CMSE 890-001: Spectral Graph Theory and Related Topics, MSU, Spring 2021

Lecture 01: Introduction to Spectral Graph Theory

January 19, 2021

Lecturer: Matthew Hirn

1 Introduction

The main book for this course is the draft of the book, *Spectral and Algebraic Graph Theory* by Daniel Spielman [1]; you can download it here. As of this writing, I am using the version dated December 4, 2019. I will make every effort to keep my notation consistent with the book's notation.

As the course progresses we will incorporate other readings, particularly for topics on graph signal processing and graph convolutional (neural) networks. Stay tuned for those.

2 Graphs

An (unweighted, undirected) *graph* $G = (V, E)$ is a collection of vertices V and edges E. The edge set consists of unordered pairs of distinct vertices:

$$E \subset \{(a, b) : a, b \in V \text{ and } a \neq b \text{ and } (a, b) = (b, a)\}.$$

A better notation for an edge is probably $\{a, b\}$ (to emphasize that the order does not matter), but it is tradition to write (a, b).

Many times we will want to add weights to the edges; these are called *weighted graphs* and are written $G = (V, E, w)$. Here $w : E \to \mathbb{R}$ gives the weight $w(a, b) = w(b, a)$ of each edge. Almost always the weights will be positive, i.e., $w(a, b) > 0$, but there might rare occasions when we want to allow for negative weights. We can (and will) view an unweighted graph as a weighted graph in which all the edge weights are equal to one.

Here is a simple example of a graph:

$$\begin{aligned} V &= \{1, 2, 3, 4, 5, 6\}, \\ E &= \{(1, 2), (1, 3), (2, 3), (3, 4), (4, 5), (4, 6)\}. \end{aligned} \tag{1}$$

Graphs have natural visual representations. We can draw dots for the vertices and lines for the edges. Figure 1 provides a drawing of the graph in (1). Here are some other abstract graphs that we will encounter in this course (see Figure 2 for drawings of them):

- The path graph:

$$\begin{aligned} V &= \{1, 2, \ldots, n\}, \\ E &= \{(1, 2), (2, 3), \ldots, (n-1, n)\}. \end{aligned}$$

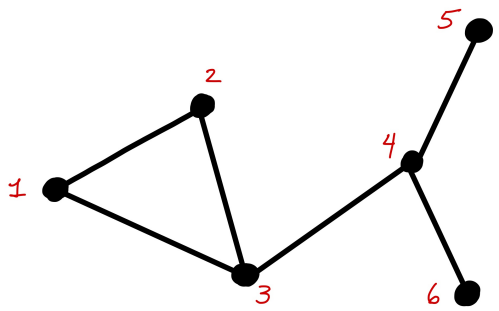

Figure 1: A drawing of the graph from (1).

- The cycle graph:

$$V = \{1, 2, \ldots, n\}\,,$$
$$E = \{(1,2), (2,3), \ldots, (n-1,n), (1,n)\}\,.$$

- The star graph:

$$V = \{1, 2, \ldots, n\}\,,$$
$$E = \{(1,2), (1,3), \ldots, (1,n)\}\,.$$

We will use these abstract graphs to gain theoretical insights, but in terms of applications, the most interesting graphs come from real world data. In this case the vertices represents objects or things, and the edges indicate some form of relationship or similarity between two objects. Here are some examples:

- Friendship graphs (like Facebook): People are vertices, edges exist between pairs of people who are friends (see Figure 3).
- Airplane route graphs: Cities are vertices, and edges exist between pairs of cities for which there is a direct flight (see Figure 4).
- Molecular graphs: Atoms are vertices and edges exists between pairs of atoms that are bonded. The graph represents a single molecule (see Figure 5).
- Gene-gene interaction graphs: Genes are vertices and two genes are connected by an edge if there is physical relationship (e.g., one gene can bind to another) or a functional relationship (see Figure 6).

In fact, almost any data set can be represented as a weighted graph. Indeed, suppose one has data set

$$\text{data set} = \{a_1, \ldots, a_n\}\,,$$

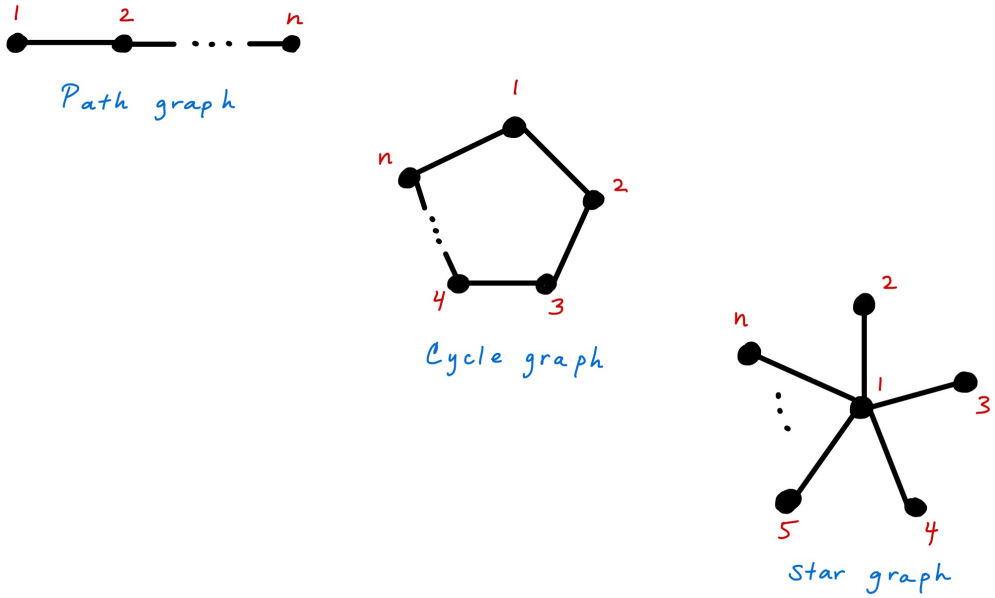

Figure 2: Drawings of the path graph, the cycle graph, and the star graph.

and you are able to provide some notion of similarity between data points through a symmetric kernel function $k(a_i, a_j) = k(a_j, a_i)$ in which $k(a_i, a_j) = 0$ means either a_i and a_j are completely dissimilar or it is impossible to measure to their similarity and $k(a_i, a_j) > 0$ means a_i and a_j have some similarity, with larger values implying a greater degree of similarity. Then we can create a weighted graph $G = (V, E, w)$ in which

$$\begin{aligned} V &= \{a_1, \ldots, a_n\}, \\ E &= \{(a_i, a_j) : k(a_i, a_j) > 0 \text{ and } i \neq j\}, \\ w(a_i, a_j) &= k(a_i, a_j). \end{aligned}$$

This is often a very useful way of thinking about a data set, and the techniques we develop in this course will help you analyze such data.

Remark 1. There are other types of graphs one can consider. These include:

- Pseudograph: Graphs with loops, i.e., we allow $(a, a) \in E$ for $a \in V$.
- Directed graphs, i.e., $(a, b) \neq (b, a)$. In other words, the edges have a directionality now.
- Mixed graphs: These are graphs with some undirected edges and some directed edges.

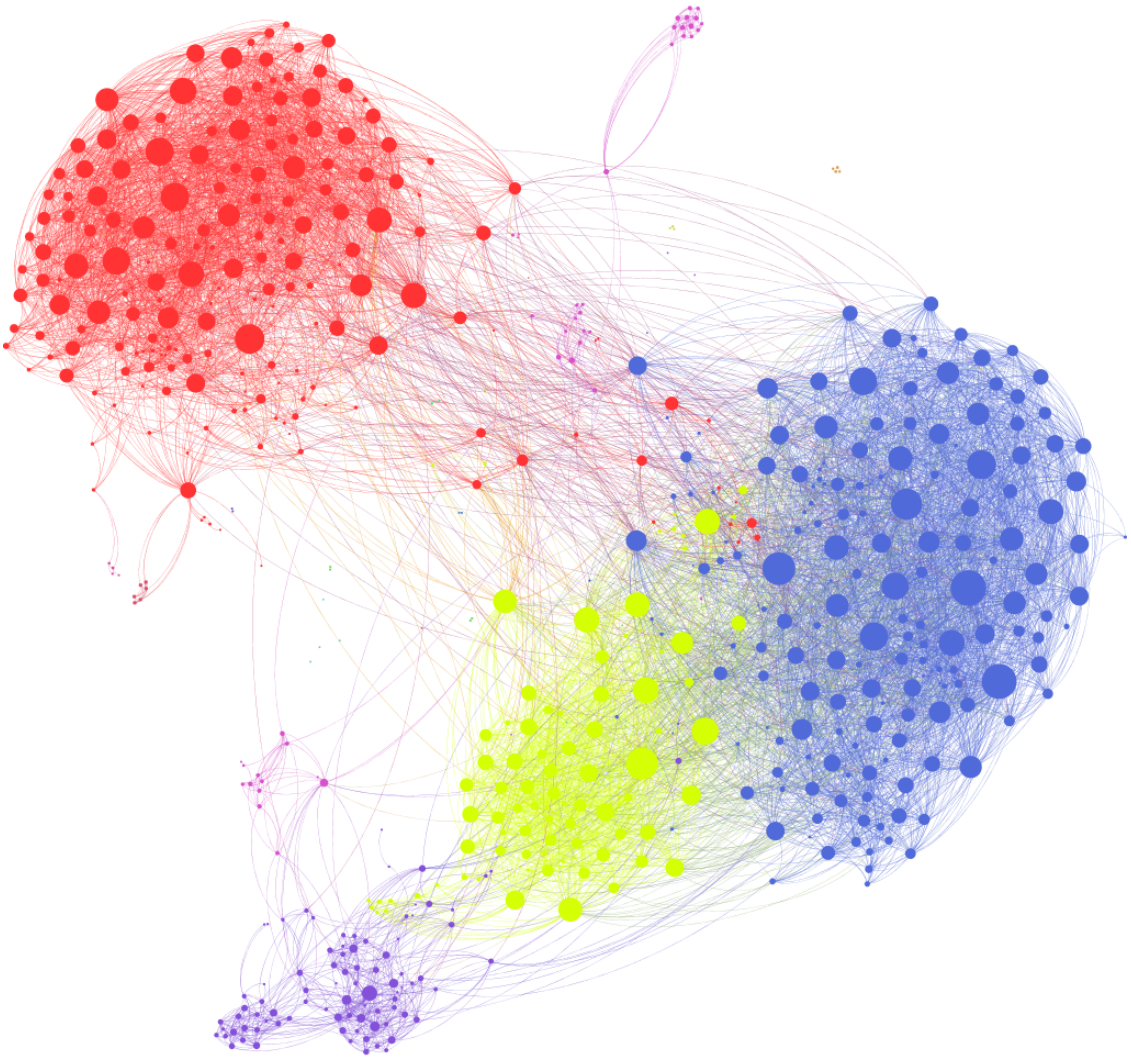

Figure 3: Facebook friendship graph of a particular person from https://griffsgraphs.wordpress.com/tag/social-network/. Per the description on the website, red are high school friends, blue are college friends, yellow are his girlfriend's friends, purple are academic colleagues, and pink are friends met from traveling.

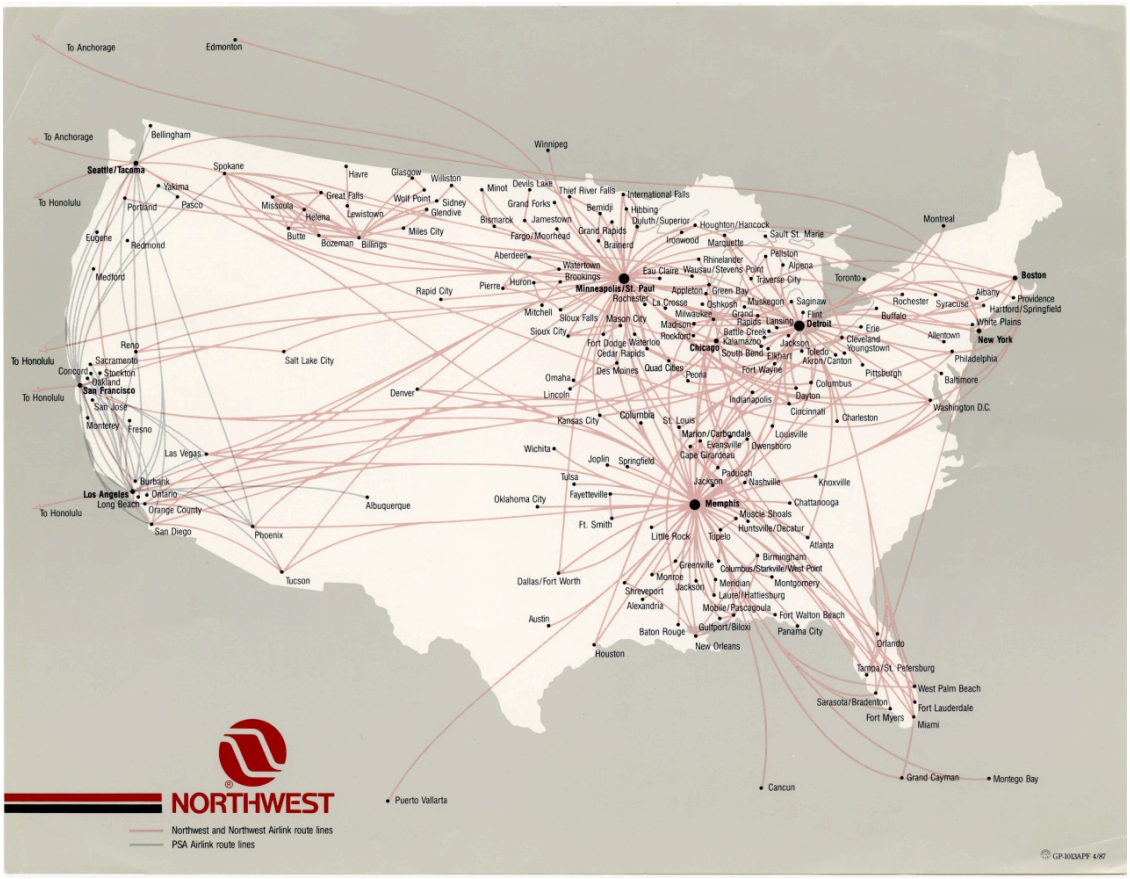

Figure 4: Northwest Airlines route map from 1987.

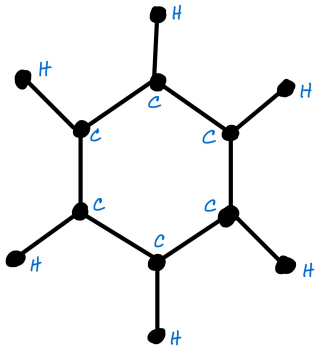

Figure 5: The Benzene molecule represented as a graph. The vertices marked with C represent carbon atoms, and the vertices marked with H represent hydrogen atoms; edges represent bonds.

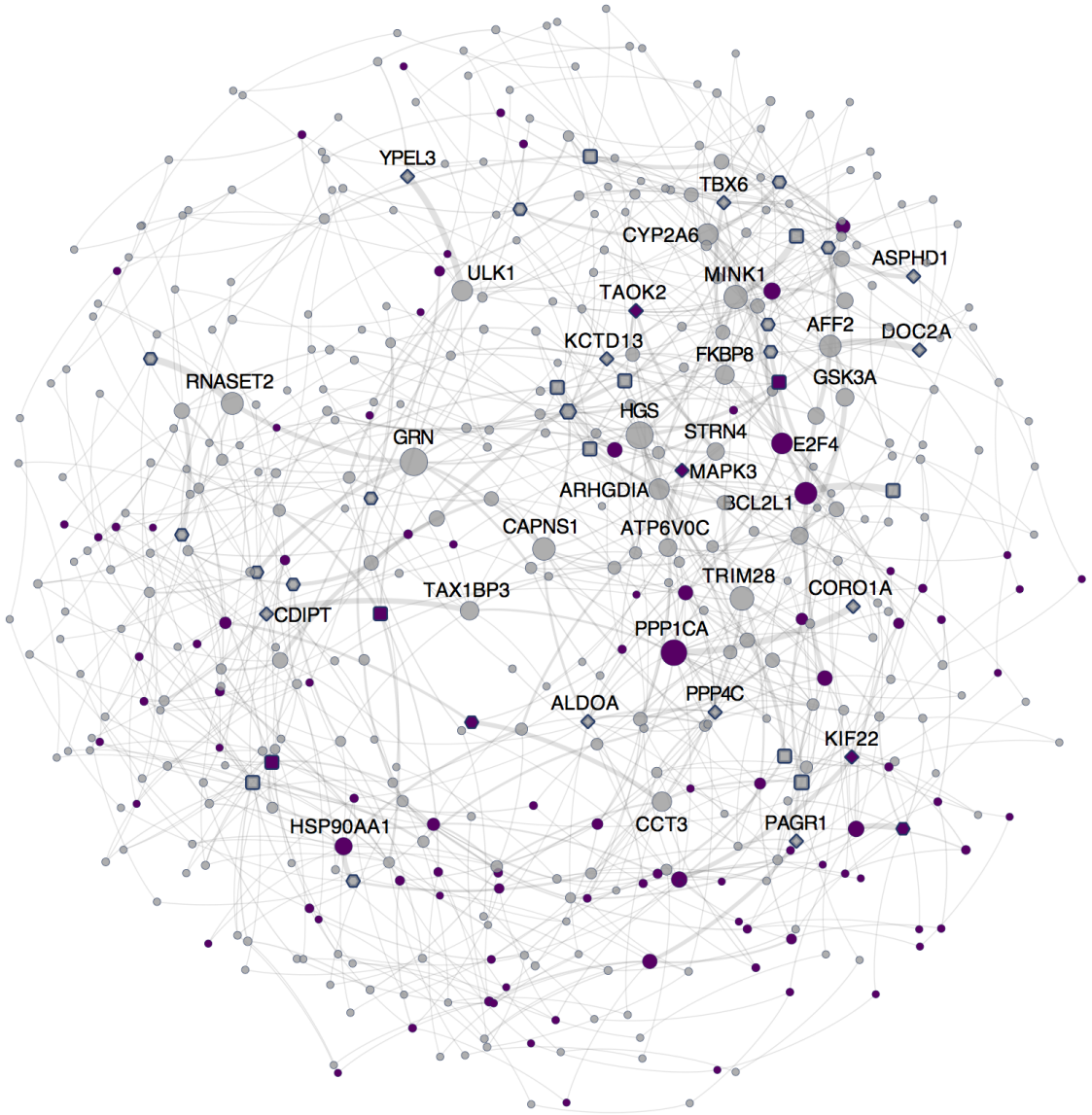

Figure 6: Gene-gene interaction graph; provided by Prof. Arjun Krishnan of CMSE!

- Multigraphs: We allow for multiple distinct edges between the same pair of vertices.
- Hypergraphs: An edge can join more than two vertices.
- Simplicial complexes: These are structures that have vertices, edges, triangles, and their higher-dimensional counterparts.

For the most part in this course we will focus on undirected, (weighted) graphs with no loops. However, we might have an opportunity to look at some of the other types of graphs listed above, depending on time and interests.

Remark 2. From here on out, if we do not say so, we will assume $G = (V, E)$ is a graph with n vertices. If we need to label the vertices, often we will use $V = \{1, \dots, n\}$ (as above).

3 Matrices for graphs

Remark 3. We will denote functions or signals on the vertices of a graph G as $\boldsymbol{x} : V \to \mathbb{R}$. It will be very useful to think of these as functions, and so we will use $\boldsymbol{x}(a)$ to denote the value of $\boldsymbol{x}$ at the vertex $a \in V$. However, it will also often be useful to think of $\boldsymbol{x}$ as an $n \times 1$ column vector, in particular, so we can do linear algebra things (like matrix multiplication).

Remark 4. We will denote $n \times n$ matrices associated to G by bold uppercase letters, such as $\boldsymbol{M} = \boldsymbol{M}_G$. Entries of these matrices will be denoted by $\boldsymbol{M}(a, b)$, to emphasize that they depend on the vertices themselves, not the order in which we may write down the vertices.

We will associate different matrices $\boldsymbol{M}$ to a graph G. Throughout much of the course, there will be three possible ways to think about such matrices:

1. As a spreadsheet that encodes the graph
2. As an operator that maps a function/vector $\boldsymbol{x}$ on the vertices to a new function/vector $\boldsymbol{M}\boldsymbol{x}$.
3. As a quadratic form that maps a function/vector $\boldsymbol{x}$ on the vertices to the number $\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}$.

3.1 Spreadsheet: The adjacency matrix

The most obvious matrix to associate to a graph $G = (V, E)$ is its *adjacency matrix*, which is defined as

$$\boldsymbol{M}_G(a, b) := \begin{cases} 1 & (a, b) \in E\,, \\ 0 & (a, b) \notin E\,. \end{cases}$$

If the graph is weighted, that is $G = (V, E, w)$, then instead we use the weighted adjacency matrix:

$$\boldsymbol{M}_G(a, b) := \begin{cases} w(a, b) & (a, b) \in E\,, \\ 0 & (a, b) \notin E\,. \end{cases}$$

While the adjacency provides a nice way to encode a graph G, it is not terribly useful since it does not provide a natural operator or a quadratic form.

3.2 Operator: Random walks

One of the most natural operators associated to a graph is the *random walk operator*. The idea is the following. Suppose you start at some vertex $a \in V$. You are allowed to step to any other vertex $b \in V$ so long as $(a, b) \in E$; incidentally this set of vertices is called the neighborhood of a:

$$N(a) = N_G(a) := \{b \in V : (a, b) \in E\}.$$

However, you don't get to choose which vertex $b \in N(v)$ you step to, but rather you pick one at random. The probabilities are uniform in an unweighted graph and they are proportional to the weights in a weighted graph. The matrix that encodes the random walk is the random walk matrix.

To define the random walk matrix we need the notion of *degree*. The degree of a vertex in an unweighted graph is its number of neighbors:

$$\deg(a) := |N(a)|.$$

In a weighted graph we weight each of the neighbors according to their degree:

$$\deg(a) := \sum_{b \in N(v)} w(a, b).$$

Let $\boldsymbol{d}$ denote the degree vector, i.e., $\boldsymbol{d}(a) := \deg(a)$, which we notice can be written as

$$\boldsymbol{d} = \boldsymbol{M}\mathbf{1},$$

where $\mathbf{1}$ is the vector of all ones (equivalently, the function that assigned $\mathbf{1}(a) = 1$ to every vertex $a \in V$). The *degree matrix* associated to a graph is the $n \times n$ diagonal matrix with $\boldsymbol{d}$ on its diagonal:

$$\boldsymbol{D}(a, b) = \boldsymbol{D}_G(a, b) := \begin{cases} \boldsymbol{d}(a) & a = b, \\ 0 & a \neq b. \end{cases}$$

The random walk matrix is defined as:

$$\boldsymbol{W} = \boldsymbol{W}_G := \boldsymbol{M}_G \boldsymbol{D}_G^{-1}.$$

Let $\boldsymbol{\delta}_a : V \to \mathbb{R}$ denote the function that assigns the value of one a and the value of zero to every other vertex in V, i.e.,

$$\boldsymbol{\delta}_a(b) := \begin{cases} 1 & b = a, \\ 0 & b \neq a. \end{cases}$$

One can think of $\boldsymbol{\delta}_a$ as a probability distribution on the vertices of G that indicates where we are going to start our random walk. It says that we are starting at the vertex a and

there is no chance we are starting anywhere else. Now let us suppose we take one step in our random walk. We want to know the probability of landing at each vertex in the graph. It will be given by:

$$\boldsymbol{W}\boldsymbol{\delta}_a\,.$$

You can verify for yourself that $\boldsymbol{W}\boldsymbol{\delta}_a(b)$ will only take nonzero values (that is, have non-zero probabilities) when $b \in N(a)$ and its entries will add up to one. If we want to know the probabilities of landing at each vertex in the graph after t steps of the random walk, it will be given by

$$\boldsymbol{W}^t\boldsymbol{\delta}_a\,.$$

Spectral theory will play an important role here as it is a very useful tool by which to analyze repeated applications of an operator, i.e., powers $\boldsymbol{W}^t$.

References

[1] Daniel A. Spielman. Spectral and algebraic graph theory. Book draft, available at: http://cs-www.cs.yale.edu/homes/spielman/sagt/, 2019.

[2] Michael Perlmutter, Feng Gao, Guy Wolf, and Matthew Hirn. Geometric scattering networks on compact Riemannian manifolds. In *Proceedings of The First Mathematical and Scientific Machine Learning Conference, Proceedings of Machine Learning Research*, volume 107, pages 570–604, 2020.