Accelerating hyperbolic t-SNE

Quadtree generalization for the upper half-plane model

Dido Dimitrov <dhdimitrov@student.tudelft.nl>

1. Introduction

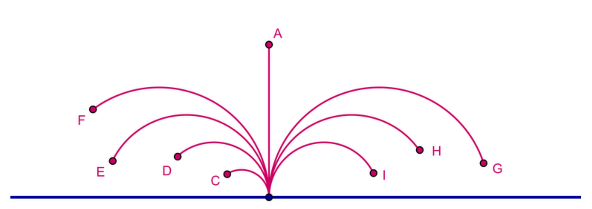
Background

- Dimensionality reduction using **t-SNE** [3]
- Hyperbolic spaces
- Accelerated data structure: **Barnes Hut** [4]

Research Gap

- No implementaion of the suggested quadree for the upper half model [2] in the context of hyperbolic t-SNE.

Research Question: Implemet the suggested quadtree for the upper half model [2] in the context of hyperbolic t-SNE and benchmark it against different embedings (error and speed).



2. Methodology

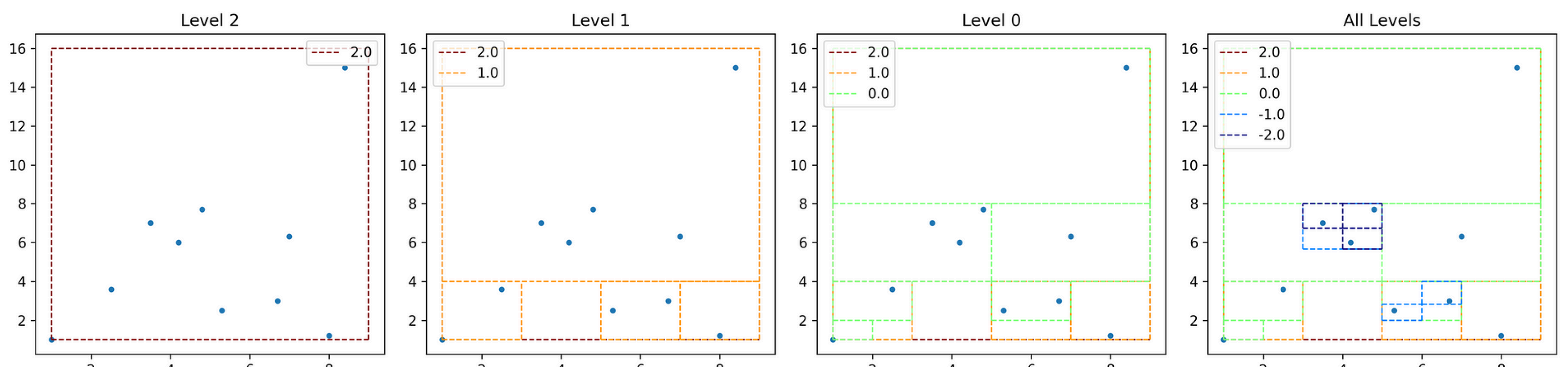
Horobox: defined by its corner points $(x_{\min}(B), z_{\downarrow}(B))$ and $(x_{\max}(B), z_{\uparrow}(B))$, which are minimal and maximal in both directions. We define $\frac{z_{\uparrow}(C)}{z_{\downarrow}(C)} = 2^{h(C)}$ and $\frac{x_{max}(C) - x_{min}(C)}{z_{\downarrow}(C)} = w(C)$ which we will call the height and the width.

Initialization the root cell

For a given set of points first we find the bounding horobox C'
If $w(C') \leq 1$ and $h(C') \leq 1$, we find smallest whole number ℓ such that $w(C') \leq 2^{\ell}$ and $h(C') \leq 2^{\ell}$ and we let $w(C) = h(C) = 2^{\ell}$
Otherwise we find the smallest whole number ℓ such that $w(C') \leq 2^{2^{\ell}-1}$ and $h(C') \leq 2^{\ell}$ and we let $h(C) = 2^{\ell}$ and $w(C) = 2^{2^{\ell}-1}$
Then we let $z_{\downarrow}(C) = z_{\downarrow}(C')$, $x_{\min}(C) = x_{\min}(C')$ and calculate $z_{\uparrow}(C), x_{max}(C)$ following the definition. C will be our root cell and ℓ the level

Spliting criteria

For a cell C' if $h(C') \leq 1$ we split into 4 cells along the axis-parallel lines $(\frac{x_{min}+x_{max}}{2}, \sqrt{z^{\downarrow}(C')z^{\uparrow}(C')})$. Otherwise we first split along $z = \sqrt{z^{\downarrow}(C')z^{\uparrow}(C')}$ then we split the bottom cell into $2^{\frac{h(C')}{2}}$ cells with equal width.

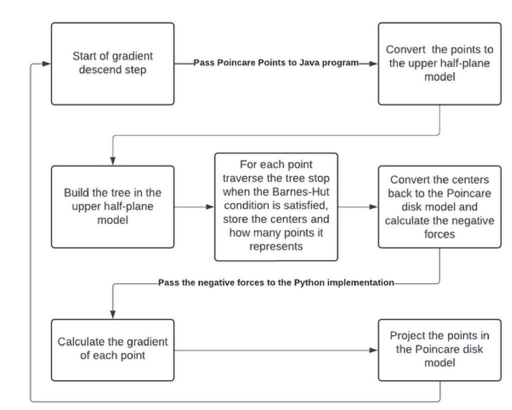


Einstein midpoint

$$m(\{v_j\}) = \sum_j \left(\frac{\gamma(v_j)}{\sum_l \gamma(v_l)} \right) v_j$$

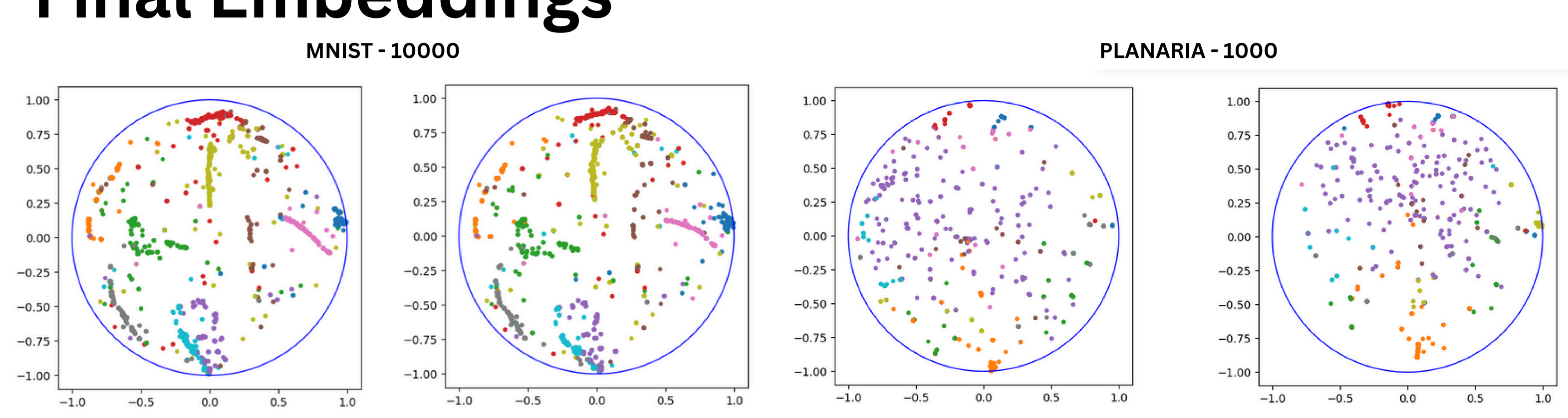
where $\gamma(v_j) = \frac{1}{\sqrt{1 - \|v_j\|^2}}$ and v_j is the coordinate of the point in Klein model

Integrating the quadtree in the context of hyperbolic t-SNE

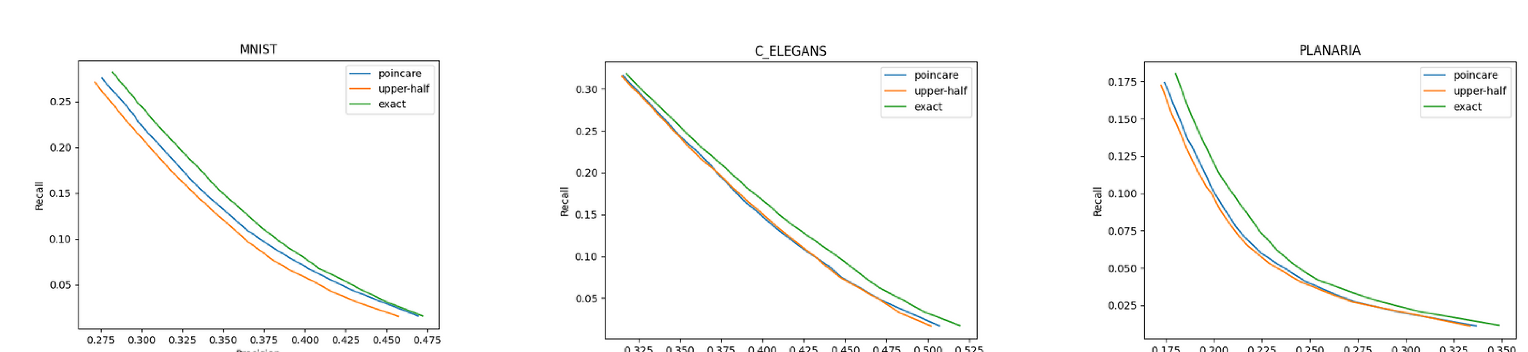


3. Results

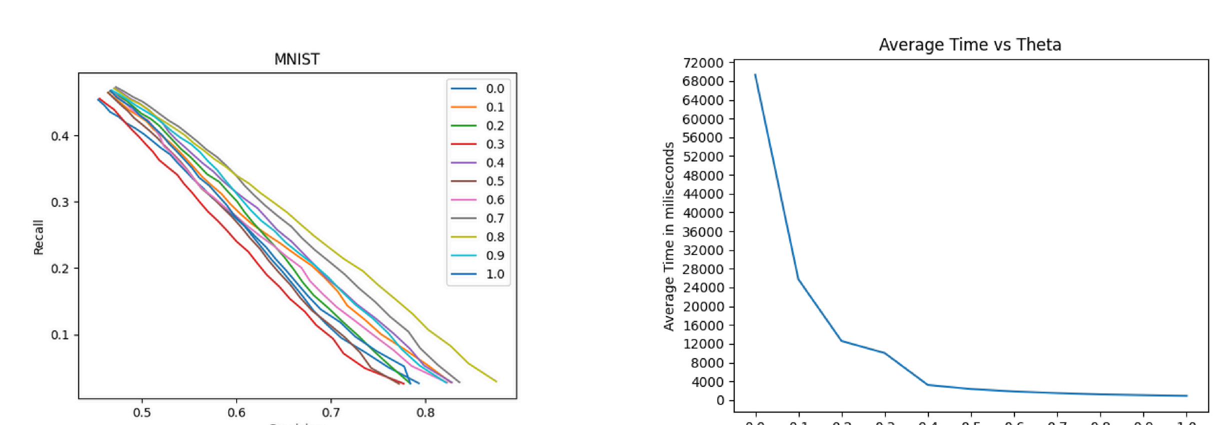
Final Embeddings



Precision and Recall



Effect of θ



Build time of the quadtree

Sample Size	Construction Time
1,000	29.95 ms
5,000	83.55 ms
10,000	120 ms
20,000	187.33 ms
40,000	308.32 ms

Time for calculating the negative forces

Sample Size	Negative forces time calculation
1,000	254.75 ms
5,000	1043.28 ms
10,000	2275.40 ms
20,000	5031.23 ms
40,000	12534.54 ms

4. Conclusion

Limitations:

- Current implementation is not fast because the comunication between the Python and JAVA relies on reading and writing files

Future works:

- Implement the quadtree in C++ to avoid reading/writing overhead
- Calculate the gradient and project points in the upper half - plane model
- Compare the method to other existing solution

References

[1] Martin Skrodzki, Hunter van Geffen, Nicolas F. Chaves-de-Plaza, Thomas Höllt, Elmar Eisemann, Klaus Hildebrandt. Accelerating hyperbolic t-SNE.

[2] Sándor Kisfaludi-Bak, Geert van Wordragen. A Quadtree, a Steiner Spanner, and Approximate Nearest Neighbours in Hyperbolic Space.

[3] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(86):2579–2605, 2008.

[4] Barnes, J., Hut, P. A hierarchical O(N log N) force-calculation algorithm. Nature 324, 446–449 (1986). https://doi.org/10.1038/324446a0

Responsible professor: Martin Skrodzki

Supervisor: Elmar Eisemanni