemannian manifold $(\mathcal{M},g)$ is a metric space equipped with the distance function $d_{(\mathcal{M},g)}$.*

Definition 2.3.9. Let $(\mathcal{M},g)$ be a Riemannian manifold and f be a C^∞ function on $\mathcal{M}$. The *gradient vector* of f at p is a tangent vector $(\operatorname{grad}f)_p\in T_p\mathcal{M}$ such that for all $v\in T_p\mathcal{M}$,

$$vf=g(p)\Big(v,(\operatorname{grad}f)_p\Big) \tag{2.19}$$

holds.

2.4 Connection and exponential map

Different from the Euclidean space case, parallelity is not trivially determined in a Riemannian manifold. In this section, we introduce the *connection* of a Riemannian manifold, which evaluates how far a vector field is from being parallel. In the following, the space of all C^∞ vector fields is denoted by $\mathfrak{X}(\mathcal{M})$

Definition 2.4.1. An affine connection in $T\mathcal{M}$ is a map

$$\nabla : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M}) \tag{2.20}$$

written $(X, Y) \mapsto \nabla_X Y$, satisfying following properties:

1. For $f_1, f_2 \in C^\infty(\mathcal{M})$ and $X_1, X_2 \in \mathfrak{X}(\mathcal{M})$,

$$\nabla_{f_1 X_1 + f_2 X_2} Y = f_1 \nabla_{X_1} Y + f_2 \nabla_{X_2} Y. \tag{2.21}$$

2. For $a_1, a_2 \in \mathbb{R}$ and $Y_1, Y_2 \in \mathfrak{X}(\mathcal{M})$,

$$\nabla_X (a_1 Y_1 + a_2 Y_2) = a_1 \nabla_X Y_1 + a_2 \nabla_X Y_2. \tag{2.22}$$

3. For $f \in C^\infty(\mathcal{M})$,

$$\nabla_X (fY) = f \nabla_X (Y) + (Xf)Y. \tag{2.23}$$

Definition 2.4.2. Let $X, Y \in \mathfrak{X}(\mathcal{M})$. The Lie bracket $[X, Y]$ is the C^∞ vector field defined by

$$[X, Y]f = X(Yf) - Y(Xf) \tag{2.24}$$

Definition 2.4.3. Let $\mathcal{M}$ be a smooth manifold and ∇ be a connection in $T\mathcal{M}$. For each smooth curve $c : I \to \mathcal{M}$, the *covariant derivative along* c is defined as the unique operator $D_t : \mathfrak{X}(c) \to \mathfrak{X}(c)$ satisfying

1. For $a_1, a_2 \in \mathbb{R}$ and $X_1, X_2 \in \mathfrak{X}(c)$,

$$D_t(a_1 X_1 + a_2 X_2) = a_1 D_t X_1 + a_2 D_t X_2 \tag{2.25}$$

2. For $f \in C^\infty(I)$,

$$D_t(fX) = \left(\frac{\mathrm{d}}{\mathrm{d}t}\bigg|_t f \right) Xf + f D_t(X). \tag{2.26}$$

3. For every extension $\tilde{X} \in \mathfrak{X}(\mathcal{M})$ of $X \in \mathfrak{X}(c)$,

$$D_t X = \nabla_{\frac{\mathrm{d}}{\mathrm{d}t}\big|_t c} \tilde{X}. \tag{2.27}$$

By a connection, which determines parallelity, we can define a geodesic as a curve whose velocity is parallel everywhere, as follows:

Definition 2.4.4. A smooth curve, $c : I \to \mathcal{M}$, is called a *geodesic* if

$$D_t \left. \frac{\mathrm{d}}{\mathrm{d}t} \right|_t c = 0. \tag{2.28}$$

A geodesic corresponds to a line in a Euclidean space, as the velocity of uniform linear motion in a Euclidean space is also parallel everywhere.

We can also define parallel transportation along a curve.

Definition 2.4.5. A smooth vector field X is said to be *parallel along* c with respect to ∇ if $D_t X = 0$.

Definition 2.4.6. Let $\mathcal{M}$ be a C^∞ manifold, p is a point on the $\mathcal{M}$ and $v \in T_p\mathcal{M}$. The unique maximal geodesic c_v starting at p with initial velocity v, the maximal geodesic that satisfies

$$\begin{aligned} c_v(0) &= p \\ \frac{\mathrm{d}}{\mathrm{d}t} c_v \Big|_0 &= v \end{aligned} \tag{2.29}$$

is called the geodesic with initial point p and initial velocity v.

Definition 2.4.7. Let $\mathcal{M}$ be a C^∞ manifold and ∇ be a connection on it. Define a subset $\mathfrak{E} \subset T\mathcal{M}$, the *domain of the exponential map*, by

$$\mathfrak{E} = \{v \in T\mathcal{M} \mid c_v \text{ is defined on an interval containing } [0,1].\}. \tag{2.30}$$

The *exponential map* $\exp : \mathfrak{E} \to \mathcal{M}$ is defined by

$$\exp(v) \coloneqq c_v(1). \tag{2.31}$$

A connection on a manifold is not unique. However, if we consider a Riemannian manifold, a manifold equipped with a metric, we can expect that the parallelity compatible with the metric gives intuitive results. Thus, we consider the following property.

Definition 2.4.8. Let $(\mathcal{M}, g)$ be a Riemannian manifold. A connection ∇ is called a *metric connection* or compatible with g if for every curve c on $\mathcal{M}$ and parallel vector fields X_1, X_2 along c $g(X_1, X_2)$ is constant along c.

In addition, we consider some symmetry.

Definition 2.4.9. A connection ∇ is called *symmetric* if

$$\nabla_X Y - \nabla_Y X = [X, Y]. \tag{2.32}$$

We can prove that for any metric there is a unique connection that is symmetric and compatible with the metric. We call it Levi-Civita connection.

Definition 2.4.10. Let $(\mathcal{M}, g)$ be a Riemannian manifold. The *Levi-Civita connection* of $(\mathcal{M}, g)$ is the unique connection of $\mathcal{M}$ that is symmetric and compatible with g.

Proposition 2.4.11 (Koszul's formula)**.** *The Levi-Civita connection ∇ of $(\mathcal{M}, g)$ is given by*

$$g(\nabla_X Y, Z) \coloneqq \frac{1}{2}(Xg(Y,Z) + Yg(Z,X) - Zg(Y,Z) - g(Y,[X,Z]) - g(Z,[Y,X]) + g(X,[Z,Y])). \tag{2.33}$$

2.5 Immersion and Embedding

Definition 2.5.1. Let $F : \tilde{\mathcal{M}} \to \mathcal{M}$ be a C^∞ map between two manifolds. At each point $p \in \tilde{\mathcal{M}}$, the *differential of F at p* is the map $\mathrm{d}_p F : T_p\tilde{\mathcal{M}} \to T_{F(p)}\mathcal{M}$ given by

$$\mathrm{d}_p F\left(\left(\frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t_0}\right) c\right) \coloneqq \left(\frac{\mathrm{d}}{\mathrm{d}t}\Big|_{t_0}\right)(F \circ c). \tag{2.34}$$

Definition 2.5.2. A C^∞ $F : \tilde{\mathcal{M}} \to \mathcal{M}$ is said to be an *immersion* at $p \in \tilde{\mathcal{M}}$ if its differential $\mathrm{d}_p F$ is injective. We call $F : \tilde{\mathcal{M}} \to \mathcal{M}$ an *immersion* if it is a immersion at every $p \in \tilde{\mathcal{M}}$. We call an immersion $F : \tilde{\mathcal{M}} \to \mathcal{M}$ an *embedding* if F is also injective.

Definition 2.5.3. Let $\tilde{\mathcal{M}}$ be a C^∞ manifold, $(\mathcal{M}, g)$ be a Riemannian manifold, and $F : \tilde{\mathcal{M}} \to \mathcal{M}$ be an immersion between the two manifolds. The metric $\tilde{g}$ on $\tilde{\mathcal{M}}$, called the *metric induced by F from g*, is defined as follows: for every $p \in \mathcal{M}$ and $v, v' \in T_p\mathcal{M}$,

$$\tilde{g}(p)(v, v') \coloneqq g(F(p))(\mathrm{d}_p F(v), \mathrm{d}_p F(v')). \tag{2.35}$$

2.6 Curvature and space form

Definition 2.6.1. Let $\mathcal{M}$ be a C^∞ manifold. The *curvature tensor* of $\mathcal{M}$ with respect to a connection ∇ is the map $R : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M})$ written $(X, Y, Z) \mapsto R(X, Y)Z$, defined by

$$R(X, Y)Z \coloneqq \nabla_X Z - \nabla_Y Z - \nabla_{[X,Y]} Z. \tag{2.36}$$

Definition 2.6.2. Let $\mathcal{M}$ be a C^∞ manifold. The *sectional curvature* of a two-dimensional linear subspace Π in $T_p\mathcal{M}$ with respect to a connection ∇ is a scalar $K(\Pi)$ defined by

$$K(\Pi) \coloneqq g(R(v_1, v_2)v_2, v_1), \tag{2.37}$$

where $\{v_1, v_2\}$ is an orthonormal basis of Π.

The following proposition gives an intuitive interpretation of the sectional curvature.

Proposition 2.6.3. *Let p be a point in a Riemannian D-manifold $(\mathcal{M}, g)$, and Π be a two-dimensional linear subspace in $T_p\mathcal{M}$. The volume (area) $\mathrm{Vol}\left(B_R(p)\right)$ of the ball (disk) $B_R(p)$*

$$B_R(p) \coloneqq \left\{ \exp(v) \;\middle|\; v \in \Pi, g(p)(v, v) \geq R^2 \right\}. \tag{2.38}$$

satisfies that as $R \to +0$

$$\mathrm{Vol}\left(B_r(p)\right) = \pi R^2 \left(1 - \frac{K(\Pi)}{24} + O()\right), \tag{2.39}$$

where K is the sectional curvature with respect to Levi-Civita connection.

For the definitions of the volume and the proof of the above proposition, see John (2018) (in particular, Problem 10-6). As the above proposition shows, the sectional curvature of Π indicates the extension speed of a space along with the tangent plane Π.

Definition 2.6.4. A Riemannian manifold $(\mathcal{M}, g)$ is called a *space form* if there exists a constant $c \in \mathbb{R}$ such that for all $p \in \mathcal{M}$, for all two-dimensional linear subspace $\Pi \in T_p\mathcal{M}$, $K(\Pi) = c$ holds.

Definition 2.6.5. The D-dimensional Euclidean space $\mathcal{R}^D$ is a D-dimensional Riemannian manifold $\left(\mathbb{R}^D, g\right)$, where $\mathbb{R}^D$ is the D-dimensional real space, which has the atlas $\left\{\left(\mathbb{R}^D, \mathrm{id}\right)\right\}$ that consists of the single chart given by $\mathbb{R}^D$ as a coordinate open set and the identity function $\mathrm{id} : \mathbb{R}^D \to \mathbb{R}^D$ as a coordinate map, and g is given by

$$g\left(\frac{\partial}{\partial x^d}, \frac{\partial}{\partial x^{d'}}\right) = \begin{cases} 1 & \text{if } d = d' \\ 0 & \text{otherwise} \end{cases}, \tag{2.40}$$

with the coordinate is denoted by the symbol $\boldsymbol{x} = \begin{bmatrix} x^1 & x^2 & \dots x^D \end{bmatrix} \in \mathbb{R}^D$.

The D-dimensional sphere $\mathcal{S}^D(R)$ with radius R is a D-dimensional Riemannian manifold $\left(\mathbb{S}^D(R), g\right)$ embedded in $\mathcal{R}^{D+1}$, where $\mathbb{S}^D(R) \subset \mathbb{R}^D$ is given by

$$\mathbb{S}^D(R) \coloneqq \left\{ \boldsymbol{x} \in \mathbb{R}^{D+1} \;\middle|\; \boldsymbol{x}^\top \boldsymbol{x} = R^2 \right\}, \tag{2.41}$$

and g is the metric induced by $\mathrm{id}\big|_{\mathbb{S}^D} : \mathbb{S}^D \to \mathbb{R}^{D+1}$.

Let define $\mathbf{I}_{(1,D)} \in \mathbb{R}^{(D+1)\times(D+1)}$ as follows:

$$\left[\mathbf{I}_{(1,D)}\right]_{d,d'} = \begin{cases} -1 & \text{if } d = d' = 0 \quad , \\ 1 & \text{if } d = d' \neq 0 \quad , \\ 0 & \text{otherwise} \quad . \end{cases} \quad d, d' = 0, 1, \dots, D. \tag{2.42}$$

The D+1-dimensional Lorentzian space $\mathcal{R}^{1,D}$ is a D+1-dimensional pseudo Riemannian manifold $\left(\mathbb{R}^{D+1}, g\right)$, where g is given by

$$g\left(\frac{\partial}{\partial x^d}, \frac{\partial}{\partial x^{d'}}\right) = \begin{cases} -1 & \text{if } d = d' = 0 \quad , \\ 1 & \text{if } d = d' \neq 0 \quad , \\ 0 & \text{otherwise} \quad , \end{cases} \tag{2.43}$$

with the coordinate is denoted by the symbol $\boldsymbol{x} = \begin{bmatrix} x^0 & x^1 & x^2 & \dots x^D \end{bmatrix} \in \mathbb{R}^{D+1}$.

The D-dimensional hyperbolic space $\mathcal{H}^D(R)$ with radius R is a D-dimensional Riemannian manifold $\left(\mathbb{S}^D(R), g\right)$ embedded in $\mathcal{R}^{1,D}$, where $\mathbb{S}^D(R) \subset \mathbb{R}^D$ is given by

$$\mathbb{H}^D(R) \coloneqq \left\{ \boldsymbol{x} \in \mathbb{R}^{D+1} \;\middle|\; \boldsymbol{x}^\top \mathbf{I}_{(1,D)} \boldsymbol{x} = R^2 \right\}, \tag{2.44}$$

and g is the metric induced by $\mathrm{id}\big|_{\mathbb{H}^D} : \mathbb{H}^D \to \mathbb{R}^{D+1}$.

In this thesis, we primarily focus on machine learning over Euclidean spaces, Spheres, and Hyperbolic spaces. This is justified based to the following theorem.

Theorem 2.6.6 (Killing-Hopf)**.** *Let $(\mathcal{M}, g)$ be a complete, simply connected Riemannian D-manifold with constant sectional curvature, $D \geq 2$. Then $(\mathcal{M}, g)$ is isometric to one of the model spaces $\mathcal{R}^D$, $\mathcal{S}^D(R)$, and $\mathcal{H}^D(R)$.*

2.7 Riemannian gradient descent

In a linear space, the gradient method is defined by the vector addition operator. On the other hand, in a Riemannian gradient descent, in which no vector addition operator is defined, the definition of a gradient descent method is not trivial.

The naïve Riemannian gradient method is

$$p^{(k+1)} \to \exp_p^{(k)} \left(-\eta (\operatorname{grad} f)_{p^{(k)}} \right), \tag{2.45}$$

where $p^{(k)}$ indicates the point at step k in optimization, and $\eta > 0$ is a learning rate. If we replace $(\operatorname{grad} f)_{p^{(k)}}$ by a stochastic gradient $\tilde{v}^{(k)}$, a random variable in $T_p\mathcal{M}$ that satisfies $\mathbb{E}\left[\tilde{v}^{(k)}\right] = (\operatorname{grad} f)_{p^{(k)}}$, we obtain the naïve Riemannian stochastic gradient (RSG) method:

$$p^{(k+1)} \to \exp_p^{(k)} \left(-\eta \tilde{v}^{(k)} \right). \tag{2.46}$$

In Chapter 3 and Chapter 4, we propose models in machine learning over space forms using the naïve RSG. In Chapter 5, we compare naïve RSG with another kind of gradient method for Riemannian manifolds: natural gradient method.

Chapter 3

Relational Data Embedding in Space Forms I: Ordinal Embedding

3.1 Motivation

In this chapter, we study the problem of ordinal embedding, a.k.a. non-metric multidimensional scale (Shepard, 1962a,b; Kruskal, 1964a,b; Shepard, 1966). Given a set of objects $1, 2 \dots N$, the weights of dissimilarity $\xi(i, j)$ for all the object pairs $i, j \in 1, 2, \dots, N$ are unknown but some ordinal relations such as $\xi(i, j) < \xi(k, l)$ can be derived. The aim of ordinal embedding is then to obtain a set of embeddings $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N$ in a low-dimensional space, so that ordinal relations are preserved. To a large extent, existing ordinal embeddings use the D-dimensional Euclidean space $\mathbb{R}^D$ to achieve

$$\xi(i, j) < \xi(k, l) \Rightarrow \|\boldsymbol{x}_i - \boldsymbol{x}_j\| < \|\boldsymbol{x}_k - \boldsymbol{x}_l\|. \tag{3.1}$$

When $i = k$ always holds, it is a special case in ordinal embedding, known as triplet embedding (Van Der Maaten and Weinberger, 2012; Wang et al., 2018).

Existing ordinal embedding methods could be roughly divided into two categories: the probabilistic-model-based (Tamuz et al., 2011; Van Der Maaten and Weinberger, 2012) and the margin-loss-based (Agarwal et al., 2007; Terada and Luxburg, 2014). The former mainly focuses on constructing a parametric probabilistic model, where the maximum likelihood estimator is used for embeddings. The latter achieves embeddings by optimizing a margin loss function. These methods are effective on preserving ordinal structure in a space of low dimension compared to original data size, but have largely ignored the optimality of a space for embedding, which is essential for embedding into a much lower-dimensional space. Ideally, the chosen low-dimensional space should be compatible with the true data structure, so that embedding can be achieved in a much lower-dimensional space with low computational cost and overfitting avoided.

However, current ordinal embedding methods use Euclidean space as a primary choice, mainly due to natural generalization of intuition-friendly and visual three-dimensional space (Ganea et al., 2018a). These methods may not be able to reflect semantic dissimilarities between objects or demand a substantial increases in model complexity and computational cost, especially when data come from hierarchical structure, whereas the hierarchical structure is exhibited in reality by many types of complex data, such as datasets with power-law distributions, in natural language area and scale-free networks (Krioukov et al., 2010; Nickel and Kiela, 2017). Take a hierarchical structure given by a complete balanced binary tree in Figure 3.1 as an example. The number of objects in each layer grows exponentially with respect to h, which is given by 2^h. However, the expanding speed of Euclidean space is polynomial (slower than exponential) as the circumference

$C(R)$ of radius R is given by $C(R) = 2\pi \sinh R \approx \pi \exp R$. This motivates us to seek a feasible non-Euclidean space that expands exponentially so as to achieve effective ordinal embeddings by capturing the hierarchical structure.

Inspired by the above, we focus on sectional curvature κ, which characterizes the expanding speed of a space. According to Bertrand-Diguet-Puiseux theorem, which claims that $C(R) = 2\pi(R - \frac{1}{6}\kappa R^3) + O(R^4)$ as $R \to +0$, achieving faster expanding speed than polynomial requires lower curvature, i.e., negative curvature. On the other hand, in essence there is no constant negative curvature space other than hyperbolic (Killing-Hopf theorem e.g., in (Lee, 2006)). Fortunately, the hyperbolic space of two dimension or higher has exponential expanding speed. Specifically, in the two-dimensional hyperbolic space the circumference is given by $C(R) = 2\pi \sinh R \approx \pi \exp R$. Such exponential expanding speed explicitly matches the hierarchical structure, as shown in Figure 3.1. Moreover, Sarkar (2011) has theoretically explained that given an arbitrary tree, we have embeddings of its vertices with arbitrary small distance distortion in the two-dimensional hyperbolic space. These facts demonstrate the hierarchy-friendly property of hyperbolic space in low-dimensional setting, which satisfies our motivation. This preferable property of hyperbolic space in embedding has been supported by recent success of hyperbolic space in many embedding settings and applications such as visualization (Lamping and Rao, 1994), self-organizing map (Ritter, 1999), multi-dimensional scaling Walter (2004); Sala et al. (2018), graph embedding (Shavitt and Tankel, 2008; Nickel and Kiela, 2017), embedding from graph Laplacian (Alanis-Lobato et al., 2016), Internet graph embedding (Shavitt and Tankel, 2008), and visualization of large taxonomies (Nickel and Kiela, 2017).

In this chapter, we propose a novel hyperbolic ordinal embedding (HOE) model to capture hierarchical structure and preserve ordinal relations simultaneously. Furthermore, we prove the suitability of hyperbolic space and limitations of Euclidean space for ordinal relation with hierarchical structure in theory.

We summarize our main contributions as follows:

- A hyperbolic ordinal embedding (HOE) is proposed to embed hierarchical structure data into an extremely low-dimensional hyperbolic space. We reformulate the ordinal embedding problem into a general metric space setting with hyperbolic space setting as a special case, and then propose two simple yet effective continuous loss functions for probabilistic-model-based and margin-loss-based models, respectively.
- We give theoretical analyses to clarify advantages of using hyperbolic space against Euclidean approach (in Section 3.6) in terms of ordinal embedding for hierarchical structural data: (1) for Euclidean space of any dimension, there exist ordinal relations that cannot be preserved in embeddings; (2) the use of hyperbolic space can achieve effective embedding with ordinal relations preserved in a space of extremely low (e.g.,2) dimensionality.
- Experiments on both artificial and real datasets have demonstrated that the proposed method outperforms existing Euclidean-space-based baselines for embedding hierarchical structure data in a significantly low-dimensional (e.g., 2, 4, 8, 16) space.

3.2 Related Work

Various ordinal embedding approaches have been proposed. Under probabilistic-model-based setting, CLK (Tamuz et al., 2011) was proposed to reduce the complexity of obtaining high quality approximations of similarity triplets via an information theoretic adaptive sampling approach. Considering that using similarity triplets is insufficient for obtaining a truthful embedding of objects, t-STE (Van Der Maaten and Weinberger, 2012) was then

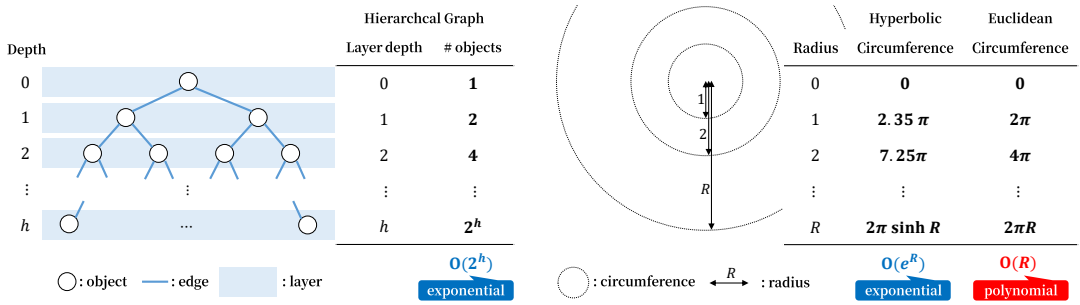

Fig. 3.1. Exponential growth of objects in a hierarchical data and space expansion speed of hyperbolic and Euclidean space.

proposed to collapse similar points and repel dissimilar points in the embedding without resulting in additional constraint violations. Under margin-loss-based setting, G-NMD (Agarwal et al., 2007) aimed to embed data when ordinal relations can be contradictory and need not be specified for all pairs of dissimilarities. Regarding that the similarities of objects may not be mutually consistent according to different tasks, McFee and Lanckriet (2011) integrated heterogeneous data so as to optimally conform to measurements of perceptual similarity. Later, LOE (Terada and Luxburg, 2014) was proposed to achieve embedding that not only preserves the ordinal constraints, but also the density structure of dataset. Though the effectiveness of existing ordinal embeddings has been demonstrated, none of them have paid attention to the compatibility of embedding space and achieved embedding in hyperbolic space.

Recently, hyperbolic space has been extensively studied in many research areas (Alanis-Lobato et al., 2016; Nickel and Kiela, 2017; Sala et al., 2018). For example, The relation between human visual space and hyperbolic geometry has been suggested Luneburg (1947); Indow (1967). Lamping and Rao (1994) proposed a scheme for visualizing and manipulating large hierarchies by laying out the hierarchy uniformly on the hyperbolic plane and map this plane onto a circular display region. Ritter (1999) proposed a self-organizing map that is based on discretizations of curved, non-Euclidean spaces. Walter (2004) proposed a projection-based visualization method for high-dimensional data sets by combining concepts from multidimensional scaling and the geometry of the hyperbolic spaces. Shavitt and Tankel (2008) embeded Internet data in hyperbolic space, since Internet structure has a highly connected core and long stretched tendrils, where most of the routing paths between nodes in the tendrils pass through the core. To enhance the efficiency of embedding of big networks, Alanis-Lobato et al. (2016) then used a Laplacian-based model for geometric analysis of big networks. Poincaré Embedding (Nickel and Kiela, 2017) aimed at learning representations of symbolic data so that it simultaneously learns the similarity and the hierarchy of objects. Later, Ganea et al. (2018b) bridged the gap between hyperbolic and Euclidean geometry in the context of neural networks and deep learning by generalizing deep neural models to the Poincaré model of the hyperbolic geometry. Balancing the trade-off between precision and dimensionality of embedding, H-MDS (Sala et al., 2018) was proposed as a general approach that can embed trees into hyperbolic space with arbitrarily low distortion. Although these approaches can achieve effective embedding by capturing hierarchy structure with hyperbolic space, the ordinal relations which often naturally exist among data cannot be utilized by them.

3.3 Hyperbolic Geometry

In this section, we introduce basic notations and then briefly review hyperbolic geometry with its real coordinate space representation.

Notations Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Z}$, and $\mathbb{Z}_{>0}$ denote the real number set, non-negative real number set, integer set, and positive integer set, respectively. We denote D-dimensional real coordinate space and $D \times D'$ real matrix space by $\mathbb{R}^D$ and $\mathbb{R}^{D \times D'}$, respectively. We let $\mathbf{0}_D \in \mathbb{R}^D$ and $\mathbf{I}_D \in \mathbb{R}^{D \times D}$ denote the D-dimensional zero vector and D-dimensional identity matrix, respectively. $\mathrm{sgn} : \mathbb{R} \to \{-1, 0, 1\}$ denotes the sign function defined by

$$\mathrm{sgn}\,(x) \coloneqq \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ +1 & x > 0 \end{cases}, \tag{3.2}$$

For $N \in \mathbb{Z}_{>0}$, we denote the set $\{1, 2, \ldots, N\}$ by $[N]$.

Hyperbolic Geometry in Coordinate Space Since there is unique hyperbolic space up to similarity if the dimension is fixed, in the following, we fix the sectional curvature of hyperbolic space to be -1, that is, $\kappa = -1$ for simplicity of discussion. For hyperbolic space, there exist several models, i.e., ways of representation in real coordinate space, such as the hyperboloid model, Klein disk model, Poincaré disk model and Poincaré upper plain model. As these models are isometric to one another, the discussion on the distance structure of hyperbolic space in one model is equivalent to that in another model. In the following, we explain hyperbolic space using the hyperboloid model. The D-dimensional hyperbolic space $\mathcal{H}^D$ is a metric space $\left(\mathbb{H}^D, d_{\mathbb{H}^D}\right)$, where $\mathbb{H}^D$ and $d_{\mathbb{H}^D} : \mathbb{H}^D \times \mathbb{H}^D \to \mathbb{R}_{\geq 0}$ are defined by

$$\begin{aligned} \mathbb{H}^D &\coloneqq \left\{ \boldsymbol{x} \in \mathbb{R}^{D+1} \;\middle|\; \boldsymbol{x}^\top \boldsymbol{G}_{\mathrm{M}} \boldsymbol{x} = -1, x^0 > 0 \right\} \\ d_{\mathbb{H}^D}(\boldsymbol{x}, \boldsymbol{y}) &\coloneqq \mathrm{arcosh}\left(-\boldsymbol{x}^\top \boldsymbol{G}_{\mathrm{M}} \boldsymbol{y}\right), \end{aligned} \tag{3.3}$$

where arcosh denotes the area hyperbolic cosine function (the inverse function of the hyperbolic cosine function), and $\boldsymbol{G}_{\mathrm{M}}$ denotes

$$\boldsymbol{G}_{\mathrm{M}} \coloneqq \begin{bmatrix} -1 & \mathbf{0}_D^\top \\ \mathbf{0}_D & \mathbf{I}_D \end{bmatrix} \in \mathbb{R}^{(D+1) \times (D+1)}. \tag{3.4}$$

3.4 Euclidean Ordinal Embedding

We consider embedding problem of $N \in \mathbb{Z}_{>0}$ objects. In the following, we identify the N objects with the integer set $[N]$. Let the sequence $\mathcal{S} = (((i_s, j_s), (k_s, l_s)), y_s)_{s=1}^S$ be an ordinal data set, in which $i_s, j_s, k_s, l_s \in [N]$ and $y_s \in \{-1, +1\}$ for $s = 1, 2, \ldots, S$. Here, if $y_s = -1$, i_s and j_s are more similar to each other than k_s and l_s i.e., the dissimilarity between i_s and j_s are larger than that between k_s and l_s, and otherwise if $y_s = +1$. An ordinal data set $\mathcal{S} = (((i_s, j_s), (k_s, l_s)), y_s)_{s=1}^S$ is called an ordinal triplet set if $i_s = k_s$ is satisfied for all $s \in [S]$. The D-dimensional Euclidean space denoted by $\mathcal{R}^D$ is a metric space $\left(\mathbb{R}^D, d_{\mathbb{R}^D}\right)$, where $d_{\mathbb{R}^D} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}_{\geq 0}$ is given by $d_{\mathbb{R}^D}(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \sqrt{(\boldsymbol{x} - \boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y})}$.

Existing ordinal embedding using the D-dimensional Euclidean space $\mathcal{R}^D$ is to obtain embedding $x_n \in \mathbb{R}^D$ for $n \in [N]$ such that

$$\operatorname{sgn}\left(d_{\mathbb{R}^D}(x_{i_s}, x_{j_s}) - d_{\mathbb{R}^D}(x_{k_s}, x_{l_s})\right) = y_s \tag{3.5}$$

is satisfied for as many $s \in [S]$ as possible.

Denote the **P**robabilistic-model-based **O**rdinal **E**mbedding and the **M**argin-loss-based **O**rdinal **E**mbedding as **POE** and **MOE**, respectively. In both Euclidean POE and MOE, the loss function of $(x_n)_{n\in[N]}$ on ordinal data $\mathcal{S} = (((i_s, j_s), (k_s, l_s)), y_s)_{s=1}^S$ is given by

$$\mathcal{L}\Big(\mathcal{S}; (x_n)_{n\in[N]}\Big) \coloneqq \frac{1}{S}\sum_{s\in[S]} \ell\Big(((i_s, j_s), (k_s, l_s), y_s); (x_n)_{n\in[N]}\Big), \tag{3.6}$$

with their own specific one point loss function ℓ of $(x_n)_{n\in[N]}$ on one point ordinal datum $(((i, j), (k, l)), y)$, which quantifies how large the contradiction of the embeddings to the one point ordinal datum is.

• Euclidean POE (EPOE) For the object quadruple $((i, j), (k, l))$, the probability of $y = -1$ is high if the distance $d_{\mathbb{X}}(x_i, x_j)$ is shorter than $d_{\mathbb{X}}(x_k, x_l)$ and the probability of $y = +1$ is high otherwise. The dependency of the distribution of y on the distances is defined by a decreasing function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Then, we have the following probabilistic model.

$$\Pr\Big(y|((i, j), (k, l)); (x_n)_{n\in[N]}\Big) \coloneqq \begin{cases} \dfrac{f(d_{\mathbb{R}^D}(x_i, x_j))}{f(d_{\mathbb{R}^D}(x_i, x_j)) + f(d_{\mathbb{R}^D}(x_k, x_l))} & y = -1 \\ \dfrac{f(d_{\mathbb{R}^D}(x_k, x_l))}{f(d_{\mathbb{R}^D}(x_i, x_j)) + f(d_{\mathbb{R}^D}(x_k, x_l))} & y = +1 \end{cases} \tag{3.7}$$

We call f a kernel function. The loss function ℓ_{prb} of $(x_n)_{n\in[N]}$ on one point ordinal datum $(((i, j), (k, l)), y)$ is given by

$$\ell_{\mathrm{prb}}\Big(((i, j), (k, l)), y); (x_n)_{n\in[N]}\Big) \coloneqq -\log \Pr\Big(y|((i, j), (k, l)); (x)_{n\in[N]}\Big). \tag{3.8}$$

Then, the loss function in EPOE of $(x_n)_{n\in[N]}$ on ordinal data $\mathcal{S} = (((i_s, j_s), (k_s, l_s)), y_s)_{s=1}^S$ is derived by substituting $\ell = \ell_{\mathrm{prb}}$ to (3.6). Take one of the most representative approaches, stochastic triplet embedding (Van Der Maaten and Weinberger, 2012), as an example. The probabilistic model given by (3.7) is reduced to that of the stochastic triplet embedding and t-distributed stochastic triplet embedding in (Van Der Maaten and Weinberger, 2012) with the Gaussian kernel $f(d) = \exp\big(-d^2\big)$ and Student's t-distribution kernel $f(d) = \left(1 + \frac{d^2}{\alpha}\right)^{\alpha}$, respectively. Note that in (Van Der Maaten and Weinberger, 2012), only are ordinal triplet data cases considered, while we above generalized it into general ordinal data cases.

• Euclidean MOE (EMOE) We define a soft margin loss for this approach (Agarwal et al., 2007; Terada and Luxburg, 2014). The soft margin loss function ℓ_{mgn} of $(x_n)_{n\in[N]}$ on one point ordinal datum $(((i, j), (k, l)), y)$ is given by

$$\ell_{\mathrm{mgn}}\Big(((i, j), (k, l)), y); (x_n)_{n\in[N]}\Big) \coloneqq \Big\{\left[\delta - (d_{\mathbb{R}^D}(x_{i_s}, x_{j_s}) - d_{\mathbb{R}^D}(x_{k_s}, x_{l_s})) \cdot y_s\right]_+\Big\}^q, \tag{3.9}$$

where $\delta \in \mathbb{R}_{\geq 0}$ is a margin hyperparameter and $q \in \mathbb{R}_{\geq 0}$ is a power index which adjusts the loss. Then, the loss function in EMOE of $(x_n)_{n\in[N]}$ on ordinal data $\mathcal{S} =$

$(((i_s, j_s), (k_s, l_s)), y_s)_{s=1}^S$ is derived by substituting $\ell = \ell_{\mathrm{mgn}}$ to (3.6). The loss function in (3.9) is reduced to that of the soft margin model in (Terada and Luxburg, 2014) if $q = 2$, and is indirectly reduced to the loss function in (Agarwal et al., 2007) if $q = 1$, whereas they obtain the distance matrix in $\mathcal{R}^D$ with $D = N$ in (Agarwal et al., 2007), instead of directly obtaining embeddings in $\mathcal{R}^D$.

3.5 Hyperbolic Ordinal Embedding

Our motivation is ordinal embedding in hyperbolic space. The key idea is to generalize existing methods into those in general metric spaces and to obtain our hyperbolic ordinal embedding as a special case.

3.5.1 General Ordinal Embedding

In this section, we obtain ordinal embedding in a general metric space $\mathcal{X} = (\mathbb{X}, d_{\mathbb{X}})$, where $\mathbb{X}$ is a point set and $d_{\mathbb{X}} : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_{\geq 0}$ is the distance function defined in $\mathbb{X}$.

Problem Settings

Sharing the same motivation as the Euclidean case, the objective of embedding objects $[N]$ in metric space $\mathcal{X}$ is to realize embedding $x_n \in \mathbb{X}$ for $n \in [N]$ such that

$$\operatorname{sgn}\left(d_{\mathbb{X}}(x_{i_s}, x_{j_s}) - d_{\mathbb{X}}(x_{k_s}, x_{l_s})\right) = y_s \tag{3.10}$$

is satisfied for as many $s \in [S]$ as possible. Therefore, the ordinal embedding is formulated as minimizing the classification loss function, as defined below.

Definition 3.5.1 (Classification Loss Function)**.** Let $[N]$ be objects and $(x_n)_{n\in[N]}$ be their embeddings. The classification loss function of $(x_n)_{n\in[N]}$ on ordinal datum $(((i, j), (k, l)), y)$, in which $i, j, k, l \in [N]$ and $y \in \{\pm 1\}$, is defined by

$$\ell_{\mathrm{cls}}\Big((((i, j), (k, l)), y); (x_n)_{n\in[N]}\Big) \coloneqq \begin{cases} 0 & \operatorname{sgn}\left(d_{\mathbb{X}}(x_{i_s}, x_{j_s}) - d_{\mathbb{X}}(x_{k_s}, x_{l_s})\right) = y_s \\ 1 & \operatorname{sgn}\left(d_{\mathbb{X}}(x_{i_s}, x_{j_s}) - d_{\mathbb{X}}(x_{k_s}, x_{l_s})\right) \neq y_s \end{cases}, \tag{3.11}$$

The classification loss function of embedding $(x_n)_{n\in[N]}$ on ordinal data $\mathcal{S} = (((i_s, j_s), (k_s, l_s)), y_s)_{s=1}^S$, in which $i_s, j_s, k_s, l_s \in [N]$ and $y_s \in \{\pm 1\}$ for all $s \in [N]$, is defined by

$$\mathcal{L}_{\mathrm{cls}}\Big(\mathcal{S}; (x_n)_{n\in[N]}\Big) \coloneqq \frac{1}{S} \sum_{s\in S} \ell_{\mathrm{cls}}\Big((((i_s, j_s), (k_s, l_s)), y); (x_n)_{n\in[N]}\Big). \tag{3.12}$$

The embedding $(x_n)_{n\in[N]}$ is called *non-contradictory to* $\mathcal{S}$ if $\mathcal{L}_{\mathrm{cls}}\Big(\mathcal{S}; (x_n)_{n\in[N]}\Big) = 0$.

Loss Functions

As the loss function in Definition 3.5.1 is a hard classification loss, it is not easy to optimize due to the discontinuity of the sign function. We first consider relaxation of the original loss function, and then introduce a probabilistic model and soft margin based loss function as a specific loss function. The ideal conditions for the loss function are listed as follows:

- The loss function should be continuous with respect to the embeddings $(x_n)_{n\in[N]}$.
- For ordinal data $(((i, j), (k, l)), -1)$ in $\mathcal{S}$, the loss function should be decreasing with respect to the distance $d_{\mathbb{X}}(x_i, x_j)$ and should be increasing with respect to $d_{\mathbb{X}}(x_k, x_l)$, and vice versa for $(((i, j), (k, l)), +1)$.

Therefore, we consider the loss function $\mathcal{L}$ of the following form

$$\mathcal{L}\Big(\mathcal{S}; (x_n)_{n\in[N]}\Big) \coloneqq \frac{1}{S}\sum_{s\in[S]} \ell\Big(((i_s, j_s), (k_s, l_s), y_s); (x_n)_{n\in[N]}\Big), \tag{3.13}$$

with one datum loss function ℓ given by

$$\ell\Big(((i, j), (k, l), y); (x_n)_{n\in[N]}\Big) \coloneqq g(d_{\mathbb{X}}(x_i, x_j), d_{\mathbb{X}}(x_k, x_l); y), \tag{3.14}$$

where $g : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times \{\pm 1\}, (d, d', y) \mapsto g(d, d'; y)$ satisfies the following:

- $g(d, d'; -1)$ is decreasing with respect to d and is increasing with respect to d'.
- $g(d, d'; +1)$ is increasing with respect to d and is decreasing with respect to d'.

This general idea allows us to apply in hyperbolic space analogical ideas to EPOE and EMOE. As a result, we obtain specific loss functions, GPOE and GMOE, as shown below.

• General POE (GPOE) One way to avoid the discontinuous loss function is to introduce a probabilistic model, as in EPOE. We design a conditional probability distribution model of y, as follows:

$$\Pr\Big(y|((i, j), (k, l)); (x_n)_{n\in[N]}\Big) \coloneqq \begin{cases} \dfrac{f(d_{\mathbb{X}}(x_i, x_j))}{f(d_{\mathbb{X}}(x_i, x_j)) + f(d_{\mathbb{X}}(x_k, x_l))} & y = -1 \quad , \\ \dfrac{f(d_{\mathbb{X}}(x_k, x_l))}{f(d_{\mathbb{X}}(x_i, x_j)) + f(d_{\mathbb{X}}(x_k, x_l))} & y = +1 \quad , \end{cases} \tag{3.15}$$

where $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a kernel function. By the above probabilistic model, one point loss function in GPOE ℓ_{prb} of $(x_n)_{n\in[N]}$ on one point ordinal datum $(((i, j), (k, l)), y)$ is given by

$$\ell_{\mathrm{prb}}\Big(((i, j), (k, l)), y); (x_n)_{n\in[N]}\Big) \coloneqq -\log \Pr\Big(y|((i, j), (k, l)); (x)_{n\in[N]}\Big). \tag{3.16}$$

Then, the loss function in GPOE of $(x_n)_{n\in[N]}$ on ordinal data $\mathcal{S} = (((i_s, j_s), (k_s, l_s)), y_s)_{s=1}^S$ is derived by substituting $\ell = \ell_{\mathrm{prb}}$ to (3.13). When $\mathbb{X}$ is the D-dimensional Euclidean space $\mathbb{R}^D$, GPOE is reduced to EPOE.

• General MOE (GMOE) Another way to avoid the discontinuous loss function is to replace it by a soft loss function, as in EMOE. We define a soft margin loss as follows. The one point soft margin loss function ℓ_{mgn} of $(x_n)_{n\in[N]}$ on one point ordinal datum $(((i, j), (k, l)), y)$ is given by

$$\ell_{\mathrm{mgn}}\Big(((i, j), (k, l)), y); (x_n)_{n\in[N]}\Big) \coloneqq \Big\{[\delta - (d_{\mathbb{X}}(x_{i_s}, x_{j_s}) - d_{\mathbb{X}}(x_{k_s}, x_{l_s})) \cdot y_s]_+\Big\}^q, \tag{3.17}$$

where $\delta \in \mathbb{R}_{\geq 0}$ is a margin hyperparameter and $q \in \mathbb{R}_{\geq 0}$ is a power index which adjusts the loss. Then, the loss function in GMOE of $(x_n)_{n\in[N]}$ on ordinal data $\mathcal{S} = (((i_s, j_s), (k_s, l_s)), y_s)_{s=1}^S$ is derived by substituting $\ell = \ell_{\mathrm{mgn}}$ to (3.13). When $\mathbb{X}$ is the D-dimensional Euclidean space $\mathbb{R}^D$, GMOE is reduced to EMOE.

3.5.2 Hyperbolic Ordinal Embedding

With the generalization in Section 3.5.1, Hyperbolic POE and MOE can be obtained by substituting $\mathbb{X} = \mathbb{H}^D$ to (3.15) and (3.17), respectively, where $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N \in \mathbb{H}^D$.

• Hyperbolic POE (HPOE) The probabilistic model of HPOE using the D-dimensional hyperbolic space $\mathcal{H}^D$ is derived by substituting $\mathbb{X} = \mathbb{H}^D$ to (3.15) as follows:

$$\Pr\Big(y|((i,j),(k,l));(\boldsymbol{x}_n)_{n\in[N]}\Big) \coloneqq \begin{cases} \dfrac{f(d_{\mathbb{H}^D}(\boldsymbol{x}_i,\boldsymbol{x}_j))}{f(d_{\mathbb{H}^D}(\boldsymbol{x}_i,\boldsymbol{x}_j))+f(d_{\mathbb{H}^D}(\boldsymbol{x}_k,\boldsymbol{x}_l))} & y=-1 \\ \dfrac{f(d_{\mathbb{H}^D}(\boldsymbol{x}_k,\boldsymbol{x}_l))}{f(d_{\mathbb{H}^D}(\boldsymbol{x}_i,\boldsymbol{x}_j))+f(d_{\mathbb{H}^D}(\boldsymbol{x}_k,\boldsymbol{x}_l))} & y=+1 \end{cases}. \tag{3.18}$$

By substituting (3.18) to (3.14), we have the one point loss function ℓ_{prb} of HPOE.

• Hyperbolic MOE (HMOE) The one point loss function of HMOE using the D-dimensional hyperbolic space $\mathcal{H}^D$ is derived by substituting $\mathbb{X} = \mathbb{H}^D$ to (3.17) as follows:

$$\ell_{\mathrm{mgn}}\Big(((i,j),(k,l)),y);(\boldsymbol{x}_n)_{n\in[N]}\Big) \coloneqq \Big\{[\delta - (d_{\mathbb{H}^D}(\boldsymbol{x}_{i_s},\boldsymbol{x}_{j_s}) - d_{\mathbb{H}^D}(\boldsymbol{x}_{k_s},\boldsymbol{x}_{l_s}))\cdot y_s]_+\Big\}^q. \tag{3.19}$$

Here, as in (3.17), $\delta \in \mathbb{R}_{\geq 0}$ is a margin hyperparameter and $q \in \mathbb{R}_{\geq 0}$ is a power index which adjusts the loss.

3.5.3 Optimization

Similar to (Van Der Maaten and Weinberger, 2012), we apply the stochastic gradient method to optimize (3.13). Note that the following optimization method can be applied to the loss function of HPOE and HMOE, because the loss functions of these methods are special cases of that in (3.13). We uniformly at random choose a subsequence $\mathcal{B}$ of $[S]$ and substitute $\mathcal{B}$ for $[S]$, then we have a stochastic loss of the loss in (3.13) as follows:

$$\tilde{\mathcal{L}}\Big(\mathcal{S};(x_n)_{n\in[N]}\Big) \coloneqq \frac{1}{|\mathcal{B}|}\sum_{s\in\mathcal{B}} \ell\Big((((i_s,j_s);(k_s,l_s)),y),(\boldsymbol{x}_n)_{n\in[N]}\Big), \tag{3.20}$$

where $|\mathcal{B}|$ denotes the number of elements in $\mathcal{B}$. Then, we use the gradient of (3.20) as a stochastic gradient of the loss function in (3.13) and then optimize the loss function in (3.13) by stochastic Riemannian sub gradient method (Zhang and Sra, 2016). The update rule is given by

$$\boldsymbol{x}_n \leftarrow \exp_{\boldsymbol{x}_n}\left(\pi_{\boldsymbol{x}_n}\left(\boldsymbol{G}_{\boldsymbol{x}_n}^{-1}\frac{\partial}{\partial \boldsymbol{x}_n}\tilde{\mathcal{L}}\right)\right), \tag{3.21}$$

where $\boldsymbol{G}_{\boldsymbol{x}_n}$ denotes the metric matrix on $\boldsymbol{x}_n$, $\pi_{\boldsymbol{x}}$ denotes the projection to the tangent space on $\boldsymbol{x}_n$, and $\exp_{\boldsymbol{x}}$ denotes the exponential map on $\boldsymbol{x}_n$. In the D-dimensional hyperbolic space, the formulae for these operations appear in e.g., (Nickel and Kiela, 2018) as follows.

$$\begin{aligned} \boldsymbol{G}_{\boldsymbol{x}} &= \boldsymbol{G}_{\mathrm{M}} \quad (\text{in (3.4)}), \\ \pi_{\boldsymbol{x}}(\boldsymbol{v}') &= \boldsymbol{v}' + (\boldsymbol{x}^\top \boldsymbol{G}_{\mathrm{M}}\boldsymbol{x})\boldsymbol{x}, \\ \exp_{\boldsymbol{x}}(\boldsymbol{v}) &= \cosh\left(\sqrt{\boldsymbol{v}^\top \boldsymbol{G}_{\mathrm{M}}\boldsymbol{v}}\right)\boldsymbol{x} + \mathrm{sinhc}\left(\sqrt{\boldsymbol{v}^\top \boldsymbol{G}_{\mathrm{M}}\boldsymbol{v}}\right)\boldsymbol{v}, \end{aligned} \tag{3.22}$$

where sinhc denotes the hyperbolic sine cardinal function, which is given by

$$\mathrm{sinhc}\, x = \begin{cases} \frac{\sinh x}{x} & x \neq 0 \\ 1 & x = 0 \end{cases}. \tag{3.23}$$

Using these formulae, we can optimize the loss function of HPOE and HMOE. Note that we can also apply the above optimization method in the Poincaré disk model of hyperbolic space by the formulae that appears in e.g., (Ganea et al., 2018a), although the result is isometric to the formulae for the hyperboloid model in this section. In both of the hyperboloid model and the Poincaré disk model, the time complexity of each operator in the above optimization is linear to dimension. Therefore, the computational cost of each step in the optimization is given by $O(D|\mathcal{B}|)$. This is not larger than the computational complexity of the previous work.

3.6 Hyperbolic vs. Euclidean

In this section, we discuss the theoretical advantages of using hyperbolic space against using Euclidean space. Our interest is the situation in which the ordinal data comes from ground-truth hierarchical structure. For formal discussion, we define such a situation as the case where we have *graphical ordinal data* of a graph that is a tree, which intuitively gives a hierarchical structure as in Figure 3.1. In this section, after defining graphical ordinal data as a preliminary, we discuss hyperbolic and Euclidean space cases.

Preliminary: Graphical Ordinal Data

Definition 3.6.1. Let $\mathcal{G} = ([N], \mathcal{E})$ be an undirected graph with a vertex set $[N]$ and an edge set $\mathcal{E}$. We denote by $d_{\mathcal{G}}$ the graph distance function. A sequence $\mathcal{S} = (((i_s, j_s), (k_s, l_s)), y_s)_{s=1}^{S}$ is called *graphical ordinal data (GOD)* of $\mathcal{G}$ when

$$\operatorname{sgn}\left(d_{\mathcal{G}}(i_s, j_s) - d_{\mathcal{G}}(k_s, l_s)\right) = y_s \tag{3.24}$$

is satisfied for all $s \in [S]$. GOD are called *graphical ordinal triplet data (GOTD)* of $\mathcal{G}$ if $i_s = k_s$ is satisfied for all $s \in [S]$, and GOD are called *complete* if for all pairs $((i, j), (k, l))$ of vertex pair such that $d_{\mathcal{G}}(i, j) - d_{\mathcal{G}}(k, l) \neq 0$, there exists $s \in [S]$ such that either of the following is satisfied.

- $((i_s, j_s), (k_s, l_s)) = ((i, j), (k, l))$ and $y_s = \operatorname{sgn}\left(d_{\mathcal{G}}(i_s, j_s) - d_{\mathcal{G}}(k_s, l_s)\right)$
- $((i_s, j_s), (k_s, l_s)) = ((k, l), (i, j))$ and $y_s = \operatorname{sgn}\left(d_{\mathcal{G}}(k_s, l_s) - d_{\mathcal{G}}(i_s, j_s)\right)$

GOTD are called *complete* if the condition above is satisfied for all pairs $((i, j), (k, l))$ of vertex pair such that $i = k$ and $d_{\mathcal{G}}(i, j) - d_{\mathcal{G}}(k, l) \neq 0$.

We are interested in the case where $\mathcal{G}$ is a tree, which corresponds to a typical hierarchical structure. We consider both the complete GOD case and complete GOTD case. Note that, as the complete GOTD are a subset of the complete GOD, to find embedding that is non-contradictory to the complete GOTD are easier than to find embedding that is non-contradictory to the complete GOD.

Hyperbolic Space Case

As shown in the following theorem, there is a non-contradictory embedding in $\mathcal{H}^D$ to complete GOD of a tree, even in $D = 2$.

Theorem 3.6.2. *For any tree $\mathcal{G}$ and GOD $\mathcal{S}$ of $\mathcal{G}$, there exists an embedding $(\boldsymbol{x}_n)_{n\in[N]}$ in $\mathcal{H}^2$ that is non-contradictory to $\mathcal{G}$.*

Corollary 3.6.3. *For any tree $\mathcal{G}$ and GOTD $\mathcal{S}$ of $\mathcal{G}$, there exists an embedding $(\boldsymbol{x}_n)_{n\in[N]}$ in $\mathcal{H}^2$ that is non-contradictory to $\mathcal{G}$.*

Theorem 3.6.2 is obtained from the result in (Sarkar, 2011), and it also gives a concrete construction of the embedding. The complete proof of Theorem 3.6.2 is given in Appendix A. Corollary 3.6.3 follows Theorem 3.6.2, because the complete GOTD are

included in the complete GOD.

Remark 3.6.4. As the D-dimensional hyperbolic space $\mathcal{H}^D$ ($D \geq 2$) includes two-dimensional hyperbolic space $\mathcal{H}^2$, the results in Theorem 3.6.2 and Corollary 3.6.3 can be applied to $\mathcal{H}^D$ ($D \geq 2$).

Euclidean Space Case Contrary to hyperbolic space, there is no non-contradictory embedding in $\mathbb{R}^D$ to complete GOTD of some trees. Before we show the results, we introduce some definitions.

Definition 3.6.5. Let $\mathcal{G} = ([N], \mathcal{E})$ be an undirected graph with vertex set $[N]$ and edge set $\mathcal{E}$. The degree $\deg(v)$ of $v \in [N]$ is defined by $\deg(v) \coloneqq |\{u \in [N] \mid (u, v) \in \mathcal{E}\}|$. We denote the maximum degree of any vertex in $\mathcal{G}$ by $\deg(\mathcal{G})$, which is defined by $\deg(\mathcal{G}) \coloneqq \max\{\deg(v) \mid v \in [N]\}$.

Definition 3.6.6. Let the D-dimensional sphere and the distance function on it be denoted by $\mathbb{S}^D$ and $d_{\mathbb{S}^D}$, respectively, which are given by

$$\mathbb{S}^D \coloneqq \left\{ \boldsymbol{x} \in \mathbb{R}^{(D+1)} \;\middle|\; \boldsymbol{x}^\top \boldsymbol{x} = 1 \right\}, \quad d_{\mathbb{S}^D}(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \arccos\left(\boldsymbol{x}^\top \boldsymbol{y}\right). \tag{3.25}$$

The $\frac{\pi}{3}$ packing number $M\left(\mathbb{S}^D, d_{\mathbb{S}^D}, \frac{\pi}{3}\right)$ of $\left(\mathbb{S}^D, d_{\mathbb{S}^D}\right)$ is the maximal number of points that can be $\frac{\pi}{3}$-separated, which is defined by

$$M\left(\mathbb{S}^D, d_{\mathbb{S}^D}, \frac{\pi}{3}\right) \coloneqq \max\left\{ N \in \mathbb{Z}_{\geq 0} \;\middle|\; \exists \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N \in \mathbb{S}^D, \forall i, j \in [N], d_{\mathbb{S}^D}(\boldsymbol{x}_i, \boldsymbol{x}_j) > \frac{\pi}{3} \right\}. \tag{3.26}$$

Note that the packing number $M\left(\mathbb{S}^D, d_{\mathbb{S}^D}, \frac{\pi}{3}\right)$ is finite for all $D \in \mathbb{Z}_{>0}$ and monotonous increasing function with respect to D, because for any $D, D' \in \mathbb{Z}_{>0}$ such that $D < D'$, $\mathbb{S}^D$ is a subspace of $\mathbb{S}^{D'}$. The following theorem clarifies the limitation of Euclidean space in ordinal embedding setting.

Theorem 3.6.7. *For any dimensionality D, for all graph $\mathcal{G}$ that is tree, if $\deg(\mathcal{G})$ is larger than $M\left(\mathbb{S}^{D-1}, d_{\mathbb{S}^{D-1}}, \frac{\pi}{3}\right)$, then no embedding $(\boldsymbol{x}_n)_{n \in [N]}$ in $\mathbb{R}^D$ is non-contradictory to the complete GOTD of $\mathcal{G}$.*

Corollary 3.6.8. *For any dimensionality D, for all graph $\mathcal{G}$ that is tree, if $\deg(\mathcal{G})$ is larger than $M\left(\mathbb{S}^{D-1}, d_{\mathbb{S}^{D-1}}, \frac{\pi}{3}\right)$, then no embedding $(\boldsymbol{x}_n)_{n \in [N]}$ in $\mathbb{R}^D$ is non-contradictory to the complete GOD of $\mathcal{G}$.*

The proof of Theorem 3.6.7 is given in Supplementary Materials. Corollary 3.6.8 follows Theorem 3.6.7, because the complete GOTD are included in the complete GOD.

Remark 3.6.9. Theorem 3.6.7 and Corollary 3.6.8 give a limitation of Euclidean space in embedding GOD of a tree. According to Theorem 3.6.2, 3.6.7, and Corollary 3.6.3, 3.6.8, two dimension is high enough in hyperbolic space for embedding of GOD of tree, but not all tree graphs can be embedded even in higher-dimensional Euclidean space. Hence, we can conclude that hyperbolic space is more suitable than Euclidean space for embedding of hierarchical ordinal data.

Remark 3.6.10 (Technical contribution of Theorem 3.6.7). Although the advantage of hyperbolic space against Euclidean space for embedding trees has been shown in graph embedding settings (e.g., (Sarkar, 2011)), the limitation of Euclidean space in embedding from ordinal triplet data given by Theorem 3.6.7 has not been clarified. Theorem 3.6.7 is not trivially derived from the graph embedding setting's results, because the requirements in the triplet ordinal data setting are weaker than those in the graph embedding setting. Specifically, all that is concerned in the triplet ordinal data setting is the distance

Table 3.1. Classification errors (mean ± standard error) in artificial datasets.

CBT-4-6	$D=2$	$D=4$	$D=8$	$D=16$
1-EMOE	0.4441 ± 0.0012	0.4313 ± 0.0012	0.3994 ± 0.0014	0.3890 ± 0.0014
2-EMOE	0.4397 ± 0.0011	0.4189 ± 0.0008	0.3986 ± 0.0008	0.3941 ± 0.0015
G-EPOE	0.4342 ± 0.0010	0.4295 ± 0.0012	0.4045 ± 0.0011	0.3831 ± 0.0010
t-EPOE	0.4424 ± 0.0010	0.4234 ± 0.0007	0.4109 ± 0.0008	0.4024 ± 0.0012
1-HMOE	0.4358 ± 0.0011	0.4138 ± 0.0009	0.4044 ± 0.0010	0.3875 ± 0.0008
2-HMOE	0.4426 ± 0.0009	0.4157 ± 0.0007	0.4085 ± 0.0012	0.3875 ± 0.0010
G-HPOE	0.4368 ± 0.0014	0.4179 ± 0.0015	0.4015 ± 0.0012	0.3899 ± 0.0007
t-HPOE	**0.4251** ± 0.0014	**0.3848** ± 0.0009	**0.3699** ± 0.0014	**0.3659** ± 0.0008
CBT-8-4	$D=2$	$D=4$	$D=8$	$D=16$
1-EMOE	0.4196 ± 0.0010	0.3901 ± 0.0010	0.3593 ± 0.0019	0.3406 ± 0.0017
2-EMOE	0.4219 ± 0.0012	0.3925 ± 0.0014	0.3650 ± 0.0013	0.3419 ± 0.0011
G-EPOE	0.4097 ± 0.0017	0.3928 ± 0.0011	0.3679 ± 0.0014	0.3365 ± 0.0014
t-EPOE	0.4252 ± 0.0014	0.3902 ± 0.0010	0.3753 ± 0.0010	0.3636 ± 0.0010
1-HMOE	0.4117 ± 0.0011	0.3779 ± 0.0008	0.3559 ± 0.0008	0.3375 ± 0.0007
2-HMOE	0.4095 ± 0.0007	0.3751 ± 0.0008	0.3500 ± 0.0013	0.3388 ± 0.0011
G-HPOE	0.4054 ± 0.0007	0.3857 ± 0.0010	0.3642 ± 0.0008	0.3362 ± 0.0012
t-HPOE	**0.3855** ± 0.0010	**0.3299** ± 0.0012	**0.3076** ± 0.0010	**0.3101** ± 0.0012
CBT-16-3	$D=2$	$D=4$	$D=8$	$D=16$
1-EMOE	0.4034 ± 0.0009	0.3595 ± 0.0014	0.3364 ± 0.0008	0.3075 ± 0.0013
2-EMOE	0.4089 ± 0.0014	0.3655 ± 0.0012	0.3374 ± 0.0008	0.3127 ± 0.0015
G-EPOE	0.3904 ± 0.0010	0.3450 ± 0.0011	0.3228 ± 0.0013	0.2865 ± 0.0009
t-EPOE	0.4023 ± 0.0011	0.3629 ± 0.0011	0.3326 ± 0.0012	0.3099 ± 0.0010
1-HMOE	0.3830 ± 0.0010	0.3427 ± 0.0011	0.3038 ± 0.0012	0.2823 ± 0.0010
2-HMOE	0.3892 ± 0.0013	0.3377 ± 0.0007	0.3100 ± 0.0013	0.2918 ± 0.0011
G-HPOE	0.3792 ± 0.0012	0.3478 ± 0.0011	0.3048 ± 0.0006	0.2863 ± 0.0012
t-HPOE	**0.3638** ± 0.0013	**0.2869** ± 0.0010	**0.2712** ± 0.0011	**0.2680** ± 0.0007

comparison in triplets, in which $i = k$, while the graph embedding setting cares uniform distortion, which corresponds to distance comparison in quadruplets, where $i \neq k$ is possible. Theorem 3.6.7 shows that Euclidean space cannot satisfy even the requirements of the ordinal triplet data setting, which are easier than the graph embedding setting.

3.7 Experiments

3.7.1 Experimental Settings

Methods To demonstrate the effectiveness of using hyperbolic space, we use the following Euclidean-space-based methods as baselines.

q-EMOE The loss function is given by ℓ_{mgn} in (3.9) with power index q, where $\mathbb{X} = \mathbb{R}^D$. In the experiments, we used $q = 1, 2$.

f-EPOE The loss function is given by ℓ_{prb} in (3.8) with kernel function f, where $\mathbb{X} = \mathbb{R}^D$. In this experiment, EPOE with the Gaussian kernel $f(d) = \exp\bigl(-d^2\bigr)$ and Student's t-distribution kernel $f(d) = \left(1 + \frac{d^2}{\alpha}\right)^{\alpha}$ are used, which we call G-EPOE (Gaussian EPOE) and t-EPOE (t-distributed EPOE).

Table 3.2. Classification errors (mean ± standard error) in real datasets.

WN-mammal	$D = 2$	$D = 4$	$D = 8$	$D = 16$
1-EMOE	0.1446 ± 0.0005	0.1120 ± 0.0004	0.0909 ± 0.0003	0.0747 ± 0.0004
2-EMOE	0.1396 ± 0.0004	0.1060 ± 0.0004	0.0842 ± 0.0004	0.0695 ± 0.0005
G-EPOE	0.1416 ± 0.0004	0.1281 ± 0.0008	0.1159 ± 0.0007	0.1058 ± 0.0004
t-EPOE	0.1598 ± 0.0007	0.1004 ± 0.0003	0.0738 ± 0.0005	0.0656 ± 0.0003
1-HMOE	0.1484 ± 0.0009	0.1022 ± 0.0009	0.0898 ± 0.0005	0.0798 ± 0.0003
2-HMOE	**0.1205** ± 0.0005	**0.0751** ± 0.0002	**0.0567** ± 0.0002	**0.0454** ± 0.0003
G-HPOE	0.1222 ± 0.0006	0.0992 ± 0.0005	0.1041 ± 0.0004	0.0909 ± 0.0005
t-HPOE	0.1438 ± 0.0010	0.1128 ± 0.0007	0.0915 ± 0.0004	0.0773 ± 0.0004
Cora	$D = 2$	$D = 4$	$D = 8$	$D = 16$
1-EMOE	0.3513 ± 0.0002	0.3258 ± 0.0002	0.3131 ± 0.0002	0.2973 ± 0.0003
2-EMOE	0.3584 ± 0.0003	0.3311 ± 0.0004	0.3091 ± 0.0002	0.2947 ± 0.0002
G-EPOE	0.3695 ± 0.0003	0.3525 ± 0.0005	0.3348 ± 0.0004	0.3103 ± 0.0003
t-EPOE	0.3629 ± 0.0003	0.3367 ± 0.0002	0.3156 ± 0.0002	0.3007 ± 0.0002
1-HMOE	0.3481 ± 0.0003	0.3245 ± 0.0002	0.3074 ± 0.0004	0.2923 ± 0.0002
2-HMOE	0.3528 ± 0.0003	0.3276 ± 0.0003	0.3051 ± 0.0002	0.2889 ± 0.0002
G-HPOE	0.3593 ± 0.0004	0.3347 ± 0.0002	0.3124 ± 0.0003	0.2967 ± 0.0002
t-HPOE	**0.3247** ± 0.0002	**0.2900** ± 0.0003	**0.2789** ± 0.0003	**0.2743** ± 0.0003

For the proposed hyperbolic methods, we use the following methods:

> q-**HMOE** The loss function is given by ℓ_{mgn} with power index q, where $\mathbb{X} = \mathbb{H}^D$. In the experiments, $q = 1, 2$.
> f-**HPOE** The loss function is given by ℓ_{prb} in (3.16) with kernel function f, where $\mathbb{X} = \mathbb{H}^D$. Similar to G-EPOE and t-EPOE, we use the same Gaussian kernel and Student's t-distrubution kernel for HPOE, and name them G-HPOE and t-HPOE, respectively.

Evaluation Protocol We conducted experiments on ordinal triplet data sets and ran each method 10 times to report their average classification errors along with standard errors. We created GOTD of ground-truth graph and randomly split the data set into training data, validation data, and test data. We trained each method on the training data, and selected a hyperparameter that gave the lowest classification error in grid-search on validation data as the best hyperparameter.

Optimization For optimization of all the methods, the stochastic Riemannian sub gradient method (Zhang and Sra, 2016) was applied. Note that this optimization method is reduced to the vanilla stochastic gradient descent method for the baselines, in which Euclidean space is used. The specific algorithm for our hyperbolic methods is given in Section 3.5.3. For all the methods, the constant learning rate was selected by grid-search.

Parameter Settings The batch size and the number of epoch in stochastic gradient descent were both fixed to 1000. In margin-loss-based methods, the margin hyperparameter δ was fixed to 1.0. The learning rate was selected from $\{0.1, 1.0, 10.0\}$ by grid-search. We report the results in $D = 2, 4, 8, 16$.

3.7.2 Experiments on Artificial Datasets

To validate the effectiveness of embedding in hyperbolic space, we constructed a typical hierarchical structure dataset, i.e., complete balanced tree (CBT).

Datasets **CBT** Denoting the m-nary complete balanced tree with the depth h by **CBT-m-h**, we use **CBT-4-6**, **CBT-8-4** and **CBT-16-3** for the experiments. Note that the number of the leaves of which are all 4096. We randomly selected 10000, 1000, and 1000 triplets for training, validation, and test, respectively in the experiments.

Results Table 3.1 shows that **t-HPOE** achieves the best result in all cases, which validate the effectiveness of using hyperbolic space. Moreover, both q-**HMOE** and f-**HPOE** performs better than the corresponding Euclidean methods in most cases. Taken $D = 2$ as an example, **t-HPOE** achieves the lowest errors with 0.4281 in **CBT-4-6**, as well as 0.3855 and 0.3638 in **CBT-8-4** and **CBT-16-3**, respectively. However, as the best performer among Euclidean methods, **G-EPOE** obtains the 0.4342, 0.4097, and 0.3904 only. This is because the expanding speed of hyperbolic space matches hierarchical structure of data, so that better embeddings can be achieved in very low dimensionality. Taking it a step further, we find that **t-HPOE** outperforms **G-EPOE** with a larger margin with lower dimensionality of space, such as 0.2865 when $D = 2$ and 0.0185 when $D = 16$ for **CBT-16-3** dataset. It is also interesting to note that superiority of **t-HPOE** decreases with increasing m. For example, **t-HPOE** achieves low errors than **G-EPOE** with 0.0061 in **CBT-4-6** and 0.0266 in **CBT-16-3** when $D = 2$. These phenomena are in line with theoretical analyses in Corollary 3.6.3 and Theorem 3.6.7.

3.7.3 Experiments on Real Datasets

We also compared the proposed methods to Euclidean-space-based methods on two real datasets that are of hierarchy.

Datasets **WN-mammal** (Nickel and Kiela, 2017) is a subset in WordNet[*1], which consists of more than 900 hyponyms of *mammal*. This dataset forms hierarchical structure, because a hypernym is often related to many hyponyms. **Cora**(Šubelj and Bajec, 2013) is a author citation dataset (McCallum et al., 2000) that contains more than 20000 computer science papers collected from web as vertices of graph. The references are parsed automatically and regarded as edges. Since reputable papers always are cited by many other papers, there should exist an underlying hierarchical structure.

The graph of each dataset is ground-truth and we derived triplets, i.e., GOTD (in Definition 3.6.1), from these graphs. Following (Liu et al., 2017), to avoid overfitting, we randomly selected 30000 triplets for training in WM-mammal, as well as 3000 triplets for validation and test each. Since Cora has a larger number of objects, we used more triplets, i.e., 100000, 10000, and 10000 for training, validation, and test, respectively.

Results The classification errors of hyperbolic methods against Euclidean baselines are given in Table 3.2 We can see that when $D = 2$, **2-HMOE** shows the lowest mean error 0.1205 in **WN-mammal** and **t-HPOE** shows the lowest error 0.3247 in **Cora**, whereas the best results of Euclidean methods are 0.1396 and 0.3513 only. This again demonstrates the effectiveness of the proposed hyperbolic methods for embedding hierarchical strurctural data in a low-dimensional space.

[*1] `https://wordnet.princeton.edu`

3.8 Conclusion

In this chapter, we have proposed a novel hyperbolic ordinal embedding (HOE) method to embed data that are of hierarchical structure in hyperbolic space. Due to the hierarchy-friendly property of hyperbolic space, HOE has effectively achieved embedding by capturing the hierarchy and preserving ordinal relations in an extremely low-dimensional space. By using a stochastic optimization method, HOE is also of high efficiency. Both theoretical and experimental results have demonstrated the outperformance of hyperbolic ordinal embedding over the Euclidean methods.

Chapter 4

Relational Data Embedding in Space Forms II: Multi-relational Graph Embedding

This part is omitted from the abridged version.

Chapter 5

Optimization in Space Forms

This part is omitted from the abridged version.

Chapter 6

Information Criterion in Space Forms

This part is omitted from the abridged version.

Chapter 7

Conclusion

7.1 Concluding Remarks

In this thesis, we have addressed the issues of machine learning over Riemannian manifolds. This thesis has contributed to the three fundamental problems: model, optimization, and evaluation.

In Chapter 3 and 4, we have proposed models— dissimilarity measure classes over Riemannian manifolds, as well as learning methods for them—for ordinal data and multi-relational graphs. In the case of ordinal data, we have focused on the distance between two points because ordinal data deals with only one kind of relation; whereas, in the case of multi-relational graphs, we also use the notion of direction to distinguish multiple relations between the data, exploiting the advantage of the Riemannian manifold using which we can define angle and direction. These methods have shown the advantage of using a Riemannian manifold for data structure and specific problem settings in which the method of using a non-Euclidean Riemannian manifold outperforms that of a Euclidean space.

In Chapter 5, we have contributed to optimizing machine learning based on Riemannian manifolds by theoretically comparing first-order stochastic optimization methods based on the traditional Euclidean metric, the natural gradient method, and the Riemannian gradient method. We have shown that even for a function with good conditions in regards to Riemannian geometry, the first-order stochastic methods based on the Euclidean metric or the natural gradient method can fail, and we can find the arbitrarily bad case for these two methods; whereas, the method based on Riemannian gradient method is more efficient. The result in this chapter motivates us to use Riemannian gradient descent based methods for optimization of functions, whose definition is derived by Riemannian geometry.

In Chapter 6, we have contributed to modelling the evaluation area for machine learning over Riemannian manifolds, providing a general calculation method of normalized maximum likelihood code lengths. Based on our novel Fourier-transform-based idea, we have obtained an explicit form of normalized maximum likelihood code length as the code length given by Bayesian predictive distribution with a complex prior. We, thereby, have given an explicit scheme to calculate the code length based on the form.

These results have systematically contributed to the fundamental elements of machine learning over Riemannian manifolds. In summary, this thesis has presented theoretical foundations and specific ways for applying machine learning over Riemannian manifolds.

7.2 Future Perspective

The models used in this thesis primarily focus on the data which do not have prior topology, such as discrete space, and we have focused on measuring its dissimilarity.

However, in reality, continuous spaces are used as inputs to many applications, such as the ones used for image, acoustic, and signal data processing. Therefore, obtaining appropriate dissimilarity measure on such spaces could be very important for future work.

In the optimization part, although we provide a counter-example for the performance guarantee of methods based on the Euclidean metric and the natural gradient method, generalizing this discussion to obtain a weaker condition that is sufficient for the failure of these methods could be considered as important future work to enable us to select the appropriate optimization methods.

In the evaluation area, although we have derived a generally applicable form for calculating the normalized maximum likelihood, the form includes a special function given by an complex integral. To use the form in machine learning applications, investigating the properties of these special functions is essential.

In all the cases, our results are one of the key foundations for future work.

Acknowledgement

I am deeply grateful to my supervisor, Prof. Kenji Yamanishi; the comments and advice of whom have been invaluable throughout the course of my study. He was also my supervisor for my Masters course, and he kindly invited me to his PhD course. His engineering philosophy on research has always been inspiring to me, and this thesis would not have been possible without it.

I would like to thank Prof. Feng Tian for his help on the research of Chapter 3 and Chapter 4. I would like to thank Prof. Takaaki Ohnishi and Prof. Taiji Suzuki, who have continuously provided me with valuable feedback. I would like to thank Dr. Shin Matsushima and Dr. Taichi Kiwaki for their help and encouragement throughout my Masters and PhD courses. I have learned a lot from them about conducting research, from problem setting to experimentation. I would like to thank Dr. Jing Wang not only for his contribution to the research of Chapter 3 and Chapter 4, but also for helping me during my PhD study. I would like to thank Mr. Yosuke Enokida for essential contributions to the research of Chapter 5. The discussion with the members of the sixth laboratory of Department of mathematical engineering, including Dr. Kohei Miyaguchi, Dr. Jing Wang, Dr. Atsushi Nitanda, Dr. Linchuan Xu, Dr. Hiroshi Kajino, Mr. Kenta Oono, and Mr. Yuichiro Suzuki was always exciting and a great guidance for my study. It was a great pleasure to be able to share time with such nice people.

Finally, I would like to thank all of my thesis committee members, Prof. Fumiyasu Komaki, Prof. Akiko Takeda, Prof. Taiji Suzuki, Dr. Kazuhiro Sato, and my supervisor.

Bibliography

Agarwal, S., Wills, J., Cayton, L., Lanckriet, G., Kriegman, D., and Belongie, S. (2007). Generalized non-metric multidimensional scaling. In *Artificial Intelligence and Statistics*, pages 11–18.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Alanis-Lobato, G., Mier, P., and Andrade-Navarro, M. A. (2016). Efficient embedding of complex networks to hyperbolic space via their laplacian. *Scientific reports*, 6:30108.

Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276.

Becigneul, G. and Ganea, O.-E. (2019). Riemannian adaptive optimization methods. In *International Conference on Learning Representations*.

Dawid, A. P. (1984). Present position and potential developments: Some personal views statistical theory the prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2):278–290.

Ganea, O.-E., Bécigneul, G., and Hofmann, T. (2018a). Hyperbolic entailment cones for learning hierarchical embeddings. In *International Conference on Machine Learning*.

Ganea, O.-E., Bécigneul, G., and Hofmann, T. (2018b). Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC.

Huffman, D. A. (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101.

Indow, T. (1967). Two interpretations of binocular visual space: Hyperbolic and euclidean. *Annals of the Japan Association for Philosophy of Science*, 3(2):51–64.

John, M., L. (2018). *Introduction to Riemannian manifolds, second edition*. Springer International Publishing.

Kraft, L. G. (1949). *A device for quantizing, grouping, and coding amplitude-modulated pulses*. PhD thesis, Massachusetts Institute of Technology.

Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., and Boguná, M. (2010). Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106.

Kruskal, J. B. (1964a). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

Kruskal, J. B. (1964b). Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129.

Lamping, J. and Rao, R. (1994). Laying out and visualizing large trees using a hyperbolic space. In *ACM symposium on User interface software and technology*, pages 13–14. ACM.

Lee, J. M. (2006). *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media.

Lehmann, E. L. and Casella, G. (2006). *Theory of point estimation*. Springer Science & Business Media.

Liu, H., Ji, R., Wu, Y., and Huang, F. (2017). Ordinal constrained binary code learning

for nearest neighbor search. In *AAAI Conference on Artificial Intelligence*.

Loring, W., T. (2011). *An introduction to manifolds, second edition*. Springer-Verlag New York.

Luneburg, R. K. (1947). Mathematical analysis of binocular vision.

McCallum, A. K., Nigam, K., Rennie, J., and Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163.

McFee, B. and Lanckriet, G. (2011). Learning multi-modal similarity. *Journal of machine learning research*, 12(Feb):491–523.

McMillan, B. (1956). Two inequalities implied by unique decipherability. *IRE Transactions on Information Theory*, 2(4):115–116.

Mitchell, T. M. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877.

Miyaguchi, K. (2018). *Learning high-dimensional models with the minimum description length principle*. PhD thesis, Graduate school of information science and technology, the University of Tokyo.

Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pages 6338–6347.

Nickel, M. and Kiela, D. (2018). Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465–471.

Ritter, H. (1999). Self-organizing maps on non-euclidean spaces. In *Kohonen maps*, pages 97–109. Elsevier.

Sala, F., De Sa, C., Gu, A., and Re, C. (2018). Representation tradeoffs for hyperbolic embeddings. In *International Conference on Machine Learning*, pages 4457–4466.

Sarkar, R. (2011). Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pages 355–366. Springer.

Savage, L. J. (1951). The theory of statistical decision. *Journal of the American Statistical association*, 46(253):55–67.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Shavitt, Y. and Tankel, T. (2008). Hyperbolic embedding of internet graph for distance estimation and overlay construction. *IEEE/ACM Transactions on Networking*, 16(1):25–36.

Shepard, R. N. (1962a). The analysis of proximities: multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140.

Shepard, R. N. (1962b). The analysis of proximities: Multidimensional scaling with an unknown distance function. ii. *Psychometrika*, 27(3):219–246.

Shepard, R. N. (1966). Metric structures in ordinal data. *Journal of Mathematical Psychology*, 3(2):287–315.

Šubelj, L. and Bajec, M. (2013). Model of complex networks based on citation dynamics. In *International conference on World Wide Web*, pages 527–530. ACM.

Tamuz, O., Liu, C., Belongie, S., Shamir, O., and Kalai, A. T. (2011). Adaptively learning the crowd kernel. In *International Conference on Machine Learning*, pages 673–680.

Terada, Y. and Luxburg, U. (2014). Local ordinal embedding. In *International Conference on Machine Learning*, pages 847–855.

Valiant, L. G. (1984). A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 436–445. ACM.

Van Der Maaten, L. and Weinberger, K. (2012). Stochastic triplet embedding. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE.

Wald, A. (1949). Statistical decision functions. *The Annals of Mathematical Statistics*, pages 165–205.

Walter, J. A. (2004). H-mds: a new approach for interactive visualization with multidimensional scaling in the hyperbolic space. *Information systems*, 29(4):273–292.

Wang, J., Tian, F., Liu, W., Wang, X., Zhang, W., and Yamanishi, K. (2018). Ranking preserving nonnegative matrix factorization. In *International Joint Conference on Artificial Intelligence*.

Zhang, H. and Sra, S. (2016). First-order methods for geodesically convex optimization. In *Conference on Learning Theory*, pages 1617–1638.

A

Appendix for Chapter 3

A.1 Proof of Theorem 3.6.2

We first introduce a Sarker's $(1+\epsilon)$ distortion embedding given by Algorithm 5 in (Sarkar, 2011), before we prove that the embedding is a concrete instance of Theorem 3.6.2. In the following, the geodesic path in $\mathcal{H}^2$ from $\boldsymbol{x} \in \mathbb{H}^2$ to $\boldsymbol{y} \in \mathbb{H}^2$ is denoted by $c(\boldsymbol{x}, \boldsymbol{y})$. Let $\mathcal{G} = ([N], \mathcal{E})$ be a tree. Let $\deg(v)$ denotes the degree of $v \in [N]$, defined by

$$\deg(v) \coloneqq |\{u \in [N] \mid (u, v) \in \mathcal{E}\}|. \tag{A.1}$$

Let the maximum degree of any vertex in $\mathcal{G}$ denoted by $\deg(\mathcal{G})$, which is defined by

$$\deg(\mathcal{G}) \coloneqq \max\{\deg(v) \mid v \in [N]\}. \tag{A.2}$$

We denote the graph distance of graph $\mathcal{G} = ([N], \mathcal{E})$ by $d_{\mathcal{G}} : [N] \times [N] \to \mathbb{Z}_{\geq 0}$. In the following, we introduce Sarker's $(1+\epsilon)$ distortion embedding for tree $\mathcal{G}$, with distortion parameter $\epsilon \in \mathbb{R}_{>0}$. By regarding object N as the root, we can regard $\mathcal{G}$ as a rooted tree. For $v \in [N-1]$, let the parent of v be denoted by $\mathrm{ch}(1; v)$ and let the $k-1$-th child of v be denoted by $\mathrm{ch}(k; v)$. For the root, let the k-th child of v be denoted by $\mathrm{ch}(k; v)$. Here $k \in [\deg(v)]$ if $v = N$, and $k \in [\deg(v) - 1]$ otherwise. In particular, $k \in [\deg(\mathcal{G})]$. Fix $\beta \in \left(0, \frac{\pi}{\deg(\mathcal{G})}\right)$. Let $\alpha = \frac{2\pi}{\deg(\mathcal{G})} - \beta$, $\nu = -2\ln\left(\tan\frac{\beta}{2}\right)$, and $\tau = \nu\frac{1+\epsilon}{\epsilon}$. For the root $v = N$, first, arbitrarily place $\boldsymbol{x}_N$ in $\mathbb{H}^2$, then $\boldsymbol{x}_{\mathrm{ch}(1;v)}$ so that $d_{\mathbb{H}^2}\left(\boldsymbol{x}_N, \boldsymbol{x}_{\mathrm{ch}(1;v)}\right) = \tau$. Then, recursively, for all objects $v \in [N]$ whose embedding has been already placed, we place the embeddings $\boldsymbol{x}_{\mathrm{ch}(k;N)}$ $(k = 2, 3, \ldots, \deg(N))$ of the children of v so that the following conditions are satisfied.

- $d_{\mathbb{H}^2}\left(\boldsymbol{x}_v, \boldsymbol{x}_{\mathrm{ch}(k;v)}\right) = \tau$.
- The angles $\left\{\angle \boldsymbol{x}_{\mathrm{ch}(k;v)}\boldsymbol{x}_v\boldsymbol{x}_{\mathrm{ch}(1;v)} \;\middle|\; k = 2, 3, \ldots, [\deg(v)]\right\}$ are mutually exclusively located in open intervals $\left\{\left(\frac{2\ell\pi}{\deg(\mathcal{G})} - \alpha, \frac{2\ell\pi}{\deg(\mathcal{G})} + \alpha\right) \;\middle|\; \ell \in [\deg(\mathcal{G}) - 1]\right\}$, where $\angle \boldsymbol{x}_{\mathrm{ch}(k;v)}\boldsymbol{x}_v\boldsymbol{x}_{\mathrm{ch}(1;v)}$ is the angle that $c\left(\boldsymbol{x}_v, \boldsymbol{x}_{\mathrm{ch}(k;v)}\right)$ makes with $c\left(\boldsymbol{x}_{\mathrm{ch}(1;v)}\right)$.

In the following, the embedding given by the above algorithm is called Sarker's $(1+\epsilon)$ distortion embedding. For any Sarker's $(1+\epsilon)$ distortion embedding, the following holds.

Theorem A.1.1 (Theorem 6 in (Sarkar, 2011))**.** *Let $\mathcal{G} = ([N], \mathcal{E})$ be a tree. For all $\epsilon \in \mathbb{R}_{>0}$ and all embeddings $(\boldsymbol{x}_n)_{n\in[N]}$ given by Sarker's $(1+\epsilon)$ distortion embedding, the following holds: for any object pair $(u, v) \in [N] \times [N]$, $\frac{1}{1+\epsilon}\tau d_{\mathcal{G}}(u, v) < d_{\mathbb{H}^2}(x_u, x_v) < \tau d_{\mathcal{G}}(u, v)$, where $\tau = \nu\frac{1+\epsilon}{\epsilon}$.*

The previous theorem directly proves Theorem 3.6.2.

Proof of Theorem 3.6.2. Let $\operatorname{diam}(\mathcal{G})$ denote the diameter of $\mathcal{G}$, defined by

$$\operatorname{diam}(\mathcal{G}) \coloneqq \max\{d_{\mathcal{G}}(u,v) \mid u,v \in [N]\}. \tag{A.3}$$

According to Theorem A.1.1 in (Sarkar, 2011), for any $\epsilon > 0$, there exists embedding $(\boldsymbol{x}_n)_{n\in[N]}$ and factor $\tau > 0$ such that for any object pair $(u,v) \in [N]\times[N]$, $\frac{1}{1+\epsilon}\tau d_{\mathcal{G}}(u,v) < d_{\mathbb{H}^2}(x_u,x_v) < \tau d_{\mathcal{G}}(u,v)$. By setting $\epsilon = \frac{1}{\operatorname{diam}(\mathcal{G})}$, we have embedding $(\boldsymbol{x}_n)_{n\in[N]}$ such that for any object pair $(u,v) \in [N]\times[N]$, $\tau[d_{\mathcal{G}}(u,v)-1] < d_{\mathbb{H}^2}(x_u,x_v) < \tau d_{\mathcal{G}}(u,v)$, which completes the proof. □

A.2 Proof of Theorem 3.6.7

Definition A.2.1. We say that $\mathcal{G} = ([N],\mathcal{E})$ includes a m-star if there exists a set of vertices $v_0, v_1, \ldots, v_m \in [N]$ such that for all $i = 1,2,\ldots,m$, $(v_0,v_i) \in \mathcal{E}$ and for all $i,j = 1,2,\ldots,m$ such that $i \neq j$, $(v_i,v_j) \notin \mathcal{E}$.

The following trivial proposition states the relation between Definition A.2.1 and tree.

Proposition A.2.2. *If a graph $\mathcal{G}$ is a tree and $\deg(\mathcal{G}) = m$, then $\mathcal{G}$ includes a m-star.*

Proof of Theorem 3.6.7. Assume that the embedding $(\boldsymbol{x}_n)_{n\in[N]}$ in $\mathbb{R}^D$ that is non-contradictory to the complete ordinal triplet data of $\mathcal{G}$. According to Proposition A.2.2, $\mathcal{G}$ includes a m-star. In this proof, the center of the sub m-star is relabeled $m+1$ and the vertices that has an edge to $m+1$ are relabeled $1,2,\ldots,m$. In the following, $\|\cdot\|_2$ denotes the 2-norm defined by $\|\boldsymbol{x}\|_2 \coloneqq \sqrt{\boldsymbol{x}^\top\boldsymbol{x}}$, and the closed ball with center $\boldsymbol{x} \in \mathbb{R}^D$ and radius $R \in \mathbb{R}_{\geq 0}$ is denoted by $B_R[\boldsymbol{x}]$, defined by $B_R[\boldsymbol{x}] \coloneqq \left\{\boldsymbol{x}' \in \mathbb{R}^D \;\middle|\; \|\boldsymbol{x}'-\boldsymbol{x}\|_2 \leq R\right\}$. Without loss of generality, we can set $\boldsymbol{x}_{m+1} = \boldsymbol{0}$. Let $R \coloneqq \min\{\|\boldsymbol{x}_n\|_2 \mid n \in [m]\}$. By the assumption of non-contradiction of embedding, for all $n,n' \in [m]$ such that $n \neq n'$, it holds that $\boldsymbol{x}_{n'} \notin B_{\|\boldsymbol{x}_n\|_2}[\boldsymbol{x}_n]$. Define $\tilde{\boldsymbol{x}}_n \coloneqq \frac{1}{\|\boldsymbol{x}_n\|_2}\boldsymbol{x}_n$. For fixed $n,n' \in [m]$ such that they satisfy $n \neq n'$ and $\|\boldsymbol{x}_n\|_2 \geq \|\boldsymbol{x}_{n'}\|_2$, define $\boldsymbol{x}'_n \coloneqq \frac{\|\boldsymbol{x}_{n'}\|_2}{\|\boldsymbol{x}_n\|_2}\boldsymbol{x}_n$. As $\boldsymbol{x}_{n'} \notin B_{\|\boldsymbol{x}_n\|_2}[\boldsymbol{x}_n]$ and $B_{\|\boldsymbol{x}'_n\|_2}[\boldsymbol{x}'_n] \subset B_{\|\boldsymbol{x}_n\|_2}[\boldsymbol{x}_n]$, it follows that $\boldsymbol{x}_{n'} \notin B_{\|\boldsymbol{x}'_n\|_2}[\boldsymbol{x}'_n]$. By multiplying factor $\frac{1}{\|\boldsymbol{x}'_n\|_2}$, we have $\tilde{\boldsymbol{x}}_{n'} \notin B_1[\tilde{\boldsymbol{x}}_n]$. Hence, it holds that $d_{\mathbb{R}^D}(\tilde{\boldsymbol{x}}_n,\tilde{\boldsymbol{x}}_{n'}) > 1$. If we regard $\tilde{\boldsymbol{x}}_1,\tilde{\boldsymbol{x}}_1,\ldots,\tilde{\boldsymbol{x}}_m$ as points in the $D-1$ dimensional unit sphere, for all $n,n' \in [m]$ such that $n \neq n'$, it holds that $d_{\mathbb{S}^{D-1}}(\tilde{\boldsymbol{x}}_n,\tilde{\boldsymbol{x}}_{n'}) > \frac{\pi}{3}$. Therefore, m cannot be larger than the $\frac{\pi}{3}$-packing number of $\mathbb{S}^{D-1}$. □

B

Appendix for Chapter 4

This part is omitted from the abridged version.