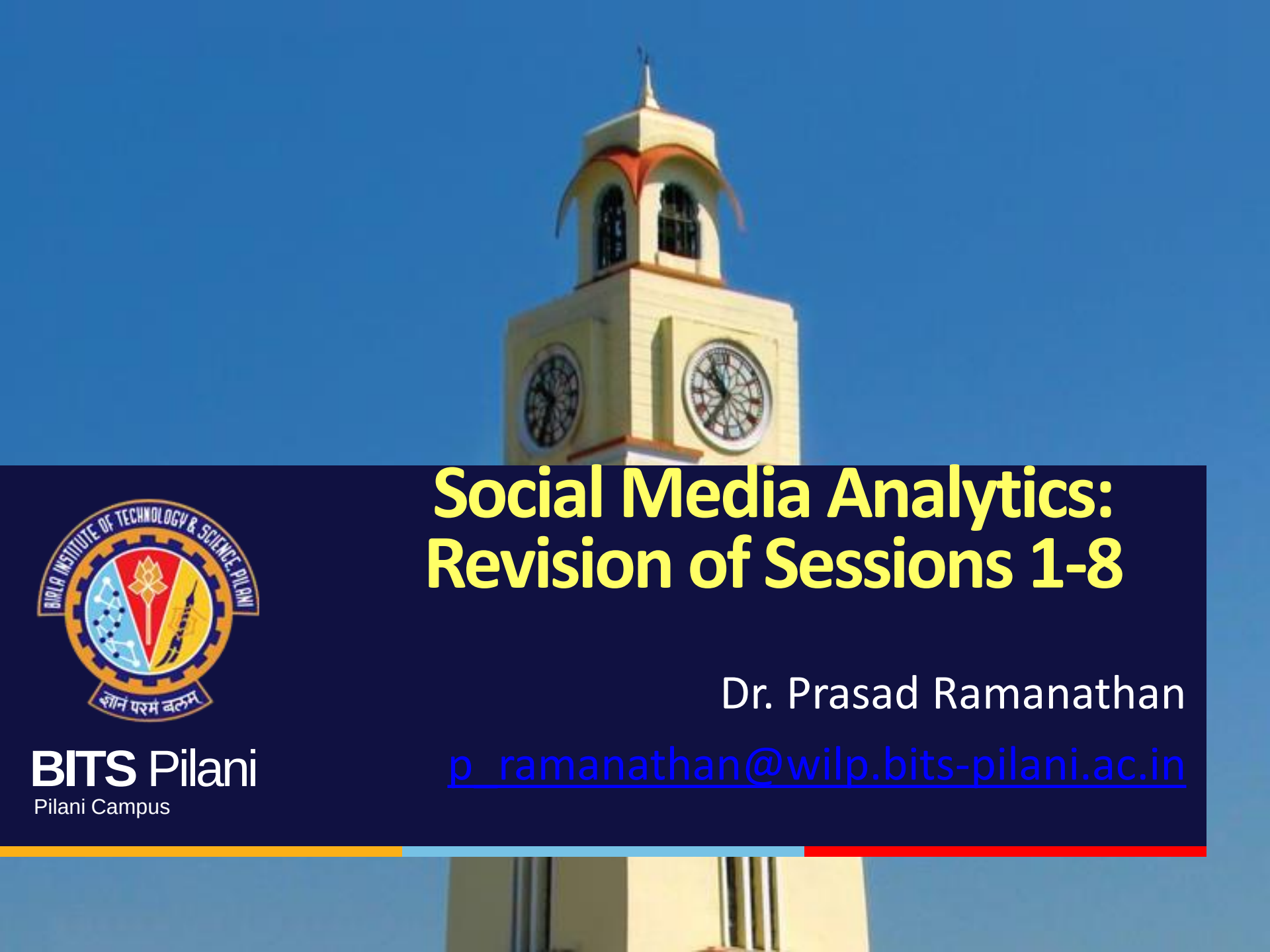


Social Media Analytics: Revision of Sessions 1-8

Dr. Prasad Ramanathan

p_ramanathan@wilp.bits-pilani.ac.in

BITS Pilani
Pilani Campus

Acknowledgment

Grateful acknowledgment to slides and course material provided by:

R. Zafarani, M. A. Abbasi, and H. Liu, *Social Media Mining: An Introduction*, Cambridge University Press, 2014.

Free book and slides at **http://socialmediamining.info/**

Social Media: Main Characteristics

innovate achieve lead

- Participation
  - social media encourages contributions and feedback from everyone who is interested. It blurs the line between broadcaster and audience.
- Openness
  - most social media services are open to feedback and participation. They encourage voting, comments and the sharing of information. There are rarely any barriers to accessing and making use of content – password-protected content is frowned on.
- Conversation
  - whereas traditional media is about “broadcast” (content transmitted or distributed to an audience) social media is better seen as a two-way conversation.
- Community
  - social media allows communities to form quickly and communicate effectively. Communities share common interests, such as a love of photography, a political issue or a favorite TV show.
- Connectedness
  - Most kinds of social media thrive on their connectedness, making use of links to other sites, resources and people.

Source: Social Media Mining, Zafarani, Abbasi & Liu

Challenges

innovate achieve lead

1. **Big Data Paradox**
   - Social media data is big, yet not evenly distributed.
   - Often little data is available for an individual

2. **Obtaining Sufficient Samples**
   - Are our samples reliable representatives of the full data?

3. **Noise Removal Fallacy**
   - Too much removal makes data more sparse
   - Noise definition is relative and complicated and is task-dependent

4. **Evaluation Dilemma**
   - When there is no ground truth, how can you evaluate?

Source: Social Media Mining, Zafarani, Abbasi & Liu

innovate achieve lead

BITS Pilani

Pilani Campus

2. Sentiment Analysis

Reference: Liu

A more practical definition

(Hu and Liu 2004; Liu, 2010, 2012)

- An *opinion* is a quintuple

  (*entity*, *aspect*, *sentiment*, *holder*, *time*)

  where
  - ***entity:*** target entity (or object).
  - ***Aspect***: aspect (or feature) of the entity.
  - ***Sentiment***: +, -, or neu, a rating, or an emotion.
  - ***holder***: opinion holder.
  - ***time***: time when the opinion was expressed.
- *Aspect-based sentiment analysis*

Opinion summary

(Hu and Liu, 2004)

Aspect/feature Based Summary of opinions about iPhone:

Aspect: **Touch screen**

Positive: 212

- *The touch screen was really cool.*
- *The touch screen was so easy to use and can do amazing things.*

…

Negative: 6

- The screen is easily scratched.
- I have a lot of difficulty in removing finger marks from the touch screen.

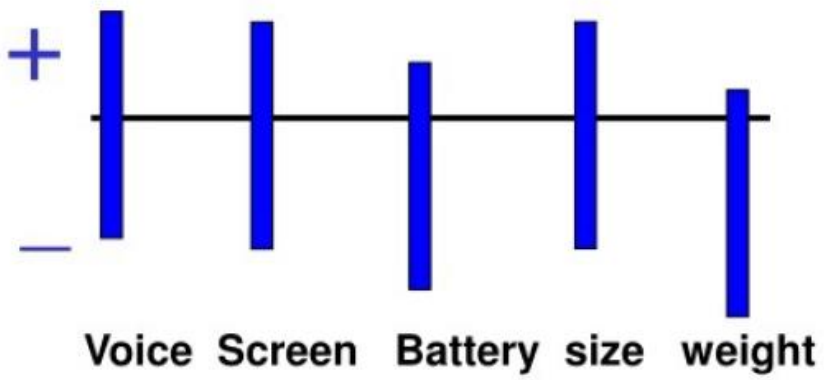
…

Aspect: **voice quality**

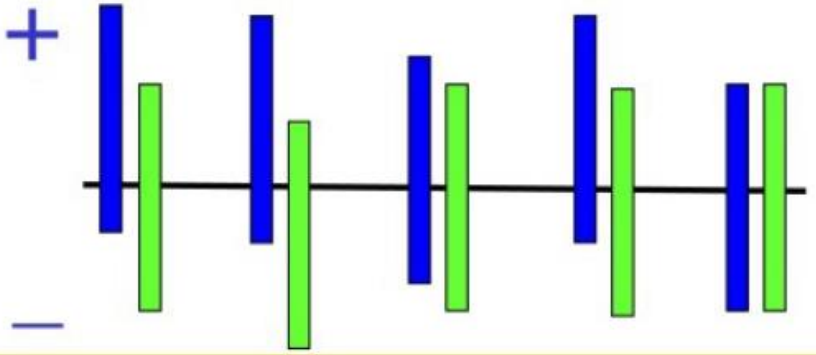
…

(Liu et al. 2005)

- Opinion Summary of 1 phone



- Opinion comparison of 2 phones



Sentiment classification

- Classify a whole opinion document (e.g., a review) based on the overall sentiment of the opinion holder (Pang et al 2002; Turney 2002)
  - Classes: Positive, negative (possibly neutral)
- An example review:
  - *"I bought an iPhone a few days ago. It is such a nice phone, although a little large. The touch screen is cool. The voice quality is great too. I simply love it!"*
  - Classification: positive or negative?
- It is basically a text classification problem

Supervised learning (Pang et al, 2002)

- Directly apply supervised learning techniques to classify reviews into positive and negative.
- Three classification techniques were tried:
  - Naïve Bayes, Maximum Entropy, Support Vector Machines (SVM)
- Features: negation tag, unigram (single words), bigram, POS tag, position.
- SVM did the best based on movie reviews.

Sentence sentiment analysis

- Usually consist of two steps
  - Subjectivity classification (Wiebe et al 1999)
    - To identify subjective sentences
  - Sentiment classification of subjective sentences
    - Into two classes, positive and negative
- But bear in mind
  - Many objective sentences can imply sentiments
  - Many subjective sentences do not express positive or negative sentiments/opinions
    - E.g., "I believe he went home yesterday."

Aspect extraction

- Goal: Given an opinion corpus, extract all aspects

- Four main approaches:
  - (1) Finding frequent nouns and noun phrases
  - (2) Exploiting opinion and target relations
  - (3) Supervised learning
  - (4) Topic modeling

Aspect sentiment classification

"*Apple is doing very well in this poor economy*"

- Lexicon-based approach: Opinion words/phrases
  - Parsing: simple sentences, compound sentences, conditional sentences, questions, modality verb tenses, etc (Hu and Liu, 2004; Ding et al. 2008; Narayanan et al. 2009).
- Supervised learning is tricky:
  - Feature weighting: consider distance between word and target entity/aspect (e.g., Boiy and Moens, 2009)
  - Use a parse tree to generate a set of target dependent features (e.g., Jiang et al. 2011)

Sentiment lexicon

- Sentiment words or phrases (also called polar words, opinion bearing words, etc). E.g.,
  - Positive: beautiful, wonderful, good, amazing,
  - Negative: bad, poor, terrible, cost an arm and a leg.
- Many of them are context dependent, not just application domain dependent.
- Three main ways to compile such lists:
  - Manual approach: not a bad idea, only an one-time effort
  - Corpus-based approach
  - Dictionary-based approach

Which approach to use?

- Both corpus and dictionary based approaches are needed.
- Dictionary usually does not give domain or context dependent meaning
  - Corpus is needed for that
- Corpus-based approach is hard to find a very large set of opinion words
  - Dictionary is good for that
- In practice, corpus, dictionary and manual approaches are all needed.

3. Information Extraction

Munindar Singh, NCSU Course on NLP, Fall 2021

Information Extraction

Named entity recognition (NER) seeks to

- Identify where each named entity is mentioned
- Identify its type: person, place, organization, . . .
- Unify distinct names for the same entity
- United = United Airlines

Foundational step for virtually any kind of advanced reasoning

- Extracting relations as to build knowledge graphs
- Extracting events
- Answering questions

Source: Munindar Singh

Named Entity Recognition

- Entities that can be named
  - For news: Person, location, organization
  - For medicine: drugs, . . .
- Even entities that aren't named, e.g., dates and numbers
- The sentence:

This Friday United is selling $100 fares to The Big Apple on their new Dreamliner

- Yields this markup:

This [*time* Friday] [*org* United] is selling [*money* $100] fares to [*loc* The Big Apple] on their new [*veh* Dreamliner]

- Challenges
  - Segmentation: what are the boundaries of an entity
  - Ambiguity: JFK can be a person, an airport, . . .
  - Exacerbated by metonymy: Washington (city, government, sports teams)

Source: Munindar Singh

IOB Tagging for NER

- Introduce $2n+1$ tags (given n types—earlier chunk, here NER)
  - B_k: Beginning of type k
  - I_k: Inside of type k
  - O: Outside of all types
- Example of IOB chunking for NER:

Woodson	,	Chancellor	of	NC	State	University
[B_{PER}]	O	[B_{PER}]	O	[B_{ORG}]	[I_{ORG}]	[I_{ORG}]

,	is	a	professor
O	O	O	O

Source: Munindar Singh

Relation Extraction

Identify and classify semantic relations between entities found in the text

- General purpose
  - Child-of: taxonomy
  - Part-whole: meronomy
  - Geospatial
- Domain specific
  - Employee of (domain of human resources)
  - Additive for (domain of chemistry)

Source: Munindar Singh

Features for Supervised Relation Extraction

- Identify *mentions* M_1 and M_2
- Important features as word embeddings
  - Headwords of M_1 and M_2
  - Concatenation of headwords of M_1 and M_2
  - Adjacent words to M_1 and M_2
  - N-grams between M_1 and M_2
- NER features
  - Types of M_1 and M_2 and their concatenation
  - Entity (constituent) level from Name, Nominal, Pronoun
  - Number of intervening entities between M_1 and M_2

Source: Munindar Singh

Extracting Temporal Expressions

- Main varieties
  - Absolute
  - Relative
  - Durational
  - How can we classify holidays, e.g., Memorial Day, Easter, Diwali?
- Often associated with lexical triggers
  - Nouns: Dusk, dawn,
  - Proper Nouns: January, Monday, Ides of March, Rosh Hashana, Ramadan
  - Adjectives: Recent, annual, former
  - Adverbs: hourly, usually
- False hits: temporal expressions used atemporally
  - 1984 (the book or movie)
  - Sunday Bloody Sunday (song by the Irish group U2)

Source: Munindar Singh

Event Extraction
How Events Link to various Entities

- Event coreference
  - Which mentions of an event refer to the same event
- Temporal expressions
  - Days, dates, times
  - Relative expressions, such as "next month"
- Normalization with respect to
  - Calendar
  - Discourse, e.g., time of utterance or reference

Source: Munindar Singh

Event Extraction: Identify Events or States from Text

- Classically, events are occurrences, not states, which are indicated by verbs such as
  - Be, is, are
  - Know, feel, believe
- In the extraction literature, events include states
  - Verbs: increased
  - Nouns: the increase
  - Gerunds: increasing
- Nonevents
  - Verbs indicating transition into an event: took effect
  - Weak or light verbs (make, take, have) that rely on a direct object to bring out an event

Source: Munindar Singh

innovate achieve lead

BITS Pilani

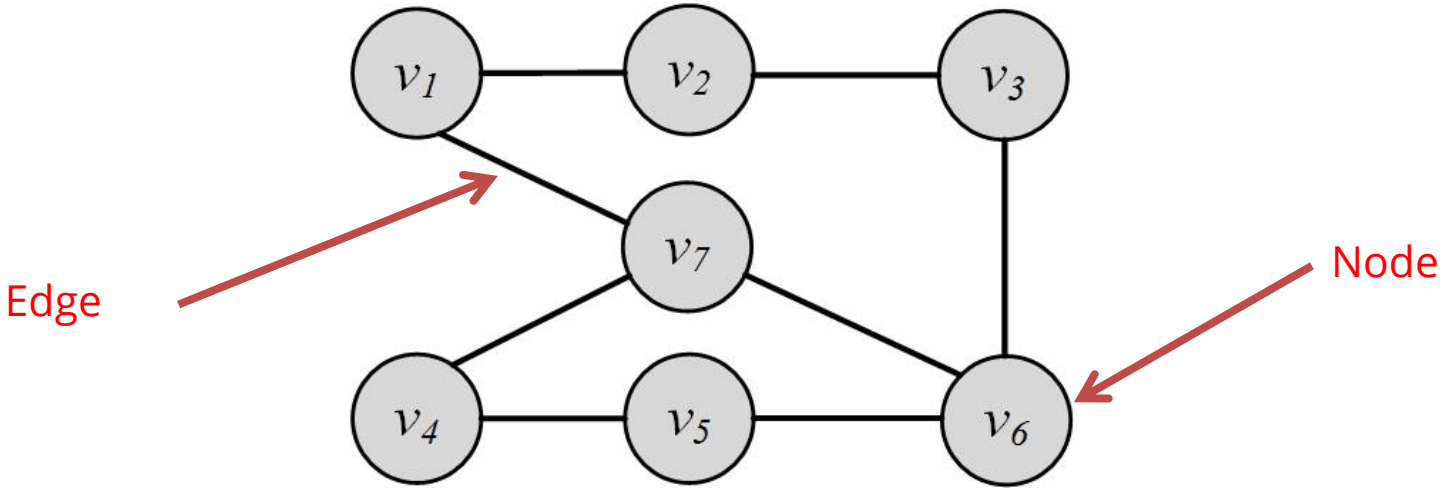
Pilani Campus

4. Graph Essentials

Nodes and Edges

A network is a graph, or a collection of points connected by lines

- Points are referred to as **nodes, actors,** or **vertices** (plural of **vertex**)
- Connections are referred to as **edges** or **ties**



Neighborhood and Degree (In-degree, out-degree)

For any node v, in an undirected graph, the set of nodes it is connected to via an edge is called its neighborhood and is represented as $N(v)$

- *In directed graphs we have incoming neighbors $N_{in}(v)$ (nodes that connect to v) and outgoing neighbors $N_{out}(v)$.*

The number of edges connected to one node is the degree of that node (the size of its neighborhood)

- Degree of a node i is usually presented using notation d_i

In Directed graphs:

d_i^{in} – In-degrees is the number of edges pointing towards a node

d_i^{out} – Out-degree is the number of edges pointing away from a node

Degree and Degree Distribution

- **Theorem 1.** The summation of degrees in an undirected graph is twice the number of edges

$$\sum_i d_i = 2|E|$$

- **Lemma 1.** The number of nodes with odd degree is even
- **Lemma 2.** In any directed graph, the summation of in-degrees is equal to the summation of out-degrees,

$$\sum_i d_i^{out} = \sum_j d_j^{in}$$

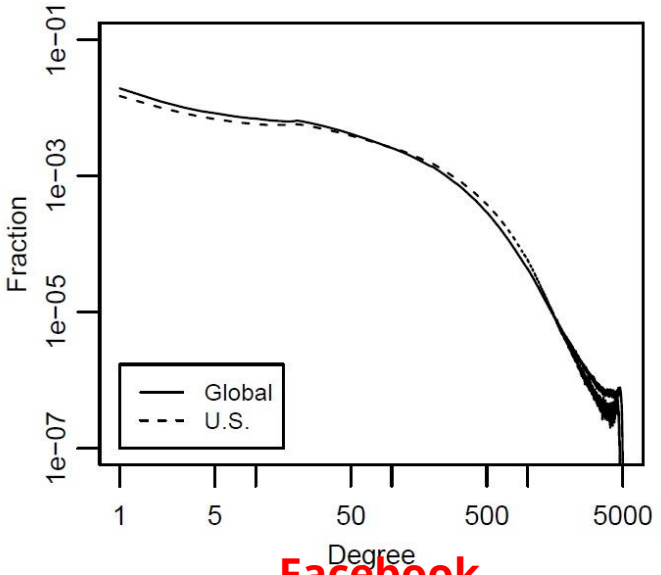
Degree Distribution Plot

The x-axis represents the degree and the y-axis represents the fraction of nodes having that degree

– On social networking sites

There exist many users with few connections and there exist a handful of users with very large numbers of friends.

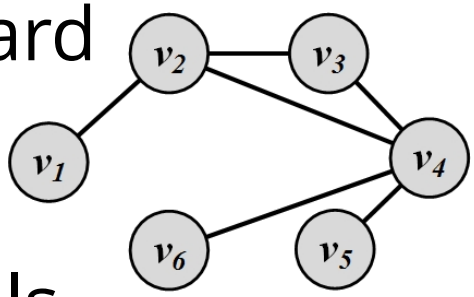
(**Power-law degree distribution)**



Facebook Degree Distribution

Graph Representation

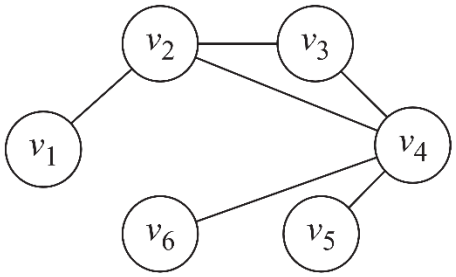
- Graph representation is straightforward and intuitive, but it cannot be effectively manipulated using mathematical and computational tools



- We are seeking representations that can store these two sets in a way such that
  - Does not lose information
  - Can be manipulated easily by computers
  - Can have mathematical methods applied easily

Adjacency Matrix (a.k.a. sociomatrix)

$$A_{ij} = \begin{cases} 1, \text{ if there is an edge between nodes } v_i \text{ and } v_j \\ 0, \text{ otherwise} \end{cases}$$



(a) Graph

	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	0	0	0	0
v_2	1	0	1	1	0	0
v_3	0	1	0	1	0	0
v_4	0	1	1	0	1	1
v_5	0	0	0	1	0	0
v_6	0	0	0	1	0	0

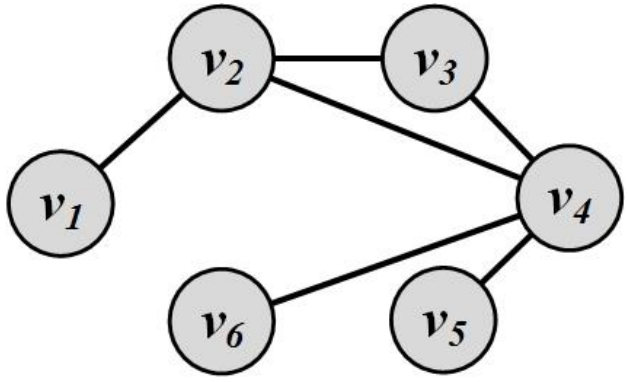
(b) Adjacency Matrix

Diagonal Entries are self-links or loops

Social media networks have very sparse Adjacency matrices

Adjacency List

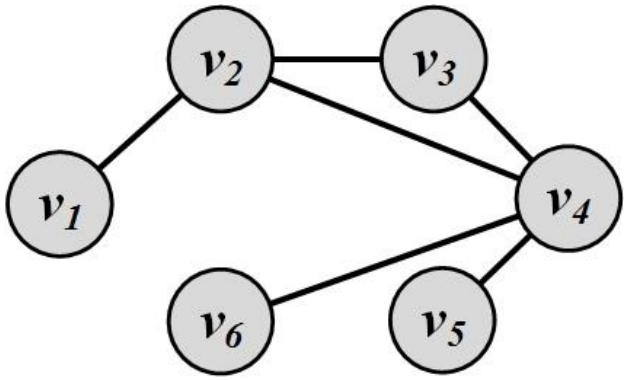
- In an adjacency list for every node, we maintain a list of all the nodes that it is connected to
- The list is usually sorted based on the node order or other preferences



Node	Connected To
v_1	v_2
v_2	v_1, v_3, v_4
v_3	v_2, v_4
v_4	v_2, v_3, v_5, v_6
v_5	v_4
v_6	v_4

Edge List

- In this representation, each element is an edge and is usually represented as (u, v), denoting that node u is connected to node v via an edge



(v_1, v_2)
(v_2, v_3)
(v_2, v_4)
(v_3, v_4)
(v_4, v_5)
(v_4, v_6)

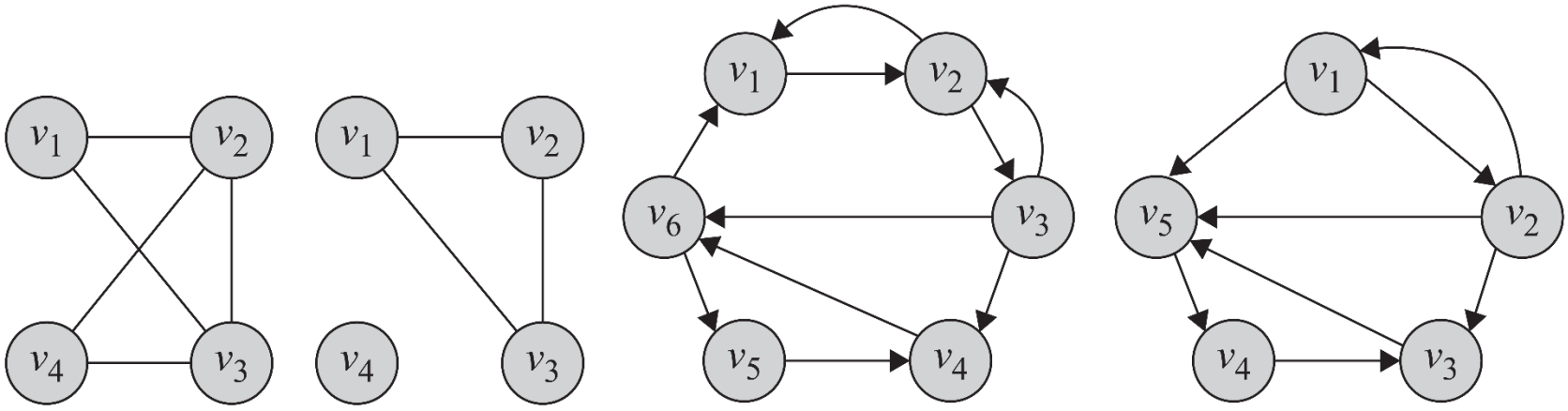
Connectivity

- **A node v_i is connected to node v_j** (or reachable from v_j) if it is adjacent to it or there exists a path from v_i to v_j.

- **A graph is connected**, if there exists a path between any pair of nodes in it
  - In a directed graph, **a graph is strongly connected** if there exists a directed path between any pair of nodes
  - In a directed graph, **a graph is weakly connected** if there exists a path between any pair of nodes, without following the edge directions

- A graph is **disconnected,** if it is not connected.

Connectivity: Example



(a) Connected (b) Disconnected (c) Strongly connected (d) Weakly connected

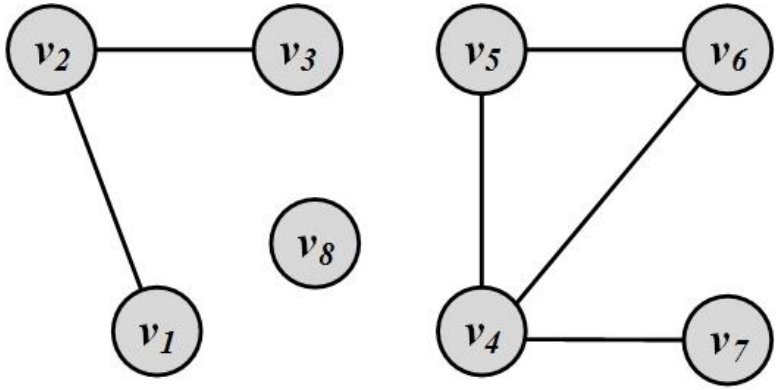
Component

- A **component** in an undirected graph is a connected **subgraph**, i.e., there is a path between every pair of nodes inside the component

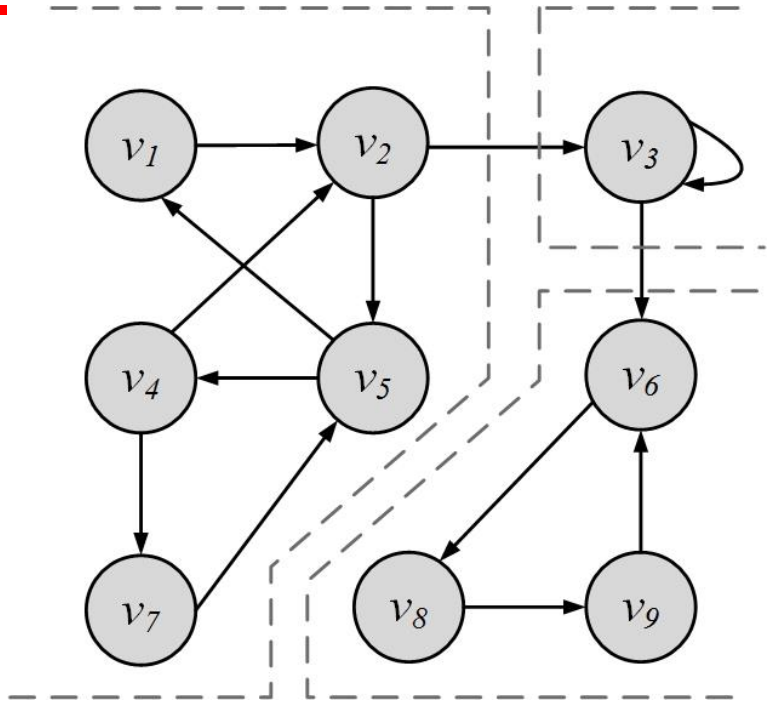
- In directed graphs, we have a **strongly connected** components when there is a path from u to v and one from v to u for every pair of nodes u and v.

- The component is **weakly connected** if replacing directed edges with undirected edges results in a connected component

Component Examples:



3 components



3 Strongly-connected components

Shortest Path

- **Shortest Path** is the path between two nodes that has the shortest length.
  - We denote the length of the shortest path between nodes v_i and v_j as $l_{i,j}$

- The concept of the neighborhood of a node can be generalized using shortest paths. An **n-hop neighborhood** of a node is the set of nodes that are within n hops distance from the node.

Diameter

The diameter of a graph is the length of the longest shortest path between any pair of nodes between any pairs of nodes in the graph

$$diameter_G = \max_{(v_i, v_j) \in V \times V} l_{i,j}$$

- How big is the diameter of the web?

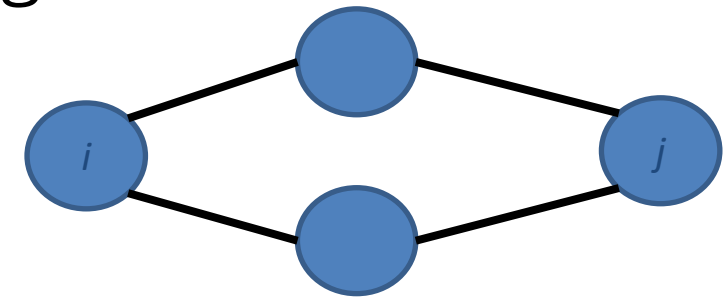
Adjacency Matrix and Connectivity

- Consider the following adjacency matrix

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ A_{d1} & A_{d2} & A_{d3} & \dots & A_{dn} \end{bmatrix}$$

- Number of Common neighbors between node i and node j

$$\sum_k A_{ik} A_{jk} = A_i \cdot A_j$$



- That's element of [ij] of matrix $A \times A^T = A^2$
- Common neighbors are paths of length 2
- Similarly, what is A^3?

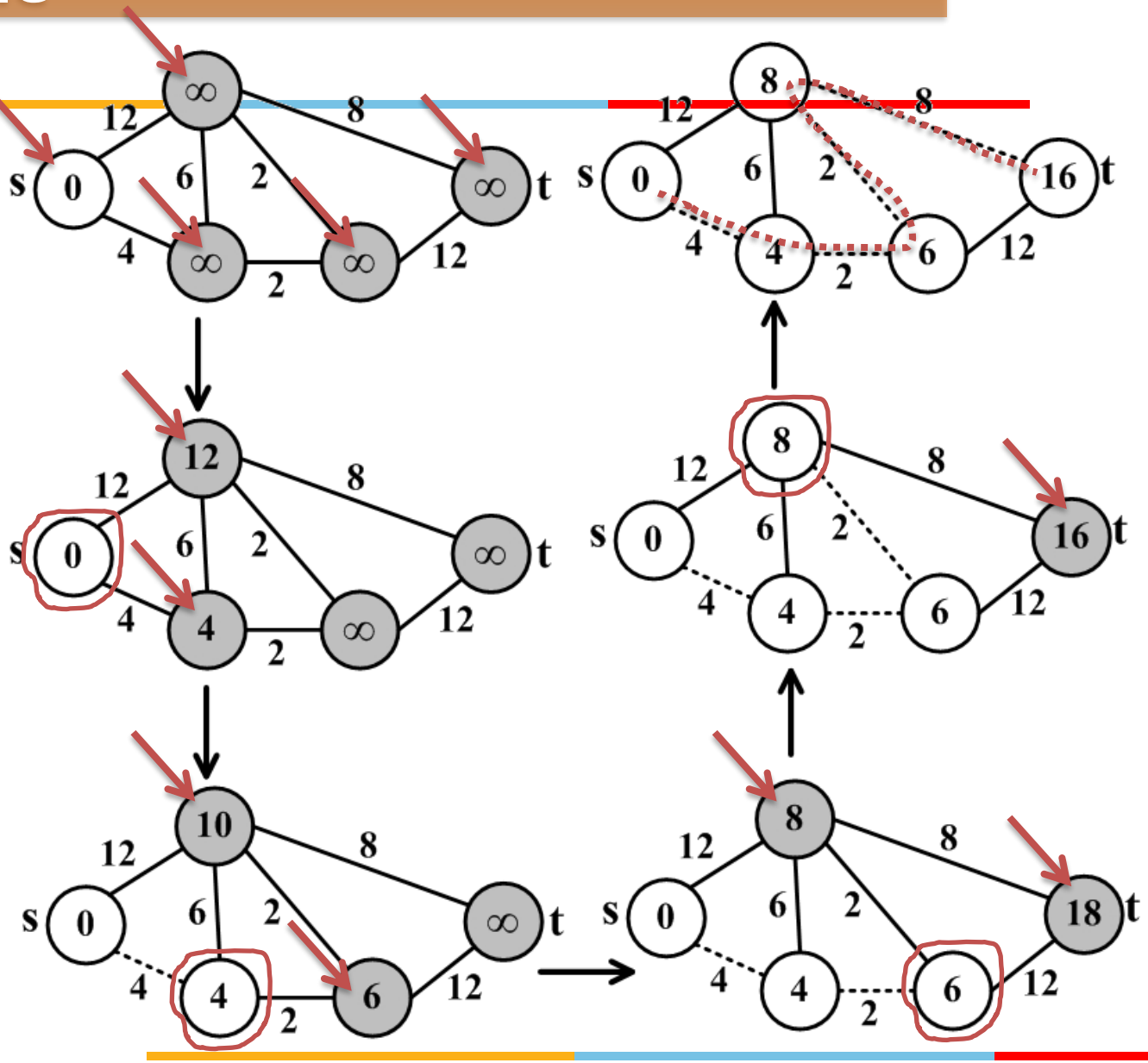
Graph/Tree Traversal

- We are interested in surveying a social media site to compute the average age of its users
  - Start from one user;
  - Employ some traversal technique to reach her friends and then friends' friends, …

- The traversal technique guarantees that
  1. All users are visited; and
  2. No user is visited more than once.

- There are two main techniques:
  - **Depth-First Search (DFS)**
  - **Breadth-First Search (BFS)**

Dijkstra's Algorithm: Execution Example



Spanning Tree

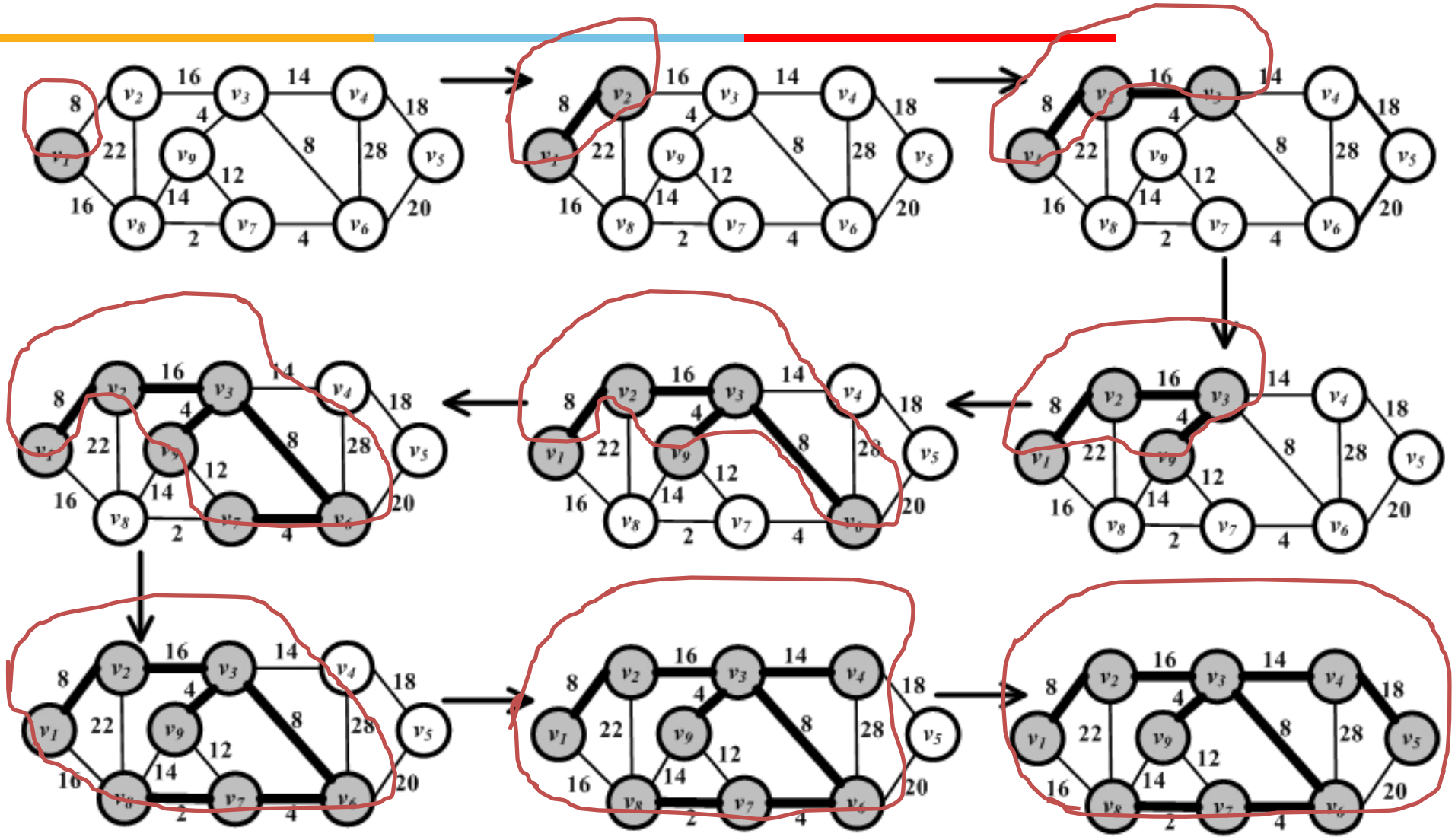
For any connected graph, the spanning tree is a subgraph and a tree that includes all the nodes of the graph. Obviously, when the original graph is not a tree, then its spanning tree includes all the nodes, but not all the edges.

There may exist multiple spanning trees for a graph. For a weighted graph and one of its spanning trees, the weight of that spanning tree is the summation of the edge weights in the tree.

Among the many spanning trees found for a weighted graph, the one with the minimum weight is called the **minimum spanning tree (MST)**

Application: Due to construction costs, the government needs to minimize the total mileage of roads built and, at the same time, needs to guarantee that there is a path (i.e., a set of roads) that connects every two cities. The minimum spanning tree is a solution to this problem.

Finding Minimum Spanning Tree: Prim's Algorithm Execution Example



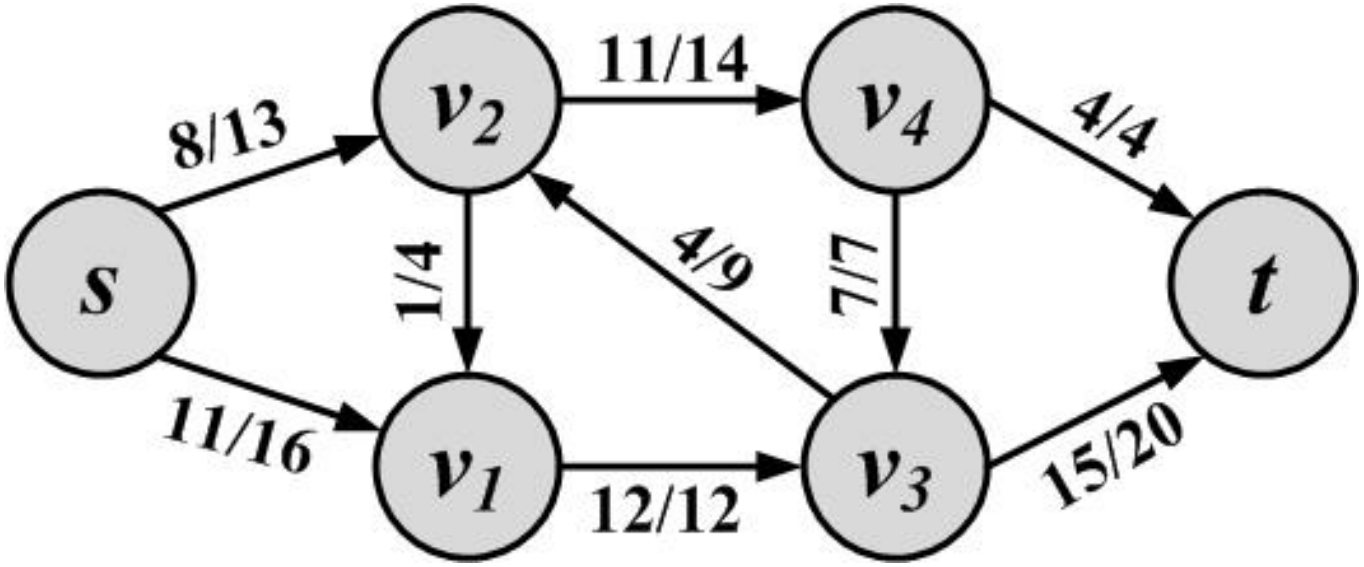
Network Flow

- Consider a network of pipes that connects an infinite water source to a water sink.
  - Given the capacity of these pipes, what is the maximum flow that can be sent from the source to the sink?

- Parallel in Social Media:
  - Users have daily cognitive/time limits (the capacity, here) of sending messages (the flow) to others,
  - What is the maximum number of messages the network should be prepared to handle at any time?

A Sample Flow Network

- Commonly, to visualize an edge with capacity c and flow f , we use the notation f/c.



Ford Fulkerson Algorithm

innovate achieve lead

BITS Pilani
Pilani Campus

5. Network Measures

Why Do We Need Measures?

- Who are the central figures (influential individuals) in the network?
  - **Centrality**
- What interaction patterns are common in friends?
  - **Reciprocity and Transitivity**
  - **Balance and Status**
- Who are the like-minded users and how can we find these similar individuals?
  - **Similarity**
- To answer these and similar questions, one first needs to define measures for quantifying **centrality**, **level of interactions**, and **similarity**, among others.

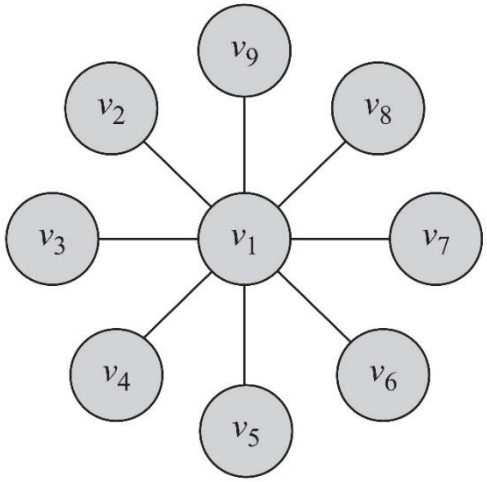
Degree Centrality

- **Degree centrality**: ranks nodes with more connections higher in terms of centrality

$$C_d(v_i) = d_i$$

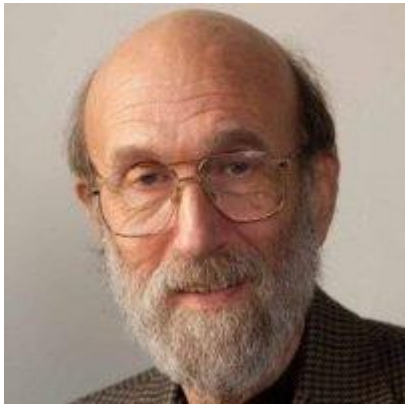
- d_i is the degree (number of friends) for node v_i
  - i.e., the number of length-1 paths (can be generalized)

In this graph, degree centrality for node v_1 is d_1=8 and for all others is $d_j = 1, j \neq 1$



Eigenvector Centrality

- Having more friends does not by itself guarantee that someone is more important
  - Having more **important friends** provides a stronger signal



Phillip Bonacich

- Eigenvector centrality generalizes degree centrality by incorporating the importance of the neighbors (undirected)

- For directed graphs, we can use incoming or outgoing edges

Katz Centrality

- A major problem with eigenvector centrality arises when it deals with directed graphs
- Centrality only passes over *outgoing* edges and in special cases such as when a node is in a directed acyclic graph centrality becomes zero
  - The node can have many edge connected to it

Elihu Katz

- To resolve this problem we add bias term β to the centrality values for all nodes

Eigenvector Centrality

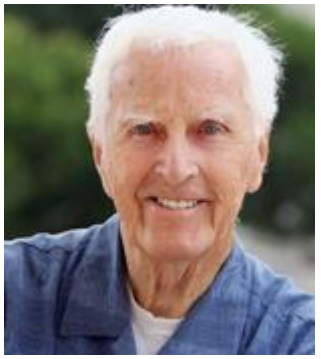
$$C_{\text{Katz}}(v_i) = \alpha \sum_{j=1}^{n} A_{j,i} C_{\text{Katz}}(v_j) + \beta$$

PageRank

- Problem with Katz Centrality:
  - In directed graphs, once a node becomes an authority (high centrality), it passes **all** its centrality along **all** of its out-links
- This is less desirable since not everyone known by a well-known person is well-known
- **Solution?**
  - We can divide the value of passed centrality by the number of outgoing links, i.e., out-degree of that node
  - Each connected neighbor gets a fraction of the source node's centrality

Betweenness Centrality

Another way of looking at centrality is by considering how important nodes are in connecting other nodes



Linton Freeman

$$C_b(v_i) = \sum_{s \neq t \neq v_i} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$$

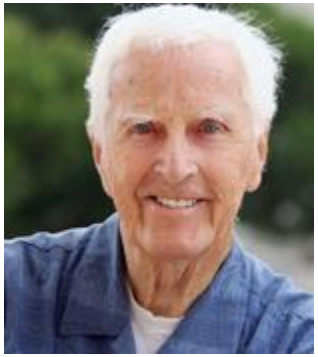
σ_{st} The number of shortest paths from vertex s to t – *a.k.a.* information pathways

$\sigma_{st}(v_i)$ The number of **shortest paths** from s to t that pass through v_i

Closeness Centrality

- The intuition is that influential/central nodes can quickly reach other nodes

- These nodes should have a smaller average shortest path length to others



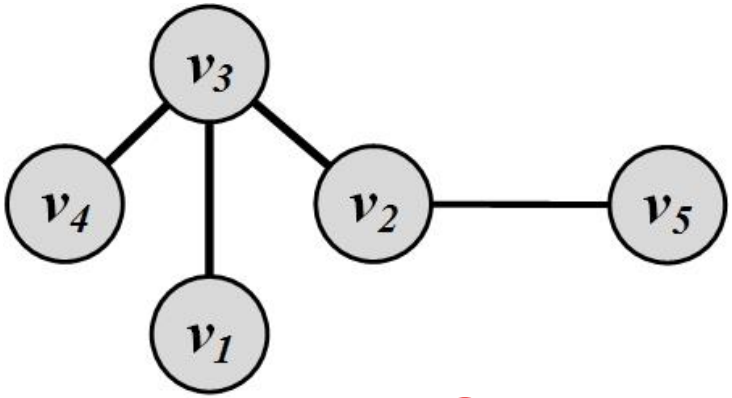
Linton Freeman

Closeness centrality: $C_c(v_i) = \frac{1}{\bar{l}_{v_i}}$

$$\bar{l}_{v_i} = \frac{1}{n-1} \sum_{v_j \neq v_i} l_{i,j}$$

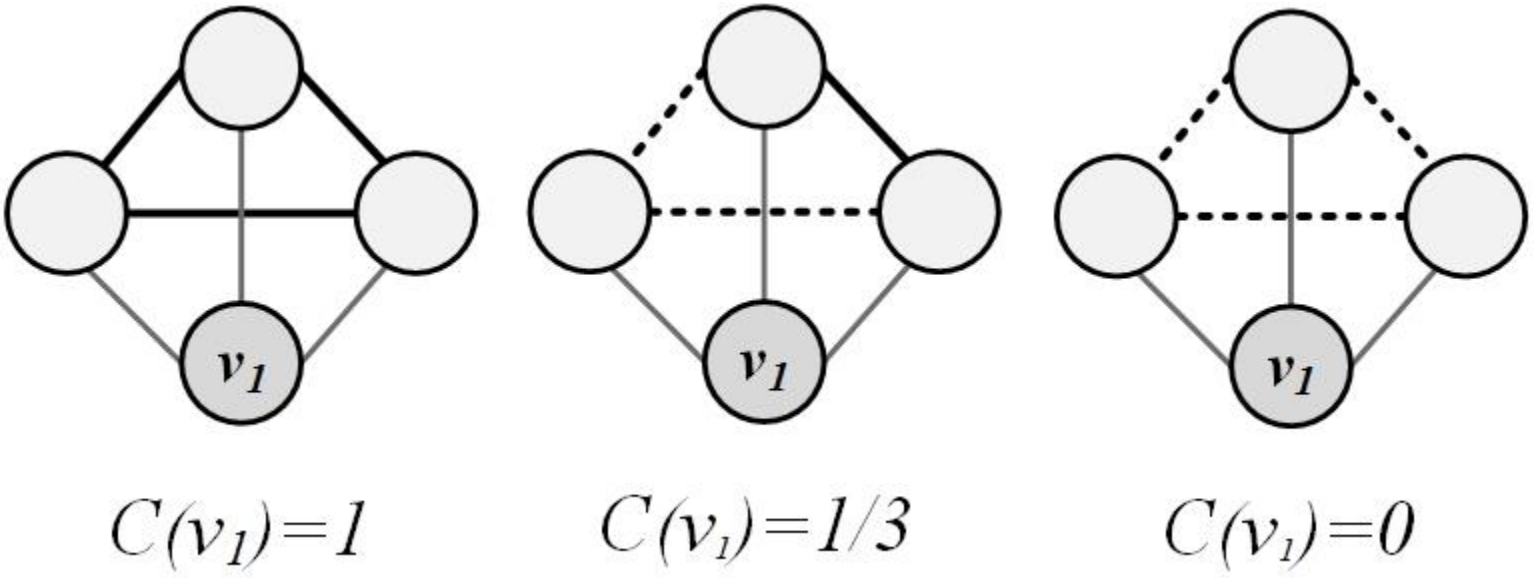
Group Centrality Example

Consider $S = \{v_2, v_3\}$



Group degree centrality = **3**
Group betweenness centrality = **3**
Group closeness centrality = **1**

Local Clustering Coefficient: Example



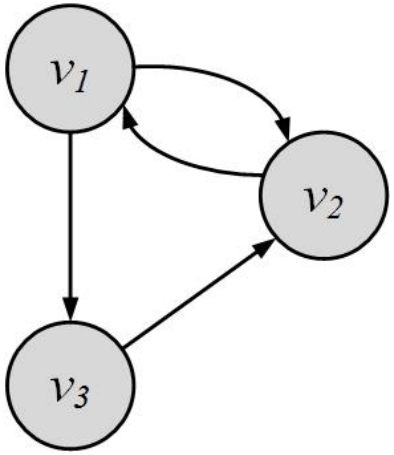
Thin lines depict connections to neighbors

Dashed lines are the missing link among neighbors

Solid lines indicate connected neighbors

- When none of neighbors are connected $C = 0$
- When all neighbors are connected $C = 1$

Reciprocity: Example



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

Reciprocal nodes: v_1, v_2

$$R = \frac{1}{m}\mathrm{Tr}(A^2) = \frac{1}{4}\mathrm{Tr}\left(\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}\right) = \frac{2}{4} = \frac{1}{2}.$$

Structural Equivalence: Definitions

Vertex similarity: $\sigma(v_i, v_j) = |N(v_i) \cap N(v_j)|$

Normalize?

Jaccard Similarity: $\sigma_{Jaccard}(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|}$

Cosine Similarity: $\sigma_{Cosine}(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{\sqrt{|N(v_i)||N(v_j)|}}$

The neighborhood $N(v)$ often excludes the node itself v.

What can go wrong?

Connected nodes not sharing a neighbor will be assigned zero similarity

Solution:

We can assume nodes are included in their neighborhoods

Regular Equivalence

v_i and v_j are similar when v_j is similar to v_i's neighbors v_k

$$\sigma_{regular}(v_i, v_j) = \alpha \sum_k A_{i,k}\sigma_{Regular}(v_k, v_j)$$

In vector format

$$\sigma_{regular} = \alpha A \sigma_{Regular}$$

<u>A vertex is highly similar to itself</u>, we guarantee this by adding an identity matrix to the equation

$$\sigma_{regular} = \alpha A \sigma_{Regular} + \mathbf{I}$$

$$\sigma_{regular} = (\mathbf{I} - \alpha A)^{-1}$$

When $\alpha < 1/\lambda_{max}$ the matrix is invertible

innovate achieve lead

BITS Pilani

Pilani Campus

6. Network Models

Key points

These three properties – **power-law degree distribution**, **high clustering coefficient**, and **small average path length** are consistently observed in real-world networks.

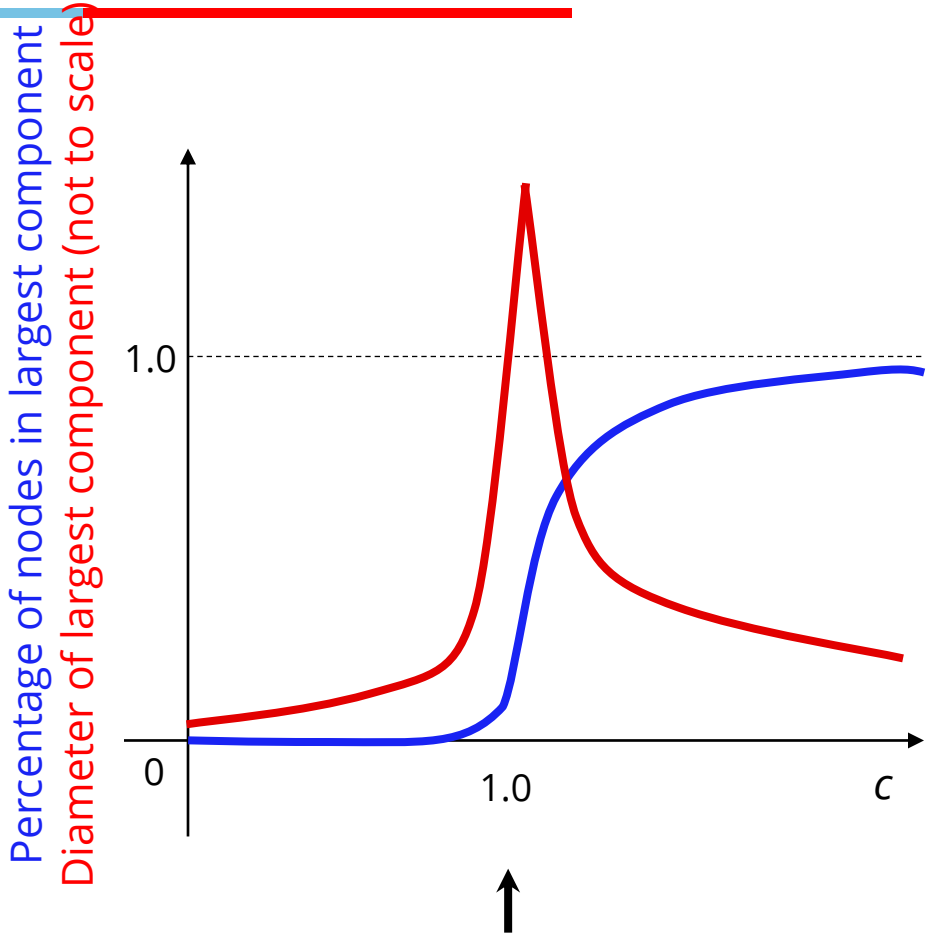
We design models based on simple assumptions on how friendships are formed, hoping that these models generate scale-free networks, with high clustering coefficient and small average path lengths.

Random Graphs

If $c < 1$:
- **small**, isolated clusters
- **small** diameters
- **short** path lengths

At $c = 1$:
- a ***giant component*** appears
- diameter **peaks**
- path lengths are **long**

For $c > 1$:
- almost **all** nodes **connected**
- diameter **shrinks**
- path lengths **shorten**



Properties of Random Graphs

Random graphs can model **average path length** in a real-world network accurately, but fail to generate a realistic **degree distribution** or **clustering coefficient**

Small World Model: Real-World Network and Simulated Graphs

	Original Network				Simulated Graph	
Network	*Size*	*Average Degree*	*Average Path Length*	*C*	*Average Path Length*	*C*
Film Actors	225,226	61	3.65	0.79	4.2	0.73
Medline Coauthorship	1,520,251	18.1	4.6	0.56	5.1	0.52
E.Coli	282	7.35	2.9	0.32	4.46	0.31
C.Elegans	282	14	2.65	0.28	3.49	0.37

Both average path lengths and clustering coefficients are modeled properly

Preferential Attachment Model: Real-World Networks and Simulated Graphs

	Original Network				Simulated Graph	
Network	*Size*	*Average Degree*	*Average Path Length*	*C*	*Average Path Length*	*C*
Film Actors	225,226	61	3.65	0.79	4.90	≈ 0.005
Medline Coauthorship	1,520,251	18.1	4.6	0.56	5.36	≈ 0.0002
E.Coli	282	7.35	2.9	0.32	2.37	0.03
C.Elegans	282	14	2.65	0.28	1.99	0.05

Average path lengths are modeled properly, whereas the clustering coefficient is underestimated

7. Data Mining Essentials

Data Mining Algorithms

Supervised Learning Algorithm

Classification (class attribute is discrete)

Assign data into predefined classes

Spam Detection

Regression (class attribute takes real values)

Predict a real value for a given data instance

Predict the price for a given house

Unsupervised Learning Algorithm

Group similar items together into some clusters

Detect communities in a given social network

Classification with Network Information

Classification with Network Information

Consider a friendship network on social media and a product being marketed to this network.

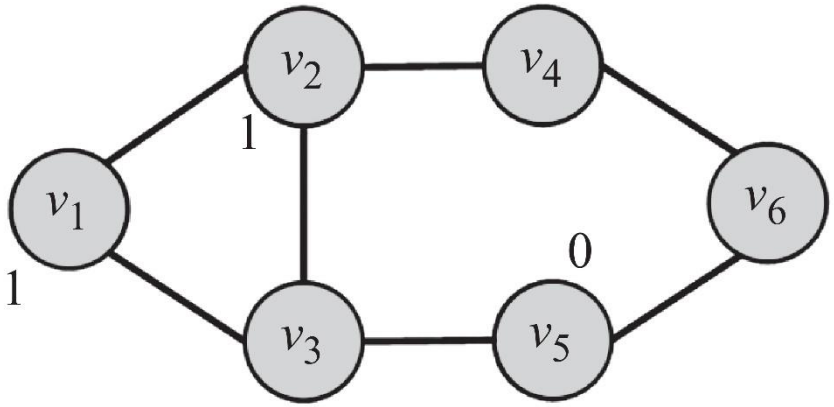
The product seller wants to know who the potential buyers are for this product.

Assume we are given the network with the list of individuals that decided to buy or not buy the product. Our goal is to predict the decision for the undecided individuals.

This problem can be formulated as a classification problem based on features gathered from individuals.

However, in this case, we have additional friendship information that may be helpful in building better classification models

Classification with Network Information



Let y_i denote the label for node i.
We can assume that

$$P(y_i = 1) \approx P(y_i = 1|N(v_i))$$

How can we estimate $\mathrm{P}(y_i = 1|N(v_i))$?

Weighted-vote Relational-Neighbor (wvRN)

wvRN provides an approach to estimate $\mathrm{P}(y_i = 1|N(v_i))$

In wvRN, to find the label of a node, we compute a weighted vote among its neighbors

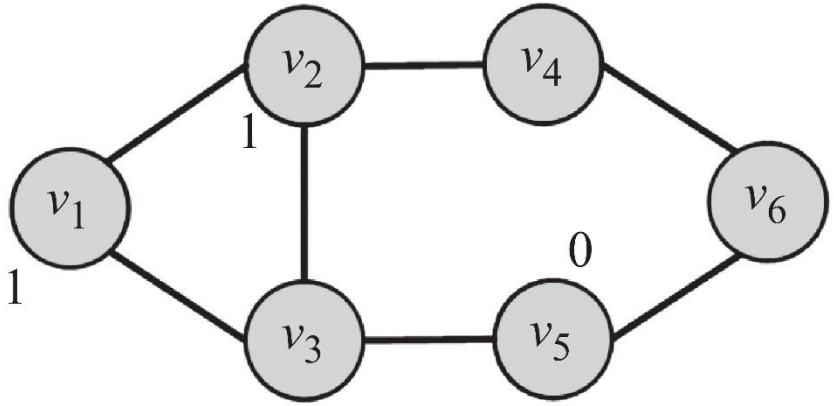
$$P(y_i = 1|N(v_i)) = \frac{1}{|N(v_i)|} \sum_{v_j \in N(v_i)} P(y_j = 1|N(v_j))$$

$\mathrm{P}(y_i = 1|N(v_i))$ is only calculated for unlabeled v_i's

We need to compute these probabilities using **some order** until convergence [i.e., they don't change]

What happens for different orders?

wvRN example



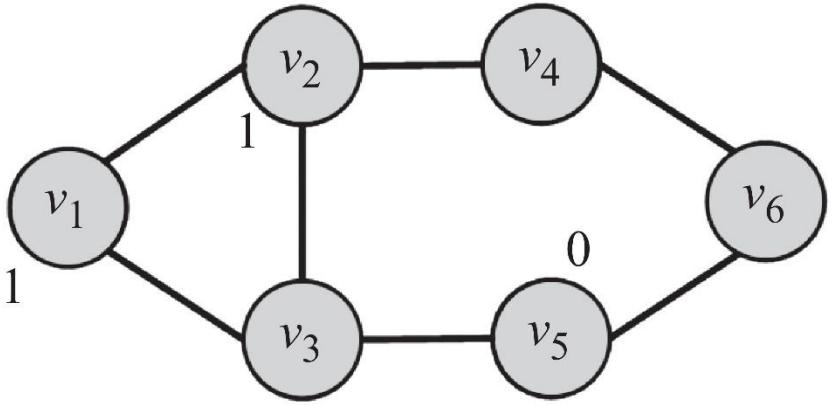
$$P(y_1 = 1|N(v_1)) = 1$$
$$P(y_2 = 1|N(v_2)) = 1$$
$$P(y_5 = 1|N(v_5)) = 0$$

$$P(y_3|N(v_3))$$
$$= \frac{1}{|N(v_3)|} \sum_{v_j \in N(v_3)} P(y_j = 1|N(v_j))$$
$$= \frac{1}{3}(P(y_1 = 1|N(v_1)) + P(y_2 = 1|N(v_2)) + P(y_5 = 1|N(v_5)))$$
$$= \frac{1}{3}(1 + 1 + 0) = 0.67$$

$$P(y_4|N(v_4)) = \frac{1}{2}(1 + 0.5) = 0.75$$

$$P(y_6|N(v_6)) = \frac{1}{2}(0.75 + 0) = 0.38$$

wvRN example



$$P_{(1)}(y_4|N(v_4)) = \frac{1}{2}(1 + 0.38) = 0.69$$

$$P_{(1)}(y_6|N(v_6)) = \frac{1}{2}(0.69 + 0) = 0.35$$

$$P_{(2)}(y_4|N(v_4)) = \frac{1}{2}(1 + 0.35) = 0.68$$

$$P_{(2)}(y_6|N(v_6)) = \frac{1}{2}(0.68 + 0) = 0.34$$

$$P_{(3)}(y_4|N(v_4)) = \frac{1}{2}(1 + 0.34) = 0.67$$

$$P_{(3)}(y_6|N(v_6)) = \frac{1}{2}(0.67 + 0) = 0.34$$

$$P_{(4)}(y_4|N(v_4)) = \frac{1}{2}(1 + 0.34) = 0.67$$

$$P_{(4)}(y_6|N(v_6)) = \frac{1}{2}(0.67 + 0) = 0.34$$

Questions?

BITS Pilani
Pilani Campus