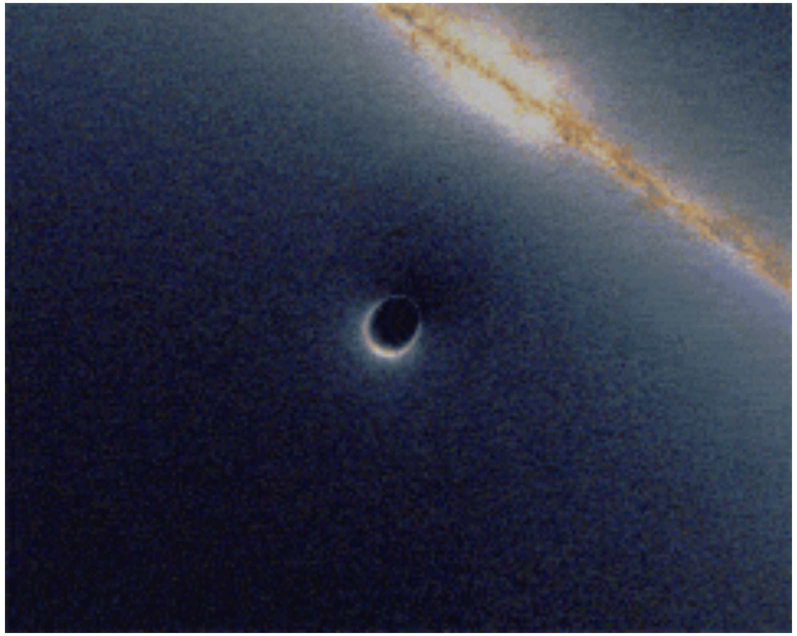
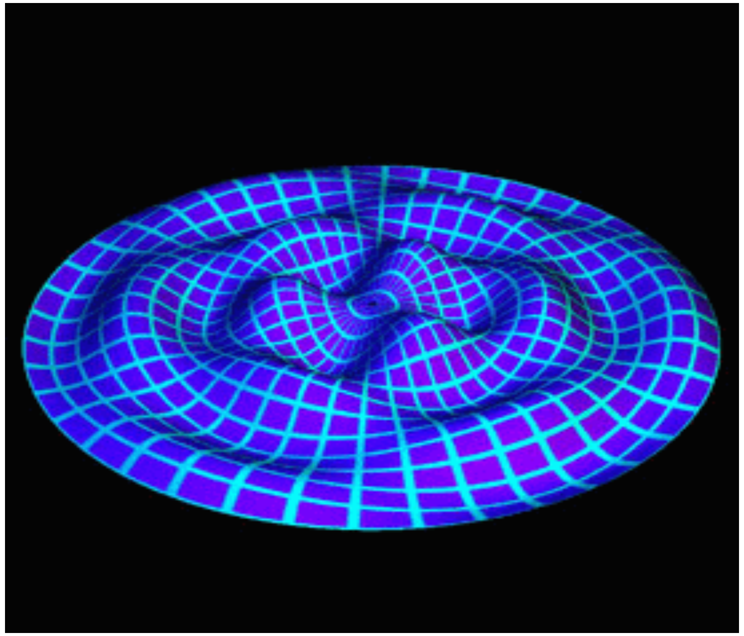
Graphs: Geometry, Operators, Spectra, and Kernels

Mike Slawinski

July 14, 2017





Graph Theory – Applications to Cylance Data

- Graph Structured Data

Graph Theory – Applications to Cylance Data

- Graph Structured Data
  - one-hot encoded vectors (nodes in a hypercube, which is itself a graph)

Graph Theory – Applications to Cylance Data

- Graph Structured Data
  - one-hot encoded vectors (nodes in a hypercube, which is itself a graph)
  - optics process/network data

Graph Theory – Applications to Cylance Data

- Graph Structured Data
  - one-hot encoded vectors (nodes in a hypercube, which is itself a graph)
  - optics process/network data
  - identity process/network data

Graph Theory – Applications to Cylance Data

- Graph Structured Data
  - one-hot encoded vectors (nodes in a hypercube, which is itself a graph)
  - optics process/network data
  - identity process/network data
  - graphs arising from decompilation (control flow, function to function, etc.)

Graph Theory – Applications to Cylance Data

- Graph Structured Data
  - one-hot encoded vectors (nodes in a hypercube, which is itself a graph)
  - optics process/network data
  - identity process/network data
  - graphs arising from decompilation (control flow, function to function, etc.)
- What would we like to be able to do with this data?

Graph Theory – Applications to Cylance Data

- Graph Structured Data
  - one-hot encoded vectors (nodes in a hypercube, which is itself a graph)
  - optics process/network data
  - identity process/network data
  - graphs arising from decompilation (control flow, function to function, etc.)
- What would we like to be able to do with this data?
  - compare nodes within a graph
    - clustering in hypercube space

Graph Theory – Applications to Cylance Data

- Graph Structured Data
  - one-hot encoded vectors (nodes in a hypercube, which is itself a graph)
  - optics process/network data
  - identity process/network data
  - graphs arising from decompilation (control flow, function to function, etc.)
- What would we like to be able to do with this data?
  - compare nodes within a graph
    - clustering in hypercube space
    - compare processes/network events in a Markov model

Graph Theory – Applications to Cylance Data

- Graph Structured Data
  - one-hot encoded vectors (nodes in a hypercube, which is itself a graph)
  - optics process/network data
  - identity process/network data
  - graphs arising from decompilation (control flow, function to function, etc.)
- What would we like to be able to do with this data?
  - compare nodes within a graph
    - clustering in hypercube space
    - compare processes/network events in a Markov model
  - compare graphs to each other – compare files via control flow comparison

Graph Theory – Applications to Cylance Data

- Graph Structured Data
  - one-hot encoded vectors (nodes in a hypercube, which is itself a graph)
  - optics process/network data
  - identity process/network data
  - graphs arising from decompilation (control flow, function to function, etc.)
- What would we like to be able to do with this data?
  - compare nodes within a graph
    - clustering in hypercube space
    - compare processes/network events in a Markov model
  - compare graphs to each other – compare files via control flow comparison

Solution: Graph Kernels (measure node similarity and graph similarity)

Kernels and Vectorization

Definition: A kernel K on a space Ω is a function $K: \Omega \times \Omega \to \mathbb{R}$, which is meant to measure the similarity between elements $x, x' \in \Omega$.

Kernels and Vectorization

Definition: A kernel K on a space Ω is a function $K: \Omega \times \Omega \to \mathbb{R}$, which is meant to measure the similarity between elements $x, x' \in \Omega$.

There is an implicit feature map $\phi: \Omega \to \mathcal{H}_K$ to a Hilbert space $\mathcal{H}_K$, in which the kernel appears as the inner product $K(x, x') =< \phi(x), \phi(x') >_{\mathcal{H}_K}$

Kernels and Vectorization

Definition: A kernel K on a space Ω is a function $K: \Omega \times \Omega \to \mathbb{R}$, which is meant to measure the similarity between elements $x, x' \in \Omega$.

There is an implicit feature map $\phi: \Omega \to \mathcal{H}_K$ to a Hilbert space $\mathcal{H}_K$, in which the kernel appears as the inner product $K(x, x') =< \phi(x), \phi(x') >_{\mathcal{H}_K}$

A Hilbert space H_K is a real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product.

Kernels and Vectorization

Definition: A kernel K on a space Ω is a function $K: \Omega \times \Omega \rightarrow \mathbb{R}$, which is meant to measure the similarity between elements $x, x' \in \Omega$.

There is an implicit feature map $\phi: \Omega \rightarrow \mathcal{H}_K$ to a Hilbert space $\mathcal{H}_K$, in which the kernel appears as the inner product $K(x, x') =< \phi(x), \phi(x') >_{\mathcal{H}_K}$

A Hilbert space H_K is a real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product.

Kernel Trick:

Example: SVM Classifier

$$z \mapsto sgn(w \cdot \varphi(z) + b) = sgn([\sum_{i=1}^{n} c_i y_i k(x_i, z)] + b) \text{ , where } w = \sum_{i=1}^{n} c_i y_i \varphi(x_i)$$

Idea is to compute similarity without actually mapping to a higher dimensional space.

Graph Kernels

Two Types:

Type I: Measure similarity between nodes based on edge structure of the graph

Graph Kernels

Two Types:

Type I: Measure similarity between nodes based on edge structure of the graph

Type II: Measure similarity between graphs based on edge structure, labeling, etc.

Kernel Development: Graph Theoretic Analogues to Smooth Manifolds

Graph Theory	Smooth Manifolds
Functions: $f: Vert(G) \rightarrow \mathbb{R}$	Functions: $f: M \rightarrow \mathbb{R}$
Variable Node Connectivity	Variable Curvature
Laplacian L	Laplacian Δ
PDEs (Heat, Wave)	PDEs (Heat, Wave)

Curvature – Comparing f to its Average

The curvature κ of a curve given by $y = f(x)$ is given by $\kappa = \frac{|y''|}{(1+y'^2)^{\frac{3}{2}}}$

Curvature – Comparing f to its Average

The curvature κ of a curve given by $y = f(x)$ is given by $\kappa = \frac{|y''|}{(1+y'^2)^{\frac{3}{2}}}$

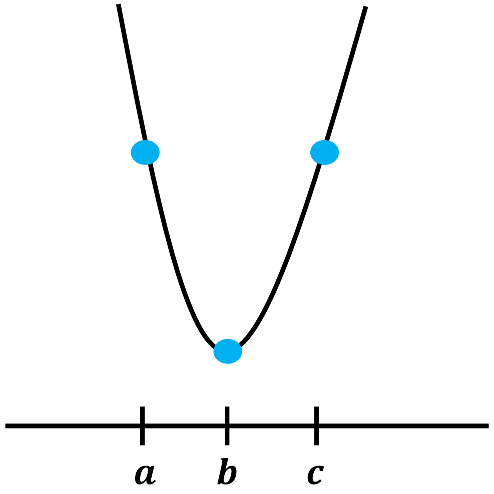
The Laplacian operator $\Delta f := \frac{\partial^2}{\partial x^2} f$ serves as a proxy for curvature when $\left|\frac{\partial}{\partial x} f \;-1\right| < \epsilon$

Curvature – Comparing f to its Average

The curvature κ of a curve given by $y = f(x)$ is given by $\kappa = \frac{|y''|}{(1+y'^2)^{\frac{3}{2}}}$

The Laplacian operator $\Delta f \coloneqq \frac{\partial^2}{\partial x^2} f$ serves as a proxy for curvature when $\left|\frac{\partial}{\partial x} f \; - 1\right| < \epsilon$

High Curvature at b



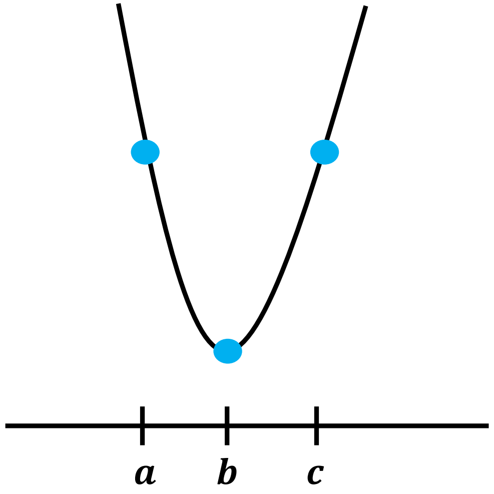
$$\boldsymbol{f(b) \ll \frac{1}{2}(f(a) + f(c))}$$

Curvature – Comparing f to its Average

The curvature κ of a curve given by $y = f(x)$ is given by $\kappa = \frac{|y''|}{(1+y'^2)^{\frac{3}{2}}}$

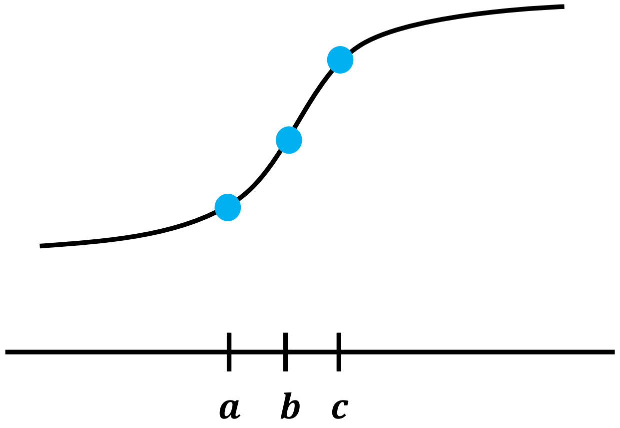
The Laplacian operator $\Delta f \coloneqq \frac{\partial^2}{\partial x^2} f$ serves as a proxy for curvature when $\left|\frac{\partial}{\partial x} f \; - 1\right| < \epsilon$

High Curvature at b



$\boldsymbol{f(b) \ll \frac{1}{2}(f(a) + f(c))}$

Low Curvature at b



$\boldsymbol{f(b) \approx \frac{1}{2}(f(a) + f(c))}$

Curvature – Comparing f to its Average

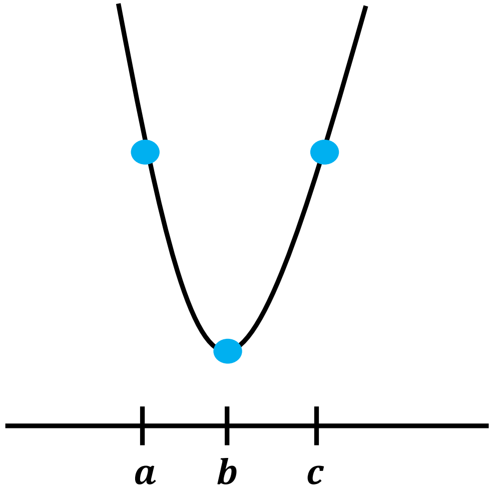
The curvature κ of a curve given by $y = f(x)$ is given by $\kappa = \frac{|y''|}{(1+y'^2)^{\frac{3}{2}}}$

The Laplacian operator $\Delta f := \frac{\partial^2}{\partial x^2} f$ serves as a proxy for curvature when $\left|\frac{\partial}{\partial x} f \ - 1\right| < \epsilon$

Key Point

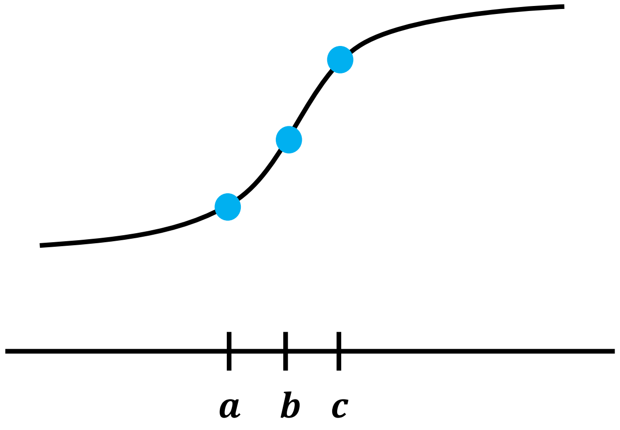
$\Delta f|_{\mathrm{b}}$ measures the extent to which $f(b)$ differs from $avg_x\ f(x)$ for x in a local spherical shell centered at b

High Curvature at b



$$\boldsymbol{f(b) \ll \frac{1}{2}(f(a) + f(c))}$$

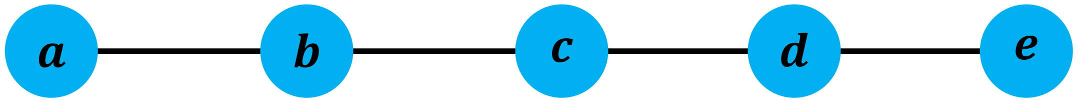
Low Curvature at b



$$\boldsymbol{f(b) \approx \frac{1}{2}(f(a) + f(c))}$$

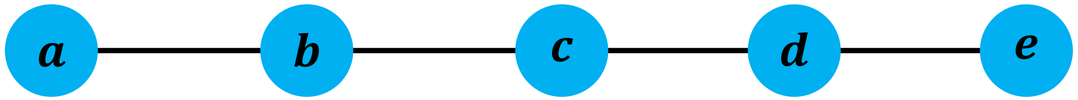
Curvature – Comparing f to its Average (Discrete Version)

Consider the graph $G =$



and a function $f : Vert(G) \to \mathbb{R}$ i.e., a vector $\big(f(a), f(b), f(c), f(d), f(e)\big) \in \mathbb{R}^5$

Curvature – Comparing f to its Average (Discrete Version)

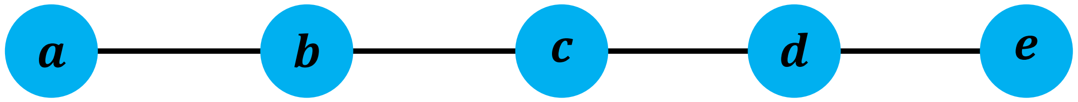
Consider the graph $G =$ 

and a function $f\colon Vert(G) \to \mathbb{R}$ i.e., a vector $\big(f(a), f(b), f(c), f(d), f(e)\big) \in \mathbb{R}^5$

Question: How can we define notions of the second derivative and the Laplacian operator on G?

Curvature – Comparing f to its Average (Discrete Version)

Consider the graph $G =$



and a function $f : Vert(G) \to \mathbb{R}$ i.e., a vector $\big(f(a), f(b), f(c), f(d), f(e)\big) \in \mathbb{R}^5$

Question: How can we define notions of the second derivative and the Laplacian operator on G?

Answer: Think of G as a discrete version of the real line and compute a difference quotient of difference quotients.

Curvature – Comparing f to its Average (Discrete Version)

Consider the graph $G = \mathbf{a} - \mathbf{b} - \mathbf{c} - \mathbf{d} - \mathbf{e}$

and a function $f\colon Vert(G) \to \mathbb{R}$ i.e., a vector $\big(f(a), f(b), f(c), f(d), f(e)\big) \in \mathbb{R}^5$

Question: How can we define notions of the second derivative and the Laplacian operator on G?

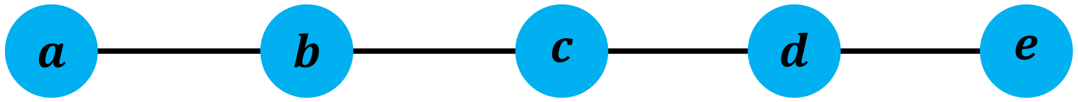
Answer: Think of G as a discrete version of the real line and compute a difference quotient of difference quotients.

$$f''(x) \approx \frac{\dfrac{f(x+\Delta x) - f(x)}{\Delta x} - \dfrac{f(x) - f(x-\Delta x)}{\Delta x}}{\Delta x} \approx \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{(\Delta x)^2}$$

Because G is a discrete structure, Δx is no longer a limiting variable, and therefore has no meaning.

Curvature – Comparing f to its Average (Discrete Version)

Consider the graph $G =$



and a function $f: Vert(G) \to \mathbb{R}$ i.e., a vector $\big(f(a), f(b), f(c), f(d), f(e)\big) \in \mathbb{R}^5$

Question: How can we define notions of the second derivative and the Laplacian operator on G?

Answer: Think of G as a discrete version of the real line and compute a difference quotient of difference quotients.

$$f''(x) \approx \frac{\dfrac{f(x+\Delta x) - f(x)}{\Delta x} - \dfrac{f(x) - f(x-\Delta x)}{\Delta x}}{\Delta x} \approx \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{(\Delta x)^2}$$

Because G is a discrete structure, Δx is no longer a limiting variable, and therefore has no meaning.

This yields $\Delta f|_c = \frac{\partial^2}{\partial x^2} f|_c = f(b) - 2f(c) + f(d)$, which measures the difference between $f(c)$ and $avg\big(f(b), f(d)\big)$

Curvature – Higher Dimensions

Curvature – Higher Dimensions

The Laplacian

Define $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \cdots + \frac{\partial^2}{\partial x_n^2}$

Curvature – Higher Dimensions

The Laplacian

Define $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \cdots + \frac{\partial^2}{\partial x_n^2}$

Application: The **Wave** Equation

Curvature – Higher Dimensions

The Laplacian

Define $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \cdots + \frac{\partial^2}{\partial x_n^2}$

Application: The **Wave** Equation

$$\frac{\partial^2 u}{\partial t^2} = \Delta u$$

Curvature – Higher Dimensions

The Laplacian

Define $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \cdots + \frac{\partial^2}{\partial x_n^2}$

Application: The **Wave** Equation

$$\frac{\partial^2 u}{\partial t^2} = \Delta u$$

Question 1: What do the solutions to this PDE look like?

Curvature – Higher Dimensions

The Laplacian

Define $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \cdots + \frac{\partial^2}{\partial x_n^2}$

Application: The **Wave** Equation

$$\frac{\partial^2 u}{\partial t^2} = \Delta u$$

Question 1: What do the solutions to this PDE look like?

Question 2: Is it possible to infer manifold structure from the structure of the operator?

Curvature – Higher Dimensions

The Laplacian

Define $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \cdots + \frac{\partial^2}{\partial x_n^2}$

Application: The **Wave** Equation

$$\frac{\partial^2 u}{\partial t^2} = \Delta u$$

Question 1: What do the solutions to this PDE look like?

Question 2: Is it possible to infer manifold structure from the structure of the operator?

Question 3: Is it possible to answer Question 2 in the graph case?

The Laplacian and the Wave Equation

Intuitively, what constraints should a wave-like function $u(x, t)$ satisfy?

The Laplacian and the Wave Equation

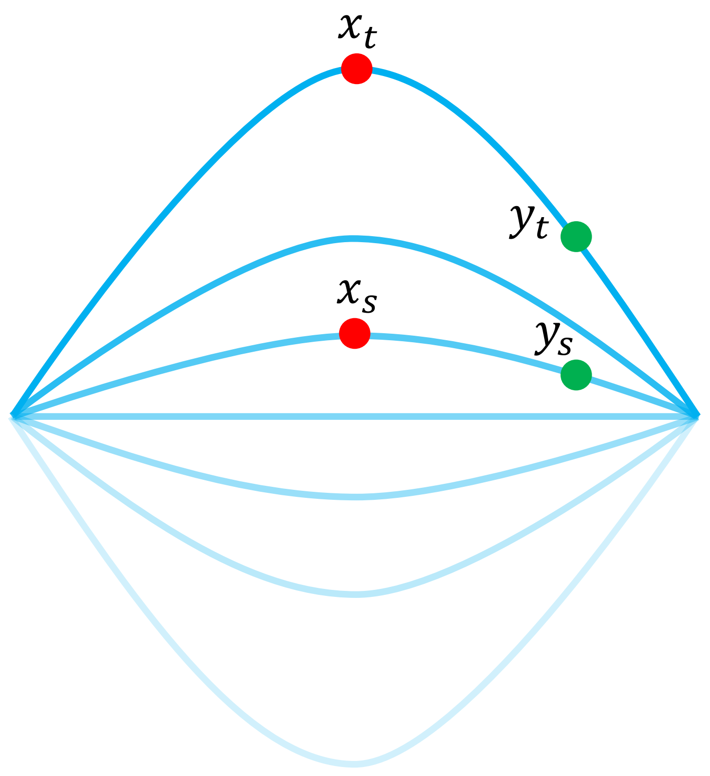
Intuitively, what constraints should a wave-like function $u(x, t)$ satisfy?

Points which are associated with high curvature should
more quickly straighten.

The Laplacian and the Wave Equation

Intuitively, what constraints should a wave-like function $u(x, t)$ satisfy?

Points which are associated with high curvature should more quickly straighten.



x_t: high curvature, high acceleration

x_t: low curvature, low acceleration

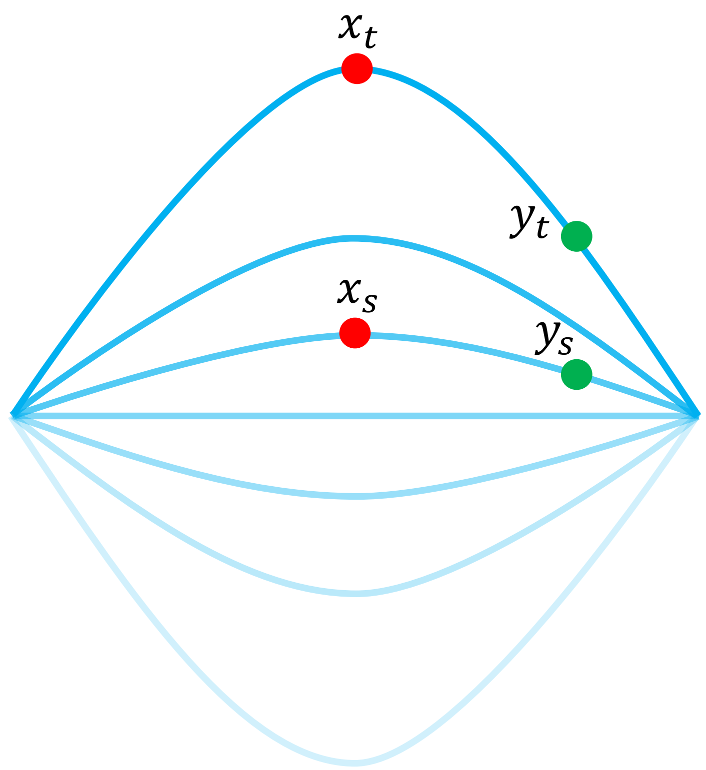
The Laplacian and the Wave Equation

Intuitively, what constraints should a wave-like function $u(x, t)$ satisfy?

Points which are associated with high curvature should more quickly straighten.

1. Curvature at x
2. Distance traveled by point x over fixed time period
3. acceleration of x

1. Δu
2. Distance traveled by point x over fixed time period
3. $\frac{\partial^2 u}{\partial t^2}$



x_t: high curvature, high acceleration

x_t: low curvature, low acceleration

The Laplacian and the Wave Equation

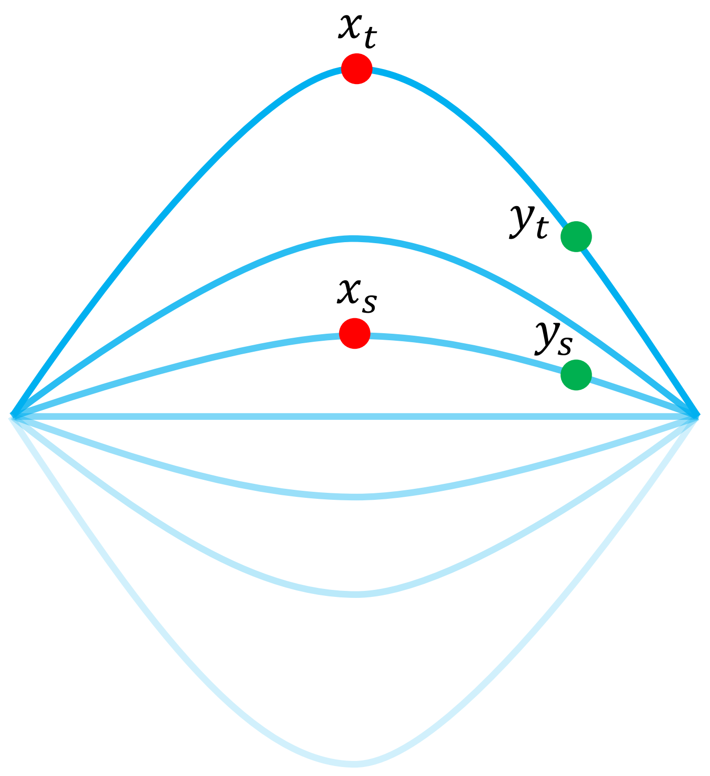
Intuitively, what constraints should a wave-like function $u(x,t)$ satisfy?

Points which are associated with high curvature should more quickly straighten.

1. Curvature at x
2. Distance traveled by point x over fixed time period
3. acceleration of x

1. Δu
2. Distance traveled by point x over fixed time period
3. $\frac{\partial^2 u}{\partial t^2}$

Extent to which $u(b)$ differs from $avg_x\, u(x)$ for x in a spherical shell centered at b $=$ Acceleration of x



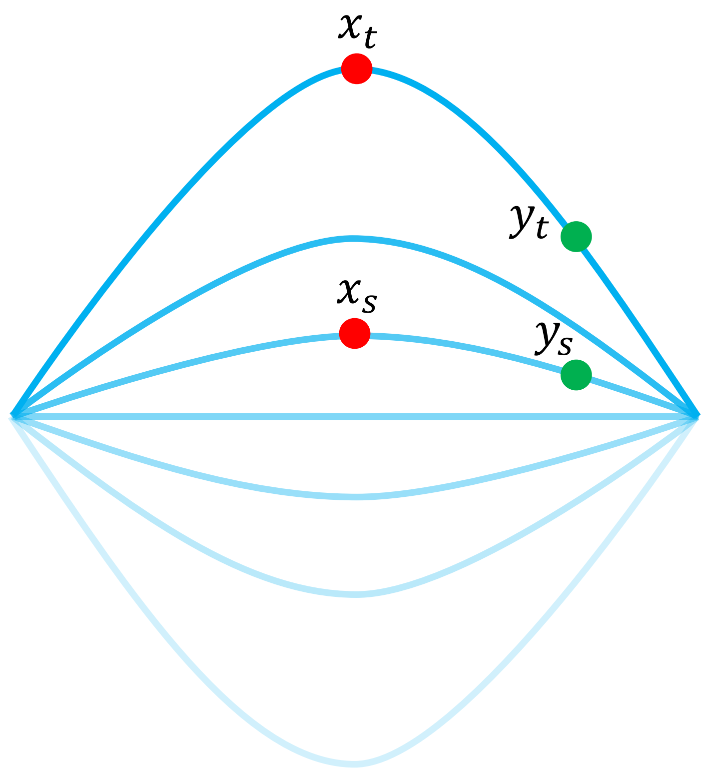
x_t: high curvature, high acceleration

x_t: low curvature, low acceleration

The Laplacian and the Wave Equation

Intuitively, what constraints should a wave-like function $u(x, t)$ satisfy?

Points which are associated with high curvature should more quickly straighten.



1. Curvature at x
2. Distance traveled by point x over fixed time period
3. acceleration of x

1. Δu
2. Distance traveled by point x over fixed time period
3. $\frac{\partial^2 u}{\partial t^2}$

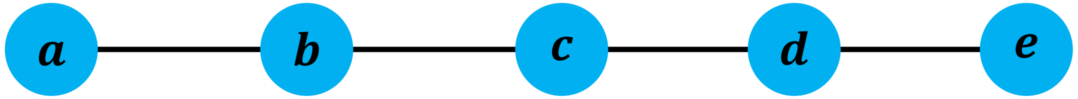
Extent to which $u(b)$ differs from $avg_x\, u(x)$ for x in a spherical shell centered at b $=$ Acceleration of x

$$\Delta u = \frac{\partial^2 u}{\partial t^2}$$

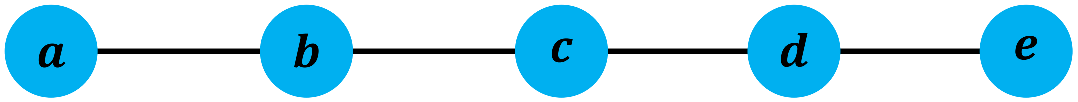
x_t: high curvature, high acceleration

x_t: low curvature, low acceleration

The Laplacian: Discrete Version

$G =$ 

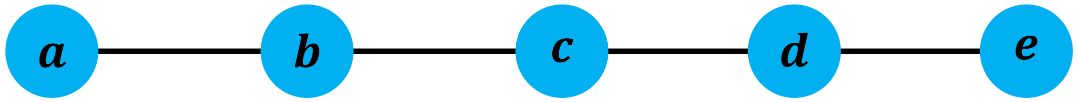
The Laplacian: Discrete Version

$G =$ 

The transformation $\Delta f = \frac{\partial^2}{\partial x^2} f|_c := f(b) - 2f(c) + f(d)$ is linear, so Δ can be described by a matrix:

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

The Laplacian: Discrete Version

$G =$ 

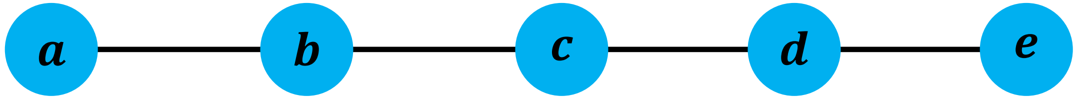
The transformation $\Delta f = \frac{\partial^2}{\partial x^2} f|_c \coloneqq f(b) - 2f(c) + f(d)$ is linear, so Δ can be described by a matrix:

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

$$\begin{aligned} \Delta \mathrm{f}\Big|_{\mathrm{c}} &= \mathrm{Lf}\Big|_{\mathrm{c}} \\ &= < \mathrm{L}_{2*}, \big(f(\mathrm{a}), f(\mathrm{b}), f(c), f(d), f(e)\big) > \\ &= -f(b) + 2f(c) - f(d) \end{aligned}$$

The Laplacian: Discrete Version

G =



The transformation $\Delta f = \frac{\partial^2}{\partial x^2} f|_c \coloneqq f(b) - 2f(c) + f(d)$ is linear, so Δ can be described by a matrix:

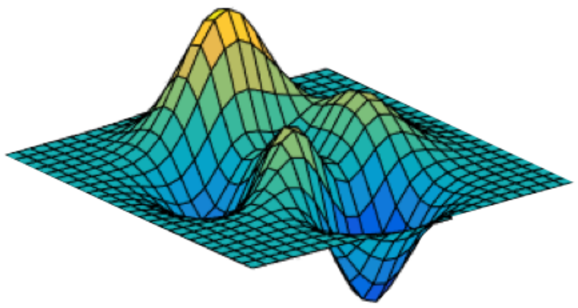
$$L = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

$$\begin{aligned} \Delta \mathrm{f}\Big|_{\mathrm{c}} &= \mathrm{Lf}\Big|_{\mathrm{c}} \\ &= < \mathrm{L}_{2*}, \big(f(\mathrm{a}), f(\mathrm{b}), f(c), f(d), f(e)\big) > \\ &= -f(b) + 2f(c) - f(d) \end{aligned}$$

What if the structure of G varies from node to node?

Laplacian on Curved Manifolds

Consider a function $f: M \to \mathbb{R}$, on a manifold M, the structure of which varies from point to point



How should we define a notion of the Laplacian ∇_M when M has interesting structure?

Laplacian on Curved Manifolds

Riemannian Geometry

Def: A Riemannian manifold is a differentiable manifold endowed with a point-varying metric:

Laplacian on Curved Manifolds

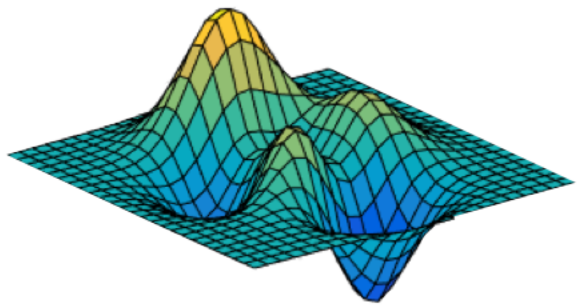
Riemannian Geometry

Def: A Riemannian manifold is a differentiable manifold endowed with a point-varying metric:

$$g_p : T_pM \times T_pM \to \mathbb{R}$$

$$(X(p), Y(p)) \mapsto g_p\big(X(p), Y(p)\big)$$

$$g_p = (g^{ij})$$



Laplacian on Curved Manifolds

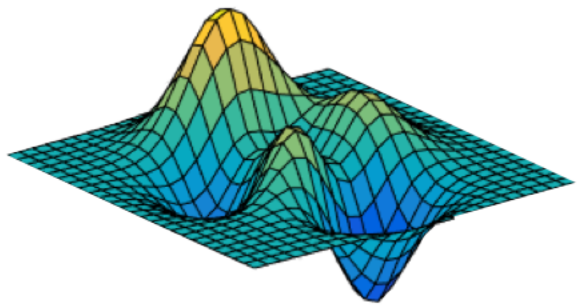
Riemannian Geometry

Def: A Riemannian manifold is a differentiable manifold endowed with a point-varying metric:

$$g_p : T_pM \times T_pM \to \mathbb{R}$$

$$(X(p), Y(p)) \mapsto g_p\big(X(p), Y(p)\big)$$

$$g_p = (g^{ij})$$



Laplacian:

$$\Delta_M f \coloneqq \frac{1}{\sqrt{|g|}} \partial_i(\sqrt{|g|} g^{ij} \partial_j f)$$

Laplacian on Curved Manifolds

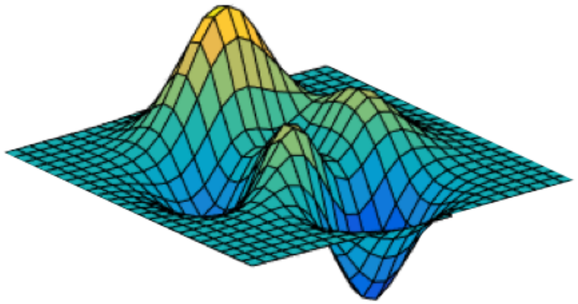
Riemannian Geometry

Def: A Riemannian manifold is a differentiable manifold endowed with a point-varying metric:

$$g_p : T_pM \times T_pM \to \mathbb{R}$$

$$(X(p), Y(p)) \mapsto g_p\big(X(p), Y(p)\big)$$

$$g_p = (g^{ij})$$



Laplacian:

$$\Delta_M f \coloneqq \frac{1}{\sqrt{|g|}} \partial_i (\sqrt{|g|} g^{ij} \partial_j f)$$

The non constant structure of the underlying manifold, the domain of $\boldsymbol{f} : \boldsymbol{M} \to \mathbb{R}$, is reflected in the operator $\boldsymbol{\Delta_{\mathrm{M}}}$.

Laplacian on Curved Manifolds

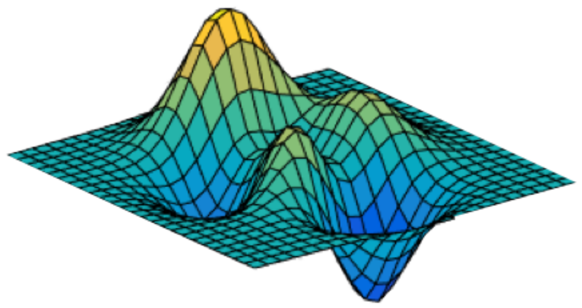
Riemannian Geometry

Def: A Riemannian manifold is a differentiable manifold endowed with a point-varying metric:

$$g_p\colon T_pM \times T_pM \to \mathbb{R}$$

$$(X(p), Y(p)) \mapsto g_p\big(X(p), Y(p)\big)$$

$$g_p = (g^{ij})$$



Laplacian:

$$\Delta_M f \coloneqq \frac{1}{\sqrt{|g|}} \partial_i(\sqrt{|g|} g^{ij} \partial_j f)$$

The non constant structure of the underlying manifold, the domain of $f\colon M \to \mathbb{R}$, is reflected in the operator Δ_{M}.

Euclidian Space Case:

$$g^{ij} = \delta_{ij}, \quad g = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$$

This yields $\Delta f = \partial_i^2 f$ the standard Laplacian

Laplacian on Curved Manifolds

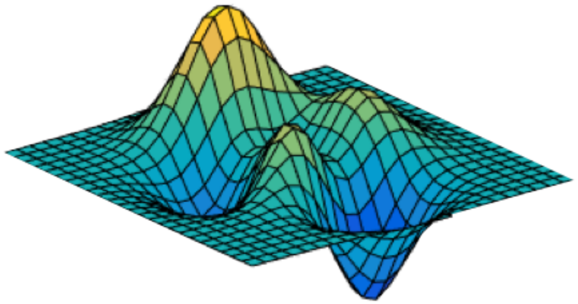
Riemannian Geometry

Def: A Riemannian manifold is a differentiable manifold endowed with a point-varying metric:

$$g_p : T_pM \times T_pM \to \mathbb{R}$$

$$(X(p), Y(p)) \mapsto g_p\big(X(p), Y(p)\big)$$

$$g_p = (g^{ij})$$



Laplacian:

$$\Delta_M f \coloneqq \frac{1}{\sqrt{|g|}} \partial_i (\sqrt{|g|} g^{ij} \partial_j f)$$

Euclidian Space Case:

$$g^{ij} = \delta_{ij}, \quad g = \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}$$

This yields $\Delta f = \partial_i^2 f$ the standard Laplacian

The non constant structure of the underlying manifold, the domain of $f: M \to \mathbb{R}$, is reflected in the operator Δ_M.

The same is true in the graph case – the structure of the underlying graph, the domain of $f: G \to \mathbb{R}$, is reflected in the operator L.

Laplacian on Nonlinear Graphs

General Definition

$$L_{ij} = \begin{cases} \deg(v_i)\,, & if \;\; i = j \\ -1, & if \; v_i \sim v_j \\ 0, & otherwise \end{cases}$$

Note that $L = D - A$,

Laplacian on Nonlinear Graphs

General Definition

$$L_{ij} = \begin{cases} \deg(v_i), & if \;\; i = j \\ -1, & if \; v_i \sim v_j \\ 0, & otherwise \end{cases}$$

Note that $L = D - A$,

where D is the degree matrix

$$D = \begin{cases} \deg(v_i), & if i = j \\ 0, & otherwise \end{cases}$$

Laplacian on Nonlinear Graphs

General Definition

$$L_{ij} = \begin{cases} \deg(v_i), & if \;\; i = j \\ -1, & if \; v_i \sim v_j \\ 0, & otherwise \end{cases}$$

Note that $L = D - A$,

where D is the degree matrix

$$D = \begin{cases} \deg(v_i), & if i = j \\ 0, & otherwise \end{cases}$$

and A is the adjacency matrix

$$A = \begin{cases} 1, & v_i \sim v_j \\ 0, & otherwise \end{cases}$$

Laplacian on Nonlinear Graphs

General Definition

$$L_{ij} = \begin{cases} \deg(v_i), & if \;\; i = j \\ -1, & if \; v_i \sim v_j \\ 0, & otherwise \end{cases}$$

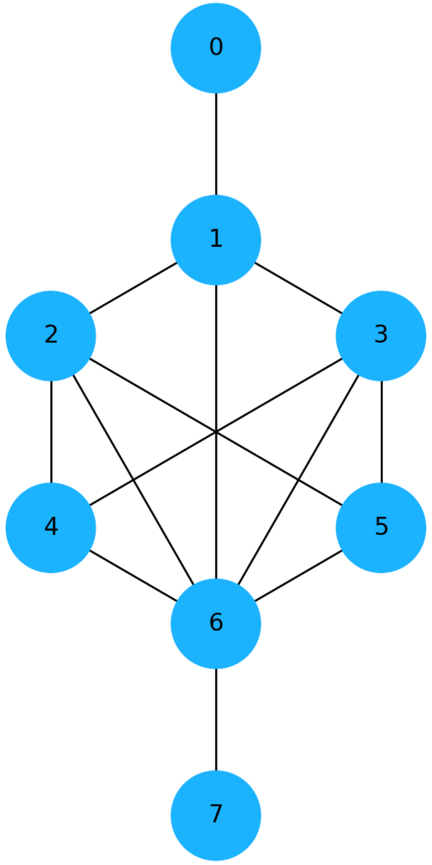
Note that $L = D - A$,

where D is the degree matrix

$$D = \begin{cases} \deg(v_i), & if i = j \\ 0, & otherwise \end{cases}$$

and A is the adjacency matrix

$$A = \begin{cases} 1, & v_i \sim v_j \\ 0, & otherwise \end{cases}$$



$$L$$

$$\begin{matrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & -1 & -1 & -1 & 0 \\ 0 & -1 & 0 & 4 & -1 & -1 & -1 & 0 \\ 0 & 0 & -1 & -1 & 3 & 0 & -1 & 0 \\ 0 & 0 & -1 & -1 & 0 & 3 & -1 & 0 \\ 0 & -1 & -1 & -1 & -1 & -1 & 6 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{matrix}$$

Laplacian on Nonlinear Graphs

General Definition

$$L_{ij} = \begin{cases} \deg(v_i), & if \;\; i = j \\ -1, & if \; v_i \sim v_j \\ 0, & otherwise \end{cases}$$

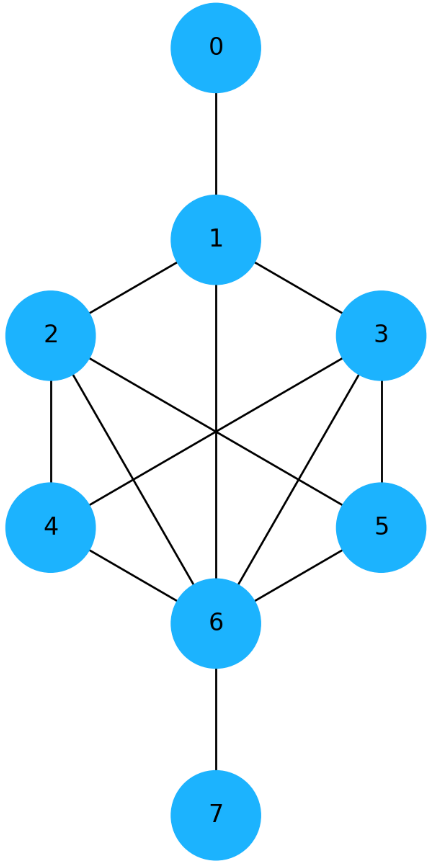
Note that $L = D - A$,

where D is the degree matrix

$$D = \begin{cases} \deg(v_i), & if i = j \\ 0, & otherwise \end{cases}$$

and A is the adjacency matrix

$$A = \begin{cases} 1, & v_i \sim v_j \\ 0, & otherwise \end{cases}$$



L

$$\begin{array}{cccccccc} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & -1 & -1 & -1 & 0 \\ 0 & -1 & 0 & 4 & -1 & -1 & -1 & 0 \\ 0 & 0 & -1 & -1 & 3 & 0 & -1 & 0 \\ 0 & 0 & -1 & -1 & 0 & 3 & -1 & 0 \\ 0 & -1 & -1 & -1 & -1 & -1 & 6 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{array}$$

The structure of the graph changes from node to node, and is reflected in the non-constant block diagonal structure of L.

Wave Equation on Graphs

Recall the wave equation $\Delta u = \frac{\partial^2 u}{\partial t^2}$.

Wave Equation on Graphs

Recall the wave equation $\Delta u = \frac{\partial^2 u}{\partial t^2}$.

Let $u(x, t)$ be a function $u: Vert(G) \times \mathbb{R} \to \mathbb{R}$, which is a solution to $\frac{\partial^2 u}{\partial t^2} = Lu$.

Wave Equation on Graphs

Recall the wave equation $\Delta u = \frac{\partial^2 u}{\partial t^2}$.

Let $u(x,t)$ be a function $u\colon Vert(G) \times \mathbb{R} \to \mathbb{R}$, which is a solution to $\frac{\partial^2 u}{\partial t^2} = Lu$.

Then u has the form $u(x,t) = e^{\sqrt{\lambda_i}t} v_i(x)$, where v_i is an eigenvector with corresponding eigenvalue λ_i.

Wave Equation on Graphs

Recall the wave equation $\Delta u = \frac{\partial^2 u}{\partial t^2}$.

Let $u(x,t)$ be a function $u: Vert(G) \times \mathbb{R} \to \mathbb{R}$, which is a solution to $\frac{\partial^2 u}{\partial t^2} = Lu$.

Then u has the form $u(x,t) = e^{\sqrt{\lambda_i} t} v_i(x)$, where v_i is an eigenvector with corresponding eigenvalue λ_i.

Upshot: The eigenvalues of L, which correspond to the vibrational frequencies of the wave solutions on G, are determined by the structure of G.

Wave Equation on Graphs

Recall the wave equation $\Delta u = \frac{\partial^2 u}{\partial t^2}$.

Let $u(x, t)$ be a function $u: Vert(G) \times \mathbb{R} \to \mathbb{R}$, which is a solution to $\frac{\partial^2 u}{\partial t^2} = Lu$.

Then u has the form $u(x, t) = e^{\sqrt{\lambda_i} t} v_i(x)$, where v_i is an eigenvector with corresponding eigenvalue λ_i.

Upshot: The eigenvalues of L, which correspond to the vibrational frequencies of the wave solutions on G, are determined by the structure of G.

Spectral Graph Theory: the study of graph structure via the spectra of various graph-associated matrices.

Wave Equation on Graphs

Recall the wave equation $\Delta u = \frac{\partial^2 u}{\partial t^2}$.

Let $u(x,t)$ be a function $u: Vert(G) \times \mathbb{R} \to \mathbb{R}$, which is a solution to $\frac{\partial^2 u}{\partial t^2} = Lu$.

Then u has the form $u(x,t) = e^{\sqrt{\lambda_i} t} v_i(x)$, where v_i is an eigenvector with corresponding eigenvalue λ_i.

Upshot: The eigenvalues of L, which correspond to the vibrational frequencies of the wave solutions on G, are determined by the structure of G.

Spectral Graph Theory: the study of graph structure via the spectra of various graph-associated matrices.

We can exploit the correspondence between frequency and wavelength to visualize graph spectra:

Laplacian Spectra - Eigenvalues

Eigenvalues correspond to frequencies of wave solutions on the domain – can render as wavelengths

Laplacian Spectra - Eigenvalues

Eigenvalues correspond to frequencies of wave solutions on the domain – can render as wavelengths

Consider the Laplacian matrix L and its associated eigenvectors and eigenvalues. We compute the spectrum

$$\Lambda = \{\lambda_1, \ldots, \lambda_{|G|}\},$$

and apply the transformation

$$\lambda \mapsto 400 + 300\frac{|\lambda|}{|G|}$$

Laplacian Spectra - Eigenvalues

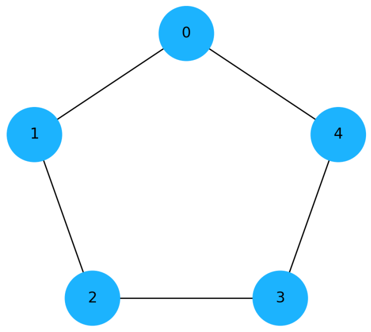
Eigenvalues correspond to frequencies of wave solutions on the domain – can render as wavelengths

Consider the Laplacian matrix L and its associated eigenvectors and eigenvalues. We compute the spectrum

$$\Lambda = \{\lambda_1, \ldots, \lambda_{|G|}\},$$

and apply the transformation

$$\lambda \mapsto 400 + 300\frac{|\lambda|}{|G|}$$



Laplacian Spectra - Eigenvalues

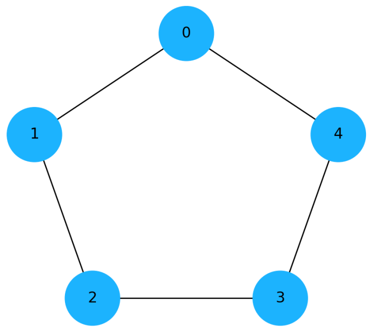
Eigenvalues correspond to frequencies of wave solutions on the domain – can render as wavelengths

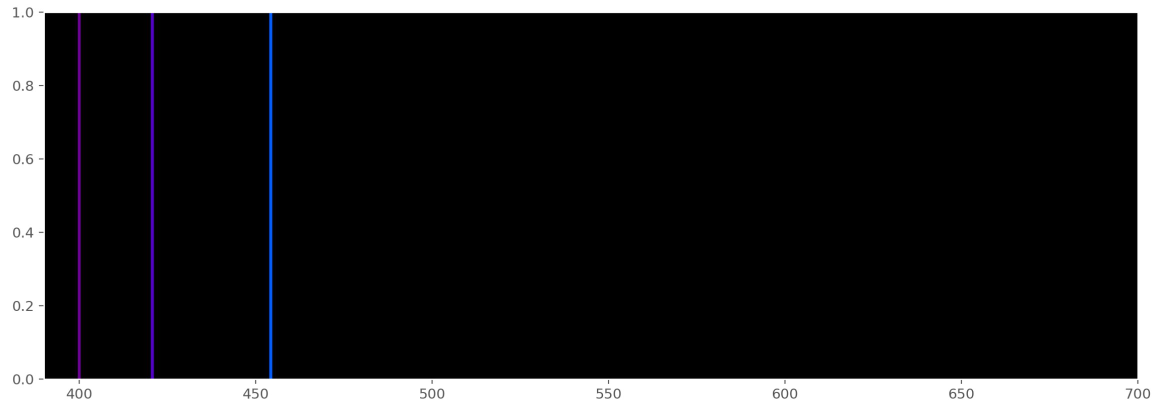
Consider the Laplacian matrix L and its associated eigenvectors and eigenvalues. We compute the spectrum

$$\Lambda = \{\lambda_1, \dots, \lambda_{|G|}\},$$

and apply the transformation

$$\lambda \mapsto 400 + 300\frac{|\lambda|}{|G|}$$





Laplacian Spectra - Eigenvalues

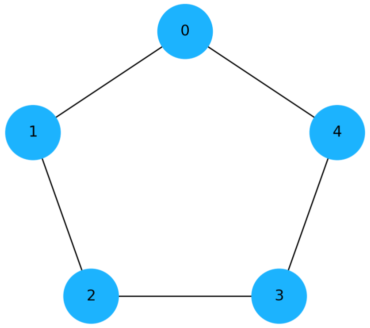
Eigenvalues correspond to frequencies of wave solutions on the domain – can render as wavelengths

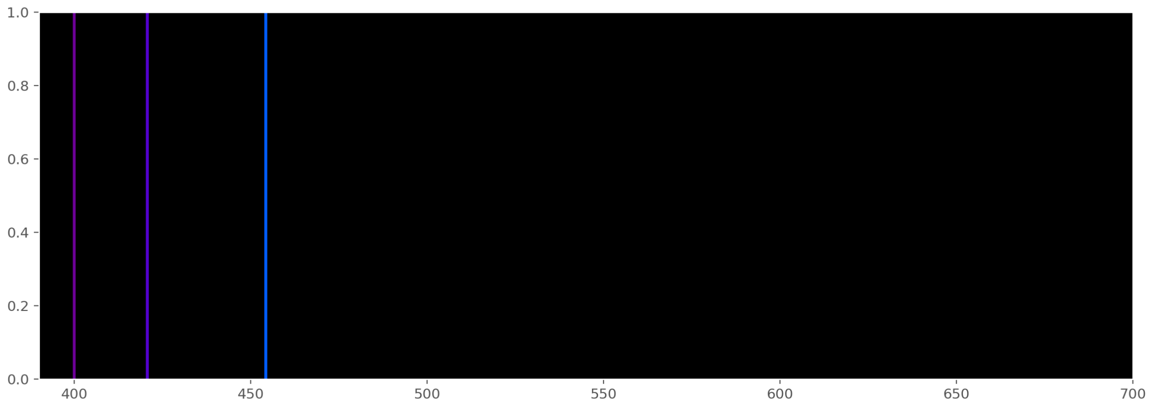
Consider the Laplacian matrix L and its associated eigenvectors and eigenvalues. We compute the spectrum

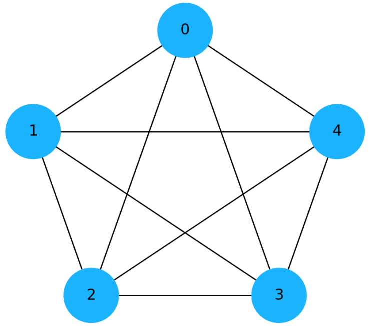
$$\Lambda = \{\lambda_1, \ldots, \lambda_{|G|}\},$$

and apply the transformation

$$\lambda \mapsto 400 + 300\frac{|\lambda|}{|G|}$$







Laplacian Spectra - Eigenvalues

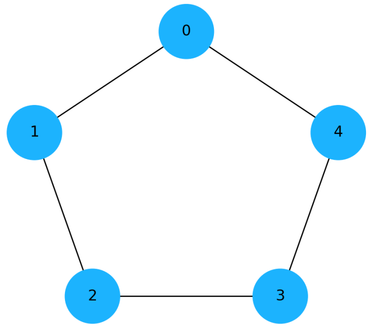
Eigenvalues correspond to frequencies of wave solutions on the domain – can render as wavelengths

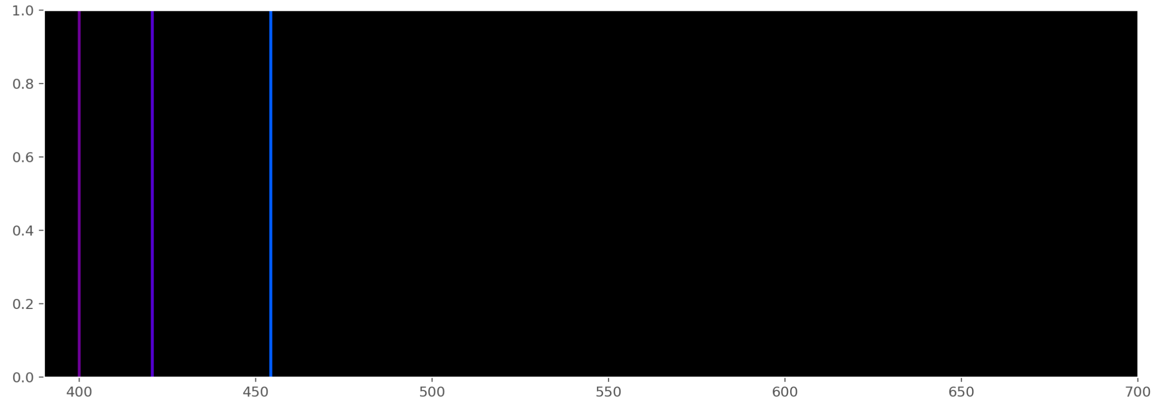
Consider the Laplacian matrix L and its associated eigenvectors and eigenvalues. We compute the spectrum

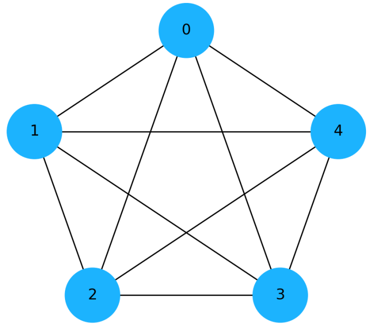
$$\Lambda = \{\lambda_1, \ldots, \lambda_{|G|}\},$$

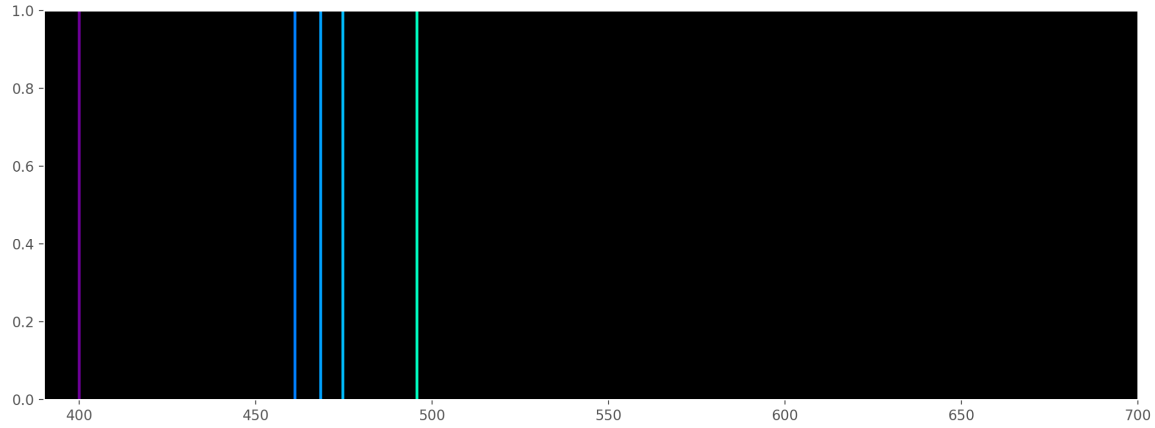
and apply the transformation

$$\lambda \mapsto 400 + 300\frac{|\lambda|}{|G|}$$









Curvature – Higher Dimensions

The Laplacian

Define $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \cdots + \frac{\partial^2}{\partial x_n^2}$

Application: The **Heat** Equation

$$\frac{\partial u}{\partial t} = \Delta u$$

Question 1: What do the solutions to this PDE look like?

Question 2: Is it possible to infer manifold structure from the structure of the operator?

Question 3: Is it possible to answer Question 2 in the graph case?

Heat Kernels on Graphs

How can we measure the similarity between nodes in a graph using heat diffusion?

Heat Kernels on Graphs

How can we measure the similarity between nodes in a graph using heat diffusion?

Consider the following process defined on the node set $Vert(G) = \{1,2, \dots, |Vert(G)|\}$:

Heat Kernels on Graphs

How can we measure the similarity between nodes in a graph using heat diffusion?

Consider the following process defined on the node set $Vert(G) = \{1,2,\ldots,|Vert(G)|\}$:

$$Z(t) = \left(Z_1(t), Z_2(t), \ldots, Z_{|V|}(t)\right)^T \text{, where}$$

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t)\right)$$

Heat Kernels on Graphs

How can we measure the similarity between nodes in a graph using heat diffusion?

Consider the following process defined on the node set $Vert(G) = \{1,2,\dots,|Vert(G)|\}$:

$$Z(t) = \left(Z_1(t), Z_2(t), \dots, Z_{|V|}(t)\right)^T, \text{ where}$$

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V: j \sim i} \left(Z_j(t) - Z_i(t)\right)$$

Intuition: Each node borrows a percentage α of heat from its neighbors based on the pairwise heat differential.

Heat Kernels on Graphs

How can we measure the similarity between nodes in a graph using heat diffusion?

Consider the following process defined on the node set $Vert(G) = \{1,2,\ldots,|Vert(G)|\}$:

$$Z(t) = \left(Z_1(t), Z_2(t), \ldots, Z_{|V|}(t)\right)^T, \text{ where}$$

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t)\right)$$

Intuition: Each node borrows a percentage α of heat from its neighbors based on the pairwise heat differential.

Notice that each node i is counted $\deg(i)$ times in this sum.

Heat Kernels on Graphs

How can we measure the similarity between nodes in a graph using heat diffusion?

Consider the following process defined on the node set $Vert(G) = \{1,2,\dots,|Vert(G)|\}$:

$$Z(t) = \left(Z_1(t), Z_2(t), \dots, Z_{|V|}(t)\right)^T, \text{ where}$$

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t)\right)$$

Intuition: Each node borrows a percentage α of heat from its neighbors based on the pairwise heat differential.

Notice that each node i is counted $\deg(i)$ times in this sum.

We can therefore describe $Z(t)$ for $t \in \mathbb{N}$ as

$$Z(t) = T(t)Z(0)$$

where $T(t) = (1 - \alpha L)^t$ for L the Laplacian.

Heat Kernels

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t) \right)$$

$$Z(t) = T(t)Z(0) = (1 - \alpha L)^t Z(0)$$

What happens to $Z(t) = T(t)Z(0)$ as $t \to \infty$?

Heat Kernels

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t) \right)$$

$$Z(t) = T(t)Z(0) = (1 - \alpha L)^t Z(0)$$

What happens to $Z(t) = T(t)Z(0)$ as $t \to \infty$?

Rewrite $T(t) = (1 - \alpha L)^t$ as $T(t) = \left(1 + \frac{\alpha H}{\frac{1}{\Delta t}}\right)^{\frac{t}{\Delta t}}$ and let $\Delta t \to 0$

Heat Kernels

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t)\right)$$

$$Z(t) = T(t)Z(0) = (1 - \alpha L)^t Z(0)$$

What happens to $Z(t) = T(t)Z(0)$ as $t \to \infty$?

Rewrite $T(t) = (1 - \alpha L)^t$ as $T(t) = \left(1 + \frac{\alpha H}{\frac{1}{\Delta t}}\right)^{\frac{t}{\Delta t}}$ and let $\Delta t \to 0$

This yields the kernel $\mathrm{K}_{-2\alpha t} \coloneqq \sigma^2 e^{-2\alpha t L}$. Ignoring σ and setting $\beta \coloneqq -2\alpha t$, we can rewrite K_β as $e^{\beta L}$,

Heat Kernels

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t) \right)$$

$$Z(t) = T(t)Z(0) = (1 - \alpha L)^t Z(0)$$

What happens to $Z(t) = T(t)Z(0)$ as $t \to \infty$?

Rewrite $T(t) = (1 - \alpha L)^t$ as $T(t) = \left(1 + \frac{\alpha H}{\frac{1}{\Delta t}}\right)^{\frac{t}{\Delta t}}$ and let $\Delta t \to 0$

This yields the kernel $\mathrm{K}_{-2\alpha t} \coloneqq \sigma^2 e^{-2\alpha t L}$. Ignoring σ and setting $\beta \coloneqq -2\alpha t$, we can rewrite K_β as $e^{\beta L}$,

where for any matrix H,

$$e^{\beta H} = \lim_{n \to \infty} \left(1 + \frac{\beta H}{n}\right)^n = I + \beta H + \frac{1}{2!}\beta H^2 + \frac{1}{3!}\beta H^3 + \cdots$$

Heat Kernels

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t) \right)$$

$$Z(t) = T(t)Z(0) = (1 - \alpha L)^t Z(0)$$

What happens to $Z(t) = T(t)Z(0)$ as $t \to \infty$?

Rewrite $T(t) = (1 - \alpha L)^t$ as $T(t) = \left(1 + \frac{\alpha H}{\frac{1}{\Delta t}}\right)^{\frac{t}{\Delta t}}$ and let $\Delta t \to 0$

This yields the kernel $\mathrm{K}_{-2\alpha t} \coloneqq \sigma^2 e^{-2\alpha t L}$. Ignoring σ and setting $\beta \coloneqq -2\alpha t$, we can rewrite K_β as $e^{\beta L}$,

where for any matrix H,

$$e^{\beta H} = \lim_{n \to \infty} \left(1 + \frac{\beta H}{n}\right)^n = I + \beta H + \frac{1}{2!}\beta H^2 + \frac{1}{3!}\beta H^3 + \cdots$$

Notice K_β satisfies $\frac{d}{d\beta} K_\beta = L K_\beta$. This is precisely the heat equation on G.

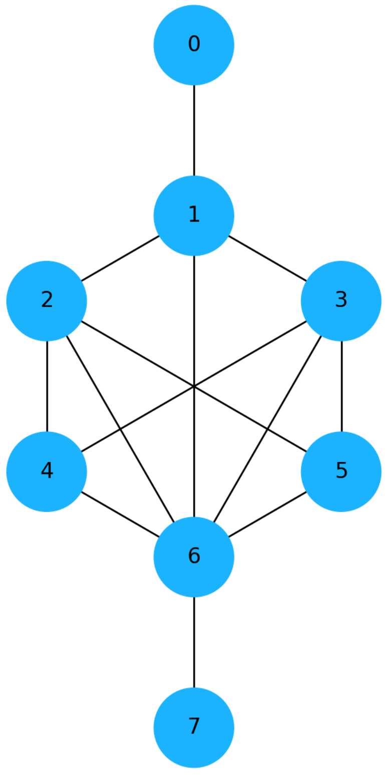
Heat (Dense and Sparse)

Example: We consider two graphs with the same node set, but with different edge structure. The goal is to test if it is possible, through heat diffusion, to measure differences in node similarity

Heat (Dense and Sparse)

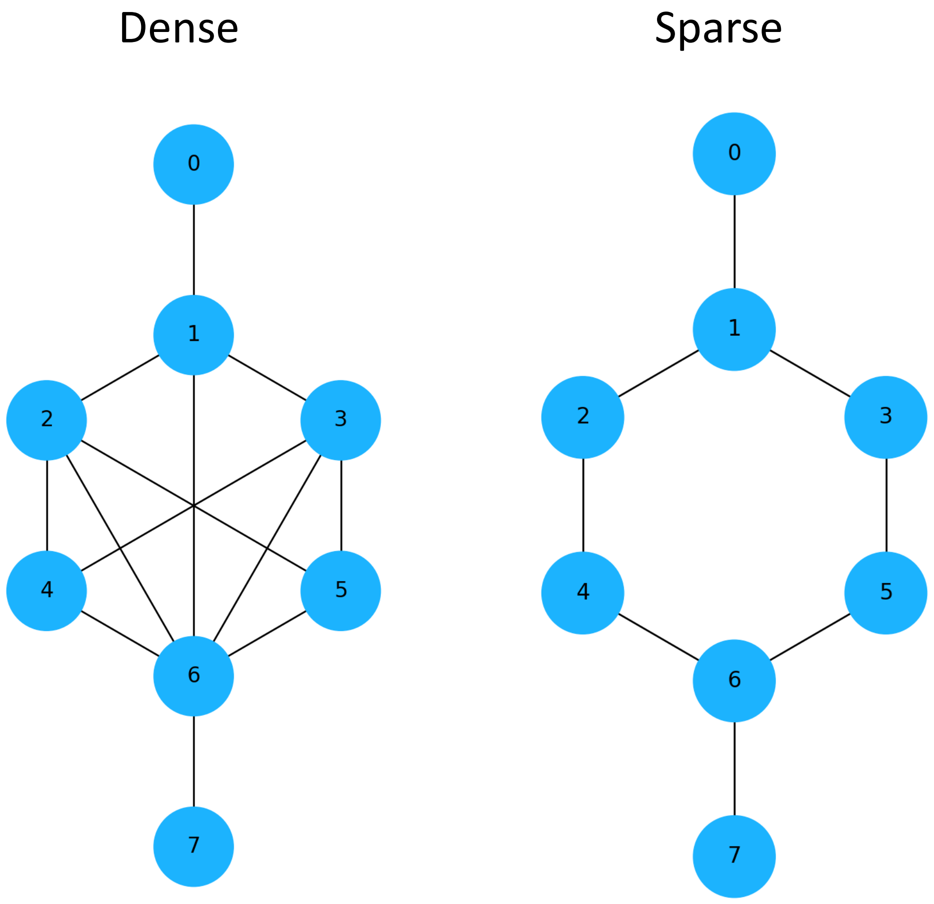
Example: We consider two graphs with the same node set, but with different edge structure. The goal is to test if it is possible, through heat diffusion, to measure differences in node similarity

Dense



Heat (Dense and Sparse)

Example: We consider two graphs with the same node set, but with different edge structure. The goal is to test if it is possible, through heat diffusion, to measure differences in node similarity

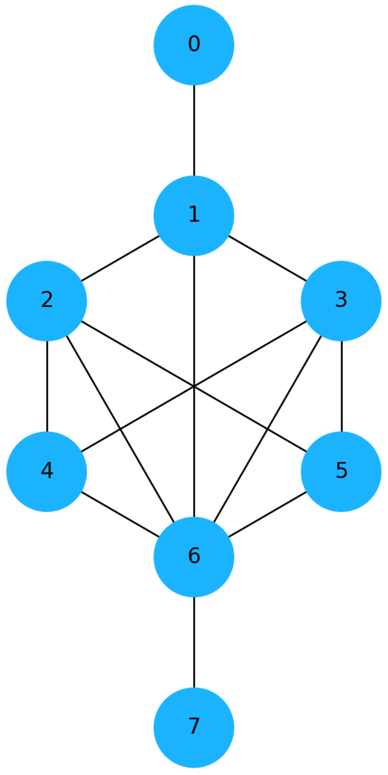


Heat (Dense and Sparse)

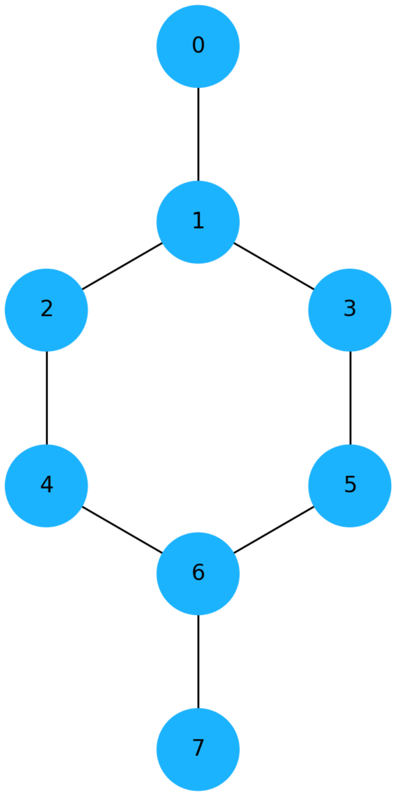
Example: We consider two graphs with the same node set, but with different edge structure. The goal is to test if it is possible, through heat diffusion, to measure differences in node similarity

Step 1: Heat node 0

Dense



Sparse

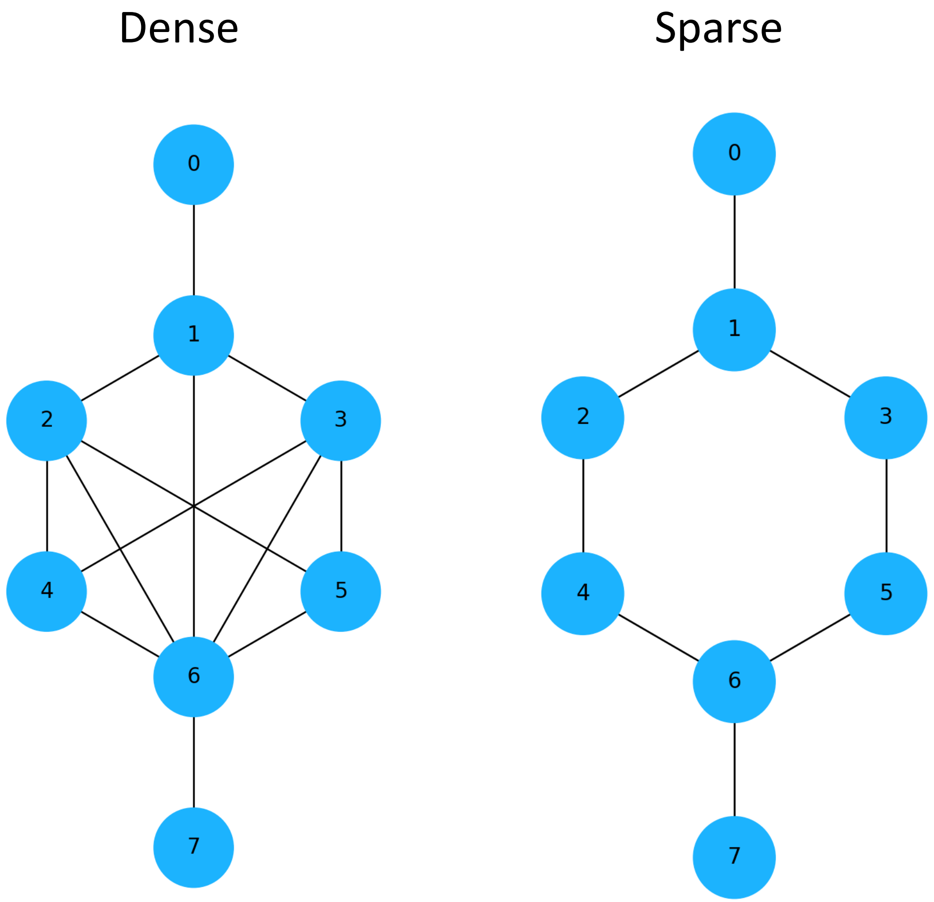


Heat (Dense and Sparse)

Example: We consider two graphs with the same node set, but with different edge structure. The goal is to test if it is possible, through heat diffusion, to measure differences in node similarity

Step 1: Heat node 0

Step 2: Observe heat in each node as it diffuses through the graph



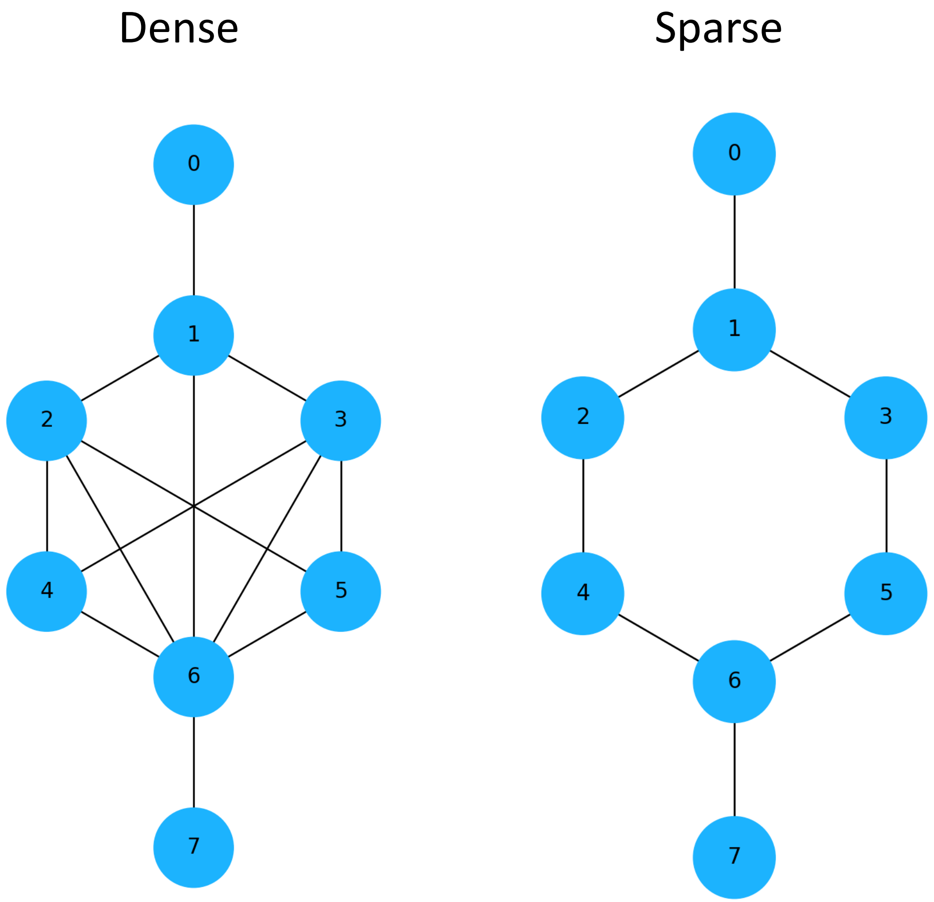
Heat (Dense and Sparse)

Example: We consider two graphs with the same node set, but with different edge structure. The goal is to test if it is possible, through heat diffusion, to measure differences in node similarity

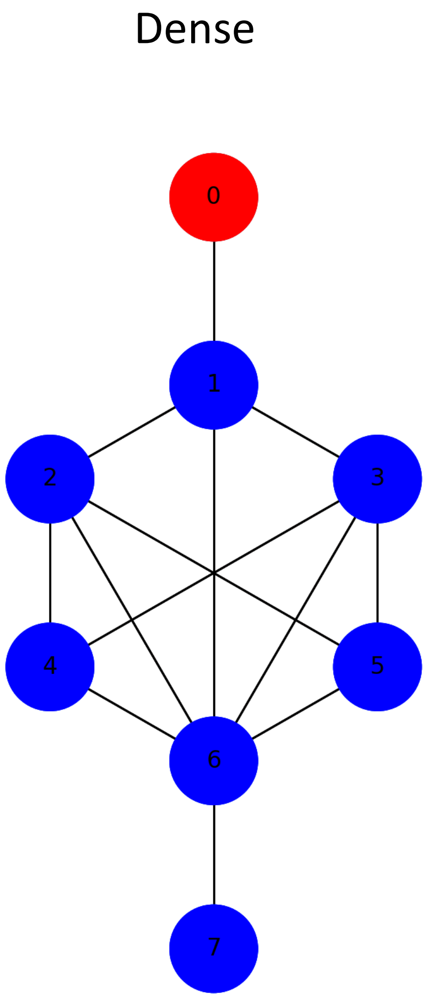
Step 1: Heat node 0

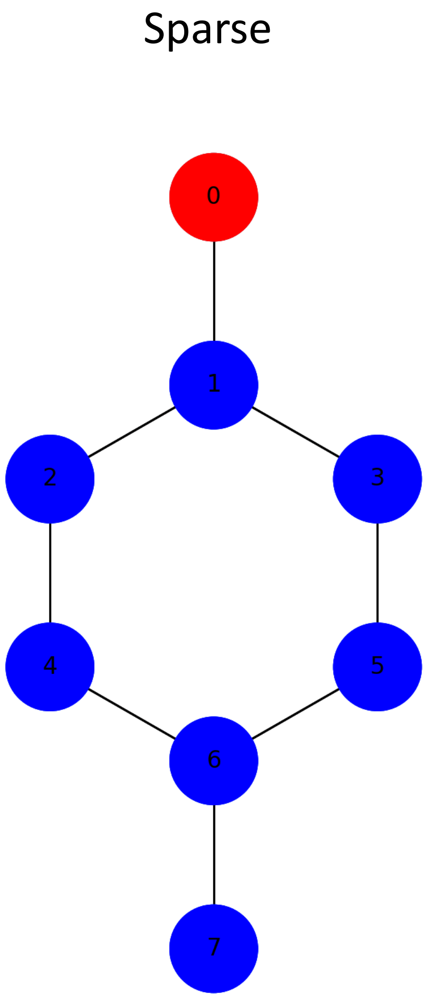
Step 2: Observe heat in each node as it diffuses through the graph

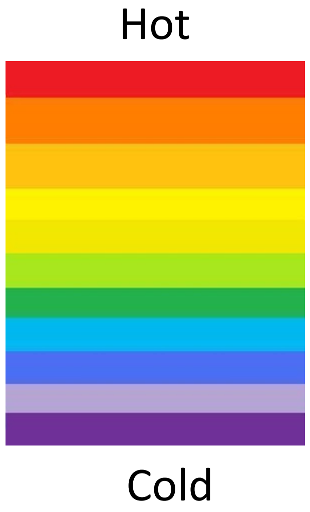
Step 3: Compute node similarity based on node to node heat transfer between times 0 and ∞



Heat (Dense vs Sparse)

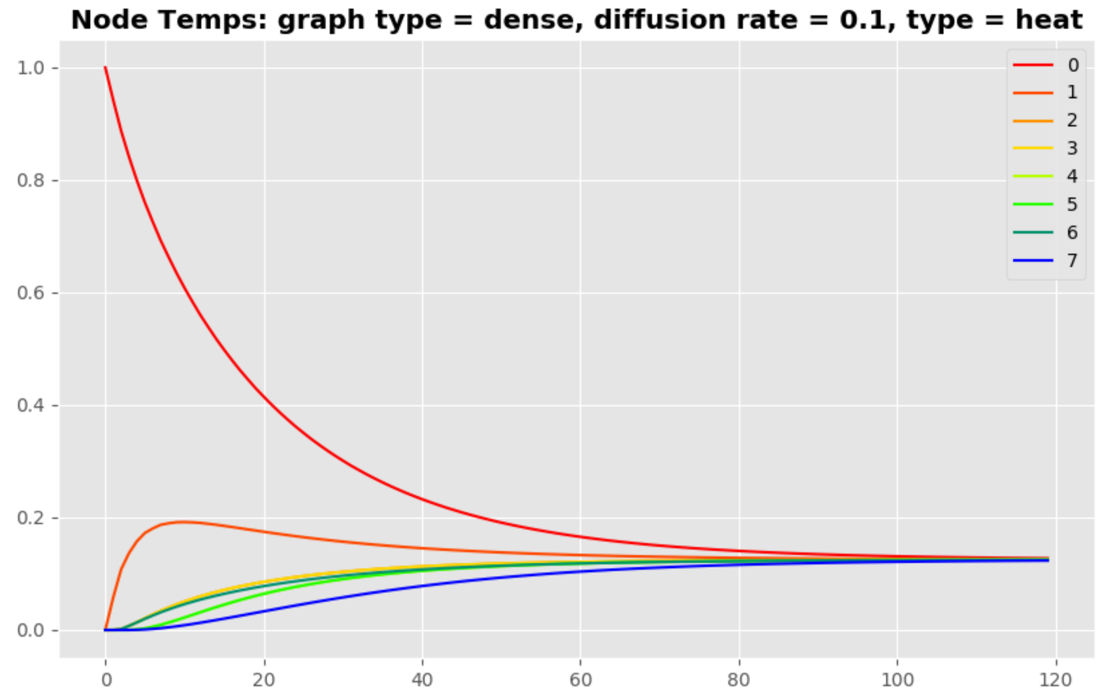




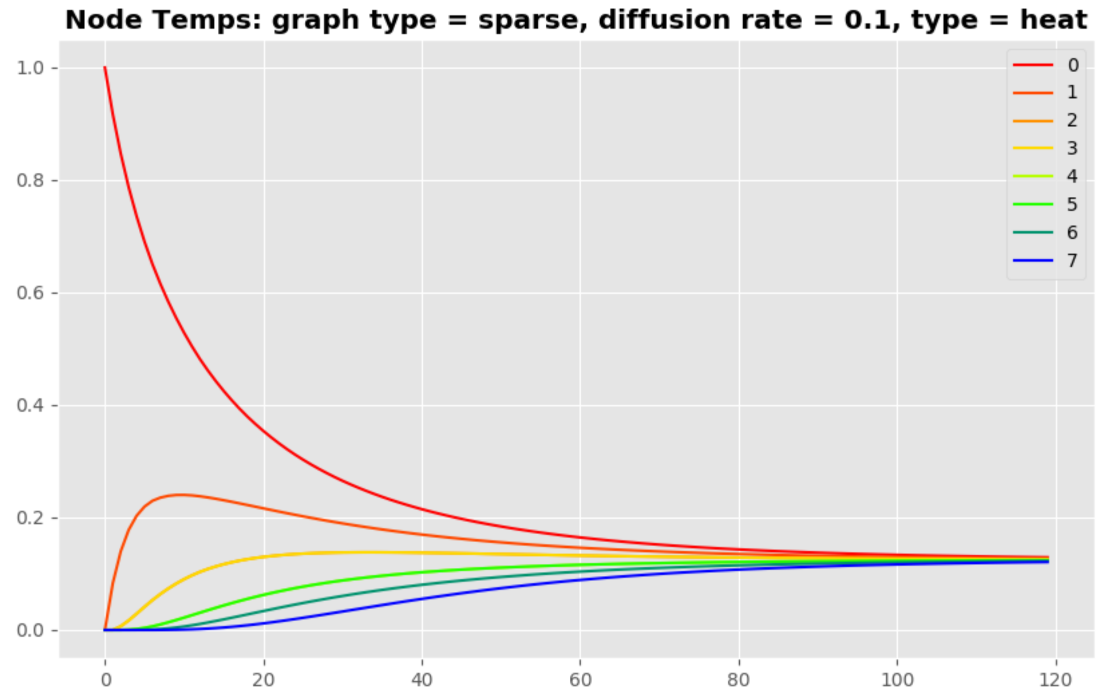


Heat (Dense vs Sparse)

Dense

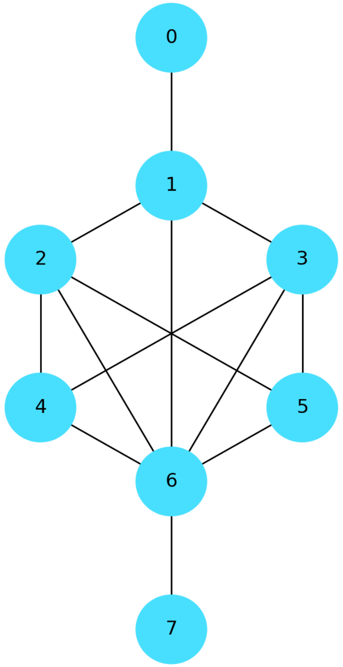


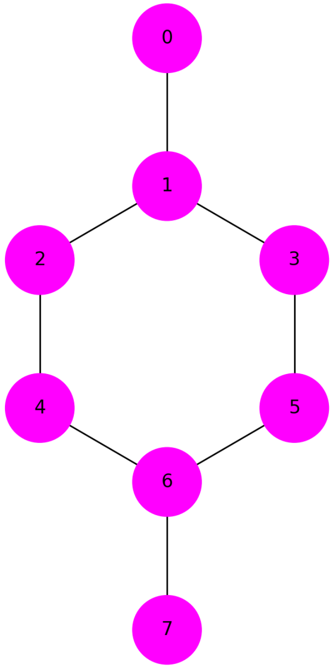
Sparse

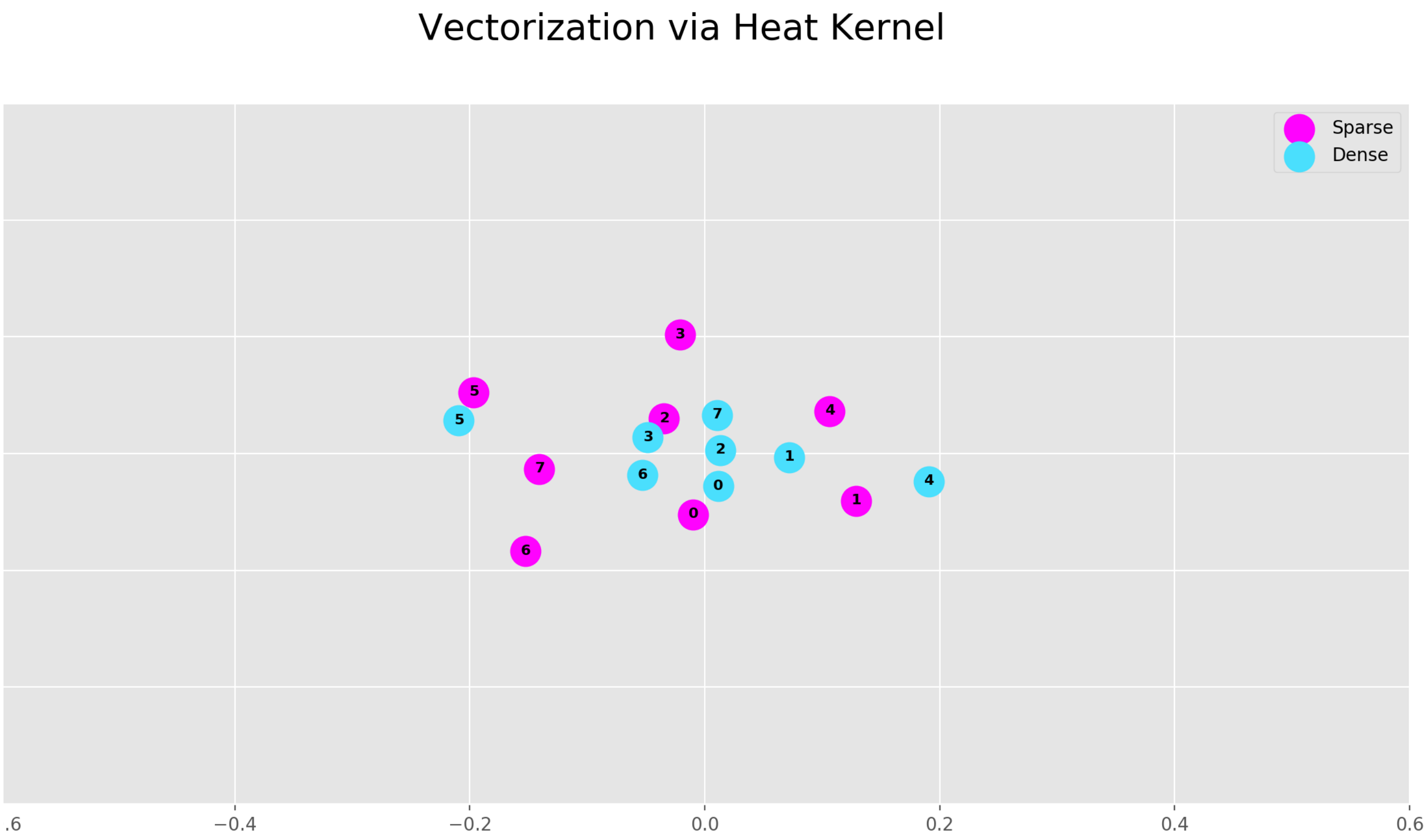


Heat Kernels - Vectorization

Vectorize each graph via Cholesky factorization of $K_\beta = F^T F$. Vectors correspond to columns of F.







Transfer Kernels

Once we have intuition, we can build our own kernels via perturbations of known kernels.

Alter process by considering a directed graph version of the above example, and only allow heat transfer along directed edges.

Transfer Kernels

Once we have intuition, we can build our own kernels via perturbations of known kernels.

Alter process by considering a directed graph version of the above example, and only allow heat transfer along directed edges.

Original heat kernel process: Undirected graph G with $V \coloneqq Vert(G)$.

$$Z(t) = \left(Z_1(t), Z_2(t), \dots, Z_{|V|}(t)\right)^T, \text{ where}$$

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t)\right)$$

$$Z(t) = T(t)Z(0)$$

where $T(t) = (1 - \alpha L)^t$ for L the Laplacian.

Note that $L = D - A$,

Transfer Kernels

Once we have intuition, we can build our own kernels via perturbations of known kernels.

Alter process by considering a directed graph version of the above example, and only allow heat transfer along directed edges.

Original heat kernel process: Undirected graph G with $V \coloneqq Vert(G)$.

$$Z(t) = \left(Z_1(t), Z_2(t), \dots, Z_{|V|}(t)\right)^T, \text{ where}$$

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t)\right)$$

$$Z(t) = T(t)Z(0)$$

where $T(t) = (1 - \alpha L)^t$ for L the Laplacian.

Note that $L = D - A$,

New heat kernel process: Directed graph G with $V \coloneqq Vert(G)$.

Transfer Kernels

Once we have intuition, we can build our own kernels via perturbations of known kernels.

Alter process by considering a directed graph version of the above example, and only allow heat transfer along directed edges.

Original heat kernel process: Undirected graph G with $V \coloneqq Vert(G)$.

$$Z(t) = \left(Z_1(t), Z_2(t), \dots, Z_{|V|}(t)\right)^T, \text{ where}$$

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t)\right)$$

$$Z(t) = T(t)Z(0)$$

where $T(t) = (1 - \alpha L)^t$ for L the Laplacian.

Note that $L = D - A$,

New heat kernel process: Directed graph G with $V \coloneqq Vert(G)$.

$$Z(t) = \left(Z_1(t), Z_2(t), \dots, Z_{|V|}(t)\right)^T, \text{ where}$$

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : i \to j} \left(Z_j(t) - Z_i(t)\right)$$

Transfer Kernels

Once we have intuition, we can build our own kernels via perturbations of known kernels.

Alter process by considering a directed graph version of the above example, and only allow heat transfer along directed edges.

Original heat kernel process: Undirected graph G with $V \coloneqq Vert(G)$.

$$Z(t) = \left(Z_1(t), Z_2(t), \dots, Z_{|V|}(t)\right)^T, \text{ where}$$

$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : j \sim i} \left(Z_j(t) - Z_i(t)\right)$$

$$Z(t) = T(t)Z(0)$$

where $T(t) = (1 - \alpha L)^t$ for L the Laplacian.

Note that $L = D - A$,

New heat kernel process: Directed graph G with $V \coloneqq Vert(G)$.

$$Z(t) = \left(Z_1(t), Z_2(t), \dots, Z_{|V|}(t)\right)^T, \text{ where}$$

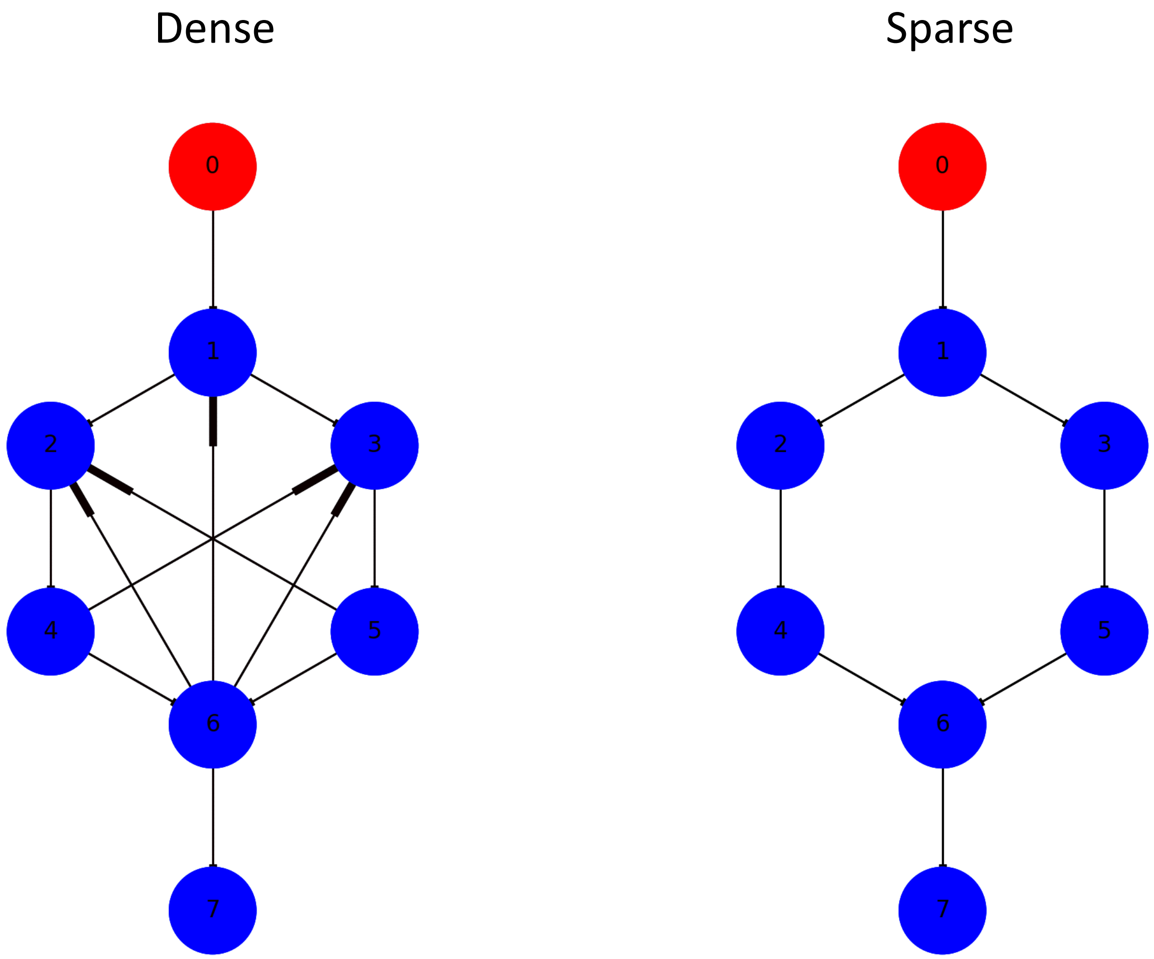
$$Z_i(t+1) = Z_i(t) + \alpha \sum_{j \in V : i \to j} \left(Z_j(t) - Z_i(t)\right)$$

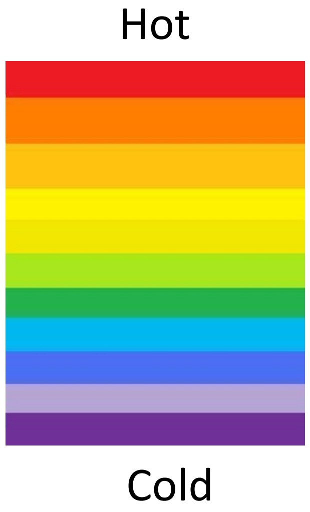
$$Z(t) = T(t)Z(0)$$

where $T(t) = (1 - \alpha L_O)^t$ for L the Laplacian.

Note that $L_O = D_{out} - A$,

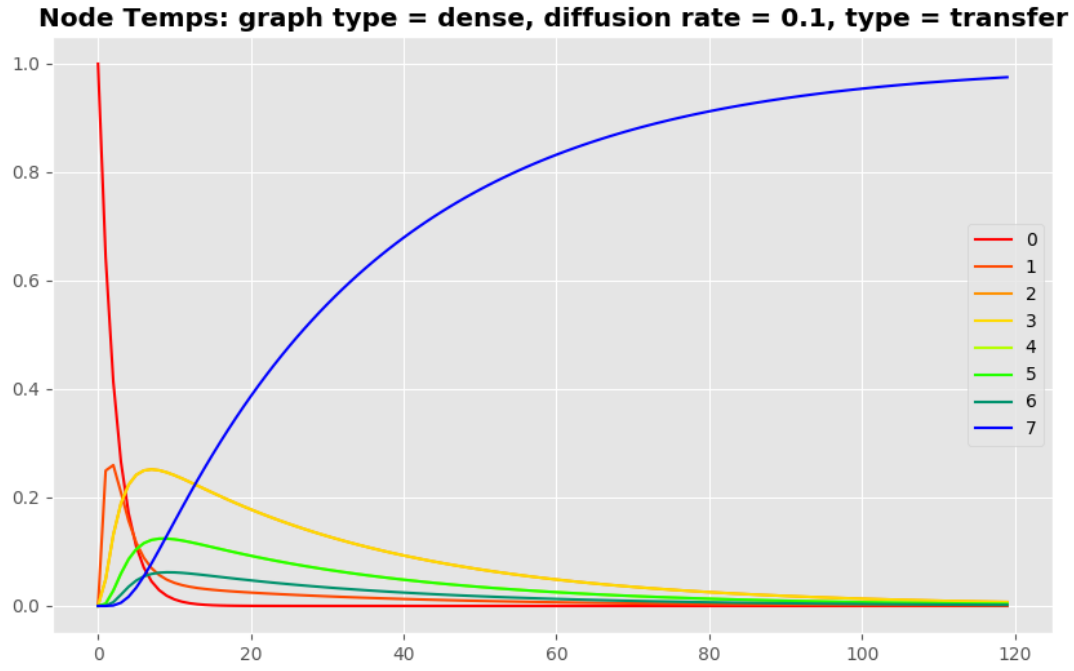
Transfer (Dense vs Sparse)



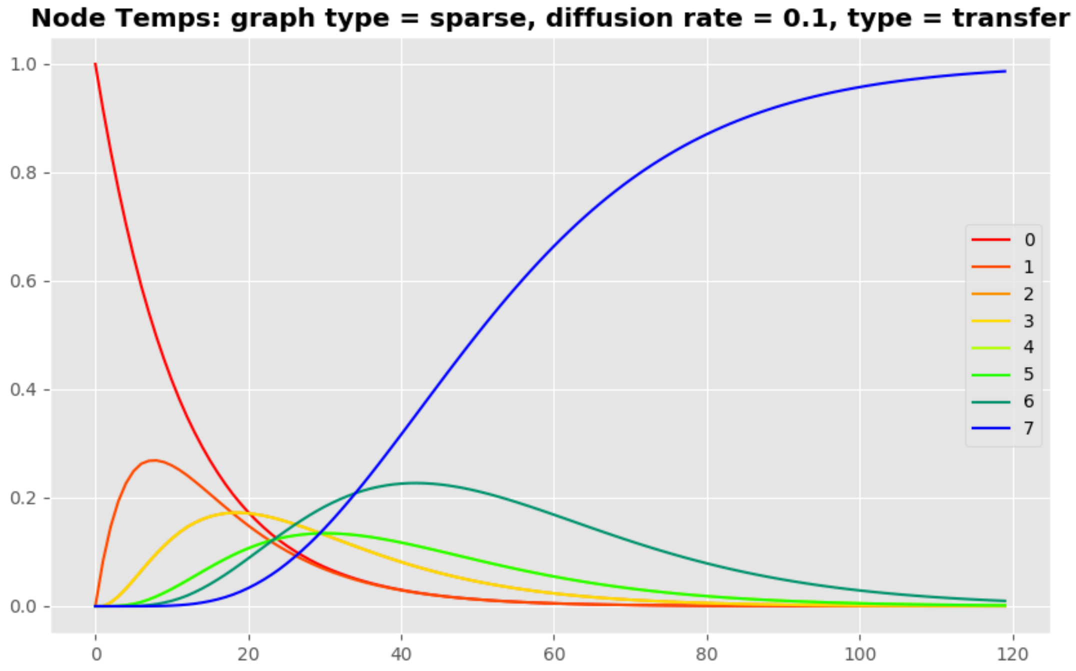


Transfer (Dense and Sparse)

Dense



Sparse



Heat Kernels - Applications

Assume we have a kernel K defined on some set Ω, that is, we have a map $K: \Omega \times \Omega \to \mathbb{R}$

Heat Kernels - Applications

Assume we have a kernel K defined on some set Ω, that is, we have a map $K: \Omega \times \Omega \to \mathbb{R}$

Question: How can we define a kernel on $\Omega^n := \Omega \times \cdots \times \Omega$?

Heat Kernels - Applications

Assume we have a kernel K defined on some set Ω, that is, we have a map $K\colon \Omega \times \Omega \to \mathbb{R}$

Question: How can we define a kernel on $\Omega^n \coloneqq \Omega \times \cdots \times \Omega$?

Answer: Construct the tensor product $K^n \coloneqq \bigotimes_{i=1}^n K$ defined by $K^n(x, x') = \prod_{i=1}^n K(x_i, x_i')$

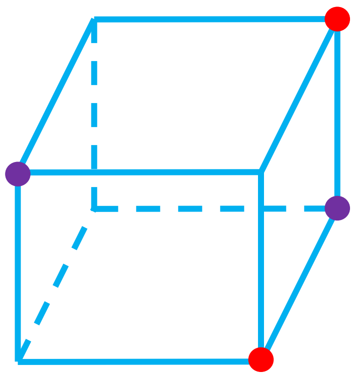
Heat Kernels - Applications

Assume we have a kernel K defined on some set Ω, that is, we have a map $K: \Omega \times \Omega \to \mathbb{R}$

Question: How can we define a kernel on $\Omega^n := \Omega \times \cdots \times \Omega$?

Answer: Construct the tensor product $K^n := \bigotimes_{i=1}^n K$ defined by $K^n(x, x') = \prod_{i=1}^n K(x_i, x_i')$

Example: Construct kernel on binary strings via the construction of a graph kernel on the hypercube

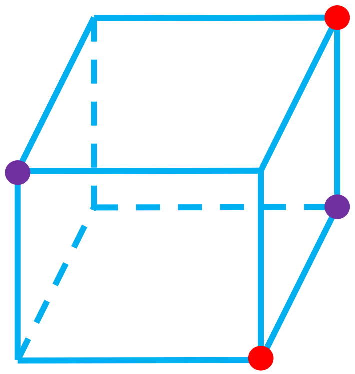


● $\{(0,0,1), (1,1,0)\}$ $\quad d\big((0,0,1),(1,1,0)\big) = 3$

● $\{(1,0,0), (1,1,1)\}$ $\quad d\big((1,0,0),(1,1,1)\big) = 2$

Idea is to measure similarity between points $(0,0,1)$ and $(1,1,0)$, and between points $(1,0,0)$ and $(1,1,1)$, while taking into account the graph structure.

Heat Kernels - Applications

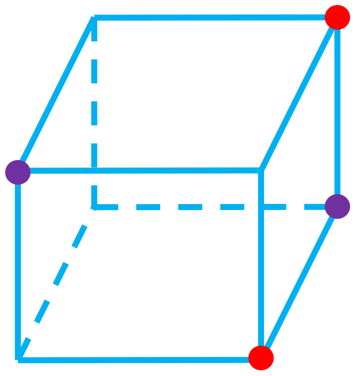


$\{(0,0,1),(1,1,0)\}$ $\quad d\big((0,0,1),(1,1,0)\big) = 3$

$\{(1,0,0),(1,1,1)\}$ $\quad d\big((1,0,0),(1,1,1)\big) = 2$

Idea is to measure similarity between points $(0,0,1)$ and $(1,1,0)$, and between points $(1,0,0)$ and $(1,1,1)$, while taking into account the graph structure.

Heat Kernels - Applications



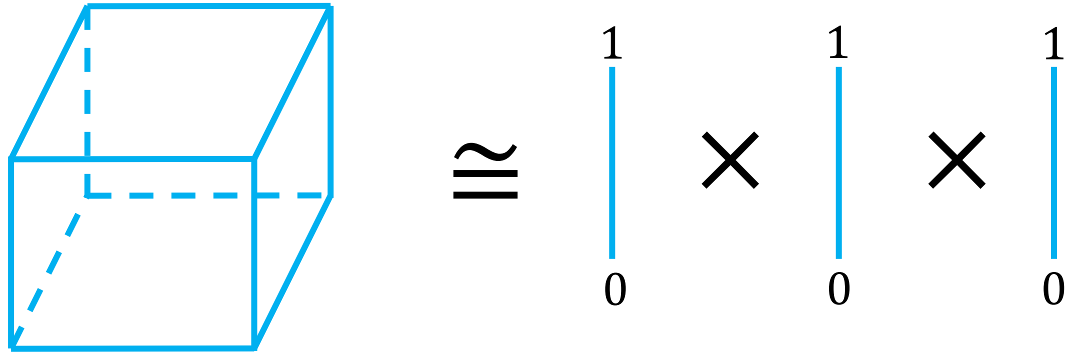
$\{(0,0,1),(1,1,0)\}$ $\quad d\big((0,0,1),(1,1,0)\big) = 3$

$\{(1,0,0),(1,1,1)\}$ $\quad d\big((1,0,0),(1,1,1)\big) = 2$

Idea is to measure similarity between points $(0,0,1)$ and $(1,1,0)$, and between points $(1,0,0)$ and $(1,1,1)$, while taking into account the graph structure.

The hypercube can be constructed by successive cross products of simpler graphs.

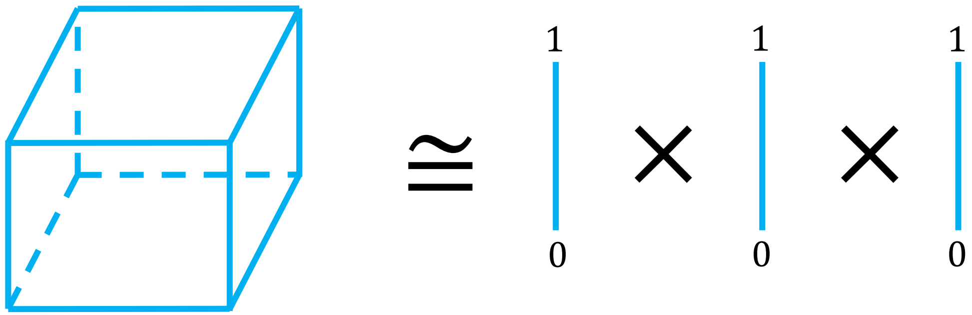
$L_{ij} = -1 + \delta_{ij} n$ for an unweighted complete graph.



Heat Kernels - Applications

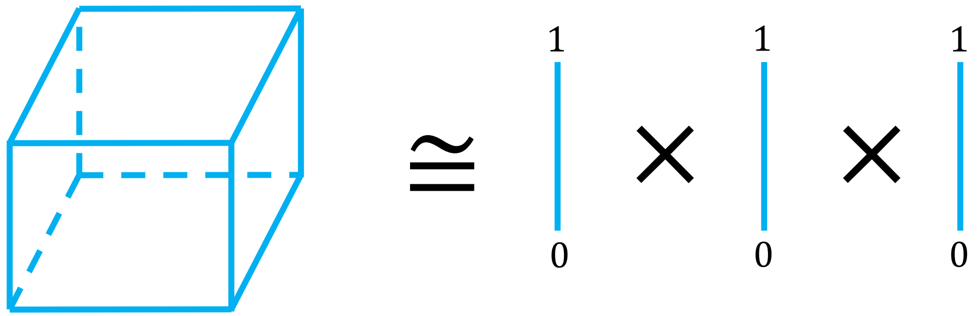
The hypercube can be constructed by successive cross products of simpler graphs.

$L_{ij} = -1 + \delta_{ij} n$ for an unweighted complete graph.



Heat Kernels - Applications

The hypercube can be constructed by successive cross products of simpler graphs.



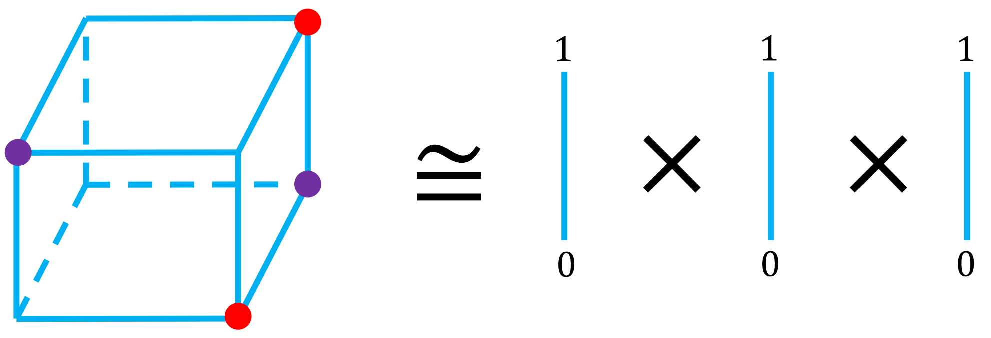
$L_{ij} = -1 + \delta_{ij}n$ for an unweighted complete graph.

Solving $\frac{d}{d\beta}K_\beta = LK_\beta$ on the complete graph with n vertices yields

$$K(i,j) = \begin{cases} \dfrac{1 + (n-1)e^{-n\beta}}{n}, & for\ i = j \\ \dfrac{1 - e^{-n\beta}}{n}, & for\ i \neq j \end{cases}$$

Heat Kernels - Applications

The hypercube can be constructed by successive cross products of simpler graphs.



$L_{ij} = -1 + \delta_{ij} n$ for an unweighted complete graph.

Solving $\frac{d}{d\beta} K_\beta = L K_\beta$ on the complete graph with n vertices yields

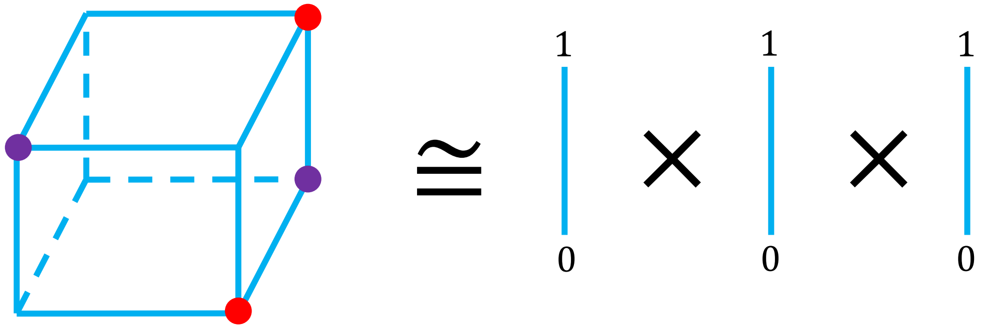
$$K(i,j) = \begin{cases} \dfrac{1 + (n-1)e^{-n\beta}}{n}, & for\ i = j \\ \dfrac{1 - e^{-n\beta}}{n}, & for\ i \neq j \end{cases}$$

and applying $K^n(x, x') = \prod_{i=1}^{n} K(x_i, x_i')$ gives the kernel $K(x, x') \propto \left(\dfrac{1 - e^{-2\beta}}{1 + e^{-2\beta}} \right)^{d(x,x')}$

$$= (\tanh \beta)^{d(x,x')}$$

Heat Kernels - Applications

The hypercube can be constructed by successive cross products of simpler graphs.



$L_{ij} = -1 + \delta_{ij} n$ for an unweighted complete graph.

Solving $\frac{d}{d\beta} K_\beta = L K_\beta$ on the complete graph with n vertices yields

$$K(i,j) = \begin{cases} \dfrac{1 + (n-1)e^{-n\beta}}{n}, & for\ i = j \\ \dfrac{1 - e^{-n\beta}}{n}, & for\ i \neq j \end{cases}$$

and applying $K^n(x, x') = \prod_{i=1}^{n} K(x_i, x_i')$ gives the kernel $K(x, x') \propto \left(\dfrac{1 - e^{-2\beta}}{1 + e^{-2\beta}}\right)^{d(x,x')}$

$$= (\tanh \beta)^{d(x,x')}$$

$$K(\bullet, \bullet) = \tanh(0.1)^3 = 0.00099$$

$$K(\bullet, \bullet) = \tanh(0.1)^2 = 0.00993$$

Thank you!

The End

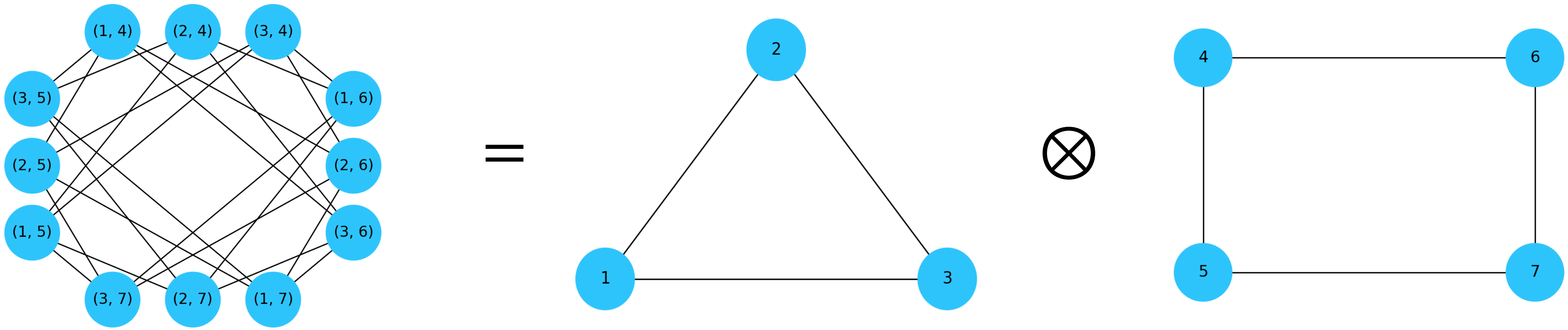
Inter Graph Diffusion Kernels

Tensor Product of Graphs

$$G_\times \coloneqq G \otimes G'$$

$$V_\times = \{(v_i, v_r'): v_i \in V, v_r' \in V'\}$$

$$E_\times = \left\{\left((v_i, v_r'), (v_j, v_s')\right): \left(v_i, v_j\right) \in E \wedge (v_r', v_s') \in E'\right\}$$



Inter Graph Diffusion Kernels

- Construction of Kernel between edge-labeled graphs

Inter Graph Diffusion Kernels

- Dotnet function to function graphs – example of kernel between these graphs
- Plot TSNE vectorization using edge-labeled kernel

Inter Graph Diffusion Kernels

- Functions defined on the set of vertices

Inter Graph Diffusion Kernels

- Lebesgue integral of functions on vertex set

Inter Graph Diffusion Kernels

- Use this vectorization and TNSE or LargeViz to plot a bunch of dotnet function to function graphs
- Good vs Bad red and blue scatter plot

Inter Graph Diffusion Kernels

Kronecker Product: Let M and N be matrices. Then

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 2 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \\ 3 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} & 4 \cdot \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}$$

Inter Graph Diffusion Kernels

- Describe how a heat kernel can serve to measure similarity
- Derivation of Heat Kernels
- String Kernel
- Vectorization

$$A \otimes B$$

$$\Phi(X) A \otimes B \; \Phi(X')$$

$$k(G, G') \coloneqq \sum_{k=0}^{\infty} \mu(k) q_{\times}^{T} W_{\times}^{k} p_{\times}$$

Inter Graph Diffusion Kernels

- Picture with graph of results
- Application to clusters within graph

$$k(G, G') \coloneqq \sum_{k=0}^{\infty} q_\times^T (P_\times D_\times P_\times^{-1})^k p_\times$$

$$q_\times^T P_\times \left(\sum_{k=0}^{\infty} \mu(k) D_\times^k) \right) P_\times^{-1} p_\times$$

$$k(G, G') \coloneqq q_\times^T P_\times e^{\lambda D_\times} P_\times^{-1} p_\times$$

$$k(G, G') \coloneqq q_\times^T P_\times (\mathrm{I} - \lambda D_\times)^{-1} P_\times^{-1} p_\times$$

References

- Chung, F.
- More…