Introduction to Graph Spectral Analysis

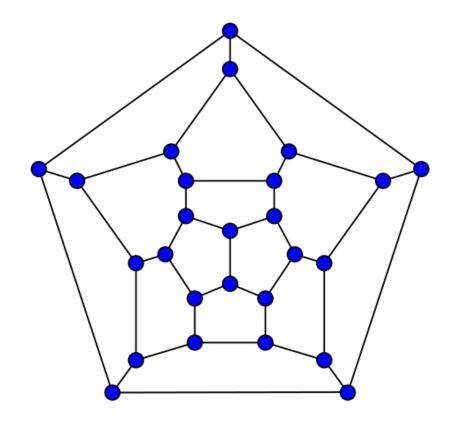
Umberto Castellani, Simone Melzi, Riccardo Marin



Tuesday 28th July 2020 Verona, Italy

Context

Graphs



Eigenvalues & Eigenvectors

Al
$$V = \lambda V$$

Where $Al \in \mathbb{R}^{m \times m}$ (Square Matrix)

eigenvectors $\rightarrow V \in \mathbb{R}^{m \times 1}$ (Column Vector)

eigenvalues $\rightarrow \lambda \in \mathbb{R}^{m \times m}$ (Diagonal Matrix)

Context

1973

W. E. Donath

Lower Bounds for the Partitioning of Graphs

Abstract: Let a k-partition of a graph be a division of the vertices into k disjoint subsets containing $m_i \geq m_i \cdots \geq m_k$ vertices. Let E_i be the number of edges whose two vertices belong to different subsets. Let $k_i \geq k_1 \cdots \geq k_k$ be the k-largest eigenvalues of a matrix, which is the sum of the adjacency matrix of the graph plus any diagonal matrix U such that the sum of all the elements of the sum matrix is zero. Then

$$E_c \ge \frac{1}{2} \sum_{i=1}^{k} - m_i \lambda_c$$

Introduction

Partitioning of gran

tioning [1, 2], pagin

may also find applica

tices of a graph into

such that the number

a given number, whi

connecting nodes in

problem of compute

what different: for a

see Ref. 1. The part

sion of that problem

in each subset is pr

cency matrix of the

and U any diagonal

(U) is the negative

tices. We derive in T

ber of cut edges in

For the case of div

using a different met

is stricter for this sp

rem 1 turns out to b

suggests exploitation

gramming. Computat

In this paper, we

Graph partitioning

A theorem is given that shows the effect of the maximum degree of any node being limited, and it is also shown that the right-hand side is a concave function of U. Computational studies are made of the ratio of upper bound to lower bound for the two-partition of a number of random graphs having up to 100 nodes.

1973

Czechoslovak Mathematical Journal, 23 (98) 1973. Praha

ALGEBRAIC CONNECTIVITY OF GRAPHS*)

MIROSLAV FIEDLER, Praha (Received April 14, 1972)

1. INTRODUCTION

Let G = (V, E) be a non-directed finite graph without loops and n Having chosen a fixed ordering $w_1, w_2, ..., w_n$ of the set V, we can n-rowed matrix A(G) whose off-diagonal entries are $a_{ik} = a_{kl} = -1$ and $a_{ik} = 0$ otherwise and whose diagonal entries a_{ii} are equal to the the vertices w_i . This matrix A(G), which is frequently used to enumerate trees of the graph G, is symmetric, singular (all the row sums are zero semidefinite $(A(G) = UU^T \text{ where } U \text{ is the } (0, 1, -1) \text{ vertex-edge adj}$ of arbitrarily directed graph G). Let $n \ge 2$ and $0 = \lambda_1 \le \lambda_2 =$... $\leq \lambda_n$ be the eigenvalues of the matrix A(G). From the Perron-Frob applied to the matrix (n-1)I - A(G) it follows that a(G) is zero if a

graph G is not connected. We shall call the second smallest eigenvalue a(G) of the matrix A(G) algebraic connectivity of the graph G. It is the purpose of this paper to find its relation to the usual vertex and edge connectivities.

We recall that many authors, e.g. A. J. HOFFMAN, M. DOOB, D. K. RAY-CHAUD-HURI, J. J. SEIDEL have characterized graphs by means of the spectra of the (0, 1) and (0, 1, -1) adjacency matrices.

2. NOTATION AND CONVENTIONS

The notation introduced above is used throughout the present paper. All matrices and vectors are considered real. The transpose of a matrix M is denoted by M^T , the identity matrix by I, the vector $(1, 1, ..., 1)^T$ by e, the universal matrix ee^T by J, the

For our purpose it is convenient to denote by W the set of all column vectors x

*) Presented at the Graph Theory Meeting in Zlatá Idka, May 1971

2000

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE VOL 22 NO. 8 ALIGLIST 2007

Normalized Cuts and Image Segmentation

Jianbo Shi and Jitendra Malik, Member, IEEE

Abstract—We propose a novel approach for solving the perceptual grouping problem in vision. Rather than focusing on local features and their consistencies in the image data, our approach aims at extracting the global impression of an image. We treat image segmentation as a graph partitioning problem and propose a novel global criterion, the normalized cut, for segmenting the graph. The normalized cut criterion measures both the total dissimilarity between the different groups as well as the total similarity within the groups. We show that an efficient computational technique based on a generalized eigenvalue problem can be used to optimize this criterion. We have applied this approach to segmenting static images, as well as motion sequences, and found the results to be very

Index Terms-Grouping, image segmentation, graph partitioning

N EARLY 75 years ago, Wertheimer [24] pointed out the to be done from the big picture downward, rather like a painter first marking out the major areas and then filling in in vision and listed several key factors, such as similarity. proximity, and good continuation, which lead to visual grouping. However, even to this day, many of the computational issues of perceptual grouping have remained unresolved. In this paper, we present a general framework for this problem, focusing specifically on the

Since there are many possible partitions of the domain I of an image into subsets, how do we pick the "right" one? there may not be a single correct answer. A Bayesian view is variational formulations also exposed two basic questions: appropriate-there are several possible interpretations in the context of prior world knowledge. The difficulty, of course, is in specifying the prior world knowledge. Some of it is low level, such as coherence of brightness, color, texture, or motion, but equally important is mid- or highlevel knowledge about symmetries of objects or object models. The second aspect is that the partitioning is inherently hierarchical. Therefore, it is more appropriate to think of returning a tree structure corresponding to a hierarchical partition instead of a single "flat" partition.

This suggests that image segmentation based on lowlevel cues cannot and should not aim to produce a complete final "correct" segmentation. The objective should instead be to use the low-level coherence of brightness, color, texture, or motion attributes to sequentially come un with hierarchical partitions. Mid- and high-level knowledge can be used to either confirm these groups or select some for further attention. This attention could result in further repartitioning or grouping. The key point is that image partitioning is

J. Shi is with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213. E-mail: ishi@cs.cmu.edu

 J. Malik is with the Electrical Engineering and Computer Science Division, University of California at Berkeley, Berkeley, CA 94720. E-mail: malik@cs berkeleu edu

Manuscript received 4 Feb. 1998; accepted 16 Nov. 1999. Recommended for acceptance by M. Shah. recommenses for acceptance by M. Shan.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107618.

Prior literature on the related problems of clustering. grouping and image segmentation is huge. The clustering community [12] has offered us agglomerative and divisive algorithms; in image segmentation, we have region-based merge and split algorithms. The hierarchical divisive approach that we advocate produces a tree, the dendrogram. While most of these ideas go back to the 1970s (and earlier), the 1980s brought in the use of Markov Random Fields [10] There are two aspects to be considered here. The first is that and variational formulations [17], [2], [14]. The MRF and

- What is the criterion that one wants to optimize?
- Is there an efficient algorithm for carrying out the optimization?

Many an attractive criterion has been doomed by the inability to find an effective algorithm to find its minimum-greedy or gradient descent type approaches fail to find global optima for these high-dimensional, nonlinear

Our approach is most related to the graph theoretic formulation of grouping. The set of points in an arbitrary feature space are represented as a weighted undirected graph G = (V, E), where the nodes of the graph are the points in the feature space, and an edge is formed between every pair of nodes. The weight on each edge, w(i, j), is a function of the similarity between nodes i and j.

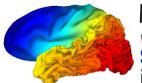
In grouping, we seek to partition the set of vertices into disjoint sets V_1, V_2, \dots, V_m , where by some measure the similarity among the vertices in a set V_i is high and, across different sets V_i , V_i is low.

To partition a graph, we need to also ask the following

- What is the precise criterion for a good partition?
- How can such a partition be computed efficiently?
- In the image segmentation and data clustering community, there has been much previous work using variations of the minimal spanning tree or limited neighborhood set approaches. Although those use efficient computational

0162-8829/00/\$10.00 © 2000 IEEE
Authorized licensed use limited to: Universita degli Studi di Verona. Downloaded on July 16,2020 at 18:56:14 UTC from IEEE Xplore. Restrictions apply.

WORKSHOP ON SPECTRAL GRAPH THEORY SEPTEMBER 23-25/2020



Workshop on

Spectral & Shape Analysis in Medical Imaging

2018

arXiv:1801.01587v6

SPECTRALNET: SPECTRAL CLUSTERING USING DEEP NEURAL NETWORKS

Uri Shaham, Kelly Stanton, Henry Li*

Yale University New Haven, CT, USA

{uri.shaham, kelly.stanton, henry.

Boaz Nadler, Ronen Basri Weizmann Institute of Science

Rehovot, Israel {boaz.nadler, ronen.basri}@gmail.co

Spectral clustering is a leading and popular t tral embedding (i.e., out-of-sample-extension learning approach to spectral clustering that Our network, which we call SpectralNet, points into the eigenspace of their associate quently clusters them. We train SpectralNet strained stochastic optimization. Stochastic of datasets, while the constraints, which are i output layer, allow us to keep the network of learned by SpectralNet naturally generalize data points. To further improve the quality of dard pairwise Gaussian affinities with affinit a Siamese network. Additional improvement network to code representations produced, end-to-end learning procedure is fully unsupe mension theory to derive a lower bound on t clustering results are reported on the Reuters licly available at https://github.com

1 Introduction

Discovering clusters in unlabeled data is a task of significant technological progress images, texts, and other types labeling, however, is often expensive, tedious, or requ provide useful tools to analyze such data and to reve

Spectral Clustering (Shi & Malik 2000: No et al. 20 popular clustering algorithm. It works by embeddi matrix, derived from the pairwise similarities betw representation to obtain the clusters. Several propert its embedding optimizes a natural cost function, m data points: moreover, this optimal embedding can tering variants arise as relaxations of graph balance spectral clustering was shown to outperform other (Von Luxburg, 2007), arguably due to its ability to clustering has a solid probabilistic interpretation,

*Equal contribution.

Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

2014

Learning Deep Representations for Graph Clustering

Fei Tian* University of Science and Technology of China Bin Gao Microsoft Research

yet. The goal of this work is to conduct some preliminary

Clustering aims to group similar patterns among mas-sive data points. Graph clustering is a key branch of clus-

tering, which tries to find disjoint partitions of graph nodes such that the connections between nodes within the same partition are much denser than those across different parti-

tions. On one hand, many real-world problems can be cast as

graph clustering such as image segmentation (Shi and Malik 2000), community detection (Smyth and White 2005), and

VLSI design (Chan, Schlag, and Zien 1994); on the other hand, it is easy to transform a clustering problem in the vec-tor space to a clustering problem on the similarity graph built

from the vector representations of the data points. Therefor

from the vector representations of the data points. Ineretore, we choose to put our focus on graph clustering in this work, and in particular, we investigate the use of stacked sparse autoencoder to perform graph clustering.

Our proposal is motivated by the similarity between au-

toencoder and spectral clustering, a state-of-the-art graph

(Dhillon, Guan, and Kulis 2007)(Satuluri and Parthasarathy

2009), spectral clustering has attracted people's great atten-tion in the past decades due to its solid theoretical foundation

and global optimal solution. Given an n-node graph, spec-tral clustering method runs an Eigenvalue Decomposition (EVD) on the normalized graph Laplacian matrix; then the

eigenvectors corresponding to the k smallest non-zero eigen-

values are extracted as the representation of the graph nodes, where k is the predefined number of clusters; after that, a k-means method is run on the graph representations to get

Note that these k eigenvectors are also the eigenvectors of

clustering method, in terms of what they actually opt Among many existing graph clustering algorithms (Karypis and Kumar 1998)(Shi and Malik 2000) (Van Dongen 2000)

Tie-Yan Liu

estigations along this direction.

Enhong Chen

University of Science and Technology of China cheneh@ustc.edu.cn

Recently deen learning has been successfully adopted in e classification. In this work, we explore the possi bility of employing deep learning in graph clustering ear embedding of the original graph by stacked au-neoder, and then runs k-means algorithm on the embedding to obtain clustering result. We show that thi simple method has solid theoretical foundation, due to the similarity between autoencoder and spectral cluste ine in terms of what they actually ontimize. Then, we demonstrate that the proposed method is more efficien and flexible than spectral clustering. First, the compu ional complexity of autoencoder is much lower that pectral clustering: the former can be linear to the nun ber of nodes in a sparse graph while the latter is su-per quadratic due to eigenvalue decomposition. Sec-ond, when additional sparsity constraint is imposed, we can simply employ the sparse autoencoder developed in the literature of deep learning; however, it is nonstraightforward to implement a sparse spectral method The experimental results on various graph datasets show effectiveness of deep learning in graph clustering

1 Introduction

Deep learning has been a hot topic in the comm of machine learning and artificial intelligence. Many algoof machine learning and artificial intelligence. Many algo-rithms, theories, and large-scale training systems towards deep learning have been developed and successfully adopted in real tasks, such as speech recognition (Dahl et al. 2012), image classification (Krizthevsky, Sutskever, and Hinton 2012), and natural language processing (Collobert et al. 2011). However, to our knowledge, the adoption of deep learning in clustering has not been adequately investigated

*This work was done when the two authors were visiting Mi-Copyright © 2014. Association for the Advancement of Artificial nce (www.aaai.org). All rights reserved.

responding to the $|\sqrt{k}|$ smallest non-zero eigenvalue

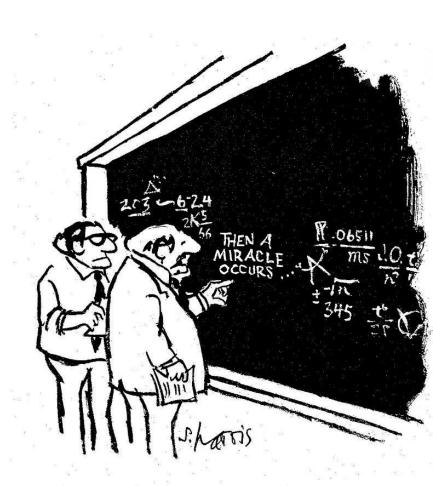
two that these k eigenvectors are also the eigenvectors of the normalized graph similarity matrix, whereas corresponding to its k largest eigenvalues. Therefore these eigenvectors can be regarded as an encoding of the normalized graph similarity matrix, and according to the Eckart-Young-Mirsky

cardinality of a set S by |S|.

such that $x^Tx = 1$, $x^Te = 0$. Any square matrix M with all zero row sums has an

Index

- Basic concepts (linear algebra, function analysis)
- Dirichlet energy and smoothness
- Laplacian
- Graph Laplacian
- Eigen-decomposition of the Laplacian
- Spectral properties
- Applications (mincut, clustering, Fourier-like analysis)



Basis

Given a vector space **V**, a subset **B** is a *basis* iif:

- Its elements are linear indipendent
- ullet And they span all the vectors in $oldsymbol{\mathsf{V}}$

The canonical base for $\mathbf{v} \in \mathbb{R}^3$

$$\alpha_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \alpha_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{v}$$

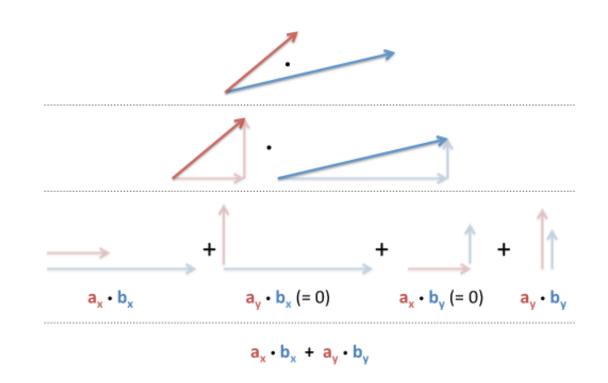
Every **v** can be written as a weighted sum of the elements of the basis.

Basis - Properties

Defining an inner product:

$$\langle \mathbf{a}, \mathbf{b} \rangle = a_0 b_0 + a_1 b_1 + \cdots + a_n b_n = \sum_{i=0}^n a_i b_i$$

It is useful to see it as a projection operation



We say our basis $B = \{b_0, b_1, \dots, b_n\}$ is:

Orthogonal
$$\Longrightarrow \forall i, j \in [0, n]$$
, if $i \neq j$ then $\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0 \Longrightarrow$ Projection gives nothing



$$\forall i, ||\mathbf{b}_i||_2 = \sqrt{\langle \mathbf{b}_i, \mathbf{b}_i \rangle} = 1$$

Projection preserves length

Basis - Coefficients

$$\mathbf{v} = \mathbf{b}_1 c_1 + \mathbf{b}_2 c_2 + \mathbf{b}_3 c_3 + \dots$$

$$\mathbf{v} \cdot \mathbf{b}_2 = \left(\mathbf{b}_1 c_1 + \mathbf{b}_2 c_2 + \mathbf{b}_3 c_3 + \dots\right) \cdot \mathbf{b}_2$$

$$\frac{\mathbf{v} \cdot \mathbf{b}_2 - \mathbf{b}_2 \mathbf{b}_1 c_1 - \mathbf{b}_2 \mathbf{b}_3 c_3 - \dots}{||\mathbf{b}_2||_2} = c_2$$
If \mathbf{B} is orthonormal $\Longrightarrow \langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0$

$$||\mathbf{b}_2||_2 = 1$$

$$\Longrightarrow \mathbf{v} \cdot \mathbf{b}_2 = c_2$$

Each coefficent is indipendent and comes from a multiplication.

Basis - Functions

For functions:

$$f(x) = \alpha_1 g_1(x) + \alpha_2 g_2(x) + \alpha_3 g_3(x) + \dots$$

Weirestrass Theorem (of approximation):

Suppose f is a continuous real-valued function defined on the real interval [a,b]. For every $\epsilon>0$, there exists a polynomial p such that for all x in [a,b], we have $|f(x)-p(x)|<\epsilon$

i.e., Given a function, there exist a polynomial arbitrary close to it.

Monomials are a basis for the functional space:

$$G(x) = \{g_i(x)\} = \{x^i\} = \{1, x, x^2, x^3, \dots\}$$

Basis – Coefficients for functions

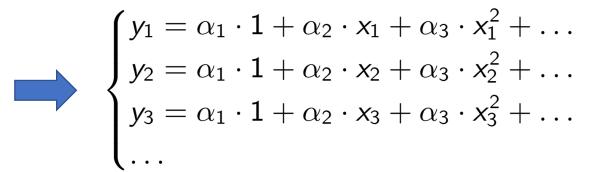
$$G(x) = \{g_i(x)\} = \{x^i\} = \{1, x, x^2, x^3, \dots\}$$

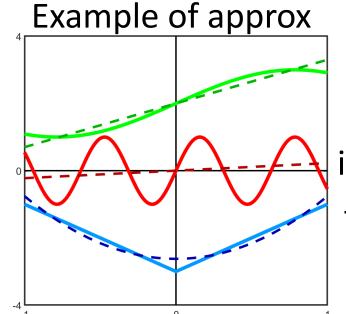
We use polynomial interpolation by sampling the function:

$$\begin{cases} y_1 = \alpha_1 g_1(x_1) + \alpha_2 g_2(x_1) + \alpha_3 g_3(x_1) + \dots \\ y_2 = \alpha_1 g_1(x_2) + \alpha_2 g_2(x_2) + \alpha_3 g_3(x_2) + \dots \\ y_3 = \alpha_1 g_1(x_3) + \alpha_2 g_2(x_3) + \alpha_3 g_3(x_3) + \dots \\ \dots \end{cases}$$

$$\begin{bmatrix} y1 \\ y2 \\ y3 \\ \dots \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots \\ 1 & x_2 & x_2^2 & \dots \\ 1 & x_3 & x_3^2 & \dots \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \dots \end{bmatrix}$$

$$y = Ga$$
 with a unknown $\implies G^{-1}y = a$





Coefficients are informative on the function behavior

Basis - Coefficients for functions (infinite sampling)

Considering infinite sampling
$$\mathbf{x} = \begin{bmatrix} -1 \\ -1 + dx \\ -1 + 2dx \\ ... \\ 1 - dx \\ 1 \end{bmatrix} \mathbf{y} = \begin{bmatrix} f(-1) \\ f(-1 + dx) \\ f(-1 + 2dx) \\ ... \\ f(1 - dx) \\ f(1) \end{bmatrix}$$

Defining an inner product $\langle \mathbf{f}, \mathbf{g} \rangle = \begin{bmatrix} f(-1) \\ f(-1) \\ f(-1 + dx) \\ f(-1 + 2dx) \\ ... \\ f(1 - dx) \\ f(1) \end{bmatrix} \cdot \begin{bmatrix} g(-1) \\ g(-1 + dx) \\ g(-1 + 2dx) \\ ... \\ g(1 - dx) \\ g(1) \end{bmatrix}$

$$= f(-1)g(-1) + f(-1 + dx)g(-1 + dx) + \cdots + f(1)g(1)$$

$$= \int_{-1}^{1} f(x)g(x)dx$$

Basis – Properties for functions

Orthogonality

$$\langle v, w \rangle = 0$$

Vectors
$$\langle v,w\rangle=0$$
 Functions $\langle f,g\rangle=\int_{-1}^1 f(x)g(x)dx=0$

$$\langle 1, x \rangle = \int_{-1}^{1} 1 \cdot x dx = \frac{x^2}{2} \Big|_{-1}^{1} = \frac{1}{2} - \frac{1}{2} = 0$$

$$\langle x, x^2 \rangle = \int_{-1}^{1} x \cdot x^2 dx = \frac{x^4}{4} \Big|_{-1}^{1} = \frac{1}{4} - \frac{1}{4} = 0$$

$$\langle 1, x \rangle = \int_{-1}^{1} 1 \cdot x dx = \frac{x^{2}}{2} \Big|_{-1}^{1} = \frac{1}{2} - \frac{1}{2} = 0$$

$$\langle x, x^{2} \rangle = \int_{-1}^{1} x \cdot x^{2} dx = \frac{x^{4}}{4} \Big|_{-1}^{1} = \frac{1}{4} - \frac{1}{4} = 0 \qquad \langle 1, 1 \rangle = \int_{-1}^{1} 1 dx = x \Big|_{-1}^{1} = 1 + 1 = 2$$

$$\langle 1, x^{2} \rangle = \int_{-1}^{1} 1 \cdot x^{2} dx = \frac{x^{3}}{3} \Big|_{-1}^{1} = \frac{1}{3} + \frac{1}{3} = \frac{2}{3}$$

Normality

$$\langle \mathsf{v}, \mathsf{v}
angle = 1$$

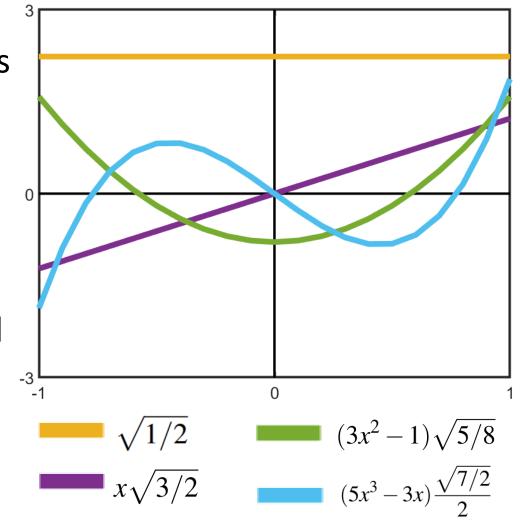
$$\langle f, f \rangle = \int_{-1}^{1} f(x)^2 = 1$$

$$\langle 1, 1 \rangle = \int_{-1}^{1} 1 dx = x \Big|_{-1}^{1} = 1 + 1 = 2$$

Basis – Orthonormalization for functions

Gram-Schmidt process over the monomial basis produces the **Legendre Polynomials**, that are orthonormals.

There are other useful properties for functional basis?

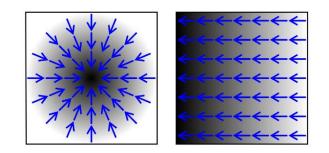


Gradient, Divergence, Laplacian

Important tools from analysis:

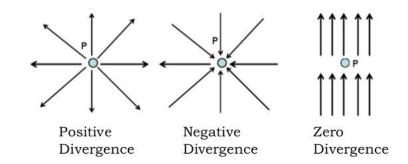
The gradient

$$\nabla : \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}, \dots\right)$$



The divergence

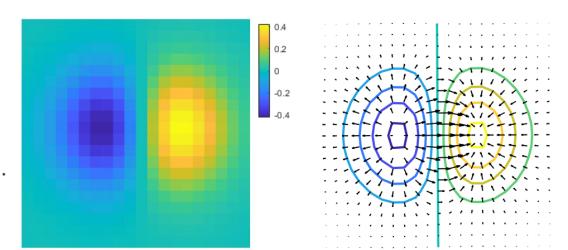
$$div: \frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z} + \dots$$

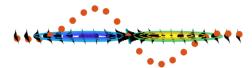


The Laplacian

$$\Delta$$
: $div \nabla = \frac{\partial^2}{\partial^2 x} + \frac{\partial^2}{\partial^2 y} + \frac{\partial^2}{\partial^2 z} + \dots$

The change of a rate of change





Note: in 1D, the Laplacian is equal to the second order derivative.

Smoothness – Dirichlet Energy

Dirichlet Energy:

$$E(f) = \int_X ||\nabla f(x)||^2 dx$$

It measures the variability of the function.

A miracle occours

First Green Identity:

$$E(f) = \int_X ||\nabla f(x)||^2 dx = \dots = \int_X f(x) \Delta f(x) dx$$

We want the orthonormal basis that minimizes this energy.

Smoothness – (Discrete) Dirichlet Energy

$$E(f) = \int_{X} f(x) \Delta f(x) dx \qquad E(\phi_{i}) = \int_{X} \phi_{i}(x) \Delta \phi_{i}(x) dx = \langle \phi_{i}, \Delta \phi_{i} \rangle$$

$$\mathbf{\Phi}_{k} = \begin{bmatrix} \phi_{0} & \phi_{1} & \phi_{2} & \dots & \phi_{k} \end{bmatrix}$$

$$\mathbf{\Phi}_{k} \in \mathbb{R}^{n \times k}$$

$$\min_{\phi_{0}} E(\phi_{0}) \quad \text{s.t.} \quad ||\phi_{0}|| = 1$$

$$\min_{\phi_i} E(\phi_i)$$
 s.t. $||\phi_i|| = 1$, $\phi_i \perp \operatorname{span}\{\phi_0, \ldots, \phi_{i-1}\}$

$$\min_{\mathbf{\Phi}_k} \operatorname{trace}(\mathbf{\Phi}_k^T \Delta \mathbf{\Phi}_k) \quad \text{s.t.} \quad \mathbf{\Phi}_k^T \mathbf{\Phi}_k = \mathbf{I}$$

Smoothness – (Discrete) Dirichlet Energy

$$\min_{\mathbf{\Phi}_k} \operatorname{trace}(\mathbf{\Phi}_k^T \Delta \mathbf{\Phi}_k) \quad \text{s.t} \quad \mathbf{\Phi}_k^T \mathbf{\Phi}_k = \mathbf{I}$$

$$\Delta \Phi_{\mathbf{k}} = \Phi_{k} \Lambda_{k}$$

The set of eigenvectors (also called *eigenfunctions*) of the Laplacian is an orthonormal basis, optimal for the smoothness.

In the 1D case they are the standard Fourier basis:

$$\Delta \sin (n\theta) = -|n|^2 \sin (n\theta)$$
$$\Delta e^{n2\pi i\theta} = -|n|^2 4\pi^2 e^{n2\pi i\theta}$$

First part summary

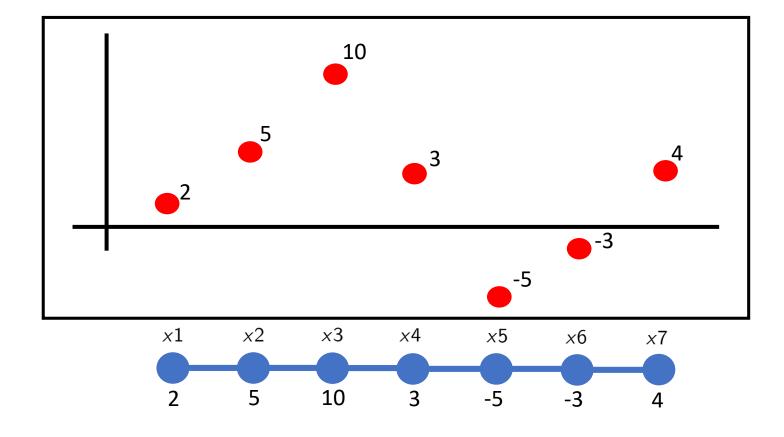
We have seen so far:

- The concept of basis
- The properties of basis (Orthogonality and Normality)
- How to extend these to the functions
- Some main tools: gradient, divergence, Laplacian
- The Dirichlet Energy to add a new property (Smoothness)
- The link with Fourier Basis

Key idea: given a domain, we need a Laplacian.



Discrete Setting



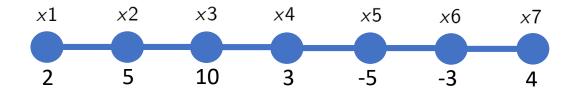
Vertices $\{x1, x2, x3, x4, x5, x6, x7\}$

Edges connect consecutive points

The y-value is a function on the graph $F: V \to \mathbb{R}$

Discrete Setting

$$f(x) = \begin{bmatrix} 2 & 5 & 10 & 3 & -5 & -3 & 4 \end{bmatrix}$$



$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$$

$$f'(x) = \begin{bmatrix} 3 & 5 & -7 & -8 & 2 & 7 & \dots \end{bmatrix}$$

$$f''(x_i) = \frac{f'(x_i) - f'(x_i - 1)}{h} = \cdots = \frac{f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)}{h^2}$$

assuming h = 1:

$$f''(x_i) = f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)$$

$$f''(x) =$$
 ... 2 -12 -1 10 5 ...

Encode distance of a point from the mean of the neighboor values.

1D Discrete Laplacian

$$f(x) = \begin{bmatrix} 2 & 5 & 10 & 3 & -5 & -3 & 4 \end{bmatrix}$$

$$f''(x_i) = f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)$$

$$f''(x_i) = Lf = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \\ 10 \\ 3 \\ -5 \\ -3 \\ 4 \end{bmatrix} = \begin{bmatrix} \dots \\ 2 \\ -12 \\ 10 \\ 5 \\ \dots \end{bmatrix}$$

$$L_{ij} = \begin{cases} 1 & \text{if } i \neq j, i \sim j \\ -d_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \text{ or more generally } \begin{cases} w_{ij} & \text{if } i \neq j, i \sim j \\ -\sum_{j \neq i, j \sim i} w_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

L = W - D, where D is the diagonal matrix of the degrees, and W is the weighted adjacency matrix.

Laplacian Eigenvectors

$$E(f) = \int_{X} ||\nabla f(x)||^{2} dx = \int_{X} f(x) \Delta f(x) dx$$

$$E(f) = \sum_{i} \langle \nabla f, \nabla f \rangle = \sum_{i} f_{i}(\Delta f)_{i}$$

$$\sum_{(i,j)\in E} (f_i - f_j)^2 = f^T L f$$

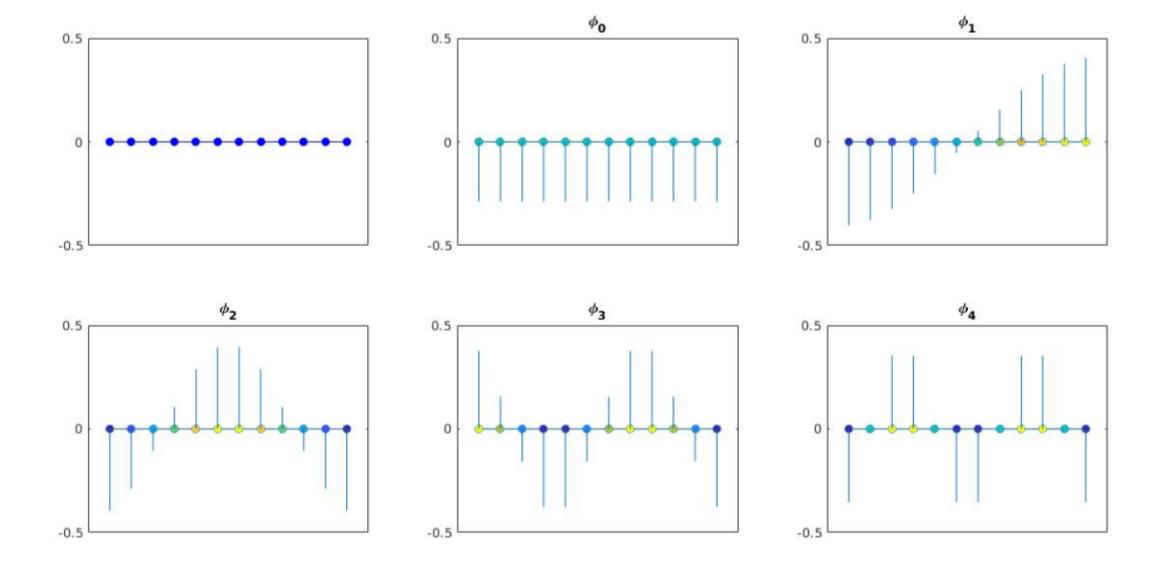
$$\lambda = f^T L f$$

$$\lambda f = L f$$

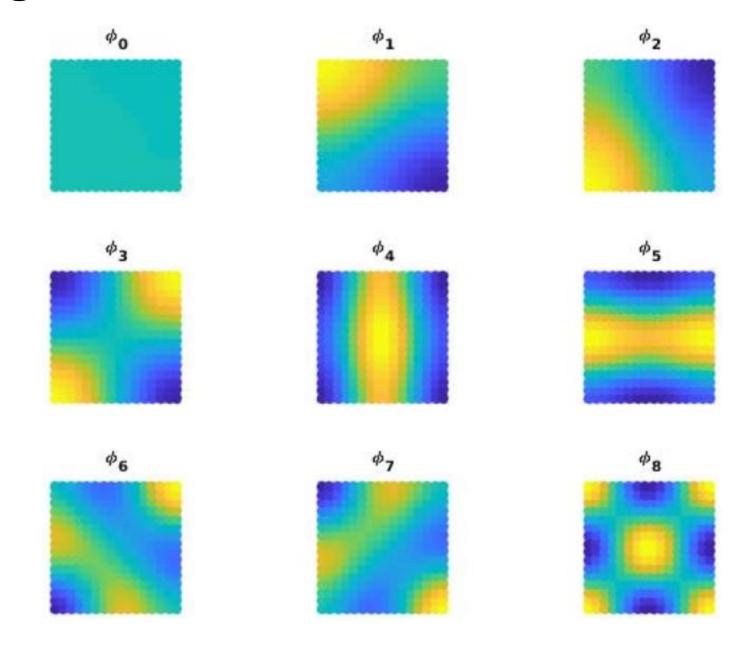
L is symmetric its eigenvalues are real and non-negative

$$\lambda_i = \phi_i^T L \phi_i \geq 0$$

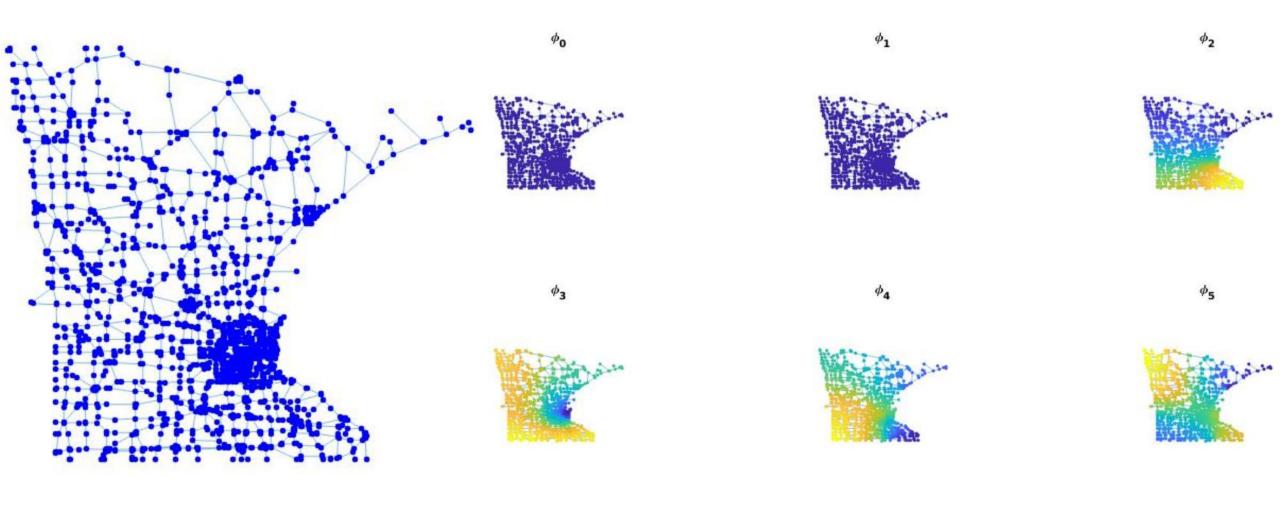
Laplacian Eigenvectors – 1D



Laplacian Eigenvectors – 2D



Laplacian Eigenvectors – Generic Graph



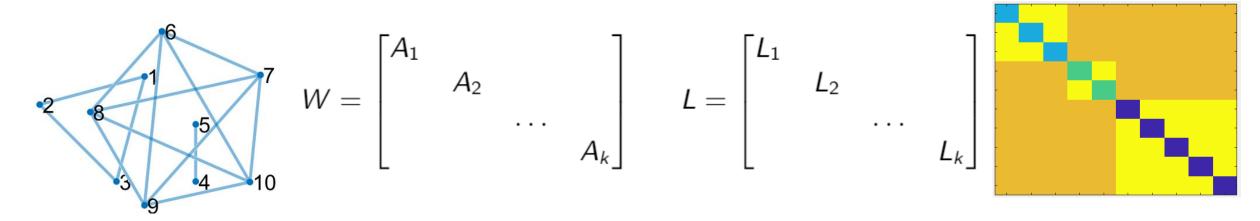
Disconnected components

Consider the eigenvalues 0 on a connected graph:

$$\sum_{(i,j)\in E} (\phi_i - \phi_j)^2 = \phi^T L \phi = 0 \implies (\phi_i - \phi_j)^2 = 0, \forall (i,j) \implies \phi_i = \phi_j$$

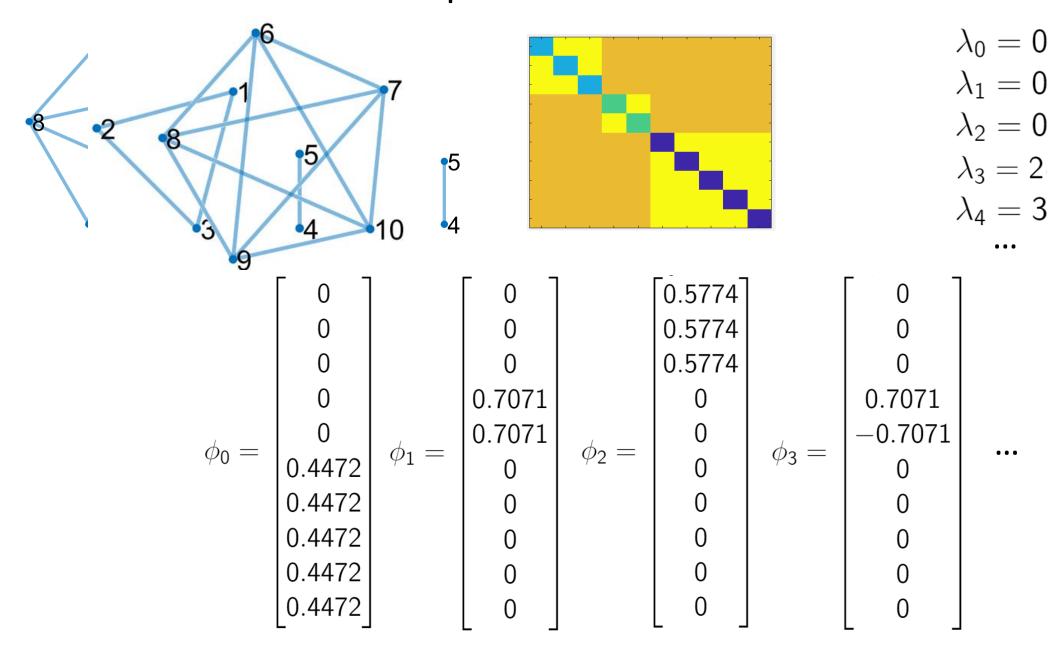
 ϕ_0 is the constant function.

Consider the adjency matrix on a disconnected graph \implies Block matrix

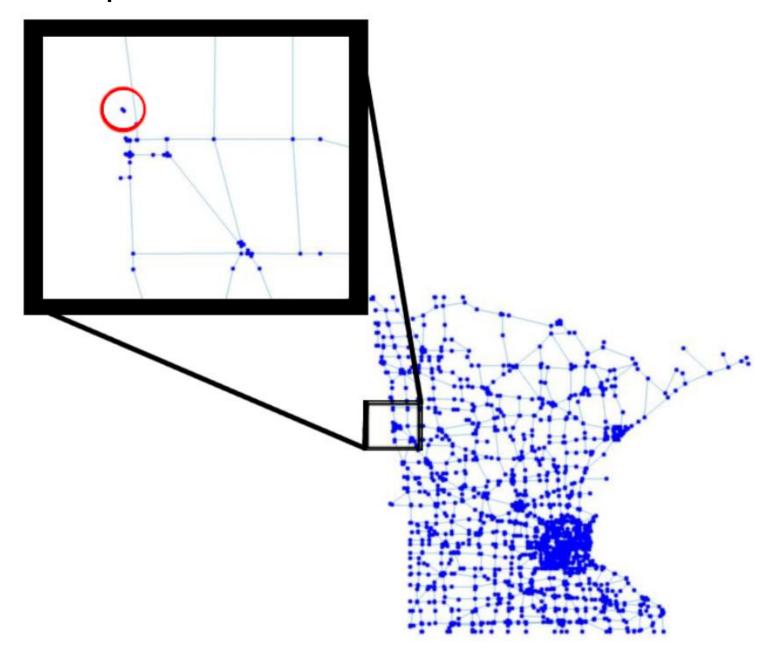


The eigenfunctions of a block matrix are the union of the eigenfunctions of the blocks \implies 0s eigenvalues as many as connected components.

Disconnected components



Disconnected components



Min-cut

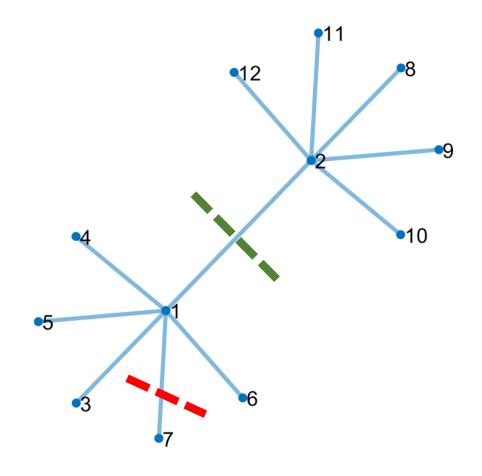
Minimum number of edges to remove to disconnect the graph.

Given a Graph (V,E), find the minimum cut s.t. the two disjoint sets $A, B \subset V$

$$min cut(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

Green and red cuts have the same cost. In general we prefer a *balanced* cut.

$$|A_1| = |\bar{A}_1| = 0.5$$



Ratiocut

Approximation of a balanced cut

min Ratiocut
$$(A_1, \ldots, A_2) = \sum_{i=1}^k \frac{cut(A_i, \bar{A}_i)}{|A_i|}$$

Case of split in two

$$\mathsf{min}\,\mathsf{Ratiocut}(A_1,ar{A_1}) = rac{\mathsf{cut}(A_1,ar{A_1})}{|A_1|} + rac{\mathsf{cut}(ar{A_1},A_1)}{|ar{A_1}|} = \epsilon$$

$$|\bar{A}_1|cut + |A_1|cut = \epsilon |A_1||\bar{A}_1|$$

$$\frac{cut}{|A_1||\bar{A}_1|} = \frac{\epsilon}{|V|}$$

$$\mathsf{min}\,\mathsf{Ratiocut}(A_1,\bar{A_1}) = \frac{\mathit{cut}}{|A_1||\bar{A_1}|}$$

maximize $|A_1||\bar{A}_1| \implies |A_1| = |\bar{A}_1| = 0.5$ minimize $cut \implies$ few «bridge» edges

Ratiocut

$$\sum_{(i,j)\in E} (\phi_{1_i} - \phi_{1_j})^2 = \lambda_1 \quad \text{Minimize Dirichlet Energy}$$

$$\phi_0 \perp \phi_1 \implies \langle \phi_0, \phi_1 \rangle = 0 \implies \sum_i \phi_{0_i} \phi_{1_i} = 0 \implies \phi_{1_1} + \phi_{1_2} + \dots = 0$$

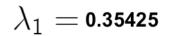


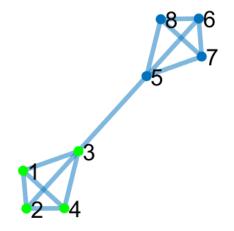
First eigenfunction is constant

Also:

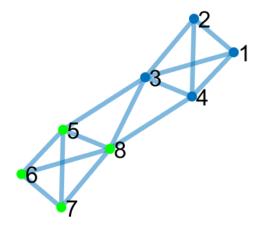
 $||\phi_1||=1$ Split the graph eavenly by sign \longrightarrow Relaxed version of RatioCut $\phi_{1_1}+\phi_{1_2}+\cdots=0$

Fiedler value

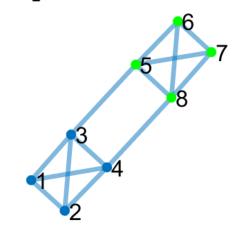




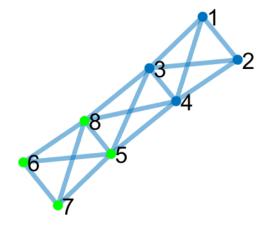
$$\lambda_1=$$
 0.94863



$$\lambda_1=$$
 0.76393



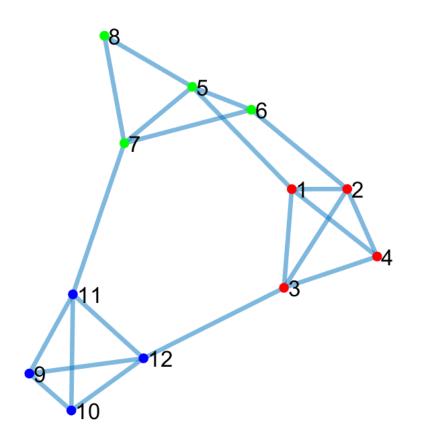
$$\lambda_1=$$
 1.1716



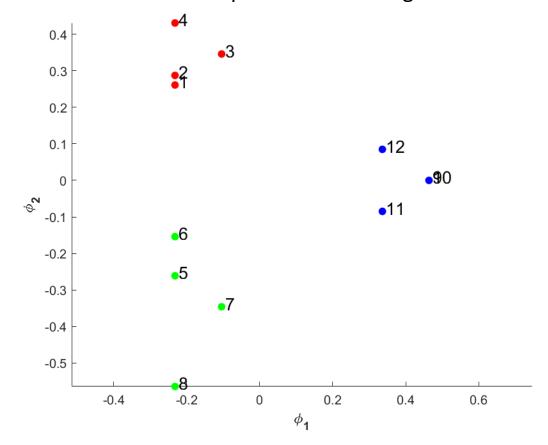
Spectral Clustering

$$\phi_1, \phi_2, \dots, \phi_d$$

$$\{\phi_1(v), \phi_2(v), \dots, \phi_d(v)\}$$



Spectral Embedding

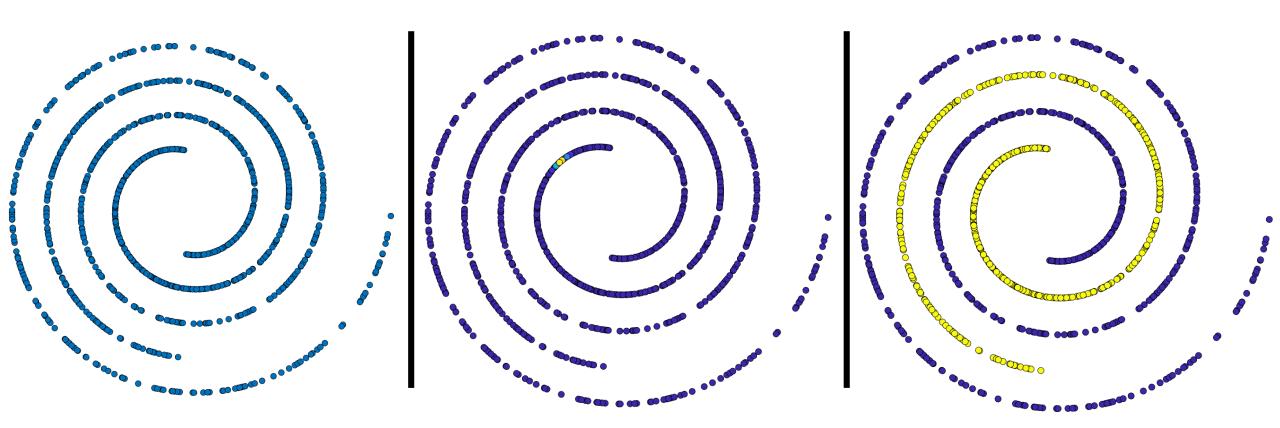


Spectral Clustering – 2D

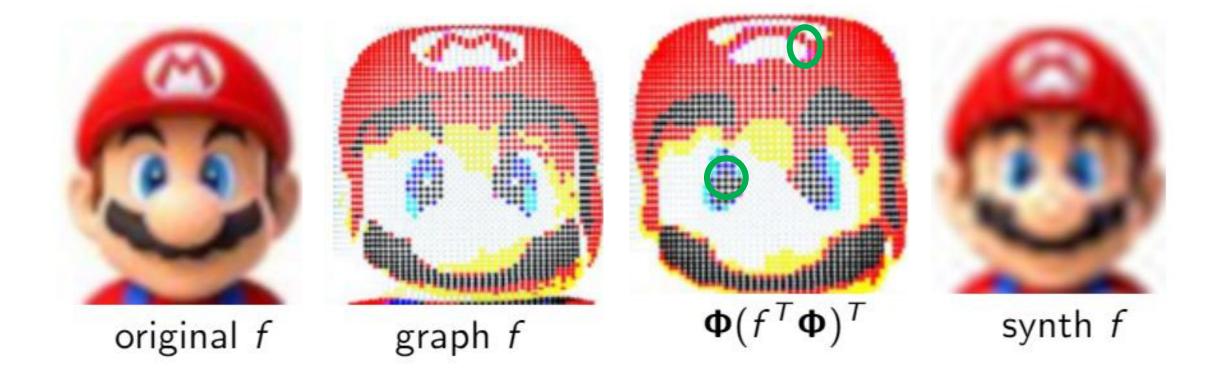
We can build a fully-connected adjancy matrix setting the weight as:

$$d(x_i,x_j)=e^{\frac{-|x_i-x_j|}{2\sigma^2}}$$

Other strategies are possible (e.g. K-NN).



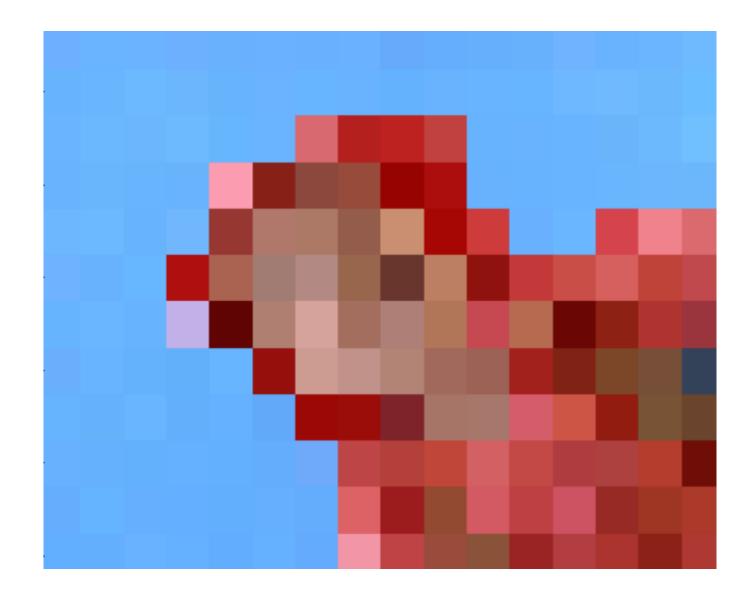
Graph Signal Processing

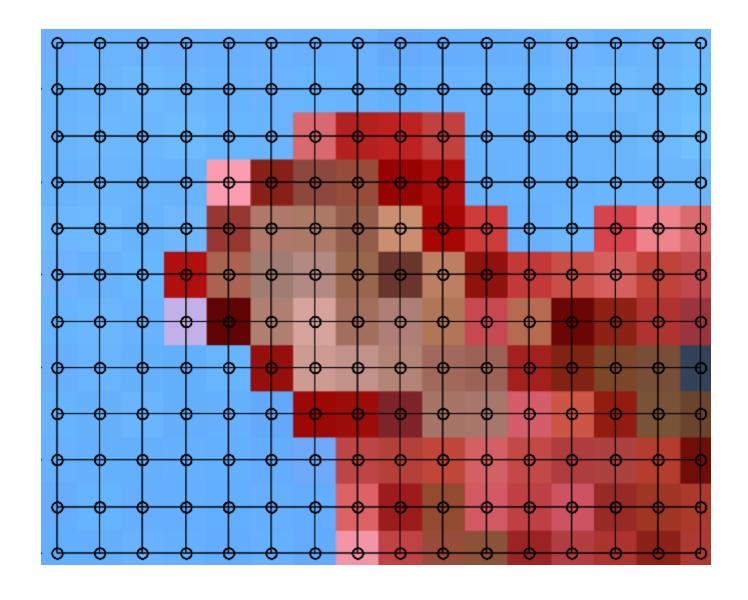


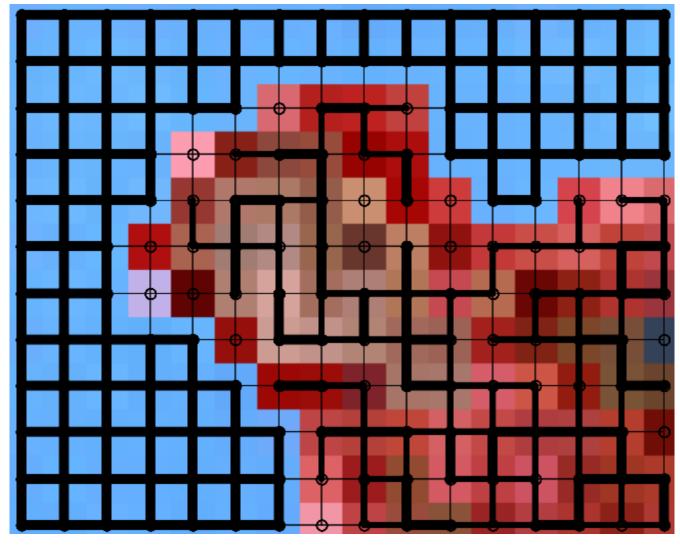
Denoising and compression

This image is $64 \times 64 = 4096$ nodes, for the 3 RGB channels: 12288 values. We have represented 3 signals with the first 500 eigenfunctions \implies 1500 values.







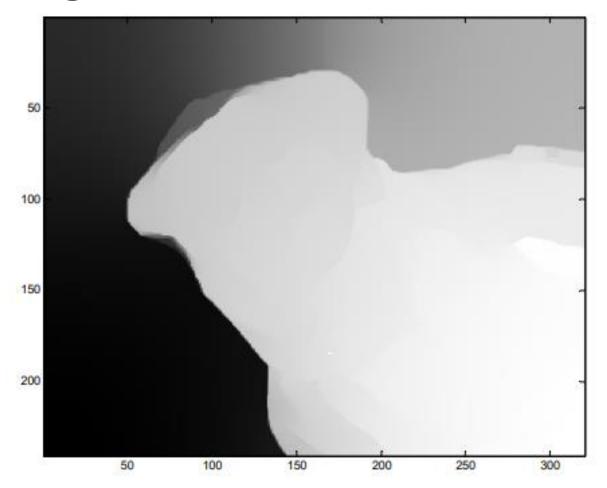


edge weight $e^{-{
m diff}({
m pixel}i,{
m pixel}j)^2/t^2}$

https://www.cs.yale.edu/homes/spielman/sgta/SpectTut.pdf

Eigen Segm.

Segmentation (Miller and Tolliver '06)





For a graph $G=(V,E,w^0)$ prescribe the number of partitions k that the edge cut is to yield. Given a valid **reweighting scheme**, iteration of the SR-Step produces a sequence of N weightings $\{w^{(N)}\}$ such that the graph $G^N=(V,E,w^N)$ is disconnected into k components by the weighting w^N .

Segmentation (Neto and Felzenszwalb '20)



Both graphs are defined over the same set of vertices, corresponding to the pixels in an image.

- 1. The graph G_{grid} is a grid over the image pixels, where each pixel is connected to the four neighboring pixels with an edge of weight 1. This graph encourages neighboring pixels to be grouped together, independent of their appearance.
- 2. The graph G_{data} is a fully connected graph that encourages pixels with similar appearance to be grouped together, independent of their location. The weights in G_{data} are based on appearance similarity of pixels, and do not depend on pixel locations,

$$w(i,j) = \exp\left(-\frac{||I(i) - I(j)||^2}{2\sigma^2}\right).$$
 (6)

In our experiments, we use a Lanczos Process to compute the second largest eigenvector of P.

Summary

We have learned:

- How to extend Laplacian to non-euclidean domain
- Why their eigenfunctions are so important
- How this operator and its eigendecomposition touch the connectivity
- Several applications of its eigendecomposition

! Exercises!

https://github.com/riccardomarin/SpectralShapeAnalysis

It could take some time to configure the environment Lunch it in advance

