Gradient Descent in Hyperbolic Space

Siddartha Devic and Michael Skinner

1 Abstract

Hyperbolic space $\mathbb{H}^n$ is a non-Euclidean geometry with negative curvature. Studies have found that for certain classes of problems where data may need to be represented hierarchically or in a tree-like structure, Hyperbolic geometries may yield better embeddings. However, recent published work has ignored the fact that one can derive gradient descent update rules directly within $\mathbb{H}^n$. The tutorial given in [10] derives and implements GD within $\mathbb{H}^n$ for the problem of finding the barycentre (Karcher mean) of a set of points on a hyperbolic manifold. We extended the work in [10] by deriving and implementing Armijo backtracking search, the Barzelia-Borwein algorithm, and Nesterov accelerated gradient descent in $\mathbb{H}^n$ [1]. We confirm that these modalities accelerate convergence in this convex optimization problem of $\mathbb{H}^2$, an aspect ignored in [10]. Moreover, we demonstrate the usefulness of these algorithms for k-means clustering in $\mathbb{H}^n$.

2 Introduction and Related Work

Hyperbolic space is a Riemannian manifold of n-dimensions having negative curvature, contrasting with Euclidian space, having curvature of 1, and elliptical space whose curvature is greater than 1. Russian mathematician Nikolai Lobachevsky published the first report describing this new geometry, and the principles were extended by other notable eighteenth-century mathematicians such as János Bolyai, Carl Friedrich Gauss, and others [5]. That a non-Euclidian geometry (that is, one not comporting with human experience in the world) could be mathematically complete and consistent had significant philosophical implications, and shifted the field of mathematics from the principal aim of explaining the material world to the modern view that mathematics deals with abstract concepts not necessarily found in nature.

Recently, investigators recognized that the negative curvature of hyperbolic space, with the attendant increase in diameter with distance from the origin, serves well as an embedding space for hierarchical data. Such data are often encoded in tree-like data structures, and the increased number of nodes as one travels from the root, owing to the branching factor, can be modeled in negative-curvature spaces; hyperbolic geometry may be regarded as a continuous analogue to trees [6]. For this reason, hyperbolic space representations have found increasing usefulness in the data sciences. Indeed, investigators recently described that representing the WordNet noun hierarchy in hyperbolic space resulted in significant improvement in the link prediction task when compared to Euclidian embeddings [6].

It is clearly difficult for humans, living in a locally Euclidian world, to visualize objects in non-Euclidian space. Therefore, several models of hyperbolic space have been described to ease mathematical management, in which objects are displayed in two dimensions. These representational options include the Poincaré disc or ball, the Klein model, and the hyperboloid/Lorenz model [5, 7, 10]. The models differ in their specific mathematical manipulations. For example, whereas distances in the Poincaré disk model are faithfully represented in two dimensions, angles are deformed; the opposite case holds true in the Klein disk model.

When points are embedded in a Reimannian manifold such as hyperbolic space, distances between points are measured along the shortest geodesic connecting them. Thus, when gradient descent algorithms are used to optimize functions of parameters embedded in such a space, the parameter update moves in a direction on the geodesic corresponding to the direction of the negative gradient. Early reports of optimization in hyperbolic space described techniques of approximating the gradient on the Poincaré disc or ball [6], but recent work suggests that gradient descent can be performed more straightforwardly, without approximation, when the data are modeled using the hyperboloid/Lorenz model [7, 10]. To simplify the computations, the

[1]Code available at https://github.com/skinn009/hyperbolic.

gradients are computed in the tangent space of the $\mathbb{H}^{n+1}$ Minkowski space in which the $\mathbb{H}^n$ hyperboloid is embedded. The gradients are then projected back onto the manifold using the exponential map. Stochastic gradient descent has been shown to converge almost surely on arbitrary Riemannian manifolds [4]. Furthermore, modified accelerated gradient methods have been proposed for tighter convergence guarantees in non-euclidean space [11]. Nonetheless, there are many algorithms that demonstrate improved convergence rates of gradient descent in euclidian space; our objective is to investigate the effectiveness of several such algorithms in hyperbolic space.

3 Mathematical Background

We use tools from differential geometry [8] to define our model of hyperbolic space, embedded in the ambient space, and constructed from the more general concept of a vector field, V.

3.1 The Minkowski Hyperboloid Model

Definition 1. A map $V \times V \to \mathbb{R}$, denoted $(v, w) \mapsto \langle v, w \rangle$, is a *bilinear form* if for all $\lambda_1, \lambda_2 \in \mathbb{R}$, $\mathbf{v_1}, \mathbf{v_2}, \mathbf{w} \in V$, we have that

$$\langle \lambda_1 \mathbf{v_1} + \lambda_2 \mathbf{v_2}, \mathbf{w} \rangle = \lambda_1 \langle \mathbf{w}, \mathbf{v1} \rangle + \lambda_2 \langle \mathbf{w}, \mathbf{v_2} \rangle$$
$$\langle \mathbf{w}, \lambda_1 \mathbf{v_1} + \lambda_2 \mathbf{v_2} \rangle = \lambda_1 \langle \mathbf{w}, \mathbf{v1} \rangle + \lambda_2 \langle \mathbf{w}, \mathbf{v_2} \rangle.$$

A bilinear form is called *symmetric* if for all $\mathbf{v}, \mathbf{w} \in V$,

$$\langle \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{w}, \mathbf{v} \rangle.$$

We now introduce the *Minkowski* bilinear form, which is the natural and precise way of defining $\mathbb{H}^n$ [9]. Nonetheless, we will see later that it is not the only representation of $\mathbb{H}^n$.

Definition 2. The *Minkowski* product is a symmetric bilinear form defined as

$$\langle \mathbf{u}, \mathbf{v} \rangle_M = u_1 v_1 + \cdots + u_{n-1} v_{n-1} - u_n v_n \text{ for } \mathbf{u}, \mathbf{v} \in \mathbb{R}^n.$$

Clearly, the Minkowski product is symmetric and also a bilinear form since it is linear in each of its parameters. However, note that the Minkowski product is *not* a traditional inner product since there are vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ such that $\langle \mathbf{v}, \mathbf{w} \rangle_M < 0$. The Minkowski product appears in physics and is also called the *Lorentzian* product. We can now use the Minkowski product to define our model of $\mathbb{H}^n$ as follows.

Definition 3. The *Minkowski Hyperboloid model* of $\mathbb{H}^n$ is given by

$$\mathbb{H}^n = \{\mathbf{x} \in \mathbb{R}^{n:1} | \langle \mathbf{x}, \mathbf{x} \rangle_{n:1} = -1, x_{n+1} > 0\}.$$

To be consistent with the notation provided by [10], we switch to the notation $\mathbb{R}^{n:1}$ and $\langle \cdot, \cdot \rangle_{n:1}$ to denote $\mathbb{R}^{n+1}$ space and $\langle \cdot, \cdot \rangle_M$ respectively.

As an example of the Minkowski Hyperboloid model, consider the case of $\mathbb{H}^2$. We have that

$$\mathbb{H}^2 = \{\mathbf{x} \in \mathbb{R}^3 | x_1^2 + x_2^2 - x_3^2 = -1, x_3 > 0\}.$$

We can rearrange the constraints to get that $x_3 = \sqrt{x_1^2 + x_2^2 + 1}$ which corresponds to exactly the upper sheet of the hyperboloid in euclidean $\mathbb{R}^3$ space (Fig. 1). The positive constraint on x_3 removes the second sheet commonly shown in figures of hyperboloids.

3.2 Gradient Updates in Hyperbolic Space

We can now generate points in hyperbolic space $\mathbb{H}^n$ simply by sampling a vector $\mathbf{y} \in \mathbb{R}^n$ and constructing $y_{n+1} = \sqrt{y_1^2 + \ldots y_n^2 + 1}$. Then the new vector $\mathbf{x} = (\mathbf{y}, y_{n+1}) \in \mathbb{H}^n$. Note that $\mathbf{x}$ is a $n+1$ dimensional vector. We now introduce our first problem using these points in $\mathbb{H}^n$.

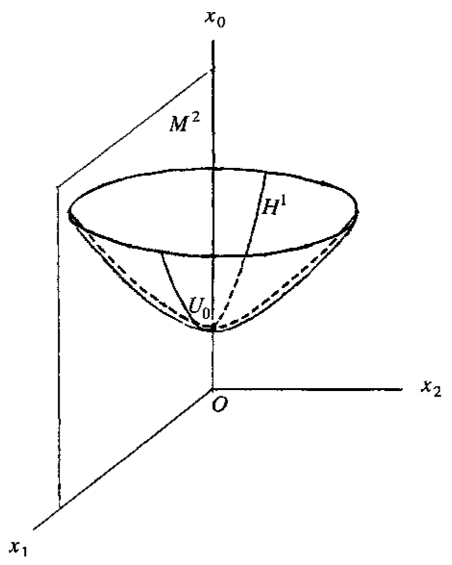

Figure 1: Hyperboloid $\mathbb{H}^2$ of one sheet in $\mathbb{R}^3$ (from [9]). One may also observe H^1 and M^2 in this diagram. Here, H^1 corresponds to the hyperboloid model one dimension lower, taken as a curve across the surface of H^2. Similarly, M^2 corresponds to the Minkowski space of H^1 which is one dimension higher.

Problem 1. Let $\mathbf{x_1}, \ldots, \mathbf{x_m} \in \mathbb{H}^n$ be given. The *barycenter* or *Karcher mean* problem seeks to find some g such that

$$g = \operatorname{argmin}_{\mathbf{p} \in \mathbb{H}^n} \frac{1}{m} \sum_{i=1}^{m} d^2_{\mathbb{H}^n}(\mathbf{p}, \mathbf{x_i}).$$

Where $d_{\mathbb{H}^n}(x, y) = \operatorname{arcosh}(-\langle x, y \rangle_{n:1})$ denotes the distance along the geodesic between points in hyperbolic space [10]. This problem is convex in hyperbolic space—the proof is beyond the scope of this project—and therefore has a global minimizer g [2].

We can also calculate the gradient of the distance function as follows.

$$\begin{aligned}
\nabla_u d_{\mathbb{H}^n} &= \nabla_u \operatorname{arcosh}(-\langle u, v \rangle_{n:1}) \\
&= \frac{1}{\sqrt{\langle u, v \rangle^2_{n:1} - 1}} \nabla_u(-\langle u, v \rangle_{n:1}) \\
&= \frac{1}{\sqrt{\langle u, v \rangle^2_{n:1} - 1}} \nabla_u(-(-u_{n+1}v_{n+1} + \sum_{i=1}^{n} u_i v_i)) \\
&= \frac{1}{\sqrt{\langle u, v \rangle^2_{n:1} - 1}} \nabla_u(u_{n+1}v_{n+1} - \sum_{i=1}^{n} u_i v_i) \\
&= \frac{1}{\sqrt{\langle u, v \rangle^2_{n:1} - 1}} (-v_1, -v_2, \ldots, -v_n, v_{n+1})
\end{aligned}$$

Where $(-v_1, -v_2, \ldots, -v_n, v_{n+1})$ is a vector. However, when we take the gradient in the ambient space, the vector of partial derivatives by the chain rule also negates the final component (since we are in Minkowski metric space). Therefore, the final gradient of the distance function in the ambient space is given by

$$\nabla_u^{\mathbb{R}^{n:1}} d_{\mathbb{H}^n}(u, v) = -((\langle u, v \rangle^2_{n:1} - 1)^{-\frac{1}{2}} \cdot v.$$

Let E denote the objective function given by

$$E = \frac{1}{m} \sum_{i=1}^{m} d^2_{\mathbb{H}^n}(\mathbf{p}, \mathbf{x_i}).$$

We have, from the derivation above, the following gradient of E.

$$\nabla_p^{\mathbb{R}^{n:1}} E = \frac{2}{m} \sum_{i=1}^{m} d_{\mathbb{H}^n}(\mathbf{p}, \mathbf{x_i}) \nabla_u^{\mathbb{R}^{n:1}} d_{\mathbb{H}^n}(\mathbf{p}, \mathbf{x_i})$$

Definition 4. A *tangent vector* to a surface S at a point p is the tangent vector at p of a curve in S passing through p. The *tangent space* T_pS of S at p is the set of all tangent vectors to S at p.

Notice that the tangent space will generally be a *set* with infinite cardinality since there can be more than a single curve on S passing through p. Next, we must project the gradient from the *ambient space* $\mathbb{R}^{n:1}$ onto the *tangent space* of $\mathbb{H}^n$. This is simply done by the following projection (w.r.t. some point p on the hyperboloid).

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

Now we have the gradient of our loss in the tangent space. This essentially corresponds to the normal gradient of a loss in Euclidean space. However, unlike in Euclidean space, we cannot simply subtract our gradient times some step size in the parameter update, for two reasons. First, because the distance metric we are working with is different, the shortest path may be curved (Fig. 5). Second, if we do subtract the gradient, the new point may not be on the hyperboloid $\mathbb{H}^n$. There is no euclidean analog to working in hyperbolic space, i.e. we cannot work in euclidean space and somehow project back to hyperbolic space. For these reasons, we use the *exponential map*, which maps elements from the tangent space back to the manifold. The hyperbolic parameter update equation is:

$$\Theta^{new} = \textsc{Exp}_\Theta(-\alpha \cdot \nabla_\Theta^{\mathbb{H}^n} E).$$

Where $\textsc{Exp}_p$, the exponential map from the tangent space back to the hyperbolic manifold, is:

$$\textsc{Exp}_p(v) = \cosh(||v||)p + \sinh(||v||)\frac{v}{||v||}. \tag{1}$$

It is worth noting that this method still works for general Riemannian manifolds, and is described in further detail in [4]. However, the exponential map will change for different spaces. In some spaces, it may be intractable to compute exactly, and so some approximation will have to be used. Fortunately, in hyperbolic space it is given by the simple closed form expression in equation 1.

4 Methods

We aim to implement the algorithm that computes the barycentre (or center of mass) of a set of points in hyperbolic space. As with the general gradient descent algorithm, choosing the optimal learning rate γ is challenging and significantly impacts algorithm convergence [3]. We investigated and implement several options that might improve convergence of the algorithm and that were not discussed in [10]. These include:

- Standard "vanilla" gradient descent
- Nesterov accelerated gradient descent
- Armijo backtracking search with gradient descent
- Barzelia-Borwein algorithm gradient descent

Owing to the difficulty of graphically representing higher dimensions, we present results obtained in the hyperboloid representation of $\mathbb{H}^2$. Points were generated randomly in the $\mathbb{R}^2$ Minkowski ambient space, and the final coordinate x_3 was constructed to lie on the hyperboloid (section 3.2). Thus, in the limit as the number of points increases, the barycentre of their mass resides at the origin, simplifying the evaluation of our algorithms' convergence. We also randomly instantiated the initial centroid to be on the circle of a certain size (usually $r = 10$), and construct the final coordinate so that the point lies on the hyperboloid. For the vanilla GD, the learning rate was constant at 0.1; this value was also used for gamma in the algorithms requiring this parameter. We also experimented with smaller γ values for AGD, Armijo, and Barzelia-Borwein. Finally, we implemented hyperbolic k-means clustering, with two initialization schemes.

5 Experimental Results

5.1 Vanilla Gradient Descent

We implement the vanilla gradient descent algorithm described in section 3.2 from scratch with numpy to generate the barycentre of points in $\mathbb{H}^2$. In Figure 2 is shown a set of points embedded on the hyperbolic sheet, where the centroid is the red dot; this is clearly different from where the euclidian centroid would lie. To test our work, we also implemented the Poincaré disc projection, which takes points in $\mathbb{H}^2$ and maps them to points on the unit circle in $\mathbb{R}^2$. This allows a *distorted* visualization of $\mathbb{H}^2$ on the unit circle (Fig. 3), where the orange dots represent the centroid generated at each iteration of the algorithm. We can see that there is convergence to the origin, as expected.

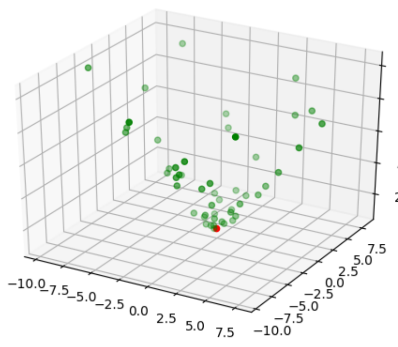

Figure 2: Sampled points on $\mathbb{H}^2$ sitting in $\mathbb{R}^3$ with Karcher mean (red). Karcher mean of $\mathbb{H}^2$ is not the same as centroid in $\mathbb{R}^3$.

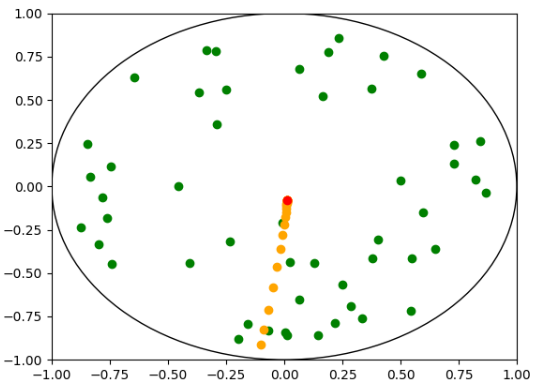

Figure 3: Same sampled points shown in $\mathbb{R}^2$ using Poincaré projection. Path to Karcher mean during GD marked in orange.

To further illustrate the non-intuitive nature of hyperbolic space, we generate points from only the first quadrant of $\mathbb{R}^3$ on $\mathbb{H}^2$ (Fig. 4). We show that the *geodesic*, the shortest path between two points, is not a straight line like in euclidean space (Fig. 5). Furthermore, Fig. 4 demonstrates that owing to the warped distances inherit in the representation, we cannot simply project points from $\mathbb{H}^2$ onto $\mathbb{R}^2$ and take the centroid there, since not only is the distance metric different, the solution must also still be in $\mathbb{H}^2$. Finally, Fig. 6 demonstrates that the Poincaré projection is not restricted to $\mathbb{H}^{\nvDash}$ and shows the projection of a set of points in $\mathbb{H}^3$ along with the convergence of the algorithm (orange points) to their Karcher mean.

5.2 Accelerated Gradient Descent Methods

We also implemented accelerated GD, Armijo line search, and Barzelia-Borwein GD. On a simple dataset with 250 points in $\mathbb{H}^2$, vanilla takes 27 steps before meeting our convergence criterion ($||\nabla_\theta^{\mathbb{H}^2} E||_{inf} \leq 1e-4$). Our implementations of vanilla GD with Armijo line search, Nesterov accelerated GD with Armijo line search, and Barzelia-Borwein GD with Armijo line search take 3, 6, and 6 iterations to converge respectively. Figure 7 shows the loss comparison, and figures 8, 9, 10, and 11 show the path taken for each of the implemented methods.

In general, it is not clear that all of these accelerated gradient methods should converge. Indeed, in dimensions higher than $\mathbb{H}^2$, the Barzelia-Borwein (BB) method did not converge to the correct centroid, and often stopped after one iteration. Our parameter update is as follows.

$$\begin{aligned}\Theta^{new} &= \textsc{Exp}_\Theta(-\alpha \cdot \nabla_\Theta^{\mathbb{H}^n} E) \\ &= \cosh(||-\alpha \cdot \nabla_\Theta^{\mathbb{H}^n} E||)\Theta + \sinh(||-\alpha \cdot \nabla_\Theta^{\mathbb{H}^n} E||)\frac{-\alpha \cdot \nabla_\Theta^{\mathbb{H}^n} E}{||-\alpha \cdot \nabla_\Theta^{\mathbb{H}^n} E||}\end{aligned}$$

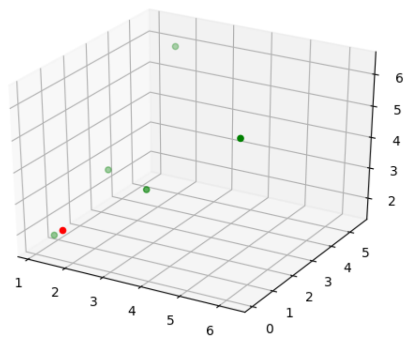
Figure 4: Points in first quadrant of the hyperboloid $\mathbb{H}^2$ depicted with their Karcher mean.

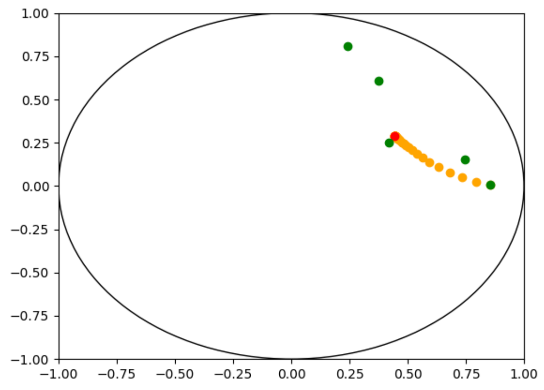
Figure 5: Poincaré projection of points in first quadrant (left). Initializing GD at a point in the point set shows us that the shortest line between two points lies on a geodesic connecting the points.

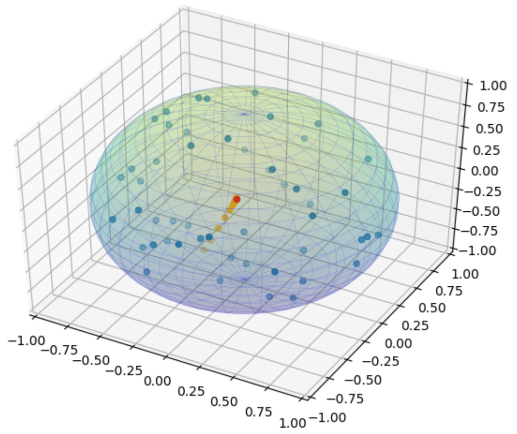
Figure 6: The Poincaré ball projection of $\mathbb{H}^3$ showing convergence of vanilla gradient descent.

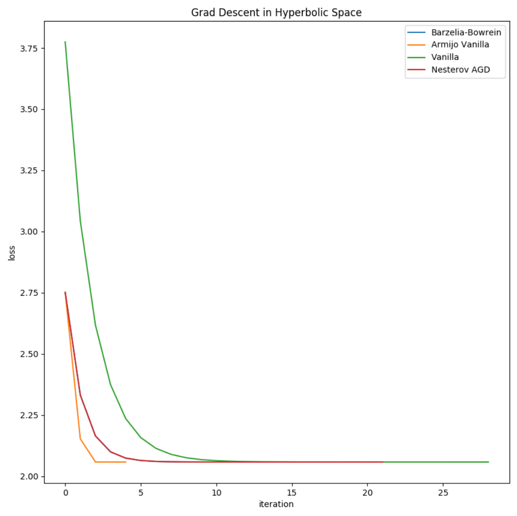

Figure 7: Loss comparison for accelerated GD methods in $\mathbb{H}^2$. Barzelia-Borwein hidden behind AGD.

As a reminder, $\cosh : \mathbb{R} \to [1, \infty)$ and $\sinh(x)$ maps to $\mathbb{R}^+$ if x is positive. The AGD and BB algorithms rely on approximating averaging and approximating local curvature with the Hessian, respectively. We speculate that these approximations may not faithfully carry over into hyperbolic space, owing to the markedly different distance metric. This warrants further investigation. Moreover, additional investigation should be done in regard to whether there are any convergence guarantees associated with using these techniques in non-euclidian space. It may be that further modification of the parameter update step is needed for accelerated gradient methods in hyperbolic space, like the method proposed for Riemannian accelerated GD in [11]. Henceforth, we simply use vanilla gradient descent in our experiments.

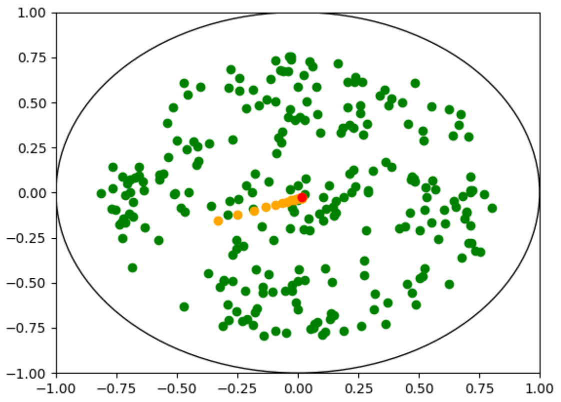
Figure 8: Vanilla gradient descent path ($\alpha = 0.1$).

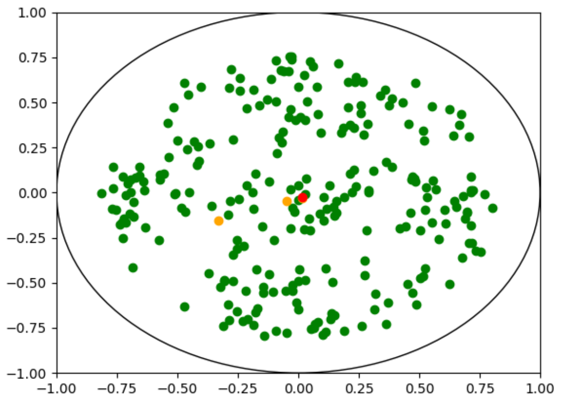
Figure 9: Vanilla GD with Armijo ($\gamma = 0.1$).

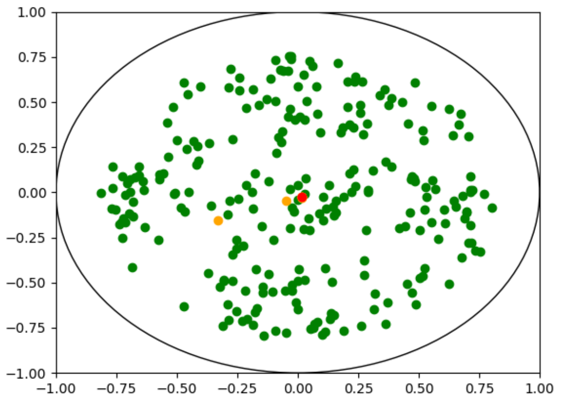
Figure 10: Accelerated GD with Armijo ($\gamma = 0.1$).

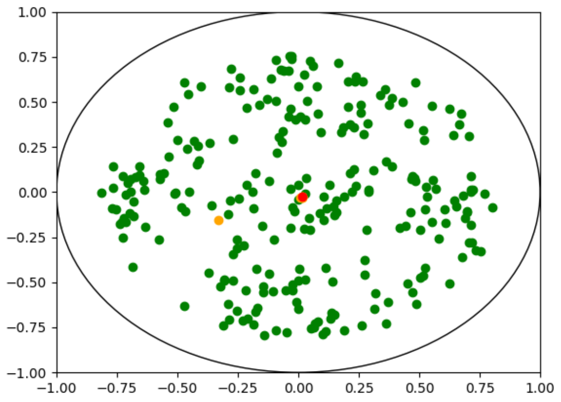
Figure 11: Barzelia-Borwein GD with Armijo ($\gamma = 0.1$).

5.3 Clustering in Hyperbolic Space

Since the barycenter problem is a subproblem within the general unsupervised learning algorithm k-means, we can formulate another problem to solve using a similar algorithm.

Problem 2. Let $\mathbf{x_1}, \ldots, \mathbf{x_m} \in \mathbb{H}^n$ be given. Then we have the following hyperbolic kmeans optimization problem

$$\min_{S_1,\ldots,S_k} \sum_{i=1}^{k} \sum_{j \in S_i} d^2_{\mathbb{H}^n}(\mathbf{x_j}, \mu_i).$$

Where $S_i \subseteq \{1, \ldots, K\}$ is the i'th cluster, $S_i \cap S_j = \emptyset$ for $i \neq j$, each point is assigned to a cluster, and μ_i denotes the centroid of the i'th cluster.

Clustering is non-convex problem and NP-hard even for the $k = 2$ case in euclidean space. We use block coordinate descent to solve approximately, and whenever we need to recompute a cluster center μ_i, we use our implementation described above.

We implement from scratch the hyperbolic k-means algorithm with a random centroid initialization scheme and the "k-means++" scheme, which is an algorithm to better initialize the centroids [1]. We test our implementation by inspecting the special case of $\mathbb{H}^2$ which can be visualized on the Poincaré disc. First, we generate points from the positive and negative octants of $\mathbb{R}^3$ that also sit on the hyperboloid $\mathbb{H}^2$ (Fig. 12). We see that with $k = 2$, our algorithm converges with to a reasonable solution. Next, we randomly sample 10 points in hyperbolic space, and generate additional points around each of these 10 points within a ball of radius 0.2. Then, we attempt to find cluster centers with the hyperbolic k-means algorithm for $k = 10$ (Fig. 13). Here, we converge to a local minima since we can see that not all cluster centers lie at the ball centers. This is to be expected, since in general the k-means loss is non-convex.

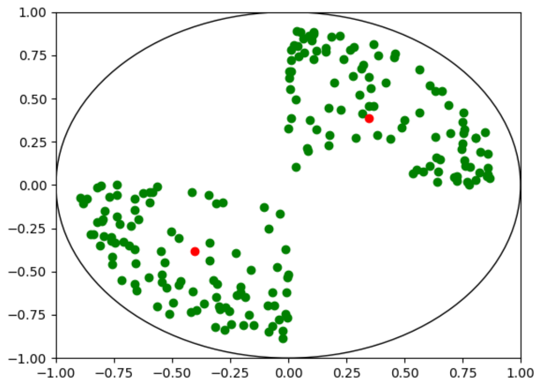

Figure 12: Randomly sampling points in positive and negative octant of $\mathbb{R}^3$ and generating points in $\mathbb{H}^2$ with them.

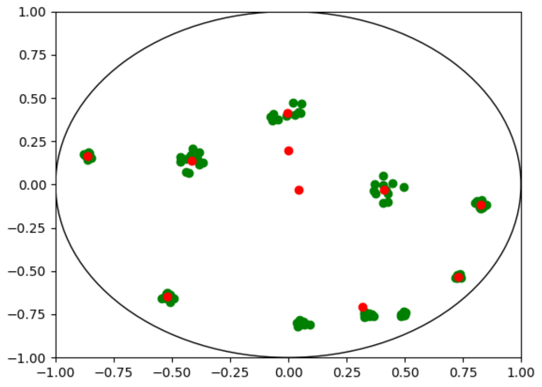

Figure 13: Randomly sample 10 points to serve as centers for hyperbolic balls in $\mathbb{H}^2$, then sample 10 additional points from within each hyperbolic ball (radius $\epsilon = 0.2$).

Both of the previous experiments were run with a random initialization scheme for cluster centers. We now benchmark our algorithm for larger dimensions and k values, with both the random and "k-means++" schemes (Table 1). In general, we see that like in euclidean space, k-means++ performs better than the random initialization.

5.4 A Note on the Convergence of Hyperbolic k-means

The k-means algorithm in euclidean space is guaranteed to converge to some local optima, as it alternates between updating cluster assignments S_i and updating cluster centroids μ_i. Similarly, in hyperbolic space the k-means algorithm will also terminate eventually. A sketch is as follows. By definition, the cluster assignment step is guaranteed to decrease the objective function, as in euclidean space. Furthermore, the cluster centroid algorithm in $\mathbb{H}^n$ has a unique solution [2]. Then the second step of the algorithm where we update the centroids μ_i will also decrease the objective. Since the objective is bounded from below (by zero or some positive value), then we simply terminate when our function changes less than $\epsilon > 0$ between episodes. Numerically, however, ϵ should be determined by the convergence criterion for centroid identification, since we are in fact approximating each μ_i within our procedure.

Finally, the original k-means++ paper proves a $\Theta(\log k)$ competitive ratio in euclidean space [1]. They also prove a competitive ratio in p-normed metric spaces, however, they lose a 2^{2p} factor in addition to the $\log k$ term. The results from the original paper do not seem to apply in non-p-normed spaces, so it is unclear if k-means++ offers the same theoretical guarantees in hyperbolic space.

Dimension	k (Number of clusters)	random init	k++ init
5	5	496.23	**494.89**
	10	**403.21**	409.37
	15	379.17	**364.62**
	20	366.44	**337.84**
10	5	771.52	**769.43**
	10	**744.82**	748.81
	15	727.83	**705.44**
	20	689.47	**652.09**
15	5	951.74	**946.49**
	10	929.26	**920.48**
	15	884.06	**853.19**
	20	842.52	**801.59**
20	5	1050.71	**1050.37**
	10	1010.34	**1001.79**
	15	996.09	**972.96**
	20	987.11	**958.64**

Table 1: k-means objective function average over 10 runs of 100 randomly sampled points for both the random initialization and the k-means++ initialization schemes.

6 Conclusions and Future Work

We have demonstrated the usefulness of several accelerated gradient descent methods in hyperbolic space $\mathbb{H}^n$. Moreover, we have implemented the k-means algorithm in this non-euclidean space. These methods may be helpful in the broader application of embedding hierarchical data structures, such as trees, for the solution of problems with such data. Given additional time, we would like to implement and compare against the AGD method proposed in [11]. Furthermore, it remains to be seen whether convergence guarantees for similar methods can hold on arbitrary Riemannian manifolds. That we found success in the negative-curvature hyperbolic space also suggests that accelerated gradient descent may also be used in positive-curvature elliptical space (which is also a Riemannian manifold).

References

[1] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics.

[2] M. Bacak. Computing medians and means in hadamard spaces, 2012.

[3] D. P. Bertsekas and A. Scientific. *Convex optimization algorithms.* Athena Scientific Belmont, 2015.

[4] S. Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58:2217–2229, 2013.

[5] M. J. Greenberg. *Euclidean and non-Euclidean geometries: Development and history.* Macmillan, 1993.

[6] M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pages 6338–6347, 2017.

[7] M. Nickel and D. Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. *arXiv preprint arXiv:1806.03417*, 2018.

[8] A. Pressley. *Elementary differential geometry.* Springer, 2012.

[9] W. F. Reynolds. Hyperbolic geometry on a hyperboloid. 1993.

[10] B. Wilson and M. Leimeister. Gradient descent in hyperbolic space. *arXiv preprint arXiv:1805.08207*, 2018.

[11] H. Zhang and S. Sra. Towards riemannian accelerated gradient methods, 2018.