From Trees to Continuous Embeddings and Back: Hyperbolic Hierarchical Clustering: A Comprehensive Report

Jarod Levy
Paul Meddeb
jarod.levy@etu.minesparis.psl.eu
paul.meddeb@etu.minesparis.psl.eu

ABSTRACT

This paper provides a thorough analysis of the paper: "From Trees to Continuous Embeddings and Back: Hyperbolic Hierarchical Clustering" [1], contextualizing and elaborating on its concepts. It extends the original work to various datasets, including textual data as well as synthetic data, and introduces dendrogram purity as a new metric for result analysis. The study identifies and discusses the original method's limitations, particularly its computational intensity and the trade-offs with its triplet strategy. We also propose an alternative sampling strategy. This report aims to provide a review offering insights into the original work's context, interpretation, extensions, and identified limitations.

KEYWORDS

Hierarchical Clustering, Hyperbolic Embeddings, Dasgupta Cost

1 INTRODUCTION

Hierarchical Clustering (HC) is a fundamental methodology in data analysis, providing an organizational framework that arranges clusters into tree structures to reveal inherent hierarchical relationships. Distinguished from several flat clustering techniques, such as k-means, which segregate datasets into disjoint subsets, HC uniquely fabricates an inclusive hierarchical architecture of clusters, typically represented through a dendrogram. Leaves correspond to data points and internal nodes correspond to clusters. HC has proven its usefulness in various applications where data show hierarchical structure such as phylogenetics [4] or community detection [7]. The principal allure of HC lies in its capacity to afford a multi-scale perspective of the data, elucidating structures across diverse levels of granularity [8, 14, 16].

HC predominantly bifurcates into two methodologies: the agglomerative, which adopts a bottom-up approach, and the divisive, implementing a top-down strategy. Agglomerative Hierarchical Clustering (HAC) commences with individual data points as solitary clusters. These clusters are then iteratively merged, ascending the hierarchy based on a predefined similarity or distance metric, culminating in a single comprehensive cluster. A pivotal element in HAC is the choice of linkage criterion, dictating the mode of distance computation between clusters. Commonly employed linkage methods encompass single linkage [12]: minimum distance, complete linkage [15]: maximum distance, average linkage [13]: average distance, and Ward's method [17]: minimizes intra-cluster variance. Divisive method is a top-down approach that begins with all data points in a single cluster and iteratively splits the cluster into smaller clusters. The process starts at the top of the hierarchy and progressively divides the dataset into finer and finer clusters. The divisive algorithm can use various criteria for splitting a cluster, often based on measures of dissimilarity or distance between data points. Divisive methods are generally considered more computationally intensive but can be more accurate, especially in cases where the underlying data structure is genuinely hierarchical. An example of a divisive method is the bisecting K-means [19] which is a variation of the standard K-means clustering algorithm. It iteratively splits the dataset into two clusters using the K-means algorithm.

Yet, a thorough examination of extant literature [5][18] reveals that theoretical advancements in HC, especially regarding the efficient optimization of these algorithms, have not kept pace with their widespread application. In this juncture, the significance of Dasgupta's cost [3] emerges, introducing a novel metric to assess cluster quality. Dasgupta's cost, conceptualized by Sanjoy Dasgupta, is a criterion for evaluating the efficacy of a hierarchical clustering output. It establishes an objective framework for assessing a tree structure generated by a hierarchical clustering algorithm, focusing on the pairwise similarities between data points. Mathematically, the cost of a tree involves considering each data point pair, and for each, multiplying the population at their lowest common ancestor by their similarity.

$$\text{cost}_G(T) = \sum_{(i,j)\in E} w_{ij}|\text{leaves}(T[i \vee j])|$$

The lower this cost, the more optimal the clustering is deemed. Dasgupta's cost has thus become a cornerstone in hierarchical clustering, providing a robust means to compare different clustering outputs and guiding the evolution of more efficacious clustering algorithms. The main issue of this cost function is its inherent discreteness as it is built on top of the Lowest Common Ancestor (LCA) concept.

This paper introduces a differentiable relaxation of Dasgupta's discrete optimization problem, building upon the foundational work of preceding methodologies such as UFit [2] and gHHC [9]. While these prior approaches have explored gradient-based HC through embedding methods, they diverge in their direct relaxation of Dasgupta's optimization problem. UFit incorporates Euclidean embeddings in conjunction with an "ultrametric fitting" problem, whereas gHHC presupposes the knowledge of leaf hyperbolic embeddings. Both methods, however, are overshadowed by discrete agglomerative algorithms in performance and lack rigorous theoretical underpinnings.

The novel methodology presented herein is underpinned by three groundbreaking concepts:

Conference'17, July 2017, Washington, DC, USA
2023. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

- The innovative parameterization of the search space through the bijection between binary trees and hyperbolic space.

In the realm of hyperbolic embeddings, these representations have shown exceptional aptitude for delineating hierarchical structures and complex networks [6] with scale-free characteristics, such as social and biological networks, and linguistic trees [10]. Hyperbolic embeddings strategically configure distances in the embedding space to mirror the hierarchical relationships inherent in the data, ensuring that closely related items are proximal in the hyperbolic space, while those from disparate hierarchy branches are distanced. This natural alignment makes hyperbolic embeddings an ideal candidate for pursuing Dasgupta's cost relaxation [11].

- The formulation of a continuous analogue for the Least Common Ancestor (LCA), which appears in the expression of Dasgupta's cost, to facilitate its differentiable relaxation.

This formulation links shortest paths in trees and hyperbolic geodesics.

- A decoding algorithm that translates continuous representations back into discrete binary trees.

This latter aspect is particularly crucial, as it enables the approximation of the discrete cost's minimizer with noteworthy precision, with ϵ, an adjustable parameter, finely balancing quality assurance against optimization challenges. Additionally, the proposed model, HypHC, incorporates extensions that augment its scalability and adaptability in handling large datasets and varied clustering scenarios. Under the assumption of perfect optimization, optimal clustering achieved through HypHC approximates the discrete cost's minimizer within a $1+\epsilon$ margin. This relaxation, when integrated with gradient-based optimization, exhibits profound potential in enhancing clustering quality, scalability, and adaptability, and can easily integrated into broader machine learning pipelines.
This paper contributes a distinctive perspective to the optimization challenges in hierarchical clustering. The subsequent sections will delve into an in-depth analysis of this paper, discussing its results, presenting various experimental validations, extensions, and limitations, and offering a critical viewpoint on its findings.

2 DESCRIPTION OF THE PAPER

The paper introduces Hyperbolic Hierarchical Clustering (HypHC) as a novel approach to hierarchical clustering that leverages hyperbolic geometry. The main objective of HypHC is to relax the traditional discrete optimization problem associated with hierarchical clustering and provide a continuous, differentiable representation that enables efficient optimization.

2.1 Continuous Tree Representation via Hyperbolic Embeddings

Trees are represented using hyperbolic embeddings of their leaves in the Poincaré disk. This is based on the insight that hyperbolic space induces a correspondence between leaves embeddings and binary trees, thanks to a decoding algorithm further introduced. Specifically, the embeddings are performed in two dimensions, and the paper acknowledges potential optimization difficulties due to precision requirements. However, a proposition from prior work is cited to address this issue by increasing the dimension [11].

2.2 Differentiable Objective Function

The continuous representation of trees alone is not sufficient for gradient-based hierarchical clustering because Dasgupta's cost requires computing the discrete Lowest Common Ancestor (LCA).

The authors leverage the similarities between geodesics in hyperbolic space and shortest paths in trees to derive a continuous analogue of the discrete LCA. Thereby, the hyperbolic LCA of embeddings x and y is defined as the point on their geodesic closest to the origin (the root). This hyperbolic LCA is then used to introduce a differentiable version of Dasgupta's cost, denoted as CHypHC:

$$CHypHC(Z;w,\tau) = X(w_{ij}+w_{ik}+w_{jk}-w_{HypHC,ijk}(Z;w,\tau))+2Xw_{ij}^{ijk}, \tag{1}$$

where

$$w_{HypHC,ijk}(Z;w,\tau) = (w_{ij},w_{ik},w_{jk})\cdot\sigma_\tau(d_o(z_i\vee z_j),d_o(z_i\vee z_k),d_o(z_j\vee z_k))^\top.$$

2.3 Hyperbolic Decoding

To derive a discrete binary tree structure from optimized embeddings, a decoding algorithm is proposed. This algorithm consists of iteratively merging the most similar pairs based on their hyperbolic LCA distance to the origin, ensuring the resulting tree structure aligns with the hierarchical clustering defined by the embeddings.

2.4 Approximation Ratio Result

The paper presents a theoretical result stating that, under certain conditions, the continuous optimization provided by HypHC yields a $(1+\epsilon)$-approximation to the minimizer of Dasgupta's cost, where ϵ can be made arbitrarily small. To achieve this result, optimization must occur over the entire set of introduced spread embeddings. For any triplet of embeddings:
$\max\{d_o(z_i\vee z_j),d_o(z_i\vee z_k),d_o(z_j\vee z_k)\}-\min\{d_o(z_i\vee z_j),d_o(z_i\vee z_k),d_o(z_j\vee z_k)\} > \delta\cdot O(n)$.
This ensures that the Lowest Common Ancestor (LCA) depths are distinguishable from each other, particularly guaranteeing that the embeddings are well-distributed across the entire disk.

2.5 Practical Considerations

Despite the continuous relaxation, optimization can be highly computationally expensive. That's why the authors propose two empirical techniques to accelerate optimization:

2.5.1 Triplet Sampling. Instead of computing the cost function for all triplets, which is on the order of n^3, the cost is approximated on a sample of triplets. The proposed sampling involves generating all unique pairs of nodes and randomly selecting a third node. In the next section, we will propose a less naive sampling heuristic to further accelerate training.

2.5.2 Greedy Decoding. The authors suggest an approximation of the initially described decoding, reducing the complexity from $O(n^2)$ to $O(n\log n)$. However, we did not understand the utility of such an approximation since it does not improve the overall complexity of the algorithm. Decoding only occurs at the very end, once the hyperbolic embeddings have been optimized. Empirically, we verify that greedy decoding does not speed up the training of the model.

2.6 Experiments

In the experiments, HypHC is evaluated on standard basic datasets, including those from the UCI Machine Learning repository and CIFAR-100, against various hierarchical clustering methods, including agglomerative clustering and UFit [2]. Performance metrics include the Dasgupta Cost (DC) for clustering quality and classification accuracy. HypHC consistently outperforms or matches the best discrete methods and significantly improves over UFit.

However, the experiments seemed somewhat ineffective to us for several reasons:

- Only very simple datasets are tested. In particular, no large dataset is tested, even though the contribution of HypHC is supposed to be its ability to scale to a larger dataset.
- The computation time is not compared between different methods.
- Only the Dasgupta cost is compared, and while it is theoretically interesting, it may not necessarily be the most informative metric.

Therefore, we present our original experiments, which allowed us to address these different points more thoroughly.

3 EXTENSION OF THE RESULTS

Our research endeavors to corroborate and expand upon the original study's conclusions, examining the efficacy of the Hyperbolic Hierarchical Clustering (HypHC) method across a broader spectrum of datasets. We have introduced dendrogram purity as an auxiliary metric to Dasgupta's cost, aiming to furnish a dual-faceted evaluation of clustering quality: a global perspective through Dasgupta's cost and a local viewpoint via dendrogram purity. The latter metric is also posited as a means to facilitate comprehension, as it offers a more intuitive assessment of cluster cohesion. This metric will establish a connection to their outcomes on downstream classification tasks. Additionally, we will evaluate the tradeoffs between clustering quality and computational effort exhibited by HypHC, and subsequently, we will present illustrative visual representations

3.1 Datasets

In our empirical assessment, we selected eight datasets, including those from the original investigation: Zoo, Iris, Glass, and CIFAR. To these, we appended results for dendrogram purity. Furthermore, we integrated the Wine dataset, characterized by its three-class structure and approximately 200 data points, and the Breast Cancer (Wisconsin) dataset (BC in Table 1.), comprising 700 data points delineated into malignant or benign categories. These were procured from the UCI Machine Learning Repository and underwent standard preprocessing to achieve uniformity with the original dataset inputs: normalization (mean of 0 and standard deviation of 1) followed by transformation of features using cosine similarities. The input for the different models is a pairwise similarity normalized graph as in the original paper.

Notably, our methodology is bolstered by the addition of two text-based datasets from the Internet Newsgroup (NG1 and NG2 in Table 1.) data bank composed of 20K documents distributed over 20 topics. A word-document matrix was constructed, and 1000 words were selected based on mutual information between words and documents in an unsupervised manner. We applied standard tf-idf term weighting and normalized each document to 1. We randomly selected 100 documents from five different topics, resulting in two datasets of 500 documents each, distributed over five classes and featuring 1000 characteristics.

3.2 Models and Training

The comparative framework of our study includes six methodologies: the HypHC from the focal paper, UFit—a recent initiative employing "ultrametric fitting" within Euclidean embeddings, and four conventional agglomerative hierarchical clustering algorithms. We did not include a divisive method in our study. It could be integrated for further research. Our training protocol adhered to the original study's regimen, applying Riemannian Adam for optimization over 50 epochs. As in the original paper, note that UFit was not specifically tuned for these datasets. Our training procedure follows the study with 50 epochs of the sampled triplets.
The experimentation was done using 1 GPU Intel Iris Graphics on a personal computer.

3.3 Metrics

To broaden our analytical perspective, we calculated dendrogram purity for each dataset and method. Dendrogram purity, a metric assessing the quality of hierarchical clustering, gauges the extent to which clustering captures inherent class labels or categories at different hierarchy levels, reflecting the purity of clusters relative to predefined labels. Its value was computed at the cutoff level corresponding to the number of classes. This metric was included to provide an alternative viewpoint, as Dasgupta's cost is often complex for lay understanding.

Dasgupta's cost, assessing the quality of a spanning tree, is based on the likelihood that two randomly chosen points and their common ancestor are in the same cluster, dependent on the tree's overall structure and the integration of points within it. In contrast, purity is a measure of a cluster's accuracy in containing single-class elements, evaluated within each cluster by examining the majority class and disregarding data structure between clusters.

We observed scenarios where Dasgupta's cost did not decrease, yet purity significantly increased. This occurred when clusters predominantly contained single-class points, but their arrangement became less optimal according to Dasgupta's cost criteria. For instance, if same-class points grouped together did not minimize average distance to their common ancestor, Dasgupta's cost remained unchanged. Therefore, it was insightful to consider these two metrics simultaneously, one offering a global view and the other a local perspective, to assess both the overall tree structure and the distinct separation of clusters.

3.4 Results

The results are presented in Table 1. For each dataset, we computed two metrics: one for the Dasgupta Cost and another for the Dendrogram Purity. The most favorable metric is denoted in bold, while the second-best metric is underscored. Instances highlighted in red signify that we were unable to calculate the metrics due to memory constraints. Instances marked in cyan indicate metrics that were not part of the original study but have been added for comparative purposes. Furthermore, we computed the percentage

	ZOO	IRIS	WINE	GLASS	BC	CIFAR	NG1	NG2
# FEATURES	16	4	13	9	9	2048	1000	1000
# CLASSES	7	3	3	6	2	10	5	5
# POINTS	101	150	178	214	699	50K	500	500
DASGUPTA / DP	10^{-5}	10^{-5}	10^{-6}	10^{-6}	10^{-7}	10^{-13}	10^{-7}	10^{-7}
AGGLOMERATIVE								
SL	2,897 / 0,858	8,12 / 0,755	1,585 / 0,536	3,018 / 0,466	8,38 / 0,865	4,149 / X	4,51 / 0,56	4,6 / 0,51
AL	2.829 / 0.817	7.939 / 0.727	1.579 / 0.454	2.906 / 0.511	8.391 / 0.846	4.056 / X	4.47 / 0.55	4.57 / 0.56
CL	2.802 / 0.807	7.95 / 0.7	1.58 / 0.537	2.939 / 0.457	8.35 / 0.75	4.078 / X	4.47 / 0.6	4.55 / 0.56
WL	2.827 / 0.912	7.938 / 0.72	1.569 / 0.616	2.92 / 0.505	8.348 / 0.96	4.06 / X	4.45 / 0.65	4.51 / 0.63
CONTINUOUS								
UFit	2.896 / **0.92**	7.916 / 0.77	1.571 / 0.588	2.925 / 0.6	8.345 / 0.96	X / X	4.35 / 0.72	4.4 / 0.75
HypHC	**2.802** / 0.843	**7.881** / **0.77**	**1.566** / **0.77**	**2.902** / **0.686**	**8.341** / **0.96**	**4.056** / X	**4.27** / **0.88**	**4.32** / **0.86**
Percentage Deviation (%)	3,4 / 7,7	3,0 / 0,0	1,2 / 15,4	4,0 / 0,9	0,6 / 0,0	2,3 / X	5,6 / 16,0	6,5 / 11,0
RUN TIME HypHC (sec)	452	492	674	646	489	X	1213	1277

Table 1: Analysis of clustering quality using discrete Dasgupta's Cost (DC) and Dendrogram Purity (DP): top scores highlighted in bold, second best underlined. Red crosses signifies computational limitations, Blue marks newly added results.

deviation between the HypHC metrics and the superior metrics from other methods. Lastly, we provide the computational runtime in seconds for 50 epochs of the HypHC method.

3.5 Analysis

Firstly, we have observed that in almost all cases, HypHC consistently yields the highest dendrogram purity value. With the exception of the Zoo and Iris datasets, which we suspect may be relatively straightforward for the method to distinguish itself, dendrogram purity consistently surpasses other methods when employing HypHC. On the other hand, HypHC consistently achieves the lowest value of the discrete Dasgupta Cost across all datasets. Notably, HypHC outperforms both the top-performing discrete methods and the single similarity-based continuous method, UFit. This finding substantiates the central thesis of the paper, which posits that direct optimization of a continuous relaxation of Dasgupta's objective can significantly enhance clustering quality at both local (DP) and global (DC) levels.

In the case of the two newsgroup datasets, we presented a scenario in which documents are clustered based on the top 1000 words exhibiting the highest mutual information scores. We held the expectation that the HypHC method would significantly outperform other methods due to its utilization of hyperbolic embeddings. This challenging task serves as an ideal context for the method to excel.

Indeed, let us consider the challenge of arranging nodes within a tree in a two-dimensional plane such that the spatial separation between any two nodes accurately reflects their hierarchical relationship. In Euclidean space, this endeavor can lead to congested representations as the depth of the tree increases. Conversely, hyperbolic space, characterized by its exponential surface area growth relative to the radius, offers an expansive canvas capable of accommodating an expanding hierarchical structure without the overlap or congestion often encountered in Euclidean embeddings.

This comprehension holds the potential to yield improved outcomes. Our findings reveal a distinct and substantial improvement in dendrogram purity when employing the HypHC method, surpassing the preceding method by over 10% in terms of purity and more than 5% in terms of Dasgupta's cost.

3.6 Running time considerations

One potential criticism pertains to the computational time required for HypHC. As we augment the number of sampled triplets, we observe a decrease in cost and an increase in dendrogram purity, as evidenced in the visualization section. However, it is essential to note that achieving robust clustering quality demands a computational complexity of $O(n^2)$, as indicated by the graph in the paper. This phase, which accounts for the most significant runtime expense within HypHC, necessitates substantial computation time for training. A summary of the runtime duration for all datasets is provided in Table 1.

For datasets with straightforward characteristics, such as Iris, the HypHC method already demands more than 8 minutes, whereas standard algorithms can produce commendable clustering results in a matter of seconds with no difference in terms of dendrogram purity. In scenarios involving uncomplicated datasets, akin to those employed in the original paper, the application of HypHC appears less feasible due to the extensive runtime requirements. Classical agglomerative methods yield comparable outcomes within seconds. Unfortunately, within our environment, we encountered constraints preventing the evaluation of this method's performance on larger datasets like CIFAR, attributed to limitations in memory and computational complexity. The authors extensively discuss the concepts of scaling and scalability. However, the examples provided predominantly involve datasets of limited size and are focused on GPU-based implementations. This is regrettable, given that a significant portion of this paper is devoted to the utilization of gradient-descent-based algorithms for hierarchical clustering, a technique particularly suitable for large datasets. It is plausible that this method could prove particularly advantageous when dealing with substantial datasets, and it would have been insightful to assess its performance in such contexts.

In summation, this study shows that the HypHC method, grounded in robust theoretical underpinnings, has demonstrated superior experimental performance. It excels in both Dasgupta's cost and dendrogram purity metrics. However, the consideration of computational efficiency remains a critical factor, particularly when the method is applied to less complex datasets or when scalability to larger datasets is required. The original paper's lack of focus on challenging datasets suggests an opportunity for future work, where HypHC's advantages may be more pronounced and distinguishable from traditional methods.

4 SCALING CHALLENGES

4.1 HypHC Performance vs. Average Linkage on Simple Datasets

We investigate the performance of HypHC in relation to the number of epochs, each comprising 10,000 samples (corresponding to the value of n^2 samples suggested by the authors) on the Zoo dataset. Intriguingly, our results (Figure 1) indicate an unexpectedly lower Dasgupta's cost for Average Linkage compared to HypHC, contrasting with the reported values in the original article's result table. Notably, the extensive training time required for HypHC (approximately 50 minutes) raises concerns, particularly when compared to the rapid clustering capabilities of the average linkage method, which achieves results in mere fractions of a second. This stark contrast prompts a reevaluation of HypHC's scalability, touted by the authors.

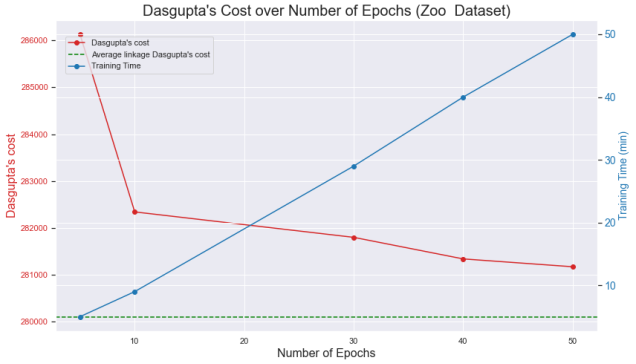

Figure 1: Evolution of Dasgupta's cost over training epochs: HypHC vs. Average Linkage on the Zoo dataset. Training was carried out here on a device without a gpu, unlike the experiments in the previous section, which explain the longer training times.

Moreover, our scrutiny exposes that the dendrogram generated by the Average Linkage method is remarkably efficient, demonstrating near-perfection, particularly in terms of purity. In contrast, the decoded tree produced by HypHC falls short in comparison (Figure 2. This observation underscores the inherent limitations associated with evaluating clustering algorithms on overly simplistic datasets such as the Zoo dataset. The disparity in performance, combined with the effectiveness of alternative methods, calls for a closer examination of HypHC's alleged gains in scalability.

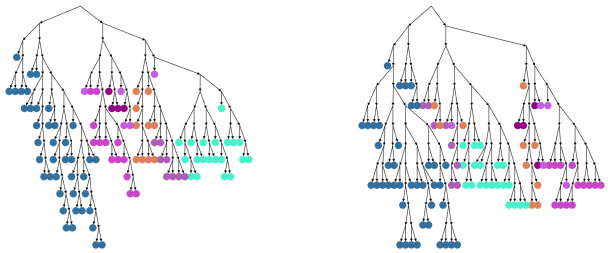

Figure 2: Comparison of dendrograms: average linkage (left) vs. HypHC (right) on the Zoo Dataset

Figure 3: Visualization of HypHC embeddings of Internet Newsgroup 1 dataset and decoded trees with increasing number of triplets. From left to right: 10K, 100K, 500K triplets.

When taking a more difficult dataset, we can see the value of the proposed method. On Figure 3, we present visualizations featuring three dendrograms generated by the HypHC method, each derived from datasets containing 10K, 100K, and 500K triplets, using the Internet newsgroup dataset 1 as the basis. Notably, as observed, augmenting the triplet count results in a noteworthy enhancement in clustering performance. Each color within these visualizations corresponds to a distinct class. While the Dasgupta's cost value experiences relatively minimal fluctuations (less than 2%) as we increase the number of triplets, the purity demonstrates substantial improvement. Our findings corroborate the assertion made in the paper that embeddings tend to gravitate toward the boundary of the Poincaré disk as hyperbolic distances assume a more "tree-like" nature. As stated by the authors, the optimal embedding closely aligns with a tree metric embedding.

4.2 Impact of Dataset Size on HypHC Performance on a Synthetic Dataset

We conducted a scaling study using a synthetic dataset generated from a Gaussian mixture, featuring six features and five clusters (Figure 4). In this investigation, we kept the number of triplets sampled and the number of training epochs constant, ensuring a fixed computational load. The primary focus was on evaluating the influence of increasing the dataset size on HypHC's performance, specifically with respect to Dasgupta's cost.

Our findings reveal that, as the dataset size increases, HypHC's Dasgupta cost diverges from that of average linkage, ultimately stabilizing at a relative deviation of approximately 7% (Figure 5).

This observation sheds light on HypHC's ability to handle larger datasets, at least for our synthetic dataset.

It's worth noting that our exploration was limited by computational resources, preventing us from extending the analysis to even larger sample sizes.

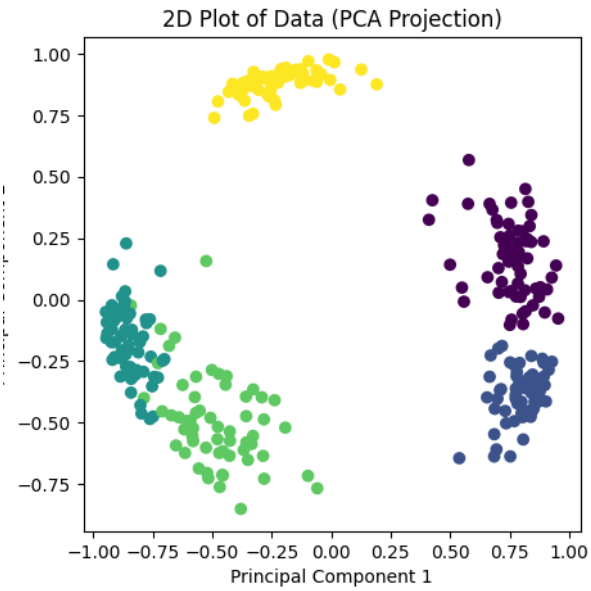

Figure 4: Synthetic dataset: Gaussian mixture with six features and five clusters

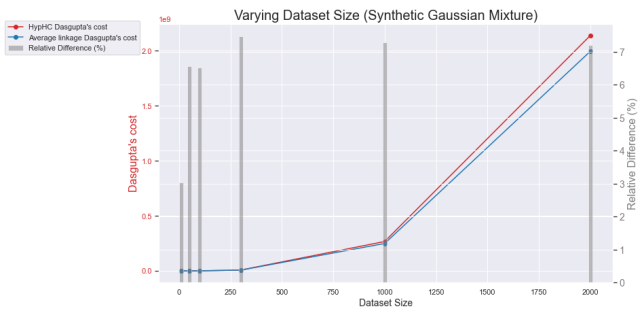

Figure 5: Evolution of HypHC and average linkage Dasgupta's cost with the number of data points (constant running time)

4.3 Enhancing Scalability: Triplet Sampling Optimization

We finally addressed the critical aspect of triplet sampling, recognizing its pivotal role in scaling HypHC. The challenge lies in the necessity of having n^2 samples per epoch, as highlighted in the original article, to achieve accurate results. Our initial experiment on the *Zoo* dataset also underscored the need to train the model on a substantial number of samples for satisfactory performance. However, the considerable computational time required becomes a significant hindrance for practical adoption of this otherwise promising method.

To mitigate this limitation, we explored a refined approach to triplet sampling. Our method prioritizes the sampling of triplets composed of two nearby points and a third distant one, particularly in the early stages of training. More precisely, for each point, we generate pairs from the $x\%$ of points that are most similar and $x\%$ of points that are most dissimilar, with x increasing as training progresses.

As an illustration, Figure 6 displays the distribution of counts for each sampled point. As anticipated, the distribution for the naive method, where triplets are randomly selected, follows a Gaussian pattern. In contrast, the distribution for our method is non-Gaussian, underscoring the fact that specific points are privileged over others in the sampling process.

The underlying hypothesis behind our method is that, at the outset of training, this sampling strategy allows us to directly select triplets with the highest cost. However, it's important to note that our method demonstrated improved training results only for the Iris dataset (Figure 7). Further experimentation with different hyperparameters would be beneficial, although regrettably, we lacked the necessary computational resources for exhaustive testing.

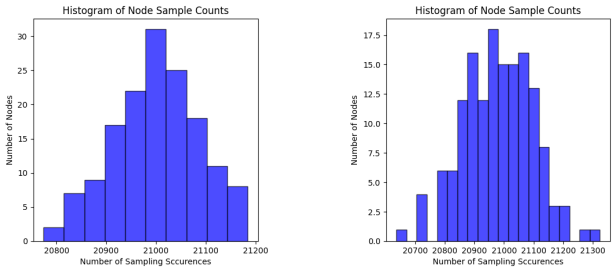

Figure 6: Distribution of sample counts for each point when training HypHC: Original naive method (left) vs. our method (right) on the Iris dataset

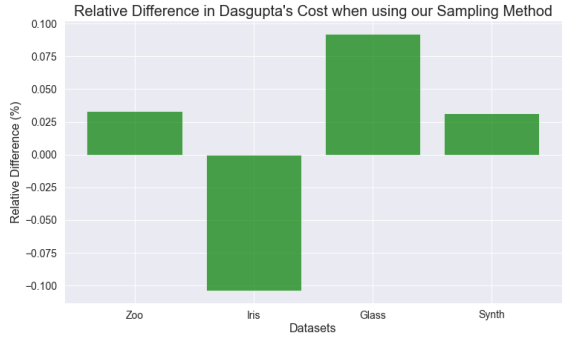

Figure 7: Relative difference in Dasgupta's cost when using our triplet sampling method on various datasets. Our method is only improving clustering for the Iris dataset

5 CONCLUSION

The paper introduces HypHC as a novel approach to hierarchical clustering, leveraging hyperbolic geometry to offer a continuous, differentiable representation for efficient optimization. Through innovative parameterization of the search space, a differentiable objective function, and a decoding algorithm, HypHC demonstrates good performance in both Dasgupta's cost and dendrogram purity metrics. The method's scalability, while promising, demands careful consideration of computational efficiency, particularly on simpler datasets. Further exploration into challenging datasets and optimizations, such as refined triplet sampling, reveals potential areas for improvement. It is worth emphasizing that HypHC could be particularly well-suited for complex and bulky datasets. In scenarios where computation time is not a major constraint, HypHC can offer compelling results that distinctly stand out from classical agglomerative methods. This approach thus opens up exciting avenues for research and application, especially in the realm of large-scale clustering.

REFERENCES

[1] Ines Chami, Albert Gu, Vaggos Chatziafratis, and Christopher Ré. 2020. From Trees to Continuous Embeddings and Back: Hyperbolic Hierarchical Clustering. arXiv:2010.00402 [cs.DS]
[2] Giovanni Chierchia and Benjamin Perret. 2019. Ultrametric fitting by gradient descent. In *Advances in neural information processing systems*. 3175–3186.
[3] Sanjoy Dasgupta. 2016. A cost function for similarity-based hierarchical clustering. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 118–127.
[4] Joseph Felsenstein. 2004. *Inferring phylogenies*. Vol. 2. Sinauer Associates, Sunderland, MA.
[5] A.K. Jain, M.N. Murty, and P.J. Flynn. 1999. Data Clustering: A Review. *ACM Computing Surveys (CSUR)* 31, 3 (1999), 264–323.
[6] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá. 2010. Hyperbolic geometry of complex networks. *Physical Review E* 82, 3 (2010), 036106.
[7] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2019. *Mining of massive data sets*. Cambridge University Press.
[8] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2019. *Mining of Massive Datasets*. Cambridge University Press.
[9] Nicholas Monath, Manzil Zaheer, Daniel Silva, Andrew McCallum, and Amr Ahmed. 2019. Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 714–722.
[10] M. Nickel and D. Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*. 6338–6347.
[11] F. Sala, C. De Sa, A. Gu, and C. Ré. 2018. Representation tradeoffs for hyperbolic embeddings. In *International Conference on Machine Learning*. 4460–4469.
[12] P.H.A. Sneath and R.R. Sokal. 1973. *Numerical taxonomy. The principles and practice of numerical classification*. W.H. Freeman, San Francisco.
[13] R.R. Sokal and C.D. Michener. 1958. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin* (1958).
[14] Michael Steinbach, George Karypis, and Vipin Kumar. 2000. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*.
[15] T. Sørensen. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content. *Kongelige Danske Videnskabernes Selskab* 5 (1948), 1–34.
[16] Therese Sørlie, Charles M. Perou, Robert Tibshirani, et al. 2001. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences of the United States of America* (2001).
[17] J.H. Ward Jr. 1963. Hierarchical Grouping to Optimize an Objective Function. *J. Amer. Statist. Assoc.* 58, 301 (1963), 236–244.
[18] R. Xu and D. Wunsch II. 2005. Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks* 16, 3 (2005), 645–678.
[19] S. Zhong. 2005. Efficient Online Spherical K-means Clustering. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2005)*, Vol. 5. 3180–3185.