Using HyperDT to classify the American Gut Dataset

Erica Koyama

ABSTRACT

Hyperbolic embeddings can capture hierarchical structures in data, making them ideal for tasks involving latent hierarchies. However, existing hyperbolic classification and regression methods, while advantageous in this respect, often face challenges such as numerical instability and computational inefficiency. HyperDT addresses these issues by leveraging hyperbolic geometry in decision trees using inner products, improving stability and scalability. Despite these advantages, HyperDT has not been tested on real datasets, and hyperbolic methods, in general, remain underexplored in the context of metagenomics. We applied HyperDT to the American Gut Dataset and observed that while the dataset exhibited latent hierarchical structure, this did not translate into improved performance for HyperDT. Instead, traditional Euclidean-based methods demonstrated comparable accuracy in predictive power to HyperDT.

1. INTRODUCTION

1.1 BACKGROUND: HYPERBOLIC EMBEDDINGS AND HYPERDT

Hyperbolic space, characterized by its constant negative curvature, offers a significant advantage over Euclidean space when capturing complex distance relationships. Its geometry allows for exponentially growing neighborhood sizes, making it ideal for representing data with latent hierarchies [1], [2]. This capability has led to the increasing integration of hyperbolic embeddings in machine learning models, with studies showing that models leveraging hyperbolic space outperform their Euclidean counterparts in tasks such as natural language processing, computer vision, and single cell classification [3], [4], [5], [6].

Despite these advantages, existing hyperbolic classification and regression methods face significant challenges, often being numerically unstable or computationally inefficient [7], [8]. As a result, machine learning algorithms must often compromise between accurate hyperbolic distance representations and algorithm performance. HyperDT addresses these issues by offering a stable, efficient, and hyperbolically appropriate version of decision trees for classification and regression tasks by using inner products to extend Euclidian decision tree models into hyperbolic space [9].

HyperDT has already demonstrated superior performance to both Euclidean and hyperbolic classifiers when tested on three toy datasets with hyperbolic geometries. Extending this approach to biology is compelling because many biological datasets have hierarchical structures, which hyperbolic methods are well-suited to capture. While this motivation has been well-established for other types of biological data, its application to metagenomics is less explored. In metagenomics, the Operational Taxonomic Unit (OTU) table's features (species) exhibit tree-like relationships, making hyperbolic approaches potentially advantageous. However, whether these hierarchical relationships extend to the samples themselves remains an open question. To explore

whether hyperbolic classification can benefit real biological data, we applied HyperDT to classify the American Gut Project dataset and compared it to competing classifiers. In addition to classification, we also generated embeddings, varying the methods used to construct them, and compared their impact on performance. Our contributions in this work are as follows:

1. We developed a Python pipeline to retrieve and preprocess data from the American Gut Project, which compiles global human microbiome data alongside metadata such as health, lifestyle, and dietary information [10].
2. We created a Python pipeline that generates embeddings from the American Gut dataset, builds a decision tree, transforms the decision tree into hyperbolic space, and benchmarks the results in terms of metric distortion and classification accuracy.

1.2 RELATED WORK

Hyperbolic embeddings have been widely explored in biological research, particularly for datasets with latent hierarchical structures. These embeddings have proven effective in modeling known phylogenetic trees [11], [12] and outperforming traditional methods in several biological prediction tasks. For instance, for protein-protein interactions and gene-disease association predictions, hyperbolic embeddings capture complex relationships more efficiently and in fewer dimensions than state-of-the-art methods based on Euclidean or spherical spaces [13]. Moreover, these embeddings have shown promise in analyzing single-cell RNA sequencing data, where variational autoencoders in hyperbolic latent space effectively capture cell developmental trajectories' branching structure [14].

2. HYPERBOLIC DECISION TREE ALGORITHM

For decision trees in hyperbolic space, traditional axis-aligned hyperplanes, commonly used in Euclidean space, cannot capture the underlying geometry of hyperbolic space as they assume flat, Euclidean geometry. In contrast homogeneous hyperplanes (i.e. ones that contain the origin) are convex under shortest paths, partitioning the space into convex, topologically continuous regions.

To adapt decision trees for hyperbolic space, we replace axis-parallel hyperplanes with homogeneous ones while maintaining the Euclidean Classification and Regression Trees (CART) framework. Furthermore, to restrict decision boundary candidates to $O(D|\mathbf{X}|)$, we limit hyperplanes to rotations of the hyperplane $x_0 = 0$ along a single axis. These hyperplanes are parametrized by the axis d and rotational angle θ, and results in the corresponding normal vectors

$$\mathbf{n}(d, \theta) \coloneqq \langle n_0 = -cos(\theta), 0, \dots, 0, n_d = sin(\theta), 0, \dots, 0 \rangle.$$

These normal vectors create hyperplanes that satisfy

$$x_0 cos(\theta) - x_d sin(\theta) = 0.$$

This leads to an $O(1)$ procedure, as $\mathbf{n}(d, \theta)$ is a sparse vector:

$$S(x) = \text{sign}(\max(0, sin(\theta)x_d - cos(\theta)x_0))$$

The resulting decision procedure is curvature-agnostic, meaning it determines a point's position relative to the geodesic decision boundary without directly calculating its location on the hyperboloid. The complexity of this decision procedure remains similar to that of Euclidean CART, and the candidate decision boundaries are chosen based on angles rather than coordinate values. The midpoint between two boundary angles is calculated using the geometry of the hyperboloid, ensuring that the splits remain consistent with the hyperbolic structure while preserving the performance benefits of CART. Benchmark results demonstrate this method to outperform traditional Euclidean classifiers on synthetic datasets with hyperbolic geometries, highlighting its ability to balance scalability and accuracy.

3. CLASSIFICATION EXPERIMENTS

3.1 DATA PREPROCESSING

We used publicly available data to compare the performance of HyperDT with Euclidean classifiers. The American Gut Project microbiome data were downloaded from the UCSD Qiita platform [15], including a filtered feature table in BIOM format (sparse) with OTUs called against against the GreenGenes reference database[16]. We assembled OTU-level metadata from GreenGenes and sample-level metadata from the American Gut Project Qiita page and aggregated all of this into a single AnnData object for downstream analysis.

Metadata entries were filtered to retain only those with meaningful content and were converted into balanced binary or multiclass formats as appropriate for classification tasks. The AnnData sparse matrix was further preprocessed by removing columns with all zeros. Distances were calculated using Aitchison metrics [17].

3.2 CREATING EMBEDDINGS

We calculated the 5-nearest neighbors using the preprocessed data and identified the graph's largest connected component. The Floyd-Warshall algorithm was used to compute the shortest path distances. These distances were isometrically embedded using the coordinate ascent mechanism, where the objective function is iteratively maximized one variable at a time [18], [19].

3.3 BENCHMARKING PROCEDURE

We benchmarked our method against SCIKIT-LEARN's standard Euclidean random forest implementation [20], using trees with depth less than or equal to 3 and at least one sample per

leaf. The benchmarking process was repeated with randomly generated unique seeds and 5-fold cross-validation to ensure robustness.

Experiments were performed on an Ubuntu 20.04.6 LTS machine with an AMD EPYC 7R13 Processor (8 cores, 2.65 GHz), 128 GB of RAM, and an NVIDIA L40S Tensor Core GPU with 48 GB of VRAM. Python 3.10.13, along with CUDA 12.2 and driver version 535.216.03, enabled GPU acceleration.

3.4 RESULTS

We observed the American Gut dataset to exhibit latent hierarchical geometry, indicated by a positive correlation between manifold curvature and average distortion. Specifically, the average distortion increased from 0.244 at a curvature of -2 to 0.249 at a curvature of 2 (Figure 1a). However, this hierarchical structure did not translate into improved performance on the classification task. When comparing the average performance of HyperDT with SCIKIT-LEARN's Euclidean decision tree model, no clear correlation was observed between curvature and classification accuracy, and curvature did not appear to favor either model (Figure 1b).

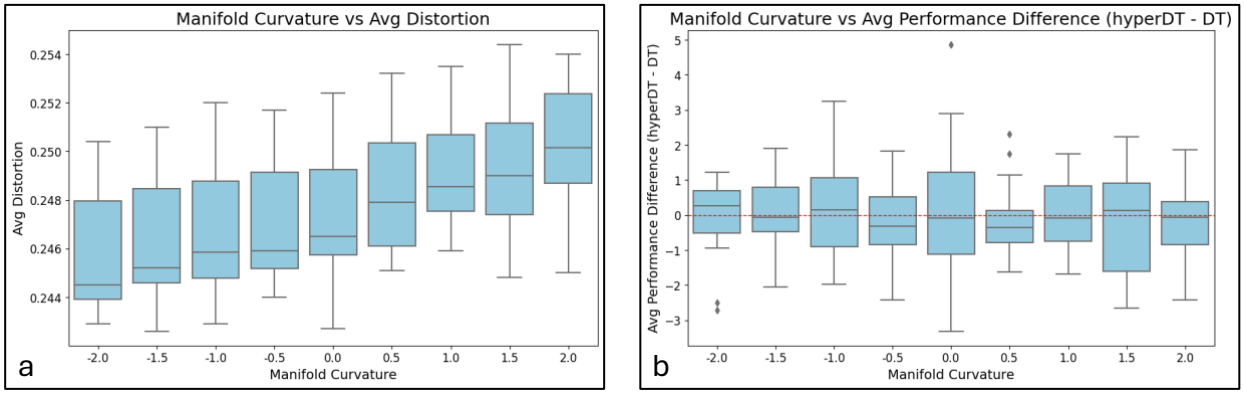

Figure 1: Impact of Manifold Curvature on Average Distortion and Classification Accuracy in the American Gut Dataset

4. CONCLUSION

We evaluated the performance of HyperDT, a hyperbolic decision tree model, on the American Gut Project dataset to investigate its utility to classify biological data with latent hierarchical structures. Our analysis revealed that while the dataset has latent hierarchical geometry, this structure did not translate into improved classification performance. HyperDT's predictive accuracy was comparable to that of SCIKIT-LEARN's Euclidean decision tree model, and no clear relationship was observed between curvature and classification performance.

While hyperbolic embeddings may excel at capturing hierarchical structures, their advantages may not always translate to improved outcomes in downstream classification tasks, especially when the dataset's geometry does not strongly align with the properties of hyperbolic space. This

underscores the importance of carefully evaluating the suitability of hyperbolic methods for specific datasets and tasks. Future work could focus on identifying the conditions under which hyperbolic embeddings offer the most benefit and applying HyperDT to datasets with known latent hierarchies.

REFERENCES

[1] I. Chami, A. Gu, V. Chatziafratis, and C. Ré, "From Trees to Continuous Embeddings and Back: Hyperbolic Hierarchical Clustering," Oct. 01, 2020, *arXiv*: arXiv:2010.00402. doi: 10.48550/arXiv.2010.00402.

[2] B. P. Chamberlain, J. Clough, and M. P. Deisenroth, "Neural Embeddings of Graphs in Hyperbolic Space," May 29, 2017, *arXiv*: arXiv:1705.10359. doi: 10.48550/arXiv.1705.10359.

[3] M. Nickel and D. Kiela, "Poincaré Embeddings for Learning Hierarchical Representations," May 26, 2017, *arXiv*: arXiv:1705.08039. doi: 10.48550/arXiv.1705.08039.

[4] M. Valentino, D. S. Carvalho, and A. Freitas, "Multi-Relational Hyperbolic Word Embeddings from Natural Language Definitions," Feb. 16, 2024, *arXiv*: arXiv:2305.07303. doi: 10.48550/arXiv.2305.07303.

[5] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky, "Hyperbolic Image Embeddings," Mar. 30, 2020, *arXiv*: arXiv:1904.02239. doi: 10.48550/arXiv.1904.02239.

[6] A. Klimovskaia, D. Lopez-Paz, L. Bottou, and M. Nickel, "Poincaré maps for analyzing complex hierarchies in single-cell data," *Nat Commun*, vol. 11, no. 1, p. 2966, Jun. 2020, doi: 10.1038/s41467-020-16822-4.

[7] Y. Jiang, P. Tabaghi, and S. Mirarab, "Learning Hyperbolic Embedding for Phylogenetic Tree Placement and Updates," *Biology*, vol. 11, no. 9, Art. no. 9, Sep. 2022, doi: 10.3390/biology11091256.

[8] X. Fan, C.-H. Yang, and B. C. Vemuri, "Horospherical Decision Boundaries for Large Margin Classification in Hyperbolic Space," Sep. 28, 2023, *arXiv*: arXiv:2302.06807. doi: 10.48550/arXiv.2302.06807.

[9] P. Chlenski, E. Turok, A. Moretti, and I. Pe'er, "Fast hyperboloid decision tree algorithms," Mar. 04, 2024, *arXiv*: arXiv:2310.13841. doi: 10.48550/arXiv.2310.13841.

[10] D. McDonald *et al.*, "American Gut: an Open Platform for Citizen Science Microbiome Research," *mSystems*, vol. 3, no. 3, p. 10.1128/msystems.00031-18, May 2018, doi: 10.1128/msystems.00031-18.

[11] T. Hughes, Y. Hyun, and D. A. Liberles, "Visualising very large phylogenetic trees in three dimensional hyperbolic space," *BMC Bioinformatics*, vol. 5, no. 1, p. 48, Apr. 2004, doi: 10.1186/1471-2105-5-48.

[12] H. Matsumoto, T. Mimori, and T. Fukunaga, "Novel metric for hyperbolic phylogenetic tree embeddings," *Biology Methods and Protocols*, vol. 6, no. 1, p. bpab006, Jan. 2021, doi: 10.1093/biomethods/bpab006.

[13] N. Li, Z. Yang, Y. Yang, J. Wang, and H. Lin, "Hyperbolic hierarchical knowledge graph embeddings for biological entities," *Journal of Biomedical Informatics*, vol. 147, p. 104503, Nov. 2023, doi: 10.1016/j.jbi.2023.104503.

[14] J. Ding and A. Regev, "Deep generative model embedding of single-cell RNA-Seq profiles on hyperspheres and hyperbolic spaces," *Nat Commun*, vol. 12, no. 1, p. 2554, May 2021, doi: 10.1038/s41467-021-22851-4.

[15] A. Gonzalez *et al.*, “Qiita: rapid, web-enabled microbiome meta-analysis,” *Nat Methods*, vol. 15, no. 10, pp. 796–798, Oct. 2018, doi: 10.1038/s41592-018-0141-9.
[16] T. Z. DeSantis *et al.*, “Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB,” *Appl Environ Microbiol*, vol. 72, no. 7, pp. 5069–5072, Jul. 2006, doi: 10.1128/AEM.03006-05.
[17] S. Weiss *et al.*, “Normalization and microbial differential abundance strategies depend upon data characteristics,” *Microbiome*, vol. 5, no. 1, p. 27, Mar. 2017, doi: 10.1186/s40168-017-0237-y.
[18] A. Gu, F. Sala, B. Gunel, and C. Re, “LEARNING MIXED-CURVATURE REPRESENTATIONS IN PRODUCTS OF MODEL SPACES,” 2019.
[19] M. Nickel and D. Kiela, “Poincaré Embeddings for Learning Hierarchical Representations,” May 26, 2017, *arXiv*: arXiv:1705.08039. doi: 10.48550/arXiv.1705.08039.
[20] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.