Using hyperbolic large-margin classifiers for biological link prediction

Asan Agibetov^{1,*} **Georg Dorffner**^{1} **Matthias Samwald**^{1}

^{1}Section for Artificial Intelligence and Decision Support, Medical University of Vienna, Austria

*asan.agibetov@meduniwien.ac.at

Abstract

Recently proposed hyperbolic neural embeddings naturally represent latent hierarchical semantic relations, and could provide a suitable bridge from the discrete world of biological networks to continuous geometric representations, enabling down-stream machine learning tasks, such as link prediction. In some cases, however, link prediction is modeled by separating hyperbolic embeddings using classifiers that operate in a flat Euclidean space, thus underexploiting the inherently curved geometric space of embeddings. Herein we present and analyze how recently introduced large-margin classifiers in hyperbolic space could be used in conjunction with hyperbolic embeddings, in order to perform biological link prediction, which exploits the curved geometry of complex biological information.

1 Introduction

Link prediction is the task of finding missing or unknown links among inter-connected entities. This assumes that entities and links can be represented as a graph, where entities are nodes and links are edges (if relationships are symmetric) or arcs (if relationships are asymmetric). When dealing with link prediction in knowledge bases, the semantic information contained is usually encoded as a knowledge graph (KG) (Sri Nurdiati and Hoede, 2008). For the purposes of this work we simply treat a knowledge graph as a graph with labelled edges (arcs), meaning that two entities may be connected with more than one link of different types. In addition, we conform to the *closed-world* assumption. This means that all the existing (asserted) links are considered *positive*, and all the links which are unknown are considered *negative*. This separation into positive and negative links naturally allows us to treat the link prediction problem as a supervised classification problem with binary classifiers (one classifier for each relation type). However, while this separation makes it possible to use a wide array of well-studied machine learning algorithms for link prediction, the main challenge is how to find the best representations for the links. This is the core subject of the recent research trend in learning suitable representations for knowledge graphs, largely dominated by so-called *neural embeddings*. Most commonly, neural embeddings are numeric representations of nodes and relations of the knowledge graph in some continuous space with vectorial structure. An overview of state-of-the-art approaches can be found in (Nickel et al., 2016).

Predicting links is especially relevant in the biomedical domain, where biological knowledge lends itself naturally to be modelled with knowledge graphs. Indeed, biological entities such as genes and gene functions can be modelled as nodes, and links among these entities as edges or arcs. Neural embeddings Θ for these biological entities could be trained with embedding models. And by training a binary classifier on these continuous numeric representations Θ, we could, for example, estimate the probability $P_l((u, v) = 1 \mid \Theta)$ of having a link l = HAS-FUNCTION (e.g., labelled edge) between nodes u = TRIM28 GENE and v = NEGATIVE REGULATION OF TRANSCRIPTION BY RNA POLYMERASE II.

More recently, researchers in machine learning have turned their attention to hyperbolic space as a better candidate for continuous geometric representation of graph-based data (Nickel and Kiela, 2017; Chamberlain et al., 2017; De Sa et al., 2018; Nickel and Kiela, 2018). This approach could be of special interest for the representation of complex biological networks, which were found to inherently exhibit a hyperbolic structure (Krioukov et al., 2010; Alanis-Lobato et al., 2016). However, as argued in (Cho et al., 2018), in many situations hyperbolic embeddings are used in classification tasks (such as link prediction) that operate in ill-fitted Euclidean space. This leads to a situation where (flat) Euclidean classifiers misuse all the learned curved information that lives in hyperbolic embeddings.

Contribution of this work. In this work we compare hyperbolic and Euclidean large-margin classifiers when used for biological link prediction with the embeddings learned in flat and curved geometric spaces. We believe that the lessons learned from this comparison will help in the identification of next steps required for end-to-end hyperbolic embedding training pipelines to adequately exploit inherently curved geometry, and to uncover latent hierarchical semantic relations of complex biological patterns.

2 Background and Methods

Hyperbolic space can not be embedded without distortion in Euclidean space (Efimov, 1963), however, there are several useful models of hyperbolic geometry formulated as a subset of Euclidean space. Two related models of hyperbolic space, popular in the deep learning community, are *hyperboloid* and *Poincare-ball*. In the first model of n-dimensional hyperbolic geometry points are represented on the forward sheet of a two-sheeted hyperboloid (generalization of hyperbola) of $(n+1)$-dimensional Minkowski space. Minkowski space, roughly speaking, is a linear ambient space endowed with a bilinear metric (generalization of inner product) given by $\langle u, v\rangle_{n+1} = -u_0 v_0 + \sum_{i=1}^{n} u_i v_i$. Thus, an n-dimensional hyperboloid $\mathbb{H}^n$ is a collection of points $\mathbb{H}^n = \{x \in \mathbb{R}^{n+1} | \langle x, x\rangle_{n+1} = -1, x_{n+1} > 0\}$. Under this setting, the distance between two points on the hyperboloid is computed with $d_{\mathbb{H}^n}(u, v) = \cosh^{-1}(-\langle u, v\rangle_{n+1})$. The second model of hyperbolic space is obtained by projecting each point of $\mathbb{H}^n$ onto the hyperplane $x_0 = 0$ using the rays emanating from $(-1, 0, \ldots, 0)$. The latter gives us a *Poincare ball model*, identified with the collection of points $\mathbb{B}^n = \{x : (x_0, \ldots, x_n) \in \mathbb{R}^n \mid \|x\|^2 < 1\}$.

Both models have found their use in literature. On one hand, the Poincare ball model is more intuitive to visualize in lower dimensions (Nickel and Kiela, 2017). On the other hand, a hyperboloid model permits much simpler expression for parameter updates with Riemannian gradient descent, and makes computation significantly faster (Wilson and Leimeister, 2018; Nickel and Kiela, 2018). These two models are equivalent and points can be converted via diffeomorphisms from one space to another. Hyperboloid to Poincare ball with $p(x_0, \ldots, x_n) = \frac{(x_1, \ldots, x_n)}{x_0+1}$, and reciprocally with its inverse $p^{-1}(x_1, \ldots, x_n) = \frac{(1+\|x\|^2, 2x_1, \ldots, 2x_n)}{1-\|x\|^2}$.

Datasets. In this work we consider two biological knowledge graphs: UMLS (subset of the Unified Medical Language System (Bodenreider, 2004) semantic network) and BIO-KG (Alshahrani et al., 2017). BIO-KG is a comprehensive and curated biological knowledge graph that incorporates knowledge from several biological ontologies and databases, including human protein interactions, human chemical-protein interactions and drug side effects and drug indication pairs. UMLS has 46 relation types, 137 biological entities and a total number of 6257 links, BIO-KG has 9 relation types, 346,225 biological entities and 1,619,239 links in total.

Neural embedding models. We compare two shallow semi-supervised neural embedding models (Nickel and Kiela, 2017; Agibetov and Samwald, 2018), which aim at learning *entity embeddings* Θ in a d-dimensional Hyperbolic $\mathbb{H}^d$ and Euclidean $\mathbb{R}^d$ spaces, respectively. Both models are simple. They embed observed connected pairs of entities (positives) close to each other, and place entities that do not share any links (generated negatives) farther apart. As in many neural embedding approaches, the weight matrix Θ of the hidden layer of the neural network represents entity embeddings (latent representations). The neural network is trained by minimizing, for each observed connected pair (u, v), the following loss function

$$\arg\min_{\Theta} \mathcal{L}(\Theta) := \sum_{(u,v)} \log \frac{e^{-d(\Theta(u), \Theta(v))}}{\sum_{(u,v') \in Neg_u} e^{-d(\Theta(u), \Theta v')}}, \tag{1}$$

where $\Theta(u)$ is the currently learned d-dimensional representation of entity u, and Neg_u represents all negative pairs of u (i.e., u, v' do not share any link). Both models have the same signature, but operate in different spaces, which means that distance d and parameters Θ are computed/updated differently. For the hyperbolic model we employ the hyperbolic distance $d_{\mathbb{H}^n}$ and Riemanian SGD with geodesic updates (Wilson and Leimeister, 2018), the implementation of which is available on GitHub [1]. The Euclidean model is trained with StarSpace toolkit [2]; details on preparing data and training neural embeddings with this model can be found in (Agibetov and Samwald, 2018).

Large-margin classification in Hyperbolic space. In (Cho et al., 2018) authors propose Hyperbolic Linear Support Vector classification as the extension of the well-known Euclidean SVM to hyperbolic geometry. Analogously to the Euclidean case, we consider a set of decision functions that lead to linear decision boundaries in the hyperbolic space. Linear decision boundaries in hyperbolic space are a set of geodesics (curves) that are obtained by intersecting the hyperboloid $\mathbb{H}^n$ with an n-dimensional hyperplane ($\langle w, x\rangle_{n+1} = 0$) in the ambient space $\mathbb{R}^{n+1}$. Authors (Cho et al., 2018) formulate the optimization problem to solve maximum margin classification with linear decision boundaries in hyperbolic space as

$$\begin{aligned}\arg\min_{w \in \mathbb{R}^{n+1}} f(w) := -\frac{1}{2}\langle w, w\rangle_{n+1} + \\ C\sum_{j=1}^{m} \max(0, \sinh^{-1}(1) - \sinh^{-1}(y^{(j)}(\langle w, x^{(j)}\rangle_{n+1}))),\end{aligned} \tag{2}$$

which closely resembles the Euclidean version, where Euclidean inner products are replaced with Minkowski inner products. The parameter C in Eq. 2 controls the tradeoff between minimizing misclassification and maximizing margin. In all our experiments we use our own Python implementation [3] of Hyperbolic Linear SVM compatible with scikit-learn (Pedregosa et al., 2011), which we based on the official open source implementation in Matlab [4].

Link prediction with neural embeddings. The usual way to perform link prediction with neural embeddings Θ is to use them as some kind of representation of a link l_i between u and v. In Euclidean space, one could leverage the underlying vector space structure and come up with link representations, such as vector addition ($l_i := \Theta(u) + \Theta(v)$) and element-wise multiplication of vector elements (Grover and Leskovec, 2016). Once we fix our link representation method, we can train binary classifiers $f(l_i)$ to perform link prediction (i.e., $f(l_i) > 0.5$ if there is a link between u and v and $f(l_i) \leq 0.5$ otherwise). Such link representations may take into account more geometrical patterns than those that rely on the notion of distance alone (e.g., Fermi-Dirac distribution $P((u,v) = 1 \,|\, \Theta) = 1/(e^{(d(\Theta(u),\Theta(v))-r)/t} + 1)$ as in (Nickel and Kiela, 2017)).

Experimental setting. For each knowledge graph we perform a nested cross-validation procedure for 10 runs. In each run, first, we split independently positive links 10 times into train (80%) and test (20%) datasets. We further generate negative links for each split dataset with the positive to negative ratio 1:1 (i.e., both train and test datasets have this ratio). We then use positive links of the train dataset to compute neural embeddings in hyperbolic $\Theta_{\mathbb{H}^n}$ and Euclidean spaces $\Theta_{\mathbb{R}^n}$ by minimizing the loss function in Eq. 1. Note that we pre-train hyperbolic embeddings on flat graph-representations of knowledge graphs, i.e., all edges are unlabelled, and each pair is connected with at most one edge (the description of this pipeline in Euclidean space in (Agibetov and Samwald, 2018)). Next, we train separate binary classifiers for each relation type with Euclidean and hyperbolic SVM classifiers. Performance of binary classifiers is evaluated with the area under the receiver-operator curve (ROC AUC), and is averaged over all 10 runs. This nested cross-validation procedure with 10 runs is computed separately in Euclidean and hyperbolic cases for dimensions $d \in \{2, 5, 10\}$. For a fair comparison we train embeddings for 500 epochs each time. In both hyperbolic and Euclidean SVMs, the parameter $C \in \{0.1, 1, 10\}$ (Eq.2) is optimized separately on the training dataset for each run.

3 Results and discussion

Table 1 summarizes results of our experiments, where we compared the classification performance of Euclidean and hyperbolic embeddings in conjunction with Euclidean and hyperbolic large-margin classifiers. Each score in this table represents an average classification score of 10 nested cross-validation runs over all relations in the knowledge graph (each relation score itself is an average over 10 runs, and the final score is the average over all relations). Our comparisons are reported for a increasing number of dimensions

[1] https://github.com/lateral/geodesic-poincare-embeddings

[2] https://github.com/facebookresearch/StarSpace

[3] https://github.com/plumdeq/hsvm

[4] https://github.com/hhcho/hyplinear

		Euclidean embeddings		Hyperbolic embeddings	
	dim d	Euc SVM	Hyp SVM	Euc SVM	Hyp SVM
UMLS					
	2	0.661 ± 0.023	0.616 ± 0.019	0.695 ± 0.026	$\mathbf{0.703 \pm 0.018}$
	5	$\mathbf{0.780 \pm 0.023}$	0.743 ± 0.024	0.735 ± 0.030	0.743 ± 0.024
	10	$\mathbf{0.793 \pm 0.025}$	0.754 ± 0.022	0.767 ± 0.031	0.742 ± 0.026
BIO-KG					
	2	$\mathbf{0.692 \pm 0.010}$	0.691 ± 0.010	0.613 ± 0.006	0.676 ± 0.009
	5	$\mathbf{0.776 \pm 0.010}$	0.771 ± 0.011	0.697 ± 0.008	0.756 ± 0.011
	10	0.732 ± 0.009	0.723 ± 0.008	0.711 ± 0.010	$\mathbf{0.763 \pm 0.010}$

Table 1: Performance comparison of flat and curved embeddings and large-margin classifiers for biological link prediction task. Link prediction is performed by training large-margin classifiers in Euclidean (Euc SVM) and hyperbolic (Hyp SVM) spaces on Euclidean and hyperbolic embeddings (classifiers and embeddings are trained separately). Embeddings are trained once per graph, while one separate classifier is trained for each type of relation. Performance of a classifier to predict a link of a certain type is measured with ROC AUC score. Each cell represents a ROC AUC score (± SD (standard deviation)) averaged over all relations in a graph (each relation ROC AUC score is itself averaged after a 10 fold cross-validation.

($d \in [2, 5, 10]$).

Results for UMLS confirmed the main hypothesis supported in (Nickel and Kiela, 2017; De Sa et al., 2018) that hyperbolic embeddings outperform Euclidean embeddings with fewer dimensions. And, as reported in (Cho et al., 2018), that large-margin classification in hyperbolic space utilizes the curved geometry of the learned embeddings better than its flat counterpart (linear euclidean SVM classifier). Moreover, the UMLS graph contains many links (e.g., PART OF) that inherently encode hierarchical semantic relations between the nodes, which are better represented in the hyperbolic space. However, as we increase the number of dimensions, Euclidean embeddings and Euclidean SVM outperform its hyperbolic competitors.

In case of a bigger and complex graph (BIO-KG) the situation seems to be the exact opposite – hyperbolic toolbox largely outperforms its Euclidean counterpart as we increase the size of dimensions ($d = 10$), while flat classifier and flat embeddings perform better with fewer dimensions ($d = 2, 5$). This could be due to the fact that 500 epochs are not enough to disentangle complex biological knowledge in the hyperbolic space in lower dimensions.

In all of our experiments Hyperbolic SVM had significantly better training performance (ROC AUC) than Euclidean SVM, which shows that the curved hyperbolic space does represent the training data better, however, has a poorer generalization trait than its Euclidean counterpart.

4 Lessons learned and future directions

The benefit of learning hyperbolic embeddings is that they require fewer dimensions to capture latent semantic and hierarchical information. This is important for scalability and interpretability (easier to visualize 2 or 3 dimensional embeddings).

From our preliminary results we observed that hyperbolic embeddings capture latent hierarchical semantic relations of the UMLS graph better than Euclidean embeddings in lower dimensions, similar to the state-of-the-art results for the reconstruction of hierarchical relationships (Nickel and Kiela, 2017, 2018; Ganea et al., 2018). For complex and big graphs, such as BIO-KG, we would recommend training hyperbolic embeddings for longer periods ($>$ 500 epochs) in order to better disentangle complex information.

While training hyperbolic embeddings is notorious for long computational time, recent advances in Riemannian SGD optimization in hyperboloid model of hyperbolic space (Wilson and Leimeister, 2018) provide us with computational tools that run much faster than analogous approaches in Poincare ball model (Nickel and Kiela, 2017) (still much slower than in the Euclidean case). Finally, we believe that in order to learn better hyperbolic embeddings (and do it faster), the next steps should be focused on end-to-end hyperbolic embedding training, where hyperbolic large-margin classifier loss is directly incorporated during the training process.

References

Asan Agibetov and Matthias Samwald. 2018. Global and local evaluation of link prediction tasks with neural embeddings. *arXiv*.

Gregorio Alanis-Lobato, Pablo Mier, and Miguel A Andrade-Navarro. 2016. Efficient embedding of complex networks to hyperbolic space via their laplacian. *Sci Rep*, 6:30108.

Mona Alshahrani, Mohammad Asif Khan, Omar Maddouri, Akira R Kinjo, Nria Queralt-Rosinach, and Robert Hoehndorf. 2017. Neuro-symbolic representation learning on biological knowledge graphs. *Bioinformatics*, 33(17):2723–2730.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Res*, 32(Database issue):D267–70.

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

Hyunghoon Cho, Benjamin DeMeo, Jian Peng, and Bonnie Berger. 2018. Large-margin classification in hyperbolic space. *arXiv*.

Christopher De Sa, Albert Gu, Christopher R, and Frederic Sala. 2018. Representation tradeoffs for hyperbolic embeddings. *arXiv*.

N.V. Efimov. 1963. Impossibility of a complete regular surface in euclidean 3-space whose gaussian curvature has a negative upper bound. *Sov. Math. (Doklady)*, 4:843–846.

Octavian-Eugen Ganea, Gary Bcigneul, and Thomas Hofmann. 2018. Hyperbolic entailment cones for learning hierarchical embeddings. *arXiv*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. *KDD*, 2016:855–864.

Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marin Bogu. 2010. Hyperbolic geometry of complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 82(3 Pt 2):036106.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *arXiv:1705.08039 [cs, stat]*.

Maximilian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. *arXiv*.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proc. IEEE*, 104(1):11–33.

Fabian Pedregosa, Gal Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Èdouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*.

S.N. Sri Nurdiati and C. Hoede. 2008. *25 years development of knowledge graph theory: the results and the challenge*. Number 2/1876 in Memorandum. Discrete Mathematics and Mathematical Programming (DMMP).

Benjamin Wilson and Matthias Leimeister. 2018. Gradient descent in hyperbolic space. *arXiv*.