Benchmarking Graph Convolutional Networks Performance with Mixed-Product Space Embedding

Fangchen Li
Department of Statistics
fcli1997@stanford.edu

Lingjue Xie
Department of Statistics
ljxie@stanford.edu

Zelin Li
Department of Statistics
jameszli@stanford.edu

Abstract

Recently non-Euclidean spaces have gained popularity due to their ability to capture the topology of data better. Embeddings in a mixed-product space further improve data representations quality by providing heterogeneous curvature suitable for different structures. We extended Graph Convolutional Network (GCN)[1] to mixed-product spaces. We benchmarked the performance of various combinations of product spaces and provided heuristics on how to choose the best mixture of space embedding.

1 Introduction

Graph convolutional neural networks (GCNs) is a powerful type of neural network designed for learning in graphs. It leverages structural information of graphs and features of nodes. However, GCNs map nodes of the graph to points in Euclidean space, which may not be a good choice for embedding graphs. Many graphs in real life, such as social networks, networks of epidemiology, knowledge graphs and text graph, often have some hierarchical structures, which can not be represented well by Euclidean space, as Euclidean space has a high distortion when embedding these graph structures. Thus, non-Euclidean spaces, such as spherical and hyperbolic space, have gained popularity due to their ability to capture the topology of data better. However, most data is not structured uniformly; for example, some graphs may have some tree-like structure in some regions and have some cycles in other regions. Thus, learning embeddings in a product space combining different spaces (spherical, hyperbolic, Euclidean) may improve data representations quality by providing a space of heterogeneous curvature suitable for diverse structures. We are interested in extending GCNs to mixed-product space, which can combine the prediction ability of GCNs and better representation of real-life graph by mixed-product space geometry. We also provide heuristics on choosing the best mixture space, which demonstrates significant improvements in node classification and linkage prediction on hierarchical and tree-like graphs.

2 Related work

Many innovations have been made recently on GCNs as the topic itself serves as a brand new approach to problems and challenges that cannot be accurately represented in the standard Euclidean space.

Wu et al.(2019)[1] have introduced a comprehensive survey of Graph Neural Networks (GNN), which divides the current state-of-the-art methods into four categories of neural networks, providing an introduction and guidelines to the GNN structures and the appropriate usages under different circumstances. We specifically investigated the category of Convolutional GNN, which includes the GCN structure that we are interested in. Chami et al.(2019)[2] recently introduced Hyperbolic Graph Convolutional Networks (HGCN), an entirely novel way of combining the more traditional GCN

[1]The code of our project is https://github.com/fal025/product_hgcn

with hyperbolic geometry to learn inductive representations of hierarchical and scale-free graphs and data. Liu et al.(2019)[3] proposed a similar neural network called Hyperbolic Graph Neural Network (HGNN), which specifically compares the Euclidean space with Poincare Ball and Lorentz model of hyperbolic geometry. However, the emphases are quite different due to the nature of the different models being compared. We referenced the method from HGCN as it proves to be more successful in the linkage prediction and the node classification tasks.

Gu et al.(2019)[4] have proposed ways to construct mixed-product spaces through learning embeddings in product manifolds. This allows for additional possibilities to represent diverse types of data for a more accurate analysis of the structure within them. Bachmann et al.(2019)[5] proved and developed the embedding of both positive and negative constant curvature spaces into a single set of equations and constructed κ-GCN to prove the concept. Such unification helps to develop the spherical space structure that can be applied in GCN.

We would like to build on these pieces of work to expand the capability of GCN to train on mixed-product spaces and evaluate their performances.

3 Dataset and Features

We demonstrate the efficacy of GCNs with mixed-product embedding on node classification and link prediction tasks with one synthetic and three real datasets of graphs which exhibit different hierarchical structures. Each graph has nodes, edges, node features and node labels. Our algorithm uses an adjacency matrix with each element, indicating whether pairs of nodes are linked or not in the graph, and node feature matrix as inputs.

CORA and PUBMED are citation networks. Nodes represent scientific papers. Edges in the graph are citations between these papers, and node labels are the particular academic areas the published paper belongs to. CORA contains 2,708 machine learning papers divided into 7 classes. Each node has a 1433-long binary feature vector indicating the presence or absence of a word from a dictionary. PUBMED contains 19,717 medicine publications in 3 classes, representing three different types of diabetes. Each node has 500 TF/IDF-weighted word frequencies as features.

AIRPORT is a flight network dataset from OpenFlights.org with 3188 nodes where nodes represent airports, edges represent the airline Routes, and node labels are the population of the country where the airport belongs. This dataset is augmented by Chami et al. (2019)[2] with geographic information. Each node has 12 geographic features including longitude, latitude, altitude, etc. and node label is the GDP of the country where the airport belongs to.

DISEASE is a tree network simulated by Chami et al. (2019)[2] based on the SIR disease spreading model. It has 2665 nodes, where the label of a node is whether the node was infected or not. Each node has 11 features indicating the susceptibility to the disease from different aspects.

CORA, PUBMED and DISEASE datasets are downloaded from the GitHub repository of HGCN. We contacted Ines Chami directly and obtained the AIRPORT dataset via email. These datasets have already been pre-processed and can be directly applied to our algorithm.

4 Method

We inherit the architecture of GCN from the paper of HyperGCN by Chami et al.(2019)[2], apply the unified formula for constant curvature spaces by Bachmann et al. (2019)[5], and integrate with methodology of constructing mixed-product space by Gu et al.(2019)[4].

4.1 Mixed-Product space

The mixed-product space is the Cartesian product space of manifolds with different dimensions. A manifold is a locally Euclidean space at any given point.

$$\mathcal{P} = \mathcal{M}_1 \times \cdots \times \mathcal{M}_k \quad M_i \in \{\mathbb{R}^{\dim M_i}, \mathbb{H}^{\dim M_i}, \mathbb{S}^{\dim M_i}\}$$

Since we cannot directly perform optimization in the mixed-product spaces, we would need to exploit the fact that the tangent space of a given point on the manifold is locally Euclidean. By the property of spaces of our interest, we can map the points between the manifolds and its tangent space back and forth via exponential and logarithm map, where the previous one maps the vector to the tangent space and the later one pulls it back onto the manifold. As described in work by Gu et al. (2019)[4] of mixture space, the exponential map on the mixed-product space in the form of is defined as

$$\exp_p(v) = (\exp_{p_1}(v_1), \cdots, \exp_{p_k}(v_k))$$

where each sub-exponential map corresponds to the exponential map defined on the subspace. Since logarithm map is the inverse map of exponential map, it is natural to be defined as following:

$$\log_p(v) = (\log_{p_1}(v_1), \cdots, \log_{p_k}(v_k))$$

Moreover, the defintion of square distance would be as follow:

$$dist(x, y)^2 = \sum_{i=1}^{k} dist(x_i, y_i)^2$$

For each space, we use the notion of gyrovector space to manipulate points of the stereographic projection of spherical spaces and Poincare Ball model. For a fixed curvature space with curvature c (in our case, +1, -1 for spherical space and Poincare Ball, respectively), the addition and scaling of gyrovectors is given by

$$x \oplus y = \frac{(1 - 2cx^T y - c\|y\|^2)x + (1 + c\|x\|^2)y}{1 - 2cx^T y + c^2\|x\|^2\|y\|^2}$$

$$s \otimes x = \tan(s \tan^{-1}\|x\|)\frac{x}{\|x\|} \quad t \in \mathbb{R}$$

And the distance from x to y and parallel transport from x to y is given by

$$d_c(x, y) = 2|c|^{-1/2} \tan^{-1}\|{-x} \oplus y\|$$

$$\gamma_{x \to y}(t) = x \oplus (t \otimes ((-x) \oplus y))$$

we also have a unified expression exponential map and logarithm map on these constant curvature space manifolds proposed in a recent work by Bachmann et al.(2019)[5]:

$$\exp_x(v) = x \oplus (\tan(|c|^{1/2}\frac{\lambda_x\|v\|}{2})\frac{v}{\|v\|})$$

$$\log_x(y) = \frac{2|c|^{1/2}}{\lambda_x} \tan^{-1}\|(-x) \oplus y\|\frac{(-x) \oplus y}{\|(-x) \oplus y\|}$$

,where $\lambda_x = \frac{2}{1+c\|x\|^2}$ is the factor come from stereographic projection of a point $x \in M \to \tilde{x} \in \mathbb{R}^{\dim M - 1}$.
With these properties equipped, we can now move to applying mixed-product space with GCN.

4.2 Product Space on GCN

Our architecture mainly follow the one described in HyperGCN by Chami et al. (2019)[2], where their proposed model is summarized as below: First, for an input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and Euclidean vertices features $(x^{0,E})_{i \in \mathcal{V}}$, the first layer(the encoder layer) is a mapping from Euclidean to mixed-product space with the exponential map

$$x^{0,\mathcal{P}} = \exp_0((0, x^{0,E}))$$

The encoder layer is followed by graph convolutional layers. For simplicity, we have two layers in our project. Each layer conducts three operations: First, the layer applies weight and biases. Second, we apply logarithm maps to project the feature onto the tangent space with respect to the center node as the origin in each subspace. After that, we aggregate the neighbour's transformed features. The result is then projected back to the mixed-product space via exponential map. Finally, the layer applies a nonlinear activation function and maps the result back to the Euclidean space. Therefore, our operations can be concluded into

$$h_i^{l,\mathcal{P}} = (W^l \otimes x_i^{l-1,\mathcal{P}}) \oplus b^l \qquad \text{(Feature Transformation)}$$

$$y_i^{l,\mathcal{P}} = AGG(h_i^{l,\mathcal{P}}) = \exp_{h_i^{\mathcal{P}}}(\sum_{j \in \mathcal{N}(i)} \log_{h_i^{l,\mathcal{P}}}(h_j^{l,\mathcal{P}})) \qquad \text{(Neighborhood aggregation)}$$

$$x_i^{l,\mathcal{P}} = \sigma^{\otimes}(y_i^{l,\mathcal{P}}) = \exp_o(\sigma(\log_o(y_i^{l,\mathcal{P}}))) \qquad \text{(Activation)}$$

The output of the last hidden layer is in the form of $(x_i^{L,\mathcal{P}})_{i\in\mathcal{V}}$, where L is the number of hidden layers in the network.

For linkage prediction, we adapt the use of Fermi-Dirac decoder, to decode the probability scores for edges

$$p((i,j)\in\mathcal{E}|x_i^{L,\mathcal{P}},x_j^{L,\mathcal{P}})=\frac{1}{1+\exp(\frac{d_{\mathcal{P}}(x_i^{L,\mathcal{P}},x_j^{L,\mathcal{P}})^2-r}{t})},$$

where r and t are hyper-parameters. The model is trained and minimizing using cross-entropy loss with negative sampling. For node classification, we use logarithm function on the mixed-product space and conduct multinomial logistic regression.

5 Experiment & Result

5.1 Experiment Setup

We comprehensively evaluate our experiments on four networks, on both node classification and link prediction tasks. We follow a similar experiment setup as the HGCN paper, Chami et al. 2019[2]. In linkage prediction, edges are split into 85/5/10% for training, dev and test sets. In node classification, nodes are split into 30/10/60% in airport and disease datasets. Standard splits are applied to CORA and PUBMED datasets. We use a training set to train GCNs with mixed-product space. Then we use the dev set to choose the hyperparameters in the decoder step in the linkage prediction task. We try the following feature dimensions for mixed-product space: [4, 8, 16, 32, 64], with each dimension being distributed according to some ratios combining the three spaces. We use ROC-AUC and F1 score to measure our algorithm's performance in linkage prediction and node classification, respectively, in the test set.

5.2 Result and Discussion

From the results of linkage prediction (see Table 1) and node classification (see Table 2) of the airport dataset, we observe that there is consistent tendency for mixed-product space formed by $\mathbb{S}\times\mathbb{H}$ to outperform the result of the HGCN model (a plain hyperboloid space), which is the baseline model we would like to compare with. We suppose such a phenomenon is due to HGCN not capturing the spherical topology hidden in the graph structure. Indeed, the airport graphs have a propensity for having circles, since there are flights that start from one city, followed by other flights, and return to the original city.

For the CORA dataset, there is no considerable difference between the result from mixed-product space and the hyperboloid space alone. HGCN performs better for the linkage prediction task, which is natural since in the citation networks we can find several highly cited paper and all other papers can be considered to be children nodes in the hierarchical structure of citation.

A similar situation occurs for the disease transmission dataset as well, where the hyperboloid space alone outperforms all mixed-product spaces on the linkage prediction task. However, since the disease dataset is specially constructed for the testing of HGCN, it contains a dominating hierarchical structure in the graph. Therefore, other choices of mixed-product spaces are not a proper representation of the graph's topology.

Finally, for the PUBMED dataset, mixed-product space produces a slightly better result than HGCN, where the best choice consists of spherical space. Such a phenomenon makes sense since the citation network in biology may be divided into interconnected fields, where each sub-networks contains a significant hierarchical structure.

Overall, we observed that for the linkage prediction task, the choice of the structure of the space matters more compared with the node classification task. We see that we cannot be sure of whether mixed-product spaces would strictly perform better or worse than the hyperboloid space alone for linkage prediction tasks while they either outperform or have the same performance than the hyperboloid space for node classification tasks. We believe that this is caused by the different nature of these two tasks. For linkage prediction, we are using the information across the entire graph for the algorithm to predict whether two nodes should be connected or not. Thus, the holistic topology of

Table 1: Linkage Prediction(LP) ROC - AUC

	AIRPORT		CORA		DISEASE	
Dimension	HGCN($\mathbb{H}^{dim}$)	Product Space	HGCN($\mathbb{H}^{dim}$)	Product Space	HGCN($\mathbb{H}^{dim}$)	Product Space
4	0.8754	0.8822 ($\mathbb{E}^4$)	0.8638	0.8553 ($\mathbb{S}^2 \times \mathbb{S}^2$)	0.5010	0.5214 ($\mathbb{S}^2 \times \mathbb{S}^2$)
8	0.8976	0.9086 ($\mathbb{S}^2 \times \mathbb{S}^2 \times \mathbb{H}^4$)	0.9107	0.8672 ($\mathbb{S}^4 \times \mathbb{S}^4$)	0.3965	0.5363 ($\mathbb{S}^4 \times \mathbb{S}^4$)
16	0.8891	**0.9286** ($\mathbb{S}^8 \times \mathbb{H}^8$)	0.9187	0.8914 ($\mathbb{H}^8 \times \mathbb{H}^8$)	0.5746	0.5684 ($\mathbb{H}^8 \times \mathbb{H}^8$)
32	0.9070	0.9251 ($\mathbb{S}^{16} \times \mathbb{H}^{16}$)	0.9351	0.9065 ($\mathbb{S}^8 \times \mathbb{S}^8 \times \mathbb{H}^{16}$)	0.6036	0.5804 ($\mathbb{H}^{16} \times \mathbb{H}^{16}$)
64	0.9203	0.9275 ($\mathbb{S}^{32} \times \mathbb{H}^{32}$)	**0.9371**	0.9176 ($\mathbb{S}^{32} \times \mathbb{H}^{32}$)	**0.6375**	0.6000 ($\mathbb{E}^{64}$)

Table 2: Node Classification(NC) F1-Score

	AIRPORT		CORA		PUBMED	
Dimension	HGCN($\mathbb{H}^{dim}$)	Product Space	HGCN($\mathbb{H}^{dim}$)	Product Space	HGCN($\mathbb{H}^{dim}$)	Product Space
4	0.4771	0.4905 ($\mathbb{S}^2 \times \mathbb{S}^2$)	0.7050	0.7050 ($\mathbb{E}^4$)	0.7600	0.7640 ($\mathbb{S}^2 \times \mathbb{S}^2$)
8	0.5210	0.5553 ($\mathbb{S}^4 \times \mathbb{S}^4$)	0.7540	0.7540 ($\mathbb{S}^8$)	0.7690	0.7690 ($\mathbb{S}^2 \times \mathbb{S}^2 \times \mathbb{H}^2 \times \mathbb{H}^2$)
16	0.6088	0.6088 ($\mathbb{S}^{16}$)	0.7890	0.7890 ($\mathbb{S}^{16}$)	0.7510	**0.7740** ($\mathbb{S}^4 \times \mathbb{S}^4 \times \mathbb{H}^4 \times \mathbb{H}^4$)
32	**0.6183**	**0.6183** ($\mathbb{S}^{32}$)	0.7820	0.7820 ($\mathbb{S}^{32}$)	0.7690	0.7720 ($\mathbb{S}^{16} \times \mathbb{S}^{16}$)
64	0.6145	**0.6183** ($\mathbb{S}^{32} \times \mathbb{H}^{16} \times \mathbb{H}^{16}$)	0.7780	**0.7920** ($\mathbb{S}^{32} \times \mathbb{S}^{32}$)	0.7640	0.7700 ($\mathbb{S}^{32} \times \mathbb{H}^{16} \times \mathbb{H}^{16}$)

the dataset matters more when choosing the structure of the space. However, for the node classification task, it may be more natural for nodes to be influenced locally instead of globally. Therefore, the local topology of the dataset matters more than the global topology. In this case, mixed-product spaces would produce better performance.

6 Conclusion and Future work

We extended the functionality of mixed-product space embedding based on the implementation of HGCN. We found that the choice of spaces corresponding to datasets' internal topology, along with the task that we perform, can significantly influence the result of the prediction. We conclude that if the hierarchical relation dominates the graph, then one should have most of the components of the space to be hyperboloid; on the other hand, if a circular structure dominates the graph, then one should choose spherical space instead. One can conduct regular cross-validation to perform a model selection for choosing the proper mixture of product space.

In the future, there are many other steps that we would like to explore further. Firstly, we want to explore more possibilities in mixed-product space, such as including non-constant curvature spaces as our component, so that we can try out our datasets in other mixed-product space to see if we can catch any exciting results. Secondly, we want to modify our current algorithm so that we can learn curvature as the GCN model's parameter for each hidden layer and add attention mechanisms to our model to further improve accuracy. Finally, we would like to test additional datasets to further understand the relationships between mixed-product spaces and different structures of datasets.

7 Team Contributions

All team members contributed equally to the project.

Acknowledgments

We greatly appreciate the help from our mentors Fred Sala, Adva Wolf, and Ines Chami, who provided us numerous help in both technical and conceptual part of our project. We also need to give special thanks to Ines Chami who guide us from the very beginning setting up the environment all the way to the very last. Without their help, we may never have the chance to go this far.

References

[1] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks, 2019.

[2] Ines Chami, Rex Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks, 2019.

[3] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks, 2019.

[4] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*, 2019.

[5] Gregor Bachmann, Gary Bécigneul, and Octavian-Eugen Ganea. Constant curvature graph convolutional networks, 2019.