Graph Drawing using the Laplacian

Based on Speilamn, Chapter 3

Gil Cohen

November 2, 2020

Overview

Embedding a graph to $\mathbb{R}$

Say we wish to embed a graph G to the reals. How should we go about it? Connected vertices should be close, so we can express this as minimizing

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{uv \in E} (\mathbf{x}(u) - \mathbf{x}(v))^2,$$

where $\mathbf{x}$ is the embedding.

Question

What is the obvious issue with this suggestion?

Embedding a graph to $\mathbb{R}$

So we normalize so that the points in $\mathbf{x}$ are not too concentrated around any point.

$$\forall p \in \mathbb{R} \quad \sum_{v \in V} (\mathbf{x}(v) - p)^2 \geq 1.$$

Question

What can we say about p?

Embedding a graph to $\mathbb{R}$

So we normalize so that the points in $\mathbf{x}$ are not too concentrated around a point.

$$\forall p \in \mathbb{R} \quad \sum_{v \in V} (\mathbf{x}(v) - p)^2 \geq 1.$$

$$p = \mathbb{E}_v \mathbf{x}(v) = \mathbf{u}^T \mathbf{x}$$

where $\mathbf{u} = \frac{1}{n} \cdot \mathbf{1}$. By shifting, we may as well assume $\mathbf{u}^T \mathbf{x} = 0$.

Embedding a graph to $\mathbb{R}$

Hence, we want to minimize $\mathbf{x}^T \mathbf{M}\mathbf{x}$ subject to $\mathbf{u}^T\mathbf{x} = 0$ and $\|\mathbf{x}\|_2 = 1$.

Question

Who is $\mathbf{x}$*?*

Examples

Figure: 20-vertex path graph embedded into $\mathbb{R}$.

Figure: 20-vertex cycle graph embedded into $\mathbb{R}$.

Examples

Figure: Depth-4 complete binary tree embedded into $\mathbb{R}$.

Examples

Figure: Third-dumbbell embedded into $\mathbb{R}$.

Examples

Figure: 5-clique embedded into $\mathbb{R}$.

Examples

Figure: Degree 2 plus symmetrization random graph embedded into $\mathbb{R}$.

My modest code

```
using LinearAlgebra
using Plots
using Luxor


function draw_on_line(M)
    n = size(M)[1]
    E = eigen(M)
    v2 = E.vectors[:,n-1]
    s = 300

    @png begin
        for i in 1:n
            circle(Point(s*v2[i],0),2,:fill)
        end

        sethue("gray")
        setline(1)
        for i in 1:n
            for j in i:n
                if (M[i,j] == 1)
                    A = Point(s*v2[i], 0)
                    B = Point(s*v2[j], 0)
                    C = Point((A.x+B.x)/2,-80+rand(-40:1:40))
                    curve(A, C, B)
                    strokepath()
                end
            end
        end
    end
end

P = clique_no_loops(10)
```

Drawing a graph on the plane

Moving to two dimensions, we now wish to find a pair $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ that minimizes

$$\sum_{uv \in E} \left\| \begin{pmatrix} \mathbf{x}(u) \\ \mathbf{y}(u) \end{pmatrix} - \begin{pmatrix} \mathbf{x}(v) \\ \mathbf{y}(v) \end{pmatrix} \right\|_2^2 \tag{1}$$

subject to $\mathbf{u}^T \mathbf{x} = \mathbf{u}^T \mathbf{y} = 0$ and $\|\mathbf{x}\| = \|\mathbf{y}\| = 1$.

$(1) = \mathbf{x}^T \mathbf{L} \mathbf{x} + \mathbf{y}^T \mathbf{L} \mathbf{y}$

Question

What other condition you would suggest we include?

This is precisely Hall's idea.

A lower bound

Theorem

Let $\mathbf{L}$ be a Laplacian matrix. For $k \geq 1$ let $\mathbf{x}_1, \ldots, \mathbf{x}_k$ be orthonormal vectors that are all orthogonal to $\mathbf{1}$. Then,

$$\sum_{i=1}^{k} \mathbf{x}_i^T \mathbf{L} \mathbf{x}_i \geq \sum_{i=2}^{k+1} \lambda_i .$$

Some room for the proof

Examples

MANAGEMENT SCIENCE
Vol. 17, No. 3, November, 1970
Printed in U.S.A.

AN r-DIMENSIONAL QUADRATIC PLACEMENT ALGORITHM*

KENNETH M. HALL†

State of California, Department of General Services

In this paper the solution to the problem of placing n connected points (or nodes) in r-dimensional Euclidean space is given. The criterion for optimality is minimizing a weighted sum of squared distances between the points subject to quadratic constraints of the form $X'X = 1$, for each of the r unknown coordinate vectors. It is proved that the problem reduces to the minimization of a sum or r positive semi-definite quadratic forms which, under the quadratic constraints, reduces to the problem of finding r eigenvectors of a special "disconnection" matrix. It is shown, by example, how this can serve as a basis for cluster identification.

Examples

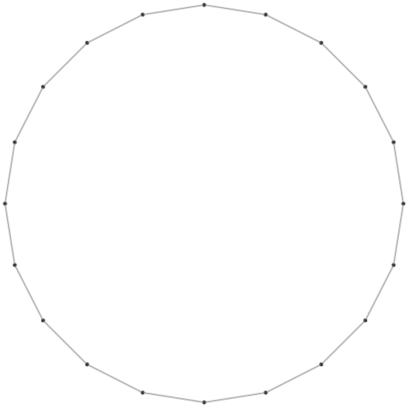

Figure: 20-cycle embedded into $\mathbb{R}^2$.

Examples

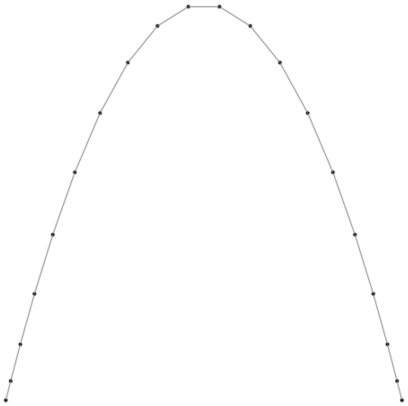

Figure: 20-vertex path embedded into $\mathbb{R}^2$.

Examples

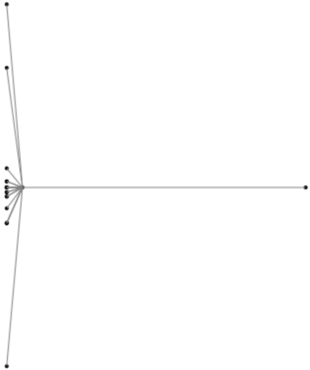

Figure: 20-vertex star graph embedded into $\mathbb{R}^2$.

Examples

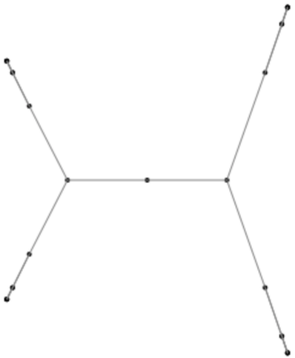

Figure: Depth-4 complete binary tree embedded into $\mathbb{R}^2$.

Examples

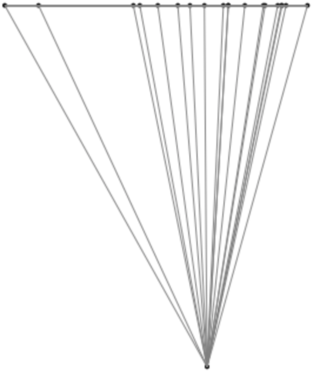

Figure: 20-vertex clique embedded into $\mathbb{R}^2$.

Examples

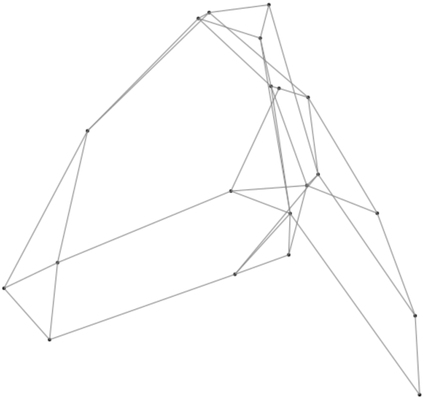

Figure: 20-vertex random graph (degree 2 symmetrized) embedded into $\mathbb{R}^2$.

Going 3D!

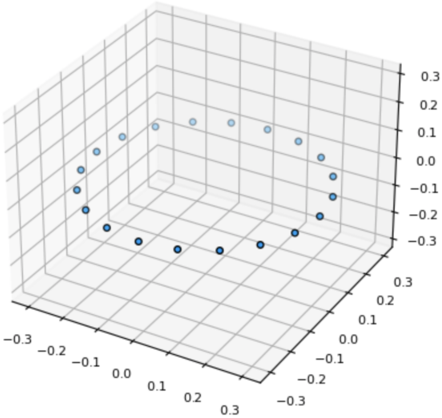

Figure: 20-cycle embedded into $\mathbb{R}^3$.

Going 3D!

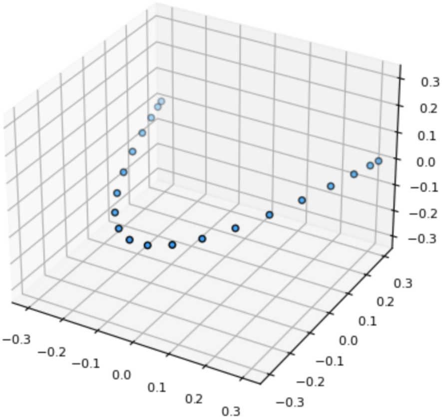

Figure: 20-vertex path embedded into $\mathbb{R}^3$.

Going 3D!

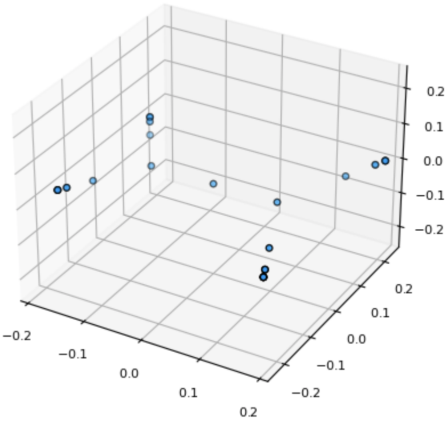

Figure: Depth-4 complete binary tree embedded into $\mathbb{R}^3$.

Going 3D!

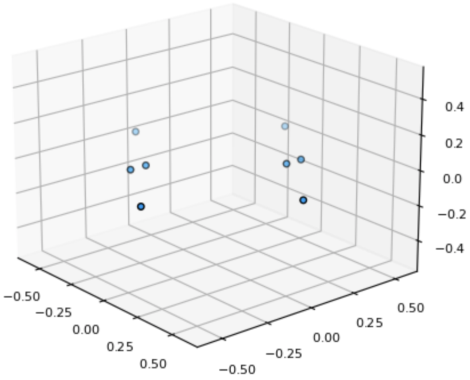

Figure: Who am I?

My modest code

```
function draw3d(M)
    n = size(M)[1]
    E = eigen(M)
    v2 = E.vectors[:,n-1]
    v3 = E.vectors[:,n-2]
    v4 = E.vectors[:,n-3]
    Plots.scatter(v2, v3, v4, leg = false, camera = (50,20))
end

draw3d(lap(cube()))
```