Geometry of Complex Networks

Savannah Crawford, Jae-Moon Hwang

Mason Experimental Geometry Lab

George Mason University
College of Science

April 25, 2019

Abstract

A complex network is a graph, $G = (V, E)$ consisting of E edges and V vertices with non-trivial features, such as sparseness, algebraic degree distribution, community structure, and exponential growth of neighbors, that often occur in graphs modeling real world systems. We aim to study the geometry of complex networks by developing an embedding, or mapping, to a metric space that preserves its topological properties. We consider both isometric embeddings, which preserve distances, and 'similarity' embeddings, which relax the isometric constraints of the embedding. Our goal is to embed complex networks into hyperbolic space, and we will explore the possible applications of this work.

Definitions

Network

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

Complex Network

A **complex network** is a network with non-trivial features that do not occur in simple networks such as lattices or random graphs but often occur in graphs modeling real systems. General features of complex networks include sparsity, algebraic degree distribution, community structure, exponential growth of neighbors.

Embedding

We want to study the geometry of complex networks by finding a map from nodes to points in a metric space: $f : V \to \Omega^n$ such that $f(v_i) = x_i$.

Isometric Embedding

Original distances between nodes are preserved in embedding: $\delta(v_i, v_j) = d_\Omega(f(v_i), f(v_j))$

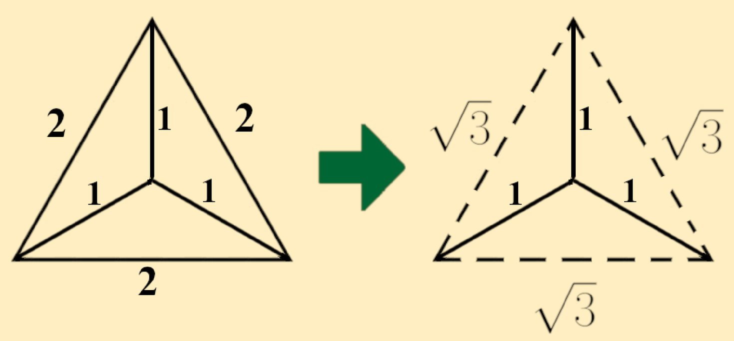
'Similarity' Embedding

- If nodes are connected in graph, then they are close in the embedding.
- If nodes are not connected in graph, then they are far apart in embedding.

$$d_\Omega(f(v_i), f(v_j)) \leq \alpha \quad \text{if } A_{ij} = 1$$
$$d_\Omega(f(v_i), f(v_j)) > \alpha \quad \text{if } A_{ij} = 0$$

Example: Star Graph

Some graphs cannot embed into any metric space isometrically, but they can be embedded with a similarity embedding. For the graph below, if we let $\alpha = 1$, we can easily get a similarity embedding.



Why 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).

Poincaré Disk Model

The disk model is defined as $\{(x, y) | \sqrt{x^2 + y^2} < 1, x, y \in \mathbb{R}\}$ with the origin at $(0, 0)$. This model has a more expensive distance metric, but it is much nicer for visualization.

Similarity Embedding

Since we are not guaranteed the existence of an isomorphic function $f : A \to \mathbb{H}^2$, we proceed with the similarity embedding. We can associate a "cost" with an embedding to determine how close it is to ideal. To minimize the cost of our function, we move points with respect to the gradient of the cost function.

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:

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

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

where α is our parameter for connection. The first sum penalizes connected nodes that are too far away, while the second sum penalizes disconnected nodes that are embedded too closely.

Distance Function

$$d_H(p_1, p_2) = arcosh\left(1 + \frac{2\left((a - x)^2 + (b - y)^2\right)}{(1 - x^2 - y^2)(1 - a^2 - b^2)}\right)$$

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

Gradient Descent Algorithm

In order to minimize the 'cost' of our embedding, we use the gradient descent algorithm.

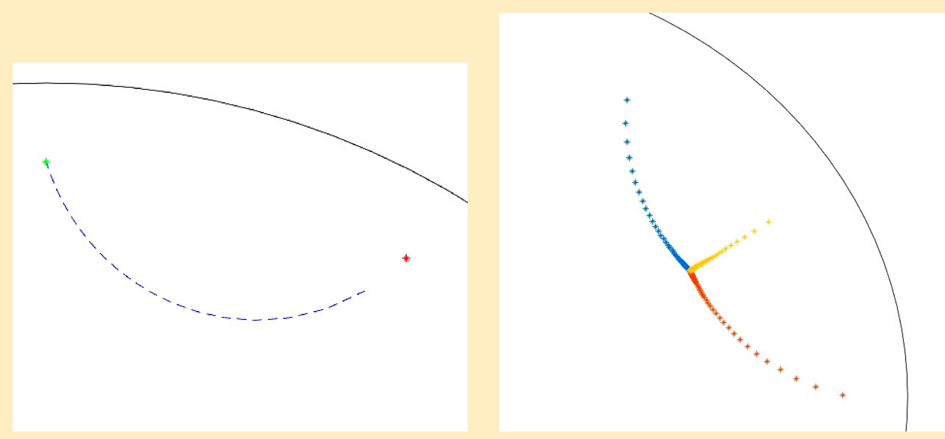
Data: A, adjacency matrix of G, with n nodes. α is the max distance for two connected points. *data* is a set of n random points in Ω^N

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 ≤ i ≤ n do
    for i < j ≤ n do
      if d_H(i,j) > α and W_ij = −1 OR d_H(i,j) < α and W_ij = 1
      then p_i := p_i − W_ij ∗ ∇J_ij ;
    end
  end
  data := data + p ∗ dt;
  p_norm := norm(p, 2)
end
```

Example: Points travel along geodesic

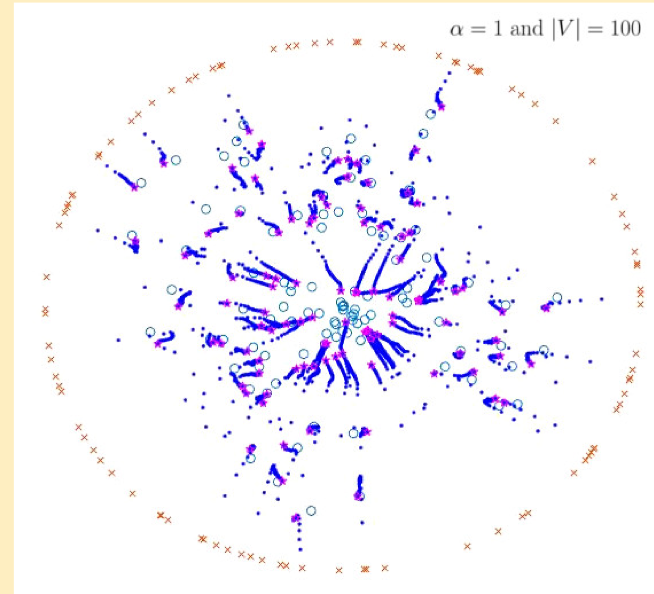
If we embed a graph of two connected points and start with initial distance $> \alpha$, the points will move toward each other along the geodesic, the shortest path between the two points.



Similarly, three points move along the geodesic toward the center of the triangle whose vertices are the original points.

Embedding with known angular component

Some networks have a known geographic locations that may correspond to their position in an embedding. From this geographic component, we can place each node at the boundary with a circle. This gives a better initial condition for the embedding and leads to faster convergence. Below is an example of 100 nodes that converged after 62 iterations.



Future Work

For some initial conditions, the algorithm never converges, even if there exists a solution. This is because the gradient has found a local minimum. We want to explore ways to better guess initial conditions to avoid local minimums and converge faster. Once our algorithm consistently finds low cost embeddings, we want to use it to find embeddings of real world networks, such as the airline transportation network.

Acknowledgments

We would like to thank our Faculty Mentor, Matt Holzer for leading this research project for us.
An additional thank you to Chris Thornton who contributed to this research project in Fall 2018.

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