Representation Tradeoffs for Hyperbolic Embeddings

Frederic Sala [1] Christopher De Sa [2] Albert Gu [1] Christopher Ré [1]

Abstract

Hyperbolic embeddings offer excellent quality with few dimensions when embedding hierarchical data structures. We give a combinatorial construction that embeds trees into hyperbolic space with arbitrarily low distortion without optimization. On WordNet, this algorithm obtains a mean-average-precision of 0.989 with only two dimensions, outperforming existing work by 0.11 points. We provide bounds characterizing the precision-dimensionality tradeoff inherent in any hyperbolic embedding. To embed general metric spaces, we propose a hyperbolic generalization of multidimensional scaling (h-MDS). We show how to perform exact recovery of hyperbolic points from distances, provide a perturbation analysis, and give a recovery result that enables us to reduce dimensionality. Finally, we extract lessons from the algorithms and theory above to design a scalable PyTorch-based implementation that can handle incomplete information.

1. Introduction

Recently, hyperbolic embeddings have been proposed as a way to capture hierarchy information for network and natural language processing tasks (Nickel & Kiela, 2017; Chamberlain et al., 2017). This approach is an exciting way to fuse structural information (for example, from knowledge graphs or synonym hierarchies) with the continuous representations favored by modern machine learning methods.

To understand the intuition behind hyperbolic embeddings' superior capacity, note that trees can be embedded with arbitrarily low distortion into the Poincaré disk, a two-dimensional model of hyperbolic space (Sarkar, 2011). In contrast, Bourgain's theorem (Linial et al., 1995) shows that Euclidean space cannot achieve comparably low distortion for trees—even using an unbounded number of dimensions.

Many graphs, such as complex networks (Krioukov et al., 2010), the Internet (Krioukov et al., 2009), and social networks (Verbeek & Suri, 2016), are known to have tree-like or hyperbolic structure and thus befit hyperbolic embeddings. Indeed, recent works show that hyperbolic representations are suitable for many hierarchies (e.g. the question answering (Q/A) system HyperQA in Tay et al. (2018), vertex classifiers in Chamberlain et al. (2017), and link prediction (Nickel & Kiela, 2017)). However, the optimization problems underlying the embedding techniques in these works are challenging, motivating us to seek fundamental insights and to understand the subtle tradeoffs involved.

We begin by considering the case where we are given an input graph that is a tree or nearly tree-like, and our goal is to produce a low-dimensional hyperbolic embedding that preserves all distances. This leads to a simple combinatorial strategy that directly places points instead of minimizing a surrogate loss function. It is both fast (nearly linear time) and has formal quality guarantees. The approach proceeds in two phases: we (1) produce an embedding of a graph into a weighted tree, and (2) embed that tree into the hyperbolic disk. In particular, we consider an extension of an elegant embedding of trees into the Poincaré disk by Sarkar (2011) and work on low-distortion graph embeddings into tree metrics (Abraham et al., 2007). For trees, this approach has nearly perfect quality. On the WordNet hypernym graph reconstruction, it obtains a nearly perfect mean average precision (MAP) of 0.989 using just 2 dimensions. The best published numbers for WordNet in Nickel & Kiela (2017) range between 0.823 and 0.87 for 5 to 200 dimensions.

We analyze this construction to extract fundamental tradeoffs. One tradeoff involves the embedding dimension, the properties of the graph, and the number of bits of precision used to represent components of embedded points—an important hidden cost. We show that for a fixed precision, the dimension required scales linearly with the length of the longest path. On the other hand, the dimension scales logarithmically with the maximum degree of the tree. This suggests that hyperbolic embeddings should have high quality on hierarchies like WordNet but require large dimensions or high precision on graphs with long chains.

To understand how hyperbolic embeddings perform for met-

[1] Department of Computer Science, Stanford University [2] Department of Computer Science, Cornell University. Correspondence to: Frederic Sala <fredsala@stanford.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

rics that are far from tree-like, we consider a more general problem: given a matrix of distances that arise from points that are embeddable in hyperbolic space of dimension d (not necessarily from a graph), find a set of points that produces these distances. In Euclidean space, the problem is known as multidimensional scaling (MDS) and is solvable using PCA. A key step is a transformation that effectively centers the points, without knowledge of their exact coordinates. It is not obvious how to center points in hyperbolic space, which is curved. We show that in hyperbolic space, a centering operation is still possible with respect to a non-standard mean. In turn, this allows us to reduce the hyperbolic MDS problem (h-MDS) to a standard eigenvalue problem that can be solved with power methods. We also extend classical PCA perturbation analysis (Sibson, 1978; 1979). When applied to distances from graphs induced by real data, h-MDS obtains low distortion on far from tree-like graphs. However, we observe that these solutions may require high precision, which is not surprising in light of our previous analysis.

Finally, we handle increasing amounts of noise in the model, leading naturally into new SGD-based formulations. Like in traditional PCA, the underlying problem is nonconvex. In contrast to PCA, there are local minima that are not global minima—an additional challenge. Our main technical result is that an SGD-based algorithm initialized with an h-MDS solution can recover the submanifold the data is on—even in some cases in which the data is perturbed by noise that can be full dimensional. Our algorithm essentially provides new recovery results for convergence of Principal Geodesic Analysis (PGA) in hyperbolic space. We implemented the resulting SGD-based algorithm using PyTorch. Finally, we note that all of our algorithms can handle incomplete distance information through standard techniques.

2. Background

We provide intuition connecting hyperbolic space and tree distances, discuss the metrics used to measure embedding fidelity, and discuss the relationship between the reconstruction and learning problems for graph embeddings.

Hyperbolic spaces The Poincaré disk $\mathbb{H}_2$ is a two-dimensional model of hyperbolic geometry with points located in the interior of the unit disk, as shown in Figure 1. A natural generalization of $\mathbb{H}_2$ is the Poincaré ball $\mathbb{H}_r$, with elements inside the unit ball. The Poincaré models offer several useful properties, chief among which is mapping conformally to Euclidean space. That is, angles are preserved between hyperbolic and Euclidean space. Distances, on the other hand, are not preserved, but are given by

$$d_H(x, y) = \text{acosh}\left(1 + 2\frac{\|x - y\|^2}{(1 - \|x\|^2)(1 - \|y\|^2)}\right).$$

There are some potentially unexpected consequences of this formula, and a simple example gives intuition about a key technical property that allows hyperbolic space to embed trees. Consider three points inside the unit disk: the origin 0, and points x and y with $\|x\| = \|y\| = t$ for some $t > 0$. As shown on the right of Figure 1, as $t \to 1$ (i.e., the points move towards the outside of the disk), in flat Euclidean space, the ratio $\frac{d_E(x,y)}{d_E(x,0)+d_E(0,y)}$ is *constant* with respect to t (blue curve). In contrast, the ratio $\frac{d_H(x,y)}{d_H(x,0)+d_H(0,y)}$ approaches 1, or, equivalently, the distance $d_H(x, y)$ approaches $d_H(x, 0) + d_H(0, y)$ (red and pink curves). That is, the shortest path between x and y is almost the same as the path through the origin. This is analogous to the property of trees in which the shortest path between two sibling nodes is the path through their parent. This tree-like nature of hyperbolic space is the key property exploited by embeddings. Moreover, this property holds for arbitrarily small angles between x and y.

Lines and geodesics There are two types of geodesics (shortest paths) in the Poincaré disk model: segments of circles that are orthogonal to the disk surface, and disk diameters (Brannan et al., 2012). Our algorithms and proofs make use of a simple geometric fact: *isometric* reflection across geodesics (preserving hyperbolic distances) is represented in this Euclidean model as a *circle inversion*.

Embeddings and fidelity measures An *embedding* is a mapping $f : U \to V$ for spaces U, V with distances d_U, d_V. We measure the quality of embeddings with several *fidelity measures*, presented here from most local to most global.

Recent work (Nickel & Kiela, 2017) proposes using the *mean average precision* (MAP). For a graph $G = (V, E)$, let $a \in V$ have neighborhood $\mathcal{N}_a = \{b_1, b_2, \ldots, b_{\deg(a)}\}$, where $\deg(a)$ denotes the degree of a. In the embedding f, consider the points closest to $f(a)$, and define R_{a,b_i} to be the smallest set of such points that contains b_i (that is, R_{a,b_i} is the smallest set of nearest points required to retrieve the ith neighbor of a in f). Then, the MAP is defined to be

$$\text{MAP}(f) = \frac{1}{|V|}\sum_{a\in V}\frac{1}{\deg(a)}\sum_{i=1}^{|\mathcal{N}_a|}\frac{|\mathcal{N}_a \cap R_{a,b_i}|}{|R_{a,b_i}|}.$$

We have $\text{MAP}(f) \le 1$, with 1 as the best case. MAP is not concerned with explicit distances, but only *ranks* between the distances of immediate neighbors. It is a *local* metric.

The standard metric for graph embeddings is distortion D. For an n point embedding,

$$D(f) = \frac{1}{\binom{n}{2}}\left(\sum_{u,v\in U : u\neq v}\frac{|d_V(f(u), f(v)) - d_U(u, v)|}{d_U(u, v)}\right).$$

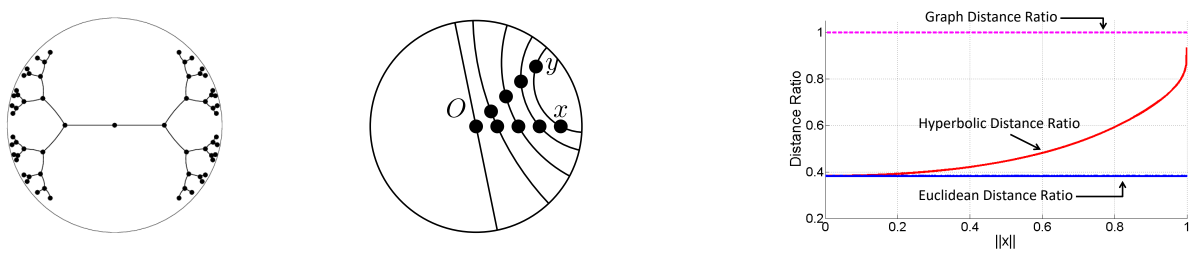

Figure 1. Left: Embedding of a binary tree in the Poincaré disk. Right: Geodesics and distances. As x and y move towards the outside of the disk (i.e., letting $\|x\|, \|y\| \to 1$), the distance $d_H(x, y)$ approaches $d_H(x, O) + d_H(O, y)$.

The best distortion is $D(f) = 0$, preserving the edge lengths exactly. This is a *global* metric, as it depends directly on the underlying distances rather than the local relationships between distances. A variant is the worst-case distortion D_{wc}, defined by

$$D_{\text{wc}}(f) = \frac{\max_{u,v \in U : u \neq v} d_V(f(u), f(v))/d_U(u, v)}{\min_{u,v \in U : u \neq v} d_V(f(u), f(v))/d_U(u, v)}.$$

That is, the wost-case distortion is the ratio of the maximal expansion and the minimal contraction of distances. Note that scaling the unit distance does not affect D_{wc}. The best worst-case distortion is $D_{\text{wc}}(f) = 1$.

Reconstruction and learning If we lack a full set of distances, we can either use the triangle inequality to recover the missing distances, or we can access the scaled Euclidean distances (the inside of the acosh in $d_H(x, y)$), and apply standard matrix completion techniques (Candes & Tao, 2010). Then we compute an embedding using any of the approaches discussed in this paper. We quantify the error introduced by this process experimentally in Section 5.

3. Combinatorial Constructions

We first focus on hyperbolic tree embeddings—a natural approach considering the tree-like behavior of hyperbolic space. We review the embedding of Sarkar (2011). We then provide novel analysis on the precision, revealing fundamental limits of hyperbolic embeddings. In particular, we characterize the bits of precision needed for hyperbolic representations. We extend the construction to r dimensions, and propose to use Steiner nodes to better embed general graphs as trees, building on Abraham et al. (2007).

Embedding trees The nature of hyperbolic space lends itself towards excellent tree embeddings. In fact, it is possible to embed trees into the Poincaré disk $\mathbb{H}_2$ with arbitrarily low distortion (Sarkar, 2011). Remarkably, trees cannot be embedded into Euclidean space with arbitrarily low distortion for *any* number of dimensions. These notions motivate the following two-step process for embedding hierarchies into hyperbolic space: (1) embed the graph $G = (V, E)$ into a tree T, and (2) embed T into the Poincaré ball $\mathbb{H}_d$. We refer to this process as the *combinatorial construction*. Note that we are not required to minimize a loss function. We begin by describing the second stage, where we extend an elegant construction from Sarkar (2011).

3.1. Sarkar's Construction

Algorithm 1 performs an embedding of trees into $\mathbb{H}_2$. The inputs are a scaling factor τ and a node a (of degree $\deg(a)$) from the tree with parent node b. Say a and b have already been embedded into $f(a)$ and $f(b)$ in $\mathbb{H}_2$. The algorithm places the children $c_1, c_2, \ldots, c_{\deg(a)-1}$ into $\mathbb{H}_2$.

A two-step process is used. First, $f(a)$ and $f(b)$ are reflected across a geodesic (using circle inversion) so that $f(a)$ is mapped onto the origin 0 and $f(b)$ is mapped onto some point z. Next, we place the children nodes to vectors $y_1, \ldots, y_{d-1}$ equally spaced around a circle with radius $\frac{e^\tau - 1}{e^\tau + 1}$ (which is a circle of radius τ in the hyperbolic metric), and maximally separated from the reflected parent node embedding z. Lastly, we reflect all of the points back across the geodesic. The isometric properties of reflections imply that all children are now at hyperbolic distance exactly τ from $f(a)$. To embed the entire tree, we place the root at the origin O and its children in a circle around it (as in Step 5 of Algorithm 1), then recursively place their children until all nodes have been placed. This process runs in linear time.

3.2. Analyzing Sarkar's Construction

Sarkar's construction works by separating children sufficiently in hyperbolic space. A key technical idea is to scale all the edges by a factor τ before embedding. We can then recover the original distances by dividing by τ. This transformation exploits the fact that hyperbolic space is not *scale invariant*. Sarkar's construction always captures neighbors perfectly, but Figure 1 implies that increasing the scale preserves the distances between farther nodes better. Indeed, if one sets $\tau = \frac{1+\varepsilon}{\varepsilon}\left(2 \log \frac{\deg_{\max}}{\pi/2}\right)$, then the worst-case distortion D of the resulting embedding is no more than

Algorithm 1 Sarkar's Construction

```
1: Input: Node $a$ with parent $b$, children to place $c_1, c_2, \dots, c_{\deg(a)-1}$, partial embedding $f$ containing an embedding for $a$ and $b$, scaling factor $\tau$
2: $(0, z) \leftarrow \text{reflect}_{f(a)\to 0}(f(a), f(b))$
3: $\theta \leftarrow \arg(z)$        {angle of $z$ from x-axis in the plane}
4: for $i \in \{1, \dots, \deg(a) - 1\}$ do
5:     $y_i \leftarrow \frac{e^\tau - 1}{e^\tau + 1} \cdot \left(\cos\left(\theta + \frac{2\pi i}{\deg(a)}\right), \sin\left(\theta + \frac{2\pi i}{\deg(a)}\right)\right)$
6: $(f(a), f(b), f(c_1), \dots, f(c_{\deg(a)-1})) \leftarrow \text{reflect}_{0\to f(a)}(0, z, y_1, \dots, y_{\deg(x)-1})$
7: Output: Embedded $\mathbb{H}_2$ vectors $f(c_1), f(c_2), \dots, f(c_{\deg(a)-1})$
```

$1 + \varepsilon$. For trees, Sarkar's construction has arbitrarily high fidelity. However, this comes at a cost: the scaling τ affects the bits of precision required. In fact, we will show that the precision scales logarithmically with the degree of the tree—but linearly with the maximum path length.

How many bits of precision do we need to represent points in $\mathbb{H}_2$? If $x \in \mathbb{H}_2$, then $\|x\| < 1$, so we need sufficiently many bits so that $1 - \|x\|$ will not be rounded to zero. This requires roughly $-\log(1 - \|x\|) = \log \frac{1}{1-\|x\|}$ bits. Say we are embedding two points x, y at distance d. As described in the background, there is an isometric reflection that takes a pair of points (x, y) in $\mathbb{H}_2$ to $(0, z)$ while preserving their distance, so without loss of generality we have that

$$d = d_H(x, y) = d_H(0, z) = \operatorname{acosh}\left(1 + 2\frac{\|z\|^2}{1 - \|z\|^2}\right).$$

Rearranging the terms, we have $(\cosh(d) + 1)/2 = (1 - \|z\|^2)^{-1} \geq (1 - \|z\|)^{-1}/2$. Thus, the number of bits we want so that $1 - \|z\|$ will not be rounded to zero is $\log(\cosh(d) + 1)$. Since $\cosh(d) = (\exp(d) + \exp(-d))/2$, this is roughly d bits. That is, in hyperbolic space, we need about d bits to express distances of d (rather than $\log d$ in Euclidean space).[1] This result will be of use below.

Consider the largest distance in the embeddings produced by Algorithm 1. If the longest path length in the tree is ℓ, and each edge has length $\tau = \frac{1}{\varepsilon}\left(2 \log \frac{\deg_{\max}}{\pi/2}\right)$, the largest distance is $O(\frac{\ell}{\varepsilon} \log \deg_{\max})$, and we require this number of bits for the representation.

Let us interpret this expression. Note that $\deg_{\max}$ is inside the $\log$ term, so that a bushy tree is not penalized much in precision. On the other hand, the longest path length ℓ is not, so that hyperbolic embeddings struggle with long paths. Moreover, by selecting an explicit graph, we derive a matching lower bound, concluding that to achieve a distortion ε, any construction requires $\Omega\left(\frac{\ell}{\varepsilon} \log(\deg_{\max})\right)$ bits. The argument follows from selecting a graph consisting of $m(\deg_{\max} + 1)$ nodes in a tree with a single root and $\deg_{\max}$ chains each of length m (shown in the appendix).

3.3. Improving the Construction

Our next contribution is a generalization of the construction from the disk $\mathbb{H}_2$ to the ball $\mathbb{H}_r$. Our construction follows the same line as Algorithm 1, but since we have r dimensions, the step where we place children spaced out on a circle around their parent now uses a hypersphere.

Spacing out points on the hypersphere is a classic problem known as *spherical coding* (Conway & Sloane, 1999). As we shall see, the number of children that we can place for a particular angle grows with the dimension. Since the required scaling factor τ gets larger as the angle decreases, we can reduce τ for a particular embedding by increasing the dimension. Note that increasing the dimension helps with bushy trees (large $\deg_{\max}$), but has limited effect on tall trees with small $\deg_{\max}$. We show

Proposition 3.1. *The generalized $\mathbb{H}_r$ combinatorial construction has distortion at most $1 + \varepsilon$ and requires at most $O(\frac{1}{\varepsilon}\frac{\ell}{r} \log \deg_{\max})$ bits to represent a node component for $r \leq (\log \deg_{\max}) + 1$, and $O(\frac{1}{\varepsilon}\ell)$ bits for $r > (\log \deg_{\max}) + 1$.*

To generalize to $\mathbb{H}_r$, we replace Step 5 in Algorithm 1 with a node placement step based on coding theory. The children are placed at the vertices of a hypercube inscribed into the unit hypersphere (and then scaled by τ). Each component of a hypercube vertex has the form $\frac{\pm 1}{\sqrt{r}}$. We index these points using binary sequences $a \in \{0, 1\}^r$ in the following way: $x_a = \left(\frac{(-1)^{a_1}}{\sqrt{r}}, \frac{(-1)^{a_2}}{\sqrt{r}}, \dots, \frac{(-1)^{a_r}}{\sqrt{r}}\right)$. We space out the children by controlling the distances by selecting a set of binary sequences a with a prescribed minimum Hamming distance—a binary error-correcting code—and placing the children at the resulting hypercube vertices. We provide more details, including our choice of code in the appendix.

3.4. Embedding into Trees

We revisit the first step of the construction: embedding graphs into trees. There are fundamental limits to how well graphs can be embedded into trees; in general, breaking long cycles inevitably adds distortion, as shown in Figure 2. We are inspired by a measure of this limit, the *δ-4 points condition* introduced in Abraham et al. (2007). A graph on n nodes that satisfies the δ-4 points condition has distortion at most $(1 + \delta)^{c_1 \log n}$ for some constant c_1. This result enables our end-to-end embedding to achieve a distortion of at most $D(f) \leq (1 + \delta)^{c_1 \log n}(1 + \varepsilon)$.

The result in Abraham et al. (2007) builds a tree with *Steiner*

[1] Although it is particularly easy to bound precision in the Poincaré model, this fact holds generally for hyperbolic space independent of model (shown in the appendix).

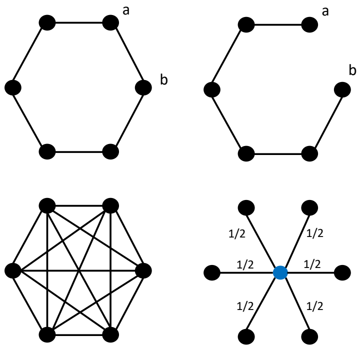

Figure 2. Top: Cycles are an issue in tree embeddings: $d_G(a,b)$ changes from 1 to 5. Bottom: Steiner nodes can help: adding a node and weighting edges maintains the pairwise distances.

nodes. These additional nodes can help control the distances in the resulting weighted tree (Figure 2). Note that Algorithm 1 readily extends to the case of weighted trees.

In summary, the key takeaways of our analysis are:

- There is a fundamental tension between precision and quality in hyperbolic embeddings.
- Hyperbolic embeddings have an exponential advantage in space compared to Euclidean embeddings for short, bushy hierarchies, but will have less of an advantage for graphs that contain long paths.
- Choosing an appropriate scaling factor τ is critical for quality. Later, we will propose to learn this scale factor automatically for computing embeddings in PyTorch.
- Steiner nodes can help improve embeddings of graphs.

4. Hyperbolic Multidimensional Scaling

In this section, we explore a fundamental and more general question than we did in the previous section: if we are given the pairwise distances arising from a set of points in hyperbolic space, can we recover the points? This enables us to produce an embedding for a desired distance metric. The equivalent problem for Euclidean distances is solved with multidimensional scaling (MDS). The goal of this section is to analyze the *hyperbolic MDS* (h-MDS) problem. We describe and overcome the additional technical challenges imposed by hyperbolic distances, and show that exact recovery is possible and interpretable. Afterwards we propose a technique for dimensionality reduction using principal geodesics analysis (PGA) that provides optimization guarantees. In particular, this addresses the shortcomings of h-MDS when recovering points that do not exactly lie on a hyperbolic manifold.

4.1. Exact Hyperbolic MDS

Suppose that there is a set of hyperbolic points $x_1, \ldots, x_n \in \mathbb{H}_r$, embedded in the Poincaré ball and written $X \in \mathbb{R}^{n \times r}$ in matrix form. We observe all the pairwise distances $d_{i,j} = d_H(x_i, x_j)$, but do not observe X: our goal is to use the observed $d_{i,j}$'s to recover X (or some other set of points with the same pairwise distances $d_{i,j}$).

The MDS algorithm in the Euclidean setting makes an important *centering*[2] assumption: the points have mean 0. If an exact embedding for the distances exists, it can be recovered from a matrix factorization. In other words, Euclidean MDS always recovers a centered embedding.

In hyperbolic space, the same algorithm does not work, but we show that it is possible to find an embedding centered at a different mean. More precisely, we introduce a new mean which we call the *pseudo-Euclidean mean*, that behaves like the Euclidean mean in that it enables recovery through matrix factorization. Once the points are recovered in hyperbolic space, they can be recentered around a more canonical mean by translating it to the origin.

Algorithm 2 is our complete algorithm, and for the remainder of this section we will describe how and why it works. We first describe the *hyperboloid model*, an alternate but equivalent model of hyperbolic geometry in which h-MDS is simpler. Of course, we can easily convert between the hyperboloid model and the Poincaré ball model. Next, we show how to reduce the problem to a standard PCA problem, which recovers an embedding centered at the points' pseudo-Euclidean mean. Finally, we discuss the meaning and implications of centering and prove that the algorithm preserves submanifolds as well—that is, if there is an exact embedding in $k < r$ dimensions centered at their canonical mean, then our algorithm will recover it.

The hyperboloid model Define Q to be the diagonal matrix in $\mathbb{R}^{r+1}$ where $Q_{00} = 1$ and $Q_{ii} = -1$ for $i > 0$. For a vector $x \in \mathbb{R}^{r+1}$, $x^T Q x$ is called the *Minkowski quadratic form.* The hyperboloid model is defined as

$$\mathbb{M}_r = \left\{ x \in \mathbb{R}^{r+1} \middle| x^T Q x = 1 \wedge x_0 > 0 \right\},$$

which is endowed with a distance measure $d_H(x,y) = \operatorname{acosh}(x^T Q y)$. For convenience, for $x \in \mathbb{M}_r$ let x_0 denote 0th coordinate $e_0^T x$, and $\vec{x} \in \mathbb{R}^r$ denote the rest of the coordinates[3]. With this notation, the Minkowski bilinear form can be written $x^T Q y = x_0 y_0 - \vec{x}^T \vec{y}$.

[2] We say that points are centered at a particular mean if this mean is at 0. The act of centering refers to applying an isometry that makes the mean of the points 0.

[3] Since $x_0 = \sqrt{1 + \|\vec{x}\|^2}$ is just a function of $\vec{x}$, we can equivalently consider just $\vec{x}$ as being a member of a model of hyperbolic space: This representation is sometimes known as the Gans model.

A new mean Given points $x_1, x_2, \dots, x_n \in \mathbb{M}_r$ in hyperbolic space, define a variance term

$$\Psi(z; x_1, x_2, \dots, x_n) = \sum_{i=1}^{n} \sinh^2(d_H(x_i, z)).$$

We define a *pseudo-Euclidean mean* to be any local minimum of this expression. Notice that this is independent of any particular model of hyperbolic space, since it is defined only through the hyperbolic distance function d_H.

Lemma 4.1. *Define $X \in \mathbb{R}^{n \times r}$ such that $X^T e_i = \vec{x}_i$ and $u \in \mathbb{R}^n$ such that $u_i = x_{0,i}$. Then*

$$\nabla_{\vec{z}} \Psi(z; x_1, x_2, \dots, x_n)|_{\vec{z}=0} = -2 \sum_{i=1}^{n} x_{0,i} \vec{x}_i = -2X^T u.$$

This means that 0 is a pseudo-Euclidean mean if and only if $0 = X^T u$. Call some hyperbolic points $x_1, \dots, x_n$ *pseudo-Euclidean centered* if their average is 0 in this sense: i.e. if $X^T u = 0$. We can always center a set of points without affecting their pairwise distances by simply finding their average, and then sending it to 0 through an isometry.

Recovery via matrix factorization Suppose we observe the pairwise distances $d_H(x_i, x_j)$ of points $x_1, x_2, \dots, x_n \in \mathbb{M}_r$. This gives the matrix Y such that

$$Y_{i,j} = \cosh\left(d_H(x_i, x_j)\right) = x_{0,i} x_{0,j} - \vec{x_i}^T \vec{x_j}. \quad (1)$$

Defining X and u as in Lemma 4.1, then in matrix form $Y = uu^T - XX^T$. Without loss of generality, suppose that the x_i are centered at their pseudo-Euclidean mean, so that $X^T u = 0$ by Lemma 4.1. This implies that u is an eigenvector of Y with positive eigenvalue, and the rest of Y's eigenvalues are negative. Therefore an eigendecomposition of Y will find $u, \hat{X}$ such that $Y = uu^T - \hat{X}\hat{X}^T$, i.e. it will directly recover X up to rotation.

In fact, running PCA on $-Y = X^T X - uu^T$ to find the n most significant non-negative eigenvectors will recover X up to rotation, and then u can be found by leveraging the fact that $x_0 = \sqrt{1 + \|\vec{x}\|^2}$. This leads to Algorithm 2, with optional post-processing steps for converting the embedding to the Poincaré ball model and for re-centering the points.

A word on centering The MDS algorithm in Euclidean geometry returns points centered at their *Karcher mean* z, which is a point minimizing $\sum d^2(z, x_i)$ (where d is the distance metric). The Karcher center is important for interpreting dimensionality reduction; we use the analogous hyperbolic Karcher mean for PGA in Section 4.2.

Algorithm 2

```
Input: Distance matrix $d_{i,j}$ and rank $r$
Compute scaled distance matrix $Y_{i,j} = \cosh(d_{i,j})$
$X \to$ PCA$(-Y, r)$
Project $X$ from hyperboloid model to Poincaré model: $x \to \frac{x}{1+\sqrt{1+\|x\|^2}}$
If desired, center $X$ at a different mean (e.g. the Karcher mean)
return $X$
```

Although Algorithm 2 returns points centered at their pseudo-Euclidean mean instead of their Karcher mean, they can be easily recentered by finding their Karcher mean and reflecting it onto the origin. Furthermore, Algorithm 2 preserves the dimension of the embedding:

Lemma 4.2. *If a set of points lie in a dimension-k geodesic submanifold, then both their Karcher mean and their pseudo-Euclidean mean lie in the same submanifold.*

This implies that centering with the pseudo-Euclidean mean preserves geodesic submanifolds: If it is possible to embed distances in a dimension-k geodesic submanifold centered and rooted at a Karcher mean, then it is also possible to embed the distances in a dimension-k submanifold centered and rooted at a pseudo-Euclidean mean, and vice versa.

4.2. Reducing Dimensionality with PGA

Given a high-rank embedding (resulting from h-MDS, for example), we may wish to find a lower-rank version. In Euclidean space, one can get the optimal lower rank embedding by simply discarding components. However, this may not be the case in hyperbolic space. Motivated by this, we study dimensionality reduction in hyperbolic space.

As hyperbolic space does not have a linear subspace structure like Euclidean space, we need to define what we mean by lower-dimensional. We follow Principal Geodesic Analysis (Fletcher et al., 2004), (Huckemann et al., 2010). Consider an initial embedding with points $x_1, \dots, x_n \in \mathbb{H}_2$ and let $d_H : \mathbb{H}_2 \times \mathbb{H}_2 \to \mathbb{R}_+$ be the hyperbolic distance. Suppose we want to map this embedding onto a one-dimensional subspace. (Note that we are considering a two-dimensional embedding and one-dimensional subspace here for simplicity, and these results immediately extend to higher dimensions.) In this case, the goal of PGA is to find a geodesic $\gamma : [0, 1] \to \mathbb{H}_2$ that passes through the mean of the points and that minimizes the squared error (or variance): $f(\gamma) = \sum_{i=1}^{n} \min_{t \in [0,1]} d_H(\gamma(t), x_i)^2$.

This expression can be simplified significantly and reduced to a minimization in Euclidean space. First, we find the mean of the points, the point $\bar{x}$ which minimizes $\sum_{i=1}^{n} d_H(\bar{x}, x_i)^2$.[4] Next, we reflect all the points x_i so that their mean is 0 in the Poincaré disk model; we can

[4] The derivative of the hyperbolic distance has a singularity, that is, $\lim_{y \to x} \partial_x |d_H(x, y)| \to \infty$ for any $x \in \mathbb{H}$. This issue can

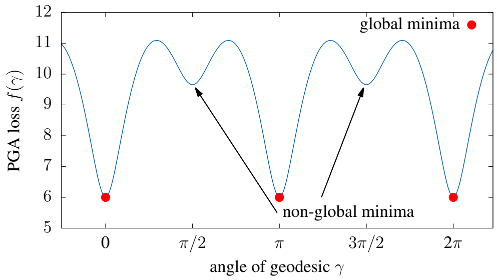

Figure 3. The PGA objective of an example task where the input dataset in the Poincaré disk is $x_1 = (0.8, 0)$, $x_2 = (-0.8, 0)$, $x_3 = (0, 0.7)$ and $x_4 = (0, -0.7)$. Note the presence of non-optimal local minima, unlike PCA.

do this using a circle inversion that maps $\bar{x}$ onto 0 Since reflections are isometric, if γ is a line through 0 and R_γ is the reflection across γ, we have that $d_H(\gamma, x) = \min_{t\in[0,1]} d_H(\gamma(t), x) = \frac{1}{2} d_H(R_l x, x)$.

Combining this with the Euclidean reflection formula and the hyperbolic metric produces

$$f(\gamma) = \frac{1}{4} \sum_{i=1}^{n} \operatorname{acosh}^2 \left(1 + \frac{8 d_E(\gamma, x_i)^2}{(1 - \|x_i\|^2)^2} \right),$$

in which d_E is the Euclidean distance from a point to a line. If we define $w_i = \sqrt{8} x_i / (1 - \|x_i\|^2)$ this reduces to the simplified expression $f(\gamma) = \frac{1}{4} \sum_{i=1}^{n} \operatorname{acosh}^2 \left(1 + d_E(\gamma, w_i)^2\right)$.

Notice that *the loss function is not convex*. We observe that there can be multiple local minima that are attractive and stable, in contrast to PCA. Figure 3 illustrates this nonconvexity on a simple dataset in $\mathbb{H}_2$ with only four examples. This makes globally optimizing the objective difficult.

Nevertheless, there will always be a region Ω containing a global optimum γ^* that is convex and admits an efficient projection, and where f is convex when restricted to Ω. Thus it is possible to build a gradient descent-based algorithm to recover lower-dimensional subspaces: for example, we built a simple optimizer in PyTorch. We also give a sufficient condition on the data for f above to be convex.

Lemma 4.3. *For hyperbolic PGA if for all i,*

$$\operatorname{acosh}^2 \left(1 + d_E(\gamma, w_i)^2\right) < \min \left(1, \frac{1}{3} \|w_i\|^2\right)$$

then f is locally convex at γ.

Table 1. Dataset statistics.

Dataset	Nodes	Edges	Comment
Bal. Tree	40	39	Tree
Phy. Tree	344	343	Tree
CS PhDs	1025	1043	Tree-like
WordNet	74374	75834	Tree-like
Diseases	516	1188	Dense
Gr-QC	4158	13428	Dense

Table 2. MAP measure for WordNet embedding compared to values in Nickel & Kiela (2017). Closer to 1 is better.

Dataset	C-$\mathbb{H}_2$	FB $\mathbb{H}_5$	FB $\mathbb{H}_{200}$
WordNet	0.989	0.823*	0.87*

As a result, if we initialize in and optimize over a region that contains γ^* and where the condition of Lemma 4.3 holds, then gradient descent will be guaranteed to converge to γ^*. We can turn this result around and read it as a recovery result: if the noise is bounded in this regime, then we are able to provably recover the correct low-dimensional embedding.

5. Experiments

We evaluate the proposed approaches and compare against existing methods. We hypothesize that for tree-like data, the combinatorial construction offers the best performance. For general data, we expect h-MDS to produce the lowest distortion, while it may have low MAP due to precision limitations. We anticipate that dimension is a critical factor (outside of the combinatorial construction). In the appendix, we report on additional datasets, combinatorial construction parameters, and the effect of hyperparameters.

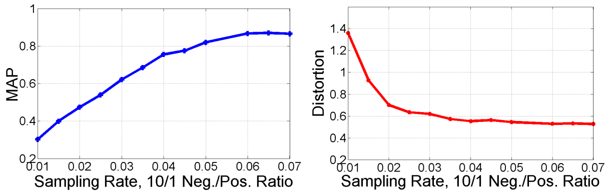

Figure 4. Learning from incomplete information. The distance matrix is sampled, completed, and embedded.

Datasets We consider trees, tree-like hierarchies, and graphs that are not tree-like. Trees include fully-balanced and phylogenetic trees expressing genetic heritage (Hofbauer et al., 2016), available at Sanderson et al. (1994). Nearly tree-like hierarchies include the WordNet hypernym graph (the largest connected component from Nickel & Kiela (2017)) and a graph of Ph.D. advisor-advisee relationships (De Nooy et al., 2011). Also included are datasets

be mitigated by minimizing d_H^2, which does have a continuous derivative throughout $\mathbb{H}$. The use of $d_H(x, y)$ is a minor instability in Nickel & Kiela (2017); Chamberlain et al. (2017)'s formulation, necessitating guarding against NaNs. We discuss this further in the appendix.

Table 3. Combinatorial and h-MDS techniques, compared against PCA and results from Nickel & Kiela (2017) (asterisks). Left (Distortion): Closer to 0 is better. Right (MAP): Closer to 1 is better.

Dataset	C-$\mathbb{H}_2$	FB $\mathbb{H}_2$	h-MDS	PT	PWS	PCA	FB	C-$\mathbb{H}_2$	FB $\mathbb{H}_2$	h-MDS	PT	PWS	PCA	FB
Bal. Tree	**0.013**	0.425	**0.077**	0.034	0.020	0.496	0.236	**1.0**	0.846	**1.0**	**1.0**	**1.0**	**1.0**	0.859
Phy. Tree	**0.006**	0.832	**0.039**	0.237	0.092	0.746	0.583	**1.0**	0.718	0.675	0.951	0.998	**1.0**	0.811
CS PhDs	**0.286**	0.542	**0.149**	0.298	0.187	0.708	0.336	**0.991**	0.567	0.463	0.799	**0.945**	0.541	0.78
Diseases	**0.147**	0.410	0.111	**0.080**	0.108	0.595	0.764	**0.822**	0.788	0.949	0.995	0.897	**0.999**	0.934
Gr-QC	**0.354**	-	0.530	**0.125**	0.134	0.546	-	0.696	-	0.710	0.733	0.504	0.738	**0.999***

Table 4. Precision and recall for WordNet entity-relationship-entity triple hyperbolic embeddings using combinatorial construction.

Relationship	Precision	Recall
'has instance'	99.97	99.98
'part of'	100.00	99.64
'domain region'	99.66	99.93

that vary in their tree nearness, such as disease relationships (Goh et al., 2007) and protein interactions (Jeong et al., 2001), both available from Rossi & Ahmed (2015). We also include the general relativity and quantum cosmology (Gr-QC) arXiv collaboration network (Leskovec et al., 2007).

Approaches Combinatorial embeddings into $\mathbb{H}_2$ use the $\varepsilon = 0.1$ precision setting; others are considered in the Appendix. We performed h-MDS in floating point precision. We include results for our PyTorch implementation (PT) of an SGD-based algorithm (described later), and a warm start version (PWS) initialized with the high-dimensional combinatorial construction. We compare against classical MDS (i.e., PCA), and the optimization-based approach Nickel & Kiela (2017), which we call FB. The experiments for h-MDS, PyTorch SGD, PCA, and FB used dimensions of 2,5,10,50,100,200; we recorded the best resulting MAP and distortion. Due to the large scale, we did not replicate the best FB numbers on large graphs (i.e., Gr-QC and WordNet); we report their best published MAP numbers (their work does not report distortion). These entries are marked with an asterisk. For the WordNet graph, FB uses the transitive closure; a weighted version of the graph captures the ancestor relationships. The full details are in appendix.

Quality In Table 3 (left), we report the distortion. As expected, for tree or tree-like graphs, the combinatorial construction has exceedingly low distortion. Because h-MDS is meant to recover points exactly, we hypothesized that h-MDS would offer very low distortion on these datasets. Table 3 confirms this: among h-MDS, PCA, and FB, h-MDS consistently offers the lowest distortion, producing, for example, a distortion of 0.039 on the phylogenetic tree. We observe that floating point h-MDS struggles with MAP. We separately confirmed that this is due to precision (by using a high-precision solver). The optimization-based approach is bolstered by appropriate initialization from the combinatorial construction.

Table 3 (right) reports the MAP measure (we additionally include WordNet results in Table 2), which is a local measure. We confirm that the combinatorial construction performs well for tree-like hierarchies, where MAP is close to 1. The construction improves on approaches such as FB that rely on optimization. On larger graphs like WordNet, our approach yields a MAP of 0.989—while their WordNet MAP result is 0.870 at 200 dimensions. This is exciting, as our approach is deterministic and linear-time.

A refined understanding of hyperbolic embeddings may be used to improve the quality and runtime of extant algorithms. Indeed, we embedded WordNet entity-relationship-entity triples (Socher et al., 2013) using the combinatorial construction in 10 dimensions, accurately preserving relationship knowledge (Table 4). This suggests that hyperbolic embeddings are effective at compressing knowledge and may useful for knowledge base completion and Q/A tasks.

SGD-Based Algorithm We built an SGD-based algorithm implemented in PyTorch. The loss function is equivalent to the PGA loss, and so is continuously differentiable.

To evaluate our algorithm's ability to deal with incomplete information, we sample the distance matrix at a ratio of non-edges to edges at $10 : 1$ following Nickel & Kiela (2017). In Figure 4, we recover a good solution for the phylogenetic tree with a small fraction of the entries; for example, we sampled approximately 4% of the graph for a MAP of 0.74 and distortion of 0.6. We also considered learning the scale of the embedding (details in the appendix). Finally, all of our techniques scale to graphs with millions of nodes.

6. Conclusion and Future Work

Hyperbolic embeddings embed hierarchical information with high fidelity and few dimensions. We explored the limits of this approach by describing scalable, high quality algorithms. We hope the techniques here encourage more follow-on work on the exciting techniques of Nickel & Kiela (2017); Chamberlain et al. (2017).

Acknowledgements

Thanks to Alex Ratner and Avner May for helpful discussion and to Beliz Gunel and Sen Wu for assistance with experiments. We gratefully acknowledge the support of DARPA under No. FA87501720095 and FA87501320039, ONR under No. N000141712266, the Moore Foundation, Okawa Research Grant, American Family Insurance, Accenture, Toshiba, the Secure Internet of Things Project, Google, VMware, Qualcomm, Ericsson, Analog Devices, and members of the Stanford DAWN project: Intel, Microsoft, Teradata, and VMware. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views, policies, or endorsements, either expressed or implied, of DARPA, DOE, NIH, ONR, or the U.S. Government.

References

Abraham, I., Balakrishnan, M., Kuhn, F., Malkhi, D., Ramasubramanian, V., and Talwar, K. Reconstructing approximate tree metrics. In *Proc. of the 26th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 43–52, Portland, Oregon, 2007.

Brannan, D., Esplen, M., and Gray, J. *Geometry*. Cambridge University Press, Cambridge, UK, 2012.

Candes, E. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Chamberlain, B. P., Clough, J. R., and Deisenroth, M. P. Neural embeddings of graphs in hyperbolic space. *arXiv preprint, arXiv:1705.10359*, 2017.

Conway, J. and Sloane, N. J. A. *Sphere Packings, Lattices and Groups*. Springer, New York, NY, 1999.

De Nooy, W., Mrvar, A., and Batagelj, V. *Exploratory social network analysis with Pajek*, volume 27. Cambridge University Press, 2011.

Fletcher, P., Lu, C., Pizer, S., and Joshi, S. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23(8): 995–1005, 2004.

Goh, K., Cusick, M., Valle, D., Childs, B., Vidal, M., and Barabási, A. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21), 2007.

Hofbauer, W., Forrest, L., Hollingsworth, P., and Hart, M. Preliminary insights from DNA barcoding into the diversity of mosses colonising modern building surfaces. *Bryophyte Diversity and Evolution*, 38(1):1–22, 2016.

Huckemann, S., Hotz, T., and Munk, A. Intrinsic shape analysis: Geodesic PCA for Riemannian manifolds modulo isometric Lie group actions. *Statistica Sinica*, 20(1): 1–58, 2010.

Jeong, H., Mason, S., Barabási, A., and Oltvai, Z. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.

Krioukov, D., Papadopoulos, F., and Boguná, A. V. M. Curvature and temperature of complex networks. *Physical Review E*, 80(035101), 2009.

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

Leskovec, J., Kleinberg, J., and Faloutsos, C. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1 (2), 2007.

Linial, N., London, E., and Rabinovich, Y. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

Nickel, M. and Kiela, D. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Long Beach, CA, 2017.

Rossi, R. A. and Ahmed, N. K. The network data repository with interactive graph analytics and visualization. In *Proc. of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. URL `http://networkrepository.com`.

Sanderson, M. J., Donoghue, M. J., Piel, W. H., and Eriksson, T. TreeBASE: a prototype database of phylogenetic analyses and an interactive tool for browsing the phylogeny of life. *American Journal of Botany*, 81(6), 1994.

Sarkar, R. Low distortion Delaunay embedding of trees in hyperbolic plane. In *Proc. of the International Symposium on Graph Drawing (GD 2011)*, pp. 355–366, Eindhoven, Netherlands, September 2011.

Sibson, R. Studies in the robustness of multidimensional scaling: Procrustes statistics. *Journal of the Royal Statistical Society, Series B*, 40(2):234–238, 1978.

Sibson, R. Studies in the robustness of multidimensional scaling: Perturbational analysis of classical scaling. *Journal of the Royal Statistical Society, Series B*, 41(2):217–229, 1979.

Socher, R., Chen, D., Manning, C. D., and Ng, A. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pp. 926–934, Lake Tahoe, NV, 2013.

Tay, Y., Tuan, L. A., and Hui, S. C. Hyperbolic representation learning for fast and efficient neural question answering. In *Proc. of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM 2018)*, pp. 583–591, Los Angeles, California, 2018.

Verbeek, K. and Suri, S. Metric embedding, hyperbolic space, and social networks. *Computational Geometry*, 59 Issue C:1–12, 2016.