Hyperbolic Embedding of Attributed Networks

David McDonald[1] , **Shan He**[2]
[1]University of Birmingham
[2]University of Birmingham
{dxm237, s.he}@cs.bham.ac.uk

Abstract

TODO

1 Introduction

Throughout our world, we observe complex systems – groups of *elements* that connect to each other by *relations* in a non-uniform way. Through these relations, these elements are able to work together and function as a coherent whole that is greater than the sum of its parts. We see this in the simple relationships amongst people that form an entire society; in the iterations between genes, proteins and metabolites that form a living organism; and in the links between pages that make up the internet. Within these systems, interactions are not controlled globally, but emerge locally based on some local organisation that gives rise to new levels of organisation. In this way, we see that the organisation of complex systems is *hierarchical*: elements belong to many different systems on many different scales, with all the levels affecting each other [Barabási and Albert, 1999]. In addition to the hierarchical organisation of elements, we observe that entities can be richly annotated with features, that are themselves organised hierarchically. For example, a paper within a citation network may be annotated with the presence of particular key words and the presence of these words may give rise to the presence or absence of higher order (or more abstract) features such as semantics or topic.

The success of machine learning algorithms often depends upon data representation [Bengio *et al.*, 2013]. Representation learning – where we learn alternative representations of data – has become common for processing information on non-Euclidean domains, such as the domain of nodes and edges that comprise these complex systems. Prediction over nodes and edges, for example, requires careful feature engineering [Grover and Leskovec, 2016] and representation learning leads to the extraction of features from a graph that a most useful for downstream tasks, without careful design or a-priori knowledge. In particular, research has shown compelling evidence that an underlying metric space underpins the emergence of behaviour in the network – for example, two elements that appear close together in this metric space are more likely to interact [Grover and Leskovec, 2016; Alanis-Lobato *et al.*, 2016a; Alanis-Lobato *et al.*, 2016b] and furthermore, that the shape of this metric space is, in fact, hyperbolic. Indeed, we can interpret a hyperbolic space is a continuous representation of a discrete tree structure that captures the hierarchical organisation of elements within a complex system [Krioukov *et al.*, 2010].

Here we propose the first ....

2 Hyperbolic Geometry

Everyone is familiar with Euclidean Geometry. This is the geometry of the world in which we live. It lives in a space that is *connected* (the space cannot be broken up as the union of open subsets), *flat* and *isotropic* (the Gaussian curvature of the space is zero for all points in the space)[1]. It also obeys all of the Euclid's postulates that seem "obviously true" in the world that we live in. These consist of axioms such as the existence of a straight line segment between two points, a circle being uniquely described by a centre and a radius, and the *parallel postulate*: that given a line l and a point P not on l, there exists exactly one line through P that is parallel to (does not intersect) l.

Moving beyond our familiar Euclidean geometry, we observe that there are three types of connected, isotropic spaces.

- Euclidean, with gaussian curvature equal to 0,
- Spherical, with strictly positive gaussian curvature,
- Hyperbolic, with strictly negative gaussian curvature.

We shall not focus on Spherical geometry here (however, it will become a useful comparison later). Instead the hyperbolic geometry shall be the focus of this work. Nearly all of Euclid's postulates hold for hyperbolic geometry. All, in fact, except for the *parallel postulate*. In hyperbolic geometry there exists a line l and a point P not on l such that at least two distinct lines parallel to l pass through P.

[1]Gaussian curvature K of a surface S at a point P, we find the normal vector at P, and then define a normal plane as a plane that intersects the surface and contains the normal vector. The intersection of a normal plane and the surface will form a curve called a normal section and the curvature of this curve is the normal curvature. For most points on most surfaces, different normal sections will have different curvatures; the maximum and minimum values of these are called the principal curvatures, call these k_1 and k_2. The Gaussian curvature is the product of the two principal curvatures $K = k_1 k_2$.

2.1 Models of Hyperbolic Geometry

A negative Gaussian curvature K for the entire space implies that every single point in the Hyperbolic space is a saddle point. This makes them a little unintuitive and hard to imagine. Furthermore, they cannot be embedded into a Euclidean space. without distortion. Krioukov et *al*. [Krioukov *et al.*, 2010] informally explain hyperbolic spaces are "larger" and have more "space" than Euclidean spaces. This is mathematically reflected by the fact that the area of a circle of radius r, does not grow quadratically with r, as we are used to in Euclidean space[2], but grows exponentially as $A = 2\pi\cosh(\zeta r - 1)$[3].

Hyperbolic space and trees are very similar. Informally, trees can be thought of as "discrete hyperbolic spaces" and can be embedded into a two dimensional hyperbolic plane without distortion [Krioukov *et al.*, 2010; De Sa *et al.*, 2018].

Because of the fundamental difficulties in representing spaces of constant negative curvature as subsets of Euclidean spaces, there are not one but many equivalent models of hyperbolic spaces. We say the models are equivalent because all models of hyperbolic geometry can be freely mapped to each other by an *isometry* (a distance preserving transformation). Each model emphasizes different aspects of hyperbolic geometry, but no model simultaneously represents all of its properties. We are free to chose the model that best fits our need.

The most popular model in the network embedding literature is the so-called *Poincaré disk* (or *Poincaré ball* for $n > 2$ dimensions.) Here, we have the entire hyperbolic plane (or higher dimensional equivalent) represented as the interior of a unit ball, sitting in a Euclidean ambient space. The boundary of the ball represents infinity in the hyperbolic system that it is modelling. Euclidean and hyperbolic distances, r_e and r_h from the disk centre, or the origin of the hyperbolic plane, are related by

$$r_e = \tanh\left(\frac{r_h}{2}\right)$$

and the shortest path between points (*geodesics*) are given by the diameters of the circle or Euclidean circle arcs that intersect the boundary of the ball perpendicularly. This model has the main advantage that it is *conformal*: the Euclidean angles in the model are equal to the hyperbolic angles in hyperbolic geometry that the model is representing. This makes the model a popular choice for embedding methods that abstract node similarity as the angular distance between them, for example: [Alanis-Lobato *et al.*, 2016a; Alanis-Lobato *et al.*, 2016b].

We have also the much less popular *Klein* model of hyperbolic geometry. This model also represents hyperbolic geometry as a unit disk (or ball) in an ambient Euclidean space. This model preserves straight lines: straight Euclidean lines map to straight hyperbolic lines. However, unlike the *Poincaré ball*, the *Klein* model is not conformal, and the Euclidean angles in the model are not equal to hyperbolic angles.

[2]Recall that the area of a circle is given by $A = \pi r^2$ in Euclidean geometry.

[3]$\zeta = \sqrt{-K}$

2.2 The Hyperboloid Model of Hyperbolic Space

Physicists use a third model of the hyperbolic space. This model is the *Hyperboloid* model, and has direct implications in the study of special relativity. Unlike the aforementioned disk models, a Hyperboloid model of n-dimensional hyperbolic geometry does not sit in an ambient Euclidean space of dimension n, but in $n+1$-dimensional Minkowski spacetime. Furthermore, the set of hyperboloid points to not form a disk in this ambient space, but in n-dimensional hyperboloid. We can actually view both the *Poincaré* and *Klein* models as (stereographic and orthographic) projections of the points from the hyperboloid to disks orthogonal to the main axis of the hyperboloid [Krioukov *et al.*, 2010]. Informally, we can see this relationships as analogous to the relationship between a projected map and a globe [Reynolds, 1993].

$n+1$-dimensional Minkowski spacetime is defined as the combination of n-dimensional Euclidean space with an additional time co-ordinate t. As is common practice, we shall henceforth denote this set $\mathbb{R}_{n:1}$. We say that point $\mathbf{x} \in \mathbb{R}_{n:1}$ has time co-ordinate x_i^0 and spacial coordinates x_i^k for $k = 1, 2, ..., n$.

We define the *Minkowski Bilinear Form* to be

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle_{\mathbb{R}^{n:1}} = -c^2 x_i^0 x_j^0 + \sum_{k=1}^{n} x_i^k x_j^k$$

where c is the speed of information flow in our system (normally set to 1 for simplified calculations). Further details have been omitted for brevity, however the reader is directed to [Clough and Evans, 2017] for more details.

This bilinear form functions an an inner product (like the Euclidean dot product that we are used to) and allows use to compute norms in a familiar way. That is

$$||\mathbf{x}||_{\mathbb{R}^{n:1}} := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbb{R}^{n:1}}}$$

However, it is possible for $\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbb{R}^{n:1}} < 0$ and so norms may be imaginary.

In fact, the points $\mathbf{x}$ satisfying $\langle \mathbf{x}, \mathbf{x} \rangle_{\mathbb{R}^{n:1}} < 0$ are of particular relevance to hyperboloid geometry as the n-dimensional hyperboloid $\mathbb{H}^n$ is comprised of just such points:

$$\mathbb{H}^n = \{\mathbf{x} \in \mathbb{R}^{n:1} \mid \langle \mathbf{x}, \mathbf{x} \rangle_{\mathbb{R}^{n:1}} = -1, x_0 > 0\}$$

The first condition defines a hyperbola of two sheets, and the second one selects the top sheet. Shortest paths (*geodesics*)between points on the model are given by the hyperbola formed by the intersection of $\mathbb{H}^n$ and the two dimensional plane containing the origin and both of the points.

The distance (along the geodesic) between two points $\mathbf{x}_u, \mathbf{x}_v \in \mathbb{H}^n$ is given by

$$d_{\mathbb{H}^n}(\mathbf{x}_u, \mathbf{x}_v) = \operatorname{arccosh}(-\langle \mathbf{x}_u, \mathbf{x}_v \rangle_{\mathbb{R}^{n:1}})$$

and is analogous to the length of the great circle connecting two points in spherical geometry[4].

We also take this opportunity to define the tangent space of a point $p \in \mathbb{H}^n$ as

$$T_p\mathbb{H}^n = \{x \in \mathbb{R}^{n:1} \mid \langle p, x \rangle_{\mathbb{R}^{n:1}} = 0\}$$

[4]The proof for the distance formula is given in [Reynolds, 1993]

We see that $T_p\mathbb{H}^n$ is the collection of all points in $\mathbb{R}^{n:1}$ that are orthogonal to p. It can be shown (in [Reynolds, 1993]) that $\langle x, x\rangle_{\mathbb{R}^{n:1}} > 0 \; \forall x \in T_p\mathbb{H}^n \; \forall p \in \mathbb{H}^n$. In other words, the tangent space of the hyperboloid is positive definite (with respect to the Minkowski bilinear form) for all points on the hyperboloid. This property actually defines $\mathbb{H}^n$ (equipped with the Minkowski bilinear form) as a Riemannian manifold [Reynolds, 1993]. Furthermore, we obtain a positive norm for any vector $x \in T_p\mathbb{H}^n$, allowing us to preform gradient descent.

2.3 Related Work

An emerging popular belief in the literature is that the underlying metric space of most complex networks is, in fact, hyperbolic. Nodes in real world networks often form a *taxonomy* – where nodes are grouped hierarchically into groups in an approximate tree structure [Papadopoulos *et al.*, 2011]. Hyperbolic spaces can be viewed as continuous representations of this tree structure and so models that embed networks into hyperbolic space have proven to be increasingly popular in the literature [Krioukov *et al.*, 2009; Krioukov *et al.*, 2010]. In fact, this assumption has already had proven success in the task of greedy forwarding of information packets where nodes use only the hyperbolic coordinates of their neighbours to ensure packets reach their intended destination [Papadopoulos *et al.*, 2010].

The most popular of all these models is the Popularity-Similarity (or PS) model [Papadopoulos *et al.*, 2011]. This model extends the "popularity is attractive" aphorism of preferential attachment [Barabási and Albert, 1999] to include node similarity as a further dimension of attachment. Nodes like to connect to popular nodes but also nodes that 'so the same thing'. The PS model sustains that the clustering and hierarchy observed in real world networks is the result of this principle [Alanis-Lobato *et al.*, 2016a], and this trade-off is abstractly represented by distance in hyperbolic space. Maximum likelihood (ML) was used in [Papadopoulos *et al.*, 2011] to search the space of all PS models with similar structural properties as the observed network, to find the one that fit it best. This was extended by the authors in [Papadopoulos *et al.*, 2015a; Papadopoulos *et al.*, 2015b]. Due to the computationally demanding task of maximum likelihood estimation, often heuristic methods are used. For example, [Alanis-Lobato *et al.*, 2016a] used Laplacian Eigenmaps to efficiently estimate the angular coordinates of nodes in the PS model. The authors then combined both approaches to leverage the performance of ML estimation against the efficiency of heuristic search with a user controlled parameter in [Alanis-Lobato *et al.*, 2016b]. Additionally, [Thomas *et al.*, 2016] propose the use of classical manifold learning techniques in the PS model setting with a framework that they call *coalescent embedding*.

Beyond the two-dimensional hyperbolic disk of the PS model, we see that embedding to an n-dimensional Poincaré ball can give more degrees of freedom to the embedding and capture further dimensions of attractiveness than just "popularity" and "similarity" [Nickel and Kiela, 2017; Chamberlain *et al.*, 2017]. By embedding graphs to trees, [De Sa *et al.*, 2018] we able to achieve state-of-the-art results by extending the work of [Sarkar, 2011].

Despite hyperbolic embedding being such an emergent field, not work has yet been done to embed attributed networks. However, it is prolific in the Euclidean domain. [Gibert *et al.*, 2012] embed into a Euclidean vector space based on the statistics of attributes and pairs of attributes, [Li *et al.*, 2017] draw from the well known fields of manifold learning and multi-view learning to align the projections based on topology and attributes and [Liao *et al.*, 2018] use deep learning. In [Niepert *et al.*, 2016], the authors generalised convolutional neural networks from regular pixel lattices to arbitrary graphs.

3 Method

3.1 Problem Definition

We consider a system of N actors given by the set V with $|V| = N$. We use E to denote the set of all interactions in our system. $E = \{(u, v)\} \subseteq V \times V$. We use the matrix $W \in \mathbb{R}^{N\times N}$ to encode the weights of these interactions, where W_{uv} is the weight of the interaction between actor u and actor v. We have that $W_{u,v} > 0 \iff (u, v) \in E$. If the network is unweighted then $W_{u,v} = 1 \; \forall (u, v) \in E$.

Furthermore, the matrix $X \in \mathbb{R}^{N\times d}$ describes the attributes of each actor in the system. We consider the problem of representing a graph given as $\mathbb{G} = (V, A, W, X)$ as set of low-dimensional vectors $\{\mathbf{x}_v \in \mathbb{H}^n \mid v \in V\}$, with $n << \min(N, d)$.

3.2 Random Walks for Learning Global Structure from Local Information and Attributes

Our proposed approach is broken into two steps:

1. Incorporate attribute information to learn a global representation of the system as study.
2. Use gradient descent to learn a low dimensional representation of the global structure.

Following previous works [Grover and Leskovec, 2016], we use a modified random-walk procedure to learn a global representation of the system.

We define the attributional similarity Y as cosine similarity of the attribute vectors of the nodes. That is

$$Y_{uv} = \frac{X_u \cdot X_v}{||X_u||||X_v||}$$

with $\cdot$ denoting the Euclidean dot product and $||\cdot||$ the Euclidean norm. Our choice of using cosine similarity is that it can handle high dimensional data well without making a strong assumption about the data. This measure could be easily changed to a more sophisticated and problem dependant measure of pairwise node attribute similarity, and this is left as future work.

We then additionally define $\bar{W}$ and $\bar{Y}$ to be the row normalized versions of W and Y respectively, such that each row sums to 1. We observe now that each row in $\bar{W}$ and $\bar{Y}$ is a probability distribution. In particular, the entry $\bar{W}_{u,v}$ encodes the probability of jumping from node u to v based on the strength of the topological link between u and v, and $\bar{Y}_{u,v}$

likewise encodes the jump probability from attribute similarity.

Beginning from a source node s in the network, we perform a fixed length walk l through the network. Each step in the walk from one node to the next is a stochastic process based on both topological structure and similarity of attributes. We define $0 \leq \alpha \leq 1$ to be a parameter that controls the trade-off a topological step and an attribute step in the walk.

Formally, we use i to denote the ith node in the walk. Then for each step we sample $\pi_i \sim U(0,1)$ and determine the ith node as follows.

$$x_0 = s$$
$$P(x_i = v \mid x_{i-1} = u) = \begin{cases} \hat{W}_{uv} & \text{if } \pi_i < \alpha, \\ \hat{Y}_{uv} & \text{otherwise.} \end{cases}$$

for $i = 1, 2, ..., l$.

3.3 Building Training Samples from Walks

Taking inspiration from natural language processing, in particular [Mikolov *et al.*, 2013a; Mikolov *et al.*, 2013b], we consider nodes that appear close together in the same walk to be "context pairs". For a source-context pair (u, v), we aim to maximise the probability of observing v, given u, $P(v|u)$.

To build the set of source-context pairs, D, we scan across all walks with a sliding window and add pairs of nodes that appear within the window. We call this window size "context-size" and it is a user defined parameter that controls the size of a local neighbourhood of a node. Previous works show that increasing context size typically improves performance, at some computational cost [Grover and Leskovec, 2016].

3.4 Negative Sampling

We define the probability of two nodes sharing a connection to be function of their distance in the embedding space. Nodes separated by a small distances share a high degree of similarity and should, therefore have a high probability of connection. Similarly, nodes very far apart in the embedding space should have a low probability of connection.

We make the assumption that a source node and neighbourhood node have a symmetric effect over each other in feature space. To this end, we define the symmetric Gaussian function

$$\hat{P}(v|u) := \exp\left(-\frac{d^2_{\mathbb{H}^n}(\mathbf{x}_u, \mathbf{x}_v)}{2\sigma^2}\right)$$

to be the unnormalized probability of observing a link between nodes source node u and context node v. We normalize the probability thusly:

$$P(v|u) := \frac{\hat{P}(v \mid u)}{Z_u}$$
$$Z_u := \sum_{v' \in V} \hat{P}(v' \mid u)$$

[4]The choice for a Gaussian function is motivated by the observation that the gradient is stable, ie: for $x, x' \in \mathbb{H}^n$ $\lim_{x' \to x} \langle x, x' \rangle_{\mathbb{R}^{n:1}} \to -1$ and $\lim_{x \to -1} \partial_x \text{arccosh}^2(-x) \to 2$. Contrast this with $\lim_{x \to -1} \partial_x \text{arccosh}(-x) \to \infty$ [De Sa *et al.*, 2018].

However, the partition function Z_u involves a summation over all nodes $v \in V$, which for large networks, is prohibitively computationally expensive [Grover and Leskovec, 2016].

Following previous works, we overcome this limitation through *negative sampling*. We define the set of negative samples for u, $N(u)$, as the set of v for we we observe no relation with u:

$$\Gamma(u) := \{v \mid (u, v) \notin D\}$$

There is no guarantee that the size of these sets is the same for all u, so we further define

$$S_m(u) := \left\{ x_i \underset{P_n}{\sim} \Gamma(u) \mid i = 1, 2, ..., m \right\}$$

to be a random sample with replacement of size m from the set of negative samples of u, according to a noise distribution P_n. Following [Grover and Leskovec, 2016], we use $P_n = U^{\frac{3}{4}}$ the unigram distribution raised to the $\frac{3}{4}$ power. This means that the probability that a node is selected as a negative sample is proportional to its occurrence probability (ie. the number of times that it appeared over all the random walks for the network).

We then define the loss function for an embedding $\Theta = \{\mathbf{x}_u \mid u \in V\}$ as the mean of negative log-likelihood of observing all the source-context pairs in D, against the negative sample noise:

$$\begin{aligned} E(\Theta) &= -\frac{1}{|D|} \sum_{(u,v) \in D} \log P(v \mid u) \\ &= -\frac{1}{|D|} \sum_{(u,v) \in D} \Bigg[-\frac{d^2_{\mathbb{H}^n}(\mathbf{x}_u, \mathbf{x}_v)}{2\sigma^2} \\ &\quad - \log \sum_{v' \in S_m(u) \cup \{v\}} \exp\left(-\frac{d^2_{\mathbb{H}^n}(\mathbf{x}_u, \mathbf{x}_{v'})}{2\sigma^2}\right) \Bigg] \end{aligned}$$

and optimize over many passes over D, until convergence to obtain the final embedding Θ[*5]. We observe that optimising E involves maximising $P(v \mid u)$ $\forall (u, v) \in D$. To do this, we must minimise $d^2_{\mathbb{H}^n}(\mathbf{x}_u, \mathbf{x}_v)$ and maximise $d^2_{\mathbb{H}^n}(\mathbf{x}_u, \mathbf{x}'_v)$ $\forall v' \in S_m(u)$. This encourages source-context pairs to be close together ion the embedding space, and u to be embedding far from the noise nodes v' [Nickel and Kiela, 2017].

3.5 Optimization on Hyperboloid

The motivation for using the hyperboloid model for complex network embedding is the simplicity at which gradient computation can be computed simply and exactly, versus previous works (like [Nickel and Kiela, 2017; De Sa *et al.*, 2018]) that use the *Poincaré ball* model and approximate gradients [Wilson and Leimeister, 2018].

We follow the example of [Wilson and Leimeister, 2018] to compute gradients with a three step procedure. The procedure will be outlined here briefly (adopting their notation), with more details given in their paper.

[5]We union $S_m(u)$ with $\{v\}$ to bound $P(v \mid u)$ between 0 and 1.

Let us suppose a cost function E that is defined over the whole ambient Minkowski space $\mathbb{R}^{n:1}$. Then E is, of course defined over $\mathbb{H}^n \subset \mathbb{R}^{n:1}$. For a given point on the hyperboloid $p \in \mathbb{H}^n$, we wish to compute the gradient of E with respect to p, denoted $\nabla_p^{\mathbb{H}^n} E \in T_p\mathbb{H}^n$. Then to perform gradient descent optimization, we will move p along $-\nabla_p^{\mathbb{H}^n} E$ by a small amount η to $p' \in T_p\mathbb{H}^n$. Finally we will map p' back to $\mathbb{H}^n$ using an exponential mapping.

To compute $\nabla_p^{\mathbb{H}^n} E$, we first compute the gradient with respect to the ambient space $\mathbb{R}^{n:1}$ as

$$\nabla_p^{\mathbb{R}^{n:1}} E = \left(-\frac{\partial E}{\partial x^0}\bigg|_p, \frac{\partial E}{\partial x^1}\bigg|_p, ..., \frac{\partial E}{\partial x^n}\bigg|_p \right)$$

We then use the familiar vector projection formula form Euclidean geometry (replacing the dot product with the Minkowski inner product) to compute the projection of the gradient with the ambient to its component in the tangent space:

$$\nabla_p^{\mathbb{H}^n} E = \nabla_p^{\mathbb{R}^{n:1}} E + \langle p, \nabla_p^{\mathbb{R}^{n:1}} E \rangle_{\mathbb{R}^{n:1}} \cdot p$$

Having computed the gradient component in the tangent space of p, we define the exponential map to take a vector $v \in T_p\mathbb{H}^n$ to its corresponding point on the hyperboloid:

$$\text{Exp}_p(v) = \cosh(||v||_{\mathbb{R}^{n:1}}) \cdot p + \sinh(||v||_{\mathbb{R}^{n:1}}) \cdot \frac{v}{||v||}$$

This is analogous to the exponential map in spherical geometry with maps points from the tangent space of a point on the sphere, back to the sphere itself[6]. For a concrete example of this (for the spherical case), imagine that p was a point on the globe. A plane flies (seemingly in a straight line) parallel to the surface of the globe in a straight direction of v for a distance of $||v||_{\mathbb{R}^{n:1}}$. Then $p' = \text{Exp}_p(v)$ is the point on the globe that the plane will land at.

So, incorporating all the preceding steps, we compute p' with

1. Calculate ambient gradient $\nabla E_p^{\mathbb{R}^{n:1}}$
2. Compute component of ambient gradient on the tangent space $\nabla_p^{\mathbb{H}^n} E$
3. Set $p' = \text{Exp}_p\left(-\eta \nabla_p^{\mathbb{H}^n} E\right)$

4 Experimental Setup

4.1 Datasets

Table 1 shows the network statistics of the three citation networks used.

4.2 Parameter Settings

Table 2 shows the parameter settings used for the following experiments. For comparison, we used the open-source implementation of the algorithm described by [Nickel and Kiela, 2017]. We used default parameters to train their embeddings.

[6]The spherical exponential map is given by $\text{Exp}_p(v) := \cos(||v||) \cdot p + \sin(||v||) \cdot \frac{v}{||v||}$. Compare this to the hyperboloid case.

Setting σ

$$\sigma(e) = \log(1 + e)$$

4.3 Network Reconstruction

An important aspect of a graph embedding is its capacity – how well does the embedding reflect the original data? To this end, we define the reconstruction experiment. After training our model to convergence upon the complete network, we shall compute distances (according the distance on the hyperboloid) in the embedding space between all pairs of nodes according to our model. We then rank node pairs by their distance in increasing order and, with a sliding threshold, compute both the average precision (AP) and the area under the receiver operating characteristic curve (AUROC). We assign the true edges in the network positive labels and all other pairs as negatives. High values of AP and AUROC suggest that our model is very capable of reconstructing the observed network topology.

4.4 Link Prediction

An additional desirable property for graph embeddings is their ability to predict links between similar node pairs that are not observed in the original network. These links may be missing due to noise in the network. Furthermore, these predicted links may appear in future networks in time-series data. To evaluate our models ability to predict missing links, we randomly select 15% of the edges in the network (5% validation and 10% for testing) and remove them. We randomly select also an equal number of non-edges in the network. We then train the model on the incomplete network, and, like the reconstruction experiment, rank the pairs of nodes based on distance. We then use the removed edges as the positives and the selected non-edges as negatives and, again, compute AP and AUROC. High-scoring embeddings show the models ability to uncover the true similarity of nodes, even with noisy structural information.

4.5 Node Classification

A third common embedding evaluation technique is node classification. Often we are provided with labels of the nodes within the system of study, however, typically this information is incomplete. We have perhaps only a very small number of labelled nodes within the system. The purpose of the node classification experiment is to see how well a node's label can be predicted, based on its position within the embedding space. We make the assumption that nodes are likely to connect to nodes of the same label, and will also display similar attributes. To this end, we devise the following experiment: We first train an embedding using topology and attributes. Note that this is unsupervised, as the embedding is performed with no knowledge of the ground truth labels. After convergence, train a logistic regression model on a subset of the labelled nodes, and then use that model to predict the labels of the other nodes in the network. We record micro-F1 and macro-F1 scores for the following labelled percentage of nodes: 2%, 3%, 4%, 5%, 6%, 7%, 8%, 9%, 10%. We use an out-of-the-box (Euclidean) logistic regression model with the Klein embedding of the network as input features. The

Network	N	$\|E\|$	d	y
Cora_ML	2995	8416	2879	7
Citeseer	4230	5358	2701	6
Pubmed	18230	79612	500	3

Table 1: Network statistics. *Key:* N is the number of nodes, $|E|$ is the number of edges, d is the dimension of node features, y is the number of classes.

Parameter	Value

Table 2: Parameter settings used.

content... |

Table 3: AUROC scores for Network Reconstruction.

content... |

Table 4: AP scores for Network Reconstruction.

content... |

Table 5: AUROC scores for Link Prediction.

content... |

Table 6: AP scores for Link Prediction.

Klein model has the desirable property that straight lines in the model correspond to straight lines in the underlying hyperbolic geometry that is being represented. Alternatively, [Ganea *et al.*, 2018] have generalised logistic regression for hyperbolic space using Möbius transformations. Using this formulation has been left as future work.

4.6 Greedy Routing of Packages

Interest has emerged in the network embedding community for efficient routing of packets of information between nodes in the network, using only a nodes local information (ie: there current position in the hidden metric space, the location of all of their neighbours and the coordinates of the target) [Kleinberg, 2007; Boguná *et al.*, 2010; Bianconi and Rahmede, 2017; Kleineberg and Helbing, 2017]. We observe this behaviour in nature as the so-called *six degrees of separation* effect. To evaluate the performance of our algorithm at this task, we randomly sample 1000 pairs of nodes (sender, receiver) from the largest connected component of the network. We restrict ourselves to the largest component to ensure that there is a path on the network between the randomly selected nodes. Starting from the source node, we repeatedly pass on the package to the neighbour that is closest to the target node in the hyperbolic space. In accordance with previous works [Kleinberg, 2007; Boguná *et al.*, 2010; Bianconi and Rahmede, 2017], if the package arrives at the target node, then we record the routing as a success and compute the so-called *stretch* given by the number of nodes in the chain from the source to the target divided by the ground truth shortest path length. If the package is passed back to a node already in the chain, then the routing is recorded as a failure. We report the number of complete routes and the mean stretch of the completed routes.

5 Results

5.1 Network Reconstruction

5.2 Link Prediction

5.3 Node Classification

5.4 Greedy Routing

6 Conclusion

References

[Alanis-Lobato *et al.*, 2016a] Gregorio Alanis-Lobato, Pablo Mier, and Miguel A Andrade-Navarro. Efficient embedding of complex networks to hyperbolic space via their laplacian. *Scientific Reports*, 6, 2016.

[Alanis-Lobato *et al.*, 2016b] Gregorio Alanis-Lobato, Pablo Mier, and Miguel A Andrade-Navarro. Manifold learning and maximum likelihood estimation for hyperbolic network embedding. *Applied Network Science*, 1(1):10, 2016.

[Barabási and Albert, 1999] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[Bengio *et al.*, 2013] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[Bianconi and Rahmede, 2017] Ginestra Bianconi and Christoph Rahmede. Emergent hyperbolic network geometry. *Scientific Reports*, 7:41974, 2017.

[Boguná *et al.*, 2010] Marián Boguná, Fragkiskos Papadopoulos, and Dmitri Krioukov. Sustaining the internet with hyperbolic mapping. *Nature communications*, 1:62, 2010.

[Chamberlain *et al.*, 2017] Benjamin Paul Chamberlain, James Clough, and Marc Peter Deisenroth. Neural embeddings of graphs in hyperbolic space. *arXiv preprint arXiv:1705.10359*, 2017.

[Clough and Evans, 2017] James R Clough and Tim S Evans. Embedding graphs in lorentzian spacetime. *PloS one*, 12(11):e0187301, 2017.

[De Sa *et al.*, 2018] Christopher De Sa, Albert Gu, Christopher Ré, and Frederic Sala. Representation tradeoffs for hyperbolic embeddings. *arXiv preprint arXiv:1804.03329*, 2018.

[Ganea *et al.*, 2018] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *arXiv preprint arXiv:1805.09112*, 2018.

[Gibert *et al.*, 2012] Jaume Gibert, Ernest Valveny, and Horst Bunke. Graph embedding in vector spaces by node attribute statistics. *Pattern Recognition*, 45(9):3072–3083, 2012.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[Kleinberg, 2007] Robert Kleinberg. Geographic routing using hyperbolic space. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1902–1909. IEEE, 2007.

[Kleineberg and Helbing, 2017] Kaj-Kolja Kleineberg and Dirk Helbing. Collective navigation of complex networks: Participatory greedy routing. *Scientific reports*, 7(1):2897, 2017.

[Krioukov *et al.*, 2009] Dmitri Krioukov, Fragkiskos Papadopoulos, Amin Vahdat, and Marián Boguñá. Curvature and temperature of complex networks. *Physical Review E*, 80(3):035101, 2009.

[Krioukov *et al.*, 2010] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010.

[Li *et al.*, 2017] Jundong Li, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, and Huan Liu. Attributed network embedding for learning in a dynamic environment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 387–396. ACM, 2017.

[Liao *et al.*, 2018] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Attributed social network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2257–2270, 2018.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[Nickel and Kiela, 2017] Maximilian Nickel and Douwe Kiela. Poincar\'e embeddings for learning hierarchical representations. *arXiv preprint arXiv:1705.08039*, 2017.

[Niepert *et al.*, 2016] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016.

[Papadopoulos *et al.*, 2010] Fragkiskos Papadopoulos, Dmitri Krioukov, Marián Boguñá, and Amin Vahdat. Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[Papadopoulos *et al.*, 2011] Fragkiskos Papadopoulos, Maksim Kitsak, M Serrano, Marián Boguná, and Dmitri Krioukov. Popularity versus similarity in growing networks. *arXiv preprint arXiv:1106.0286*, 2011.

[Papadopoulos *et al.*, 2015a] Fragkiskos Papadopoulos, Rodrigo Aldecoa, and Dmitri Krioukov. Network geometry inference using common neighbors. *Physical Review E*, 92(2):022807, 2015.

[Papadopoulos *et al.*, 2015b] Fragkiskos Papadopoulos, Constantinos Psomas, and Dmitri Krioukov. Network mapping by replaying hyperbolic growth. *IEEE/ACM Transactions on Networking (TON)*, 23(1):198–211, 2015.

[Reynolds, 1993] William F Reynolds. Hyperbolic geometry on a hyperboloid. *The American mathematical monthly*, 100(5):442–455, 1993.

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

[Thomas *et al.*, 2016] Josephine Maria Thomas, Alessandro Muscoloni, Sara Ciucci, Ginestra Bianconi, and Carlo Vittorio Cannistraci. Machine learning meets network science: dimensionality reduction for fast and efficient embedding of networks in the hyperbolic space. *arXiv preprint arXiv:1602.06522*, 2016.

[Wilson and Leimeister, 2018] Benjamin Wilson and Matthias Leimeister. Gradient descent in hyperbolic space. *arXiv preprint arXiv:1805.08207*, 2018.