Unsupervised Hyperbolic Representation Learning via Message Passing Auto-Encoders

Jiwoong Park\textsuperscript{1}Junho Cho\textsuperscript{1}Hyung Jin Chang\textsuperscript{2}Jin Young Choi\textsuperscript{1}

\textsuperscript{1}ASRI, Dept. of ECE., Seoul National University \textsuperscript{2}School of Computer Science, University of Birmingham

\{ptywoong,junhocho,jychoi\}@snu.ac.kr, h.j.chang@bham.ac.uk

Abstract

Most of the existing literature regarding hyperbolic embedding concentrate upon supervised learning, whereas the use of unsupervised hyperbolic embedding is less well explored. In this paper, we analyze how unsupervised tasks can benefit from learned representations in hyperbolic space. To explore how well the hierarchical structure of unlabeled data can be represented in hyperbolic spaces, we design a novel hyperbolic message passing auto-encoder whose overall auto-encoding is performed in hyperbolic space. The proposed model conducts auto-encoding the networks via fully utilizing hyperbolic geometry in message passing. Through extensive quantitative and qualitative analyses, we validate the properties and benefits of the unsupervised hyperbolic representations. Codes are available at https://github.com/junhocho/HGCAE.

1. Introduction

A fundamental problem of machine learning is learning useful representations from high-dimensional data. There are many supervised representation learning methods that achieve good performances for downstream tasks \cite{25, 22, 30, 60} on several data domains such as images and graphs. In recent years, with the success of deep learning, various large-scale real-world datasets have been collated \cite{25, 24, 56, 49}. However, the larger these datasets and the closer they are to the real world, the expense and effort required to label the data increases proportionally. Thus, unsupervised representation learning is an increasingly viable approach to extract useful representation from real-world datasets.

Recently, many works \cite{39, 40, 11, 16, 5, 1, 19} utilize hyperbolic geometry \cite{23} to learn representations by understanding the underlying nature of the data domains. It is well known that complex networks contain latent hierarchies between large groups and the divided subgroups of nodes and can be approximated as trees that grow exponentially with their depth \cite{23}. Based on this fact, previous works which involve graphs \cite{3, 38, 39, 40, 11, 36} showed the effectiveness of learning representation using hyperbolic spaces (a continuous version of trees) where distances increase exponentially when moving away from the origin. More recently, works \cite{5, 27, 1} have been conducted which learn more powerful representations via conducting message passing (graph convolution) \cite{12, 22, 54} in hyperbolic spaces.

In addition, it has been successfully shown that grafting hyperbolic geometry onto computer vision tasks is promising \cite{19}. They observed a high degree of hyperbolicity \cite{10} in the activations of image datasets obtained from pre-trained convolutional networks. Also, it has been shown that the hyperbolic distance between learned embeddings and the origin of the Poincaré ball could be considered as a measurement of the model’s confidence. Using these analyses, \cite{19} added a single layer of hyperbolic neural networks \cite{11} to deep convolutional networks and showed the benefits of hyperbolic embeddings on few-shot learning and person re-identification. Another work \cite{28} also demonstrated the suitability of hyperbolic embeddings on zero-shot learning. However, most of the existing hyperbolic representation learning works \cite{19, 28, 5, 27, 1} mainly focus on a supervised setting, and the effect of hyperbolic geometry on unsupervised representation learning has not been explored deeply so far \cite{32, 15, 36}.

In this paper, we explore the benefits of hyperbolic geometry to carry out unsupervised representation learning upon various data domains. Our motivation is to learn high-quality node embeddings of the graphs that are hierarchical and tree-like without supervision via considering the geometry of the embedding space. To do so, we present a novel hyperbolic graph convolutional auto-encoder (HGCAE) by combining hyperbolic geometry and message passing \cite{12}. Every layer of HGCAE performs message passing in the hyperbolic space and its corresponding tangent space where curvature values can be trained. This is primarily in contrast to the Poincaré variational auto-encoder (P-VAE) \cite{32} whose latent space is the Poincaré ball and conducts mes-
sage passing in Euclidean space. The HGCAE conducts auto-encoding the graphs from diverse data domains, such as images or social networks, in the hyperbolic space such as the Poincaré ball and hyperboloid. To fully utilize hyperbolic geometry for representation learning, we adopt a geometry-aware attention mechanism [16] when conducting message passing. Through extensive experiments and analyses using the learned representation in the hyperbolic latent spaces, we present the following observations on hierarchically structured data:

- The proposed auto-encoder, which combines message passing based on geometry-aware attention and hyperbolic spaces, can learn useful representations for downstream tasks. On various networks, the proposed method achieves state-of-the-art results on node clustering and link prediction tasks.

- Image clustering tasks can benefit from embeddings in hyperbolic latent spaces. We achieve comparable results to state-of-the-art image clustering results by learning representations from the activations of neural networks.

- Hyperbolic embeddings of images, the results of unsupervised learning, can recognize the underlying data structures such as a class hierarchy without any supervision of ground-truth class hierarchy.

- We show that the sample’s hyperbolic distance from the origin in hyperbolic space can be utilized as a criterion to choose samples, therefore improving the generalization ability of a model for a given dataset.

2. Related Works

Hyperbolic embedding of images. Khrulkov et al. [19] validated hyperbolic embeddings of images via measuring the degree of hyperbolicity of image datasets. Many datasets such as CIFAR10/100 [24], CUB [56] and MiniImageNet [45] showed high degrees of hyperbolicity. In particular, the ImageNet dataset [47] is organized by following the hierarchical structure of WordNet [35]. These observations suggest that hyperbolic geometry can be beneficial in analyzing image manifolds by capturing not only semantic similarities but also hierarchical relationships between images. Furthermore, Khrulkov et al. [19] empirically showed that the distance between the origin and the image embeddings in the Poincaré ball can be regarded as the measure of the model’s confidence. They observed that the samples which are easily classified are located near the boundary, while those more ambiguous samples lie near the origin of the hyperbolic space. Recent works of hyperbolic image embeddings [19, 28] add one or two layers of hyperbolic layers [11] after Euclidean convolutional networks.

Graph auto-encoding via hyperbolic geometry. Some recent works [15, 32, 51] attempted to auto-encode graphs in hyperbolic space. Their models attempted to learn latent representations in the hyperbolic space via grafting hyperbolic geometry onto a variational auto-encoder model [20]. [15, 32] encoded the node representation via message passing [22] in Euclidean space, then the encoded representation was projected onto the hyperbolic space. Similar to these concurrent models, our auto-encoder framework learns latent node representations of the graph in hyperbolic latent spaces. Differing from these models, our work considers hyperbolic geometry throughout the auto-encoding process. Each encoder and decoder layer of the proposed model conducts message passing by utilizing geometry-aware attention in the hyperbolic space and its tangent space.

3. Hyperbolic Geometry

A real, smooth manifold \( M \) is a set of points \( x \), that is locally similar to linear space. At each point \( x \in M \), the tangent space at \( x \), \( T_xM \), is a real vector space whose dimensionality is same as \( M \). A Riemannian manifold is defined as a tuple \((M, g)\) that is possessing metric tensor \( g_x : T_xM \times T_xM \rightarrow \mathbb{R} \). The metric tensor provides geometric notions such as geodesic, angle and volume. There exist mapping between the manifold and the tangent space: exponential map and logarithmic map. The exponential map \( \exp_x : T_xM \rightarrow M \) projects the vector on the tangent space \( T_xM \) back to the manifold \( M \), while the logarithmic map \( \log_x : M \rightarrow T_xM \) is the inverse mapping of the exponential map as \( \log_x(\exp_x(v)) = v \).

The hyperbolic space is a Riemannian manifold with constant negative sectional curvature equipped with hyperbolic geometry. This paper deals with two hyperbolic spaces: ‘Poincaré ball’ and ‘hyperboloid’. The Poincaré ball \( \mathbb{B} \) is highly effective for visualizing and analyzing the hyperbolic latent space. Meanwhile, the hyperboloid \( \mathbb{H} \) can provide stable optimization since, unlike distance function of Poincaré ball, there is no division in the distance function [40]. A review of Riemannian geometry and details of the hyperboloid model are presented in the supplementary material.

Poincaré ball. The \( n \)-dimensional Poincaré ball with constant negative curvature \( K(K < 0) \) \( (\mathbb{B}_n^K, g^K_x) \) is defined:

\[
\mathbb{B}_n^K = \{x \in \mathbb{R}^n : \|x\|^2 < -1/K\},
\]

where \( \| \cdot \| \) denotes Euclidean norm. The metric tensor is \( g^K_x = (\lambda^K_x)\|g^K_x\| \), where \( \lambda^K_x = \frac{2}{1+K\|x\|^2} \) is the conformal factor and \( g^K_x = \text{diag}([1, 1, \ldots, 1]) \) denotes Euclidean metric tensor. The origin of \( \mathbb{B}_n^K \) is \( o = (0, \ldots, 0) \in \mathbb{R}^n \). The distance between two points \( x, y \in \mathbb{B}_n^K \) is defined as

\[
d_{\mathbb{B}^K}(x, y) = \frac{1}{\sqrt{-K}} \arccosh \left( 1 - \frac{2K\|x - y\|^2}{(1 + K\|x\|^2)(1 + K\|y\|^2)} \right).
\]
Figure 1: The overall architecture of HGCAE in a two-layer auto-encoder (i.e., the encoder and decoder have two layers each) whose hyperbolic space is hyperboloid. This figure describes three things: 1) how the node of the graph (red dot) conducts message passing (Eq. (8) and (11)) with its neighbors (yellow dot), 2) the process of embedding the output of encoder in hyperboloid latent space (blue-purple space), and 3) reconstruction of Euclidean node attributes at the end of the decoder.

For points \( x \in \mathbb{P}_K^n \), tangent vector \( v \in T_x \mathbb{P}_K^n \), and \( y \neq 0 \), the exponential map \( \exp_x : \mathbb{P}_K^n \rightarrow \mathbb{P}_K^n \) and the logarithmic map \( \log_x : \mathbb{P}_K^n \rightarrow T_x \mathbb{P}_K^n \) are defined as:

\[
\exp^K_x(v) = x \oplus_K \left( \tanh(\frac{-K\lambda}{2}) \frac{v}{\sqrt{-K|v|}} \right), \quad (3)
\]

\[
\log^K_x(y) = \frac{2}{\sqrt{-K\lambda}} \arctanh \left( \frac{y}{\sqrt{-K|y|}} \right) \left\| \frac{u}{\|u\|} \right\|, \quad (4)
\]

where \( u = -x \oplus_K y \) and \( \oplus_K \) denotes Möbius addition [53] for \( x, y \in \mathbb{P}_K^n \) as

\[
x \oplus_K y = \frac{(1 - 2K\langle x, y \rangle - K|y|^2)x + (1 + K|x|^2)y}{1 - 2K\langle x, y \rangle + K^2|x|^2|y|^2}. \quad (5)
\]

**Mapping between two models.** Two hyperbolic models, Poincaré ball and hyperboloid, are equivalent and transformations between two models retain many geometric properties including isometry. There exist diffeomorphisms \( p_{\mathbb{P}^{n}_K \rightarrow \mathbb{P}^{n}_K} \) and \( p_{\mathbb{P}^{n}_K \rightarrow \mathbb{H}^{n,1}_K} \) between the two models, Poincaré ball \( \mathbb{P}^{n}_K \) and hyperboloid \( \mathbb{H}^{n,1}_K \) [27, 5], as follows:

\[
p_{\mathbb{P}^{n}_K \rightarrow \mathbb{P}^{n}_K}(x_0, x_1, \ldots, x_n) = \left( x_1, \ldots, x_n \right) \sqrt{\frac{1}{K} |x_0| + 1} \quad (6)
\]

\[
p_{\mathbb{P}^{n}_K \rightarrow \mathbb{H}^{n,1}_K}(x_1, \ldots, x_n) = \left( 1 - K|x|^2, 2x_1, \ldots, 2x_n \right) \frac{1}{1 + K|x|^2}. \quad (7)
\]

**4. Methodology**

HGCAE is designed to fully utilize hyperbolic geometry in the auto-encoding process along with leveraging the power of graph convolutions via a geometry-aware attention mechanism. Each layer conducts message passing in hyperbolic space whose curvature value is trainable. Before conducting message passing, we need to map the given input data points, \( x^{Euc} \), defined in Euclidean space to the hyperbolic manifold. We map the Euclidean feature into hyperbolic manifold via \( h^1_i = \exp^K_{o_i}(x^{Euc}_i) \), where \( K_1 \) and \( h^1_i \) denote a trainable curvature value and the \( i \)-th node’s representation of the first layer respectively. When the hyperbolic space is hyperboloid model, we use \( (0, x^{Euc}) \in \mathbb{R}^{n+1} \) as an input of an exponential map as [5] did. The overall architecture of HGCAE is presented in Fig. 1.

**4.1. Geometry-Aware Message Passing**

**Linear transformation.** Message passing in the HGCAE consists of two steps: the linear transformation of a message and aggregating messages from neighbors. The \( i \)-th node’s message passing result at the \( l \)-th layer \( z^l_i \) is as follows:

\[
z^l_i = \exp^K_{o_i} \left( \sum_{j \in \mathcal{N}(i)} \alpha^l_{ij} \left( W^l \log^K_{o_j}(h^l_j) + b^l_j \right) \right), \quad (8)
\]

where \( W^l, b^l, \mathcal{N}(i), \) and \( \alpha^l_{ij} \) denote a weight matrix, a bias term, the set of direct neighbors of node \( i \) including itself, and the relative importance (attention score) of the neighbor node \( j \) to the node \( i \) at the \( l \)-th layer respectively. Based on [11], we map the points in the hyperbolic manifold to the tangent space via the logarithmic map since the linear transformation cannot be performed directly in hyperbolic spaces. Then, the messages are linearly transformed on the tangent space of the origin in which inherits many properties of the ambient Euclidean space.

**Aggregation.** After performing linear transformation, we aggregate messages from neighbors via an attention mechanism. The majority of message passing algorithms that use attention mechanisms learn the relative importance of each node’s neighbors based on node feature not only in Euclidean space [54] but also in hyperbolic space [5]. However, only considering node features for learning their relative importance does not take into account the geometry of the space, and this might result in an imprecise attention score. To make full use of the Riemannian metric of the hyperbolic manifolds, we adopt a geometry-aware attention mechanism [16] by utilizing the distance between linearly transformed node features on the hyperbolic space.

Let \( y^l_i = W^l \log^K_{o}(h^l_i) + b^l_i \), then the attention score at the
4.2. Nonlinear Activation

The nonlinear activations, \( \sigma \), such as ReLU can be directly applied to the points in the Poincaré ball, in contrast to the points on the hyperboloid \([27]\). Thus, when the hyperboloid model is used, we map the points to the Poincaré ball using Eq. (6) first. Next, we apply the nonlinear activation in the Poincaré ball and then return the result to the hyperboloid using the Eq. (7).

Since the curvature value of each layer in HGCAE is trainable, each layer can have different curvature values from other layers. Thus, a step for locating the result of the nonlinear activation in the hyperbolic space having a curvature value of the next layer is required. First, we map the results of the nonlinear activation to the tangent space of the current layer, \( \mathbb{T}_o \mathcal{M}_{K_l} \), using logarithmic map, \( \log_{K_l} \). Next, the points in the tangent space are mapped to the next layer’s hyperbolic space via an exponential map of the next layer \( \exp^{K_{l+1}} \). The equations for performing such nonlinear activation and mapping to the hyperbolic space of the next layer in the cases of Poincaré ball and hyperboloid are as follows respectively:

\[
h_{l+1}^i = \exp_{K_{l+1}}^{( \log_{K_l} (\sigma(z^i)))},
\]

\[
h_{l+1}^i = \exp_{K_{l+1}}^{( \log_{K_l} (p_{P \to H}(\sigma(p_{E \to P}(z^i)))))}.
\]

4.3. Loss Function

Our HGCAE reconstructs both the affinity matrix (graph structure) \( A \) and the Euclidean node attributes \( X^{Euc} \) at the end of the encoder and the decoder, respectively. To reconstruct the Euclidean node attributes \( X^{Euc} \), the aggregated representations in the hyperbolic space of the decoder’s last layer are mapped to the tangent space of the origin \( \mathbb{T}_o \mathcal{M} \). Then, the loss of representations \( \mathcal{L}_{REC−X} \) is defined as the mean square error between \( X^{Euc} \) and \( \hat{X}^{Euc} \),

\[
\|X^{Euc} - \hat{X}^{Euc}\|_2^2.
\]

For reconstructing the structure of the graph, the hyperbolic distance between the latent representations (the output of the encoder) of two nodes is utilized. To calculate the probability score of an edge which links between two nodes, we adopt the Fermi-Dirac distribution \([23, 39]\),

\[
\hat{A}_{ij} = \left[e^{d^2_{\mathcal{M}_{K_l}}(h_i, h_j)/t} + 1\right]^{-1},
\]

where \( d_{\mathcal{M}_{K_l}}(\cdot, \cdot) \) and \( t \) denote the hyperbolic distance and the temperature, respectively.

The results of link prediction and node clustering are presented in Tables 2 and 3 respectively. From the results, we see that HGCAE-P outperforms other models in terms of P@K and hits@K metrics, which indicates its effectiveness in capturing the hyperbolic properties of real-world datasets.

The reconstructed affinity matrix, and hyperparameters respectively. The loss function for the affinity matrix is defined by the cross entropy loss with negative sampling:

\[
\mathcal{L}_{REC−A} = - \sum_{e \in E} \log p(A[e]),
\]

where \( p(A[e]) \) is the probability score of an edge which links between two nodes, \( e \), and \( A \) is the reconstructed affinity matrix.

The overall loss function of HGCAE is

\[
\mathcal{L} = \mathcal{L}_{REC−A} + \lambda \mathcal{L}_{REC−X},
\]

where \( \lambda \) is a regularizer for the relative importance between the attributes and structure.

5. Experiments

This section explores the effectiveness of unsupervised hyperbolic embeddings on various data domains via quantitative and qualitative analyses. We use 9 real-world complex network datasets and 3 image datasets. The statistics of the datasets are summarized in Table 1. The details of the datasets, the compared methods, and the experimental details are described in the supplementary material. For node clustering and link prediction tasks on the 9 network datasets, we evaluate HGCAE-P and HGCAE-H, which denote HGCAE models whose latent spaces are Poincaré ball and hyperboloid, respectively. For the tasks of image clustering and visual data analysis, we use HGCAE-P because Poincaré ball is a powerful tool for visualizing and analyzing properties of hyperbolic visual embeddings.
Table 2: Link prediction performances.

<table>
<thead>
<tr>
<th></th>
<th>Cora</th>
<th>Citeseer</th>
<th>Wiki</th>
<th>Pubmed</th>
<th>BlogCatalog</th>
<th>Amazon Photo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>AP</td>
<td>AUC</td>
<td>AP</td>
<td>AUC</td>
<td>AP</td>
</tr>
<tr>
<td>GAE [21]</td>
<td>0.910</td>
<td>0.920</td>
<td>0.895</td>
<td>0.899</td>
<td>0.930</td>
<td>0.948</td>
</tr>
<tr>
<td>VGAE [21]</td>
<td>0.914</td>
<td>0.926</td>
<td>0.908</td>
<td>0.920</td>
<td>0.936</td>
<td>0.950</td>
</tr>
<tr>
<td>ARGA [41]</td>
<td>0.924</td>
<td>0.932</td>
<td>0.919</td>
<td>0.930</td>
<td>0.934</td>
<td>0.947</td>
</tr>
<tr>
<td>ARVGA [41]</td>
<td>0.924</td>
<td>0.926</td>
<td>0.924</td>
<td>0.930</td>
<td>0.947</td>
<td>0.948</td>
</tr>
<tr>
<td>GALA [42]</td>
<td>0.929</td>
<td>0.937</td>
<td>0.944</td>
<td>0.948</td>
<td>0.936</td>
<td>0.931</td>
</tr>
<tr>
<td>DBGAN [66]</td>
<td>0.945</td>
<td>0.951</td>
<td>0.945</td>
<td>0.958</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HGCAE-P</td>
<td>0.956</td>
<td>0.957</td>
<td>0.960</td>
<td>0.963</td>
<td>0.955</td>
<td>0.956</td>
</tr>
<tr>
<td>HGCAE-H</td>
<td>0.956</td>
<td>0.957</td>
<td>0.970</td>
<td></td>
<td>0.962</td>
<td>0.962</td>
</tr>
</tbody>
</table>

Table 3: Node clustering performances.

<table>
<thead>
<tr>
<th></th>
<th>Cora</th>
<th>Citeseer</th>
<th>Wiki</th>
<th>Pubmed</th>
<th>BlogCatalog</th>
<th>Amazon Photo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACC</td>
<td>NMI</td>
<td>ACC</td>
<td>NMI</td>
<td>ACC</td>
<td>NMI</td>
</tr>
<tr>
<td>Kmeans [29]</td>
<td>0.492</td>
<td>0.321</td>
<td>0.540</td>
<td>0.305</td>
<td>0.417</td>
<td>0.440</td>
</tr>
<tr>
<td>GAE [21]</td>
<td>0.532</td>
<td>0.434</td>
<td>0.505</td>
<td>0.246</td>
<td>0.460</td>
<td>0.468</td>
</tr>
<tr>
<td>MGAE [57]</td>
<td>0.684</td>
<td>0.511</td>
<td>0.660</td>
<td>0.412</td>
<td>0.514</td>
<td>0.485</td>
</tr>
<tr>
<td>ARGA [41]</td>
<td>0.640</td>
<td>0.449</td>
<td>0.573</td>
<td>0.350</td>
<td>0.458</td>
<td>0.437</td>
</tr>
<tr>
<td>ARVGA [41]</td>
<td>0.638</td>
<td>0.450</td>
<td>0.544</td>
<td>0.261</td>
<td>0.386</td>
<td>0.338</td>
</tr>
<tr>
<td>GALA [42]</td>
<td>0.745</td>
<td>0.576</td>
<td>0.693</td>
<td>0.441</td>
<td>0.544</td>
<td>0.503</td>
</tr>
<tr>
<td>DBGAN [66]</td>
<td>0.748</td>
<td>0.560</td>
<td>0.670</td>
<td>0.407</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HGCAE-P</td>
<td>0.746</td>
<td>0.572</td>
<td>0.693</td>
<td>0.422</td>
<td>0.459</td>
<td>0.467</td>
</tr>
<tr>
<td>HGCAE-H</td>
<td>0.767</td>
<td>0.599</td>
<td>0.715</td>
<td>0.453</td>
<td>0.530</td>
<td>0.435</td>
</tr>
</tbody>
</table>

Table 4: Link prediction task compared with P-VAE.

<table>
<thead>
<tr>
<th>Phylogenetic CS PhDs Diseases</th>
<th>AUC</th>
<th>AP</th>
<th>AUC</th>
<th>AP</th>
<th>AUC</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vgae [21]</td>
<td>0.542</td>
<td>0.540</td>
<td>0.565</td>
<td>0.564</td>
<td>0.898</td>
<td>0.918</td>
</tr>
<tr>
<td>P-VAE [32]</td>
<td>0.590</td>
<td>0.555</td>
<td>0.598</td>
<td>0.567</td>
<td>0.923</td>
<td>0.936</td>
</tr>
<tr>
<td>HGCAE-P</td>
<td>0.688</td>
<td>0.712</td>
<td>0.673</td>
<td>0.640</td>
<td>0.926</td>
<td>0.914</td>
</tr>
</tbody>
</table>

Figure 2: 2-dimensional embeddings in Euclidean, Poincaré ball, and hyperboloid latent space on Cora dataset. Same color indicates same class. On hyperbolic latent spaces, most of the nodes are located on the boundary and well-clustered with the nodes in the same class.

5.2. Image Clustering

In this experiment, we illustrate that image clustering can benefit from hyperbolic geometry. The training sets of ImageNet-10 and ImageNet-Dogs [7], which are subsets of ImageNet [25], are used for evaluation. In the manner of the researches [11, 16, 19] which impose hyperbolic geometry on the activations of neural networks, we used the activations of PICA [18], one of the most recent models developed for deep image clustering. After obtaining activations from the pre-trained networks of PICA, we built the graph by mutual $k$ nearest neighbors between activations. Then,
both the activations and the graph were used as inputs of HGCAE-P. Extensive baselines and state-of-the-art image clustering methods [29, 64, 14, 4, 2, 31, 37, 55, 44, 63, 65, 20, 62, 59, 7, 6, 58, 18] were compared. Furthermore, we also trained two auto-encoder models, GAE [21], and hyperbolic auto-encoder (HAE) whose layers are hyperbolic feed-forward layers [11]. The image clustering results are reported in Table 5. The metrics, ACC, NMI and Adjusted Rand Index (ARI), were used for evaluation. The results demonstrate that applying hyperbolic geometry along with using additional information of the approximated image manifold via nearest neighbor graphs can achieve better results than the Euclidean counterparts. We can also observe that HAE, the auto-encoder which naively applies hyperbolic geometry, does not work well, while our model performs better via the message passing fully utilizing hyperbolic geometry.

### 5.3. Structure-Aware Unsupervised Embeddings

In this experiment, we observe the unsupervised hyperbolic image embeddings’ ability to recognize the latent structure of visual datasets that have hierarchical structures. ImageNet [25] is constructed following the hierarchy of WordNet [35], therefore, its classes of ImageNet-10 [7] also have hierarchical structures. However, it is difficult to explore the effectiveness of hyperbolic embeddings since the classes of ImageNet-10 are biased to a certain root. Thus, we have constructed a new dataset, ImageNet-BNCR, that has a Balanced Number of Classes across Roots. For ImageNet-BNCR, we have chosen three roots, Artifact, Natural objects, and Animal, which have a large number of leaf classes. Each root contains balanced child nodes of \{Ambulance, Dogsled, School bus\}, \{Lemon, Jackfruit, Granny Smith\}, and \{Flamingo, Bald eagle, Lionfish\}, respectively. On the leaf classes of ImageNet-10, \{Container ship, Airliner, Airship, Sports car, Trailer truck, Soccer ball\}, \{Orange\}, and \{Maltese dog, Snow leopard, King penguin\} are the child nodes of the roots Artifact, Natural objects, and Animal, respectively. The class hierarchies of ImageNet-10 and ImageNet-BNCR are shown in Fig. 3.

We extracted 1000-dimensional features by training a convolutional auto-encoder (CAE) [31] on the ImageNet-10 and ImageNet-BNCR datasets. Then, after building the graph using mutual k nearest neighbors between extracted features, we trained three auto-encoder models (HGCAE-P, GAE [21], and HAE) whose latent space is 2-dimensional without the ground truth hierarchy structure of labels. The embedding results of the 1000-dimensional CAE features via UMAP [34] and three auto-encoders are presented in Fig. 4. We can observe that the embeddings of HGCAE-P are better clustered than others, according to the classes of each root in Fig. 3. On the ImageNet-10, in the same root Artifact, the embeddings of descendants of Craft and Wheeled vehicle are clustered respectively. The embeddings of the ImageNet-BNCR are clustered more distinctly ac-

---

1. http://image-net.org/index
According to the root of class hierarchy than with ImageNet-10. On the other hand, the embeddings of the root Natural objects, \{Lemon, Jackfruit, Granny Smith\}, are located closer to each other since the geodesic distance between each leaf label is small. Our distinction from HAE implies that the additional information on image manifolds approximated by nearest neighbor graphs is helpful. In contrast to the representations of CAE and GAE, we can see that the hyperbolic representations belonging to the same root are located near the boundary of the space. In addition, to quantitatively validate the ability to recognize the latent hierarchical structure of the data without direct learning of label hierarchy, we cluster 2-dimensional embeddings of the three auto-encoders with three ground truth label settings according to the class hierarchy in Fig. 3: I. Root

5.4. Hyperbolic Distance to Filter Training Samples

In this experiment, we show that hyperbolic distance can help to choose training samples beneficial to the generalization ability of neural networks. To this end, we obtained the latent embeddings of ImageNet-10 \([7]\) and ImageNet-BNCR via HGCAE-P model. Then, the hyperbolic distance (Eq. (2)) of each embedding from the origin was computed. Fig. 6 shows some samples near the boundary or near the origin in the histogram of the hyperbolic distance from embeddings to the origin. We can see that the samples near the boundary can be easily classified, whereas those near the origin are harder to classify. In general, the easy samples are not influential to learn an exact decision boundary. On the other hand, the hard samples make the decision boundary over-fitted, i.e., they work like noises located at the soft margin region near the decision boundary \([8]\). This illustration intuitively shows that the Hyperbolic Distance from the Origin (HDO) of a sample could give a clue which samples are influential or beneficial to learn the decision boundary crucial for the generalization ability of a classifier.

To verify this intuition, we conducted an experiment on the image classification task. On ImageNet-10 and ImageNet-BNCR, we trained the VGG-11 \([50]\) classifier by adding further samples near the boundary/median of the dis-
Figure 6: Histogram and images according to the hyperbolic distance from the origin (HDO) on ImageNet-10 and ImageNet-BNCR. The feature of images inside red (blue) color box have high (low) HDO, so are located near the boundary (origin) of hyperbolic space.

Figure 7: Top-1 classification error (%) on ImageNet-10 and ImageNet-BNCR.

6. Conclusion

In this paper, we explored the properties of unsupervised hyperbolic representations. We derived the representations from geometry-aware message passing auto-encoders whose whole operations were conducted in hyperbolic spaces. Then, we conducted extensive experiments and analyses on the low-dimensional latent representations in hyperbolic spaces. The experimental results support the conclusion that taking advantage of hyperbolic geometry can improve the performances of unsupervised tasks; node clustering, link prediction, and image clustering. We observed that the proposed method could yield unsupervised hyperbolic image embeddings reflecting the latent structure of the visual datasets that have a hierarchical structure. Lastly, we demonstrated that the hyperbolic distance from origin for a sample could be utilized to determine the additional data crucial for a classifier’s generalisation ability.

Acknowledgement: We thank Esha Dasgupta for careful reading and insightful comments on our manuscript. This work was supported by Ministry of Science and ICT, Korea: IITP grant [No.2014-3-00123, Development of High Performance Visual BigData Discovery Platform] and ITRC support program [IITP-2020-2020-0-01789] supervised by the IITP.
References


[4] Deng Cai, Xiaofei He, Xuanhui Wang, Hujun Bao, and Jiawei Han. Locality preserving nonnegative matrix factorization. In IJCAI, volume 9, pages 1010–1015, 2009.


[31] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hi-
erarchical feature extraction. In International Conference on Artificial Neural Networks, pages 52–59. Springer, 2011. 6


[43] Peter Petersen, S Axler, and KA Ribet. Riemannian geometry, volume 171. Springer, 2006. 2


[61] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Chang. Network representation learning with rich text
information. In Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015. 4


