Hyperbolic Manifold Regression

Gian Maria Marconi[1] **Lorenzo Rosasco**[1,3,4] **Carlo Ciliberto**[2]

[1]Istituto Italiano di Tecnologia, Genoa, Italy [2] Imperial College of London, London, UK

[3]Massachusetts Institute of Technology, MA, USA [4]University of Genoa, Genoa, Italy

Abstract

Geometric representation learning has recently shown great promise in several machine learning settings, ranging from relational learning to language processing and generative models. In this work, we consider the problem of performing manifold-valued regression onto an hyperbolic space as an intermediate component for a number of relevant machine learning applications. In particular, by formulating the problem of predicting nodes of a tree as a manifold regression task in the hyperbolic space, we propose a novel perspective on two challenging tasks: 1) hierarchical classification via label embeddings and 2) taxonomy extension of hyperbolic representations. To address the regression problem we consider previous methods as well as proposing two novel approaches that are computationally more advantageous: a parametric deep learning model that is informed by the geodesics of the target space and a non-parametric kernel-method for which we also prove excess risk bounds. Our experiments show that the strategy of leveraging the hyperbolic geometry is promising. In particular, in the taxonomy expansion setting, we find that the hyperbolic-based estimators significantly outperform methods performing regression in the ambient Euclidean space.

1 Introduction

Representation learning is a key paradigm in machine learning and artificial intelligence. It has enabled important breakthroughs in computer vision (Krizhevsky et al., 2012; He et al., 2016) natural language processing (Mikolov et al., 2013; Bojanowski et al., 2016; Joulin et al., 2016), relational learning (Nickel et al., 2011; Perozzi et al., 2014), generative modeling (Kingma and Welling, 2013; Radford et al., 2015), and many other areas (Bengio et al., 2013; LeCun et al., 2015). Its objective is typically to infer latent feature representations of objects (e.g., images, words, entities, concepts) such that their similarity or distance in the representation space captures their *semantic* similarity. For this purpose, the geometry of the representation space has recently received increased attention (Wilson et al., 2014; Falorsi et al., 2018; Davidson et al., 2018; Xu and Durrett, 2018). Here, we focus on Riemannian representation spaces and in particular on hyperbolic geometry. Nickel and Kiela (2017) introduced Poincaré embeddings to infer hierarchical representations of symbolic data, which led to substantial gains in representational efficiency and generalization performance. Hyperbolic representations have since been extended to other manifolds (Nickel and Kiela, 2018; De Sa et al., 2018), word embeddings (Tifrea et al., 2018; Le et al., 2019), recommender systems (Chamberlain et al., 2019), and image embeddings (Khrulkov et al., 2019).

However, it is yet an open problem how to efficiently integrate hyperbolic representations with standard machine learning methods which often make a Euclidean or vector space assumption. The work of Ganea et al. (2018) establishes some fundamental steps in this direction by proposing a generalization of fully connected neural network layers from Euclidean space to hyperbolic space. However most of the experiments shown were from hyperbolic to Euclidean space using recurrent models. In this paper we focus on the task of learning manifold-valued functions from Euclidean on to hyperbolic space that allows us to leverage its hierarchical structure for supervised learning. For this purpose, we propose two novel approaches: a deep learning model trained with a geodesic-based loss to learn hyperbolic-valued functions and a non-parametric

Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

kernel-based model for which we provide a theoretical analysis.

We illustrate the effectiveness of this strategy on two challenging tasks, i.e., hierarchical classification via label embeddings and taxonomy expansion by predicting concept embeddings from text. For standard classification tasks, label embeddings have shown great promise as they allow to scale supervised learning methods to datasets with massive label spaces (Chollet, 2016; Veit et al., 2018). By embedding labels in hyperbolic space according to their natural hierarchical structure (e.g, the underlying WordNet taxonomy of ImageNet labels) we are then able to combine the benefits of hierarchical classification with the scalability of label embeddings. Moreover, the continuous nature of hyperbolic space allows the model to *invent* new concepts by predicting their placement in a pre-embedded base taxonomy. We exploit this property for a novel task which we refer to as *taxonomy expansion*: Given an embedded taxonomy $\mathcal{T}$, we infer the placement of unknown novel concepts by predicting their features onto the embedding. In contrast to hierarchical classification, the predicted embeddings are here full members of the taxonomy, i.e., they can themselves act as parents of other points. For both tasks, we show empirically that the proposed strategy can often lead to more effective estimators than its Euclean counterpart. These findings support the thesis of this work that leveraging the hyperbolic geometry can be advantageous for several machine learning settings. Additionally, we observe that the hyperbolic-based estimators introduced in this work achieve comparable performance to the previously proposed hyperbolic neural networks (Ganea et al., 2018). This suggests that, in practice, it is not necessary to work with hyperbolic layers as long as the training procedure exploits the geodesic as an error measure. This is advantageous from the computational perspective, since we found our proposed approaches to be generally significantly easier to train in practice.

The remainder of this paper is organized as follows: In Section 2 we briefly review hyperbolic embeddings and related concepts such as Riemannian optimization. In Section 3, we introduce our proposed methods and prove excess risk bounds for the kernel-based method. In Section 4 we evaluate our methods on the tasks of hierarchical classification and taxonomy expansion.

2 Hyperbolic Representations

Hyperbolic space is the unique, complete, simply connected Riemannian manifold with constant negative sectional curvature. There exist multiple equivalent models for hyperbolic space. To estimate the embeddings using stochastic optimization we will employ the Lorentz model due to its numerical advantages. For analysis, we will map embeddings into the Poincaré disk which provides an intuitive visualization of hyperbolic embeddings. This can be easily done because the two models are isometric Nickel and Kiela (2018). We review both manifolds in the following.

Lorentz Model. Let $\boldsymbol{u}$, $\boldsymbol{v} \in \mathbb{R}^{n+1}$ and let $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathcal{L}} = -u_0 v_0 + \sum_{i=1}^{n} u_n v_n$ denote the *Lorentzian scalar product*. The Lorentz model of n-dimensional hyperbolic space is then defined as the Riemannian manifold $\mathcal{L}^n = (\mathcal{H}^n, g_{\mathcal{L}})$, where

$$\mathcal{H}^n = \{\boldsymbol{u} \in \mathbb{R}^{n+1} : \langle \boldsymbol{u}, \boldsymbol{u} \rangle_{\mathcal{L}} = -1, x_0 > 0\}, \quad (1)$$

denotes the upper sheet of a two-sheeted n-dimensional hyperboloid and where $g_{\mathcal{L}}(\boldsymbol{u}) = \mathrm{diag}([-1, 1, \ldots, 1])$ is the associated metric tensor. Furthermore, the distance on $\mathcal{L}$ is defined as

$$d_{\mathcal{L}}(\boldsymbol{u}, \boldsymbol{v}) = \mathrm{acosh}(-\langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\mathcal{L}}). \quad (2)$$

An advantage of the Lorentz model is that its exponential map has as simple, closed-form expression. As showed by Nickel and Kiela (2018), this allows us to perform Riemannian optimization efficiently and with increased numerical stability. In particular, let $\boldsymbol{u} \in \mathcal{L}^n$ and let $\boldsymbol{z} \in \mathcal{T}_{\boldsymbol{u}}\mathcal{L}^n$ denote a point in the associated tangent space. The exponential map $\exp_{\boldsymbol{u}} : \mathcal{T}_{\boldsymbol{u}}\mathcal{L}^n \to \mathcal{L}^n$ is then defined as

$$\exp_{\boldsymbol{u}}(\boldsymbol{z}) = \cosh(\|\boldsymbol{z}\|_{\mathcal{L}})\boldsymbol{u} + \sinh(\|\boldsymbol{z}\|_{\mathcal{L}})\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|_{\mathcal{L}}}. \quad (3)$$

Poincaré ball. The Poincaré ball model is the Riemannian manifold $\mathcal{P}^n = (\mathcal{B}^n, g_p)$, where $\mathcal{B}^n = \{\boldsymbol{u} \in \mathbb{R}^n : \|\boldsymbol{u}\| < 1\}$ is the *open* n-dimensional unit ball and where $g_p(\boldsymbol{u}) = 4/(1 - \|\boldsymbol{u}\|^2)^2$ is the associated metric tensor. The distance function on $\mathcal{P}$ is defined as

$$d_p(\boldsymbol{u}, \boldsymbol{v}) = \mathrm{acosh}\left(1 + 2\frac{\|\boldsymbol{u} - \boldsymbol{v}\|^2}{(1 - \|\boldsymbol{u}\|^2)(1 - \|\boldsymbol{v}\|^2)}\right). \quad (4)$$

An advantage of the Poincaré ball is that it provides an intuitive model of hyperbolic space which is well suited for analysis and visualization of the embeddings. It can be seen from Eq. (4), that the distance within the Poincaré ball changes smoothly with respect to the norm of $\boldsymbol{u}$ and $\boldsymbol{v}$. This locality property of the distance is key for representing hierarchies efficiently (Hamann, 2018). For instance, by placing the root node of a tree at the origin of $\mathcal{B}^n$, it would have relatively small distance to all other nodes, as its norm is zero. On the other hand, leaf nodes can be placed close to the boundary of the ball, as the distance between points grows quickly with a norm close to one.

Hyperbolic embeddings. We consider supervised datasets $\{x_i, c_i\}_{i=1}^m \in \mathcal{X} \times \mathcal{C}$ where class labels c_i can be organized according to a taxonomy or class hierarchy $\mathcal{T} = (\mathcal{C}, \mathcal{E})$. Edges $(i, j) \in \mathcal{E}$ indicate that c_i *is-a* c_j. To compute hyperbolic embeddings of all c_i that capture these hierarchical relationships of $\mathcal{T}$, we follow the works of Nickel and Kiela (2017, 2018) and infer the embedding from pairwise similarities. In particular, let $\gamma : \mathcal{C} \times \mathcal{C} \to \mathbb{R}_+$ be the similarity function such that

$$\gamma(c_i, c_j) = \begin{cases} 1, & \text{if } c_i,\ c_j \text{ are adjacent in } clos(\mathcal{T}) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $clos(\mathcal{T})$ is the transitive closure of $\mathcal{T}$. Furthermore, let $\mathcal{N}(i, j) = \{\ell : \gamma(i, \ell) < \gamma(i, j)\} \cup \{j\}$ denote the set of concepts that are *less* similar to c_i then c_j (including c_j) and let $\phi(i, j) = \arg\min_{k \in \mathcal{N}(i,j)} d(\boldsymbol{u}_i, \boldsymbol{u}_k)$ denote the nearest neighbor of c_i in the set $\mathcal{N}(i, j)$. We then learn embeddings $\Theta = \{\boldsymbol{u}\}_{i=1}^m$ by optimizing

$$\min_{\Theta} -\sum_{i,j} \log \Pr(\phi(i, j) = j \mid \Theta) \tag{6}$$

with

$$\Pr(\phi(i, j) = j \mid \Theta) = \frac{e^{d(\boldsymbol{u}_i, \boldsymbol{u}_j)}}{\sum_{k \in \mathcal{N}(i,j)} e^{d(\boldsymbol{u}_i, \boldsymbol{u}_k)}}. \tag{7}$$

Eq. (7) can be interpreted as a ranking loss that aims to extract latent hierarchical structures from $\mathcal{C}$. For computational efficiency, we follow Jean et al. (2014) and randomly subsample $\mathcal{N}(i, j)$ on large datasets. To infer the embeddings θ we then minimize Eq. (7) using Riemannian SGD (Bonnabel, 2013). In RSGD, updates to the parameters θ are computed via

$$\theta_{t+1} = \exp_{\theta_t}\Big(-\eta \sum_{j \in B} \operatorname{grad}_{\mathcal{L}} f_j(\theta_t) \Big), \tag{8}$$

where $\operatorname{grad}_{\mathcal{L}} f(\theta_t) \in \mathcal{T}_\theta \mathcal{L}$ denotes the *Riemannian gradient*, η denotes the learning rate, and $B = [j_1, \dots, j_B]$ is a set of random uniformly sampled indexes.

By computing hyperbolic embeddings of $\mathcal{T}$, we have then recast the learning problem from a discrete tree $\mathcal{D} = \{x_i, c_i\}_{i=1}^m, c_i \in \mathcal{C}$ to its embedding in a continuous manifold $\mathcal{D}^e = \{x_i, y_i\}_{i=1}^m$ with $y_i \in \mathcal{L}$. This allows us to apply manifold regression techniques as discussed in the following.

3 Manifold Valued Prediction in Hyperbolic Space

We study the problem of learning $f : \mathcal{X} \to \mathcal{Y} \subset \mathcal{L}^n$ a map taking values in the hyperbolic space, often referred to as *manifold regression* (Steinke and Hein, 2009; Steinke et al., 2010). We assume for simplicity that $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathcal{L}^n$ are compact subsets. In particular, we assume a training dataset $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^m$ of points independently sampled from a joint distribution ρ on $\mathcal{X} \times \mathcal{Y}$ and aim to find an estimator for the minimizer of the expected risk

$$\min_{f : \mathcal{X} \to \mathcal{Y}} \ \mathcal{E}(f) \quad \mathcal{E}(f) = \int d_{\mathcal{L}}(f(\boldsymbol{x}), \boldsymbol{y})^2 \ d\rho(\boldsymbol{x}, \boldsymbol{y}). \tag{9}$$

Here we consider $\mathcal{L}^n$ as target space and $d_{\mathcal{L}}$ as loss function, but all results extend to $\mathcal{P}$. Eq. (9) is the natural generalization of standard vector-valued ridge regression (indeed the geodesic of $\mathcal{Y} = \mathbb{R}^n$ is the Euclidean distance $d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|$). We tackle this problem proposing two novel approaches: one leveraging recent results on structured prediction and one using geodesic neural networks.

Structured Prediction. Rudi et al. (2018) proposed a new approach to address manifold regression problems. The authors adopted a perspective based on structured prediction and interpreted the target manifold $\mathcal{Y}$ as a "structured" output. While standard structured prediction studies settings where $\mathcal{Y}$ is a discrete (often finite) space (Bakir et al., 2007), this extension allowed the authors to design a kernel-based approach for structured prediction for which they provided a theoretical analysis under suitable assumptions on the output space. We formulate the corresponding Hyperbolic Structured Prediction (HSP) estimator when applying this strategy to our problem (namely $\mathcal{Y} \subset \mathcal{L}^n$). In particular, we have $f_{hsp} : \mathcal{X} \to \mathcal{L}^n$ the function such that for any test point $\boldsymbol{x} \in \mathcal{X}$

$$f_{hsp}(\boldsymbol{x}) = \operatorname*{arg\,min}_{\boldsymbol{y} \in \mathcal{L}^n} \ \sum_{i=1}^m \alpha_i(\boldsymbol{x}) \ d_{\mathcal{L}}(\boldsymbol{y}, \boldsymbol{y}_i)^2, \tag{10}$$

where the weights $\alpha(\boldsymbol{x}) = (\alpha_1(\boldsymbol{x}), \dots, \alpha_n(\boldsymbol{x}))^\top \in \mathbb{R}^m$ are learned by solving a variant of kernel ridge regression: given $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a reproducing kernel on the input space, we obtain

$$\alpha(\boldsymbol{x}) = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{v}(\boldsymbol{x}), \tag{11}$$

where $\mathbf{K} \in \mathbb{R}^{m \times m}$ is the empirical kernel matrix and $\mathbf{v} \in \mathbb{R}^m$ is the evaluation vector with entries with entries respectively $\mathbf{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\mathbf{v}(\boldsymbol{x})_i = k(\boldsymbol{x}, \boldsymbol{x}_i)$ for $i, j \in \{1, \dots, m\}$.

In line with most literature on structured prediction, the estimator in (10) requires solving an optimization problem at every test point. Hence, while this approach offers a significant advantage at training time (when learning the weights α), it can lead to a more expensive operation at test time. To solve this problem in practice we resort to RSGD as defined in (8).

Rudi et al. (2018), studied the generalization properties of estimators of the form of (10). The authors proved that under suitable assumptions on the regularity of the output manifold, it was possible to give bounds on the excess risk in terms of the number of training examples available. The following theorem specializes this result to the case of f_{hsp}. A key role will be played by the $(s, 2)$-Sobolev space $W^{s,2}$ of functions from $\mathcal{L}^n$ to $\mathbb{R}$, which generalizes the standard notion on Euclidean domains (see Hebey, 2000).

Theorem 1. *Let $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^m$ be sampled independently according to ρ on $\mathcal{X} \times \mathcal{Y}$ with $\mathcal{Y} \subset \mathcal{L}^m$ compact sets. Let f_{hsp} defined as in (10) with weights (11) learned with reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with reproducing kernel Hilbert space (RKHS) $\mathcal{F}$. If the map $x \mapsto \int d_{\mathcal{L}}(\cdot, \boldsymbol{y})^2 \, d\rho(y|x)$ belongs to $W^{s,2}(\mathcal{Y}) \otimes \mathcal{F}$ with $s > n/2$, then for any $\tau \in (0, 1]$*

$$\mathcal{E}(f_{hsp}) - \inf_f \mathcal{E}(f) \quad \leq \quad \left\| d^2_{\mathcal{L}} \right\|_{s,2} \mathsf{q} \tau^2 \, \frac{1}{n^{1/4}}, \qquad (12)$$

holds with probability at least $1 - 8e^{-\tau}$, where q is a constant not depending on n, τ or $\|d_{\mathcal{L}}\|_{s,2}$.

The result guarantees a learning rate of order $O(n^{-1/4})$. We comment on the assumptions and constants appearing in Thm. 1. First, we point out that, albeit the requirement $\int d_{\mathcal{L}}(\cdot, \boldsymbol{y})^2 \, d\rho(y|x) \in \mathcal{F} \otimes W^{s,2}(\mathcal{Y})$ can seem overly abstract, it reduces to a standard assumption in statistical learning theory. Informally, it corresponds to a regularity assumption on the conditional mean embedding of the distribution $\rho(\cdot|x)$ (see the work of Song et al. (2013) for more details), and can be interpreted as requiring the solution of (9) to belong to the hypotheses space associated to the kernel k. Second, we comment on the constant in (12) that depending on the geodesic distance. In particular, we note that by Thm. 2 of Rudi et al. (2018) the squared geodesic on any compact subset of $\mathcal{L}^n$ belongs to $W^{s,2}(\mathcal{Y})$ for any $s \geq 0$. Hence $\left\| d^2_{\mathcal{L}} \right\|_{s,2} < +\infty$ also for any $s > n/2$, as required by Thm. 1.

Proof. The proof of Thm. 1 is a specialization of Thm. 2 and 4 by Rudi et al. (2018). We recall a key assumption that is required to apply such results.

Assumption 1. *$\mathcal{M}$ is a complete n-dimensional smooth connected Riemannian manifold, without boundary, with Ricci curvature bounded below and positive injectivity radius.*

The assumption above imposes basic regularity conditions on the output manifold. A first implication is indeed that.

Proposition 2 (Thm. 2 in Rudi et al. (2018))**.** *Let $\mathcal{M}$ satisfy assumption 1 and let $\mathcal{Y} \subset \mathcal{M}$ is a compact geodesically convex subset of $\mathcal{M}$. Then, the squared geodesic distance $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is smooth on $\mathcal{Y}$. Moreover, by the proof of Thm.1 in the appendix of Manifold Structured Prediction (Rudi et al., 2018), we have $d^2 \in W^{s,2}(\mathcal{Y})$ for any $s > n/2$.*

Leveraging standard results from Riemannian geometry, we can guarantee that the manifolds considered in this paper satisfy the above requirements. For simplicity, we restrict on $\mathcal{M}$ corresponding to an open bounded ball in either $\mathcal{P}^n$ or $\mathcal{L}^n$. In particular,

- $\mathcal{M}$ has sectional curvature constantly equal to -1. Hence the Ricci curvature is bounded from below since we are in a bounded ball in either $\mathcal{P}^n$ or $\mathcal{L}^n$.
- The injectivity radius is positive (actually lower bounded by $1/(2 \cdot 9^{2+[n/2]})$ with $[n/2]$ the integer parts of $n/2$), see Main Theorem by Martin (1989).

We see that we are in the hypotheses of Prop. 2, from which we conclude the following.

Corollary 3. *For any $s \geq 0$, the geodesic distance $d_{\mathcal{L}}$ (respectively $d_{\mathcal{P}}$) belongs to $W^{s,2}(\mathcal{Y})$ for any compact subspace of $\mathcal{L}^n$ (respectively $\mathcal{P}^n$).*

This guarantees us that we are in the hypotheses of (Rudi et al., 2018, Thm. 4), from which Thm. 1 follows. We note in particular that $d^2_{\mathcal{L}}$ takes the role of the loss function $\triangle$ in the original theorem. Which needs to be a so-called "Structure Encoding Loss Function". The latter is guaranteed by Cor. 3 above. □

Neural Network with Geodesic loss (NN-G). As an alternative to the non-parametric model f_{hsp}, we consider also a parametric method based on deep neural networks. An important challenge when dealing with manifold regression is how to design a suitable model for the estimator. While neural networks of the form $g_\theta \colon \mathbb{R}^d \to \mathbb{R}^k$ (parametrized by some weights θ) have proven to be powerful models for regression and feature representation (LeCun et al., 2015; Bengio et al., 2013; Xiao et al., 2016; Ngiam et al., 2011), it is unclear how to enforce the constraint for a candidate function to take values on the manifold since their canonical forms are designed to act between linear spaces. To address this limitation, we consider in the following the Poincaré ball model and develop a neural architecture mapping the Euclidean space into the open unit ball. In particular, let the element-wise hyperbolic tangent be defined as

$$h \colon \mathbb{R}^k \to \{\boldsymbol{x} \in \mathbb{R}^k : \|\boldsymbol{x}\|_\infty < 1\} \qquad (13)$$

$$(x_1, \ldots, x_k) \mapsto (\tanh x_1, \ldots, \tanh x_k), \qquad (14)$$

which maps a linear space onto the open ℓ_∞ ball. Moreover, we define a "squashing" function

$$s\colon \{\boldsymbol{x} \in \mathbb{R}^k : \|\boldsymbol{x}\|_\infty < 1\} \to \{\boldsymbol{x} \in \mathbb{R}^k : \|\boldsymbol{x}\|_2 < 1\} \tag{15}$$

$$s(\boldsymbol{x}) = \begin{cases} \boldsymbol{x} \mapsto \boldsymbol{x}\frac{\|\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_2}, & \text{if } \boldsymbol{x} \neq \mathbf{0} \\ \mathbf{0}, & \text{if } \boldsymbol{x} = \mathbf{0} \end{cases} \tag{16}$$

where $\mathbf{0}$ is the vector of all zeros. Since $\|\boldsymbol{x}\|_\infty < \|\boldsymbol{x}\|_2$, this function is continuous and maps the open ℓ_∞ ball into the open ℓ_2 ball. And because both s and h are bijective continuous function with continuous inverse, the composition $s \circ h\colon \mathbb{R}^k \to \{\boldsymbol{x} \in \mathbb{R}^k : \|\boldsymbol{x}\|_2 < 1\}$ is also a homeomorphism from $\mathbb{R}^k$ into the open ball ℓ_2 and therefore also on the Poincaré model manifold. By composing $s \circ h$ with the neural network feature extractor g_θ we obtain a deep model that jointly learns features into a linear space and maps them to the hyperbolic manifold:

$$f_{nng} = s \circ h \circ g_\theta\colon \mathbb{R}^d \to \mathcal{P}^k. \tag{17}$$

Note that the homeomorphism $s \circ h$ is sub-differentiable. Therefore learning the parameters θ of this model is akin to training a classical deep learning architecture with activation functions at the output layer corresponding to $s \circ h$. The key difference here lies in the loss used for training. In this setting, analogously to the task addressed by HSP, we replaced of the standard mean-squared error (Euclidean) loss with the squared geodesic distance between predictions and true labels.

Hyperbolic embeddings and manifold regression. In this work we propose to leverage the hyperbolic geometry to address machine learning tasks where hierarchical structures play a central role. In particular, we combine label embeddings approaches with hyperbolic regression to perform hierarchical classification. We do this by following a two step procedure: assuming a hierarcy $\mathcal{T}$, we consider an augmented $\mathcal{T}_\mathcal{X}$ where each example x_i corresponds to a child to its associated class c_i from the original $\mathcal{T}$. Then, we embed $\mathcal{T}_\mathcal{X}$ into the hyperboilic space using the procedure reviewed in Section 2. We compute similarity scores $\gamma(\cdot, \cdot)$ in the transitive closure of $\mathcal{T}_\mathcal{X}$, using either a Gaussian kernel on the features – when both nodes have a corresponding representation available – or otherwise employing the original γ. This allows us to incorporate information about feature similarities within the label embedding.

4 Experiments

We evaluate our proposed methods for hyperbolic manifold regression on the following experiments:

Hierarchical Classification via Label Embeddings. For this task, the goal is to classify examples with a single label from a class hierarchy with tree structure. We begin by computing label embeddings of the class hierarchy via hyperbolic representations. We then learn to regress examples onto label embeddings and classify them using the nearest label in the target space, i.e., by denoting $\boldsymbol{y}_c \in \mathcal{L}^n$ the embedding of class c and taking $f : \mathbb{R}^d \to \mathcal{L}^n$.

$$\widehat{c} = \operatorname*{arg\,min}_{c \in \mathcal{C}} d(f(\boldsymbol{x}), \boldsymbol{y}_c) \tag{18}$$

Taxonomy expansion. For this task, the goal is to expand an existing taxonomy based on feature information about new concepts. As for hierarchical classification, we first embed the existing taxonomy in hyperbolic space and then learn to regress onto the label embeddings. However, a key difference is that a new example c can themselves act as the parent of another class c'.

Models and training details. For hierarchical classification, we compare to standard baselines such as top-down classification with logistic regression (TD-LR) and hierarchical SVM (HSVM). Furthermore, since both tasks can be regarded as regression problems onto the Poincaré ball (which has a canonical embedding in $\mathbb{R}^k$) we also compare to kernel regularized least squares regression (KRLS) and a neural network with squared Euclidean loss (NN-E). In both cases, we constrain predictions to remain within the Poincaré ball via the projection

$$\text{proj}(\boldsymbol{y}) = \begin{cases} \boldsymbol{y}/\|\boldsymbol{y}\| - \varepsilon & \text{if } \|\boldsymbol{y}\| \geq 1 \\ \boldsymbol{y} & \text{otherwise} \end{cases},$$

where ε is a small constant to ensure numerical stability, equal to $\varepsilon = 10^{-6}$. These regression baselines allows us to evaluate the advantages of training manifold-valued models with squared geodesic loss compared to standard methods that are agnostic of the underlying geodesics.

For kernel-based methods, we employ a Gaussian kernel selecting the bandwidth $\sigma \in [10^{-1}, 10^2]$ and regularization parameter $\lambda \in [10^{-6}, 10^{-2}]$ via cross-validation. Both parameter ranges are logarithmically spaced. For HSP inference we use RSGD with batch size equal to 50 and a maximum of 40000 iterations. We stop the minimization if the the gradient Euclidean norm is smaller than 10^{-5} (In most cases the inference stops before the 10000 iteration). The learning rate for RSGD is chosen via cross-validation on the interval $[10^{-5}, 10^{-1}]$. For the neural network models (NN-G, NN-E) we use the same architecture for g_θ: each layer is a fully connected network

$$z^\ell = \psi(W_\ell z^{\ell-1} + b_\ell)$$

		Model - Performance (Relative Rank)									
		TD-LR		HSVM		NN-E		NN-G		HSP	
News-20	μF1	77.07	(3)	**80.79**	(1)	63.91	(5)	72.67	(4)	80.28	(2)
	MF1	77.94	(3)	**80.04**	(1)	64.21	(5)	72.70	(4)	79.56	(2)
Imclef07a	μF1	73.86	(3)	74.98	(2)	65.49	(5)	67.49	(4)	**75.95**	(1)
	MF1	36.03	(3)	**50.44**	(1)	26.76	(5)	31.20	(4)	46.41	(2)
Wipo	μF1	36.85	(2)	**38.48**	(1)	16.87	(5)	16.69	(6)	31.94	(3)
	MF1	52.18	(3)	52.21	(2)	42.77	(5)	42.86	(4)	**52.41**	(1)
Diatoms	μF1	**54.01**	(1)	48.97	(3)	9.25	(5)	11.31	(4)	53.20	(2)
	MF1	55.53	(2)	44.61	(3)	14.90	(4)	14.61	(5)	**62.10**	(1)
	Avg. Rank		(2.5)		(1.75)		(4.88)		(4.38)		(1.75)

Table 1: Hierarchical classification on benchmark datasets. We report micro-F1 (μF1), macro-F1 (MF1), as well as the rank relative to all other models on a dataset, e.g., (1) for the the best performing model.

where $\psi(x) = \max(0, x)$ is a ReLU non-linearity and $\theta = \{W \in \mathbb{R}^{s/2 \times s}, b \in \mathbb{R}^{s/2}\}$, with s the dimension of the previous layer (with the exception of the first and last layer which must fit input and output dimensions). We use a depth of 5 layers with intermediate dimensionalities $s \in (1024, 1024, 512, 256, 128)$ for taxonomy expansion and $s \in (2048, 2048, 1024, 512, 256)$ for hierarchical classification. We did not find significant improvements with deeper architectures in performance. We train the deep models using mini-batch stochastic gradient descent, with a scheduler until the model reaches convergence on the training loss. For taxonomy expansion we also compare our algorithms with a hyperbolic neural networks (HNN) as introduced by Ganea et al. (2018). This architecture is trained with Riemannian Stochastic Gradient Descent until convergence and has the same structure and the same number of parameters of NN-G and NN-E. Because NN-G uses fully connected layers until the homeomorphic transformation, it can be trained with traditional optimizers such as stochastic gradient descent or Adam (Kingma and Welling, 2013). In our experiments, we observe that this can be an important advantage as these models require typically one third of the training time compared to HNNs.

4.1 Hierarchical classification

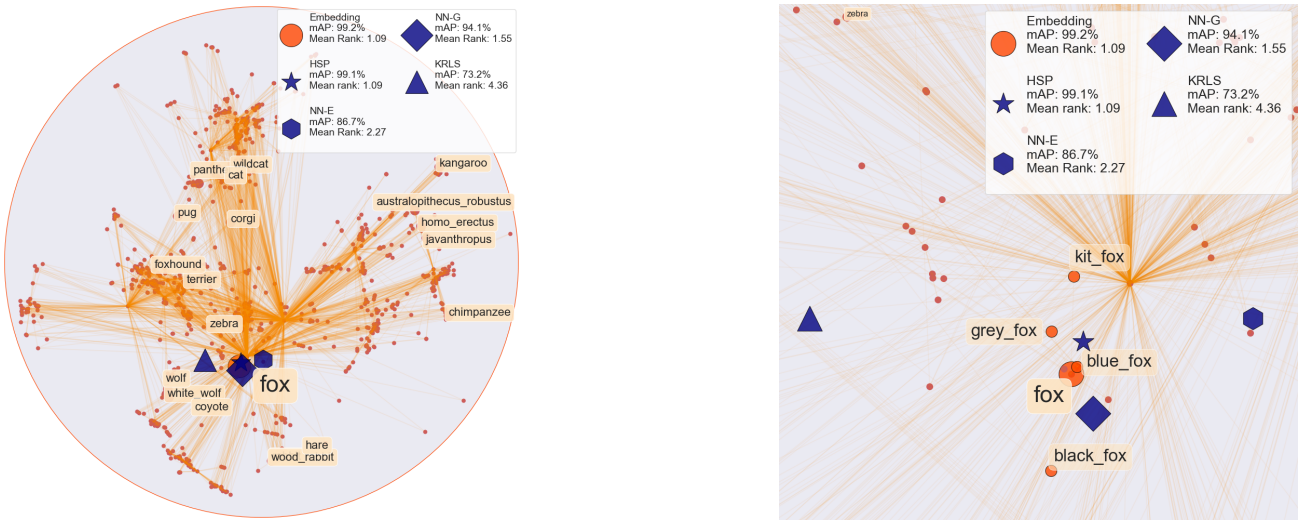
For hierarchical classification, we are given a supervised training set $\mathcal{D} = \{x_i, c_i\}_{i=1}^m$ where the class labels c_i are organized in a tree $\mathcal{T}$. We first embed the augmented hierarchy $\mathcal{T}_x$ as discussed in Section 3 and learn a regression function $\hat{f}\colon \mathbb{R}^d \to \mathcal{L}^n$ using $\mathcal{D}^e = \{x_i, y_i\}_{i=1}^m$. For a test point $\boldsymbol{x}' \in \mathbb{R}^d$, we first map it onto the target manifold $\hat{y} = \hat{f}(\boldsymbol{x}')$ and then classify $\hat{y}$ according to Eq. (18). For evaluation, we use various benchmark datasets for hierarchical classification[1], and Newsgroups-20[2] for which we manually extract TF-IDF features $x_i \in \mathbb{R}^{10000}$ from the original documents. We compute an embedding for the augmented hierarchies of each dataset. To make sure to obtain a good embedding, we perform parameter-tuning in order to attain mAP of at least 0.99. We then train HSP, NN-G and NN-E as described above and measure classification performance in terms of μF1 and macroF1 scores. As a baseline we also train Hierarchical SVM (HSVM) (Vateekul et al., 2012) and Top-Down Logisitic Regression (TD-LR) (Naik and Rangwala, 2018).

Table 1 shows the results of our experiments. It can be seen that the hyperbolic structured predictor achieves results comparable to state-of-the-art on this task although we did not explicitly optimize the embedding or training loss for hierarchical classification. We also observe that while NN-G outperforms NN-E, both algorithms perform significantly worse on Wipo and Diatoms datasets. Interestingly, these two datasets are significantly smaller compared to Newsgroup-20 and Imclef07a in terms of number of training points ($\sim 1K$ Vs $\sim 10K$ training samples). This seems to suggest that NN-G and NN-E models have a higher sample complexity.

4.2 Taxonomy expansion

For taxonomy expansion, we assume a similar setting as for hierarchical classification. We are given a dataset $\mathcal{D} = \{x_i, c_i\}_{i=1}^m$ where concepts c_i are organized in a taxonomy $\mathcal{T}$ and for each concept we have an additional feature representation x_i. Again, we first embed the augmented hierarchy $\mathcal{T}_x$ as discussed in Section 3 and split it in train $\mathcal{D}^e_{\text{train}}$ and test set $\mathcal{D}^e_{\text{test}}$. We vary

[1] `https://sites.google.com/site/hrsvmproject/`
[2] `http://qwone.com/~jason/20Newsgroups/`



(a) WordNet mammals embedding

(b) Close-up of predicted embedding for 'Fox'

Figure 1: Overview and close-up of predicted positions for entity 'Fox'. Models that do not use the geometry of the hyperbolic manifold fail at positioning the entity, while the geodesic neural network and the hyperbolic structured predictor position the entity accordingly to its real neighbours.

the size of the test set, i.e., the number of unknown concepts in $\mathcal{T}$ such that $|\mathcal{D}_{\text{test}}| \in \{5, 10, 20, 30, 50\}$. Whenever necessary, we also create a validation set from $\mathcal{D}_{\text{train}}$ for model selection with a $80:20$ ratio for model selection. We then train all regression functions $\hat{f}\colon \mathbb{R}^d \to \mathcal{L}^n$ using $\mathcal{D}^e_{\text{train}}$ and predict embeddings for $\mathcal{D}_{\text{test}}$. In contrast to hierarchical classification, the predicted points $\hat{\boldsymbol{y}} = f(\boldsymbol{x})$ can themselves act as parents of other points, i.e., they are full members of the taxonomy $\mathcal{T}$. To assess the quality of the predictions we use mean average prediction (mAP) as proposed by Nickel and Kiela (2017). We report mAP for the predicted points as well as for the points originally embedded by the Lorentz embedding (Orig). This experiment is repeated 20 times for a given size of the test set, each time selecting a new training-test split. In our experiments, we consider the following datasets:

WordNet Mammals. For WordNet Mammals, the goal is to expand an existing taxonomy by predicting concept embeddings from text. For this purpose, we take the mammals hierarchy of WordNet and retrieve for each node its corresponding Wikipedia page. If a page is missing, we remove the corresponding node and if a page has multiple candidates we disambiguate manually. The transitive closure of $\mathcal{T}$ has 1036 nodes and 11222 edges. Next, we pre-process the retrieved Wikipedia descriptions by removing all non alphabetical characters, tokenizing words and removing stopwords using NLTK (Loper and Bird, 2002). Finally, we associate to each concept $c_i \in \mathcal{T}$ the TF-IDF vector of its Wikipedia description as feature representation $x_i \in \mathbb{R}^{10000}$ computed using Scikit-learn (Pedregosa et al., 2011). We then embed $\mathcal{T}$ following Section 2 and obtain an embedding with mAP 0.86 and mean rank 4.74. This dataset is particularly difficult given the way features were collected: Wikipedia pages have a high variance in quality and amount of content, while some pages are detailed and rich in information other barely contain a full sentence.

Synthetic datasets. To better control for noise in the feature representations, we also generate datasets based on synthetic random trees, i.e., a smaller tree with 226 nodes and 1228 edges and a larger tree with 2455 nodes and 30829 edges after transitive closure. For each node we take as feature vector the corresponding row of the adjacency matrix of the transitive closure of the tree. We project these rows on the first d principal components of the adjacency matrix, where $d = 50$ for the small tree and $d = 500$ for the big tree. We then embed the nodes of the graph in $\mathcal{L}^5$ using both the tree structure and similarity scores computed using the vector features. The similarity is computed by a Gaussian kernel with σ equal to the average tenth nearest neighbour of the dataset.

Results We provide the results of our evaluation for different sizes on $\mathcal{D}^e_{\text{test}}$ in Table 2. It can be seen that all hyperbolic-based methods can successfully predict the embeddings of unknown concepts when the test set is small. The performance degrades as the size of the test set increases, since it becomes harder to leverage the original structure of the graph. While all methods are affected by this trend, we note that algorithms using the geodesic loss tend to perform better than those working in the linear space. This suggest that taking

		Number of new concepts				
		5	10	20	30	50
Wordnet Mammals	Orig	0.86 ± 0.06	0.88 ± 0.06	0.87 ± 0.03	0.87 ± 0.03	0.88 ± 0.02
	KRLS	0.54 ± 0.14	0.37 ± 0.07	0.26 ± 0.04	0.22 ± 0.03	0.15 ± 0.02
	NN-E	0.61 ± 0.12	0.47 ± 0.08	0.38 ± 0.04	0.31 ± 0.03	0.20 ± 0.03
	NN-G	0.79 ± 0.08	$\mathbf{0.74 \pm 0.06}$	0.63 ± 0.06	$\mathbf{0.61 \pm 0.06}$	$\mathbf{0.50 \pm 0.04}$
	HNN	$\mathbf{0.82 \pm 0.05}$	0.73 ± 0.05	0.63 ± 0.04	0.57 ± 0.05	0.46 ± 0.04
	HSP	0.72 ± 0.10	0.69 ± 0.07	$\mathbf{0.69 \pm 0.07}$	0.58 ± 0.09	$\mathbf{0.50 \pm 0.06}$
Synthetic Small	Orig	0.94 ± 0.03	0.93 ± 0.03	0.94 ± 0.02	0.94 ± 0.02	0.94 ± 0.01
	KRLS	0.63 ± 0.16	0.51 ± 0.12	0.36 ± 0.06	0.27 ± 0.03	0.21 ± 0.02
	NN-E	0.76 ± 0.07	0.72 ± 0.09	0.63 ± 0.09	0.56 ± 0.09	0.45 ± 0.08
	NN-G	0.80 ± 0.07	0.73 ± 0.06	0.61 ± 0.06	0.55 ± 0.05	0.45 ± 0.04
	HNN	$\mathbf{0.82 \pm 0.01}$	0.71 ± 0.07	0.60 ± 0.05	0.51 ± 0.04	0.41 ± 0.04
	HSP	$\mathbf{0.82 \pm 0.08}$	$\mathbf{0.76 \pm 0.07}$	$\mathbf{0.66 \pm 0.05}$	$\mathbf{0.60 \pm 0.04}$	$\mathbf{0.50 \pm 0.03}$
Synthetic Large	Orig	0.81 ± 0.06	0.79 ± 0.05	0.80 ± 0.03	0.80 ± 0.02	0.80 ± 0.01
	KRLS	0.30 ± 0.05	0.20 ± 0.02	0.13 ± 0.01	0.09 ± 0.01	0.07 ± 0.00
	NN-E	0.69 ± 0.09	0.68 ± 0.09	0.64 ± 0.05	0.61 ± 0.05	0.59 ± 0.05
	NN-G	0.77 ± 0.07	0.72 ± 0.07	0.71 ± 0.04	$\mathbf{0.69 \pm 0.04}$	$\mathbf{0.65 \pm 0.03}$
	HNN	$\mathbf{0.83 \pm 0.7}$	$\mathbf{0.79 \pm 0.4}$	$\mathbf{0.72 \pm 0.06}$	0.64 ± 0.06	0.63 ± 0.02
	HSP	0.76 ± 0.09	0.70 ± 0.07	0.69 ± 0.04	0.67 ± 0.05	0.63 ± 0.03

Table 2: Mean average precision for taxonomy expansion on WordNet mammals and synthetic data

into account the local geometry of the embedding is indeed beneficial in estimating the relative position of novel points in the space.

We conclude by noting that all hyperbolic-based methods have comparable performance across the three settings. However, we point out that HSP and NN-G offer significant practical advantages over HNN: in all our experiments they were faster to train and in general more amenable do model design. In particular, since HSP is based on a kernel method, it has relatively fewer hyperparameters and requires only solving a linear system at training time. NN-G consists of a standard neural architecture with the homeomorphism activation function introduced in Section 3 and trained with the geodesic loss. This allows one to leverage all current packages available to train neural networks, significantly reducing both modeling and training times.

5 Conclusion

In this paper, we showed how to recast supervised problems with hierarchical structure as manifold-valued regressions in the hyperbolic manifold. We then proposed two algorithms for learning manifold-valued functions mapping from Euclidean to hyperbolic space: a non-parametric kernel-based method for which we also proved generalization bounds and a parametric deep-learning model that is informed by the geodesics of the output space. The latter makes possible to leverage traditional neural network layers for regression on hyperbolic space without resorting to hyperbolic layers, thus requiring a smaller training time. We evaluated both methods empirically on the task of hierarchical classification and showed that hyperbolic structured prediction shows strong generalization performance. We also showed that hyperbolic manifold regression enables new applications in supervised learning. By exploiting the continuous representation of hierarchies in hyperbolic space we were able to place unknown concepts in the embedding of a taxonomy using manifold regression. Moreover, by comparing to hyperbolic neural networks we showed that for this application, the key step is leveraging the geodesic of the manifold. In this work, we have aimed at developing a foundation for regressing onto hyperbolic representations. In future work, we plan to exploit this framework in dedicated methods for hierarchical machine learning and extending the applications to manifold product spaces.

Acknowledgments

We thank Maximilian Nickel for his invaluable support and feedback throughout this project. Without him, this paper would have not been possible.

This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216, and the Italian Institute of Tech-

nology. We gratefully acknowledge the support of NVIDIA Corporation for the donation of the Titan Xp GPUs and the Tesla k40 GPU used for this research. This work has been carried out at the Machine Learning Genoa (MaLGa) center, Università di Genoa (IT) L. R. acknowledges the financial support of the European Research Council (grant SLING 819789), the AFOSR projects FA9550-17-1-0390 and BAA-AFRL-AFOSR-2016-0007 (European Office of Aerospace Research and Development), and the EU H2020-MSCA-RISE project NoMADS - DLV-777826.

References

G. Bakir, T. Hofmann, B. Schölkopf, A. Smola, B. Taskar, and S. Vishwanathan. Predicting structured data. neural information processing, 2007.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35 (8):1798–1828, 2013.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

S. Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Trans. Automat. Contr.*, 58(9):2217–2229, 2013. doi: 10.1109/TAC.2013.2254619. URL `http://dx.doi.org/10.1109/TAC.2013.2254619`.

B. P. Chamberlain, S. R. Hardwick, D. R. Wardrope, F. Dzogang, F. Daolio, and S. Vargas. Scalable hyperbolic recommender systems. *arXiv preprint arXiv:1902.08648*, 2019.

F. Chollet. Information-theoretical label embeddings for large-scale image classification. *arXiv preprint arXiv:1607.05691*, 2016.

T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

C. De Sa, A. Gu, C. Ré, and F. Sala. Representation Tradeoffs for Hyperbolic Embeddings. In *International Conference on Machine Learning*, pages 4460–4469, jul 2018. URL `http://proceedings.mlr.press/v80/sala18a.htmlhttp://arxiv.org/abs/1804.03329`.

L. Falorsi, P. de Haan, T. R. Davidson, N. De Cao, M. Weiler, P. Forré, and T. S. Cohen. Explorations in homeomorphic variational auto-encoding. *arXiv preprint arXiv:1807.04689*, 2018.

O.-E. Ganea, G. Bécigneul, and T. Hofmann. Hyperbolic Neural Networks. In *Advances in neural information processing systems*, pages 5345–5355, 2018. URL `https://papers.nips.cc/paper/7780-hyperbolic-neural-networks`.

M. Hamann. On the tree-likeness of hyperbolic spaces. *Mathematical Proceedings of the Cambridge Philosophical Society*, 164(2):345–361, 2018. doi: 10.1017/S0305004117000238.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

E. Hebey. *Nonlinear analysis on manifolds: Sobolev spaces and inequalities*, volume 5. American Mathematical Soc., 2000.

S. Jean, K. Cho, R. Memisevic, and Y. Bengio. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*, 2014.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky. Hyperbolic image embeddings. *arXiv preprint arXiv:1904.02239*, 2019.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

M. Le, S. Roller, L. Papaxanthos, D. Kiela, and M. Nickel. Inferring concept hierarchies from text corpora via hyperbolic embeddings. *arXiv preprint arXiv:1902.00913*, 2019.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

E. Loper and S. Bird. NLTK. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics* -, volume 1, pages 63–70, Morristown, NJ, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118108.1118117. URL `http://portal.acm.org/citation.cfm?doid=1118108.1118117`.

G. J. Martin. Balls in hyperbolic manifolds. *Journal of the London mathematical society*, 2(2):257–264, 1989.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013. URL `http://arxiv.org/abs/1310.4546`.

A. Naik and H. Rangwala. *Large Scale Hierarchical Classi cation : State of the Art*. 2018. ISBN 9783030016197.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

M. Nickel and D. Kiela. Poincaré Embeddings for Learning Hierarchical Representations. In *Neural Information Processing Systems Proceedings*, pages 6338–6347, 2017.

M. Nickel and D. Kiela. Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. In *International Conference on Machine Learning*, jul 2018. URL `http://proceedings.mlr.press/v80/nickel18a.htmlhttp://arxiv.org/abs/1806.03417`.

M. Nickel, V. Tresp, and H. Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*, pages 809–816, 2011.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011. ISSN ISSN 1533-7928. URL `http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html`.

B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

A. Rudi, C. Ciliberto, G. M. Marconi, and L. Rosasco. Manifold Structured Prediction. (Nips):1–18, 2018. URL `http://arxiv.org/abs/1806.09908`.

L. Song, K. Fukumizu, and A. Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013.

F. Steinke and M. Hein. Non-parametric regression between manifolds. In *Advances in Neural Information Processing Systems*, pages 1561–1568, 2009.

F. Steinke, M. Hein, and B. Schölkopf. Nonparametric regression between general riemannian manifolds. *SIAM Journal on Imaging Sciences*, 3(3):527–563, 2010.

A. Tifrea, G. Bécigneul, and O.-E. Ganea. Poincar\'e glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018.

P. Vateekul, M. Kubat, and K. Sarinnapakorn. Top-down optimized svms for hierarchical multi-label classification: A case study in gene function prediction. *Intelligent Data Analysis*, 2012.

A. Veit, M. Nickel, S. Belongie, and L. van der Maaten. Separating self-expression and visual content in hashtag supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5919–5927, 2018.

R. C. Wilson, E. R. Hancock, E. Pekalska, and R. P. Duin. Spherical and hyperbolic embeddings of data. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2255–2269, 2014.

T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1249–1258, 2016.

J. Xu and G. Durrett. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4503–4513, 2018.