Statistical Data Analysis

Dr. Jana de Wiljes

5. Januar 2022

Universität Potsdam

Using the Triangle Inequality to Accelerate k-Means

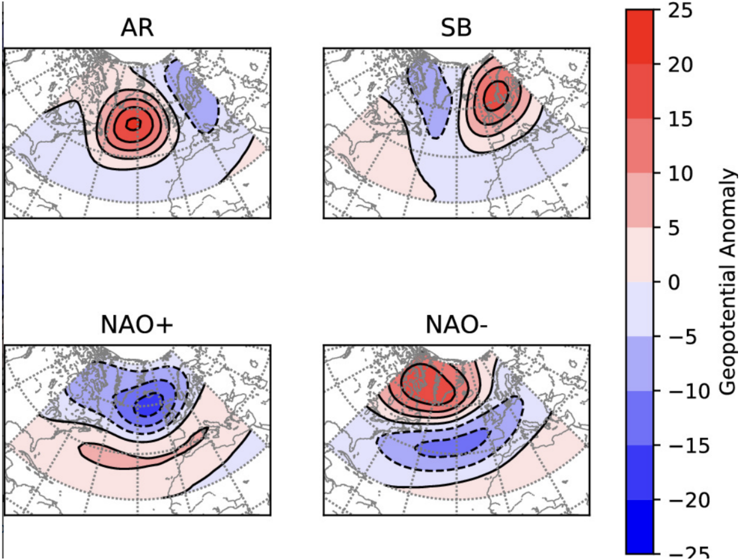
Algorithm:

1. Initialize the centre of the cluster $\theta_1, \ldots, \theta_K \in \mathbb{R}^n$ randomly
2. Set lower bounds to $l(x_m, \theta_i) = 0$ for all θ_i and x_m
3. Assign each x_m to its closest initial center $\theta(x_m) = \arg\min_h ||\theta_h - x_m||_2^2$ (avoid redundant calculations using Lemma 1)
4. Each time $||\theta_h - x_m||_2^2$ is computed, set $l(x_m, \theta_h) = ||\theta_h - x_m||_2^2$
5. Assign upper bounds $u(x_m) = min_i||\theta_i - x_m||_2^2$
6. Repeat till a stopping criterion is fulfilled {
   - 6.1 **for all** θ_i and θ_j, compute $||\theta_i - \theta_j||_2^2$. **For all** centers θ_i, compute $s(\theta_i) = \frac{1}{2}\min_j ||\theta_i - \theta_j||_2^2$
   - 6.2 Identify all points x_m such that $u(x_m) \leq s(\theta(x_m))$.
   - 6.3 **for all** centers θ_i **for all** remaining points x_m check
     - $\theta_i \neq \theta(x_m)$ and
     - $u(x_m) > l(x_m, \theta_i)$ and
     - $u(x_m) > \frac{1}{2}||\theta(x_m) - \theta_i||_2^2$

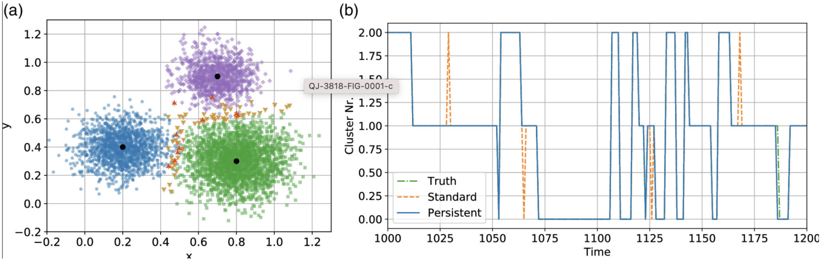
     If conditions $r(x_m) =$ true are true compute $\|x_m - \theta(x_m)\|$ and assign $r(x_m) =$ false. Otherwise $\|x_m - \theta(x_m)\|_2^2 = u(x_m)$.
   - 6.4 if $\|x_m - \theta(x_m)\|_2^2 > l(x_m, \theta_i)$ or $\|x_m - \theta(x_m)\} > \frac{1}{2}||\theta(x_m) - \theta_i||_2^2$ then
     - compute $\|(x_m - \theta_i)\|_2^2$
     - if $\|(x_m - \theta_i)\|_2^2 < \|(x_m - \theta(x_m))\|_2^2$ then assign $\theta(x_m) = \theta_i$
7. **for all** centers θ_i, let $m(\theta_i)$ be the mean of the points assigned to θ_i
8. **for all** points x_m and **for all** centers θ_i assign $l(x_m, \theta_i) = \max\{l(x_m, \theta_i) - \|\theta_i - m(\theta_i)\|_2^2, 0\}$
9. **for all** points x_m, assign $u(x_m) = u(x_m) + \|m(\theta(x_m)) - \theta(x_m)\|$ and $r(x_m) =$ true
10. replace each center θ_i with $m(\theta_i)$
11. **return** $\theta_1, \ldots, \theta_K$

Example: pattern recognition for atmospheric circulation regimes

Regime

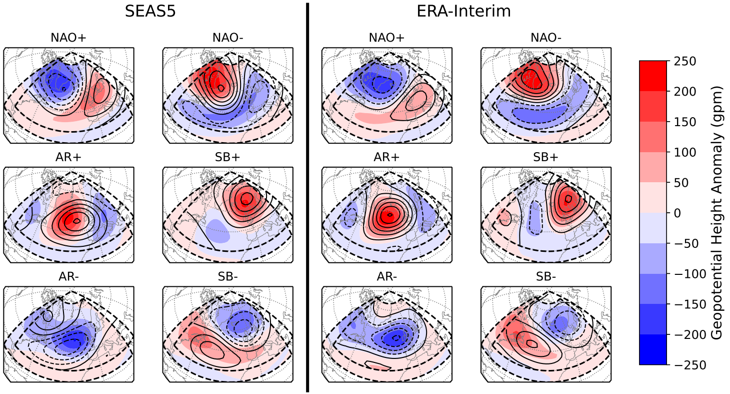


Time persistency constraint

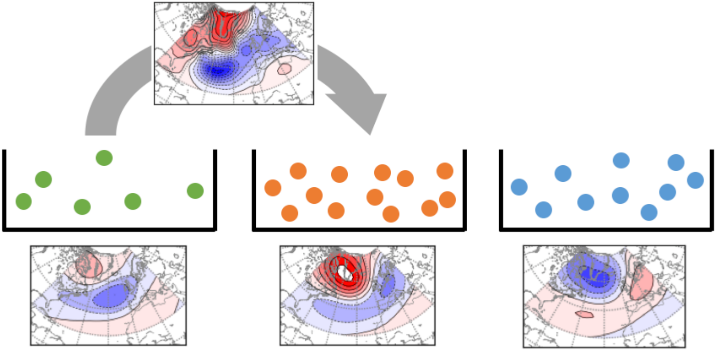


$$\sum_{t=1}^{T-1} |\gamma_k(t+1) - \gamma_k(t)| \leq N_C \quad \forall k$$

k-means clustering for different domains



k-means clustering for different domains



Optimisation problem

$$\mathbf{L}(\Theta, \Gamma) = \sum_{t=0}^{T} \sum_{n=1}^{N} \sum_{i=1}^{k} \gamma_i(t, n) \|x_{t,n} - \theta_i\|^2$$

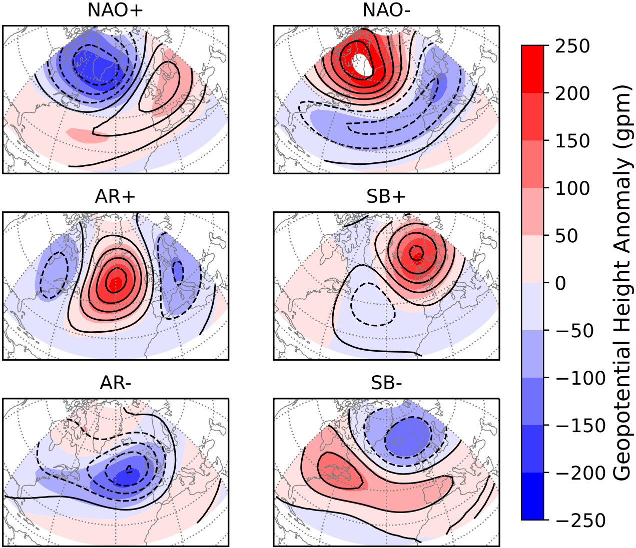
with

$$\sum_{i=1}^{k} \gamma_i(t, n) = 1, \qquad \forall t \in [0, T], \quad \forall n \in [1, N].$$

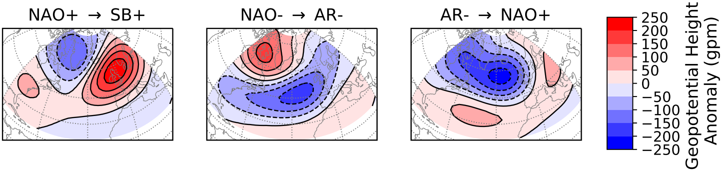
and

$$\sum_{i=1}^{k} \sum_{n_1, n_2} |\gamma_i(t, n_1) - \gamma_i(t, n_2)| \leq \phi \cdot C_{\text{eq}}, \qquad \forall t \in [0, T],$$

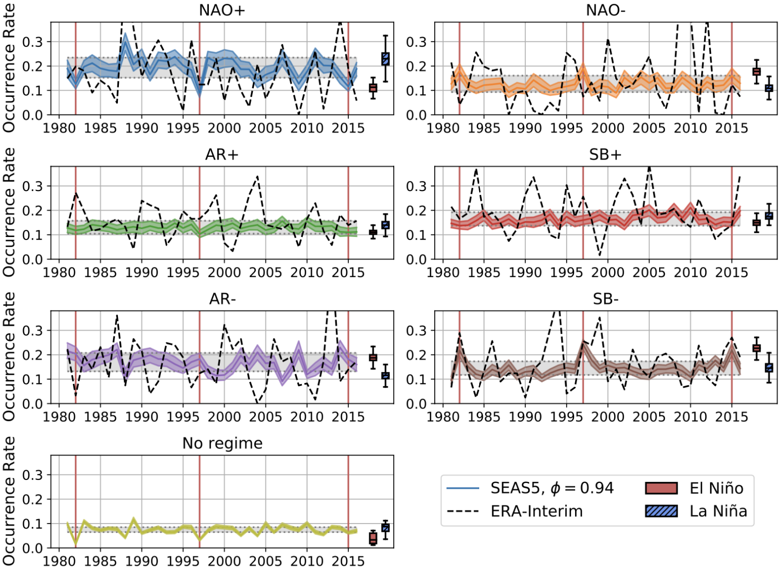
Ensemble persistency constraint



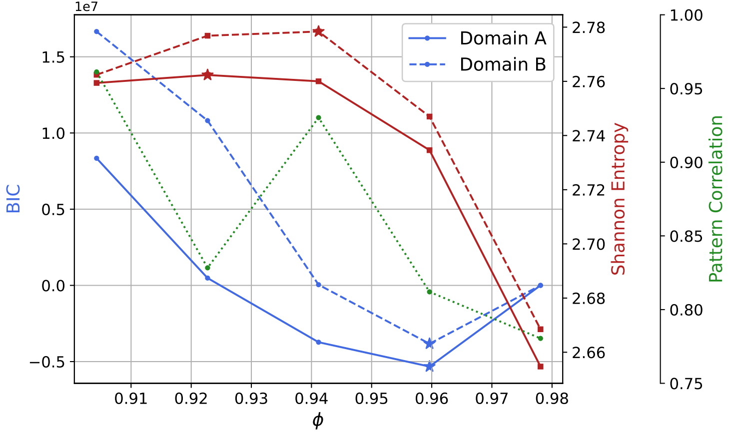
Ensemble persistency constraint



Occurrence rates

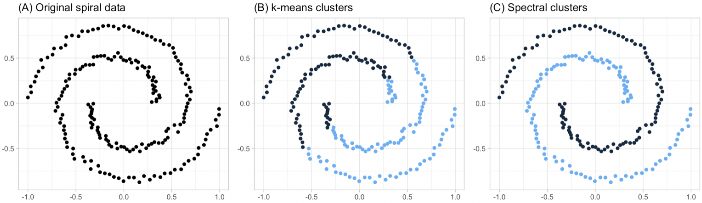


Optimal ϕ



1e7
1.5
1.0
0.5
0.0
−0.5
BIC
0.91
0.92
0.93
0.94
0.95
0.96
0.97
0.98
ϕ
2.78
2.76
2.74
2.72
2.70
2.68
2.66
Shannon Entropy
1.00
0.95
0.90
0.85
0.80
0.75
Pattern Correlation
Domain A
Domain B

K-Means vs Spectral Clustering



Eigenvalues and Eigenvectors

Definition

Let V be a K-Vector space, $f\colon V \to V$ an Endomorphismus, $\lambda \in K$. The scalar λ is called **Eigenvalue** of f, if there is a vector $v \in V, v \neq 0$, so that

$$f(v) = \lambda \cdot v.$$

The vector v is called **Eigenvector** of f an Eigenvalue λ.

Note: An Eigenvalue λ can be $0 \in K$, but an Eigenvector is always $\neq 0$.

Theorem

Let V be a K-vector space, $n = \dim V < \infty$ and $f\colon V \to V$ an Endomorphismus. The following two are equivalent:

1. *V has a basis of Eigenvectors of f.*
2. *There is a Basis $\mathcal{B}$ of V, so that*

$$M_{\mathcal{B}}^{\mathcal{B}}(f) = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} \text{ with } \lambda_i \in K.$$

Characteristic Polynom

Definition

Let $A \in K^{n \times n}$ and $\lambda \in K$ abitrary. Then

$$\mathrm{Eig}(A, \lambda) := \{v \in K^n \mid Av = \lambda v\}$$

is called the **Eigenspace** of A with respect to λ.

$$\chi_A(t) := \det(A - tE) \in K[t]$$

is called the **charakteristisches Polynom** of A.

Remark: For a matrix $A \in K^{n \times n}$ the following holds:

$$\lambda \in K \text{ is an Eigenvalue of } A \Leftrightarrow \mathrm{Eig}(A, \lambda) \neq 0.$$

Theorem

Let $A \in K^{n \times n}$ and $\lambda \in K$. Then:

$$\lambda \text{ is an Eigenvalue of } A \Leftrightarrow \lambda \text{ is a root of } \chi_A(t).$$

Multiplicity

Definition

Let $P(t) \in K[t]$ be a Polynom. $P(t)$ can be decomposed over K in **Linear factors** if and only if there are $\lambda_1, \dots, \lambda_n \in K, c \in K$, so that

$$P(t) = c \cdot (t - \lambda_1) \cdots (t - \lambda_n) = c \cdot \prod_{j=1}^{r} (t - \lambda'_j)^{m_j},$$

where $m_j \in \mathbb{N}$ and $\lambda'_1, \dots, \lambda'_r \in \{\lambda_1, \dots, \lambda_n\}$ are pairwise different. m_j is called the **Multiplicity** of the root λ'_j. It holds that

$$\sum_{j=1}^{r} m_j = n.$$

Example

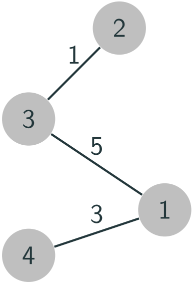
Example

Example

What is a graph (formally)?

The objects on the following slides will play a major role in this course.

- $G = (V, E, \omega)$, where $V \neq \emptyset$ is a set (called the **vertex set**), $E \subset \binom{V}{2} = \{\{u, v\} : u, v \in V\}$ (called the **edge set**) and $\omega : E \to \mathbb{R}^+$, is called a **(weighted) graph**
- usually we choose (or rename) $V = \{1, 2, \dots, n\}$ and use the notations $ij = \{i, j\}$ for $\{i, j\} \in E$ and $\omega_{ij} = \omega(ij)$
- for every $i \in V$ define $N(i) := \{j \in V : ij \in E\}$, called the **neighbourhood** of i (in G); elements of $N(i)$ are called **neighbours** of i (those elements are **adjacent** to i)



$w(23) = 1$,
$N(4) = \{1\}$,
$d(1) = |\{3, 4\}| = 2$

Graph classes

Well known graph classes are:

- the **path graph** P_n has vertex set $\{1, 2, \ldots, n\}$ and edge set $\{\{1, 2\}, \{2, 3\}, \ldots, \{n-1, n\}\}$
- the **cycle graph** C_n has vertex set $\{1, 2, \ldots, n\}$ and edge set $\{\{1, 2\}, \{2, 3\}, \ldots, \{n-1, n\}, \{n, 1\}\}$
- the **complete graph** K_n consists of n vertices which are all adjacent to each other
- the **complete bipartite graph** $K_{m,n}$ has two sets V_1 and V_2 of vertices of sizes m and n, such that the edge set consists of all possible edges between V_1 and V_2

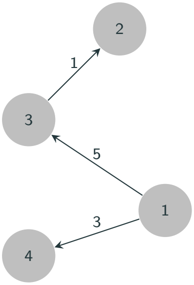
A set of vertices in a graph which are all adjacent to each other (they **induce** a complete (sub)graph), is called **clique**.

The graph $K_{1,n}$ is called a **star**.

What is a digraph (formally)?

Edges can have a direction.

- $G = (V, E, \omega)$, where $V \neq \emptyset$ is a set, $E \subset V \times V$ (this is sometimes also called the **set of arcs**) and $\omega : E \to \mathbb{R}^+$, is called a **(weighted) digraph**
- for $(i, j) \in E$ the vertex i is called **predecessor** of j and j is called **successor** of i
- similar notation simplifications as before
- $N^+(i) := \{j \in V : (i, j) \in E\}$ is the **out-neighbourhood** of i, $N^-(i) := \{j \in V : (j, i) \in E\}$ is the **in-neighbourhood** of i
- $d^+(i) := |N^+(i)|$ is the **out-degree** of i and $d^-(i) := |N^-(i)|$ is the **in-degree** of i



$N^-(3) = \{1\}$,
$N^+(4) = \emptyset$,
$d^+(1) = 2$,
$d^-(2) = 1$

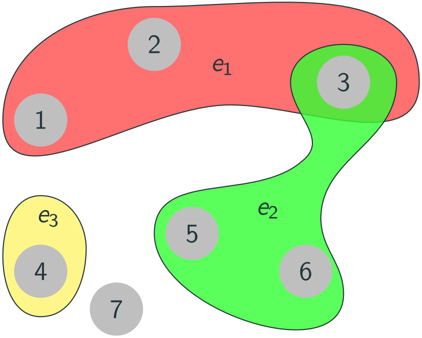
Example of a multigraph

It is sometimes necessary to allow multiple edges between two vertices or a **loop** (a self-edge). In that case we use the term **multigraph**.

What is a hypergraph (formally)?

Sometimes more than two vertices need to form an edge (certain real life situations' have this property).

- natural generalisation is a **hypergraph** $H = (V, E)$, where
  - $V \neq \emptyset$ is (also) a set, but
  - E can be an arbitrary subset (the elements are called **hyperedges**) of the power set $\mathcal{P}(V)$
- if all hyperedges are of the same size r, then H is called r-**uniform**



Storing graphs

Certain matrices and lists can be associated with a graph (we will see more examples later).

- **affinity matrix** $W(G)$:

$$w_{ij} = \begin{cases} \omega_{ij} & \text{if } \{i, j\} \in E, \\ 0 & \text{else.} \end{cases}$$

- **adjacency matrix** $A(G)$: special case of $W(G)$, where $w_{ij} = 1$ for all $ij \in E$.
- **adjacency list**:
    - associate list to every vertex containing its neighbours
    - call list of these lists adjacency list of the graph (treated differently in the literature)
    - not very useful for mathematical arguments
    - especially useful (for storing) when $A(G)$ is sparse

All the above constructions are valid for directed graphs.

How to transform a hypergraph into a graph?

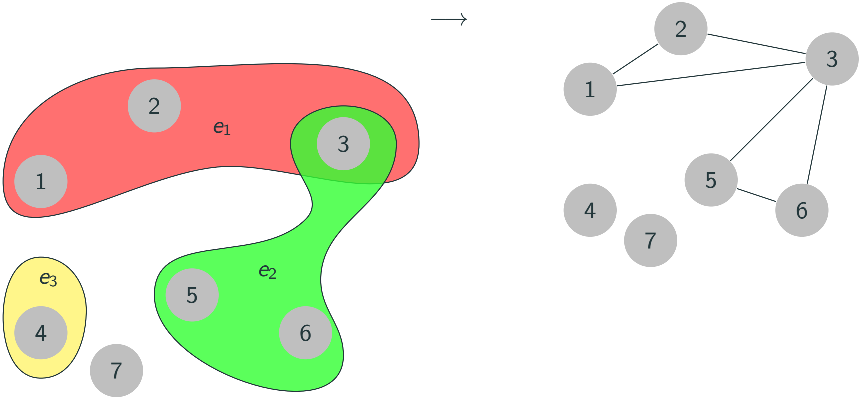
The following constructions are standard.

- clique expansion
  - the vertex set is V
  - each hyperedge e is replaced by an edge for every pair of vertices in e
  - this construction yields cliques for every hyperedge
- star expansion
  - vertex set is $V \cup E$
  - edge between u and e iff $u \in e$
  - every hyperedge corresponds to a star
- there are more...

Clique expansion

The clique expansion $G^x = (V^x, E^x)$ is constructed from $H = (V, E)$ via:

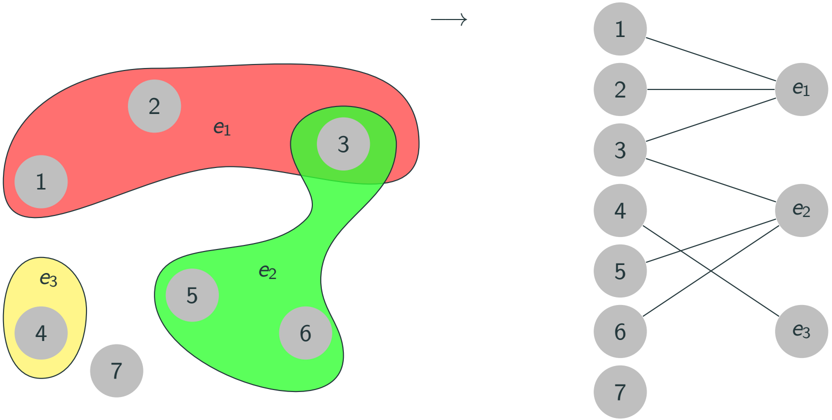
- $V^x = V$
- $E^x = \{\{i, j\} : \exists e \in E \text{ with } i, j \in e\}$



Star expansion

The star expansion $G^* = (V^*, E^*)$ is constructed from $H = (V, E)$ via:

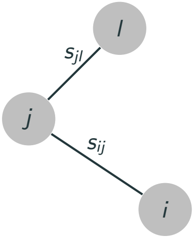
- $V^* = V \cup E$
- $E^* = \{\{i, e\} : i \in e, e \in E\}$



What if data without network structure is given?

Solution: Build your own graph!

- given a set of data points $x_1, x_2 \ldots, x_n$ and some notion of similarity[1] $s_{ij} \geq 0$ between all pairs of data points x_i and x_j
- build graph $G = (V, E)$, where the vertex i represents the data point x_i, so $V = \{1, 2, \ldots, n\}$
- $\{i, j\} \in E$ if $s_{ij} > 0$
- edge weight $\omega_{ij} = s_{ij}$ (edge weights represent similarities)
- G is called **similarity graph** (although with this particular choice of edges it is often referred to as the **fully connected graph**)



graph for $\{x_i, x_j, x_l\}$ with $s_{ij}, s_{jl} > 0$ and $s_{il} = 0$

The ε-neighbourhood graph

The ε-**neighbourhood graph** is constructed as follows:

- vertices are data points
- fix some $\varepsilon > 0$
- connect all vertices whose similarities are smaller than ε
- since ε is usually small, values of existing edges are roughly of the same scale
- hence usually unweighted

The (mutual) k-nearest neighbour graph

The k-**nearest neighbour graph** is constructed as follows:

- vertices are data points
- fix some $k > 0$
- connect i to the k nearest (w.r.t. s_{ij}) k vertices via an edge starting at i
- obtain an undirected graph by ignoring the directions

The **mutual** k-**nearest neighbour graph** is constructed as follows:

- vertices are data points
- fix some k
- connect i to the k nearest (w.r.t. s_{ij}) k vertices via an edge starting at i
- obtain an undirected graph by deleting all non symmetric edges

Spectral clustering

- mathematical foundation by Donath & Hoffman and Fiedler in 1973
- applications in various fields/for various problems
  - image segmentation
  - educational data mining
  - entity resolution
  - speech separation
  - ...

Laplacian matrix (and another graph definition)

The degree matrix $D(G)$ is given by

$$d_{ij} = \begin{cases} \sum_{l \in N(i)} w_{il} & \text{if } i = j, \\ 0 & \text{else.} \end{cases}$$

Laplacian matrix:

$$L(G) = D(G) - W(G)$$

We also need:

$$\mathrm{vol}(A) = \sum_{ij \in E, i, j \in A} \omega_{ij} \text{ for } A \subset V \text{ (no double counting!)}$$