Lorentzian Distance Learning for Hyperbolic Representations

Marc T. Law [1 2 3] **Renjie Liao** [1 2] **Jake Snell** [1 2] **Richard S. Zemel** [1 2]

Abstract

We introduce an approach to learn representations based on the Lorentzian distance in hyperbolic geometry. Hyperbolic geometry is especially suited to hierarchically-structured datasets, which are prevalent in the real world. Current hyperbolic representation learning methods compare examples with the Poincaré distance. They try to minimize the distance of each node in a hierarchy with its descendants while maximizing its distance with other nodes. This formulation produces node representations close to the centroid of their descendants. To obtain efficient and interpretable algorithms, we exploit the fact that the centroid w.r.t the squared Lorentzian distance can be written in closed-form. We show that the Euclidean norm of such a centroid decreases as the curvature of the hyperbolic space decreases. This property makes it appropriate to represent hierarchies where parent nodes minimize the distances to their descendants and have smaller Euclidean norm than their children. Our approach obtains state-of-the-art results in retrieval and classification tasks on different datasets.

1. Introduction

Generalizations of Euclidean space are important forms of data representation in machine learning. For instance, kernel methods (Shawe-Taylor et al., 2004) rely on Hilbert spaces that possess the structure of the inner product and are therefore used to compare examples. The properties of such spaces are well-known and closed-form relations are often exploited to obtain efficient, scalable, and interpretable training algorithms. While representing examples in a Euclidean space is appropriate to compare lengths and angles, non-Euclidean representations are useful when the task requires specific structure.

A common and natural non-Euclidean representation space is the spherical model (*e.g.* (Wang et al., 2017)) where the data lies on a unit hypersphere $\mathcal{S}^d = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 = 1\}$ and angles are compared with the cosine similarity function. Recently, some machine learning approaches (Nickel & Kiela, 2017; 2018; Ganea et al., 2018) have considered representing hierarchical datasets with the hyperbolic model. The motivation is that any finite tree can be mapped into a finite hyperbolic space while approximately preserving distances (Gromov, 1987), which is not the case for Euclidean space (Linial et al., 1995). Since hierarchies can be formulated as trees, hyperbolic spaces can be used to represent hierarchically structured data where the high-level nodes of the hierarchy are represented close to the origin whereas leaves are further away from the origin.

An important question is the form of hyperbolic geometry. Since their first formulation in the early nineteenth century by Lobachevsky and Bolyai, hyperbolic spaces have been used in many domains. In particular, they became popular in mathematics (*e.g.* space theory and differential geometry), and physics when Varicak (1908) discovered that special relativity theory (Einstein, 1905) had a natural interpretation in hyperbolic geometry. Various hyperbolic geometries and related distances have been studied since then. Among them are the Poincaré metric, the Lorentzian distance (Ratcliffe, 2006), and the gyrodistance (Ungar, 2010; 2014).

In the case of hierarchical datasets, machine learning approaches that learn hyperbolic representations designed to preserve the hierarchical similarity order have typically employed the Poincaré metric. Usually, the optimization problem is formulated so that the representation of a node in a hierarchy should be closer to the representation of its children and other descendants than to any other node in the hierarchy. Based on (Gromov, 1987), the Poincaré metric is a sensible dissimilarity function as it satisfies all the properties of a distance metric and is thus natural to interpret.

In this paper, we explain why the squared Lorentzian distance is a better choice than the Poincaré metric. One analytic argument relies on Jacobi Field (Lee, 2006) properties of Riemannian centers of mass (also called "*Karcher means*" although Karcher (2014) strongly discourages the use of that term). One other interesting property is that its centroid can be written in closed form.

[1] University of Toronto, Canada [2] Vector Institute, Canada [3] NVIDIA, work done while affiliated with the University of Toronto. Correspondence to: Marc Law <law@cs.toronto.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

Contributions: The main contributions of this paper are the study of the Lorentzian distance. We show that interpreting the squared Lorentzian distances with a set of points is equivalent to interpreting the distance with their centroid. We also study the dependence of the centroid with some hyperparameters, particularly the curvature of the manifold that has an impact on its Euclidean norm which is used as a proxy for depth in the hierarchy. This is the key motivation for our theoretical work characterizing its behavior *w.r.t.* the curvature. We relate the Lorentzian distance to other hyperbolic distances/geometries and explore its performance on retrieval and classification problems.

2. Background

In this section, we provide some technical background about hyperbolic geometry and introduce relevant notation. The interested reader may refer to (Ratcliffe, 2006).

2.1. Notation and definitions

To simplify the notation, we consider that vectors are row vectors and $\|\cdot\|$ is the ℓ_2-norm. In the following, we consider three important spaces.

Poincaré ball: The Poincaré ball $\mathcal{P}^d$ is defined as the set of d-dimensional vectors with Euclidean norm smaller than 1 (*i.e.* $\mathcal{P}^d = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| < 1\}$). Its associated distance is the Poincaré distance metric defined in Eq. (3).

Hyperboloid model: We consider some specific hyperboloid models $\mathcal{H}^{d,\beta} \subseteq \mathbb{R}^{d+1}$ defined as follows:

$$\mathcal{H}^{d,\beta} := \{\mathbf{a} = (a_0, \cdots, a_d) \in \mathbb{R}^{d+1} : \|\mathbf{a}\|^2_{\mathcal{L}} = -\beta, a_0 > 0\} \quad (1)$$

where $\beta > 0$ and $\|\mathbf{a}\|^2_{\mathcal{L}} = \langle \mathbf{a}, \mathbf{a} \rangle_{\mathcal{L}}$ is the *squared Lorentzian norm* of $\mathbf{a}$. The squared Lorentzian norm is derived from the *Lorentzian inner product* defined for all $\mathbf{a} = (a_0, \cdots, a_d) \in \mathcal{H}^{d,\beta}, \mathbf{b} = (b_0, \cdots, b_d) \in \mathcal{H}^{d,\beta}$ as:

$$\langle \mathbf{a}, \mathbf{b} \rangle_{\mathcal{L}} := -a_0 b_0 + \sum_{i=1}^{d} a_i b_i \leq -\beta \quad (2)$$

It is worth noting that $\langle \mathbf{a}, \mathbf{b} \rangle_{\mathcal{L}} = -\beta$ iff $\mathbf{a} = \mathbf{b}$. Otherwise, $\langle \mathbf{a}, \mathbf{b} \rangle_{\mathcal{L}} < -\beta$ for all pairs $(\mathbf{a}, \mathbf{b}) \in (\mathcal{H}^{d,\beta})^2$. Vectors in $\mathcal{H}^{d,\beta}$ are a subset of positive time-like vectors[1]. The hyperboloid $\mathcal{H}^{d,\beta}$ has constant negative curvature $-1/\beta$. Moreover, every vector $\mathbf{a} \in \mathcal{H}^{d,\beta}$ satisfies $a_0 = \sqrt{\beta + \sum_{i=1}^d a_i^2}$. We note $\mathcal{H}^d := \mathcal{H}^{d,1}$ the space obtained when $\beta = 1$; it is called the *unit hyperboloid* model and is the main hyperboloid model considered in the literature.

Model space: Finally, we note $\mathcal{F}^d \subseteq \mathbb{R}^d$ the output vector space of our model (*e.g.* the output representation of some neural network). We consider that $\mathcal{F}^d = \mathbb{R}^d$.

2.2. Optimizing the Poincaré distance metric

Most methods that compare hyperbolic representations (Nickel & Kiela, 2017; 2018; Ganea et al., 2018; Gulcehre et al., 2019) consider the Poincaré distance metric defined for all $\mathbf{c} \in \mathcal{P}^d, \mathbf{d} \in \mathcal{P}^d$ as:

$$\mathsf{d}_{\mathcal{P}}(\mathbf{c}, \mathbf{d}) = \cosh^{-1}\left(1 + 2\frac{\|\mathbf{c} - \mathbf{d}\|^2}{(1 - \|\mathbf{c}\|^2)(1 - \|\mathbf{d}\|^2)}\right) \quad (3)$$

which satisfies all the properties of a distance metric and is therefore natural to interpret. Direct optimization in $\mathcal{P}^d$ of problems using the distance formulation in Eq. (3) is numerically unstable for two main reasons (see for instance (Nickel & Kiela, 2018) or (Ganea et al., 2018, Section 4)). First, the denominator depends on the norm of examples, so optimizing over $\mathbf{c}$ and $\mathbf{d}$ when either of their norms is close to 1 leads to numerical instability. Second, elements have to be re-projected onto the Poincaré ball at each iteration with a fixed maximum norm. Moreover, Eq. (3) is not differentiable when $\mathbf{c} = \mathbf{d}$ (see proof in appendix).

For better numerical stability of their solver, Nickel & Kiela (2018) propose to use an equivalent formulation of $\mathsf{d}_{\mathcal{P}}$ in the unit hyperboloid model. They use the fact that there exists an invertible mapping $h : \mathcal{H}^{d,\beta} \to \mathcal{P}^d$ defined for all $\mathbf{a} = (a_0, \cdots, a_d) \in \mathcal{H}^{d,\beta}$ as:

$$h(\mathbf{a}) := \frac{1}{1 + \sqrt{1 + \sum_{i=1}^d a_i^2}}(a_1, \cdots, a_d) \in \mathcal{P}^d \quad (4)$$

When $\beta = 1, \mathbf{a} \in \mathcal{H}^d, \mathbf{b} \in \mathcal{H}^d$, we have the following equivalence:

$$\mathsf{d}_{\mathcal{H}}(\mathbf{a}, \mathbf{b}) = \mathsf{d}_{\mathcal{P}}(h(\mathbf{a}), h(\mathbf{b})) = \cosh^{-1}(-\langle \mathbf{a}, \mathbf{b} \rangle_{\mathcal{L}}) \quad (5)$$

Nickel & Kiela (2018) show that optimizing the formulation in Eq. (5) in $\mathcal{H}^d$ is more stable numerically.

Duality between spherical and hyperbolic geometries: One can observe from Eq. (5) that preserving the order of Poincaré distances is equivalent to preserving the reverse order of Lorentzian inner products (defined in Eq. (2)) since the $\cosh^{-1}$ function is monotonically increasing on its domain $[1, +\infty)$. The relationship between the Poincaré metric and the Lorentzian inner product is actually similar to the relationship between the geodesic distance $\cos^{-1}(\langle \mathbf{p}, \mathbf{q} \rangle)$ and the cosine $\langle \mathbf{p}, \mathbf{q} \rangle$ (or the squared Euclidean distance $\|\mathbf{p} - \mathbf{q}\|^2 = 2 - 2\langle \mathbf{p}, \mathbf{q} \rangle$) when $\mathbf{p}$ and $\mathbf{q}$ are on a unit hypersphere $\mathcal{S}^d$ because of the duality between these geometries (Ratcliffe, 2006). The hyperboloid $\mathcal{H}^{d,\beta}$ can be seen as a half hypersphere of imaginary radius $i\sqrt{\beta}$. In the same way as kernel methods that consider inner products in Hilbert spaces as similarity measures, we consider in this paper the Lorentzian inner product and its induced distance.

[1] A vector $\mathbf{a}$ that satisfies $\langle \mathbf{a}, \mathbf{a} \rangle_{\mathcal{L}} < 0$ is called time-like and it is called positive iff $a_0 > 0$.

3. Lorentzian distance learning

We present the (squared) *Lorentzian distance function* $\mathrm{d}_{\mathcal{L}}$ which has been studied in differential geometry but not used to learn hyperbolic representations to the best of our knowledge. Nonetheless, it was used in contexts where representations are not hyperbolic (Liu et al., 2010; Sun et al., 2015) (*i.e.* not constrained to belong to some hyperboloid). We give the formulation of the Lorentzian centroid when representations are hyperbolic and show that its Euclidean norm, which is used as a proxy for depth in the hierarchy, depends on the curvature $-1/\beta$. We show that studying (squared) Lorentzian distances with a set of points is equivalent to studying the distance with their centroid. Moreover, we exploit results in (Karcher, 1987) to explain why the Lorentzian distance is a better choice than the Poincaré distance. Finally, we discuss optimization details.

3.1. Lorentzian distance and mappings

The squared Lorentzian distance (Ratcliffe, 2006) is defined for all pair $\mathbf{a} \in \mathcal{H}^{d,\beta}, \mathbf{b} \in \mathcal{H}^{d,\beta}$ as:

$$\mathrm{d}^2_{\mathcal{L}}(\mathbf{a},\mathbf{b}) = \|\mathbf{a}-\mathbf{b}\|^2_{\mathcal{L}} = -2\beta - 2\langle \mathbf{a},\mathbf{b}\rangle_{\mathcal{L}} \tag{6}$$

It satisfies all the axioms of a distance metric except the triangle inequality.

Mapping: Current hyperbolic machine learning models exploit $\mathrm{d}_{\mathcal{H}}$. They re-project at each iteration their learned representations onto the Poincaré ball (Nickel & Kiela, 2017) or use the exponential map of the unit hyperboloid model $\mathcal{H}^d$ (Nickel & Kiela, 2018; Gulcehre et al., 2019) to directly optimize on it. Since our approach does not necessarily consider that $\beta = 1$, we consider the invertible mapping $g_\beta : \mathcal{F}^d \to \mathcal{H}^{d,\beta}$, called a *local parametrization* of $\mathcal{H}^{d,\beta}$, defined for all $\mathbf{f} = (f_1, \cdots, f_d) \in \mathcal{F}^d$ as:

$$g_\beta(\mathbf{f}) := (\sqrt{\|\mathbf{f}\|^2 + \beta}, f_1, \cdots, f_d) \in \mathcal{H}^{d,\beta} \tag{7}$$

As mentioned in Section 2.1, $\mathcal{F}^d$ is the output space of our model, *i.e.*, $\mathbb{R}^d$ in practice. The pair $(\mathcal{H}^{d,\beta}, g_\beta^{-1})$ where $g_\beta^{-1} : \mathcal{H}^{d,\beta} \to \mathcal{F}^d$ is called a *chart* of the manifold $\mathcal{H}^{d,\beta}$.

We then compare two examples $\mathbf{f}_1 \in \mathcal{F}^d$ and $\mathbf{f}_2 \in \mathcal{F}^d$ with $\mathrm{d}^2_{\mathcal{L}}$ by calculating:

$$\mathrm{d}^2_{\mathcal{L}}(g_\beta(\mathbf{f}_1), g_\beta(\mathbf{f}_2)) = -2\beta - 2\langle g_\beta(\mathbf{f}_1), g_\beta(\mathbf{f}_2)\rangle_{\mathcal{L}} \tag{8}$$

$$= -2\left[\beta + \langle \mathbf{f}_1, \mathbf{f}_2\rangle - \sqrt{\|\mathbf{f}_1\|^2+\beta}\sqrt{\|\mathbf{f}_2\|^2+\beta}\right] \tag{9}$$

Preserved order of Euclidean norms: Although examples are compared with $\mathrm{d}^2_{\mathcal{L}}$ in the hyperbolic space $\mathcal{H}^{d,\beta}$ where all the points have the same Lorentzian norm, it is worth noting that the order of the Euclidean norms of examples is preserved along the three spaces $\mathcal{F}^d$, $\mathcal{H}^{d,\beta}$ and $\mathcal{P}^d$ with the mappings g_β and h. The preservation with g_β is straightforward: $\forall \mathbf{f}_1, \mathbf{f}_2 \in \mathcal{F}^d, \|\mathbf{f}_1\| < \|\mathbf{f}_2\| \iff \sqrt{2\|\mathbf{f}_1\|^2+\beta} = \|g_\beta(\mathbf{f}_1)\| < \|g_\beta(\mathbf{f}_2)\|$. The proof of the following theorem is given in the appendix:

Theorem 3.1 (Order of Euclidean norms)**.** *The following order is preserved for all $\mathbf{a} \in \mathcal{F}^d, \mathbf{b} \in \mathcal{F}^d$ with $h \circ g_\beta$ where h and g_β are defined in Eq. (4) and Eq. (7), respectively:*

$$\|\mathbf{a}\| < \|\mathbf{b}\| \iff \|h(g_\beta(\mathbf{a}))\| < \|h(g_\beta(\mathbf{b}))\| \tag{10}$$

In conclusion, the Euclidean norms of examples can be compared equivalently in any space. This is particularly useful if we want to study the Euclidean norm of centroids.

3.2. Centroid properties

The center of mass (here called centroid) is a concept in statistics motivated in (Fréchet, 1948) to estimate some statistical dispersion of a set of points (*e.g.* the variance). It is the minimizer of an expectation of (squared) distances with a set of points and was extended to Riemannian manifolds in (Grove & Karcher, 1973). We now study the centroid *w.r.t.* the squared Lorentzian distance. Ideally, we would like the centroid of node representations to be (close to) the representation of their lowest common ancestor.

Lemma 3.2 (Center of mass of the Lorentzian inner product)**.** *The point $\boldsymbol{\mu} \in \mathcal{H}^{d,\beta}$ that maximizes the following problem $\max_{\boldsymbol{\mu}\in\mathcal{H}^{d,\beta}} \sum_{i=1}^n \nu_i \langle \mathbf{x}_i, \boldsymbol{\mu}\rangle_{\mathcal{L}}$ where $\forall i, \mathbf{x}_i \in \mathcal{H}^{d,\beta}$, $\forall i, \nu_i \geq 0, \sum_i \nu_i > 0$ is unique and formulated as:*

$$\boldsymbol{\mu} = \sqrt{\beta}\frac{\sum_{i=1}^n \nu_i \mathbf{x}_i}{|\|\sum_{i=1}^n \nu_i \mathbf{x}_i\|_{\mathcal{L}}|} \tag{11}$$

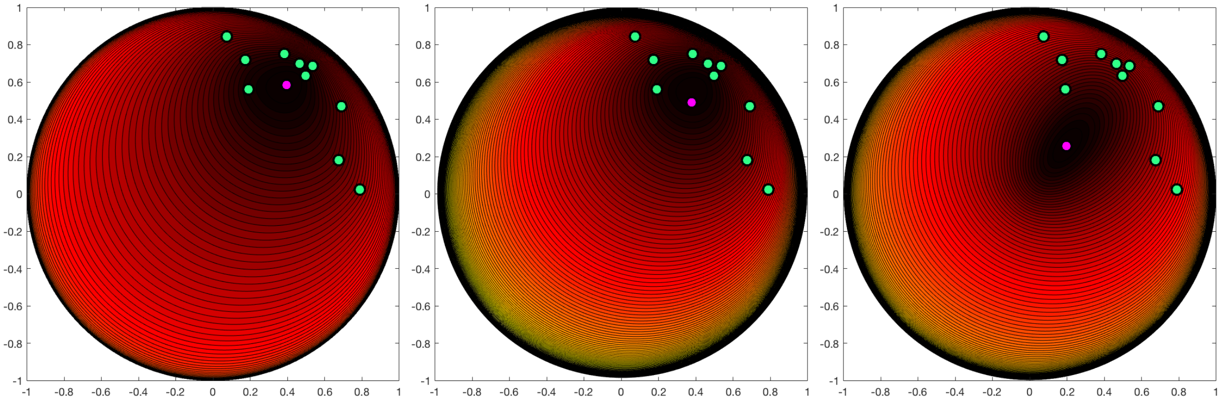
where $|\|\mathbf{a}\|_{\mathcal{L}}| = \sqrt{|\|\mathbf{a}\|^2_{\mathcal{L}}|}$ is the modulus of the imaginary Lorentzian norm of the positive time-like vector $\mathbf{a}$. The proof is given in the appendix. The centroid formulation in Eq. (11) generalizes the centroid formulations given in (Galperin, 1993; Ratcliffe, 2006) for any number of points and any value of the constant curvature $-1/\beta$.

Theorem 3.3 (Centroid of the squared Lorentzian distance)**.** *The point $\boldsymbol{\mu} \in \mathcal{H}^{d,\beta}$ that minimizes the problem $\min_{\boldsymbol{\mu}\in\mathcal{H}^{d,\beta}} \sum_{i=1}^n \nu_i \mathrm{d}^2_{\mathcal{L}}(\mathbf{x}_i, \boldsymbol{\mu})$ where $\forall i, \mathbf{x}_i \in \mathcal{H}^{d,\beta}$, $\nu_i \geq 0, \sum_i \nu_i > 0$ is given in Eq. (11)*

The proof exploits the formulation given in Eq. (6). The formulation of the centroid $\boldsymbol{\mu}$ can be used to perform hard clustering where a uniform measure over the data in a cluster is assumed (*i.e.* $\forall i, \nu_i = \frac{1}{n}$ or equivalently in this context $\forall i, \nu_i = 1$). One can see that the centroid of one example is the example itself. We now study its Euclidean norm.

Theorem 3.4. *The Euclidean norm of the centroid of different points in $\mathcal{P}^d$ decreases as $\beta > 0$ decreases.*

The proof is given in the appendix. Fig. 1 illustrates the 2-dimensional Poincaré ball representation of the centroid



(a) Poincaré distance (b) $\beta = 1$, Squared Lorentzian dist. (c) $\beta = 0.1$, Squared Lorentzian dist.

Figure 1. $n = 10$ examples are represented in green in a Poincaré ball. Their centroid *w.r.t.* $\mathsf{d}_{\mathcal{H}}$ and $\mathsf{d}^2_{\mathcal{L}}$ for different values of $\beta \in \{1, 10^{-1}\}$ (and $\forall i \in \{1, \cdots, n\}, \nu_i = \frac{1}{n}$) is in magenta and the level sets represent the sum of the distances between the current point and the 10 examples. Smaller values of β induce smaller Euclidean norms of the centroid.

of a set of 10 different points *w.r.t.* the Poincaré distance and the squared Lorentzian distance for different values of $\beta > 0$. One can see that the centroid *w.r.t.* the Poincaré metric does not have smaller norm. On the other hand, the Euclidean norm of the Lorentzian centroid does decrease as β decreases, it can then be enforced to be smaller by choosing lower values of $\beta > 0$. Centroids *w.r.t.* other hyperbolic distances are illustrated in the appendix.

We provide in the following some side remarks that are useful to understand the behavior of the Lorentzian distance.

Theorem 3.5 (Nearest point)**.** *Let $\mathcal{B} \subseteq \mathcal{H}^{d,\beta}$ be a subset of $\mathcal{H}^{d,\beta}$, and let $\boldsymbol{\mu} \in \mathcal{H}^{d,\beta}$ be the centroid of the set $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathcal{H}^{d,\beta}$ w.r.t. the squared Lorentzian distance (see Theorem 3.3). We have the following relation:*

$$\operatorname*{arg\,min}_{\mathbf{b} \in \mathcal{B}} \sum_{i=1}^{n} \nu_i \mathsf{d}^2_{\mathcal{L}}(\mathbf{x}_i, \mathbf{b}) = \operatorname*{arg\,min}_{\mathbf{b} \in \mathcal{B}} \mathsf{d}^2_{\mathcal{L}}(\boldsymbol{\mu}, \mathbf{b}) \qquad (12)$$

The theorem shows that distances with a set of points can be compared with only one point which is the centroid. We will show the interest of this theorem in the last experiment in Section 4.3.

Moreover, from the Cauchy-Schwarz inequality, as β tends to 0, the Lorentzian distance in Eq. (9) to $\mathbf{f}_1$ tends to 0 for any vector $\mathbf{f}_2$ that can be written $\mathbf{f}_2 = \tau \mathbf{f}_1$ with $\tau \geq 0$. The distance is greater otherwise. Therefore, the Lorentzian distance with a set of points tends to be smaller along the ray that contains elements that can be written as $\tau \boldsymbol{\mu}$ where $\tau \geq 0$ and $\boldsymbol{\mu}$ is their centroid (see illustrations in the appendix).

Curvature adaption: From Jacobi field properties, when the chosen metric is the Poincaré metric $\mathsf{d}_{\mathcal{H}}$ on the unit hyperboloid $\mathcal{H}^d$, the Hessian of $\mathsf{d}_{\mathcal{H}}$ has the eigenvalue 0 along the radial direction. However, the eigenvalues of the Hessian are principal curvatures of the level surface. Since the vector field is more important and to get a "better curvature adaption" (Karcher, 2014) (*i.e.* nonzero eigenvalues), Karcher (1987) recommends the "modified distance" $(-1 + \cosh(\mathsf{d}_{\mathcal{H}}))$ which is equal to $\frac{1}{2}\mathsf{d}^2_{\mathcal{L}}$ on $\mathcal{H}^d$.

Hyperbolic centroids in the literature: To the best of our knowledge, two other works exploit hyperbolic centroids. Sala et al. (2018) use a centroid related to the Poincaré metric but do not have a closed-form solution for it so they compute it via gradient descent. Gulcehre et al. (2019) optimize a problem based on the Poincaré metric but exploit the gyro-centroid (Ungar, 2014) which belongs to another type of hyperbolic geometry. When the set contains only 2 points, it is the minimizer of the Einstein addition of the gyrodistances between it and the two points of the set by using the gyrotriangle inequality. Otherwise, it can be seen as a point which preserves left gyrotranslation. One limitation of the gyrocentroid is that it cannot be seen as a minimizer of an expectation of distances (unlike Fréchet means) since the Einstein addition is not commutative.

3.3. Optimization and solver

Hyperbolic approaches that optimize the Poincaré distance on the hyperboloid (Nickel & Kiela, 2018) as defined in Eq. (5) exploit Riemannian stochastic gradient descent. This choice of optimizer is motivated by the fact that the domain of Eq. (5) cannot be considered to be a vector space (*e.g.* $\mathbb{R}^{d+1} \times \mathbb{R}^{d+1}$). Indeed, $\mathsf{d}_{\mathcal{H}}(\mathbf{a}, \mathbf{b})$ is not defined for pairs $\mathbf{a} \in \mathbb{R}^{d+1}$, $\mathbf{b} \in \mathbb{R}^{d+1}$ that satisfy $\langle \mathbf{a}, \mathbf{b} \rangle_{\mathcal{L}} > -1$ due to the definition of $\cosh^{-1}$. Therefore, the directional derivative of $\mathsf{d}_{\mathcal{H}}$ lacks a suitable vector space structure, and standard optimizers cannot be used.

On the other hand, the squared Lorentzian distance relies only on the formulation of the Lorentzian inner product in Eq. (2) which is well-defined and smooth for any pair

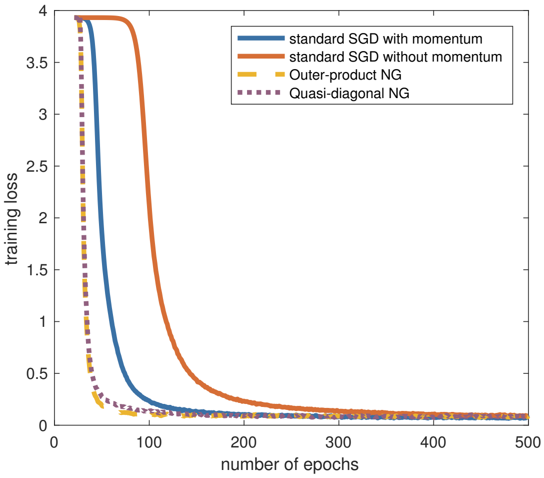


Figure 2. Training loss obtained by different solvers as a function of the number of epochs for the ACM dataset.

$\mathbf{a} \in \mathbb{R}^{d+1}$, $\mathbf{b} \in \mathbb{R}^{d+1}$. We formulate our training loss functions as taking representations in $\mathcal{F}^d = \mathbb{R}^d$ that are mapped to $\mathcal{H}^{d,\beta}$ via $g_\beta : \mathcal{F}^d \to \mathcal{H}^{d,\beta}$ and compared with the Lorentzian inner product for which a directional derivative in vector space exists. Therefore, methods of real analysis apply (see Chapter 3.1.1 of (Absil et al., 2009)), and the chain rule is used to derive standard optimization techniques such as SGD. In the case of the unit hyperboloid model, the geodesic distance $\mathsf{d}_\mathcal{H}$ (*i.e.* Poincaré) and the vector space distance $\mathsf{d}_\mathcal{L}$ (*i.e.* Lorentzian) are increasing functions of each other due to the monotonicity of $\cosh^{-1}$. Therefore, by trying to decrease the Lorentzian distance in vector space, the standard SGD also decreases the geodesic distance.

It is worth mentioning that due to the formulation of g_β, the parameter space is not Euclidean and has a Riemannian structure. The direction in parameter space which gives the largest change in the objective per unit of change is parameterized by the inverse of the Fisher information matrix and called *natural gradient* (NG) (Amari, 1998). We have tried different optimizers such as the standard SGD with and without momentum, and Riemannian metric methods that approximate the NG (Ollivier, 2015). We experimentally observed that the standard SGD with momentum provides a good tradeoff in terms on running time and retrieval evaluation metrics. Although its convergence is slightly worse than NG approximation methods, it is much simpler to use since it does not require storing gradient matrices and returns better retrieval performance. Momentum methods are known to account for the curvature directions of the parameter space. The convergence comparison of the SGD and NG approximations for the ACM dataset is illustrated in Fig. 2. A detailed comparison of standard SGD and NG optimizers is provided in the appendix. In the following, we only use the standard SGD with momentum.

4. Experiments

We evaluate the Lorentzian distance in three different tasks. The first two experiments consider the same evaluation protocol as the baselines and therefore do not exploit the formulation of the center of mass which does not exist in closed form for the Poincaré metric. The first experiment considers similarity constraints based on subsumption in hierarchies. The second experiment performs binary classification to determine whether or not a test node belongs to a specific subtree of the hierarchy. The final experiment extracts a category hierarchy from a dataset where the taxonomy is partially provided.

4.1. Representation of hierarchies

The goal of the first experiment is to learn the hyperbolic representation of a hierarchy. To this end, we consider the same evaluation protocol as (Nickel & Kiela, 2018) and the same datasets. Each dataset can be seen as a directed acyclic graph (DAG) where a parent node links to its children and descendants in the tree to create a set of edges $\mathcal{D}$. More exactly, Nickel & Kiela (2017) consider subsumption relations in a hierarchy (*e.g.* WordNet nouns). They consider the set $\mathcal{D} = \{(u, v)\}$ such that each node u is an ancestor of v in the tree. For each node u, a set of negative nodes $\mathcal{N}(u)$ is created: each example in $\mathcal{N}(u)$ is not a descendant of u in the tree. Their problem is then formulated as learning the representations $\{\mathbf{f}_i\}_{i=1}^n$ that minimize:

$$L_\mathcal{D} = -\sum_{(u,v)\in\mathcal{D}} \log \frac{e^{-\mathsf{d}(\mathbf{f}_u,\mathbf{f}_v)}}{\sum_{v'\in\mathcal{N}(u)\cup\{v\}} e^{-\mathsf{d}(\mathbf{f}_u,\mathbf{f}_{v'})}} \tag{13}$$

where d is the chosen distance function. Eq. (13) tries to get similar examples closer to each other than dissimilar ones. Nickel & Kiela (2017) and Nickel & Kiela (2018) optimize the Poincaré metric with different solvers. We replace $\mathsf{d}(\mathbf{f}_u, \mathbf{f}_v)$ by $\mathsf{d}^2_\mathcal{L}(g_\beta(\mathbf{f}_u), g_\beta(\mathbf{f}_v))$ as defined in Eq. (9).

As in (Nickel & Kiela, 2018), the quality of the learned embeddings is measured with the following metrics: the mean rank (MR), the mean average precision (MAP) and the Spearman rank-order correlation ρ (SROC) between the Euclidean norms of the learned embeddings and the normalized ranks of nodes. Following (Nickel & Kiela, 2018), the normalized rank of a node u is formulated:

$$\text{rank}(u) = \frac{\text{sp}(u)}{\text{sp}(u) + \text{lp}(u)} \in [0, 1] \tag{14}$$

where $\text{lp}(u)$ is the longest path between u and one of its descendant leaves; $\text{sp}(u)$ is the shortest path from the root of the hierarchy to u. A leaf in the hierarchy tree then has a normalized rank equal to 1, and nodes closer to the root have smaller normalized rank. ρ is measured as the SROC between the list of Euclidean norms of embeddings and their normalized rank.

Table 1. Evaluation of Taxonomy embeddings. MR: Mean Rank (lower is better). MAP: Mean Average Precision (higher is better). ρ: Spearman's rank-order correlation (higher is better).

Method		$\mathsf{d}_{\mathcal{P}}$ in $\mathcal{P}^d$ optimizer proposed in (Nickel & Kiela, 2017)	$\mathsf{d}_{\mathcal{P}}$ in $\mathcal{H}^d$ optimizer proposed in (Nickel & Kiela, 2018)	Ours $\beta = 0.01$ $\lambda = 0$	Ours $\beta = 0.1$ $\lambda = 0$	Ours $\beta = 1$ $\lambda = 0$	Ours $\beta = 0.01$ $\lambda = 0.01$
WordNet Nouns	MR	4.02	2.95	**1.46**	1.59	1.72	1.47
	MAP	86.5	92.8	94.0	93.5	91.5	**94.7**
	ρ	58.5	59.5	40.2	45.2	43.1	**71.1**
WordNet Verbs	MR	1.35	1.23	**1.11**	1.14	1.23	1.13
	MAP	91.2	93.5	**94.6**	93.7	91.9	94.0
	ρ	55.1	56.6	36.8	38.7	37.2	**73.0**
EuroVoc	MR	1.23	1.17	**1.06**	**1.06**	1.09	**1.06**
	MAP	94.4	**96.5**	**96.5**	96.0	95.0	96.1
	ρ	61.4	**67.5**	41.8	44.2	45.6	61.7
ACM	MR	1.71	1.63	**1.03**	1.06	1.16	1.04
	MAP	94.8	97.0	**98.8**	96.9	94.1	98.1
	ρ	62.9	65.9	53.9	55.9	46.7	**66.4**
MeSH	MR	12.8	12.4	1.31	**1.30**	1.40	1.33
	MAP	79.4	79.9	90.1	**90.5**	85.5	90.3
	ρ	74.9	76.3	46.1	47.2	41.5	**78.7**

Our optimization problems learns representations $\{\mathbf{f}_i \in \mathcal{F}^d\}_{i=1}^n$ that minimize the following problem:

$$L_{\mathcal{D}} + \lambda \sum_{\{(u,v):\text{rank}(u)<\text{rank}(v)\}} \max(\|\mathbf{f}_u\|^2 - \|\mathbf{f}_v\|^2, 0) \quad (15)$$

where $\lambda \geq 0$ is a regularization parameter. The second term tries to satisfy the order of the embedding norms to match the normalized ranks. Using Theorem 3.1, we optimize Euclidean norms in $\mathcal{F}^d$. This problem tries to minimize the distances between similar examples to preserve orders of distances so that ancestors are closer to their descendants than to unrelated nodes. This indirectly enforces an example similar to a set of examples to be close to their centroids.

Datasets: We consider the following datasets: (1) *2012 ACM Computing Classification System:* is the classification system for the computing field used by ACM journal. (2) *EuroVoc:* is a thesaurus maintained by the European Union. (3) *Medical Subject Headings (MeSH):* (Rogers, 1963) is a medical thesaurus provided by the U.S. National Library of Medicine. (4) *Wordnet:* (Miller, 1998) is a large lexical database. As in (Nickel & Kiela, 2018), we consider the noun and verb hierarchy of WordNet. More details about these datasets can be found in (Nickel & Kiela, 2018).

Implementation details: Following (Nickel & Kiela, 2017)[2], we implemented our method in Pytorch 0.3.1. We use the standard SGD optimizer with a learning rate of 0.1 and momentum of 0.9. For the largest datasets *Wordnet Nouns* and *MeSH*, we stop training after 1500 epochs. We stop training at 3000 epochs for the other datasets. The mini-batch size is 50, and the number of sampled negatives per example is 50. The weights of the embeddings are initialized from the continuous uniform distribution in the interval $[-10^{-4}, 10^{-4}]$. The dimensionality of our embeddings is 10. To sample $\{(u, v) : \text{rank}(u) < \text{rank}(v)\}$ in a mini-batch, we randomly sample η examples from the set of positive and negative examples that are sampled for $L_{\mathcal{D}}$, we then select 5% of the possible ordered pairs. $\eta = 150$ for all the datasets, except for WordNet nouns where $\eta = 50$ and MeSH where $\eta = 100$ due to their large size.

Results: We compare in Table 1 the Poincaré distance metric as optimized in (Nickel & Kiela, 2017; 2018) with our method for different values of β and λ (indicated in the table). Here we separately analyze the case where $\lambda = 0$, which corresponds to using the same constraints as those reported in (Nickel & Kiela, 2018), and where $\lambda > 0$.

- Case where $\lambda = 0$: Our approach obtains better Mean Rank and Mean Average Precision scores than (Nickel & Kiela, 2018) for small values of β when we use the same constraints. The fact that the retrieval performance of our approach changes with different values of $\beta \in \{0.01, 0.1, 1\}$ shows that the curvature of the space has an impact on the distances between examples and on the behavior of the model. As explained in Section 3.2, the squared Lorentzian distance tends to behave more radially (*i.e.* the distance tends to decrease along the ray that can be written $\tau\boldsymbol{\mu}$ where $\tau \geq 0$ and $\boldsymbol{\mu}$ is the centroid) as $\beta > 0$ decreases. Children then tend to have larger Euclidean norm than their parents while being close *w.r.t.* the Lorentzian distance. We eval-

[2] We use the source code available at `https://github.com/facebookresearch/poincare-embeddings`

Table 2. Test F1 classification scores for four different subtrees of WordNet noun tree.

Dataset	animal.n.01	group.n.01	worker.n.01	mammal.n.01
(Ganea et al., 2018)	$99.26 \pm 0.59\%$	$91.91 \pm 3.07\%$	$66.83 \pm 11.83\%$	$91.37 \pm 6.09\%$
Euclidean dist	$99.36 \pm 0.18\%$	$91.38 \pm 1.19\%$	$47.29 \pm 3.93\%$	$77.76 \pm 5.08\%$
$\log_{\mathbf{0}}$ + Eucl	$98.27 \pm 0.70\%$	$91.41 \pm 0.18\%$	$36.66 \pm 2.74\%$	$56.11 \pm 2.21\%$
Ours ($\beta = \lambda = 0.01$)	$99.57 \pm 0.24\%$	$99.75 \pm 0.11\%$	$94.50 \pm 1.21\%$	$96.65 \pm 1.18\%$
Ours ($\beta = 0.01$, $\lambda = 0$)	$99.77 \pm 0.17\%$	$99.86 \pm 0.03\%$	$96.32 \pm 1.05\%$	$97.73 \pm 0.86\%$

uate in the appendix the percentage of nodes that have a Euclidean norm greater than their parent in the tree. More than 90% of pairs (parent,child) satisfy the desired order of Euclidean norms. The percentage increases with smaller values of β, this illustrates our point on the impact on the Euclidean norm of the center of mass.

On the other hand, our approach obtains worse performance for the ρ metric which evaluates how the order of the Euclidean norms is correlated with their normalized rank/depth in the hierarchy tree. This result is expected due to the formulation of the constraints of $L_{\mathcal{D}}$ that considers only local similarity orders between pairs of examples. The loss $L_{\mathcal{D}}$ does not take into account the global structure of the tree; it only considers whether pairs of concepts subsume each other or not. The worse ρ performance of the Lorentzian distance $\mathsf{d}_{\mathcal{L}}$ could be due to the fact that $\mathsf{d}_{\mathcal{L}}$ does not take into account global structure of the hierarchy tree but does tend to preserve local structure (as shown in Table 3).

- Case where $\lambda > 0$: As a consequence of the worse ρ performance, we evaluate the performance of our model when including normalized rank information during training. As can be seen in Table 1, this improves the ρ performance and outperforms the baselines (Nickel & Kiela, 2017; 2018) for all the evaluation metrics on some datasets. The Mean Rank and Mean Average Precision performances remain comparable with the case where $\lambda = 0$. Global rank information can then be exploited during training without having a significant impact on retrieval performance. Increasing λ even more can improve ρ.

In conclusion, we have shown that the Lorentzian distance outperforms the standard Poincaré distance for retrieval (*i.e.* mean rank and MAP). In particular, retrieval performance can be improved by tuning the hyperparameter β.

4.2. Binary classification

Another task of interest for hyperbolic representations is to determine whether a given node belongs to a specific subtree of the hierarchy or not. We follow the same binary classification protocol as (Ganea et al., 2018) on the same datasets. We describe their protocol below.

(Ganea et al., 2018) extract some pre-trained hyperbolic embeddings of the WordNet nouns hierarchy, those representations are learned with the Poincaré metric. They then consider four subtrees whose roots are the following synsets: *animal.n.01*, *group.n.01*, *worker.n.01* and *mammal.n.01*. For each subtree, they consider that every node that belongs to it is positive and all the other nodes of Wordnet nouns are negative. They then select 80% of the positive nodes for training, the rest for test. They select the same percentage of negative nodes for training and test. At test time, they evaluate the F1 score of the binary classification performance. They do it for 3 different training/test splits. We refer to (Ganea et al., 2018) for details on the baselines.

Our goal is to evaluate the relevance of the Lorentzian distance to represent hierarchical datasets. We then use the embeddings trained with our approach (with $\beta = 0.01$ and different values of λ) and classify a test node by assigning it to the category of the nearest training example *w.r.t.* the Lorentzian distance (with $\beta = 1$). The test performance of our approach is reported in Table 2. It outperforms the classification performance of the baselines which are based on Poincaré or Euclidean distance. This shows that our embeddings can also be used to perform classification.

4.3. Automatic hierarchy extraction

The previous experiments compare the performances of the Lorentzian distance and the standard Poincaré distance but do not exploit the closed form expression of the center of mass since we use the same evaluation protocols as the baselines (*i.e.* we use the same pairwise constraints). We now exploit the formulation of our centroids to efficiently extract the category hierarchy of a dataset whose taxonomy is partially provided. In particular, we test our approach on the CIFAR-100 (Krizhevsky & Hinton, 2009) dataset which contains 100 classes/categories containing 600 images each. The 100 classes are grouped into 20 superclasses[3] (*e.g.* the classes *apples, mushrooms, oranges, pears, sweet peppers* belong to the superclass *fruits and vegetables*). We consider the hierarchy formed by these superclasses to learn hyberbolic representations with a neural network.

[3] The detailed class hierarchy of CIFAR-100 is available at `https://www.cs.toronto.edu/~kriz/cifar.html`

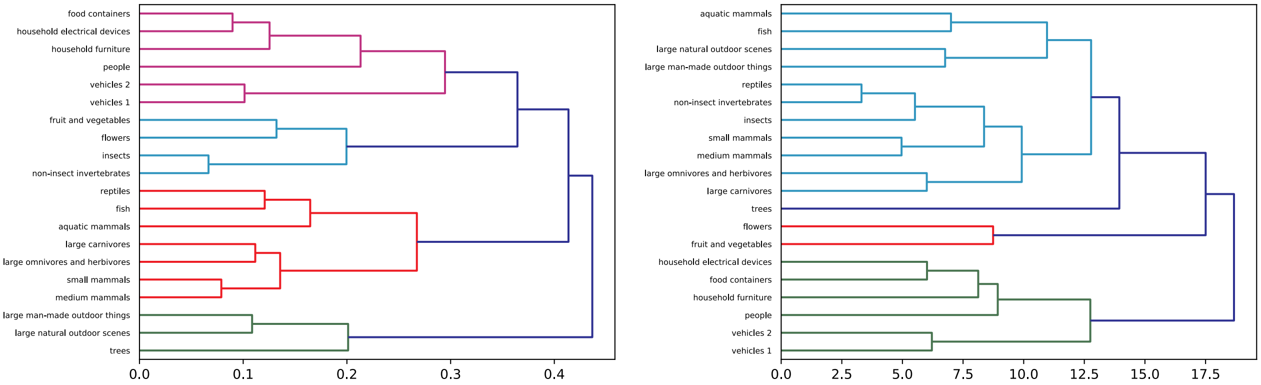


Figure 3. Taxonomy extracted for CIFAR-100 by optimizing Eq. (18) with the Lorentzian distance (left) and Euclidean distance (right)

Let $\{\mathcal{C}_c\}_{c=1}^{k}$ be the set of categories (with $k = 100$) and $\{\mathcal{A}_a\}_{a=1}^{\kappa}$ be the set of superclasses (with $\kappa = 20$) of the dataset. We formulate the posterior probability:

$$p(\mathcal{C}_c|\mathbf{x}_i) = \frac{\exp(-\mathsf{d}(\mathbf{x}_i, \boldsymbol{\mu}_{\mathcal{C}_c}))}{\sum_{m=1}^{k} \exp(-\mathsf{d}(\mathbf{x}_i, \boldsymbol{\mu}_{\mathcal{C}_m}))} \tag{16}$$

$$p(\mathcal{A}_a|\mathbf{x}_i) = \frac{\exp(-\mathsf{d}(\mathbf{x}_i, \boldsymbol{\mu}_{\mathcal{A}_a}))}{\sum_{e=1}^{\kappa} \exp(-\mathsf{d}(\mathbf{x}_i, \boldsymbol{\mu}_{\mathcal{A}_e}))} \tag{17}$$

where d is the chosen distance. Our goal is to learn the representations $\mathbf{x}_i$ and $\boldsymbol{\mu}$ that minimize the following loss:

$$-\sum_{\mathbf{x}_i \in \mathcal{C}_c} \log(p(\mathcal{C}_c|\mathbf{x}_i)) - \alpha \sum_{\mathbf{x}_i \in \mathcal{A}_a} \log(p(\mathcal{A}_a|\mathbf{x}_i)) \tag{18}$$

where $\alpha \geq 0$ is a regularization parameter. Eq. (18) tries to group samples which belongs to the same (super)-class into a single cluster. It corresponds to the *supervised clustering* task (Law et al., 2016; 2019). Following works on clustering, the optimal values of the different $\boldsymbol{\mu}$ are the centers of mass of the elements belonging to the respective (super-)classes. If the chosen metric is the squared Lorentzian distance, the optimal value of $\boldsymbol{\mu}_{\mathcal{C}_c}$ is:

$$\boldsymbol{\mu}_{\mathcal{C}_c} = \sqrt{\beta} \frac{\sum_{i=1}^{n} \nu_i \mathbf{x}_i}{|\|\sum_{i=1}^{n} \nu_i \mathbf{x}_i\|_{\mathcal{L}}|} \quad \text{s.t. } \nu_i = \mathbf{1}_{\mathcal{C}_c}(\mathbf{x}_i) \tag{19}$$

where $\mathbf{1}$ is the indicator function and $n = 600$ is the size of the mini-batch. We trained a convolutional neural network φ so that its output of the i-th image is $\mathbf{f}_i \in \mathcal{F}^d$ and $\mathbf{x}_i = g_\beta(\mathbf{f}_i)$ (see details in the appendix). One advantage of our loss function is that its complexity is linear in the number of examples and linear in the number of (super-)classes. The algorithm is then more efficient than pairwise constraint losses such as Eq. (13) where the cardinality of the negative nodes $\mathcal{N}(u)$ is so large that it has to be subsampled (*e.g.* with the sampling strategy in (Jean et al., 2015)).

From Theorem 3.5, each category can be represented by a single centroid. We then trained a neural network with output dimensionality $d = 10$ on the whole dataset, and extracted the super-class centroids on which we applied hierarchical agglomerative clustering based on complete-linkage clustering (Defays, 1977). Fig. 3 illustrates the extracted taxonomies when the chosen distance d is the squared Lorentzian distance (left) or the squared Euclidean distance (right). The hierarchical clusters extracted with hyperbolic representations are more natural. For instance, *insects* are closer to *flowers* with the Lorentzian distance. This is explained by the fact that bees (known for their role in pollination) are in the *insects* superclass. *Trees* are also closer to outdoor scenes and things with the Lorentzian distance. The *reptile* superclass contains aquatic animals (*e.g.* turtles and crocodiles), which is why it is close to *fish* and *aquatic mammals*.

5. Conclusion

In this paper, we proposed a distance learning approach based on the Lorentzian distance to represent hierarchically-structured datasets. Unlike most of the literature that considers the unit hyperboloid model, we show that the performance of the learned model can be improved by decreasing the curvature of the chosen hyperboloid model. We give a formulation of the centroid *w.r.t.* the squared Lorentzian distance as a function of the curvature, and we show that the Euclidean norm of its projection in the Poincaré ball decreases as the curvature decreases. Hierarchy constraints are generally formulated such that high-level nodes are similar to all their descendants and thus their representation should be close to the centroid of the descendants. Decreasing the curvature implicitly enforces high-level nodes to have smaller Euclidean norm than their descendants and is therefore is more appropriate for learning representations of hierarchically-structured datasets.

Acknowledgements

We thank Kuan-Chieh Wang and the anonymous reviewers for helpful feedback on early versions of this manuscript. This work was supported by Samsung and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

References

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

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

Daniel Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, 1977.

Albert Einstein. Zur elektrodynamik bewegter körper. *Annalen der physik*, 322(10):891–921, 1905.

Maurice Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. *Ann. Inst. H. Poincaré*, 10(3):215–310, 1948.

GA Galperin. A concept of the mass center of a system of material points in the constant curvature spaces. *Communications in Mathematical Physics*, 154(1):63–84, 1993.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 2018.

Mikhael Gromov. Hyperbolic groups. In *Essays in group theory*, pp. 75–263. Springer, 1987.

Karsten Grove and Hermann Karcher. How to conjugatec 1-close group actions. *Mathematische Zeitschrift*, 132(1): 11–20, 1973.

Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. Hyperbolic attention networks. In *International Conference on Learning Representations*, 2019.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pp. 1–10, 2015.

Hermann Karcher. *Riemannian comparison constructions*. SFB 256, 1987.

Hermann Karcher. Riemannian center of mass and so called karcher mean. *arXiv preprint arXiv:1407.2087*, 2014.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Marc T Law, Yaoliang Yu, Matthieu Cord, and Eric P Xing. Closed-form training of mahalanobis distance for supervised clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3909–3917, 2016.

Marc T Law, Jake Snell, Amir massoud Farahmand, Raquel Urtasun, and Richard S Zemel. Dimensionality reduction for representing the knowledge of probabilistic models. In *International Conference on Learning Representations*, 2019.

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

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

Risheng Liu, Zhouchen Lin, Zhixun Su, and Kewei Tang. Feature extraction by learning lorentzian metric tensor and its extensions. *Pattern Recognition*, 43(10):3298–3306, 2010.

George Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

Maximilian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. *International Conference on Machine Learning*, 2018.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pp. 6338–6347, 2017.

Yann Ollivier. Riemannian metrics for neural networks i: feedforward networks. *Information and Inference: A Journal of the IMA*, 4(2):108–153, 2015.

John Ratcliffe. *Foundations of hyperbolic manifolds*, volume 149. Springer Science & Business Media, 2006.

Frank B Rogers. Medical subject headings. *Bulletin of the Medical Library Association*, 51:114–116, 1963.

Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International Conference on Machine Learning*, pp. 4457–4466, 2018.

John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

Ke Sun, Jun Wang, Alexandros Kalousis, and Stéphane Marchand-Maillet. Space-time local embeddings. In *Advances in Neural Information Processing Systems*, pp. 100–108, 2015.

Abraham Albert Ungar. *Barycentric calculus in Euclidean and hyperbolic geometry: A comparative introduction*. World Scientific, 2010.

Abraham Albert Ungar. *Analytic hyperbolic geometry in n dimensions: An introduction*. CRC Press, 2014.

Vladimir Varicak. Beiträge zur nichteuklidischen geometrie. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 17:70–83, 1908.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: l2 hypersphere embedding for face verification. In *Proceedings of the 2017 ACM on Multimedia Conference*, pp. 1041–1049. ACM, 2017.