1 Distance

Observation: “∗” is closer to “⋄” than to “◦” and “□” in Euclidean distance.

Clusters: each circle forms one cluster.

What wrong with it?
Use diffusion-based distance and take advantage of the local information:

(a) If placing a heat source at “⋄”, the heat is likely to diffuse faster to “□” than to “∗” and “◦”.

(b) Why? The resistance between “⋄” and “◦” could be higher than that between “∗” and “□”.

Question: Can we derive a distance based on the heat diffusion process on graph?

2 Random walk on the networks

Given a weighted graph $G = (V, E, W)$, one can construct a random walk (a diffusion process) associated to this graph. This graph may be constructed from data points or the dataset itself has a graph structure including social network and biological network.
We consider a random walk \( \{X_t\} \) on the vertices \( \{1, \ldots, n\} \) as follows:

\[
P(X_{t+1} = j | X_t = i) = \frac{w_{ij}}{d_i}, \quad d_i = \sum_{j=1}^{n} w_{ij}, \quad \forall t \geq 0, \quad 1 \leq i, j \leq n \tag{2.1}
\]

where \( t \in \mathbb{Z}_{\geq 0} \) is the time index and \( X_t \) is the state of this random walk at time \( t \). In other words, starting from a vertex, the random walker picks the next state with probability proportional to the value of weight.

**Markov property:** Note that the state \( X_{t+1} \) only depends on \( X_t \) and does not rely on any information before time \( t \). This is called *Markov property*.

**Question:** why (2.1) defines a probability distribution?

\[
\sum_{j=1}^{n} \frac{w_{ij}}{d_i} = \frac{1}{d_i} \sum_{j=1}^{n} w_{ij} = 1, \quad \forall 1 \leq i \leq n.
\]

**Definition 2.1.** The Markov transition matrix \( P \) is defined by

\[
P_{ij} = P(X_{t+1} = j | X_t = i) = \frac{w_{ij}}{d_i}.
\]

If written in matrix form, it holds that

\[
P = D^{-1}W.
\]

### 2.1 Examples

Let’s simulate a random walk on the network.

![Illustration of the network and first 100 states of the Markov chain. The random walker spends a long time before moving to the other complete graph.](image-url)

Figure 1: Illustration of the network and first 100 states of the Markov chain. The random walker spends a long time before moving to the other complete graph.
Consider such a network: two complete graphs is glued with an edge.

\[
A = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{bmatrix}, \quad P = \begin{bmatrix}
1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\
1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\
1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\
1/5 & 1/5 & 1/5 & 1/5 & 1/5 & 0 & 0 & 0 \\
0 & 0 & 0 & 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\
0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\
0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\
0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 \\
\end{bmatrix}
\]

Starting from \(X_0 = 1\), we simulate one trajectory of \(\{X_t\}_{t \geq 0}\) based on the transition probability:

\[
P_{ij} = \mathbb{P}(X_{t+1} = j | X_t = i).
\]

### 2.2 Spectra of transition matrix

**Lemma 2.1.** The leading eigenvalue and eigenvector of \(P\) are 1 and \(1_n\) respectively.

**Proof:** The proof is straightforward:

\[
P1_n = D^{-1}W1_n = D^{-1}d = 1_n.
\]

\(\square\)

**Question:** why is 1 one of leading eigenvalues?

**Question:** when does the transition matrix has a unique leading eigenvalue, i.e., \(\lambda_1(P) = 1\) with multiplicity 1.

The answer follows from our result for Laplacian. Recall that

\[
P = D^{-1}W \iff D^{\frac{1}{2}}PD^{-\frac{1}{2}} = D^{\frac{1}{2}}WD^{-\frac{1}{2}} = I_n - \mathcal{L}.
\]

Let \(\lambda_{\ell,\downarrow}(P)\) be the eigenvalues of \(P\) in descent order; and \(\lambda_{\ell,\uparrow}(\mathcal{L})\) be the eigenvalues of \(\mathcal{L}\) in ascent order. Then

\[
\lambda_{\ell,\downarrow}(P) + \lambda_{\ell,\uparrow}(\mathcal{L}) = 1.
\]

**Lemma 2.2.** The leading eigenvalue of \(P\) is unique if and only if the graph is connected.

### 2.3 Stationary distribution

**Question:** Suppose \(\mathbb{P}(X_i = i) = \pi_i^t\). When does \(X_{t+1}\) share the same distribution with \(X_t\)?

\[
\mathbb{P}(X_{t+1} = i) = \sum_{j=1}^n \mathbb{P}(X_{t+1} = i, X_t = j)
\]

\[
= \sum_{j=1}^n \mathbb{P}(X_{t+1} = i | X_t = j)\mathbb{P}(X_t = j)
\]

\[
= \sum_{j=1}^n \pi_j^t P_{ji} = \pi_i^{t+1}.
\]
If \( \pi^t = \pi^{t+1} \), it holds that
\[
\pi_i = \sum_{j=1}^{n} \pi_j P_{ji}, \quad \forall 1 \leq i \leq n.
\]

**Definition 2.2.** The row vector \( \pi \in \mathbb{R}^{1 \times n} \) is called the stationary distribution if
\[
\pi P = \pi, \quad \sum_{i=1}^{n} \pi_i = 1, \quad \pi \geq 0.
\]
i.e., the leading left eigenvector of \( P \).

**Question:** does it exist? If so, is it unique?

In fact, the stationary distribution for random walk on finite network equals the degree of each node.

**Lemma 2.3.** The stationary distribution is
\[
\pi_i = \frac{d_i}{\text{Vol}(G)}, \quad \text{Vol}(G) = \sum_{i=1}^{n} d_i = \sum_{i,j} w_{ij}
\]
where \( \text{Vol}(G) \) is the volume of the graph \( G \).

We leave the proof to the readers. This Lemma directly implies that the stationary distribution of the example
\[
\pi = \frac{1}{34} [4, 4, 4, 5, 4, 4, 4, 5].
\]

**Theorem 2.4** ("Space average equals time average."). For a random walk with any initialization, it holds that
\[
\lim_{N \to \infty} \frac{\# \{ t : X_t = i, 1 \leq t \leq N \}}{N} = \pi_i
\]
for random walk on a finite connected graph.

We simulate the random walk in Figure 1 to illustrate this theorem. One trajectory is simulated and compute the frequency of each state appearing in the whole sequence. Figure 2 implies that \( N_i(t) \) converges to \( \pi_i \) as \( t \to \infty \).

### 3 Diffusion on the graph

Consider a random walker, starting from \( X_0 = i \), what is the probability of \( \mathbb{P}(X_t = j|X_0 = i) \)? Note that
\[
\mathbb{P}(X_t = j|X_0 = i)
\]
defines a probability mass function associated to the state \( i \). This probability mass function actually characterizes the diffusion process of one particle moving on the graph from a given initial value.

**Question:** How to compute \( \mathbb{P}(X_t = j|X_0 = i) \)?
Theorem 3.1. The probability is given by
\[ P(X_t = j|X_0 = i) = (e_i^\top P^t)_j \]
where \( e_i \) is the \( i \)th vector of the canonical basis in \( \mathbb{R}^n \). This is exactly the \((i,j)\)-entry of \( P^t \).

Proof: Let’s prove by induction. If \( t = 0 \), then
\[ P(X_0 = j|X_0 = i) = \begin{cases} 1, & j = i, \\ 0, & j \neq i. \end{cases} \]
In other words, \( P(X_0|X_0 = i) = e_i^\top \) holds. Now we assume for \( 0, \cdots, t-1 \), the result holds, i.e.,
\[ P(X_s|X_0 = i) = e_i^\top P^s, \quad 0 \leq s \leq t-1. \]

Then
\[
P(X_t = j|X_0 = i) = \sum_{k=1}^{n} P(X_t = j, X_{t-1} = k|X_0 = i)
\]
\[
= \sum_{k=1}^{n} \frac{P(X_t = j, X_{t-1} = k, X_0 = i) P(X_{t-1} = k, X_0 = i)}{P(X_0 = i) P(X_{t-1} = k, X_0 = i)}
\]
\[
= \sum_{k=1}^{n} P(X_t = j|X_{t-1} = k, X_0 = i) P(X_{t-1} = k|X_0 = i)
\]
\[
= \sum_{k=1}^{n} P(X_t = j|X_{t-1} = k) P(X_{t-1} = k|X_0 = i)
\]
\[
= \sum_{k=1}^{n} (e_i^\top P^{t-1})_k P_{kj}
\]
\[
= \sum_{k=1}^{n} (P^{t-1})_{ik} P_{kj} = (P^t)_{ij}
\]
where the fourth equality follows from Markov property.
3.1 Simulation of the diffusion process

Now we pick two points on the concentric circles and compute the diffusion starting from these two initializations by studying the conditional probability:

\[ p_t(i, k) := \mathbb{P}(X_t = k|X_0 = i) = (e_i^\top P^t)_k \]

The colorbar in the following figures reflects the magnitude of \( p_t(i, k) \) at time \( t \). In this case, we construct the weight matrix by using Gaussian kernel function:

\[ w_{ij} = \exp \left( -\frac{\|x_i - x_j\|^2}{2\sigma^2} \right). \]

We choose Gaussian kernel because the actual diffusion is also characterized by Gaussian function.

![Two concentric circles, \( X_0 = 250 \)](image)

Two concentric circles, \( X_0 = 250 \)

![Two concentric circles, \( X_0 = 1000 \)](image)

Two concentric circles, \( X_0 = 1000 \)

Figure 3: Diffusion on graphs from two initializations

![Diffusion at \( t = 1, X_0 = 250 \)](image)

Diffusion at \( t = 1, X_0 = 250 \)

![Diffusion at \( t = 1, X_0 = 1000 \)](image)

Diffusion at \( t = 1, X_0 = 1000 \)

Figure 4: Transition matrix at time \( t = 1 \)
4 Diffusion maps

How to define a “notion” of distance between two nodes on the graph? Diffusion maps \cite{1} provide a method to compute the distance between nodes based on the diffusion process on the network.
The distance between state $i$ and $j$ is derived from the probability measure of this Markov chain at time $t$ starting from $i$ and $j$:

$$p_t(i, k) = \mathbb{P}(X_t = k | X_0 = i) = (e_i^\top P^t)_k$$

and the diffusion distance is obtained by

$$D_t(i, j) = \sqrt{\sum_{k=1}^{n} \frac{1}{\pi_k} |p_t(i, k) - p_t(j, k)|^2}, \quad \pi_k = \frac{d_k}{\text{Vol}(G)}.$$ 

In matrix form, we have

$$D^2_t(i, j) = \frac{1}{\text{Vol}(G)} (e_i - e_j)^\top P D^{-1} P^\top (e_i - e_j).$$

It is the weighted $\ell^2$ distance between the probability measure $\mathbb{P}(X_t | X_0 = i)$ and $\mathbb{P}(X_t | X_0 = j)$.

1. It reflects the connectivity of the data at any given time scale $t$. As $t \to \infty$, the diffusion distance between any two points vanishes.

2. The distance $D_t(i, j)$ basically sums over all paths of length $t$ connecting from $i$ to $j$ as well as from $j$ to $i$. Thus it is more robust than shortest path (geodesic distance)

We present an example here by looking at how $P^t$ evolves w.r.t. time $t$, as shown in Figure 9 and 10.

**Theorem 4.1.** The diffusion distance satisfies

$$D^2_t(i, j) = \sum_{\ell=1}^{n} \lambda^2_t(\psi_\ell(i) - \psi_\ell(j))^2$$

where $\psi_\ell$ is the $\ell$th eigenvector of $P$, i.e.,

$$P \psi_\ell = \lambda_\ell \psi_\ell, \quad 1 \leq \ell \leq n, \quad 1 \geq \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -1$$

and $\psi_\ell(i)$ is the $i$th entry of $\psi_\ell$. 

---

**Figure 8:** Transition matrix at time $t = 1000$
Figure 9: $t$-step transition matrix $P^t$. 
Figure 10: $t$-step transition matrix $P^t$.

**Proof:** Note that

$$D^2_t(i, j) = \text{Vol}(G)(e_i - e_j)^\top PD^{-1}P^\top(e_i - e_j)$$

since

$$e_i^\top P = [p_t(i, 1), \cdots, p_t(i, n)], \quad \pi_k = \frac{d_k}{\text{Vol}(G)}.$$  

Let $\Psi = [\psi_1, \cdots, \psi_n]$ be the matrix consisting of all the eigenvectors of $P$. It holds that

$$P\Psi = \Psi\Lambda \iff P = \Psi\Lambda\Psi^{-1}.$$
\[ \Lambda = \text{diag}(\lambda_1, \cdots, \lambda_n) \].

In fact \( \Psi \) satisfies \( \Psi^\top D \Psi = I_n \). It follows from
\[ P \Psi = \Psi \Lambda \iff D^{-1} W \Psi = \Psi \Lambda \iff D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \Psi = D^{-\frac{1}{2}} \Psi \Lambda \]

In other words, \( D^{\frac{1}{2}} \Psi \) are the eigenvectors of the normalized graph Laplacian. Therefore,
\[ \Psi^\top D \Psi = I_n, \quad \Psi^{-1} D^{-1} \Psi^{-\top} = I_n \]

since the eigenvectors of symmetric matrices are perpendicular to one another. Now we compute \( D^2_t(i, j) \):
\[ PD^{-1} P^\top = \Psi \Lambda \Psi^{-1} D^{-1} \Psi^{-\top} \Lambda \Psi^\top = \Psi \Lambda^2 \Psi^\top \]

Therefore, the distance between \( i \) and \( j \) is given by the kernel \( \Psi \Lambda^2 \Psi^\top \).

Thus we have proven
\[ D^2_t(i, j) = \text{Vol}(G)(e_i - e_j)^\top \Psi \Lambda^2 \Psi^\top (e_i - e_j). \]

What does it mean? One can find an embedding of the data points via diffusion.

**Definition 4.1 (Diffusion maps).** The diffusion maps are defined as
\[ \Psi_t(i) := \begin{bmatrix} \lambda_1^{t} \psi_1(i) \\ \lambda_2^{t} \psi_2(i) \\ \vdots \\ \lambda_n^{t} \psi_n(i) \end{bmatrix}, \quad 1 \leq i \leq n \]

where \( \psi_\ell \) is the \( \ell \)th eigenvector of \( P \). Each component in \( \Psi_t(i) \) is termed diffusion coordinate.

**Remark:** it is important to note that the diffusion maps are time dependent, i.e., depending on the time parameter \( t \). This allows us to have a representation of dataset at different time scale.

By construction of diffusion maps, we have
\[ D^2_t(i, j) = \text{Vol}(G)\|\Psi_t(i) - \Psi_t(j)\|^2 \]

where
\[ \|\Psi_t(i) - \Psi_t(j)\|^2 = \sum_{\ell=2}^{m} \lambda_\ell^{2t}(\psi_\ell(i) - \psi_\ell(j))^2. \]

In other words, we embed each vertex \( i \) to \( \Psi_t(i) \in \mathbb{R}^m \) whose Euclidean distance equals diffusion distance.

Note that in most cases, the eigenvalue \( \lambda_\ell \) decays very fast. Therefore, only a few coordinates are needed in some cases. Therefore, we consider the truncated diffusion maps:
\[ \Psi_{t,m}(i) := \begin{bmatrix} \lambda_1^{t} \psi_1(i) \\ \lambda_2^{t} \psi_2(i) \\ \vdots \\ \lambda_m^{t} \psi_m(i) \end{bmatrix} \]

where \( m(t, \delta) = \max\{ \ell : \lambda_\ell^{t} \geq \delta \lambda_2^{t} \} \) for some small \( \delta \).
4.1 Examples

4.1.1 n-cycle graphs

In this case, let’s compute the diffusion maps.

\[
P = \begin{bmatrix}
0 & 1/2 & 0 & \cdots & 0 & 0 & 1/2 \\
1/2 & 0 & 1/2 & \cdots & 0 & 0 & 0 \\
0 & 1/2 & 0 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 1/2 & 0 \\
0 & 0 & 0 & \cdots & 1/2 & 0 & 1/2 \\
1/2 & 0 & 0 & \cdots & 0 & 0 & 0 \\
\end{bmatrix}
\]

The second and third eigenvectors \( \psi \) of \( P \) are

\[
\psi_2 = \begin{bmatrix}
cos \left( \frac{2\pi}{n} \right), \cdots, cos \left( \frac{2(n-1)\pi}{n} \right), 1
\end{bmatrix}
\]

\[
\psi_3 = \begin{bmatrix}
sin \left( \frac{2\pi}{n} \right), \cdots, sin \left( \frac{2(n-1)\pi}{n} \right), 1
\end{bmatrix}
\]

with \( \lambda_2 = \lambda_3 = \cos \left( \frac{2\pi}{n} \right) \). As a result, the diffusion maps are given by

\[
\Psi_{t,2}(i) = \cos \left( \frac{2\pi}{n} \right) \begin{bmatrix}
cos \left( \frac{2\pi i}{n} \right) \\
sin \left( \frac{2\pi i}{n} \right)
\end{bmatrix}
\]

4.1.2 Swiss roll

Suppose the dataset is in the shape “swiss roll”. We construct the weight matrix and Markov transition matrix from the data. Then use the first two eigenvectors of \( P \) to embed the data points in 2D, shown in Figure 11.

The toy example is generated by

\[
X = \text{sort} \left( \text{rand}(n,1) \right) \ast 14; \\
Y = \text{rand}(n,1) \ast 10; \\
data = [X \ast \cos(X) \ Y \ X \ast \sin(X)];
\]

\[
sigma = 5;
\]

\[
pdist\_data = \text{squareform} \left( \text{pdist(data)} \right); \\
W = \exp \left( -pdist\_data \cdot 2/(2*\sigma^2) \right); \\
D = \text{diag} \left( \text{sum}(W) \right); \\
P = D \backslash W;
\]

\[
[eigP\_vec, eigP\_val] = \text{eig} \left( P \right); \\
eigP\_val = \text{real} \left( \text{diag} \left( eigP\_val \right) \right); \\
[\sim, id] = \text{sort} \left( \text{abs} \left( eigP\_val \right), '\text{descend}' \right); \\
Phi = \text{real} \left( \begin{bmatrix}
eigP\_val(id==1) \cdot t \ast eigP\_vec(:,id==1), \ldots \\
eigP\_val(id==2) \cdot t \ast eigP\_vec(:,id==2), eigP\_val(id==3) \cdot t \ast eigP\_vec(:,id==3)
\end{bmatrix} \right);
\]

For the connection to commute time distance and effective resistance, find more details in [2].
Figure 11: Swiss roll and Embedded data in 2D

References
