Graphs as metric spaces

Let G be a connected graph. We now shall view graphs as metric spaces. Define the distance between two vertices $v \neq w$ as follows:

$$d_G(v,w) := \inf \{ w(P) : P \text{ is a path from } v \text{ to } w, \text{ where } w \text{ - denotes the length of the path } P \}.$$

Set, $d_G(v,v) = 0 \ \forall v \in V$, $d_G(v,w) = d_G(w,v)$.

EXR: Show that (V, d_G) is a metric space.

§ Adjacency Matrices: Graphs and Matrices

Suppose $G = (V, E)$ is a simple graph where $|V| = n$.

Suppose that the vertices of G are listed arbitrarily as $v_1, v_2, \ldots, v_n$. The adjacency matrix A of G, with respect to this listing of vertices, is the "$n \times n$" zero-one matrix with 1 as its (i,j)th entry when v_i & v_j are adjacent, and 0 as its (i,j)th entry when v_i and v_j are not adjacent.

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G \\ 0 & \text{, otherwise.} \end{cases}$$

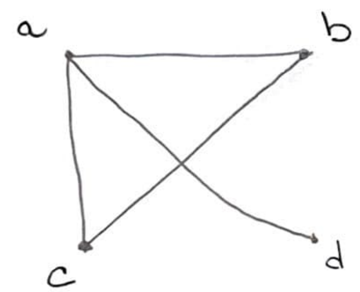
Qn: How many different adjacency Matrix G (within-vertices) can have?

1. The adjacency Matrix of a graph is based on the ordering chosen of the vertices. Hence there are $\underline{n!}$.

2. The adjacency matrix of a simple graph is symmetric, $a_{ij} = a_{ji}$,

- An symmetric matrix & hence has Real eigen-values.

Example: Use an adjacency matrix to represent the following graph.



- We order: $\underset{v_1}{a}\ \underset{v_2}{b}\ \underset{v_3}{c}\ \underset{v_4}{d}$.

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Example: Draw a graph with the adjacency Matrix:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

with respect to the ordering of vertices. a, b, c, d.

For any Matrix $A_{n\times n}$, $(A^k)_{ij} = \sum_{i_1,\dots,i_{k-1}} a_{i,i_1} a_{i_1,i_2} \cdots a_{i_{k-1},j}$

for every integer $k \geq 0$ and the sum runs over all sequences $i_1, \dots, i_{k-1}$ with each $i_r \in \{1, \dots, n\}$.

Pf. • $k = 0$, $A^0 = I \Rightarrow (A^0)_{ij} = \delta_{ij}$

• $k=1$, $A^1 = A$, so $(A^1)_{ij} = a_{ij}$.

Inductive step: Assume the formula holds for some $k \geq 1$.
We will prove it for $k+1$

$$A^{k+1} = \underline{A^k A}$$

$$(A^{k+1})_{ij} = \sum_{m=1}^{n} (A^k)_{im} a_{mj}$$

Apply inductive hypothesis to $(A^k)_{im}$

$$(A^k)_{im} = \sum_{i_1,\dots,i_{k-1}} a_{i,i_1} a_{i_1,i_2} \cdots a_{i_{k-1},m}$$

$$(A^{k+1})_{ij} = \sum_{m=1}^{n} \sum_{i_1,\dots,i_{k-1}} a_{i,i_1} a_{i_1,i_2} \cdots a_{i_{k-1}m}\, a_{mj}$$

Combine the sums: rename the indices $i_1, \dots, i_{k-1}, m$, which we may rename as $i_1, \dots, i_k$. Thus we have

$$(A^{k+1})_{ij} = \sum_{i_1,\dots,i_{k-1}} a_{i,i_1} a_{i_2,i_2} \cdots a_{i_{k-1},i_k} \cdot a_{i_k j}.$$

Lemma: Let G be a graph on n-vertices and A be its adjacency matrix. Show that $A^l(i,j)$ is the number of walks of length l from i to j.

Pf. By defn, $A^{l}(i,j) = \sum_{i_1,\dots,i_{l-1}} A_{i,i_1} A_{i_1,i_2} \cdots A_{i_{l-1},j}$

and since A is a 0-1 valued matrix, we get that

$$A_{i,i_1} A_{i_1,i_2} \cdots A_{i_{l-1},j} \in \{0,1\}.$$

Note that $A_{i,i_1} A_{i_1,i_2} \cdots A_{i_{l-1},j} = 1$ if $\underbrace{i \sim i_1 \sim i_2 \cdots i_{l-1} \sim j}_{\text{i.e } i, i_1 \dots i_{l-1}, j \text{ is a walk of length } l}$

Lemma: Let G be a connected graph on n-vertex. If $d(i,j) = m$, then $I, A, \dots, A^m$ are linearly independent.

Pf: Assume $i \neq j$. Since there are no path from i to j of length less than m.

$$A^{k}_{ij} = 0 \quad \text{for all } k < m$$

and $A^{m}_{ij} > 0$. Thus, if $I, A, \dots, A^m$ are linearly dependent with co-efficients $c_0, \dots, c_m$, then by the above observation & positivity assumption of entries of A^k for all k, we have that $c_m = 0$.

$\Rightarrow I, A, \dots, A^{m-1}$ are L.dep.

Since $d(i,j) = m$, there exists j' s.t $d(i,j') = m-1$

Now apply pre. arg. $\Rightarrow c_{m-1} = 0$.

§ Properties of Trees:

Theorem: (Characterization of trees)

Let $G = (V, E)$ be a graph with n vertices and m edges. The following statements are equivalent:

1. G is a tree.
2. There is a unique path between any two vertices in G.
3. G is connected but $G \setminus \{e\}$ is disconnected for every edge e of G.
4. G is connected, & $m = n-1$.
5. G is acyclic, and $m = n-1$.
6. G is acyclic, but $G + xy$ is cyclic for every $x, y \in V$ with $xy \notin E$

Pf. (1) ⇒ (2)
Since G is connected, there is atleast one path between any two vertices in G. So, assume that there are atleast two paths between some pair of vertices, say between x & y. Let P_1 & P_2 be two distinct paths from x to y. Then $P_1 \cup P_2$ contains a cycle. →←.

(2) ⇒ (3) Since, any two vertices in G are connected by a unique path. Let xy be any edge in E. Then,

$P = xy$ is a path from x to y. So it must be a unique path from x to y. If we remove xy from G, then there is no path from x to y. Hence $G \setminus xy$ is disconnected.

(3) ⇒ (4): Already done

(4) ⇒ (5): We have to show that every connected graph G with n-vertices, $n-1$ edges are acylic. (Already done)!

(5) ⇒ (6) Suppose that G is acyclic and that $m = n-1$. Let G_i, $1 \le i \le k$ be the connected components of G.

Since G is acyclic, $\Rightarrow$ G_i is acyclic for $1 \le i \le k$. Hence, each G_i, $1 \le i \le k$ is a tree. Let n_i & m_i, $1 \le i \le k$, be the number of vertices & edges G_i. $\Rightarrow$ $m_i = n_i - 1$.

Therefore, $m = \sum_{i=1}^{k} m_i = \sum_{i=1}^{k} (n_i - 1) = n - k$

$\Rightarrow$ $\underset{\uparrow \atop \text{hypoth}}{n-1 = m} = n - k$ $\Rightarrow$ $\underline{k = 1}$. $\Rightarrow$ Number of conn. component is 1.

$\Rightarrow$ G must be connected, hence G is a tree. Any two vertices in G are connected by a unique path. Thus adding any edge to G creates a cycle.

(6) $\Rightarrow$ (1): Suppose that G is acyclic but xy is cyclic for every x, y in V with $xy \notin E$. We must show that G is connected. Let u & v be arbitrary vertices in G. If u & v are not already adjacent, adding the edge uv creates a cycle in which all edges but uv belong to G. Thus, there is a path from u to v & since u & v were chosen arbitrarily, G is connected.

<u>Spanning tree</u>: A subgraph $T = (V_1, E_1)$ of a Graph $G = (V, E)$ is a spanning tree if

(i) T is a tree &

(ii) $V_1 = V$

<u>Theorem</u>: A graph admits a spanning tree iff G is connected.

<u>Pf</u>: Suppose G admits a spanning tree, say T. We will show that G is connected. Let u, and v be any two arbitrary vertices of G. Since T is a spanning subgraph of G, u & v are vertices of T as well. As T is connected, there is a

path $P(u,v)$ from u to v in T. As T is a subgraph of G, $P(u,v)$ is also a path in G. Hence G is connected.

Let G be connected graph with $\underline{n}$-vertices and $\underline{m}$-edges.
We construct a spanning tree in G. Let $k = (m-n)+1$
Define G_i, $0 \le i \le k$, recursively, as follows:

$$G_i = \begin{cases} G & \text{if } i = 0 \\ G_{i-1} - e_i, & \text{where } e_i \text{ is an edge in some cycle of } G_{i-1}, \text{ if } 1 \le i \le k. \end{cases}$$

Since, G_i has exactly $m-i \, (= (n-1)+k-i)$ edges, G_i is cyclic for each i, $0 \le i \le \underline{k-1}$. So, each G_i, $0 \le i \le k-1$, has a cycle. If G_{i-1} is connected, then G_i is also connected, as e_i belongs to some cycle of G_{i-1}, $0 \le i \le k-1$. Hence G_K is connected & has exactly $n-1$ edges. So, G_K is a tree. Let $T = G_K$. Now T is a spanning tree of G.

$\underline{\underline{\text{Qn:}}}$ Number of distinct spanning trees of a complete Graph?

$\underline{\text{Thm:}}$ $\underline{\tau(K_n) = n^{n-2}}$ for $n \ge 1$.

$\underline{\underline{\text{Pf:}}}$ K_n - Complete graph (every pair of distinct vertices is connected by a edge).

K_n has exactly $\binom{n}{2}$ edges.

Each K_1, K_2 has exactly one Spanning tree

$\Rightarrow$ $\tau(K_n) = n^{n-2}$ for $n=1$ & $n=2$

So assume that $n \geq 3$. Assume that $V(K_n) = \{1, 2, \ldots, n\}$.

Let X be the set of all spanning trees of K_n

& Y be the sequences $a_1, a_2, \ldots, a_{n-2}$ of length $n-2$, such that $a_i \in \{1, 2, \ldots, n\}$.

$\Rightarrow$ Y has n^{n-2} sequences.

To show that there are n^{n-2} spanning trees of K_n, it is enough to produce a function

$f : X \to Y$ which is a bijection.

Let T be any spanning tree of K_n. Define (Prufer Code)

$f(T) = a_1, a_2, \ldots, a_{n-2}$, a unique seq. of length $n-2$ s.t $a_i \in \{1, 2, \ldots, n\}$ for each i, $1 \leq i \leq n-2$, in the following way. Among all the vertices of degree one, let s_1 be the vertex s.t s_1 as an integer is minimum

Let t_1 be the vertex adjacent to s_1 in T. Assign t_1 to a_1.

Then, delete the vertex s_1 from T.

Next, among all the vertices of degree 1 in $T - \{s_1\}$,

Let s_2 be the vertex s.t. s_2 as an integer is minimum.

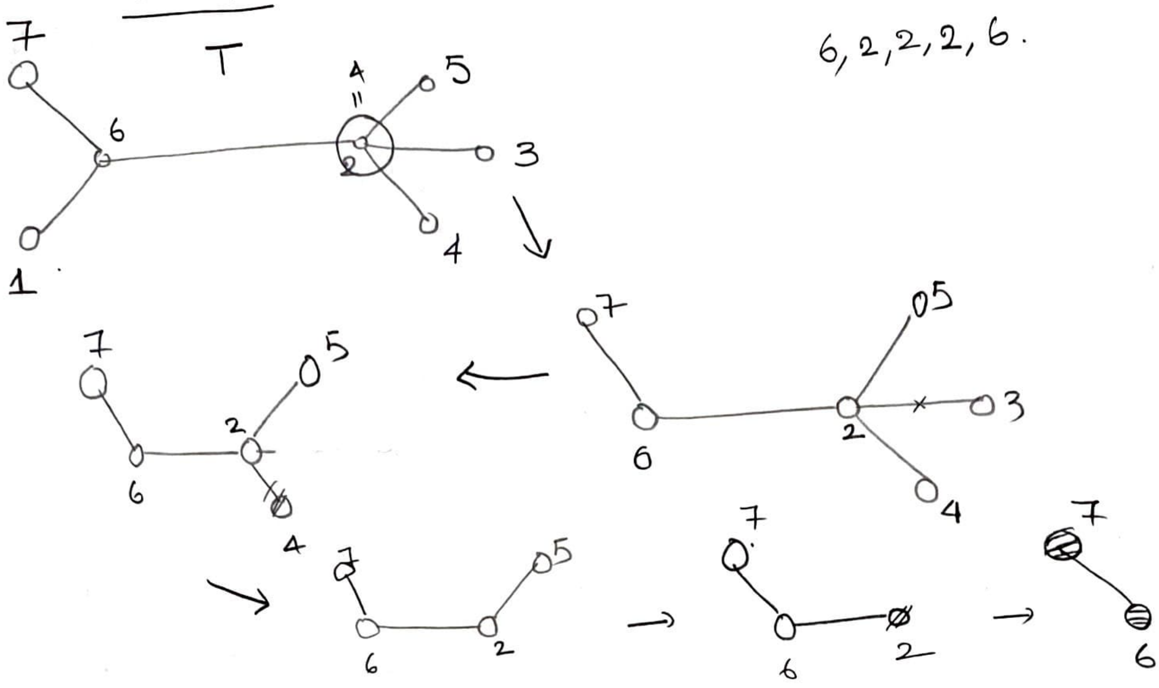
Let t_2 be the vertex adjacent to s_2 in $T - \{s_1\}$.

Assign t_2 to a_2. Then delete the vertex s_2 from $T - \{s_1\}$. Repeat this process until a_{n-2} has been defined & a tree with just two vertices remains

$f^2 = f^2$

example

6, 2, 2, 2, 6.



$f(T) := 6, 2, 2, 2, 6$

To show that f is a bijection, we have to prove that

{ (i) no seq. is produced by two different spanning trees of K_n.
(ii) every seq of Y is produced by some spanning tree of K_n.

We shall achieve this by showing f has an inverse.

i.e, we can construct a spanning tree of K_n from a seq. $a_1, a_2, \ldots a_{n-2}$ of Y.

Let T be any spanning tree of K_n, and let $f(T) = a_1, a_2, \ldots, a_{n-2}$.

Then $\deg(k) =$ number of times k appears $+ 1$.
vertex k

So, let $f(T) = a_1, a_2, \ldots, a_{n-2}$. We construct T as follows:

Let s_1 be the vertex s.t s_1 is the least integer in $\{1, 2, \ldots, n\}$ that does not appear in the ~~vertex~~ $a_1, a_2 \ldots a_{n-2}$.

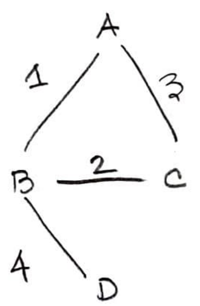
Join $s_1 \to a_1$. Then, let s_2 be the vertex s.t s_2 is the least integer in $\{1, 2, \ldots, n\} \setminus \{s_1\}$ that does not appear in the seq. $a_2, \ldots, a_{n-2}$. Join s_2 to a_2. Follow this procedure until s_{n-2} is obtained from the seq a_{n-2}. Join s_{n-2} to a_{n-2}. The T (tree) is obtained by adding the two remaining ver. in $N - \{s_1, \ldots s_{n-2}\}$.

Minimum Spanning Tree:

Let $G = (V, E)$ be a connected weighted graph and C be the cost matrix of G. Let $T = (V, E)$ be a spanning tree of G. The cost of T, denoted by $c(T)$, is defined as follows:

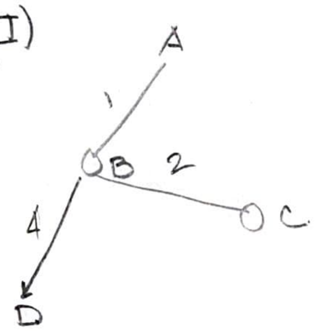
Defn: The cost of a spanning tree $T = (V, E')$ of a weighted graph $G = (V, E)$ with cost matrix C is defined by: $c(T) = \sum_{e \in E'} c(e)$

Defn: A spanning tree T of a weighted connected graph is called a minimum spanning tree (MST) if $c(T) \le c(T')$ for any other spanning tree.



Possible Spanning tree:

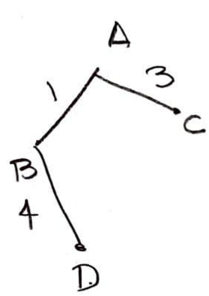
(I)



Weight $= 1+2+4 = 7$

$c(T_I) = 7$

(II)



W = 8

$c(T_{II}) = 8$.

Qn: How to find MST?

§ Minimum Spanning tree Algorithm:

We will discuss two popular algorithms to construct minimum spanning tree of a weighted connected graph.

1) Kruskal's Algorithm (Greedy) 2) Prim's Algorithm.

We first discuss Kruskal's Formally: The algorithm arranges the edges of the graph G in the non-decreasing order of their costs. Starts with the graph $T = (V, E')$, where $E' = \phi$ initially. It then examines each edge for inclusion into T. If the current edge "e" under examination does-not form a cycle with the so far selected edges, the edge "e" is selected.

Kruskal's algorithm: Input Graph (weighted) $G = (V, E, w)$.

Step-1: Initialize with $D = \phi \subset E$ & $M = (V(G), \phi)$

Step-2: Select one of the smallest (in terms of weight) edges in $E \setminus D$ call it e.

Step-3: $M \cup e$ does-not create a cycle, set $M = M \cup e$

Step-4: Set $D = D \cup e$. If $|E(M)| < n-1$ & $D \neq E$, go to step-2 else to Step 5

Step-5: Output M.

Prim's (Prim's-Dykstra-Janik's Algorithm): Input graph weighted connected graph $G = (V, E, w)$.

Step-1: $M = D = \phi$ & $S = \{v\}$, $T = V - S$ for some $v \in V$.

Prim-Dijkstra-Jarnik's algorithm:

Input graph weighted connected Graph $G = (V, E, w)$.

Step 1: Initialize with $M = D = \phi$ & $S = \{v\}$, $T = V - S$ for some $v \in V$.

It uses the fact that a connected graph with n-vertices and (n-1) edges is a tree. It starts with vertex set $V' = \{v\}$ where v is any arbitrary vertex of G & E', where $E' = \phi$. It then selects a least cost edge $e = xy$ with $x \in V'$ & $y \in V \setminus V'$ & updates $E' = E' \cup \{e\}$ and $V' = V' \cup \{y\}$. It stops when $V' = V$. Thus it maintains through that $G' = (V', E')$ is connected. Once $V' = V$, G' becomes a spanning tree of G.

Theorem: Prim's Algorithm produces a minimum spanning tree in a connected weighted graph.

Pf. Let G be a connected weighted graph and let T be a subgraph produced by Prim's algorithm. Since, G is connected, T is a spanning tree of G. Next, we show that T is a minimum spanning tree of G. Suppose, to the contrary, that T is not a ~~min~~ MST of G. Let $E(T) = \{f_1, f_2, \cdots, f_{n-1}\}$ s.t $w(f_i) \leq w(f_{i+1})$ $1 \leq i \leq n-2$.

Note that G may have more than one minimum spanning tree.

Let T_1 be a MST of G having maximum number of edges in common with T. Let i be the smallest index $1 \leq i \leq n-1$ s.t f_i is not an edge of T_1. For $i=1$, let $U = \{u\}$, where u is the first vertex added to V' by the Prim's algorithm. If $i \geq 2$, then let U be the vertex set of the subgraph induced by the edges $f_1, f_2, \cdots, f_{i-1}$. Now, f_i joins a vertex of U to a vertex of $V-U$. Let $T_2 = T_1 + f_i$, Now T_2 has a unique cycle C containing f_i. The cycle C contains an edge e_0 that joins a vertex of U to a vertex of $V-U$. Let $T_3 = T_1 + f_i - e_0$. Then T_3 is a spanning tree of G. Since f_i & e_0 are both edges from U to $V \setminus U$, & f_i is selected by Prim's algo, $w(f_i) \leq w(e_0)$
$\Rightarrow w(T_3) \leq w(T_1)$. But T_3 has more edge com. to T_1 ⨳