Jochem Lippes
J.P.J.Lippes@student.tudelft.nl
Jochem.lippes@gmail
Supervisor: Martin Skrodzki

Approximating nearest neighbours in hyperbolic space

An acceleration data structure for the Klein Disc model of hyperbolic space

TU Delft

1. Background

- **T-distributed stochastic neighbour embedding**
  - Used to embed high dimensional data in low dimensions, maintaining local neighbourhoods.
- **Hyperbolic space**
  - Non-Euclidean geometry better suited for embedding hierarchical data structures.
- **Hyperbolic t-SNE**
  - Embedding high dimensional data using t-SNE into hyperbolic space.
- **Accelerating hyperbolic t-SNE**
  - An altered version of a polar quadtree data structure made to accelerate hyperbolic t-SNE into the Poincaré Disk model, using the midpoints of cells to summarize groups of points according to their midpoint. (see fig. 2, 3)
- **Klein Disk model**
  - Similar to Poincaré disk model. (see fig 1)
  - Straight lines appear as straight chords on Euclidean unit circle.

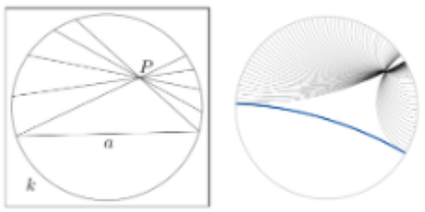

Fig 1. Left geodesics of Klein disk model, not intersecting with line A through point P, right Poincare disk models geodesics through a point not intersecting with blue line.

2. Questions

- **How does the performance, in terms of computational efficiency and quality of results, of the data structure for hyperbolic t-SNE designed for the Poincaré Disc model, compare to one designed for the Klein Disc model?**
- Is it possible to create an acceleration data stucture that works for t-SNE in the Klein Disk model.

3. Method

- Adapt the implementation for the Poincaré Disk model by swapping out the geometric calculations with those for the Klein Disk Model.
- Test whether the implementation for the Klein Disk model works by comparing the acceleration to an exact implementation.
- Compare the runtime and of the two implementations.
- Compare the quality of results in terms of the amount of retained closest neighbours between the two implementations.

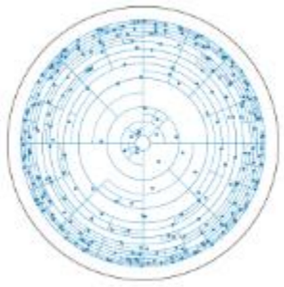

Fig 2.Polar quadtree containing points, split into cells.

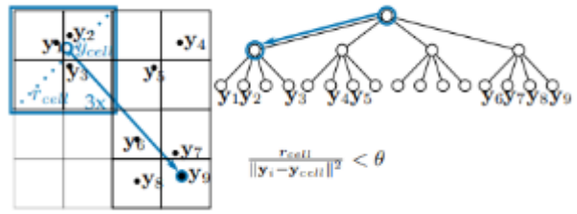

Fig 3. Depiction of how acceleration works using quadtree in regular Euclidean 2-dimensional space. Points in top-left quadrilateral are sufficiently far away from the query point and are thus summarized using their midpoint.

4. Results

- Acceleration data structure successfully accelerates computations while maintaining close quality of results to exact solution.

For the datasets tested, our implementation runs faster than one for the Poincaré Disk model.

- Quality of results is worse for Klein Disk model.

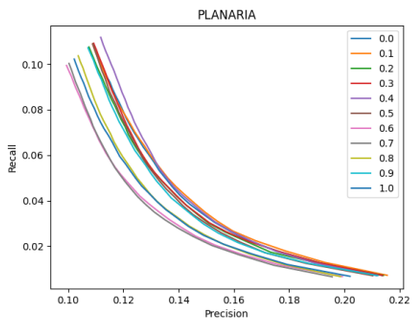

Fig 4. Graph showing precision/recall for different values of θ parameter that steers how much is approximated. Note the quality does not decrease significantly when approximated (higher θ)

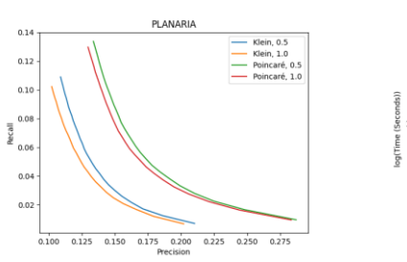

Fig 6. Graph showing precision/recall for Poincaré vs Klein implementation. Note that Poincaré performs significantly better for both accelerated and exact.

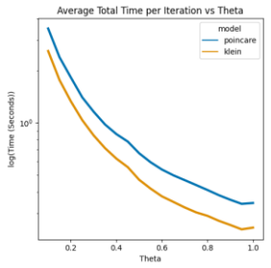

Fig 5. comparing runtimes between the two implementations for different values of θ.

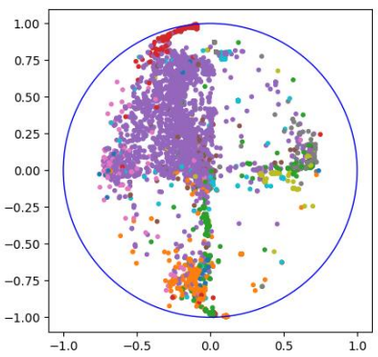

Fig 7. Embedding of Planaria dataset obtained using our implementation

5. Conclusions

- More testing with different datasets necessary to fully conclude on usability of implementation.
- Tests that have been run indicate Klein model runs faster, but with worse results.
- For the results of datasets we have tested the gain in run time is not worth the drop in results, as such we do not recommend using our implementation over the one for the Poincaré Disk for these datasets.