Geometry of Complex Networks

Savannah Crawford, Jae-Moon Hwang
Matt Holzer

George Mason University, MEGL

May 3, 2019

Table of Contents

Networks

- A network is graph, $G = (V, E)$
- with adjacency matrix A such that $A_{ij} = 1$ if nodes i and j are connected, and $A_{ij} = 0$ if not.

Complex Networks

- non-trivial features that do not occur in simple networks such as lattices or random graphs but often occur in graphs modelling real systems.
- General features of complex networks: sparse, algebraic degree distribution (i.e. hubs exist), community structure, exponential growth of neighbors.

- a map from nodes to points in a metric space:

$$f : V \to \Omega^n$$

such that $f(v_i) = x_i$

Embedding Options

Isometric Embedding

- Original distances between nodes are preserved in embedding:

$$\delta(v_i, v_j) = d_\Omega(f(v_i), f(v_j))$$

Similarity Embedding

- If nodes are connected in graph, they are close in the embedding
- If nodes are not connected in graph, they are far apart in embedding
- α is a chosen parameter to determine closeness

$$d_\Omega(f(v_i), f(v_j)) < \alpha \quad \text{if } A_{ij} = 1$$

$$d_\Omega(f(v_i), f(v_j)) > \alpha \quad \text{if } A_{ij} = 0$$

Why would you ditch isomorphic embeddings?

Because not all graphs have isometric embeddings in a given metric space.

Example

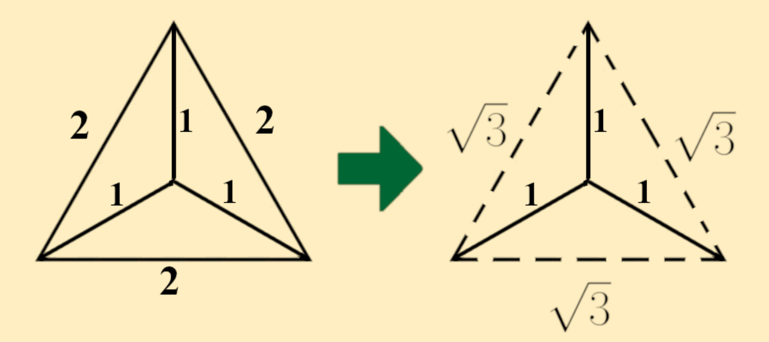


Figure: The above example does not have an isometric embedding in $\mathbb{R}^2$, but it does have a similarity embedding with $\alpha = 1$

Similarities between Complex Networks and Hyperbolic Space

- We will use hyperbolic space $\mathbb{H}^n$.
- Many real work networks share similarities with trees and trees are naturally related to hyperbolic space.
- Previous studies have demonstrated that real work networks have hyperbolic features (thin triangles for example) and our work last semester showed that complex networks behave like trees.

Why Hyperbolic space?

Table: Characteristics of Euclidean, spherical, and hyperbolic geometries

Property	**Euclidean**	**Spherical**	**Hyperbolic**
Curvature K	0	>0	<0
Parallel lines	1	0	∞
Triangles are	Normal	Thick	Thin
Sum of angles	π	$> \pi$	$< \pi$

Star Graphs in $\mathbb{R}^2$ vs $\mathbb{H}^2$

A star graph is a graph where one node, call this the parent, is connected to every other node, and all other nodes, call them children, are only connected to the parent node.

In $\mathbb{R}^2$, the largest graph that embeds has only 6 children, regardless of α. In $\mathbb{H}^2$, the largest graph that embeds increases as α increases.

α	1	1.5	1.75	2	3	4	5
Max # of Children	6	6	8	9	14	23	38

Table: The max number of children for a star graph in $\mathbb{H}^2$ given α

Models for $\mathbb{H}^2$

- We considered two models for hyperbolic space, Poincaré's Upper Half-Plane Model and Disk Model.
- The Upper Half-Plane Model is defined as $\{(x,y) \mid y > 0 \text{ and } x,y \in \mathbb{R}\}$
- The Disk Model is defined as all the points within the unit circle centered at the origin.
- Both are models for 2-dimensional hyperbolic space.
- The Upper Half-Plane model is easier to do computations with, but the Disk Model is better for visualization.
- Fortunately, the models are isomorphic, and we can convert between both models easily.

Distance in $\mathbb{H}_{PD}$

- $d_H(p_1, p_2)$ is the distance function for our metric space.
- For the Poincaré disk model, $d_H(p_1, p_2)$ is

$$arcosh\left(1 + \frac{2((a-x)^2 + (b-y)^2)}{(1-x^2-y^2)(1-a^2-b^2)}\right)$$

where $p_1 = (x, y)$ and $p_2 = (a, b)$.

Cost Function

Let A be the adjacency matrix of a network of n vertices and let B be the 'anti' matrix such that $A + B = 1_{nxn}$.

$$J_i = \sum_{j=1}^{n} A_{ij}\Phi(d_H(x_i, x_j)^2) + \sum_{j=1}^{n} B_{ij}\Psi(d_H(x_i, x_j)^2)$$

Such that $\Phi(z)$, $\Psi(z)$, and their derivatives ($\phi(z)$ and $\psi(z)$ respectively) are as follows:

$$\Phi(z) = \begin{cases} z - \alpha^2 & z > \alpha \\ 0 & z \leq \alpha \end{cases}, \quad \phi(z) = \begin{cases} 1 & z > \alpha \\ 0 & z \leq \alpha \end{cases}$$

$$\Psi(z) = \begin{cases} \alpha^2 - z & z \leq \alpha^2 \\ 0 & z > \alpha^2 \end{cases}, \quad \psi(z) = \begin{cases} -1 & z \leq \alpha \\ 0 & z > \alpha \end{cases}$$

Gradient Descent

Data: A, adjacency matrix of G, with n nodes. α is the max distance for two connected points. *data* is a set of n random points in $\mathbb{H}^2$

Result: *data* which is a $nx2$ array for which the constraints for a similarity are met

$W := A - B$;
while $p_{norm} > threshold$ **do**
    $p := 0_{n,2}$;
    for $1 \leq i \leq n$ **do**
        for $i < j \leq n$ **do**
            if $d_H(i,j) > \alpha$ and $W_{ij} = 1$ **OR** $d_H(i,j) < \alpha$ and $W_{ij} = -1$ **then** $p_i := p_i - W_{ij} * \nabla J_{ij}$;
        end
    end
    $data := data + p * dt$;
    $p_{norm} := norm(p, 2)$
end

Simple geodesic

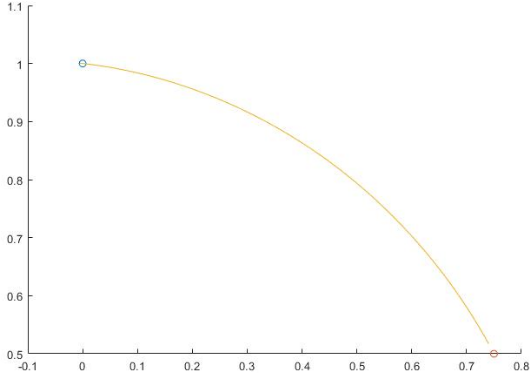
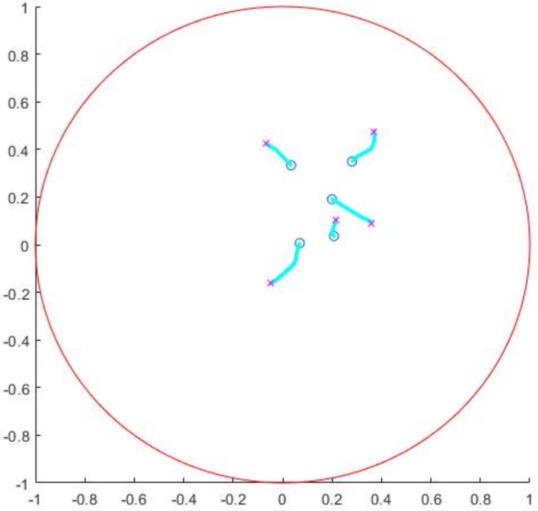


Figure: When two points move toward each other guided by the gradient, they move along the geodesic

Example

- The previous slide shows the geodesic (shortest path) between two points in hyperbolic space.
- As mentioned before, that distance is a curve when viewed in Euclidean space.
- Our algorithm is able to find this geodesic with given any two points.
- This then gives us a way to find a minimal cost embedding of graphs in a given metric space.

Star Graph Example



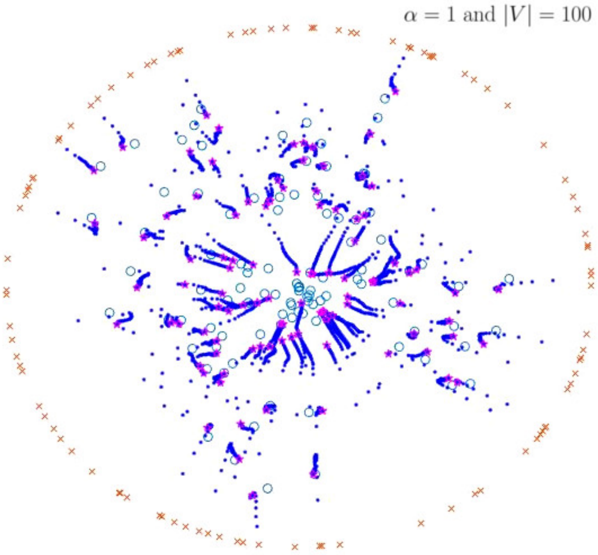
Star Graph Example (cont.)

- The smaller blue circles are random points. The red x's are where they end up after we put the points through our algorithm.
- The paths that the points take are shown by the blue lines.
- The algorithm obtains a low cost embedding of the initial points using an adjacency matrix that represents a star graph.

Gradient Descent with Angular Component

- Gradient Descent issue: There are many local minimums, and/or it is hard to select "good" initial conditions.
- Possible resolution: Use knowledge about the angular component of a node to choose better initial conditions.
- This relies on each node having an angular component associated with it, so it will only work for some networks, such as the airline transportation network.
- Begin with each node at $(r\cos\theta, r\sin\theta)$ with some fixed r and θ corresponding to that node (for example geographic location)
- Run the algorithm as before
- Potentially works better because it prevents enemy nodes from meeting

With known angular component



With known angular component (cont.)

- We generated 100 random points as a representative graph. We then computed the distances between every pair of points to see whether or not they are of a certain distance α.
- If the distances between two points is $\leq \alpha$, then they are connected. If not, then they are considered disconnected. This creates our adjacency matrix of the representative graph.
- From here, we preserve the angular component, but "forget" the radial component of each point.
- We set all the points at the same radius but with their respective angular component. We then run them through our algorithm to see if it can "learn" a similar graph. Our algorithm found a zero cost embedding after 62 iterations.

So how do we find an angular component in the real world?

- For an airplane network, the idea is relatively simple.
- Using the geographic locations of each airport, we can then find an angular component using the Earth as our "disk" (or ball).
- However, this means we need to expand into H^3 with the Poincare disk model.

Going into $\mathbb{H}^3$

- For future work, we would most likely need to go up a dimension from $\mathbb{H}^2$ to $\mathbb{H}^3$.
- Fortunately, the Poincaré disk model can be extended up to another dimension. In that case, the bounds of the model become of the unit ball.
- We would then need to recompute gradients and parts of our code so that it reflects that third component.

References

1. Poincarè disk model,
   https://en.wikipedia.org/wiki/Poincarè_disk_model (updated Jan 2019)
2. Large-scale curvature of networks. Narayan, Onuttom and Saniee., Iraj. Phys. Rev. E 84, 066108 Published 13 December 2011
3. Weisstein, Eric W. "Hyperbolic Geometry." From MathWorld–A Wolfram Web Resource.
   http://mathworld.wolfram.com/HyperbolicGeometry.html