CMSC 35410: Spectral Methods for Machine Learning and Network Analysis Autumn 2023

Lecture 3: Spectral Graph Theory Basics

Instructors: Lorenzo Orecchia, Konstantinos Ameranis

1 The Computer Science way

In Computer Science, Spectral Graph Theory is usually introduced by equating graphs with matrices[Chu, Spi07].

Undirected Graphs Unless explicitly stated, all graphs that we will deal with in this part of the course are undirected graphs, i.e. the edges are unordered pairs of vertices and have no direction. They will also be loopless and simple (no multiple edges). A (weighted) undirected graph is formally described as a tuple $G = (V, E, \mathbf{w} \in \mathbb{R}^E_{>0})$, where V is a set of vertices, $E \subseteq V \times V$ is a collection of edges, i.e. unordered pairs of vertices from V and $\mathbf{w}$ are the associated positive weights. The graph is unweighted if $\mathbf{w} = \mathbf{1}$. As we only consider finite graphs, we can assume without loss of generality that $V = [n] = \{i \in \mathbb{N} : 1 \leq i \leq n\}$, for some integer $n \geq 1$.

These assumptions may seem restrictive, but they are exactly what is needed for modeling interactions between pairs of vertices where the weight may represent the strength of the interaction. For instance, the weights of the edges may represent a measure of similarity between their endpoints, as in similarity-based learning. The main reason why we want to limit ourselves to undirected graphs is that they can naturally be represented as *symmetric* square matrices.

The Adjacency Matrix The first graph matrix we introduce is the most intuitive representation of an undirected graph.

Definition 1. The adjacency matrix $\mathbf{A}_G$ of a graph $G = (V, E, \mathbf{w})$ is a $n \times n$ square matrix with

$$(\mathbf{A}_G)_{ij} = \begin{cases} w_{ij}, & \{i,j\} \in E \\ 0, & \text{otherwise} \end{cases}$$

It should be clear by the undirectedness of G that $\mathbf{A}_G$ is symmetric. While the adjacency matrix is a simple way of defining a graph matrix, it turns out that a normalized version of the adjacency defines a more natural operator of $\mathbb{R}^n$: the random walk operator. To define this operator, we first need to introduce an auxiliary graph matrix, the degree matrix.

The Degree Matrix Recall that the degree d_i of a vertex $i \in V$ for a weighted graph $G = (V, E, w)$ is defined as $d_i \overset{\text{def}}{=} \sum_{\{i,j\} \in E} w_{ij}$. The degree distribution will often be the natural measure over the vertices of the graph. For this reason, it is convenient to have a diagonal matrix representing degrees.

Definition 2. The degree matrix D_G of $G = (V, E, w)$ is the diagonal matrix such that, for all $i \in [n]$

$$(\mathbf{D}_G)_{ii} = d_i = \sum_{\{i,j\} \in E} w_{ij}.$$

A quick observation: when the graph G is regular, i.e., all degrees are equal to some d, then the degree matrix is just a multiple of the identity, $\mathbf{D}_G = d\mathbf{I}$. For the rest of the course, we will omit the subscript indicating the graph G when the context is clear.

1.1 Natural Random Walk on G

The Natural Random Walk and Its Matrix Consider a particle moving in a random fashion over the vertices V of the graph G and transitioning from vertex to vertex at discrete time intervals. Let $X(t) \in V$ be a random variable describing the position of the particle at time $t \in \mathbb{Z}$. The weights on the edges of G regulate the random transitions of $X(t)$ as follows: suppose that at time t the particle is at vertex v, that is $X(t) = v$. At time $t + 1$, the particle randomly picks an incident edge $\{u, v\}$, with probability proportional to its weight w_{uv}. It then moves to the other end of the edge,i.e., $X(t + 1) =$ u. Formally:

$$\Pr[X(t+1) = u | X(t) = v] = \begin{cases} \frac{w_{uv}}{d_v}, & \{u, v\} \in E \\ 0, & \text{ow.} \end{cases} \tag{1}$$

We refer to this discrete-time Markov chain as the *natural random walk over the graph.* Let $\mathbf{p}^{(t)} \in \mathbb{R}^n_{\geq 0}$ be a vector describing the probability distribution of $X(t)$ over the vertices of the graph. By Equation 1, we can compute the evolution of $p^{(t)}$ as follows:

$$p_i^{(t+1)} = \sum_{\{i,j\} \in E} \Pr\{X_{t+1} = i | X_t = j\} \Pr\{X_t = j\} = \sum_{\{i,j\} \in E} \frac{w_{ij}}{d_j} \cdot p_j^{(t)} = (\mathbf{A}\mathbf{D}^{-1}\mathbf{p}^{(t)})_i. \tag{2}$$

In words, when we apply one step of the random walk the amount of probability mass at i is the sum of all the contributions from neighbors of i.

Definition 3. The random walk matrix $\mathbf{P}_G$ of $G = (V, E, w)$ is defined as

$$\mathbf{P}_G = \mathbf{A}_G \mathbf{D}_G^{-1}$$

Formally, the random walk matrix $\mathbf{W}_G$ is also the transition probability matrix for the Markov chain $X(t)$. Note that there is nothing random about W_G. It just helps in describing the probability distributions arising from the random walk process.

The Laplacian Matrix The random walk matrix described a natural process, but it is unfortunately not symmetric in general, unless the graph is regular. A related symmetric matrix that turns out to be the most useful for Spectral Graph Theory is the Laplacian matrix. Here is one definition of the Laplacian. We will see another in the next chapter.

Definition 4. The Laplacian matrix $\mathbf{L}_G$ of $G = (V, E, \mathbf{w})$ is the matrix $\mathbf{L}_G \overset{\text{def}}{=} \mathbf{D}_G - \mathbf{A}_G$. Equivalently:

$$(\mathbf{L}_G)_{ij} = \begin{cases} d_i, & i = j \\ -w_{ij}, & \{i, j\} \in E \\ 0, & \text{otherwise} \end{cases}$$

Next Steps The next steps in the standard CS development of the subject is to apply the spectral theorem to the symmetric matrix $\mathbf{L}_G$ to understand the behavior of $\mathbf{P}_G^t$ by exploiting the relation

$$\mathbf{P}_G = \mathbf{I} - \mathbf{L}_G \mathbf{D}_G^{-1}.$$

By contrast, in this course, we will discuss more in depth how the Laplacian matrix arises as a discrete analogue of the Laplacian differential operator over the space of functions on a continuous domain, such as a manifold.

2 The Partial Differential Equations (PDE) way

Historically, spectral methods were developed in the late 1970s to solve certain families of smooth partial differential equations over $\mathbb{R}^n$. Indeed, throughout the course, we will often use the same techniques to analyse and implement solvers for the discrete analogues of these PDEs. Because I believe that the continuous pictures, albeit mathematically harder to formulate, reveals some important intuition about spectral methods over graphs, in this section I will try to introduce and motivate the basic operators in spectral graph theory via an informal discussion of their continuous counterparts. A great reference for all continuous instances of the Laplacian operator is the book "The Laplacian on a Riemannian Manifold" by Rosenberg [Ros97].

2.1 The Laplacian Δ over Compact Subsets of $\mathbb{R}^n$

We will define the Laplacian Δ as the second-order differential operator defined by:

$$(\Delta f)(\mathbf{x}_0) = -\sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}\Big|_{\mathbf{x}_0} = -\nabla \cdot \nabla f = -\mathrm{div}\,(\nabla f)(\mathbf{x}_0)$$

Even though this can be defined for any twice-differentiable function over $\mathbb{R}^n$, for this discussion it will be easier to consider the action of Δ over the space $C_0^\infty([0,1]^n)$, i.e., smooth functions over $[0,1]^n$ that vanish on the boundary. Notice that Δ acts as a linear operator over this space, taking in $f \in C_0^\infty([0,1]^n)$ and returning another function Δf in the same vector space of functions. We can also equip this space with the following natural notion of inner product:

$$\langle f, g\rangle = \int_{[0,1]^n} f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}.$$

This allows us to define a notion of adjoint, i.e. a generalization of the transpose, for the linear operator Δ. By applying integration by parts twice and exploiting the compactness of the support of f and g, we find that:

$$\langle \Delta f, g\rangle = \langle \nabla f, \nabla g\rangle = \langle f, \Delta g\rangle.$$

This shows that the linear operator Δ is a self-adjoint operator, the infinite-dimensional analogue of a symmetric matrix. Hence, we may expect Δ to exhibit an analogue to an orthonormal basis of eigenvectors. Indeed, such a basis is given by the cosine functions $\{\cos(2\pi\, \boldsymbol{\lambda} \cdot \mathbf{x})\}_{\boldsymbol{\lambda} \in \mathbb{Z}^n}$, which are just the real part of the Fourier basis functions. It is not hard to compute the corresponding eigenvalues:

$$(\Delta \cos(2\pi\, \boldsymbol{\lambda} \cdot \mathbf{x}))(x_0) = -\|\boldsymbol{\lambda}\|^2 \cdot \cos(2\pi\, \boldsymbol{\lambda} \cdot \mathbf{x}).$$

It is possible to extend this reasoning to more interesting function spaces such as functions over $\mathbb{R}^n$, where the orthonormal systems yields the Fourier transform. However, the correct choice of function space for this piece of math to work is fairly complex (see Rigged Hilbert Space).

Applications of the Laplacian The Laplacian is used in three very important PDEs. Three out of four fundamental PDEs in Chapter [Eva22] Evans's standard PDE textbook involve the Laplacian. They are:

- **Laplace's equation** describes the steady state in a field

$$\Delta f = 0$$

- **Heat equation** connects the time derivative to the spatial gradient. Given a function $f^{(t)} : [0, \infty) \times \mathbb{R}^n$

$$\frac{df}{dt} = -\Delta f$$

- **Wave equation** describes a standing wave

$$\frac{d^2 f}{dt^2} = -\Delta f$$

What makes the Laplacian such an ubiquitous object in these models of natural phenomena? To answer this question, it is helpful to visualize the interpretation of the Laplacian as the divergence of the gradient of a function (or potential) f. You may equivalently think of $f(\mathbf{x})$ as a temperature or the concentration of a substance at $\mathbf{x}$. This difference in temperature (or concentration) leads to a flow of heat (or of the substance) from high temperature to low temperature regions. This takes the form of a vector field $-\nabla f(\mathbf{x})$. When flows moves along this vector field, the net in-flux at a point $\mathbf{x}$ is given by the negative divergence of $-\nabla f(\mathbf{x})$. Combining these two facts, the change in temperature at $\mathbf{x}$ becomes:

$$\frac{df}{dt}(\mathbf{x}) = (-\operatorname{div})(-\nabla f(\mathbf{x})) = -(\Delta f)(\mathbf{x}).$$

Generalized Laplacian We can slightly generalize the notion of Laplacian by introduce a notion of conductivity $w(\mathbf{x})$ at each point. The flow of heat at $\mathbf{x}$ takes now the form $-w(\mathbf{x})\nabla f(\mathbf{x})$ and the resulting Laplacian is:

$$\Delta_w(f) = -\operatorname{div}(w \cdot \nabla f).$$

2.2 Graph Discretization

From this perspective, spectral graph theory naturally arises when discretizing the continuous space into a graph with edges connecting neighboring centers, with the caveat that typical discretizations of physical phenomena involve grids or meshes that are good representation of physical space, while general spectral algorithms often involve graphs, such as the Web graphs, that do not have natural low-dimensional representations. We now derive a few more graph matrices based on discretizing the continuous differential operators in the discussion above. In particular, we will be interested in discrete analogue of the gradient operator ∇ and the divergence operator div.

Gradient Operator Given a vector field $\mathbf{x} \in \mathbb{R}^V$ we would like the equivalent to the gradient operator. This operator should return flows across each edge, so its result should belong in $\mathbb{R}^E$. At this point, it is necessary to associate with each edge $e = \{i, j\} \in E$ an arbitrary orientation, indicating the positive direction of flows over e. In this way, for any flow vector $f \in \mathbb{R}^E$, $f_e > 0$ indicates a net flow from i to j, while $f_e > 0$ indicates a net flow from j to i. Whenever the orientation of an edge $e = \{i, j\}$ matters, we will refer to e as (i, j) to emphasize this dependence. With this convention established, the (negative) gradient operator is represented by the following incidence matrix.

Definition 5. The **incidence matrix** $\mathbf{B}_G \in \mathbb{R}^{E \times V}$ is the unique matrix such that for all edges $e = (u, v) \in E$ and $\mathbf{x} \in \mathbb{R}^V$:

$$(\mathbf{B}_G\mathbf{x})_{(v,u)} = x_u - x_v$$

From this equation above we can see that $(\mathbf{B}_G)_{(v,u)} = \mathbf{e}_u - \mathbf{e}_v$, where $\mathbf{e}_i$ is the ith normal basis vector. Equivalently:

$$(\mathbf{B}_G)_{(u,v),i} = \begin{cases} 1 & \text{if } \ i = v \\ -1 & \text{if } \ i = u \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Divergence Operator In order to complete the discretization of the Laplacian operator, we need to discretize the negative divergence operator -div. Let $\mathbf{C}$ be the corresponding matrix. As div takes vector fields to functions, the operator corresponding to C should take flow vectors in $\mathbb{R}^E$ to vectors over the vertices in $\mathbb{R}^E$. We can than take the matrix C to be in $\mathbb{R}^{V \times E}$. Given a flow vector $\mathbf{f} \in \mathbb{R}^E$, the negative divergence at vertex i should measure the total influx of flow into i, i.e.,

$$(\mathbf{Cf})_i = \sum_{(j,i) \in E} \mathbf{f}_{(j,i)} - \sum_{(i,j) \in E} \mathbf{f}_{(i,j)}.$$

Comparing this with Equation 3 readily reveals that C is none other than $\mathbf{B}_G^T$, the transpose of the incidence matrix. Notice that this fact was already prefigured by the alternative notation $\nabla \cdot \mathbf{F}$ for the divergence of a vector field $\mathbf{F}$.

The Laplacian, again With these new graph matrices in tow, we can now show that the Laplacian matrix $\mathbf{L}_G$, introduced in Definition 4, arises as a discretization of the differential operator Δ. We will show this for the generalized case of Δ_w, where the heat conductivity function w will be discretized into the edge weights $\mathbf{w}$. For this purpose, for a weighted graph $G = (V, E, \mathbf{w})$, we introduce a diagonal matrix $\mathbf{W}_G \in \mathbb{R}^{E \times E}$ of weights $\mathbf{w}$. With this notation, the propose discretization of Δ_w becomes the operator with matrix representation $\mathbf{B}_G^T\mathbf{W}_G\mathbf{B}_G$.

Theorem 1. *For a weighted undirected graph $G = (V, E, \mathbf{w})$, the following matrix representations of the Laplacian are equal:*

$$\mathbf{L}_G = \mathbf{D}_G - \mathbf{A}_G = \mathbf{B}_G^T\mathbf{W}_G\mathbf{B}_G.$$

We will prove this theorem in the next section, where we study various basic properties of the graph Laplacian.

3 Basic Properties of the Laplacian

The product form of the Laplacian in Theorem 1 is particularly useful when re-expressed as a sum of symmetric rank-1 matrices as follows:

$$\mathbf{L}_G = \sum_{\{u,v\}\in E} w_{uv}(\mathbf{e}_u - \mathbf{e}_v)(\mathbf{e}_u - \mathbf{e}_v)^T. \tag{4}$$

This expression makes it immediately clear that the orientation of the edges chosen in the definition of $\mathbf{B}_G$ does not change the form of the Laplacian matrix. In particular, for an undirected edge $e = \{u, v\}$, this allows us to identify the edge Laplacian

$$\mathbf{L}_e \overset{\text{def}}{=} (\mathbf{e}_u - \mathbf{e}_v)(\mathbf{e}_u - \mathbf{e}_v)^T = \begin{array}{c c} & \begin{array}{ccccc} & u & & v & \end{array} \\ \begin{array}{c} \\ u \\ \\ v \\ \\ \end{array} & \left(\begin{array}{ccccc} \dots & \vdots & \dots & \vdots & \dots \\ \dots & 1 & \dots & -1 & \dots \\ \dots & \vdots & \dots & \vdots & \dots \\ \dots & -1 & \dots & 1 & \dots \\ \dots & \vdots & \dots & \vdots & \dots \end{array}\right) \end{array}$$

and to conveniently write $\mathbf{L}_G$ as a weighted sum of edge Laplacians:

$$\mathbf{L}_G = \sum_{e\in E} w_e \mathbf{L}_e,$$

More generally, we can prove the following theorem about sums of graphs on the same vertex set.

Theorem 2. *Let the sum $G + H$ of two weighted undirected graphs $G = (V, E_G, \mathbf{w}_G)$ and $H = (V, E_H, \mathbf{w}_H)$ on the same vertex set V as the graph having edge set equal to the union of E_G and E_H and weights equal to the sum of the weights in G and H (where the weight is taken to be 0 if the edge is absent). Then:*

$$\mathbf{L}_{G+H} = \mathbf{L}_G + \mathbf{L}_H.$$

Finally, from Equation 4, we can derive the following important properties of the Laplacian:

1. **Action of $\mathbf{L}_G$**: Consider how $\mathbf{L}_G$ acts on a generic vector $\mathbf{x} \in \mathbb{R}^V$:

   $$(\mathbf{L}_G x)_i = ((\mathbf{D}_G - \mathbf{A}_G)x)_i = d_i x_i - \sum_{j\sim i} w_{ij}x_j = d_i\left(x_i - \frac{\sum_{j\sim i} w_{ij}x_j}{d_i}\right).$$

   This means that $(L_G x)_i$ can be thought of as a measure of how much x_i differs from the weighted average value of x at neighbors of i.

2. **Quadratic Form**: For any $\mathbf{x} \in \mathbb{R}^V$,

   $$\mathbf{x}^T\mathbf{L}_G\mathbf{x} = (\mathbf{B}_G\mathbf{x})^T\mathbf{W}_G(\mathbf{B}_G\mathbf{x}) = \sum_{\{u,v\}\in E} w_{uv}\cdot \mathbf{x}^T(\mathbf{e}_u - \mathbf{e}_v)(\mathbf{e}_u - \mathbf{e}_v)^T\mathbf{x} = \sum_{(v,u)\in E} w_{vu}(x_u - x_v)^2.$$

3. **Diagonal Entries**: By plugging in $\mathbf{x} = \mathbf{e}_i$ into the quadratic form we can retrieve the i-th diagonal element.
$$(\mathbf{L}_G)_{ii} = \mathbf{e}_i^T \mathbf{L}_G \mathbf{e}_i = \sum_{j \sim i} w_{ij} = d_i$$

4. **Off-diagonal Entries**: By computing the bilinear form with respect to $\mathbf{e}_i$ and $\mathbf{e}_j$, we can recover the off-diagonal elements of $\mathbf{L}_G$:
$$(\mathbf{L}_G)_{ij} = \mathbf{e}_i^T \mathbf{L}_G \mathbf{e}_j = \sum_{(v,u) \in E} w_{vu}(-\delta_{iu})\delta_{vj} = -w_{ij},$$
where δ indicates the Kronecker delta.

The last two statements complete the proof of Theorem 1.

4 Connection between Heat Equation and Natural Random Walk on G

In the next lecture, we will study the solutions of the heat equation on a graph. We will show that the correct discretization of the heat equation for this discrete case is the following:

$$\frac{d\mathbf{x}(t)}{dt} = -\mathbf{D}_G^{-1}\mathbf{L}_G\mathbf{x}(t) = -\left(\mathbf{I} - \mathbf{D}_G^{-1}\mathbf{A}_G\right)\mathbf{x}(t) = -\left(x_i(t) - \frac{\sum_{j \sim i} w_{ij}x_j(t)}{d_i}\right)$$

Here, we want to quickly show that solutions to a discrete-time version of the heat equation fully capture the natural random walk process described in Section 1.1. Using a simple forward Euler integration technique with step length $h = 1$, we can rewrite the discretized heat equation as follows:
$$\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)} = -h \cdot \left(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}\right)\mathbf{x}^{(t)} \Rightarrow \mathbf{x}^{(t+1)} = \mathbf{D}_G^{-1}\mathbf{A}_G\mathbf{x}^{(t)}. \tag{5}$$
Equivalently, in this discrete-time evolution, at time $t+1$ each vertex i takes up a value equal to the average of its neighbors' values at time t, i.e.:

$$x_i^{(t+1)} = (\mathbf{D}_G^{-1}\mathbf{A}_G\mathbf{x}^{(t)})_i = \frac{\sum_{i \sim j} w_{ij}x_j^{(t)}}{d_i}.$$

The process $\{\mathbf{x}^{(t)}\}_{t \in \mathbb{N}}$ should be compared with the evolution of the probability vectors $\mathbf{p}^(t)$ of the natural random walk in Section 1.1, whose update rule is:

$$\mathbf{p}^{(t+1)} = \mathbf{A}_G\mathbf{D}_G^{-1}\mathbf{p}^{(t)}. \tag{6}$$

It is easy to see that the updates are of Equations 5 and Equation 6 are equivalent up to the change of representation:
$$\mathbf{p}^{(t)} = \mathbf{D}_G\mathbf{x}^{(t)}.$$

This equivalence can be interpreted in a natural way by considering the amount of probability mass $\mathbf{p}^{(t)}$ at a vertex i as being uniformly distributed over the edges leaving i. In this way, the amount of mass on each edge going out from i is exactly $x_i^{(t)}$. This interpretation allows us to visualize the action of $\mathbf{A}_G$ as sending the mass $\mathbf{x}^{(t)}$ over the edges and aggregating at each vertex, while the action of $\mathbf{D}_G^{-1}$ distributes this mass over the edges again. If we are keeping track of the mass over the edges, the right sequence of operations is left-multiplication by $\mathbf{D}_G^{-1}\mathbf{A}_G$. Conversely, if we are keeping track of the mass over the vertices $\mathbf{p}^{(t)}$, the right sequence of operations is left multiplication by $\mathbf{A}_G\mathbf{D}_G^{-1}$.

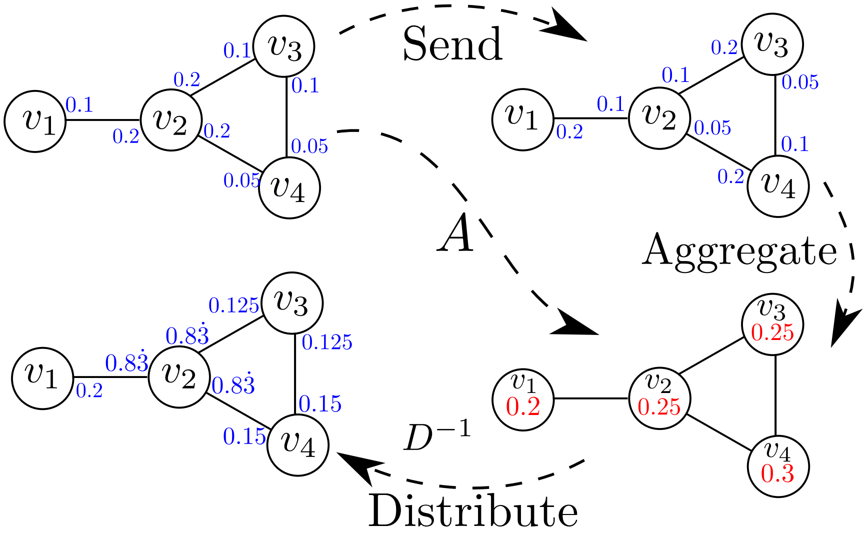

Figure 1: The action of $\mathbf{W}_G^T$ on the vector $\mathbf{x} = (0.1, 0.6, 0.2, 0.1)$.

Hilbert Space Interpretation One can think of the $\mathbf{x}$ vectors as living in the Hilbert space with inner product $\langle\cdot,\cdot\rangle_{\mathbf{D}} : \mathbb{R}^V \to \mathbb{R}$ given by:

$$\langle \mathbf{x}, \mathbf{y} \rangle_D = \mathbf{x}^T\mathbf{D}\mathbf{y}$$

And similarly the $\mathbf{p}$ vectors live in the dual space with inner product: $\langle\cdot,\cdot\rangle_{\mathbf{D}^{-1}} : \mathbb{R}^V \to \mathbb{R}$ given by:

$$\langle \mathbf{p}, \mathbf{q} \rangle_{\mathbf{D}^{-1}} = \mathbf{p}^T\mathbf{D}^{-1}\mathbf{q}$$

for any $\mathbf{x}$ the corresponding vector $\mathbf{p}$ will equal $\Phi(\mathbf{x})$ where Φ is the isomorphism guaranteed by the Riesz representation theorem[1].

5 Notes

Graph Storage How are graphs and matrices stored in computer memory? Storing the full adjacency matrix requires $O(n^2)$ memory while the graph might only have very few edges. At the same time, queries and operations on the edges can be performed very fast, mostly in O(1) time. When the graph is sparse – i.e. $|E| = O(m)$, an adjacency list representation may be preferred, even though some operations may be slightly slower. For a discussion, see:
http://en.wikipedia.org/wiki/Graph_%28abstract_data_type%29#Representations

[1] Read more about it: https://en.wikipedia.org/wiki/Riesz_representation_theorem

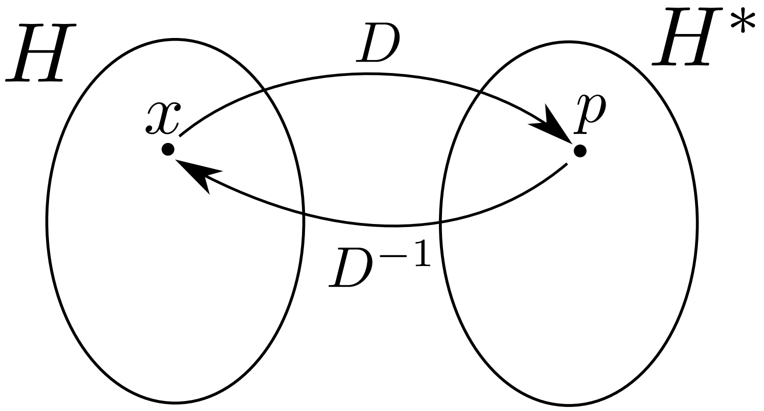

Figure 2: Vertex probability vectors p live in the dual Hilbert space to that of edge mass vectors x.

Similarly, for matrices, different formats can be used according to whether the matrix is dense or sparse. For instance, you can have a look here:
`http://netlib.org/linalg/html_templates/node90.html`
The compressed column storage used by MATLAB can be thought of as a particular implementation of the adjacency list idea.

References

[Chu] Fan R. K. Chung. *Spectral Graph Theory.* American Mathematical Soc.

[Eva22] Lawrence C. Evans. *Partial Differential Equations.* American Mathematical Society, March 2022.

[Ros97] Steven Rosenberg. *The Laplacian on a Riemannian Manifold: An Introduction to Analysis on Manifolds.* London Mathematical Society Student Texts. Cambridge University Press, Cambridge, 1997.

[Spi07] Daniel A. Spielman. Spectral Graph Theory and its Applications. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 29–38, October 2007.

5.1 Exercises

Exercise 1. *Let G be an undirected graph with unit edge weights. A* walk *of length $t \in Z_+$ between vertices u and v in G is a sequence of vertices $(u = v_0, v_1, \ldots, v_t = v)$ such that $\{v_i, v_{i+1}\} \in E$ for all $i \in \{0, \ldots, t-1\}$. Show that, for all $t \in \mathbb{Z}_+$:*

$$(A^t)_{i,j} = \textit{Number of walks of length } t \textit{ between vertices } i \textit{ and } j$$

Exercise 2. *Let G be an undirected graph and L its Laplacian matrix. Show that the vector $\mathbf{1} := (1, ..., 1) \in \mathbb{R}^n$ is an eigenvector of L. What is the corresponding eigenvalue?*