MAT 280: Harmonic Analysis on Graphs & Networks
Lecture 10: Distances on Graphs

Naoki Saito

Department of Mathematics
University of California, Davis

October 29, 2019

Outline

1. Need of a Variety of Graph Distances

2. The (Shortest) Path Distance

3. Resistance Distance

4. Commute-Time Distance

Outline

1. Need of a Variety of Graph Distances
2. The (Shortest) Path Distance
3. Resistance Distance
4. Commute-Time Distance

Need of a Variety of Graph Distances

- Suppose a simple, weighted, and connected graph $G(V,E)$ is already built from the dataset $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$.
- Assume the data vector $\boldsymbol{x}_i$ is associated with the vertex v_i.
- Let $w(e) = w_e \geq 0$ be an edge weight of the edge $e \in E(G)$, which we also write a_{ij}, i.e., (i,j)th entry of the weighted adjacent matrix A if $e = (v_i, v_j)$.
- In many applications, we want to know the *similarity* or *dissimilarity* between vertices v_i and v_j. Note that v_i and v_j may not be adjacent (i.e., they may not be endpoints of a single edge).
- Such applications include: search engines; data mining; social network analysis; pattern recognition; image processing, . . .

Need of a Variety of Graph Distances

- Suppose a simple, weighted, and connected graph $G(V,E)$ is already built from the dataset $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$.
- Assume the data vector $\boldsymbol{x}_i$ is associated with the vertex v_i.
- Let $w(e) = w_e \geq 0$ be an edge weight of the edge $e \in E(G)$, which we also write a_{ij}, i.e., (i,j)th entry of the weighted adjacent matrix A if $e = (v_i, v_j)$.
- In many applications, we want to know the *similarity* or *dissimilarity* between vertices v_i and v_j. Note that v_i and v_j may not be adjacent (i.e., they may not be endpoints of a single edge).
- Such applications include: search engines; data mining; social network analysis; pattern recognition; image processing, . . .

Need of a Variety of Graph Distances

- Suppose a simple, weighted, and connected graph $G(V,E)$ is already built from the dataset $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$.
- Assume the data vector $\boldsymbol{x}_i$ is associated with the vertex v_i.
- Let $w(e) = w_e \geq 0$ be an edge weight of the edge $e \in E(G)$, which we also write a_{ij}, i.e., (i,j)th entry of the weighted adjacent matrix A if $e = (v_i, v_j)$.
- In many applications, we want to know the *similarity* or *dissimilarity* between vertices v_i and v_j. Note that v_i and v_j may not be adjacent (i.e., they may not be endpoints of a single edge).
- Such applications include: search engines; data mining; social network analysis; pattern recognition; image processing, ...

Need of a Variety of Graph Distances

- Suppose a simple, weighted, and connected graph $G(V,E)$ is already built from the dataset $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$.
- Assume the data vector $\boldsymbol{x}_i$ is associated with the vertex v_i.
- Let $w(e) = w_e \geq 0$ be an edge weight of the edge $e \in E(G)$, which we also write a_{ij}, i.e., (i,j)th entry of the weighted adjacent matrix A if $e = (v_i, v_j)$.
- In many applications, we want to know the *similarity* or *dissimilarity* between vertices v_i and v_j. Note that v_i and v_j may not be adjacent (i.e., they may not be endpoints of a single edge).
- Such applications include: search engines; data mining; social network analysis; pattern recognition; image processing, ...

Need of a Variety of Graph Distances

- Suppose a simple, weighted, and connected graph $G(V,E)$ is already built from the dataset $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$.
- Assume the data vector $\boldsymbol{x}_i$ is associated with the vertex v_i.
- Let $w(e) = w_e \geq 0$ be an edge weight of the edge $e \in E(G)$, which we also write a_{ij}, i.e., (i,j)th entry of the weighted adjacent matrix A if $e = (v_i, v_j)$.
- In many applications, we want to know the *similarity* or *dissimilarity* between vertices v_i and v_j. Note that v_i and v_j may not be adjacent (i.e., they may not be endpoints of a single edge).
- Such applications include: search engines; data mining; social network analysis; pattern recognition; image processing, . . .

Outline

1. Need of a Variety of Graph Distances

2. The (Shortest) Path Distance

3. Resistance Distance

4. Commute-Time Distance

The (Shortest) Path Distance

- Most common metric on a graph
- Let $u, v \in V(G)$ be any two vertices of G. Then, the *(shortest) path distance* (a.k.a. the *geodesic distance*) between u and v is the sum of the weights along the path connecting u and v such that it becomes minimum, i.e.,

$$d_{\text{path}}(u, v) := \min_{P \in \mathscr{P}(u,v)} \sum_{e \in P} w(e)$$

- If G is directed, then $d_{\text{path}}(u, v) \neq d_{\text{path}}(v, u)$ in general.
- Efficient algorithms to compute $d_{\text{path}}(\cdot, \cdot)$ exist, e.g., the A^* algorithm of Hart-Nilsson-Raphael (1968), etc. Yet, in general, its complexity is at least polynomial time w.r.t. n. $\Longrightarrow$ See the excellent Wikipedia page!

The (Shortest) Path Distance

- Most common metric on a graph
- Let $u, v \in V(G)$ be any two vertices of G. Then, the *(shortest) path distance* (a.k.a. the *geodesic distance*) between u and v is the sum of the weights along the path connecting u and v such that it becomes minimum, i.e.,

$$d_{\text{path}}(u, v) := \min_{P \in \mathscr{P}(u,v)} \sum_{e \in P} w(e)$$

- If G is directed, then $d_{\text{path}}(u, v) \neq d_{\text{path}}(v, u)$ in general.
- Efficient algorithms to compute $d_{\text{path}}(\cdot, \cdot)$ exist, e.g., the A^* algorithm of Hart-Nilsson-Raphael (1968), etc. Yet, in general, its complexity is at least polynomial time w.r.t. n. $\Longrightarrow$ See the excellent Wikipedia page!

The (Shortest) Path Distance

- Most common metric on a graph
- Let $u, v \in V(G)$ be any two vertices of G. Then, the *(shortest) path distance* (a.k.a. the *geodesic distance*) between u and v is the sum of the weights along the path connecting u and v such that it becomes minimum, i.e.,

$$d_{\text{path}}(u,v) := \min_{P \in \mathscr{P}(u,v)} \sum_{e \in P} w(e)$$

- If G is directed, then $d_{\text{path}}(u,v) \neq d_{\text{path}}(v,u)$ in general.
- Efficient algorithms to compute $d_{\text{path}}(\cdot,\cdot)$ exist, e.g., the A^* algorithm of Hart-Nilsson-Raphael (1968), etc. Yet, in general, its complexity is at least polynomial time w.r.t. n. $\Longrightarrow$ See the excellent Wikipedia page!

The (Shortest) Path Distance

- Most common metric on a graph
- Let $u, v \in V(G)$ be any two vertices of G. Then, the *(shortest) path distance* (a.k.a. the *geodesic distance*) between u and v is the sum of the weights along the path connecting u and v such that it becomes minimum, i.e.,

$$d_{\mathrm{path}}(u,v) := \min_{P \in \mathscr{P}(u,v)} \sum_{e \in P} w(e)$$

- If G is directed, then $d_{\mathrm{path}}(u,v) \neq d_{\mathrm{path}}(v,u)$ in general.
- Efficient algorithms to compute $d_{\mathrm{path}}(\cdot,\cdot)$ exist, e.g., the A^* algorithm of Hart-Nilsson-Raphael (1968), etc. Yet, in general, its complexity is at least polynomial time w.r.t. n. $\Longrightarrow$ See the excellent Wikipedia page!

Outline

1. Need of a Variety of Graph Distances

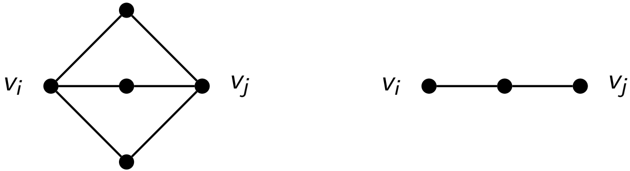
2. The (Shortest) Path Distance

3. Resistance Distance

4. Commute-Time Distance

Resistance Distance

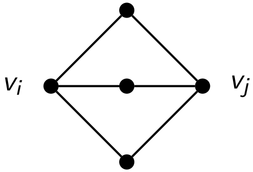
- The shortest path distance may not be always relevant. Consider the following two subgraphs where each $w_e \equiv 1$:

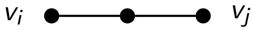


- In both cases in the above, $d_{\mathrm{path}}(v_i, v_j) = 2$. But it is clear that there are more paths connecting v_i and v_j in the subgraph in the left than in the right.
- Hence, it is reasonable that the "distance" between v_i and v_j should be smaller in the left than in the right.

Resistance Distance

- The shortest path distance may not be always relevant. Consider the following two subgraphs where each $w_e \equiv 1$:

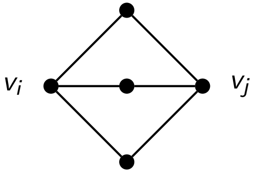


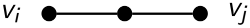


- In both cases in the above, $d_{\text{path}}(v_i, v_j) = 2$. But it is clear that there are more paths connecting v_i and v_j in the subgraph in the left than in the right.
- Hence, it is reasonable that the "distance" between v_i and v_j should be smaller in the left than in the right.

Resistance Distance

- The shortest path distance may not be always relevant. Consider the following two subgraphs where each $w_e \equiv 1$:





- In both cases in the above, $d_{\text{path}}(v_i, v_j) = 2$. But it is clear that there are more paths connecting v_i and v_j in the subgraph in the left than in the right.
- Hence, it is reasonable that the "distance" between v_i and v_j should be smaller in the left than in the right.

Resistance Distance . . .

- Now, let's interpret the edge weights as *resistances* in an *electrical network*.
- For any v_i, v_j, $i \neq j$, suppose that a battery is connected across them so that one unit of a current flows in at v_i and out in v_j.
- Then, the voltage (potential) difference is the effective resistance between v_i and v_j by Ohm's law ($V = I \cdot R$), which is called the *resistance distance* between v_i and v_j and written as $r(v_i, v_j) = r_{ij}$.
- Hence, $1/r_{ij}$ is the *conductance* between v_i and v_j, which is a measure of *connectivity* (or *similarity*) between them, which should be the edge weight a_{ij}.
- Hence, given resistance distances $\{r_{ij}\}$, one can construct the corresponding weighted adjacency matrix A by

$$a_{ij} = \begin{cases} 1/r_{ij} & \text{if } v_i \sim v_j \text{ and } r_{ij} \gneqq 0; \\ 0 & \text{otherwise.} \end{cases}$$

- But, how can we compute $\{r_{ij}\}$ if A is given? Note that r_{ij} should be defined even if $v_i \nsim v_j$.

Resistance Distance . . .

- Now, let's interpret the edge weights as *resistances* in an *electrical network*.
- For any v_i, v_j, $i \neq j$, suppose that a battery is connected across them so that one unit of a current flows in at v_i and out in v_j.
- Then, the voltage (potential) difference is the effective resistance between v_i and v_j by Ohm's law ($V = I \cdot R$), which is called the *resistance distance* between v_i and v_j and written as $r(v_i, v_j) = r_{ij}$.
- Hence, $1/r_{ij}$ is the *conductance* between v_i and v_j, which is a measure of *connectivity* (or *similarity*) between them, which should be the edge weight a_{ij}.
- Hence, given resistance distances $\{r_{ij}\}$, one can construct the corresponding weighted adjacency matrix A by

$$a_{ij} = \begin{cases} 1/r_{ij} & \text{if } v_i \sim v_j \text{ and } r_{ij} \gneqq 0; \\ 0 & \text{otherwise.} \end{cases}$$

- But, how can we compute $\{r_{ij}\}$ if A is given? Note that r_{ij} should be defined even if $v_i \nsim v_j$.

Resistance Distance . . .

- Now, let's interpret the edge weights as *resistances* in an *electrical network*.
- For any v_i, v_j, $i \neq j$, suppose that a battery is connected across them so that one unit of a current flows in at v_i and out in v_j.
- Then, the voltage (potential) difference is the effective resistance between v_i and v_j by Ohm's law ($V = I \cdot R$), which is called the *resistance distance* between v_i and v_j and written as $r(v_i, v_j) = r_{ij}$.
- Hence, $1/r_{ij}$ is the *conductance* between v_i and v_j, which is a measure of *connectivity* (or *similarity*) between them, which should be the edge weight a_{ij}.
- Hence, given resistance distances $\{r_{ij}\}$, one can construct the corresponding weighted adjacency matrix A by

$$a_{ij} = \begin{cases} 1/r_{ij} & \text{if } v_i \sim v_j \text{ and } r_{ij} \gneqq 0; \\ 0 & \text{otherwise.} \end{cases}$$

- But, how can we compute $\{r_{ij}\}$ if A is given? Note that r_{ij} should be defined even if $v_i \nsim v_j$.

Resistance Distance . . .

- Now, let's interpret the edge weights as *resistances* in an *electrical network*.
- For any v_i, v_j, $i \neq j$, suppose that a battery is connected across them so that one unit of a current flows in at v_i and out in v_j.
- Then, the voltage (potential) difference is the effective resistance between v_i and v_j by Ohm's law ($V = I \cdot R$), which is called the *resistance distance* between v_i and v_j and written as $r(v_i, v_j) = r_{ij}$.
- Hence, $1/r_{ij}$ is the *conductance* between v_i and v_j, which is a measure of *connectivity* (or *similarity*) between them, which should be the edge weight a_{ij}.
- Hence, given resistance distances $\{r_{ij}\}$, one can construct the corresponding weighted adjacency matrix A by

$$a_{ij} = \begin{cases} 1/r_{ij} & \text{if } v_i \sim v_j \text{ and } r_{ij} \gneqq 0; \\ 0 & \text{otherwise.} \end{cases}$$

- But, how can we compute $\{r_{ij}\}$ if A is given? Note that r_{ij} should be defined even if $v_i \nsim v_j$.

Resistance Distance . . .

- Now, let's interpret the edge weights as *resistances* in an *electrical network*.
- For any v_i, v_j, $i \neq j$, suppose that a battery is connected across them so that one unit of a current flows in at v_i and out in v_j.
- Then, the voltage (potential) difference is the effective resistance between v_i and v_j by Ohm's law ($V = I \cdot R$), which is called the *resistance distance* between v_i and v_j and written as $r(v_i, v_j) = r_{ij}$.
- Hence, $1/r_{ij}$ is the *conductance* between v_i and v_j, which is a measure of *connectivity* (or *similarity*) between them, which should be the edge weight a_{ij}.
- Hence, given resistance distances $\{r_{ij}\}$, one can construct the corresponding weighted adjacency matrix A by

$$a_{ij} = \begin{cases} 1/r_{ij} & \text{if } v_i \sim v_j \text{ and } r_{ij} \gneqq 0; \\ 0 & \text{otherwise.} \end{cases}$$

- But, how can we compute $\{r_{ij}\}$ if A is given? Note that r_{ij} should be defined even if $v_i \nsim v_j$.

Resistance Distance . . .

- Now, let's interpret the edge weights as *resistances* in an *electrical network*.
- For any v_i, v_j, $i \neq j$, suppose that a battery is connected across them so that one unit of a current flows in at v_i and out in v_j.
- Then, the voltage (potential) difference is the effective resistance between v_i and v_j by Ohm's law ($V = I \cdot R$), which is called the *resistance distance* between v_i and v_j and written as $r(v_i, v_j) = r_{ij}$.
- Hence, $1/r_{ij}$ is the *conductance* between v_i and v_j, which is a measure of *connectivity* (or *similarity*) between them, which should be the edge weight a_{ij}.
- Hence, given resistance distances $\{r_{ij}\}$, one can construct the corresponding weighted adjacency matrix A by

$$a_{ij} = \begin{cases} 1/r_{ij} & \text{if } v_i \sim v_j \text{ and } r_{ij} \gneqq 0; \\ 0 & \text{otherwise.} \end{cases}$$

- But, how can we compute $\{r_{ij}\}$ if A is given? Note that r_{ij} should be defined even if $v_i \nsim v_j$.

Resistance Distance . . .

- The amazing fact is the following relationship between the resistance distance and the *(Moore-Penrose) pseudoinverse* $L^{\dagger}(G)$ of the unnormalized graph Laplacian $L(G)$!

$$\begin{aligned} r_{ij} &= (L^{\dagger})_{ii} + (L^{\dagger})_{jj} - (L^{\dagger})_{ij} - (L^{\dagger})_{ji} \\ &= \left\langle \boldsymbol{e}_i - \boldsymbol{e}_j, L^{\dagger}(\boldsymbol{e}_i - \boldsymbol{e}_j) \right\rangle \\ &= (L^{\dagger})_{ii} + (L^{\dagger})_{jj} - 2(L^{\dagger})_{ij} \quad \text{if } G \text{ is undirected;} \end{aligned}$$

- Hence, the resistance matrix $R(G) = (r_{ij})$ can be computed via $L^{\dagger}(G)$.
- If G is *sparse*, then one can utilize a *sparse Cholesky factorization* of $L(G)$ to compute ith column vector $\boldsymbol{\ell}_i^{\dagger}$ of $L^{\dagger}$ as follows:
  1. Compute the projection of $\boldsymbol{e}_i$ onto range$(L(G))$, say $\boldsymbol{y}_i$ via $\boldsymbol{y}_i = (I - \mathbf{1}_n \mathbf{1}_n^{\mathsf{T}}/n)\boldsymbol{e}_i$ [Note null$(L(G)) = $ span$\{\mathbf{1}_n\}$ if G is connected.]
  2. Find a solution $\widehat{\boldsymbol{\ell}_i}$ of $L\boldsymbol{\ell} = \boldsymbol{y}_i$ where the Cholesky factorization of L should be utilized.
  3. Project the result on the row space of L (which is the same as the column space thanks to $L^{\mathsf{T}} = L$) to compute the ith column vector $\boldsymbol{\ell}_i^{\dagger} = (I - \mathbf{1}_n \mathbf{1}_n^{\mathsf{T}}/n)\widehat{\boldsymbol{\ell}_i}$.

Resistance Distance . . .

- The amazing fact is the following relationship between the resistance distance and the *(Moore-Penrose) pseudoinverse* $L^{\dagger}(G)$ of the unnormalized graph Laplacian $L(G)$!

$$\begin{aligned} r_{ij} &= (L^{\dagger})_{ii} + (L^{\dagger})_{jj} - (L^{\dagger})_{ij} - (L^{\dagger})_{ji} \\ &= \left\langle \boldsymbol{e}_i - \boldsymbol{e}_j, L^{\dagger}(\boldsymbol{e}_i - \boldsymbol{e}_j) \right\rangle \\ &= (L^{\dagger})_{ii} + (L^{\dagger})_{jj} - 2(L^{\dagger})_{ij} \quad \text{if } G \text{ is undirected;} \end{aligned}$$

- Hence, the resistance matrix $R(G) = (r_{ij})$ can be computed via $L^{\dagger}(G)$.
- If G is *sparse*, then one can utilize a *sparse Cholesky factorization* of $L(G)$ to compute ith column vector $\boldsymbol{\ell}_i^{\dagger}$ of $L^{\dagger}$ as follows:
  1. Compute the projection of $\boldsymbol{e}_i$ onto range$(L(G))$, say $\boldsymbol{y}_i$ via $\boldsymbol{y}_i = (I - \mathbf{1}_n \mathbf{1}_n^{\mathsf{T}}/n)\boldsymbol{e}_i$ [Note null$(L(G)) = $ span$\{\mathbf{1}_n\}$ if G is connected.]
  2. Find a solution $\widehat{\boldsymbol{\ell}_i}$ of $L\boldsymbol{\ell} = \boldsymbol{y}_i$ where the Cholesky factorization of L should be utilized.
  3. Project the result on the row space of L (which is the same as the column space thanks to $L^{\mathsf{T}} = L$) to compute the ith column vector $\boldsymbol{\ell}_i^{\dagger} = (I - \mathbf{1}_n \mathbf{1}_n^{\mathsf{T}}/n)\widehat{\boldsymbol{\ell}_i}$.

Resistance Distance . . .

- The amazing fact is the following relationship between the resistance distance and the *(Moore-Penrose) pseudoinverse* $L^\dagger(G)$ of the unnormalized graph Laplacian $L(G)$!

$$\begin{aligned} r_{ij} &= (L^\dagger)_{ii} + (L^\dagger)_{jj} - (L^\dagger)_{ij} - (L^\dagger)_{ji} \\ &= \left\langle \boldsymbol{e}_i - \boldsymbol{e}_j, L^\dagger(\boldsymbol{e}_i - \boldsymbol{e}_j) \right\rangle \\ &= (L^\dagger)_{ii} + (L^\dagger)_{jj} - 2(L^\dagger)_{ij} \quad \text{if } G \text{ is undirected;} \end{aligned}$$

- Hence, the resistance matrix $R(G) = (r_{ij})$ can be computed via $L^\dagger(G)$.
- If G is *sparse*, then one can utilize a *sparse Cholesky factorization* of $L(G)$ to compute ith column vector $\boldsymbol{\ell}_i^\dagger$ of $L^\dagger$ as follows:
  1. Compute the projection of $\boldsymbol{e}_i$ onto $\mathrm{range}(L(G))$, say $\boldsymbol{y}_i$ via $\boldsymbol{y}_i = (I - \mathbf{1}_n\mathbf{1}_n^\top/n)\boldsymbol{e}_i$ [Note $\mathrm{null}(L(G)) = \mathrm{span}\{\mathbf{1}_n\}$ if G is connected.]
  2. Find a solution $\widehat{\boldsymbol{\ell}_i}$ of $L\boldsymbol{\ell} = \boldsymbol{y}_i$ where the Cholesky factorization of L should be utilized.
  3. Project the result on the row space of L (which is the same as the column space thanks to $L^\top = L$) to compute the ith column vector $\boldsymbol{\ell}_i^\dagger = (I - \mathbf{1}_n\mathbf{1}_n^\top/n)\widehat{\boldsymbol{\ell}_i}$.

Resistance Distance ...

- The amazing fact is the following relationship between the resistance distance and the *(Moore-Penrose) pseudoinverse* $L^\dagger(G)$ of the unnormalized graph Laplacian $L(G)$!

$$\begin{aligned} r_{ij} &= (L^\dagger)_{ii} + (L^\dagger)_{jj} - (L^\dagger)_{ij} - (L^\dagger)_{ji} \\ &= \left\langle \boldsymbol{e}_i - \boldsymbol{e}_j, L^\dagger(\boldsymbol{e}_i - \boldsymbol{e}_j) \right\rangle \\ &= (L^\dagger)_{ii} + (L^\dagger)_{jj} - 2(L^\dagger)_{ij} \quad \text{if } G \text{ is undirected;} \end{aligned}$$

- Hence, the resistance matrix $R(G) = (r_{ij})$ can be computed via $L^\dagger(G)$.
- If G is *sparse*, then one can utilize a *sparse Cholesky factorization* of $L(G)$ to compute ith column vector $\boldsymbol{\ell}_i^\dagger$ of $L^\dagger$ as follows:
  1. Compute the projection of $\boldsymbol{e}_i$ onto $\mathrm{range}(L(G))$, say $\boldsymbol{y}_i$ via $\boldsymbol{y}_i = (I - \mathbf{1}_n \mathbf{1}_n^\top / n)\boldsymbol{e}_i$ [Note $\mathrm{null}(L(G)) = \mathrm{span}\{\mathbf{1}_n\}$ if G is connected.]
  2. Find a solution $\widehat{\boldsymbol{\ell}_i}$ of $L\boldsymbol{\ell} = \boldsymbol{y}_i$ where the Cholesky factorization of L should be utilized.
  3. Project the result on the row space of L (which is the same as the column space thanks to $L^\top = L$) to compute the ith column vector $\boldsymbol{\ell}_i^\dagger = (I - \mathbf{1}_n \mathbf{1}_n^\top / n)\widehat{\boldsymbol{\ell}_i}$.

Resistance Distance . . .

- The amazing fact is the following relationship between the resistance distance and the *(Moore-Penrose) pseudoinverse* $L^\dagger(G)$ of the unnormalized graph Laplacian $L(G)$!

$$\begin{aligned} r_{ij} &= (L^\dagger)_{ii} + (L^\dagger)_{jj} - (L^\dagger)_{ij} - (L^\dagger)_{ji} \\ &= \left\langle \boldsymbol{e}_i - \boldsymbol{e}_j, L^\dagger(\boldsymbol{e}_i - \boldsymbol{e}_j) \right\rangle \\ &= (L^\dagger)_{ii} + (L^\dagger)_{jj} - 2(L^\dagger)_{ij} \quad \text{if } G \text{ is undirected;} \end{aligned}$$

- Hence, the resistance matrix $R(G) = (r_{ij})$ can be computed via $L^\dagger(G)$.
- If G is *sparse*, then one can utilize a *sparse Cholesky factorization* of $L(G)$ to compute ith column vector $\boldsymbol{\ell}_i^\dagger$ of $L^\dagger$ as follows:
  1. Compute the projection of $\boldsymbol{e}_i$ onto $\mathrm{range}(L(G))$, say $\boldsymbol{y}_i$ via $\boldsymbol{y}_i = (I - \mathbf{1}_n \mathbf{1}_n^\top / n)\boldsymbol{e}_i$ [Note $\mathrm{null}(L(G)) = \mathrm{span}\{\mathbf{1}_n\}$ if G is connected.]
  2. Find a solution $\widehat{\boldsymbol{\ell}_i}$ of $L\boldsymbol{\ell} = \boldsymbol{y}_i$ where the Cholesky factorization of L should be utilized.
  3. Project the result on the row space of L (which is the same as the column space thanks to $L^\top = L$) to compute the ith column vector $\boldsymbol{\ell}_i^\dagger = (I - \mathbf{1}_n \mathbf{1}_n^\top / n)\widehat{\boldsymbol{\ell}_i}$.

Resistance Distance . . .

- The amazing fact is the following relationship between the resistance distance and the *(Moore-Penrose) pseudoinverse* $L^{\dagger}(G)$ of the unnormalized graph Laplacian $L(G)$!

$$\begin{aligned} r_{ij} &= (L^{\dagger})_{ii} + (L^{\dagger})_{jj} - (L^{\dagger})_{ij} - (L^{\dagger})_{ji} \\ &= \left\langle \boldsymbol{e}_i - \boldsymbol{e}_j, L^{\dagger}(\boldsymbol{e}_i - \boldsymbol{e}_j) \right\rangle \\ &= (L^{\dagger})_{ii} + (L^{\dagger})_{jj} - 2(L^{\dagger})_{ij} \quad \text{if } G \text{ is undirected;} \end{aligned}$$

- Hence, the resistance matrix $R(G) = (r_{ij})$ can be computed via $L^{\dagger}(G)$.
- If G is *sparse*, then one can utilize a *sparse Cholesky factorization* of $L(G)$ to compute ith column vector $\boldsymbol{\ell}_i^{\dagger}$ of $L^{\dagger}$ as follows:
  1. Compute the projection of $\boldsymbol{e}_i$ onto $\mathrm{range}(L(G))$, say $\boldsymbol{y}_i$ via $\boldsymbol{y}_i = (I - \mathbf{1}_n \mathbf{1}_n^{\mathsf{T}}/n)\boldsymbol{e}_i$ [Note $\mathrm{null}(L(G)) = \mathrm{span}\{\mathbf{1}_n\}$ if G is connected.]
  2. Find a solution $\widehat{\boldsymbol{\ell}_i}$ of $L\boldsymbol{\ell} = \boldsymbol{y}_i$ where the Cholesky factorization of L should be utilized.
  3. Project the result on the row space of L (which is the same as the column space thanks to $L^{\mathsf{T}} = L$) to compute the ith column vector $\boldsymbol{\ell}_i^{\dagger} = (I - \mathbf{1}_n \mathbf{1}_n^{\mathsf{T}}/n)\widehat{\boldsymbol{\ell}_i}$.

Intermezzo: The (Moore-Penrose) Pseudoinverse

Definition

The *pseudoinverse* $A^\dagger \in \mathbb{C}^{n\times m}$ of a general matrix $A \in \mathbb{C}^{m\times n}$ is defined to be the unique matrix $X \in \mathbb{C}^{n\times m}$ that satisfies the following *Moore-Penrose conditions*: i) $AXA = A$; ii) $XAX = X$; iii) $(AX)^* = AX$; iv) $(XA)^* = XA$, where A^* is the Hermitian transposition of A.

Using the SVD of $A = U\Sigma V^*$ where $U \in \mathbb{C}^{m\times m}$, $V \in \mathbb{C}^{n\times n}$ are unitary and $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_r, 0, \cdots, 0) \in \mathbb{R}^{m\times n}$, $r = \mathrm{rank}(A) \le \min(m,n)$, $A^\dagger$ can be expressed as

$$A^\dagger = V\Sigma^\dagger U^*, \quad \Sigma^\dagger := \mathrm{diag}(1/\sigma_1, \cdots, 1/\sigma_r, 0, \cdots, 0) \in \mathbb{R}^{n\times m}.$$

Note that $AA^\dagger$ and $A^\dagger A$ are the orthogonal projectors onto $\mathrm{range}(A)$ and $\mathrm{range}(A^*)$, respectively.

Intermezzo: The (Moore-Penrose) Pseudoinverse

Definition

The *pseudoinverse* $A^\dagger \in \mathbb{C}^{n \times m}$ of a general matrix $A \in \mathbb{C}^{m \times n}$ is defined to be the unique matrix $X \in \mathbb{C}^{n \times m}$ that satisfies the following *Moore-Penrose conditions*: i) $AXA = A$; ii) $XAX = X$; iii) $(AX)^* = AX$; iv) $(XA)^* = XA$, where A^* is the Hermitian transposition of A.

Using the SVD of $A = U\Sigma V^*$ where $U \in \mathbb{C}^{m \times m}$, $V \in \mathbb{C}^{n \times n}$ are unitary and $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_r, 0, \cdots, 0) \in \mathbb{R}^{m \times n}$, $r = \mathrm{rank}(A) \le \min(m, n)$, $A^\dagger$ can be expressed as

$$A^\dagger = V\Sigma^\dagger U^*, \quad \Sigma^\dagger := \mathrm{diag}(1/\sigma_1, \cdots, 1/\sigma_r, 0, \cdots, 0) \in \mathbb{R}^{n \times m}.$$

Note that $AA^\dagger$ and $A^\dagger A$ are the orthogonal projectors onto $\mathrm{range}(A)$ and $\mathrm{range}(A^*)$, respectively.

Intermezzo: The (Moore-Penrose) Pseudoinverse

Definition

The *pseudoinverse* $A^\dagger \in \mathbb{C}^{n\times m}$ of a general matrix $A \in \mathbb{C}^{m\times n}$ is defined to be the unique matrix $X \in \mathbb{C}^{n\times m}$ that satisfies the following *Moore-Penrose conditions*: i) $AXA = A$; ii) $XAX = X$; iii) $(AX)^* = AX$; iv) $(XA)^* = XA$, where A^* is the Hermitian transposition of A.

Using the SVD of $A = U\Sigma V^*$ where $U \in \mathbb{C}^{m\times m}$, $V \in \mathbb{C}^{n\times n}$ are unitary and $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_r, 0, \cdots, 0) \in \mathbb{R}^{m\times n}$, $r = \mathrm{rank}(A) \le \min(m,n)$, $A^\dagger$ can be expressed as

$$A^\dagger = V\Sigma^\dagger U^*, \quad \Sigma^\dagger := \mathrm{diag}(1/\sigma_1, \cdots, 1/\sigma_r, 0, \cdots, 0) \in \mathbb{R}^{n\times m}.$$

Note that $AA^\dagger$ and $A^\dagger A$ are the orthogonal projectors onto $\mathrm{range}(A)$ and $\mathrm{range}(A^*)$, respectively.

Intermezzo: The EP Matrices

Definition

If $A \in \mathbb{C}^{m\times m}$ commutes with its pseudoinverse, i.e., $A^\dagger A = AA^\dagger$, then it is called an *EP matrix* (EP for Equal Projection).

The properties of an EP matrix:

- If $(\lambda_i \neq 0, \boldsymbol{\phi}_i)$ is an eigenpair of A, then $(1/\lambda_i, \boldsymbol{\phi}_i)$ is an eigenpair of $A^\dagger$.
- If $(\lambda_i = 0, \boldsymbol{\phi}_i)$ is an eigenpair of A, then $(\lambda_i = 0, \boldsymbol{\phi}_i)$ is also an eigenpair of $A^\dagger$.

Intermezzo: The EP Matrices

Definition

If $A \in \mathbb{C}^{m \times m}$ commutes with its pseudoinverse, i.e., $A^\dagger A = AA^\dagger$, then it is called an *EP matrix* (EP for Equal Projection).

The properties of an EP matrix:

- If $(\lambda_i \neq 0, \boldsymbol{\phi}_i)$ is an eigenpair of A, then $(1/\lambda_i, \boldsymbol{\phi}_i)$ is an eigenpair of $A^\dagger$.
- If $(\lambda_i = 0, \boldsymbol{\phi}_i)$ is an eigenpair of A, then $(\lambda_i = 0, \boldsymbol{\phi}_i)$ is also an eigenpair of $A^\dagger$.

Properties of the Pseudoinverse $L^\dagger(G)$

- $\left(L^\dagger\right)^\top = L^\dagger$
- $\operatorname{rank}\left(L^\dagger\right) = n-1$.
- $\operatorname{null}(L) = \operatorname{null}(L^\dagger) = \operatorname{span}\{\mathbf{1}_n\}$.
- $L^\dagger$ is *doubly centered* just like L, i.e., its column sum and row sum are zeros.
- $L^\dagger$ is positive semidefinite just like L is.
- $L^\dagger = \left(L - \mathbf{1}_n\mathbf{1}_n^\top/n\right)^{-1} + \mathbf{1}_n\mathbf{1}_n^\top/n$.
- Since L represents the *local* properties of G (e.g., connectivities, etc.), $L^\dagger$ represents the *global* properties of $G \Longrightarrow$ *What is the relationship between $L^\dagger$ and the integral operator commuting with L in Lecture 2?*

Properties of the Pseudoinverse $L^{\dagger}(G)$

- $\left(L^{\dagger}\right)^{\top} = L^{\dagger}$
- $\operatorname{rank}\left(L^{\dagger}\right) = n-1.$
- $\operatorname{null}(L) = \operatorname{null}(L^{\dagger}) = \operatorname{span}\{\mathbf{1}_n\}.$
- $L^{\dagger}$ is *doubly centered* just like L, i.e., its column sum and row sum are zeros.
- $L^{\dagger}$ is positive semidefinite just like L is.
- $L^{\dagger} = \left(L - \mathbf{1}_n\mathbf{1}_n^{\top}/n\right)^{-1} + \mathbf{1}_n\mathbf{1}_n^{\top}/n.$
- Since L represents the *local* properties of G (e.g., connectivities, etc.), $L^{\dagger}$ represents the *global* properties of $G \Longrightarrow$ *What is the relationship between $L^{\dagger}$ and the integral operator commuting with L in Lecture 2?*

Properties of the Pseudoinverse $L^\dagger(G)$

- $\left(L^\dagger\right)^\top = L^\dagger$
- $\operatorname{rank}\left(L^\dagger\right) = n - 1$.
- $\operatorname{null}(L) = \operatorname{null}(L^\dagger) = \operatorname{span}\{\mathbf{1}_n\}$.
- $L^\dagger$ is *doubly centered* just like L, i.e., its column sum and row sum are zeros.
- $L^\dagger$ is positive semidefinite just like L is.
- $L^\dagger = \left(L - \mathbf{1}_n \mathbf{1}_n^\top / n\right)^{-1} + \mathbf{1}_n \mathbf{1}_n^\top / n$.
- Since L represents the *local* properties of G (e.g., connectivities, etc.), $L^\dagger$ represents the *global* properties of $G \Longrightarrow$ *What is the relationship between $L^\dagger$ and the integral operator commuting with L in Lecture 2?*

Properties of the Pseudoinverse $L^\dagger(G)$

- $\left(L^\dagger\right)^\mathsf{T} = L^\dagger$
- $\operatorname{rank}\left(L^\dagger\right) = n-1.$
- $\operatorname{null}(L) = \operatorname{null}(L^\dagger) = \operatorname{span}\{\mathbf{1}_n\}.$
- $L^\dagger$ is *doubly centered* just like L, i.e., its column sum and row sum are zeros.
- $L^\dagger$ is positive semidefinite just like L is.
- $L^\dagger = \left(L - \mathbf{1}_n\mathbf{1}_n^\mathsf{T}/n\right)^{-1} + \mathbf{1}_n\mathbf{1}_n^\mathsf{T}/n.$
- Since L represents the *local* properties of G (e.g., connectivities, etc.), $L^\dagger$ represents the *global* properties of $G \Longrightarrow$ *What is the relationship between $L^\dagger$ and the integral operator commuting with L in Lecture 2?*

Properties of the Pseudoinverse $L^\dagger(G)$

- $\left(L^\dagger\right)^\mathsf{T} = L^\dagger$
- $\operatorname{rank}\left(L^\dagger\right) = n-1$.
- $\operatorname{null}(L) = \operatorname{null}(L^\dagger) = \operatorname{span}\{\mathbf{1}_n\}$.
- $L^\dagger$ is *doubly centered* just like L, i.e., its column sum and row sum are zeros.
- $L^\dagger$ is positive semidefinite just like L is.
- $L^\dagger = \left(L - \mathbf{1}_n\mathbf{1}_n^\mathsf{T}/n\right)^{-1} + \mathbf{1}_n\mathbf{1}_n^\mathsf{T}/n$.
- Since L represents the *local* properties of G (e.g., connectivities, etc.), $L^\dagger$ represents the *global* properties of $G \Longrightarrow$ *What is the relationship between $L^\dagger$ and the integral operator commuting with L in Lecture 2?*

Properties of the Pseudoinverse $L^{\dagger}(G)$

- $\left(L^{\dagger}\right)^{\mathsf{T}} = L^{\dagger}$
- $\operatorname{rank}\left(L^{\dagger}\right) = n - 1$.
- $\operatorname{null}(L) = \operatorname{null}(L^{\dagger}) = \operatorname{span}\{\mathbf{1}_n\}$.
- $L^{\dagger}$ is *doubly centered* just like L, i.e., its column sum and row sum are zeros.
- $L^{\dagger}$ is positive semidefinite just like L is.
- $L^{\dagger} = \left(L - \mathbf{1}_n \mathbf{1}_n^{\mathsf{T}}/n\right)^{-1} + \mathbf{1}_n \mathbf{1}_n^{\mathsf{T}}/n$.
- Since L represents the *local* properties of G (e.g., connectivities, etc.), $L^{\dagger}$ represents the *global* properties of $G \Longrightarrow$ *What is the relationship between $L^{\dagger}$ and the integral operator commuting with L in Lecture 2?*

Properties of the Pseudoinverse $L^\dagger(G)$

- $\left(L^\dagger\right)^\mathsf{T} = L^\dagger$
- $\operatorname{rank}\left(L^\dagger\right) = n - 1$.
- $\operatorname{null}(L) = \operatorname{null}(L^\dagger) = \operatorname{span}\{\mathbf{1}_n\}$.
- $L^\dagger$ is *doubly centered* just like L, i.e., its column sum and row sum are zeros.
- $L^\dagger$ is positive semidefinite just like L is.
- $L^\dagger = \left(L - \mathbf{1}_n \mathbf{1}_n^\mathsf{T}/n\right)^{-1} + \mathbf{1}_n \mathbf{1}_n^\mathsf{T}/n$.
- Since L represents the *local* properties of G (e.g., connectivities, etc.), $L^\dagger$ represents the *global* properties of G $\Longrightarrow$ *What is the relationship between $L^\dagger$ and the integral operator commuting with L in Lecture 2?*

Outline

1 Need of a Variety of Graph Distances

2 The (Shortest) Path Distance

3 Resistance Distance

4 Commute-Time Distance

Commute-Time Distance

- Is quite similar to the resistance distance.
- Is intimately related to *random walks* on G.
- The *commute time* between v_i and v_j is *the expected time it takes the random walk to travel from v_i to v_j and back.*
- The commute time $c(v_i, v_j)$ is intimately related to the resistance distance $r(v_i, v_j)$:

$$c(v_i, v_j) = c_{ij} = \mathrm{vol}(V(G)) \cdot r(v_i, v_j) = \mathrm{vol}(V(G)) \cdot r_{ij}.$$

Commute-Time Distance

- Is quite similar to the resistance distance.
- Is intimately related to *random walks* on G.
- The *commute time* between v_i and v_j is *the expected time it takes the random walk to travel from v_i to v_j and back.*
- The commute time $c(v_i, v_j)$ is intimately related to the resistance distance $r(v_i, v_j)$:

$$c(v_i, v_j) = c_{ij} = \mathrm{vol}(V(G)) \cdot r(v_i, v_j) = \mathrm{vol}(V(G)) \cdot r_{ij}.$$

Commute-Time Distance

- Is quite similar to the resistance distance.
- Is intimately related to *random walks* on G.
- The *commute time* between v_i and v_j is *the expected time it takes the random walk to travel from v_i to v_j and back*.
- The commute time $c(v_i, v_j)$ is intimately related to the resistance distance $r(v_i, v_j)$:

$$c(v_i, v_j) = c_{ij} = \mathrm{vol}(V(G)) \cdot r(v_i, v_j) = \mathrm{vol}(V(G)) \cdot r_{ij}.$$

Commute-Time Distance

- Is quite similar to the resistance distance.
- Is intimately related to *random walks* on G.
- The *commute time* between v_i and v_j is *the expected time it takes the random walk to travel from v_i to v_j and back*.
- The commute time $c(v_i, v_j)$ is intimately related to the resistance distance $r(v_i, v_j)$:

$$c(v_i, v_j) = c_{ij} = \mathrm{vol}(V(G)) \cdot r(v_i, v_j) = \mathrm{vol}(V(G)) \cdot r_{ij}.$$

Intermezzo 2: Random Walks on Graphs

- The *Markov chain* describing the sequence of vertices in G (weighted, undirected, simple, and connected) visited by a random walker is called a *random walk*.
- A random variable $s(t)$ represents the state (i.e., vertex) of the Markov chain/random walker at time t.
- The random walk is defined with the following single-step *transition probability* of jumping from the state $v_i = s(t)$ at time t to an adjacent vertex $v_j = s(t+1)$ at time $t+1$:

$$\Pr(s(t+1) = v_j \,|\, s(t) = v_i) = a_{ij}/d_i =: p_{ij}.$$

- The transition probabilities depend only on the current state and not on the past states, i.e., the first-order Markov chain.
- Since G is connected, the Markov chain is *irreducible*, i.e., every state can be reached from any other state.
- Let $\boldsymbol{\pi}(t) = [\pi_1(t), \dots, \pi_n(t)]^{\mathsf{T}}$ where $\pi_i(t) := \Pr(s(t) = v_i)$, and let $P = (p_{ij}) = D^{-1}A$ be the *transition matrix*.

Intermezzo 2: Random Walks on Graphs

- The *Markov chain* describing the sequence of vertices in G (weighted, undirected, simple, and connected) visited by a random walker is called a *random walk*.
- A random variable $s(t)$ represents the state (i.e., vertex) of the Markov chain/random walker at time t.
- The random walk is defined with the following single-step *transition probability* of jumping from the state $v_i = s(t)$ at time t to an adjacent vertex $v_j = s(t+1)$ at time $t+1$:

$$\Pr(s(t+1) = v_j \,|\, s(t) = v_i) = a_{ij}/d_i =: p_{ij}.$$

- The transition probabilities depend only on the current state and not on the past states, i.e., the first-order Markov chain.
- Since G is connected, the Markov chain is *irreducible*, i.e., every state can be reached from any other state.
- Let $\boldsymbol{\pi}(t) = [\pi_1(t), \ldots, \pi_n(t)]^{\mathsf{T}}$ where $\pi_i(t) := \Pr(s(t) = v_i)$, and let $P = (p_{ij}) = D^{-1}A$ be the *transition matrix*.

Intermezzo 2: Random Walks on Graphs

- The *Markov chain* describing the sequence of vertices in G (weighted, undirected, simple, and connected) visited by a random walker is called a *random walk*.
- A random variable $s(t)$ represents the state (i.e., vertex) of the Markov chain/random walker at time t.
- The random walk is defined with the following single-step *transition probability* of jumping from the state $v_i = s(t)$ at time t to an adjacent vertex $v_j = s(t+1)$ at time $t+1$:

$$\Pr(s(t+1) = v_j \,|\, s(t) = v_i) = a_{ij}/d_i =: p_{ij}.$$

- The transition probabilities depend only on the current state and not on the past states, i.e., the first-order Markov chain.
- Since G is connected, the Markov chain is *irreducible*, i.e., every state can be reached from any other state.
- Let $\boldsymbol{\pi}(t) = [\pi_1(t), \ldots, \pi_n(t)]^{\mathsf{T}}$ where $\pi_i(t) := \Pr(s(t) = v_i)$, and let $P = (p_{ij}) = D^{-1}A$ be the *transition matrix*.

Intermezzo 2: Random Walks on Graphs

- The *Markov chain* describing the sequence of vertices in G (weighted, undirected, simple, and connected) visited by a random walker is called a *random walk*.
- A random variable $s(t)$ represents the state (i.e., vertex) of the Markov chain/random walker at time t.
- The random walk is defined with the following single-step *transition probability* of jumping from the state $v_i = s(t)$ at time t to an adjacent vertex $v_j = s(t+1)$ at time $t+1$:

$$\Pr(s(t+1) = v_j \,|\, s(t) = v_i) = a_{ij}/d_i =: p_{ij}.$$

- The transition probabilities depend only on the current state and not on the past states, i.e., the first-order Markov chain.
- Since G is connected, the Markov chain is *irreducible*, i.e., every state can be reached from any other state.
- Let $\boldsymbol{\pi}(t) = [\pi_1(t), \ldots, \pi_n(t)]^{\mathsf{T}}$ where $\pi_i(t) := \Pr(s(t) = v_i)$, and let $P = (p_{ij}) = D^{-1}A$ be the *transition matrix*.

Intermezzo 2: Random Walks on Graphs

- The *Markov chain* describing the sequence of vertices in G (weighted, undirected, simple, and connected) visited by a random walker is called a *random walk*.
- A random variable $s(t)$ represents the state (i.e., vertex) of the Markov chain/random walker at time t.
- The random walk is defined with the following single-step *transition probability* of jumping from the state $v_i = s(t)$ at time t to an adjacent vertex $v_j = s(t+1)$ at time $t+1$:

$$\Pr(s(t+1) = v_j \,|\, s(t) = v_i) = a_{ij}/d_i =: p_{ij}.$$

- The transition probabilities depend only on the current state and not on the past states, i.e., the first-order Markov chain.
- Since G is connected, the Markov chain is *irreducible*, i.e., every state can be reached from any other state.
- Let $\boldsymbol{\pi}(t) = [\pi_1(t), \ldots, \pi_n(t)]^{\mathsf{T}}$ where $\pi_i(t) := \Pr(s(t) = v_i)$, and let $P = (p_{ij}) = D^{-1}A$ be the *transition matrix*.

Intermezzo 2: Random Walks on Graphs

- The *Markov chain* describing the sequence of vertices in G (weighted, undirected, simple, and connected) visited by a random walker is called a *random walk*.
- A random variable $s(t)$ represents the state (i.e., vertex) of the Markov chain/random walker at time t.
- The random walk is defined with the following single-step *transition probability* of jumping from the state $v_i = s(t)$ at time t to an adjacent vertex $v_j = s(t+1)$ at time $t+1$:

$$\Pr(s(t+1) = v_j \,|\, s(t) = v_i) = a_{ij}/d_i =: p_{ij}.$$

- The transition probabilities depend only on the current state and not on the past states, i.e., the first-order Markov chain.
- Since G is connected, the Markov chain is *irreducible*, i.e., every state can be reached from any other state.
- Let $\boldsymbol{\pi}(t) = [\pi_1(t), \ldots, \pi_n(t)]^\mathsf{T}$ where $\pi_i(t) := \Pr(s(t) = v_i)$, and let $P = (p_{ij}) = D^{-1}A$ be the *transition matrix*.

Intermezzo 2: Random Walks on Graphs . . .

- Then, the evolution of the Markov chain is characterized by

$$\boldsymbol{\pi}(t+1) = P^{\mathsf{T}}\boldsymbol{\pi}(t) \Leftrightarrow \boldsymbol{\pi}^{\mathsf{T}}(t+1) = \boldsymbol{\pi}^{\mathsf{T}}(t)P\,.$$

- Let $\boldsymbol{\pi}^0 = \boldsymbol{\pi}(0)$ be the initial distribution.
- The *stationary distribution* $\boldsymbol{\pi}^\infty$ satisfies $\boldsymbol{\pi}^\infty = P^{\mathsf{T}}\boldsymbol{\pi}^\infty$, i.e., the eigenvector of P^{T} (a.k.a. the *left* eigenvector of P) corresponding to the eigenvalue 1.
- Can show $\boldsymbol{\pi}^\infty = [d_1, \dots, d_n]^{\mathsf{T}} / \mathrm{vol}(V)$.

Proof: Since $L = D - A = D(I - P)$, $P^{\mathsf{T}} = I - LD^{-1}$. Now, we have $P^{\mathsf{T}}D\mathbf{1}_n = D\mathbf{1}_n - L\mathbf{1}_n = D\mathbf{1}_n$. Hence, $D\mathbf{1}_n = [d_1, \dots, d_n]^{\mathsf{T}}$ is the eigenvector of P^{T} corresponding to the eigenvalue 1 modulo normalization constants. To make it as a probability distribution over V, we need to normalize it by $\mathrm{vol}(V) = \sum_j d_j$, i.e., $\boldsymbol{\pi}^\infty = D\mathbf{1}_n / \mathrm{vol}(V)$. □

Intermezzo 2: Random Walks on Graphs . . .

- Then, the evolution of the Markov chain is characterized by

$$\boldsymbol{\pi}(t+1) = P^{\mathsf{T}}\boldsymbol{\pi}(t) \Leftrightarrow \boldsymbol{\pi}^{\mathsf{T}}(t+1) = \boldsymbol{\pi}^{\mathsf{T}}(t)P\,.$$

- Let $\boldsymbol{\pi}^0 = \boldsymbol{\pi}(0)$ be the initial distribution.
- The *stationary distribution* $\boldsymbol{\pi}^\infty$ satisfies $\boldsymbol{\pi}^\infty = P^{\mathsf{T}}\boldsymbol{\pi}^\infty$, i.e., the eigenvector of P^{T} (a.k.a. the *left* eigenvector of P) corresponding to the eigenvalue 1.
- Can show $\boldsymbol{\pi}^\infty = [d_1,\ldots,d_n]^{\mathsf{T}}/\operatorname{vol}(V)$.

Proof: Since $L = D - A = D(I-P)$, $P^{\mathsf{T}} = I - LD^{-1}$. Now, we have $P^{\mathsf{T}}D\mathbf{1}_n = D\mathbf{1}_n - L\mathbf{1}_n = D\mathbf{1}_n$. Hence, $D\mathbf{1}_n = [d_1,\ldots,d_n]^{\mathsf{T}}$ is the eigenvector of P^{T} corresponding to the eigenvalue 1 modulo normalization constants. To make it as a probability distribution over V, we need to normalize it by $\operatorname{vol}(V) = \sum_j d_j$, i.e., $\boldsymbol{\pi}^\infty = D\mathbf{1}_n/\operatorname{vol}(V)$. □

Intermezzo 2: Random Walks on Graphs . . .

- Then, the evolution of the Markov chain is characterized by

$$\boldsymbol{\pi}(t+1) = P^{\mathsf{T}}\boldsymbol{\pi}(t) \Leftrightarrow \boldsymbol{\pi}^{\mathsf{T}}(t+1) = \boldsymbol{\pi}^{\mathsf{T}}(t)P\,.$$

- Let $\boldsymbol{\pi}^0 = \boldsymbol{\pi}(0)$ be the initial distribution.
- The *stationary distribution* $\boldsymbol{\pi}^\infty$ satisfies $\boldsymbol{\pi}^\infty = P^{\mathsf{T}}\boldsymbol{\pi}^\infty$, i.e., the eigenvector of P^{T} (a.k.a. the *left* eigenvector of P) corresponding to the eigenvalue 1.
- Can show $\boldsymbol{\pi}^\infty = [d_1, \ldots, d_n]^{\mathsf{T}} / \operatorname{vol}(V)$.

Proof: Since $L = D - A = D(I-P)$, $P^{\mathsf{T}} = I - LD^{-1}$. Now, we have $P^{\mathsf{T}}D\mathbf{1}_n = D\mathbf{1}_n - L\mathbf{1}_n = D\mathbf{1}_n$. Hence, $D\mathbf{1}_n = [d_1, \ldots, d_n]^{\mathsf{T}}$ is the eigenvector of P^{T} corresponding to the eigenvalue 1 modulo normalization constants. To make it as a probability distribution over V, we need to normalize it by $\operatorname{vol}(V) = \sum_j d_j$, i.e., $\boldsymbol{\pi}^\infty = D\mathbf{1}_n / \operatorname{vol}(V)$. □

Intermezzo 2: Random Walks on Graphs . . .

- Then, the evolution of the Markov chain is characterized by

$$\boldsymbol{\pi}(t+1) = P^{\mathsf{T}}\boldsymbol{\pi}(t) \Leftrightarrow \boldsymbol{\pi}^{\mathsf{T}}(t+1) = \boldsymbol{\pi}^{\mathsf{T}}(t)P\,.$$

- Let $\boldsymbol{\pi}^0 = \boldsymbol{\pi}(0)$ be the initial distribution.
- The *stationary distribution* $\boldsymbol{\pi}^\infty$ satisfies $\boldsymbol{\pi}^\infty = P^{\mathsf{T}}\boldsymbol{\pi}^\infty$, i.e., the eigenvector of P^{T} (a.k.a. the *left* eigenvector of P) corresponding to the eigenvalue 1.
- Can show $\boldsymbol{\pi}^\infty = [d_1,\ldots,d_n]^{\mathsf{T}} / \mathrm{vol}(V)$.

Proof: Since $L = D - A = D(I-P)$, $P^{\mathsf{T}} = I - LD^{-1}$. Now, we have $P^{\mathsf{T}}D\mathbf{1}_n = D\mathbf{1}_n - L\mathbf{1}_n = D\mathbf{1}_n$. Hence, $D\mathbf{1}_n = [d_1,\ldots,d_n]^{\mathsf{T}}$ is the eigenvector of P^{T} corresponding to the eigenvalue 1 modulo normalization constants. To make it as a probability distribution over V, we need to normalize it by $\mathrm{vol}(V) = \sum_j d_j$, i.e., $\boldsymbol{\pi}^\infty = D\mathbf{1}_n / \mathrm{vol}(V)$. □

Intermezzo 2: Random Walks on Graphs . . .

- Then, the evolution of the Markov chain is characterized by

$$\boldsymbol{\pi}(t+1) = P^{\mathsf{T}}\boldsymbol{\pi}(t) \Leftrightarrow \boldsymbol{\pi}^{\mathsf{T}}(t+1) = \boldsymbol{\pi}^{\mathsf{T}}(t)P\,.$$

- Let $\boldsymbol{\pi}^0 = \boldsymbol{\pi}(0)$ be the initial distribution.
- The *stationary distribution* $\boldsymbol{\pi}^\infty$ satisfies $\boldsymbol{\pi}^\infty = P^{\mathsf{T}}\boldsymbol{\pi}^\infty$, i.e., the eigenvector of P^{T} (a.k.a. the *left* eigenvector of P) corresponding to the eigenvalue 1.
- Can show $\boldsymbol{\pi}^\infty = [d_1, \ldots, d_n]^{\mathsf{T}} / \operatorname{vol}(V)$.

Proof: Since $L = D - A = D(I - P)$, $P^{\mathsf{T}} = I - LD^{-1}$. Now, we have $P^{\mathsf{T}} D\mathbf{1}_n = D\mathbf{1}_n - L\mathbf{1}_n = D\mathbf{1}_n$. Hence, $D\mathbf{1}_n = [d_1, \ldots, d_n]^{\mathsf{T}}$ is the eigenvector of P^{T} corresponding to the eigenvalue 1 modulo normalization constants. To make it as a probability distribution over V, we need to normalize it by $\operatorname{vol}(V) = \sum_j d_j$, i.e., $\boldsymbol{\pi}^\infty = D\mathbf{1}_n / \operatorname{vol}(V)$. □

Intermezzo 2: Random Walks on Graphs . . .

There is a formal equivalence between Ncut and transition probabilities of the random walk:

Proposition (Meila and Shi (2001))

Let G be connected and non bipartite. Assume that we run the random walk $s(t)$ starting with $s(0)$ in the stationary distribution $\boldsymbol{\pi}^\infty$. For disjoint subsets $X, Y \subset V$, denote $\Pr(Y \mid X) := \Pr(s(1) \in Y \mid s(0) \in X)$. Then

$$\mathrm{Ncut}(X, X^c) = \Pr(X^c \mid X) + \Pr(X \mid X^c).$$

Intermezzo 2: Random Walks on Graphs . . .

Proof: First of all, observe that

$$\begin{aligned}\Pr(s(0) \in X, s(1) \in Y) &= \sum_{x \in X, y \in Y} \Pr(s(0) = x, s(1) = y) = \sum_{x \in X, y \in Y} \boldsymbol{\pi}_x^\infty p_{xy} \\ &= \sum_{x \in X, y \in Y} \frac{d_x}{\mathrm{vol}(V)} \cdot \frac{a_{xy}}{d_x} = \frac{1}{\mathrm{vol}(V)} \sum_{x \in X, y \in Y} a_{xy}.\end{aligned}$$

Using this, we obtain

$$\begin{aligned}\Pr(s(1) \in Y \,|\, s(0) \in X) &= \frac{\Pr(s(0) \in X, s(1) \in Y)}{\Pr(s(0) \in X)} \\ &= \frac{1}{\mathrm{vol}(V)} \left(\sum_{x \in X, y \in Y} a_{xy} \right) \cdot \left(\frac{\mathrm{vol}(X)}{\mathrm{vol}(V)} \right)^{-1} \\ &= \frac{\sum_{x \in X, y \in Y} a_{xy}}{\mathrm{vol}(X)}.\end{aligned}$$

Now, it is clear that the proposition holds. □

Average First-Passage Time/Cost

- The *average first-passage time* $m(v_k \mid v_i) = m(k \mid i)$ is defined as the average number of steps that a random walker, starting in state $v_i \neq v_k$, will take to enter state v_k for the first time. More precisely, we define the minimum time until hitting state v_k, when staring from state v_i, as $T_{ik} := \min(t \geq 0 \mid s(t) = v_k; s(0) = v_i)$ for one realization of the stochastic process.
- The random walker may pass through v_k repeatedly; the minimum time corresponds to the first passage.
- The average first-passage time is the expected first-passage time: $m(k \mid i) := E[T_{ik} \mid s(0) = v_i]$.
- Similarly, the *average first-passage cost* $o(v_k \mid v_i) = o(k \mid i)$ is the average cost incurred by the random walker starting from state v_i to reach state v_k for the first time. Let $\gamma(j \mid i)$ be the cost of transition from state v_i to v_j. For example, $\gamma(j \mid i) = 1/a_{ij}$ (if $i \sim j$) is a possibility.

Average First-Passage Time/Cost

- The *average first-passage time* $m(v_k \,|\, v_i) = m(k \,|\, i)$ is defined as the average number of steps that a random walker, starting in state $v_i \neq v_k$, will take to enter state v_k for the first time. More precisely, we define the minimum time until hitting state v_k, when staring from state v_i, as $T_{ik} := \min(t \geq 0 \,|\, s(t) = v_k; s(0) = v_i)$ for one realization of the stochastic process.
- The random walker may pass through v_k repeatedly; the minimum time corresponds to the first passage.
- The average first-passage time is the expected first-passage time: $m(k \,|\, i) := E[T_{ik} \,|\, s(0) = v_i]$.
- Similarly, the *average first-passage cost* $o(v_k \,|\, v_i) = o(k \,|\, i)$ is the average cost incurred by the random walker starting from state v_i to reach state v_k for the first time. Let $\gamma(j \,|\, i)$ be the cost of transition from state v_i to v_j. For example, $\gamma(j \,|\, i) = 1/a_{ij}$ (if $i \sim j$) is a possibility.

Average First-Passage Time/Cost

- The *average first-passage time* $m(v_k \mid v_i) = m(k \mid i)$ is defined as the average number of steps that a random walker, starting in state $v_i \neq v_k$, will take to enter state v_k for the first time. More precisely, we define the minimum time until hitting state v_k, when staring from state v_i, as $T_{ik} := \min(t \geq 0 \mid s(t) = v_k; s(0) = v_i)$ for one realization of the stochastic process.
- The random walker may pass through v_k repeatedly; the minimum time corresponds to the first passage.
- The average first-passage time is the expected first-passage time: $m(k \mid i) := E[T_{ik} \mid s(0) = v_i]$.
- Similarly, the *average first-passage cost* $o(v_k \mid v_i) = o(k \mid i)$ is the average cost incurred by the random walker starting from state v_i to reach state v_k for the first time. Let $\gamma(j \mid i)$ be the cost of transition from state v_i to v_j. For example, $\gamma(j \mid i) = 1/a_{ij}$ (if $i \sim j$) is a possibility.

Average First-Passage Time/Cost

- The *average first-passage time* $m(v_k \,|\, v_i) = m(k \,|\, i)$ is defined as the average number of steps that a random walker, starting in state $v_i \neq v_k$, will take to enter state v_k for the first time. More precisely, we define the minimum time until hitting state v_k, when staring from state v_i, as $T_{ik} := \min(t \geq 0 \,|\, s(t) = v_k; s(0) = v_i)$ for one realization of the stochastic process.
- The random walker may pass through v_k repeatedly; the minimum time corresponds to the first passage.
- The average first-passage time is the expected first-passage time: $m(k \,|\, i) := E[T_{ik} \,|\, s(0) = v_i]$.
- Similarly, the *average first-passage cost* $o(v_k \,|\, v_i) = o(k \,|\, i)$ is the average cost incurred by the random walker starting from state v_i to reach state v_k for the first time. Let $\gamma(j \,|\, i)$ be the cost of transition from state v_i to v_j. For example, $\gamma(j \,|\, i) = 1/a_{ij}$ (if $i \sim j$) is a possibility.

Average First-Passage Time & Average Commute Time

- The recurrence relations computing $m(k\,|\,i)$ and $o(k\,|\,i)$ can easily be obtained by first-step analysis:

$$m(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ 1 + \sum_{j=1}^{n} p_{ij}\, m(k\,|\,j) & \text{otherwise.} \end{cases}$$

$$o(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ \sum_{j=1}^{n} p_{ij} \left(\gamma(j\,|\,i) + o(k\,|\,j) \right) & \text{otherwise.} \end{cases}$$

- A closely related quantity, the *average commute time* $c(v_i, v_j) = c(i, j) = c_{ij}$ is defined as the average number of steps that a random walker, starting in state v_i, will take to enter state v_j, $i \neq j$ for the first time and *go back to v_i*.
- That is, $c(i, j) = m(j\,|\,i) + m(i\,|\,j)$.
- Note that $c(i, j) = c(j, i)$ but $m(j\,|\,i) \neq m(i\,|\,j)$ in general.
- $c(i, j)$ is a *metric* on G:
    1. $c(i, j) \geq 0$, and the equality holds iff $v_i = v_j$;
    2. $c(i, j) = c(j, i)$;
    3. $c(i, j) \leq c(i, k) + c(k, j)$.

Average First-Passage Time & Average Commute Time

- The recurrence relations computing $m(k\,|\,i)$ and $o(k\,|\,i)$ can easily be obtained by first-step analysis:

$$m(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ 1 + \sum_{j=1}^{n} p_{ij}\, m(k\,|\,j) & \text{otherwise.} \end{cases}$$

$$o(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ \sum_{j=1}^{n} p_{ij} \left(\gamma(j\,|\,i) + o(k\,|\,j)\right) & \text{otherwise.} \end{cases}$$

- A closely related quantity, the *average commute time* $c(v_i, v_j) = c(i, j) = c_{ij}$ is defined as the average number of steps that a random walker, starting in state v_i, will take to enter state v_j, $i \neq j$ for the first time and *go back to v_i*.
- That is, $c(i, j) = m(j\,|\,i) + m(i\,|\,j)$.
- Note that $c(i, j) = c(j, i)$ but $m(j\,|\,i) \neq m(i\,|\,j)$ in general.
- $c(i, j)$ is a *metric* on G:
  1. $c(i, j) \geq 0$, and the equality holds iff $v_i = v_j$;
  2. $c(i, j) = c(j, i)$;
  3. $c(i, j) \leq c(i, k) + c(k, j)$.

Average First-Passage Time & Average Commute Time

- The recurrence relations computing $m(k\,|\,i)$ and $o(k\,|\,i)$ can easily be obtained by first-step analysis:

$$m(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ 1 + \sum_{j=1}^{n} p_{ij}\, m(k\,|\,j) & \text{otherwise.} \end{cases}$$

$$o(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ \sum_{j=1}^{n} p_{ij} \left(\gamma(j\,|\,i) + o(k\,|\,j)\right) & \text{otherwise.} \end{cases}$$

- A closely related quantity, the *average commute time* $c(v_i, v_j) = c(i, j) = c_{ij}$ is defined as the average number of steps that a random walker, starting in state v_i, will take to enter state v_j, $i \neq j$ for the first time and *go back to v_i*.
- That is, $c(i, j) = m(j\,|\,i) + m(i\,|\,j)$.
- Note that $c(i, j) = c(j, i)$ but $m(j\,|\,i) \neq m(i\,|\,j)$ in general.
- $c(i, j)$ is a *metric* on G:
  1. $c(i, j) \geq 0$, and the equality holds iff $v_i = v_j$;
  2. $c(i, j) = c(j, i)$;
  3. $c(i, j) \leq c(i, k) + c(k, j)$.

Average First-Passage Time & Average Commute Time

- The recurrence relations computing $m(k\,|\,i)$ and $o(k\,|\,i)$ can easily be obtained by first-step analysis:

$$m(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ 1 + \sum_{j=1}^{n} p_{ij}\, m(k\,|\,j) & \text{otherwise.} \end{cases}$$

$$o(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ \sum_{j=1}^{n} p_{ij} \big(\gamma(j\,|\,i) + o(k\,|\,j)\big) & \text{otherwise.} \end{cases}$$

- A closely related quantity, the *average commute time* $c(v_i, v_j) = c(i, j) = c_{ij}$ is defined as the average number of steps that a random walker, starting in state v_i, will take to enter state v_j, $i \neq j$ for the first time and *go back to v_i*.
- That is, $c(i, j) = m(j\,|\,i) + m(i\,|\,j)$.
- Note that $c(i, j) = c(j, i)$ but $m(j\,|\,i) \neq m(i\,|\,j)$ in general.
- $c(i, j)$ is a *metric* on G:
  1. $c(i, j) \geq 0$, and the equality holds iff $v_i = v_j$;
  2. $c(i, j) = c(j, i)$;
  3. $c(i, j) \leq c(i, k) + c(k, j)$.

Average First-Passage Time & Average Commute Time

- The recurrence relations computing $m(k\,|\,i)$ and $o(k\,|\,i)$ can easily be obtained by first-step analysis:

$$m(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ 1 + \sum_{j=1}^{n} p_{ij}\, m(k\,|\,j) & \text{otherwise.} \end{cases}$$

$$o(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ \sum_{j=1}^{n} p_{ij} \left(\gamma(j\,|\,i) + o(k\,|\,j) \right) & \text{otherwise.} \end{cases}$$

- A closely related quantity, the *average commute time* $c(v_i, v_j) = c(i, j) = c_{ij}$ is defined as the average number of steps that a random walker, starting in state v_i, will take to enter state v_j, $i \neq j$ for the first time and *go back to v_i*.
- That is, $c(i, j) = m(j\,|\,i) + m(i\,|\,j)$.
- Note that $c(i, j) = c(j, i)$ but $m(j\,|\,i) \neq m(i\,|\,j)$ in general.
- $c(i, j)$ is a *metric* on G:
    1. $c(i, j) \geq 0$, and the equality holds iff $v_i = v_j$;
    2. $c(i, j) = c(j, i)$;
    3. $c(i, j) \leq c(i, k) + c(k, j)$.

Average First-Passage Time & Average Commute Time

- The recurrence relations computing $m(k\,|\,i)$ and $o(k\,|\,i)$ can easily be obtained by first-step analysis:

$$m(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ 1 + \sum_{j=1}^{n} p_{ij}\, m(k\,|\,j) & \text{otherwise.} \end{cases}$$

$$o(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ \sum_{j=1}^{n} p_{ij} \left(\gamma(j\,|\,i) + o(k\,|\,j) \right) & \text{otherwise.} \end{cases}$$

- A closely related quantity, the *average commute time* $c(v_i, v_j) = c(i, j) = c_{ij}$ is defined as the average number of steps that a random walker, starting in state v_i, will take to enter state v_j, $i \neq j$ for the first time and *go back to v_i*.
- That is, $c(i, j) = m(j\,|\,i) + m(i\,|\,j)$.
- Note that $c(i, j) = c(j, i)$ but $m(j\,|\,i) \neq m(i\,|\,j)$ in general.
- $c(i, j)$ is a *metric* on G:
  1. $c(i, j) \geq 0$, and the equality holds iff $v_i = v_j$;
  2. $c(i, j) = c(j, i)$;
  3. $c(i, j) \leq c(i, k) + c(k, j)$.

Average First-Passage Time & Average Commute Time

- The recurrence relations computing $m(k\,|\,i)$ and $o(k\,|\,i)$ can easily be obtained by first-step analysis:

$$m(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ 1 + \sum_{j=1}^{n} p_{ij}\, m(k\,|\,j) & \text{otherwise.} \end{cases}$$

$$o(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ \sum_{j=1}^{n} p_{ij} \left(\gamma(j\,|\,i) + o(k\,|\,j)\right) & \text{otherwise.} \end{cases}$$

- A closely related quantity, the *average commute time* $c(v_i, v_j) = c(i,j) = c_{ij}$ is defined as the average number of steps that a random walker, starting in state v_i, will take to enter state v_j, $i \neq j$ for the first time and *go back to v_i*.
- That is, $c(i,j) = m(j\,|\,i) + m(i\,|\,j)$.
- Note that $c(i,j) = c(j,i)$ but $m(j\,|\,i) \neq m(i\,|\,j)$ in general.
- $c(i,j)$ is a *metric* on G:
  1. $c(i,j) \geq 0$, and the equality holds iff $v_i = v_j$;
  2. $c(i,j) = c(j,i)$;
  3. $c(i,j) \leq c(i,k) + c(k,j)$.

Average First-Passage Time & Average Commute Time

- The recurrence relations computing $m(k\,|\,i)$ and $o(k\,|\,i)$ can easily be obtained by first-step analysis:

$$m(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ 1 + \sum_{j=1}^{n} p_{ij}\, m(k\,|\,j) & \text{otherwise.} \end{cases}$$

$$o(k\,|\,i) = \begin{cases} 0 & \text{if } i = k; \\ \sum_{j=1}^{n} p_{ij} \left(\gamma(j\,|\,i) + o(k\,|\,j)\right) & \text{otherwise.} \end{cases}$$

- A closely related quantity, the *average commute time* $c(v_i, v_j) = c(i,j) = c_{ij}$ is defined as the average number of steps that a random walker, starting in state v_i, will take to enter state v_j, $i \neq j$ for the first time and *go back to v_i*.
- That is, $c(i,j) = m(j\,|\,i) + m(i\,|\,j)$.
- Note that $c(i,j) = c(j,i)$ but $m(j\,|\,i) \neq m(i\,|\,j)$ in general.
- $c(i,j)$ is a *metric* on G:
  1. $c(i,j) \geq 0$, and the equality holds iff $v_i = v_j$;
  2. $c(i,j) = c(j,i)$;
  3. $c(i,j) \leq c(i,k) + c(k,j)$.

Commute-Time Distance & Resistance Distance

- Hence, $c(i,j)$ is called the *commute-time distance* between v_i and v_j.
- That amazing property of the resistance distance carries over to the commute-time distance:

$$\begin{aligned} c_{ij} &= \mathrm{vol}(V)\cdot r_{ij} \\ &= \mathrm{vol}(V)\left\langle \boldsymbol{e}_i-\boldsymbol{e}_j, L^{\dagger}(\boldsymbol{e}_i-\boldsymbol{e}_j)\right\rangle \\ &= \mathrm{vol}(V)\left((L^{\dagger})_{ii}+(L^{\dagger})_{jj}-2(L^{\dagger})_{ij}\right). \end{aligned}$$

- Hence, both $\sqrt{r_{ij}}$ and $\sqrt{c_{ij}}$ are nothing but a *Mahalanobis distance* with a weighting matrix $L^{\dagger}$ and $\mathrm{vol}(V)L^{\dagger}$, respectively.

Commute-Time Distance & Resistance Distance

- Hence, $c(i, j)$ is called the *commute-time distance* between v_i and v_j.
- That amazing property of the resistance distance carries over to the commute-time distance:

$$\begin{aligned} c_{ij} &= \mathrm{vol}(V) \cdot r_{ij} \\ &= \mathrm{vol}(V) \left\langle \boldsymbol{e}_i - \boldsymbol{e}_j, L^\dagger (\boldsymbol{e}_i - \boldsymbol{e}_j) \right\rangle \\ &= \mathrm{vol}(V) \left((L^\dagger)_{ii} + (L^\dagger)_{jj} - 2(L^\dagger)_{ij} \right). \end{aligned}$$

- Hence, both $\sqrt{r_{ij}}$ and $\sqrt{c_{ij}}$ are nothing but a *Mahalanobis distance* with a weighting matrix $L^\dagger$ and $\mathrm{vol}(V)L^\dagger$, respectively.

Commute-Time Distance & Resistance Distance

- Hence, $c(i,j)$ is called the *commute-time distance* between v_i and v_j.
- That amazing property of the resistance distance carries over to the commute-time distance:

$$\begin{aligned} c_{ij} &= \mathrm{vol}(V)\cdot r_{ij} \\ &= \mathrm{vol}(V)\left\langle \boldsymbol{e}_i-\boldsymbol{e}_j, L^{\dagger}(\boldsymbol{e}_i-\boldsymbol{e}_j)\right\rangle \\ &= \mathrm{vol}(V)\left((L^{\dagger})_{ii}+(L^{\dagger})_{jj}-2(L^{\dagger})_{ij}\right). \end{aligned}$$

- Hence, both $\sqrt{r_{ij}}$ and $\sqrt{c_{ij}}$ are nothing but a *Mahalanobis distance* with a weighting matrix $L^{\dagger}$ and $\mathrm{vol}(V)L^{\dagger}$, respectively.

Commute-Time Distance & Resistance Distance . . .

- The above formula also allows us to interpret $\sqrt{r_{ij}}$ and $\sqrt{c_{ij}}$ as Euclidean distances on $V(G)$, i.e., to *embed* $v_i \in V(G)$ on a point $\boldsymbol{z}_i \in \mathbb{R}^n$.
- Let $L = \Phi\Lambda\Phi^{\mathsf{T}}$ be the eigendecomposition of L. Then, $L^{\dagger} = \Phi\Lambda^{\dagger}\Phi^{\mathsf{T}}$ where $\Lambda^{\dagger} = \operatorname{diag}\left(\lambda_0^{\dagger}, \ldots, \lambda_{n-1}^{\dagger}\right)$ is defined as

$$\lambda_i^{\dagger} := \begin{cases} 1/\lambda_i & \text{if } \lambda_i \neq 0; \\ 0 & \text{otherwise.} \end{cases}$$

- Hence, $\boldsymbol{z}_i^{\mathsf{T}} = \left(\Phi\sqrt{\Lambda^{\dagger}}\right)_{i,1:n}$ for the resistance distance and $\boldsymbol{z}_i^{\mathsf{T}} = \left(\Phi\sqrt{\mathrm{vol}(V)\Lambda^{\dagger}}\right)_{i,1:n}$ for the commute-time distance.

Commute-Time Distance & Resistance Distance . . .

- The above formula also allows us to interpret $\sqrt{r_{ij}}$ and $\sqrt{c_{ij}}$ as Euclidean distances on $V(G)$, i.e., to *embed* $v_i \in V(G)$ on a point $\boldsymbol{z}_i \in \mathbb{R}^n$.
- Let $L = \Phi\Lambda\Phi^{\mathsf{T}}$ be the eigendecomposition of L. Then, $L^{\dagger} = \Phi\Lambda^{\dagger}\Phi^{\mathsf{T}}$ where $\Lambda^{\dagger} = \mathrm{diag}\left(\lambda_0^{\dagger}, \ldots, \lambda_{n-1}^{\dagger}\right)$ is defined as

$$\lambda_i^{\dagger} := \begin{cases} 1/\lambda_i & \text{if } \lambda_i \neq 0; \\ 0 & \text{otherwise.} \end{cases}$$

- Hence, $\boldsymbol{z}_i^{\mathsf{T}} = \left(\Phi\sqrt{\Lambda^{\dagger}}\right)_{i,1:n}$ for the resistance distance and $\boldsymbol{z}_i^{\mathsf{T}} = \left(\Phi\sqrt{\mathrm{vol}(V)\Lambda^{\dagger}}\right)_{i,1:n}$ for the commute-time distance.

Commute-Time Distance & Resistance Distance . . .

- The above formula also allows us to interpret $\sqrt{r_{ij}}$ and $\sqrt{c_{ij}}$ as Euclidean distances on $V(G)$, i.e., to *embed* $v_i \in V(G)$ on a point $\boldsymbol{z}_i \in \mathbb{R}^n$.
- Let $L = \Phi\Lambda\Phi^{\mathsf{T}}$ be the eigendecomposition of L. Then, $L^{\dagger} = \Phi\Lambda^{\dagger}\Phi^{\mathsf{T}}$ where $\Lambda^{\dagger} = \operatorname{diag}\left(\lambda_0^{\dagger}, \ldots, \lambda_{n-1}^{\dagger}\right)$ is defined as

$$\lambda_i^{\dagger} := \begin{cases} 1/\lambda_i & \text{if } \lambda_i \neq 0; \\ 0 & \text{otherwise.} \end{cases}$$

- Hence, $\boldsymbol{z}_i^{\mathsf{T}} = \left(\Phi\sqrt{\Lambda^{\dagger}}\right)_{i,1:n}$ for the resistance distance and $\boldsymbol{z}_i^{\mathsf{T}} = \left(\Phi\sqrt{\mathrm{vol}(V)\Lambda^{\dagger}}\right)_{i,1:n}$ for the commute-time distance.

Commute-Time Distance & Resistance Distance . . .

- Compare these embeddings with the one used in the spectral clustering (with L) in Lecture 7, i.e., $\boldsymbol{y}_i^{\mathsf{T}} = \Phi_{i,1:k}$, $k \le n$.
- $\{\boldsymbol{z}_i\}$ and $\{\boldsymbol{y}_i\}$ could be considerably different. For example, in the optimal case where G consists of k disconnected components, the first k eigenvalues of L are zeros and the corresponding columns of Φ are the indicator vectors of these k components. However, the first k columns of the matrix $\Phi\sqrt{\Lambda^\dagger}$ are zero vectors.
- On the other hand, if G is connected, $\boldsymbol{z}_i$'s got more influenced by the eigenvectors corresponding to the small eigenvalues because $\lambda_i^\dagger = 1/\lambda_i$ if $\lambda_i \neq 0$.

Commute-Time Distance & Resistance Distance . . .

- Compare these embeddings with the one used in the spectral clustering (with L) in Lecture 7, i.e., $\boldsymbol{y}_i^\mathsf{T} = \Phi_{i,1:k}$, $k \le n$.
- $\{\boldsymbol{z}_i\}$ and $\{\boldsymbol{y}_i\}$ could be considerably different. For example, in the optimal case where G consists of k disconnected components, the first k eigenvalues of L are zeros and the corresponding columns of Φ are the indicator vectors of these k components. However, the first k columns of the matrix $\Phi\sqrt{\Lambda^\dagger}$ are zero vectors.
- On the other hand, if G is connected, $\boldsymbol{z}_i$'s got more influenced by the eigenvectors corresponding to the small eigenvalues because $\lambda_i^\dagger = 1/\lambda_i$ if $\lambda_i \neq 0$.

Commute-Time Distance & Resistance Distance . . .

- Compare these embeddings with the one used in the spectral clustering (with L) in Lecture 7, i.e., $\boldsymbol{y}_i^{\mathsf{T}} = \Phi_{i,1:k}$, $k \le n$.
- $\{\boldsymbol{z}_i\}$ and $\{\boldsymbol{y}_i\}$ could be considerably different. For example, in the optimal case where G consists of k disconnected components, the first k eigenvalues of L are zeros and the corresponding columns of Φ are the indicator vectors of these k components. However, the first k columns of the matrix $\Phi\sqrt{\Lambda^\dagger}$ are zero vectors.
- On the other hand, if G is connected, $\boldsymbol{z}_i$'s got more influenced by the eigenvectors corresponding to the small eigenvalues because $\lambda_i^\dagger = 1/\lambda_i$ if $\lambda_i \neq 0$.