



**BITS** Pilani  
Pilani Campus

# Social Media Analytics: Overview

Dr. Prasad Ramanathan

[p\\_ramanathan@wilp.bits-pilani.ac.in](mailto:p_ramanathan@wilp.bits-pilani.ac.in)

# 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

- Facebook
- X (formerly Twitter)
- Instagram
- Pinterest
- Snapchat
- ...

# What is Social Media?

Social Media is the use of electronic and Internet tools for the purpose of sharing and discussing information and experiences with other human beings in more efficient ways.

# Social Media Landscape 2015

## Social Media Landscape 2015



FredCavazza.net

# Social Media Landscape 2023

## Networking

tinder badoo  
match bumble clover  
happn Hinge hangout  
okcupid Grindr  
ravery UNTAPPO  
care2 'etoro'  
ASMALLWORLD  
nextdoor

Untappd  
Bettermode azar  
diaspora SLOWLY  
NING hoop  
Spacehey TAGGED  
LYNK  
Microsoft 365  
Google Workspace Zoho Workplace SharePoint ONLYOFFICE  
monday lumapps SIMPLER  
Notion Confluence  
unify nifty kissflow  
gliffy Creately  
Lucidchart cacao  
Dropbox Paper notejoy OneNote  
boxNOTES todoist Simpliconote  
Trello backlog Planner miro  
Basecamp teamwork Quip coda wrike  
asana workfront  
jive Airtable Podio smartsheet

Decentraland CRYPTOXOELS  
SOMNIUM hubs SPACE immersed  
SECOND LIFE Bitmoji avakinlife  
VR CHAT SANSA VR HIBERWORLD  
ZEPETO neopets sinespace

## Collaborating

## Publishing

eventbrite bio.fm  
Linktree feedlink  
compite.bio  
evite ancestry  
classmates



Meet Zoom  
Jamespot talkspirit  
Whaller Yammer's chatter  
twist Loop SYMPHONY  
THREADS Chime Google Chat

## Discussing

Medium  
SQUARESPACE  
Blogger WIX  
Typepad opendairy  
Svble LIVEJOURNAL ghost



substack  
Gumroad  
upscribe  
Buttondown  
NEWGROUNDS  
myspace WT.Social  
WattPad Pillowfort  
mastodon Ethereum World  
ello Mirror Peepeth  
Cortex BitCloud  
orbis DeSo Sigle

## Sharing

Apple Podcasts  
PodBean  
Google Podcasts buzzsprout majelan  
Infoplease Fandom  
Citizendium SCHOLARPEDIA  
WIKIPÉDIA



parazzi DISPO  
BeReal. Liveln  
Locket Widget  
LiveStatus  
anobii primitives  
ShowMe showtime  
Tripadvisor couchsurfing  
yelp

## Messaging

slideshare SCRIBD issuu  
studocu  
slashdot ARTIFACT pocket  
Post digg  
Scoop FLIPBOARD  
Instapaper

FIREWORK Playhouse SHOPSHOPS  
SmugMug 500px  
flickr  
Tenor  
imgur

## Collaborating

@FredCavazza

V 1.0

FredCavazza.net

# Social Media: Examples



- A wiki article
- Web reviews and ratings of a popular pizza place in your city
  - E.g., Yelp.com
- An online social network of your professional contacts
  - E.g., Facebook.com, LinkedIn.com
- An iPhone application that informs you where parking is likely available
  - FasPark

# Types of Social Media



- Online Social Networking
- Publishing
  - Blogging
  - Wiki
- Micro blogging
- Social News
- Social Bookmarking
- Media Sharing
  - Video Sharing
  - Photo Sharing
  - Podcast Sharing
- Opinion, Review, and Ratings Websites
- Answers
- Entertainment



Online Social Networks



Blogging



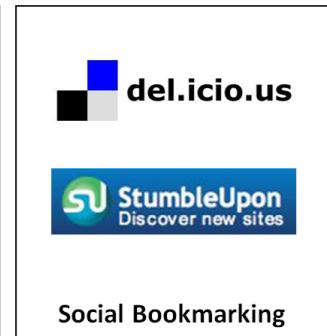
Microblogging



Wikis



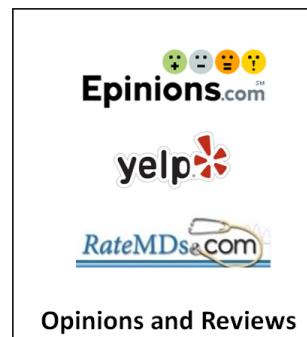
Social News



Social Bookmarking



Media Sharing



Opinions and Reviews



Answers

# Online Social Networking

innovate

achieve

lead

Online Social Networks are web-based services that allow individuals and communities to connect with real world friends and acquaintances online

- Interactions
  - Friendship interaction
    - Friends, like, comments, ...
  - Media Sharing
  - Sending and receiving messages

- Examples
  - Facebook.com
  - MySpace.com
  - Bebo.com
  - Orkut.com

This screenshot shows a MySpace profile for a user named 'Pei Pei'. The profile includes a profile picture, a bio stating 'Seattle United States', and statistics like 'Profile Views: 208' and 'Last Login: 3/18/2010'. It features links for 'My Music', 'Music Videos', 'Charts', 'New Releases', 'Featured Playlists', 'Karaoke', 'Shows', and 'Forums'. Below the bio, there's a 'Contacting Pei Pei' section with options like 'Send Message', 'Add to Friends', 'IM / Call', and 'Add to Group'. At the bottom, there's a 'General Info' section showing 'Member Since: 3/1/2010', 'Band Members: Sayuri Wijaya Gould', 'Influences: Too many to list them all... The Beach Boys, Pink Floyd, Zee Avi, A Fine Frenzy, Black Whale, Damien Rice, and more.', and 'Type of Label: Unsigned'.

This screenshot shows a Facebook page for 'Barack Obama'. The page has a large profile picture of Barack Obama. It includes sections for 'Wall', 'Info', 'OFA Store', 'Photos', 'Join OFA', and 'Video'. There are posts from August 2010, including one about voting and another from Pei Pei. The 'Information' section lists him as the 'President of the United States'. The 'Upcoming Shows' section shows a performance at 'Freshly's Cafe' on March 20, 2010. The 'About' section contains a bio about his musical interests and a note about Ramadan. The page has over 12 million likes.

# Blogging



A blog is a journal-like website for users, a.k.a. bloggers, to contribute textual and multimedia content, arranged in reverse chronological order

- Maintained both individually or by a community
  - See a tutorial at KDD  
[http://videolectures.net/kdd08\\_liu\\_briat/](http://videolectures.net/kdd08_liu_briat/)

- Usages:
  - Sharing information and opinions with friends and strangers
  - Disseminating subject-specific content
  - Who is the influencer  
[http://videolectures.net/wsdm08\\_agarwal\\_iib/](http://videolectures.net/wsdm08_agarwal_iib/)

A screenshot of the Marriott International website. At the top, there's a banner with the text "Marriott on the move". Below it is a photo of Bill Marriott, Chairman &amp; CEO of Marriott International. A "Featured Post" section shows an article titled "In Good Company" posted on 10/06/2011 at 9:18 AM. There's also a "Listen to Blog" button. The sidebar includes links for "Home", "Categories" (Books, Brands, Current Affairs, Diversity, Education, Employment, Environment, Film, Food and Drink, Government, Health, Operations, Personal, Service, Sports, Technology, Television, Travel, Web/Tech), and a "RSS Feeds" section with links to My Yahoo!, Google, and RSS. On the right, there's a "Search" bar and links for "Follow Us" (Renaissance Life, Courtyard Connection, Marriott in the Kitchen) and "Links" (Marriott.com, Customer Care). A sidebar on the far right displays "THE WORLD'S MOST ETHICAL COMPANIES" with a link to www.ethisphere.com.

A screenshot of the tuaw.com website, which is described as "The Unofficial Apple Weblog". The main article is titled "Flash-based iPod: who cares?" and was posted on Dec 4, 2004, at 6:30 PM ET by Barb Dybdahl. The article discusses John Gruber's concerns about Apple releasing a flash-based version of the iPod. It includes several images of iPods and a sidebar with "RESOURCES" and "SPONSORED TEXT LINKS". The sidebar also features a "RECENT COMMENTS" section with a link to "IceRocket — A new way to search: www.icerocket.com".

# Microblogging



Microblogging can be considered as a counterpart to blogging, but with limited content

- Usage
  - communication medium
  - social interaction
  - citizen journalism
- Service Providers:
  - X (formerly Twitter)
  - Google buzz

Mario Armstrong (@marioarmstrong) FOLLOWING YOU  
Tweets abt Tech, Life & Inspiration! Emmy winner! TV dude on HLN,CNN, NBC TODAY show! People lover,Vegan,Shoe addict! 1.5m followers at Socialcam.com/Mario  
Television and Online - <http://marioarmstrong.com>

12,936 TWEETS 1,444 FOLLOWING 12,128 FOLLOWERS Following

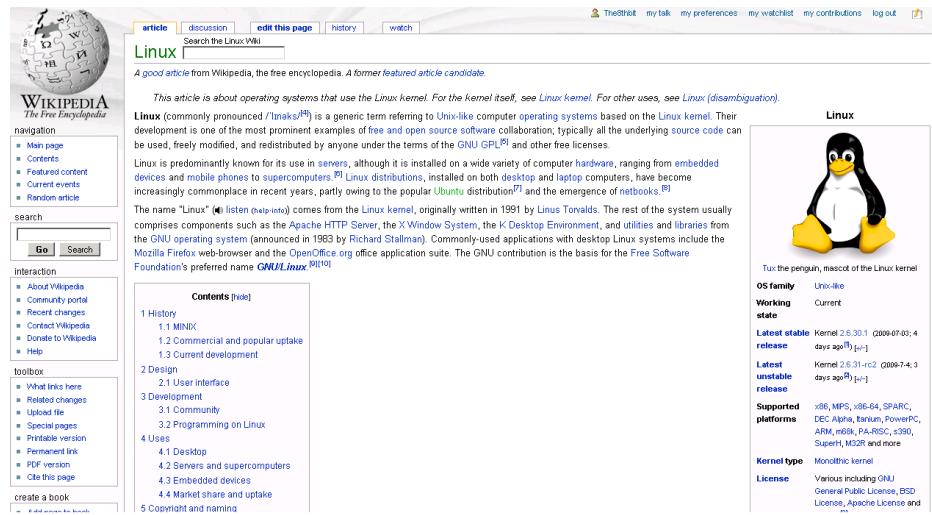
Tweets

Mario Armstrong (@marioarmstrong) @gadrienne1983 thx ALI & I'm following u now :-)  
View conversation

Mario Armstrong (@marioarmstrong) @avcilio ahnh u know! New twitter, iPhone 5, trying to get my own

A wiki is a collaborative editing environment that allows users to develop Web pages using a simplified markup language

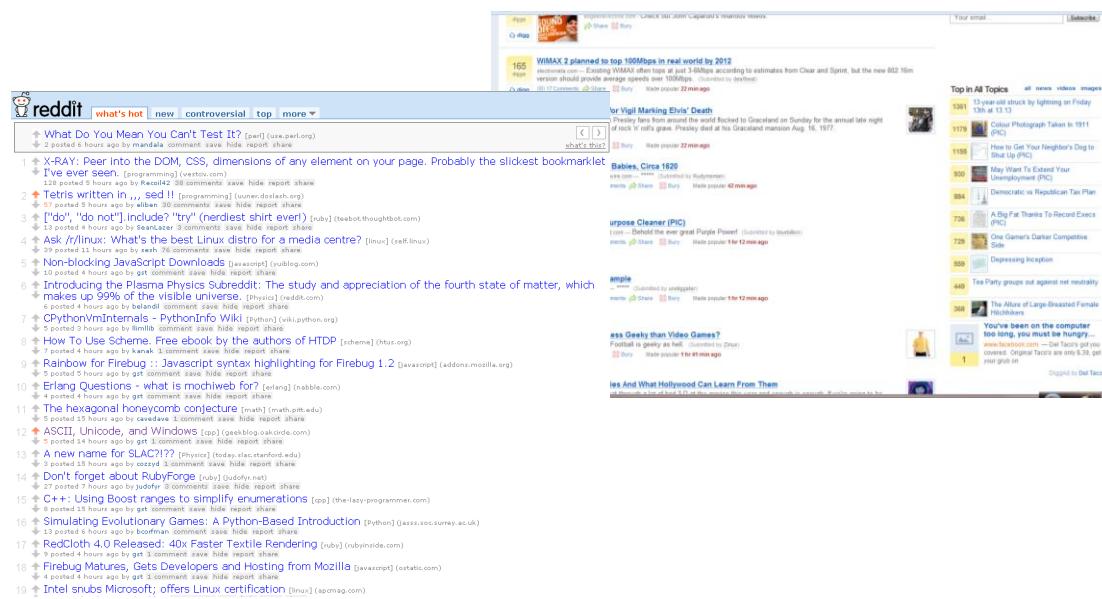
- Wikipedia allows interested individuals to collaboratively develop articles on a variety of subjects.
- Using the wisdom of crowds effectively, it has become a comprehensive repository of information useful to a variety of individuals



The image shows two related Wikipedia pages. On the left is the main article "Linux" (The Free Encyclopedia), featuring the Wikipedia logo, a search bar, and a sidebar with links like "Main page", "Contents", and "Recent changes". The main content discusses the history of Linux, mentioning Linus Torvalds and the Apache HTTP Server. On the right is the article "Linux (kernel)", which includes a sub-section on the "Linux kernel", mentioning its stable releases (Kernel 2.6.30.1) and unstable releases (Kernel 2.6.31-rc2). It also features a large image of Tux, the Linux mascot penguin.

Social News refers to the sharing and selection of news stories and articles by a community of users.

- Users can share articles that they believe would interest the community
- Samples:
  - Digg.com
  - Slashdot
  - Fark
  - Reddit



# Social Bookmarking



Social Bookmarking sites allow users to bookmark web content for storage, organization and sharing.

- These bookmarks can be tagged with metadata to categorize and provide context to the shared content, allowing users to organize information making it easy to search and identify relevant information.
- Samples
  - Delicious.com
  - StumbleUpon.com

The screenshot shows the Delicious.com homepage. At the top, there's a banner with the text "The tastiest bookmarks on the web. Save your own or see what's fresh now!" and a "Learn More" button. Below the banner is a search bar with the placeholder "Search the biggest collection of bookmarks in the universe...". To the right of the search bar are "Join Now" and "Sign In" buttons. A "HIDE INTRO" link is located at the bottom right of the banner area. The main content area features a "Popular Bookmarks" section with a grid of bookmark cards. Each card includes the title of the bookmark, a small thumbnail, the number of saves (e.g., 62, 61, 81, 61, 122, 68, 90, 83, 94, 92), and a list of tags. To the right of the bookmark grid is a "Popular Tags" sidebar with a scrollable list of tags like "design", "blog", "video", etc. At the very bottom of the page is a footer with links to "delicious | about | blog | terms of service | privacy policy | copyright policy | forums | support" and a "What's new?" link.

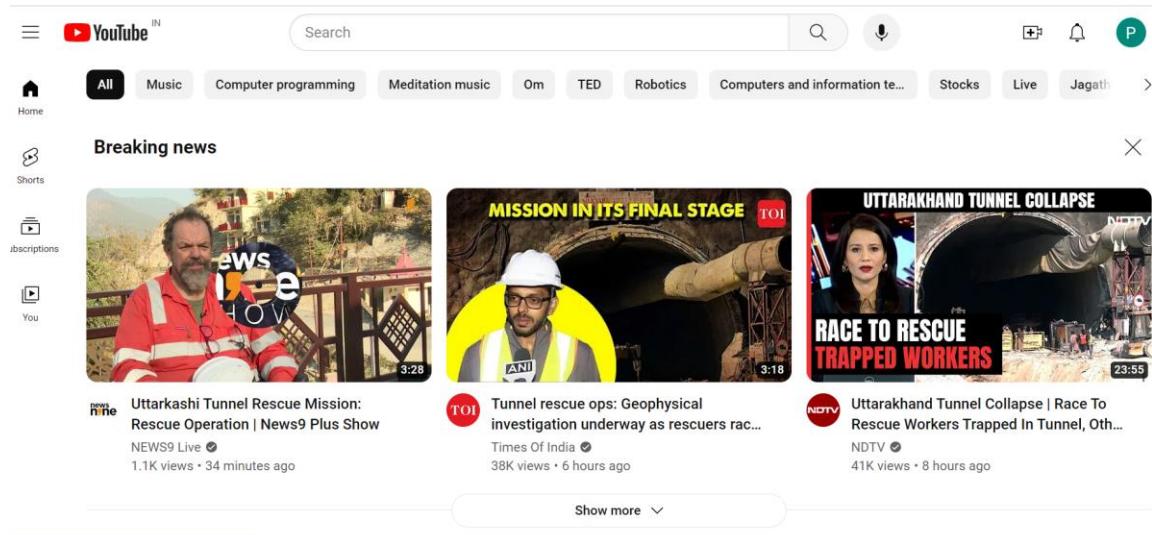
# Media Sharing



Media sharing is an umbrella term that refers to the sharing of a variety of media on the web.

Users share such multimedia content of possible interest to others

- Samples:
  - Video Sharing:
    - YouTube.com
  - Photo Sharing:
    - Flickr.com, picasa.com
  - Document Sharing:
    - Scribd.com,  
Slideshare.com
  - Livecasting:
    - Justin.tv, Ustream.com



# Opinion, Review, and Ratings Websites



Opinion, review, and ratings websites are websites whose primary function is to collect and publish user-submitted content in the form of subjective commentary on existing products, services, entertainment, businesses, places, etc. Some commercial sites may serve a secondary purpose as review sites by publishing product reviews submitted by customers.

- Examples
  - Cnet.com
  - Epinions.com
  - yelp.com
  - tripadvisor.com

The collage includes:

- A screenshot of the Yelp website showing a search for "Croissant" in San Francisco, listing results like "Tartine Bakery > Menu > Breakfast Pastries > Croissant".
- A screenshot of the GSMarena website featuring a search bar and a list of smartphone brands under "PHONE FINDER".
- A screenshot of a news or review site showing a post titled "Honor Pad X9 review" with a thumbnail image of the device.
- A screenshot of a news or review site showing a post titled "Poco C65/Redmi 13C review" with a thumbnail image of the phone.
- A screenshot of a news or review site showing a post titled "Moto G54 (Power edition) review" with a thumbnail image of the phone.
- A screenshot of a news or review site showing a post titled "Google Pixel 8 Pro vs Google Pixel 7" with a thumbnail image of both phones.

# Socially-Provided Answers



In these sites, users who require certain guidance, advice or knowledge can ask questions. Other users from the community can answer these questions based on knowledge acquired from previous experiences, personal opinions or from relevant research.

- Unlike review and opinion sites, which contain self-motivated contribution of opinions, answer sites contain knowledge shared in response to a specific query.
- Samples:
  - WikiAnswers, Yahoo Answers, Quora

Search Google Analytics Questions and Topics [Add Question](#)

Question added to topic Google Analytics:  
**What percentage of visits would Omniture / Google Analytics / Coremetrics etc miss?**  
Assuming client-side integration, compared with the numbers from the web servers and proxy logs.  
Follow · ⚡ Repost · 0 Answers · 5:55pm

---

Answer added in topic Google Analytics:  
**How can I track Pinterest in Google Analytics?**  
1 Ross Allen, Front End Engineer at Airbnb  
Their Javascript pinit.js file (<http://assets.pinterest.com/js/pinit.js>) doesn't seem to add any callbacks, so the best you can do is track clicks on the 'Pin It' button in Goo... [\(more\)](#)  
Upvote · ⚡ Repost · 2 Answers · 5:17pm

---

Answer added in topic Google Analytics:  
**Google Analytics: Why would someone from an email marketing company tell me that Google analytics does not track visits from Mac users?**  
2 Anon User  
The person was seeing if you were gullible enough to be a good fit with their product.  
Sales 101.  
Upvote · ⚡ Repost · 4 Answers · 3:52pm

[Share Topic](#) · [Invite People](#)

[Twitter](#) [Facebook](#) [Quora](#)

---

**Top Answerers**

 <b>Mike Sullivan</b> 20 Answers	 <b>Ozberk Olcer</b> 20 Answers Director of Web Analytics in SEM AS. (Google Analytics Certified Partner)
 <b>Shay Sharon</b> 22 Answers	 <b>AJ Kohn</b> 17 Answers
 <b>Christopher O'Donnell</b> 11 Answers	

---

Followed by 5455 People



# Main Characteristics

- **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.

# Social Media Mining & Analytics

---



**Social Media Mining** is the process of representing, analyzing, and extracting meaningful patterns from social media data

**Social Media Analytics** is the ability to gather and find meaning in data gathered from social channels to support business decisions — and measure the performance of actions based on those decisions through social media.

# Why Social Media Analytics is Important?

---

Social media analytics helps companies address user experiences and use them to:

- Spot trends related to offerings and brands
- Understand conversations — what is being said and how it is being received
- Derive customer sentiment towards products and services
- Gauge response to social media and other communications
- Identify high-value features for a product or service
- Uncover what competitors are saying and its effectiveness
- Map how third-party partners and channels may affect performance

# Social Media Mining: An Interdisciplinary Field

---

- Individuals – Social Atoms
- Communities – Social Molecules
- Contents, Sites, Networks – Entities
  
- Social media can be considered a world of social atoms (i.e., individuals), entities (e.g., content, sites, networks, etc.), and interactions between individuals and entities.
- Social theories and social norms govern the interactions between individuals and entities.
- Social media mining...collect information about **individuals** and **entities**, measure their **interactions**, and **discover patterns** to understand human **behavior**

# Challenges

---

## 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?

# Course outline

## Part I: Essentials (Chapters 2-5)

We learn to answer questions such as the following:

1. Who are the most important people in a social network?
2. How do people befriend others?
3. How can we find interesting patterns in user-generated content?

## Part II: Communities and Interactions (Chapters 6 and 7)

To analyze how communities are formed, how they evolve, how the qualities of detected communities are evaluated, ways in which information diffusion in social media can be studied. We aim to answer general questions such as the following:

1. How can we identify communities in a social network?
2. When someone posts an interesting article on a social network, how far can the article be transmitted in that network?

## Part III: Application (Chapters 8-10)

Exemplify social media mining using real-world problems in dealing with social media: measuring influence, recommending in a social environment, and analyzing user behavior. We aim to answer these questions:

1. How can we measure the influence of individuals in a social network?
2. How can we recommend content or friends to individuals online?
3. How can we analyze the behavior of individuals online?

# What will you learn in this course?

---

- **Technology Enablers for Social Media Analysis** - Learn representative algorithms and tools
  - Natural Language Processing: Entities, Relationships, Sentiment
  - Machine Learning: Segmentation, Cluster Analysis
  - Graph & Network Analysis: Influencers
  - Community & Interaction Analysis
- **Applications of SMA**
  - Influence
  - Recommendation
  - Behaviour Analytics
  - Social Media Marketing
  - Disaster Management



# How can you apply the learnings?

---

- Product Development
- Customer Experience
- Competitive Analysis
- Operational Efficiency

# Case Study: Coca-Cola

DATA SCIENCE

## The Power of Social Media Analytics: Case Study of Coca-Cola



Harshini Bhat  
Data Science Consultant At AlmaBetter

5 mins 7575 Published on 10 Aug, 2023

### Share a Coke:

[https://en.wikipedia.org/wiki/Share\\_a\\_Coke#:~:text=Share%20a%20Coke%20is%20a,followed%20by%20a%20person's%20name.](https://en.wikipedia.org/wiki/Share_a_Coke#:~:text=Share%20a%20Coke%20is%20a,followed%20by%20a%20person's%20name.)



**BITS** Pilani  
Pilani Campus

# Questions?



# Sentiment Analysis and Opinion Mining

---

# Introduction

- Sentiment analysis (SA) or **opinion mining**
  - computational study of opinion, sentiment, appraisal, evaluation, and emotion.
- **Why is it important?**
  - Opinions are key influencers of our behaviors.
    - Our beliefs and perceptions of reality are conditioned on how others see the world. Whenever we need to make a decision we often seek out the opinions from others.
    - Rise of social media → opinion data
  - Rise of AI and chatbots:
    - Emotion and sentiment are key to human communication

# Terms defined - Merriam-Webster

- **Sentiment:** an attitude, thought, or judgment prompted by feeling.
  - A sentiment is more of a feeling.
  - *“I am concerned about the current state of the economy.”*
- **Opinion:** a view, judgment, or appraisal formed in the mind about a particular matter.
  - a concrete view of a person about something.
  - *“I think the economy is not doing well.”*

# SA: A fascinating problem!

- Intellectually challenging & many applications.
  - A popular research area in NLP, and data mining  
(Shanahan, Qu, and Wiebe, 2006 (edited book); Surveys - Pang and Lee 2008; Liu, 2006, 2012, and 2015)
  - spread from CS to management and social sciences  
(Hu, Pavlou, Zhang, 2006; Archak, Ghose, Ipeirotis, 2007; Liu Y, et al 2007; Park, Lee, Han, 2007; Dellarocas, Zhang, Awad, 2007; Chen & Xie 2007).
  - A large number of companies in the space globally
    - > 300 in the US alone.
- It touches every aspect of NLP & also is confined.
  - A “simple” semantic analysis problem.
- A major technology from NLP.
  - But it is hard.

# Roadmap

- Sentiment analysis problem
- Document sentiment classification
- Sentence subjectivity & sentiment classification
- Aspect-based sentiment analysis
- Mining comparative opinions
- Opinion lexicon generation
- Some interesting sentences
- **Summary**

# Two main types of opinions

(Jindal and Liu 2006; Liu, 2010)

- **Regular opinions:** Sentiment/opinion expressions on some target entities
  - Direct opinions:
    - “The touch screen is really cool.”
  - Indirect opinions:
    - “After taking the drug, my pain has gone.”
- **Comparative opinions:** Comparison of more than one entity.
  - E.g., “iPhone is better than Blackberry.”
- We focus on regular opinions first, and just call them opinions.

# (I): Definition of opinion

- **Id:** **Abc123** on **5-1-2008** -- “*I bought an iPhone yesterday. It is such a nice phone. The touch screen is really cool. The voice quality is great too. It is much better than my Blackberry. However, my mom was mad with me as I didn't tell her before I bought the phone. She thought the phone was too expensive*”
- **Definition:** An **opinion** is a quadruple (Liu, 2012),  
**(target, sentiment, holder, time)**
- This definition is concise, but not easy to use.
  - Target can be complex, e.g., “*I bought an iPhone. The voice quality is amazing.*”
    - **Target** = **voice quality**? (not quite)

# 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*

# Our example blog in quintuples

- **Id:** Abc123 **on** 5-1-2008 “*I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is great too. It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, ...*”
- **In quintuples**
  - (iPhone, GENERAL, +, Abc123, 5-1-2008)
  - (iPhone, touch\_screen, +, Abc123, 5-1-2008)
  - ....
- We will discuss comparative opinions later.

## (II): Opinion summary (Hu and Liu 2004)

- **With a lot of opinions, a summary is necessary.**
  - Not traditional text summary: from long to short.
  - Text summarization: defined operationally based on algorithms that perform the task
- **Opinion summary (OS) can be defined precisely,**
  - not dependent on how summary is generated.
- **Opinion summary needs to be quantitative**
  - 60% positive is very different from 90% positive.
- **Main form of OS: *Aspect-based opinion summary***

# 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**.

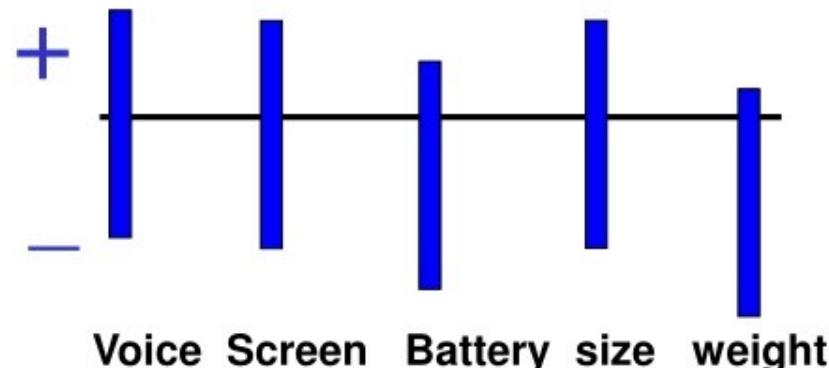
...

Aspect: **voice quality**

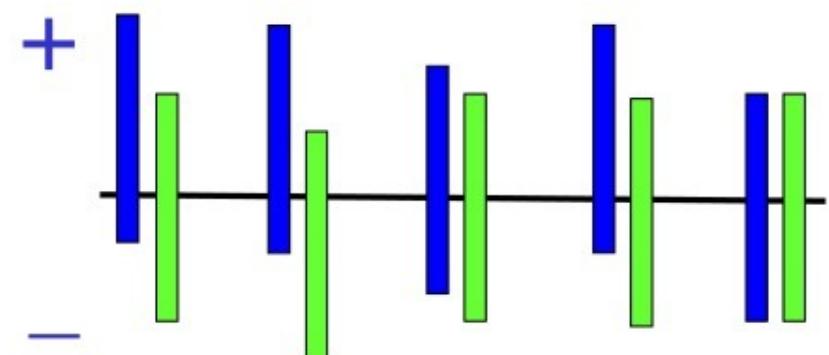
...

(Liu et al. 2005)

- Opinion Summary of 1 phone



- Opinion comparison of 2 phones



# Aspect-based opinion summary

**bing**

HP printer

ALL RESULTS

Shopping

POPULAR FEATURES

- all
- Affordability
- Speed
- Print Quality
- Reliability
- Ease Of Use
- Brand
- Installation
- Size
- Compatibility

SHOPPING

HP LaserJet 1020 - prin

from \$119.99

The HP high-qu



[user reviews](#) [product reviews](#)

user reviews

speed 96%

The quality is as good as any laserjet printer I've used and the speed is fast.  
Love Reading [www.amazon.com](http://www.amazon.com) 3/17/2006 more...

Quick and fast transaction.  
Arthur L. Taylor [www.amazon.com](http://www.amazon.com) 2/5/2008 more...

It's small and fast and very reliable.  
Muffinhead's mom [www.amazon.com](http://www.amazon.com) 1/9/2007 more...

Google products sony camera [Search Products](#)

Sony Cyber-shot DSC-W370 14.1 MP Digital Camera (Silver)

[Overview](#) - [Online stores](#) - [Nearby stores](#) - [Reviews](#) - [Technical specifications](#) - [Similar items](#) - [Accessories](#)

 \$140 [online](#), \$170 [nearby](#)

★★★★☆ 159 reviews [+1](#) [0](#)

**Reviews**

Summary - Based on 159 reviews

1 2 3 stars 4 stars 5 stars

**What people are saying**

<a href="#">pictures</a>	 "We use the product to take quickly photos."
<a href="#">features</a>	 "Impressive panoramic feature."
<a href="#">zoom/lens</a>	 "It also record better and focus better on sunny days."
<a href="#">design</a>	 "It has the slightest grip but it's sufficient."
<a href="#">video</a>	 "Video zoom is choppy."
<a href="#">battery life</a>	 "Even better, the battery lasts long."
<a href="#">screen</a>	 "I Love the Sony's 3" screen which I really wanted."

[view: positive comments \(44\)](#)

# Summarization

(AddStructure.com)



(1,043 customer reviews)

## Pros

- Great Price (518)
- Good Sound Quality (895)
  - ▲ Easy Setup (138)
  - ▼ Cons
- Remote (9)
- Inputs (8)
- Little product flaws (8)



(435 customer reviews)

## Pros

- Great Picture Quality (256)
- Good Sound Quality (77)
  - ▲ Easy Setup (60)
  - ▼ Cons
- Speakers (5)
- Changing Channels (4)
- Volume (3)

"The only down side is there is no **input** to connect to a computer."

)

"The only "bad" thing we have noticed is that there is quite a delay when you **change channels**."

# Not just ONE problem

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

- target **entity**: Named entity extraction, more
- **aspect** of **entity**: Aspect extraction
- **sentiment**: Sentiment classification
- opinion **holder**: Information/data extraction
- **time**: Information/data extraction

- There are more problems ...

- Other NLP problems

- Synonym grouping (voice = sound quality)
- Coreference resolution
- .....

# Reason for Opinion/Sentiment

- **Definition:** A reason for an opinion is the justification or explanation of the opinion.
- There are two main cases, e.g.,
  - (1). “*I hate this car as it eats too much gas.*”
    - Negative about an entity due to a bad aspect.
      - This can be identified by negative aspects
  - (2). “*This car is too small.*” (Shuai et al. 2016)
    - Negative about an aspect because of a reason
      - This can be identified by aspect specific sentiment

# Qualifier of Opinion

- **Definition:** A qualifier of an opinion limits or modifies the meaning of the opinion.
- It tells what an opinion is good for, e.g.,
  - “*This car is too small for a tall person.*”
  - “*The picture quality of night shots is bad*”
- Not every opinion comes with an explicit reason and/or an explicit qualifier.
  - No reason and no qualifier, e.g.,
    - “This car is bad.”

# Two closely related concepts

- Subjectivity and emotion.
- Sentence subjectivity
  - An **objective sentence** presents some factual information,
  - while a **subjective sentence** expresses some personal feelings, views, emotions, or beliefs.
- Emotion
  - A mental state that arises spontaneously rather than through conscious effort and is often accompanied by physiological changes.

# Subjectivity

- Subjective expressions come in many forms, e.g. (Wiebe 2000; Wiebe et al 2004; Riloff et al 2006),
  - opinions,
  - allegations,
  - desires,
  - beliefs,
  - suspicions,
  - speculations.
- Many subjective sentences contain no positive or negative opinion
  - “*I think he went home after the class.*”

# Subjective Opinions

- **Definition:** A subjective opinion is a regular or comparative opinion given in a subjective statement.
- For example,
  - “Coke tastes great.”
  - “I think Google’s profit will go up next month.”
  - “This camera is a masterpiece.”
  - “We are seriously concerned about this new policy.”
  - “Coke tastes better than Pepsi.”

# Objective Sentences with Opinion

- Most opinion sentences are subjective, but objective (or factual) sentences can imply opinions too (Liu, 2010).
- They express desirable or undesirable facts
  - “The machine stopped working in the second day”
  - “We brought the mattress yesterday, and a body impression has formed.”
  - “After taking the drug, there is no more pain”
- Such sentences are very hard to handle

# Affect, Emotion and Mood

- Three closely related and confusing terms
- Dictionary definitions:
  - **Affect**. Feeling or emotion, especially as manifested by facial expression or body language.
  - **Emotion**. A mental state that arises spontaneously rather than through conscious effort and is often accompanied by physiological changes.
  - **Mood**. A state of mind or emotion.
  - **Feeling**. An affective state of consciousness, e.g., resulting from emotions, sentiments, or desires.

# Definitions from Psychology

- **Affect**: a neurophysiological state consciously accessible as the simplest raw feeling.
- **Emotion**: the indicator of affect. Owing to cognitive processing, emotion is a compound (not primitive) feeling concerned with an object, such as a person, an event, a thing, or a topic
- **Mood**: a feeling or affective state that typically lasts longer than emotion and tends to be more unfocused and diffused

# Emotion

- No agreed set of basic emotions of people.
  - Based on Parrott (2001), people have six basic emotions,
    - love, joy, surprise, anger, sadness, and fear.
- Although related, emotions and opinions are not equivalent.
  - Opinion: rational (+/-) view on something
    - Cannot say “I like”
  - Emotion: focusing on an inner feeling
    - Can say “I am angry.” or “There is sadness in her eyes”

Anger	Disgust	Contempt, loathing, revulsion
	Envy	Jealousy
	Exasperation	Frustration
	Irritability	Aggravation, agitation, annoyance, crosspatch, grouchy, grumpy
	Rage	Anger, bitter, dislike, ferocity, fury, hatred, hostility, outrage, resentment, scorn, spite, vengefulness, wrath
	Torment	Torment
Fear	Horror	Alarm, fear, fright, horror, hysteria, mortification, panic, shock, terror
	Nervousness	Anxiety, apprehension (fear), distress, dread, suspense, uneasiness, worry
Joy	Cheerfulness	Amusement, bliss, gaiety, glee, jolliness, joviality, joy, delight, enjoyment, gladness, happiness, jubilation, elation, satisfaction, ecstasy, euphoria
	Contentment	Pleasure
	Enthrallment	Enthrallment, rapture
	Optimism	Eagerness, hope
	Pride	Triumph
	Relief	Relief
	Zest	Enthusiasm, excitement, exhilaration, thrill, zeal
Love	Affection	Adoration, attractiveness, caring, compassion, fondness, liking, sentimentality, tenderness
	Longing	Longing
	Lust/Sexual desire	Desire, infatuation, passion
Sadness	Disappointment	Dismay, displeasure
	Neglect	Alienation, defeatism, dejection, embarrassment, homesickness, humiliation, insecurity, insult, isolation, loneliness, rejection
	Sadness	Depression, despair, gloom, glumness, grief, melancholy, misery, sorrow, unhappy, woe
	Shame	Guilt, regret, remorse
	Suffering	Agony, anguish, hurt
	Sympathy	Pity, sympathy
Surprise	Surprise	Amazement, astonishment

# Cause for Emotion

- Emotions have causes as emotions are usually caused by some internal or external events.
- “cause” not “reason” because an emotion is an effect produced by a cause (usually an event) rather than a justification or explanation in support of an opinion.
  - “After hearing of **his brother’s death**, he burst into tears,”

# Definition of Emotion

- **Definition (Emotion):** It is a quintuple,  
*(entity, aspect, emotion\_type, feeler, time)*
  - E.g., “*I am so mad with the hotel manager because he refused to refund my booking fee*”
    - Entity: hotel
    - Aspect: manager
    - emotion\_type anger
    - feeler: I
    - time: unknown
  - The definition can also include *the cause*.

# Emotion Expressions

- 1. use emotion or mood words or phrases
  - E.g., love, disgust, angry, and upset
- 2. describe emotion-related behaviors,
  - “He cried after he saw his mother” and
  - “After he received the news, he jumped up and down for a few minutes like a small boy.”
- 3. use intensifiers: Common English intensifiers include
  - very, so, extremely, dreadfully, really, awfully, etc.

# Emotion Expressions (conted.)

- 4. use superlatives –
  - many superlative expressions also express emotions, for example, “*This car is simply the best*”
- 5. use pejorative (e.g., “*He is a fascist.*”), laudatory (e.g., “*He is a saint.*”), and sarcastic expressions (e.g., “*What a great car, it broke the second day*”)
- 6. use swearing, cursing, insulting, blaming, accusing, and threatening expressions

# Roadmap

- Sentiment analysis problem
- Document sentiment classification
- Sentence subjectivity & sentiment classification
- Aspect-based sentiment analysis
- Mining comparative opinions
- Opinion lexicon generation
- Some interesting sentences
- **Summary**

# 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**

# Assumption and goal

- **Assumption:** The doc is written by a single person and express opinion/sentiment on a single entity.
- **Reviews usually satisfy the assumption.**
  - Almost all research papers use reviews
  - Positive: 4 or 5 stars, negative: 1 or 2 stars
- **Forum postings and blogs do not**
  - They may mention and compare multiple entities
  - Many such postings express no sentiments

# 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.

# Features for supervised learning

- The problem has been studied by numerous researchers.
- **Key:** feature engineering. A large set of features have been tried by researchers. E.g.,
  - Terms frequency and different IR weighting schemes
  - Part of speech (POS) tags
  - Opinion words and phrases
  - Negations
  - Syntactic dependency

# Lexicon-based approach (Taboada *et al.* (2011))

- Using a set of sentiment terms, called the **sentiment lexicon**
  - Positive words: great, beautiful, amazing, ...
  - Negative words: bad, terrible awful, unreliable, ...
- The SO value for each sentiment term is assigned a value from [-5, +5].
  - Consider *negation*, *intensifier* (e.g., very), and *diminisher* (e.g., barely)
- Decide the sentiment of a review by aggregating scores from all sentiment terms

# Deep learning

- Recently, deep neural networks have been used for sentiment classification. E.g.,
  - Socher et al (2013) used deep learning to work on the sentence parse tree based on words/phrases compositionality in the framework of distributional semantics
  - Many papers ...
  - Also related
    - Irsoy and Cardie (2014) extract opinion expressions
    - Xu, Liu and Zhao (2014) identify opinion & target relations

# Review rating prediction

- Apart from classification of positive or negative sentiments,
  - research has also been done to **predict the rating scores** (e.g., 1–5 stars) of reviews (Pang and Lee, 2005; Liu and Seneff 2009; Qu, Ifrim and Weikum 2010; Long, Zhang and Zhu, 2010).
  - Training and testing are reviews with star ratings.
- **Formulation:** The problem is formulated as regression since the rating scores are ordinal.
- Again, feature engineering and model building.

# Roadmap

- Sentiment analysis problem
- Document sentiment classification
- Sentence subjectivity & sentiment classification
- Aspect-based sentiment analysis
- Mining comparative opinions
- Opinion lexicon generation
- Some interesting sentences
- **Summary**

# 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."

# Assumption

- **Assumption:** Each sentence is written by a single person and expresses a single positive or negative opinion/sentiment.
- **True for simple sentences**, e.g.,
  - “I like this car”
- **But not true for many compound and “complex” sentences**, e.g.,
  - “I like the picture quality but battery life sucks.”
  - “Apple is doing very well in this poor economy.”

# Subjectivity and sentiment classification

(Yu and Hazivassiloglou, 2003)

- **Subjective sentence identification:** a few methods were tried, e.g.,
  - Sentence similarity.
  - Naïve Bayesian classification.
- **Sentiment classification (positive, negative or neutral)** (also called **polarity**): it uses a similar method to (Turney, 2002), but
  - with more seed words (rather than two) and based on log-likelihood ratio (LLR).
  - For classification of each word, it takes the average of LLR scores of words in the sentence and use cutoffs to decide positive, negative or neutral.

# Segmentation and classification

(Wilson et al 2004)

- Since a single sentence may contain multiple opinions and subjective and factual clauses
- A study of automatic clause sentiment classification was presented in (Wilson et al 2004)
  - to classify clauses of every sentence by the *strength* of opinions being expressed in individual clauses, down to four levels
    - *neutral, low, medium, and high*
- Clause-level may not be sufficient
  - “Apple is doing very well in this lousy economy.”

# Supervised & unsupervised methods

- Numerous papers have been published on using supervised machine learning (Pang and Lee 2008; Liu 2015).
  - Again, deep neural networks have been used (Socher et al 2013) working on the sentence parse tree, words/phrases compositionality in the framework of distributional semantics.
  - Many more papers ...
- Lexicon-based methods have been applied too (e.g., Hu and Liu 2004; Kim and Hovy 2004).

# Roadmap

- Sentiment analysis problem
- Document sentiment classification
- Sentence subjectivity & sentiment classification
- Aspect-based sentiment analysis
- Mining comparative opinions
- Opinion lexicon generation
- Some interesting sentences
- **Summary**

# We need to go further

- Sentiment classification at both the document and sentence (or clause) levels are useful, **but**
  - They do not find what people liked and disliked.
- **They do not identify the targets of opinions**, i.e.,
  - Entities and their aspects
  - Without knowing targets, opinions are of limited use.
- **We need to go to the entity and aspect level.**
  - *Aspect-based opinion mining and summarization* (Hu and Liu 2004).
  - We thus need the full opinion definition.

# Recall the opinion 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*

# 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

# (1) Frequent nouns and noun phrases

(Hu and Liu 2004)

- Nouns (NN) that are frequently mentioned are likely to be true **aspects** (frequent aspects).
- Why?
  - Most aspects are nouns or noun phrases
  - When product aspects/features are discussed, the words they use often converge.
  - Those frequent ones are usually the main aspects that people are interested in.

# Using part-of relationship and the Web

(Popescu and Etzioni, 2005)

- Improved (Hu and Liu, 2004) by removing some frequent noun **phrases** that may not be aspects.
- It identifies **part-of** relationship
  - Each noun **phrase** is given a pointwise mutual information score between the phrase and **part discriminators** associated with the product class, e.g., a scanner class.
  - E.g., “of scanner”, “scanner has”, etc, which are used to find parts of scanners by searching on the Web:

$$PMI(a,d) = \frac{hits(a \sqcap d)}{hits(a)hits(d)},$$

## (2) Exploiting opinion & target relation

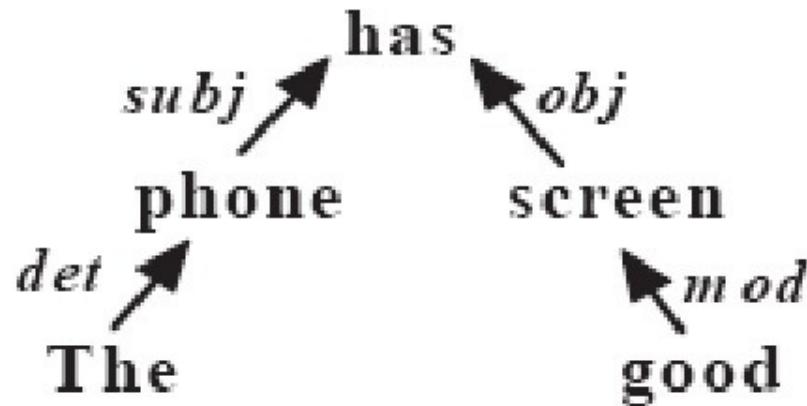
- Key idea: opinions have targets, i.e., opinion terms are used to modify aspects and entities.
  - “The pictures are absolutely amazing.”
  - “This is an amazing software.”
- The syntactic relation is approximated with the nearest noun phrases to the opinion word in (Hu and Liu 2004).
- The idea was generalized to
  - syntactic dependency in (Zhuang et al 2006)
  - double propagation in (Qiu et al 2009). A similar idea also in (Wang and Wang 2008)

# Extract aspects using DP (Qiu et al. 2009; 2011)

- *Double propagation* (DP)
  - Based on the definition earlier, **an opinion should have a target**, entity or aspect.
- Use dependency of opinions & aspects to extract both aspects & opinion words.
  - Knowing one helps find the other.
  - E.g., “*The rooms are spacious*”
- It extracts both aspects and opinion words.
  - A domain independent method.

# The DP method

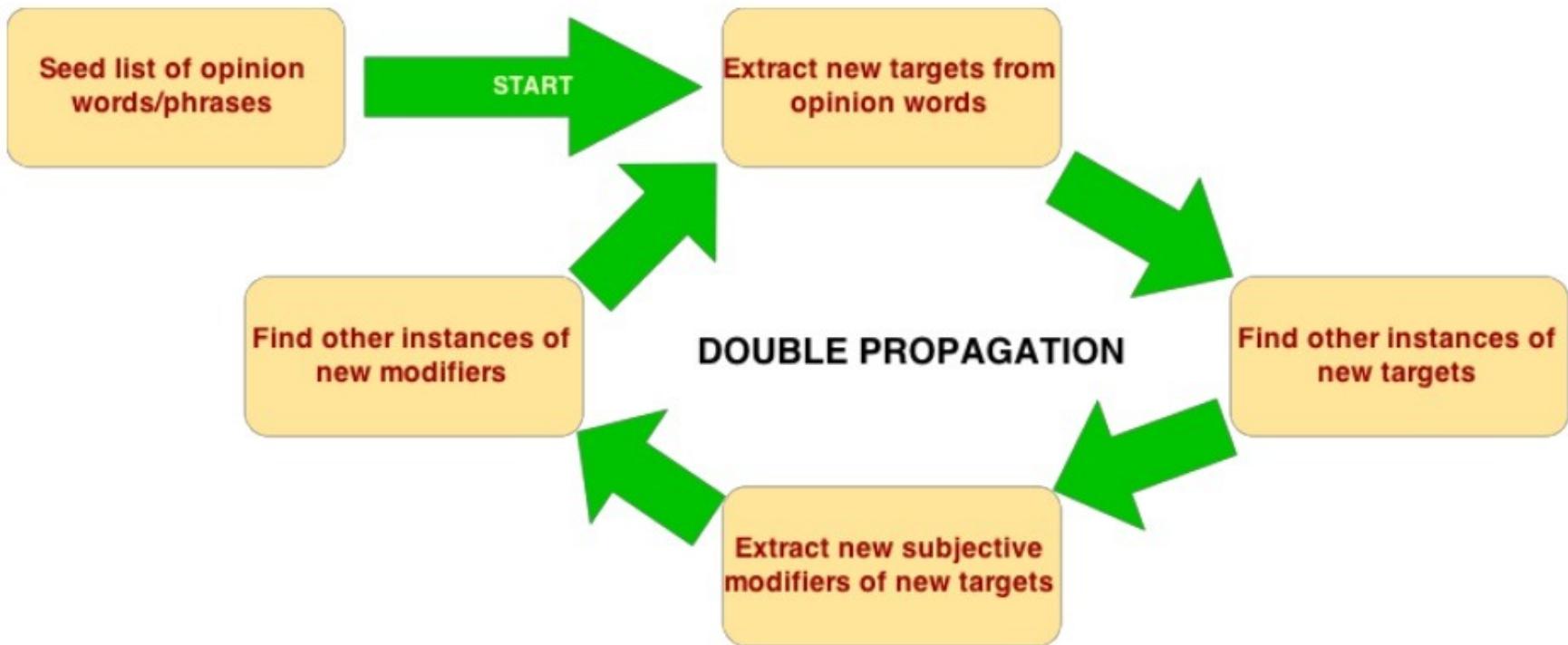
- DP is a bootstrapping method
  - Input: a set of seed opinion words,
  - no aspect seeds needed
- Based on dependency grammar (Tesniere 1959).
  - “This phone has good screen”



# Rules from dependency grammar

	Relations and Constraints	Output	Examples
R1 <sub>1</sub>	$O \rightarrow O\text{-}Dep \rightarrow F$ s.t. $O \in \{O\}$ , $O\text{-}Dep \in \{MR\}$ , $POS(F) \in \{NN\}$	$f = F$	<i>The phone has a <u>good</u> “screen”.</i> $good \rightarrow mod \rightarrow screen$
R1 <sub>2</sub>	$O \rightarrow O\text{-}Dep \rightarrow H \leftarrow F\text{-}Dep \leftarrow F$ s.t. $O \in \{O\}$ , $O/F\text{-}Dep \in \{MR\}$ , $POS(F) \in \{NN\}$	$f = F$	<i>“iPod” is the <u>best</u> mp3 player.</i> $best \rightarrow mod \rightarrow player \leftarrow subj \leftarrow iPod$
R2 <sub>1</sub>	$O \rightarrow O\text{-}Dep \rightarrow F$ s.t. $F \in \{F\}$ , $O\text{-}Dep \in \{MR\}$ , $POS(O) \in \{JJ\}$	$o = O$	same as R1 <sub>1</sub> with <i>screen</i> as the known word and <i>good</i> as the extracted word
R2 <sub>2</sub>	$O \rightarrow O\text{-}Dep \rightarrow H \leftarrow F\text{-}Dep \leftarrow F$ s.t. $F \in \{F\}$ , $O/F\text{-}Dep \in \{MR\}$ , $POS(O) \in \{JJ\}$	$o = O$	same as R1 <sub>2</sub> with <i>iPod</i> is the known word and <i>best</i> as the extract word.
R3 <sub>1</sub>	$F_{i(j)} \rightarrow F_{i(j)}\text{-}Dep \rightarrow F_{j(i)}$ s.t. $F_{j(i)} \in \{F\}$ , $F_{i(j)}\text{-}Dep \in \{CONJ\}$ , $POS(F_{i(j)}) \in \{NN\}$	$f = F_{i(j)}$	<i>Does the player play dvd with <u>audio</u> and “video”?</i> $video \rightarrow conj \rightarrow audio$
R3 <sub>2</sub>	$F_i \rightarrow F_i\text{-}Dep \rightarrow H \leftarrow F_j\text{-}Dep \leftarrow F_j$ s.t. $F_i \in \{F\}$ , $F_i\text{-}Dep = F_j\text{-}Dep$ , $POS(F_j) \in \{NN\}$	$f = F_j$	<i>Canon “G3” has a great <u>len</u>.</i> $len \rightarrow obj \rightarrow has \leftarrow subj \leftarrow G3$
R4 <sub>1</sub>	$O_{i(j)} \rightarrow O_{i(j)}\text{-}Dep \rightarrow O_{j(i)}$ s.t. $O_{j(i)} \in \{O\}$ , $O_{i(j)}\text{-}Dep \in \{CONJ\}$ , $POS(O_{i(j)}) \in \{JJ\}$	$o = O_{i(j)}$	<i>The camera is <u>amazing</u> and “easy” to use.</i> $easy \rightarrow conj \rightarrow amazing$
R4 <sub>2</sub>	$O_i \rightarrow O_i\text{-}Dep \rightarrow H \leftarrow O_j\text{-}Dep \leftarrow O_j$ s.t. $O_i \in \{O\}$ , $O_i\text{-}Dep = O_j\text{-}Dep$ , $POS(O_j) \in \{JJ\}$	$o = O_j$	<i>If you want to buy a <u>sexy</u>, “cool”, accessory-available mp3 player, you can choose iPod.</i> $sexy \rightarrow mod \rightarrow player \leftarrow mod \leftarrow cool$

# The DP method again



# Select the optimal set of rules

(Liu et al., 2015)

- Instead of manually deciding a fixed set of dependency rules/relations as in DP,
  - the paper proposed to select rules automatically.
    - based on rule induction (Liu, Hsu & Ma 1998) in ML.
  - The input has all dependency relations/rules.
  - The system selects an “optimal” subset.
- The selected rule subset performs extraction significantly better.
  - Some rules in DP were actually not selected.

# Extract opinion, target and relation

(Xu, Liu and Zhao, 2014)

- An opinion relation has three components:
  - a correct opinion word, a correct opinion target and the linking relation between them.
- This paper proposed a deep learning approach to identify them.
- Due to the problem of obtaining negative training data,
  - It applied the idea of one-class classification.
  - Final network: One-Class Deep Neural Network

# Explicit and implicit aspects

(Hu and Liu, 2004)

- **Explicit aspects**: Aspects explicitly mentioned as nouns or noun phrases in a sentence
  - “The **picture quality** is of this phone is great.”
- **Implicit aspects**: Aspects not explicitly mentioned in a sentence but are implied
  - “This car is so **expensive**.”
  - “This phone will not easily **fit in a pocket**.”
  - “Included **16MB** is stingy.”
- Some work has been done (Su et al. 2009; Hai et al 2011)

### (3) Using supervised learning

- Using sequence labeling methods such as
  - Hidden Markov Models (HMM) (Jin and Ho, 2009)
  - Conditional Random Fields (Jakob and Gurevych, 2010).
  - Other supervised or partially supervised learning.
- (Liu, Hu and Cheng 2005; Kobayashi et al., 2007; Li et al., 2010; Choi and Cardie, 2010; Yu et al., 2011; Fang and Huang, 2012).

# Identify aspect synonyms (Carenini et al 2005)

- Once aspect expressions are discovered, group them into aspect categories.
  - E.g., **power usage** and **battery life** are the same.
- Method:** based on some similarity metrics, but it needs **a taxonomy of aspects**.
  - Mapping:** The system maps each discovered aspect to an aspect node in the taxonomy.
  - Similarity metrics:** string similarity, synonyms and other distances measured using WordNet.

# Group aspect synonyms (Zhai et al. 2011a, b)

- Unsupervised learning:
  - Clustering: EM-based.
  - Constrained topic modeling: Constrained-LDA
    - By intervening Gibbs sampling.
- A variety of information/similarities are used to cluster aspect expressions into aspect categories.
  - Lexical similarity based on WordNet
  - Distributional information (surrounding words context)
  - Syntactical constraints (sharing words, in the same sentence)

# EM method

- WordNet similarity

$$Jcn(w_1, w_2) = \frac{1}{IC(w_1) + IC(w_2) - 2 \times Res(w_1, w_2)}$$

- EM-based probabilistic clustering

$$P(w_t | c_j) = \frac{1 + \sum_{i=1}^{|D|} N_{ti} P(c_j | d_i)}{|V| + \sum_{m=1}^{|V|} \sum_{i=1}^{|D|} N_{mi} P(c_j | d_i)}$$

$$P(c_j) = \frac{1 + \sum_{i=1}^{|D|} P(c_j | d_i)}{|C| + |D|}$$

$$P(c_j | d_i) = \frac{P(c_j) \prod_{k=1}^{|d_i|} P(w_{d_i,k} | c_j)}{\sum_{r=1}^{|C|} P(c_r) \prod_{k=1}^{|d_i|} P(w_{d_i,k} | c_r)}$$

## (4) Topic Modeling

- Aspect extraction has two tasks:
  - (1) extract aspect expressions
  - (2) cluster them (same: “picture,” “photo,” “image”)
- Top models such as pLSA (Hofmann 1999) and LDA (Blei et al 2003) perform both tasks at the same time. A topic is basically an aspect.
  - A document is a distribution over topics
  - A topic is a distribution over terms/words, e.g.,
    - *{price, cost, cheap, expensive, ...}*
    - Ranked based on probabilities (not shown).

# Many Related Models and Papers

- Use topic models to model aspects.
- Jointly model both aspects and sentiments
- Knowledge-based modeling: Unsupervised models are often insufficient
  - Not producing coherent topics/aspects
  - To tackle the problem, *knowledge-based topic models* have been proposed
    - Guided by user-specified prior domain knowledge.
    - Seed terms or constraints

# 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)

# 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)

# A lexicon-based method (Ding et al. 2008)

- **Input:** A set of opinion words and phrases. A pair  $(a, s)$ , where  $a$  is an aspect and  $s$  is a sentence that contains  $a$ .
- **Output:** whether the opinion on  $a$  in  $s$  is +ve, -ve, or neutral.
- Two steps:
  - Step 1: split the sentence if needed based on BUT words (but, except that, etc).
  - Step 2: work on the segment  $s_f$  containing  $a$ . Let the set of opinion words in  $s_f$  be  $w_1, \dots, w_n$ . Sum up their orientations (1, -1, 0), and assign the orientation to  $(a, s)$  based on:

$$\text{Orientation} = \frac{\sum_{i=1}^n w_i \cdot o}{d(w_i, a)}$$

where  $w_i.o$  is the opinion orientation of  $w_i$ .  $d(w_i, a)$  is the distance from  $a$  to  $w_i$ .

# Sentiment shifters (e.g., Polanyi and Zaenen 2004)

- Sentiment/opinion shifters (also called **valence shifters**) are words and phrases that can shift or change opinion orientations.
  - Negation words like *not*, *never*, *cannot*, etc., are the most common type.
  - Many other words and phrases can also alter opinion orientations. E.g., **modal auxiliary verbs** (e.g., *would*, *should*, *could*, etc)
    - “The brake could be improved.”
- Very complicated, see (Liu 2015)

# Sentiment shifters (contd)

- Some **presuppositional** items can change opinions too, e.g., *barely* and *hardly*
  - “It hardly works.” (comparing to “it works”)
  - It presupposes that better was expected.
- Words like *fail*, *omit*, *neglect* behave similarly,
  - “This camera fails to impress me.”
- Sarcasm changes orientation too
  - “What a great car, it did not start the first day.”
- Jia, Yu and Meng (2009) designed some rules based on parsing to find the scope of negation.

# Basic rules of opinions (Liu, 2010; 2012)

- Opinions/sentiments are governed by many rules, e.g., (many such rules)
  - *Opinion word or phrase*: “I love this car”
    - P ::= a positive opinion word or phrase
    - N ::= an negative opinion word or phrase
  - *Desirable or undesirable facts*: “After my wife and I slept on it for two weeks, I noticed a mountain in the middle of the mattress”
    - P ::= desirable fact
    - N ::= undesirable fact

# Basic rules of opinions

- *Producing and consuming resources and wastes:*  
“This washer uses a lot of water”

PO ::= produce a large quantity of or more resource

- | produce no, little or less waste
- | consume no, little or less resource
- | consume a large quantity of or more waste

NE ::= produce no, little or less resource

- | produce some or more waste
- | consume a large quantity of or more resource
- | consume no, little or less waste

# Basic rules of opinions

- *Producing and consuming resources and wastes:*  
“This washer uses a lot of water”

PO ::= produce a large quantity of or more resource

- | produce no, little or less waste
- | consume no, little or less resource
- | consume a large quantity of or more waste

NE ::= produce no, little or less resource

- | produce some or more waste
- | consume a large quantity of or more resource
- | consume no, little or less waste

# Basic rules of opinions

- *Producing and consuming resources and wastes:*  
“This washer uses a lot of water”

PO ::= produce a large quantity of or more resource

- | produce no, little or less waste
- | consume no, little or less resource
- | consume a large quantity of or more waste

NE ::= produce no, little or less resource

- | produce some or more waste
- | consume a large quantity of or more resource
- | consume no, little or less waste

# Roadmap

- Sentiment analysis problem
- Document sentiment classification
- Sentence subjectivity & sentiment classification
- Aspect-based sentiment analysis
- Mining comparative opinions
- Opinion lexicon generation
- Some interesting sentences
- **Summary**

# Comparative Opinions

(Jindal and Liu, 2006)

- *Gradable*

- *Non-Equal Gradable*: Relations of the type *greater or less than*
  - “*The sound of phone A is better than that of phone B*”
- *Equative*: Relations of the type *equal to*
  - “*Camera A and camera B both come in 7MP*”
- *Superlative*: Relations of the type *greater or less than all others*
  - “*Camera A is the cheapest in market*”

# Analyzing Comparative Opinions

- **Objective:** Given an opinionated document  $d$ ,  
**Extract comparative opinions:**  
 $(E_1, E_2, A, po, h, t)$ ,  
 $E_1$  and  $E_2$ ; entity sets being compared  
 $A$ : their shared aspects - the comparison is based on  
 $po$ : preferred entity set  
 $h$ : opinion holder  
 $t$ : time when the comparative opinion is posted.
- **Note:** not positive or negative opinions.

## An example

- Consider the comparative sentence
  - “*Canon’s optics is better than those of Sony and Nikon.*”
  - Written by John in 2010.
- The extracted comparative opinion/relation:
  - (*{Canon}, {Sony, Nikon}, {optics}, preferred:{Canon}, John, 2010*)

# Common comparatives

- In English, comparatives are usually formed by adding *-er* and superlatives are formed by adding *-est* to their **base adjectives** and **adverbs**
- Adjectives and adverbs with two syllables or more and not ending in *y* do not form comparatives or superlatives by adding *-er* or *-est*.
  - Instead, *more*, *most*, *less*, and *least* are used before such words, e.g., *more beautiful*.
- Irregular comparatives and superlatives, i.e., *more*, *most*, *less*, *least*, *better*, *best*, *worse*, *worst*, etc

# Some techniques (Jindal and Liu, 2006, Ding et al, 2009)

- Identify comparative sentences
  - Supervised learning
- Extraction of different items
  - Label sequential rules
  - Conditional random fields (CRF)
- Determine preferred entities (opinions)
  - Lexicon-based methods: Parsing and opinion lexicon
- (Yang and Ko, 2011) is similar to (Jindal and Liu 2006)

# Analysis of comparative opinions

- Grable comparative sentences can be dealt with *almost* as normal opinion sentences.
  - E.g., “*optics of camera A is better than that of camera B*”
  - Positive: (camera A, *optics*)
  - Negative: (camera B, *optics*)
- **Difficulty:** recognize non-standard comparatives
  - E.g., “*I am so happy because my new iPhone is nothing like my old slow ugly Droid.*”

# Identifying preferred entities

(Ganapathibhotla and Liu, 2008)

- The following rules can be applied

Comparative Negative ::= increasing comparative N  
| decreasing comparative P

Comparative Positive ::= increasing comparative P  
| decreasing comparative N

- E.g., “*Coke tastes better than Pepsi*”
- “*Nokia phone’s battery life is longer than Moto phone*”

- Context-dependent comparative opinion words
  - Using context pair: (aspect, JJ/JJR)
  - Deciding the polarity of (battery\_life, longer) in a corpus

# Roadmap

- Sentiment analysis problem
- Document sentiment classification
- Sentence subjectivity & sentiment classification
- Aspect-based sentiment analysis
- Mining comparative opinions
- Opinion lexicon generation
- Some interesting sentences
- **Summary**

# Sentiment (or opinion) lexicon

- **Sentiment lexicon:** lists of words and expressions used to express people's subjective feelings and sentiments/opinions.
  - Not just individual words, but also phrases and idioms, e.g., "cost an arm and a leg"
- They are instrumental for sentiment analysis.
- There seems to be endless variety of sentiment bearing expressions.
  - We have compiled more than 6,700 individual words.
  - There are also a large number of phrases.

# 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**

# Corpus-based approaches

- Rely on syntactic patterns in large corpora.  
(Hazivassiloglou and McKeown, 1997; Turney, 2002; Yu and Hazivassiloglou, 2003; Kanayama and Nasukawa, 2006; Ding, Liu and Yu, 2008)
  - Can find domain dependent orientations (positive, negative, or neutral).
- (Turney, 2002) and (Yu and Hazivassiloglou, 2003) are similar.
  - Assign opinion orientations (polarities) to words/phrases.
  - (Yu and Hazivassiloglou, 2003) is slightly different from (Turney, 2002)
    - use more seed words (rather than two) and use log-likelihood ratio (rather than PMI).

# Corpus-based approaches (contd)

- **Sentiment consistency:** Use conventions on connectives to identify opinion words (Hazivassiloglou and McKeown, 1997). E.g.,
  - **Conjunction:** conjoined adjectives usually have the same orientation.
    - E.g., “This car is *beautiful* **and** *spacious*.” (conjunction)
  - AND, OR, BUT, EITHER-OR, and NEITHER-NOR have similar constraints.
  - **Learning using**
    - **log-linear model:** determine if two conjoined adjectives are of the same or different orientations.
    - **Clustering:** produce two sets of words: positive and negative

# Context dependent opinion

- Find domain opinion words is insufficient. A word may indicate different opinions in same domain.
  - “The battery life is *long*” (+) and “It takes a *long* time to focus” (-).
- Ding, Liu and Yu (2008) and Ganapathibhotla and Liu (2008) exploited sentiment consistency (both inter and intra sentence) based on contexts
  - It finds context dependent opinions.
  - Context: (adjective, aspect), e.g., (long, battery\_life)
  - It assigns an opinion orientation to the pair.

# The Double Propagation method

(Qiu et al 2009, 2011)

- The same DP method can also use dependency of opinions & aspects to extract new opinion words.
- Based on dependency relations
  - Knowing an aspect can find the opinion word that modifies it
    - E.g., “The **rooms** are **spacious**”
  - Knowing some opinion words can find more opinion words
    - E.g., “The **rooms** are **spacious** and **beautiful**”

# Opinions implied by objective terms

- Most opinion words are “subjective words,” e.g., good, bad, hate, love, and crap.
- But objective nouns can imply opinions too.
  - E.g., “After sleeping on the mattress for one month, a **valley/body impression** has formed in the middle.”
- Resource usage descriptions may also imply opinions (as mentioned in rules of opinions)
  - E.g., “This washer uses a lot of water.”
- See (Zhang and Liu, 2011a; 2011b) for details.

# Dictionary-based methods

- Typically use WordNet's synsets and hierarchies to acquire opinion words
  - Start with a small seed set of opinion words.
  - Bootstrap the set by searching for synonyms and antonyms in WordNet iteratively (Hu and Liu, 2004; Kim and Hovy, 2004; Kamps et al 2004).
- Use additional information (e.g., glosses) from WordNet (Andreevskaia and Bergler, 2006) and learning (Esuti and Sebastiani, 2005). (Dragut et al 2010) uses a set of rules to infer orientations.

# Semi-supervised learning

(Esuti and Sebastiani, 2005)

- Use supervised learning
  - Given two seed sets: positive set P, negative set N
  - The two seed sets are then expanded using synonym and antonymy relations in an online dictionary to generate the expanded sets P' and N'.
- P' and N' form the training sets.
- Using all the glosses in a dictionary for each term in P' ↗ N' and converting them to a vector
- Build a binary classifier
  - Tried various learners.

# 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.

# Roadmap

- Sentiment analysis problem
- Document sentiment classification
- Sentence subjectivity & sentiment classification
- Aspect-based sentiment analysis
- Mining comparative opinions
- Opinion lexicon generation
- Some interesting sentences
- **Summary**

# Some interesting sentences

- “Trying out Chrome because Firefox keeps crashing.”
  - Firefox - negative; no opinion about chrome.
  - We need to segment the sentence into clauses to decide that “crashing” only applies to Firefox(?).
- But how about these
  - “I changed to Audi because BMW is so expensive.”
  - “I did not buy BWM because of the high price.”
  - “I am so happy that my iPhone is nothing like my old ugly Droid.”

# Some interesting sentences (contd)

- Sarcastic sentences
  - “What a great car, it stopped working in the second day.”
- Sarcastic sentences are common in political blogs, comments and discussions.
  - They make political opinions difficult to handle
- Some initial work by (Tsur, et al. 2010)

# Some interesting sentences (contd)

- Sarcastic sentences
  - “What a great car, it stopped working in the second day.”
- Sarcastic sentences are common in political blogs, comments and discussions.
  - They make political opinions difficult to handle
- Some initial work by (Tsur, et al. 2010)

# Some interesting sentences (contd)

- Sarcastic sentences
  - “What a great car, it stopped working in the second day.”
- Sarcastic sentences are common in political blogs, comments and discussions.
  - They make political opinions difficult to handle
- Some initial work by (Tsur, et al. 2010)

# Some more interesting sentences

- “My goal is to get a tv with good picture quality”
- “The top of the picture was brighter than the bottom.”
- “When I first got the airbed a couple of weeks ago it was wonderful as all new things are, however as the weeks progressed I liked it less and less.”
- “Google steals ideas from Bing, Bing steals market shares from Google.”

# Opinion mining is hard!

- “This past Saturday, I bought a Nokia phone and my girlfriend bought a Motorola phone with Bluetooth. We called each other when we got home. The voice on my phone was not so clear, worse than my previous Samsung phone. The battery life was short too. My girlfriend was quite happy with her phone. I wanted a phone with good sound quality. So my purchase was a real disappointment. I returned the phone yesterday.”

# Roadmap

- Sentiment analysis problem
- Document sentiment classification
- Sentence subjectivity & sentiment classification
- Aspect-based sentiment analysis
- Mining comparative opinions
- Opinion lexicon generation
- Some interesting sentences
- **Summary**

# Summary

- This chapter presented
  - The problem of sentiment analysis
    - It provides a structure to the unstructured text.
  - Main research directions and their representative techniques.
- Still many problems not attempted or studied.
- None of the subproblems is solved.

## Summary (contd)

- It is a fascinating NLP or text mining problem.
  - Every sub-problem is highly challenging.
  - But it is also restricted (semantically).
- Despite the challenges, applications are flourishing!
  - It is useful to every organization and individual.
- The general NLU is probably too hard, but can we solve this highly restricted problem?
  - We have a good chance.



**BITS** Pilani  
Pilani Campus

# Social Media Analytics: Information Extraction

Dr. Prasad Ramanathan

[p\\_ramanathan@wilp.bits-pilani.ac.in](mailto:p_ramanathan@wilp.bits-pilani.ac.in)

# Acknowledgments

---

- Course material from the following sources are gratefully acknowledged:
  - [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

# 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)

# Named Entity Types

Type	Tag	Sample Categories
People	PER	People, characters
Organization	ORG	Companies, teams
Location	LOC	Regions, mountains, seas
Geopolitical Entity	GPE	Countries, provinces
Facility	FAC	Bridges, buildings, airports
Vehicle	VEH	Planes, trains, automobiles

# 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 <sub>PER</sub> ]	O	[B <sub>PER</sub> ]	O	[B <sub>ORG</sub> ]	[I <sub>ORG</sub> ]	[I <sub>ORG</sub> ]

,	is	a	professor
O	O	O	O

# Feature-Based Named Entity



- ▶ Word-based features

## This word

Identity  
Embedding  
POS

Base-phrase label (IOB tag)

Presence in a gazetteer (list of place names)

## Neighboring Words

Identity  
Embedding  
POS

Base-phrase label (IOB tag)

- ▶ Character-based features, geared toward unknown words

## This word

Specific prefix up to length 4

Specific suffix up to length 4

All upper case

Hyphenated

Word shape

Short word shape

## Neighboring Words

Word shape  
Short word shape

# Word Shape & Short Word Shape

---

- ▶ Word shape: a pattern based on the symbols in a word
  - ▶ Map upper case letter to X
  - ▶ Map lower case letter to x
  - ▶ Digit to d
  - ▶ Retain hyphens, apostrophes, periods
  - ▶ L'Occitane  $\Rightarrow X'Xxxxxxx (X'Xx^7)$
  - ▶ DC10-30  $\Rightarrow XXdd-dd (X^2d^2-d^2)$
  - ▶ I.M.F.  $\Rightarrow X.X.X.$
- ▶ Short word shape: reduce consecutive character types to one
  - ▶ L'Occitane  $\Rightarrow X'Xx$
  - ▶ DC10-30  $\Rightarrow Xd-d$
  - ▶ I.M.F.  $\Rightarrow X.X.X.$

# Computing NER

- ▶ Sequence labeling via
  - ▶ Neural models
  - ▶ Maximum Entropy Markov Models (logistic regression plus Viterbi)
  - ▶ Both rely of inputs such as
    - ▶ Features of current, preceding, and following words
    - ▶ Labels of preceding words
- ▶ Rules: multiple passes each seeking to improve recall
  - ▶ High-precision rules for unambiguous names
  - ▶ Substrings of identified names
  - ▶ Domain-specific name lists
  - ▶ Sequence labeling (probabilistic, as above) to complete the list

# 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)

# Generic Relations

Relation	Type Pair	Example
Physical:Located	PER-GPE	IBM, head-quartered in Armonk NY,
Part:Whole:Subsidiary	ORG-ORG	XYZ, the parent of ABC,
Person:Social:Family	PER-PER	Clinton's daughter, Chelsea
Org-	PER-ORG	Microsoft founder, Bill Gates,
Affiliation:Founder		

# Structured Information on the Web



- ▶ Wikipedia Infoboxes
  - ▶ Provide structure for facts suited to a given entry
  - ▶ Structured facts are relations
- ▶ Resource Description Framework (RDF), a W3C recommendation (standard)
  - ▶ Expresses statements as triples in the form of
  - ▶ Subject, Predicate, Object
- ▶ Crowdsourced ontologies such as DBpedia
- ▶ WordNet: to be discussed later
- ▶ Infoboxes in web search results: provided by a webmaster

# Lexico-Syntactic Patterns

A hypernym describes a more broad term, for example cutlery, or dog. A hyponym is a more specialised and specific word, for example: spoon would be a hyponym of cutlery and labrador would be a hyponym of dog.

- ▶ (Hearst patterns) Hyponym relations are often apparent in the syntax
  - ▶ Seeing “A, such as B, ...”
  - ▶ We can conclude that B is a hyponym of A
- ▶ Coordination applies naturally by forcing type agreement
  - ▶ Seeing “A, such as B and C, ...”
  - ▶ We can conclude that B is a hyponym of A
  - ▶ We can conclude that C is a hyponym of A
- ▶ Key idea: identify lexical markers of hyponym-hypernym relations
  - ▶ Including
  - ▶ Especially: Z, especially X, ...
  - ▶ And other: X, Y, and other Zs,

# Regular Expressions as Generalized Patterns



- ▶ per, position of org
  - ▶ Relates the instance of person as holder of the specified position in the referenced organization instance
  - ▶ [<sub>PER</sub> George Marshall], [<sub>POSITION</sub> Secretary of State] of the [<sub>ORG</sub> United States]
- ▶ per (named| appointed| . . . ) per (Prep?) position
  - ▶ [<sub>PER</sub> Truman] appointed [<sub>PER</sub> Marshall] [<sub>POSITION</sub> Secretary of State]

## 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  $M_2$

# 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)

# 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

# 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

# Demos

---

- Info Extraction-for-soc-sci\_v1.ipynb
- Aspect Based Sentiment Analysis.ipynb
- Aspect Based Senti Analysis\_Transformers version.ipynb

# Additional Study Material

---

- [How Search Engines like Google Retrieve Results: Introduction to Information Extraction using Python and spaCy](#)
- [Hands-on NLP Project: A Comprehensive Guide to Information Extraction using Python](#)



**BITS** Pilani  
Pilani Campus

# Questions?





**BITS** Pilani  
Pilani Campus

# Social Media Analytics: Graph Essentials

Dr. Prasad Ramanathan

[p\\_ramanathan@wilp.bits-pilani.ac.in](mailto:p_ramanathan@wilp.bits-pilani.ac.in)

# Acknowledgment

---

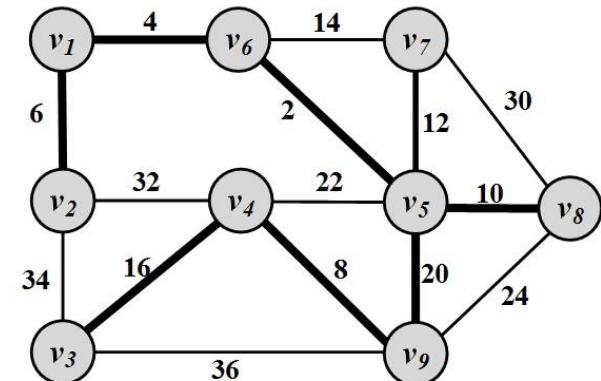
- Course material from the following source is gratefully acknowledged:

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/>**

- A network is a graph.
  - Elements of the network have meanings
- Network problems can usually be represented in terms of graph theory

## Twitter example:

- Given a piece of information, a network of individuals, and the cost to propagate information among any connected pair, find the minimum cost to disseminate the information to all individuals.

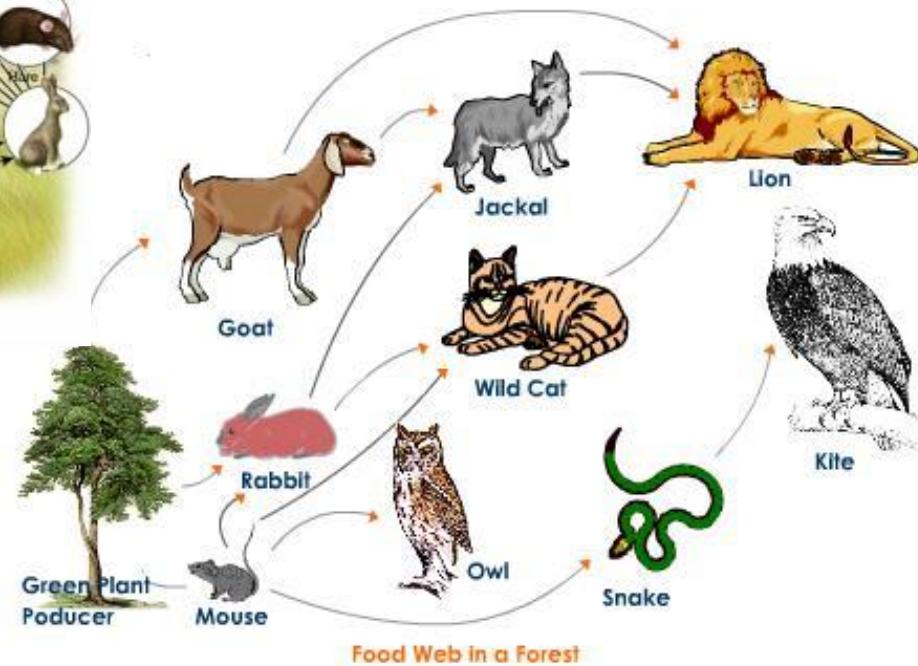
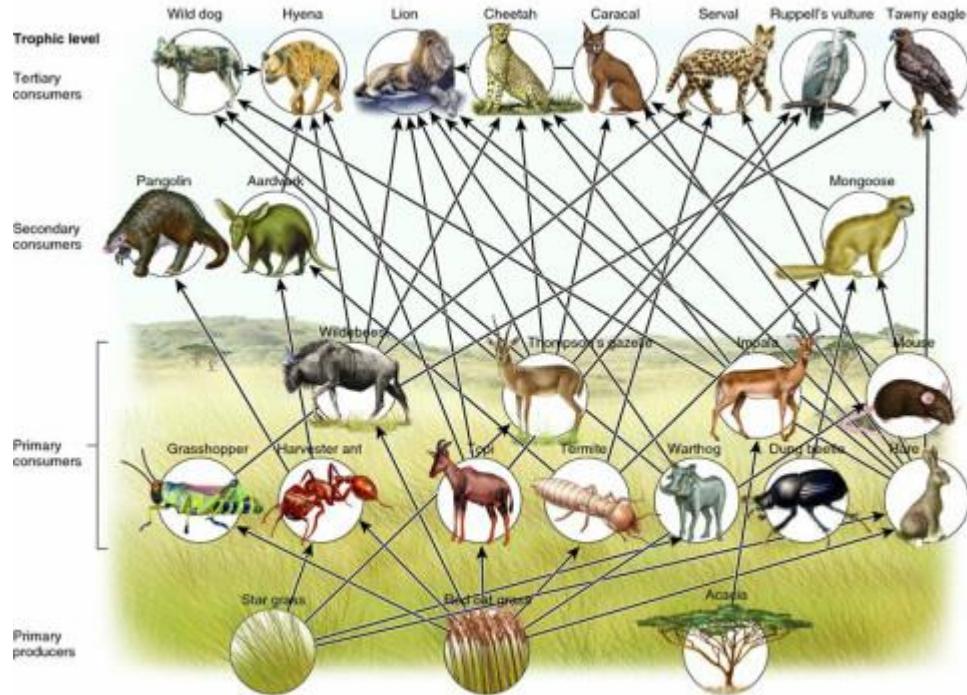


# Food Web

innovate

achieve

lead



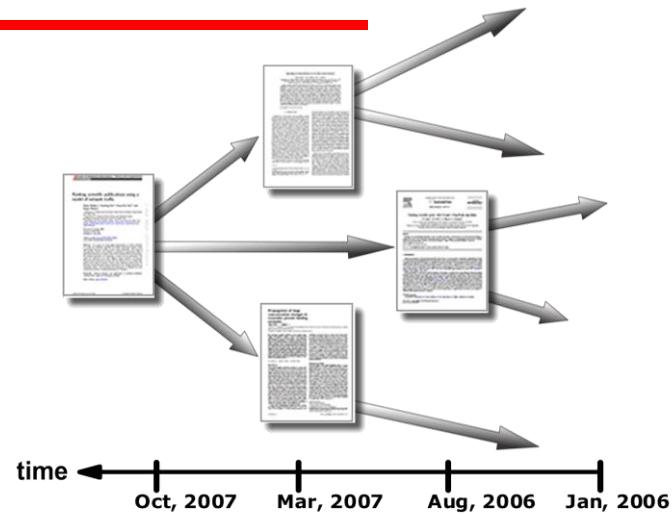
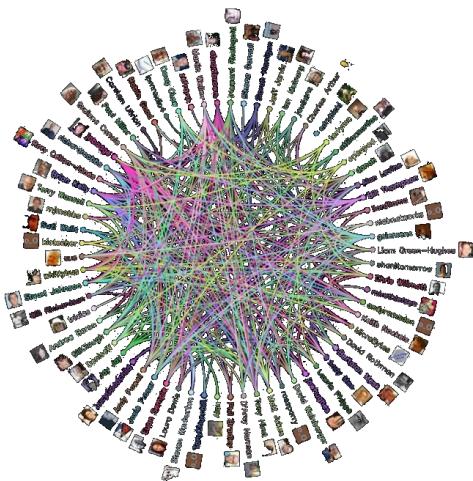
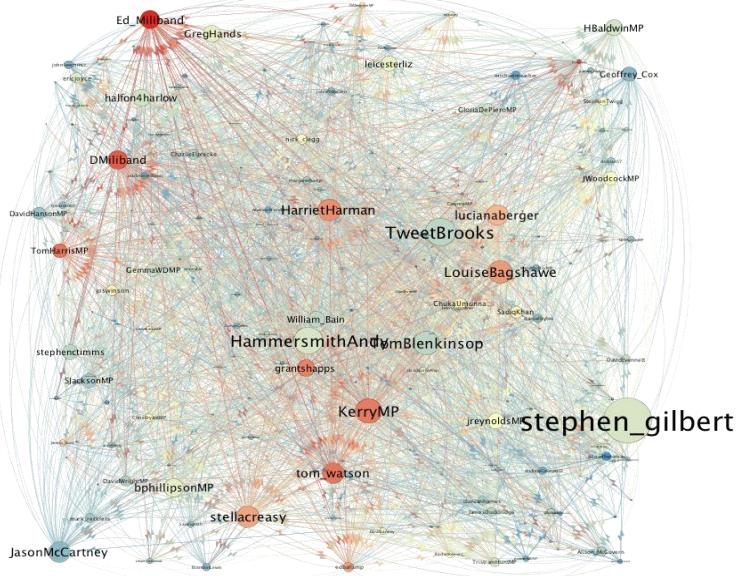
# Networks are Pervasive

innovate

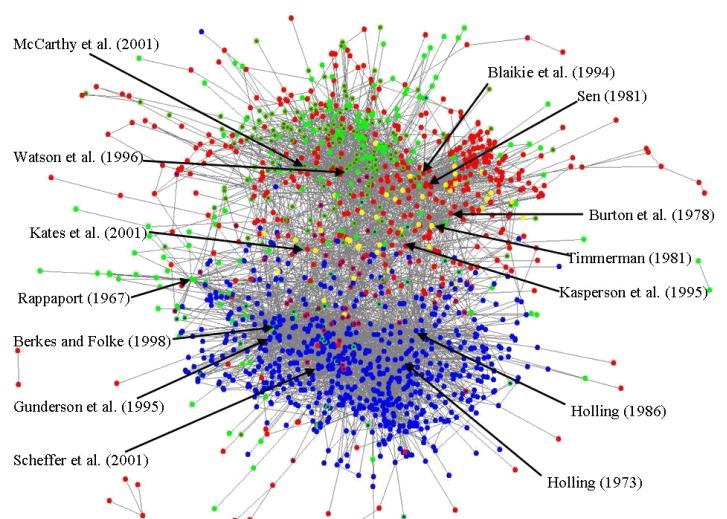
achieve

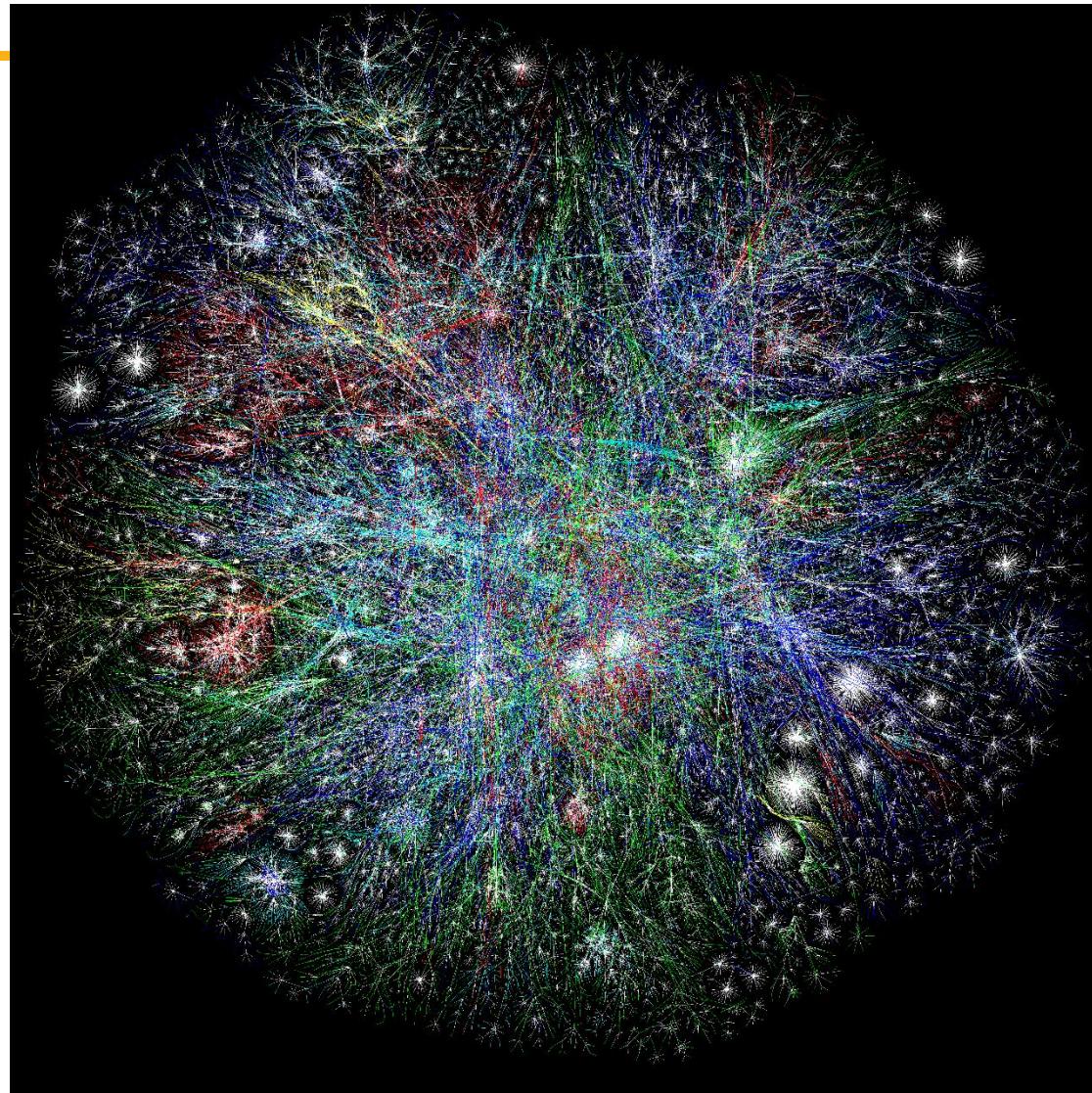
lead

## Twitter Networks



## Citation Networks





# Network of the US Interstate Highways



A network of interstates



# NY State Road Network

innovate

achieve

**lead**



# Social Networks and Social Network Analysis

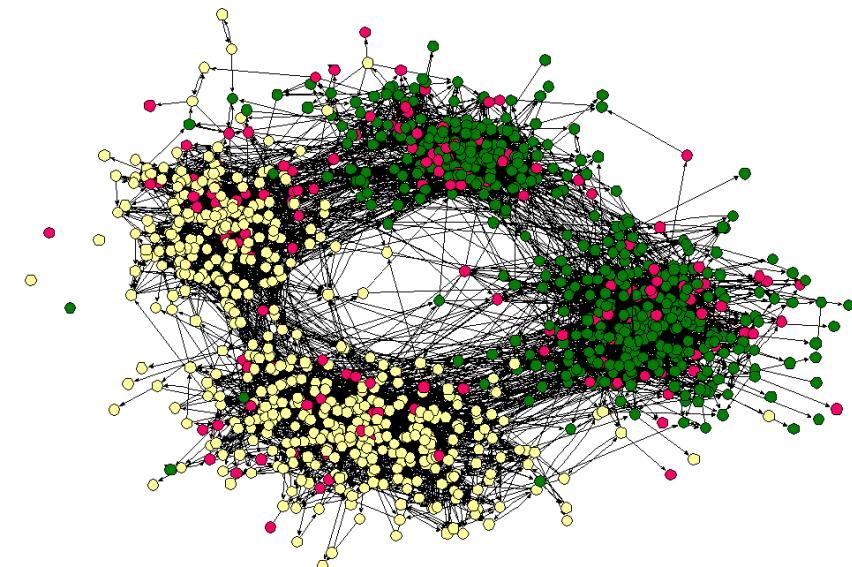


- A social network
  - A network where elements have a social structure
    - A set of **actors** (such as individuals or organizations)
    - A set of **ties** (connections between individuals)
- Social networks examples:
  - your family network, your friend network, your colleagues ,etc.
- To analyze these networks we can use **Social Network Analysis** (SNA)
- Social Network Analysis is an interdisciplinary field from social sciences, statistics, graph theory, complex networks, and now computer science

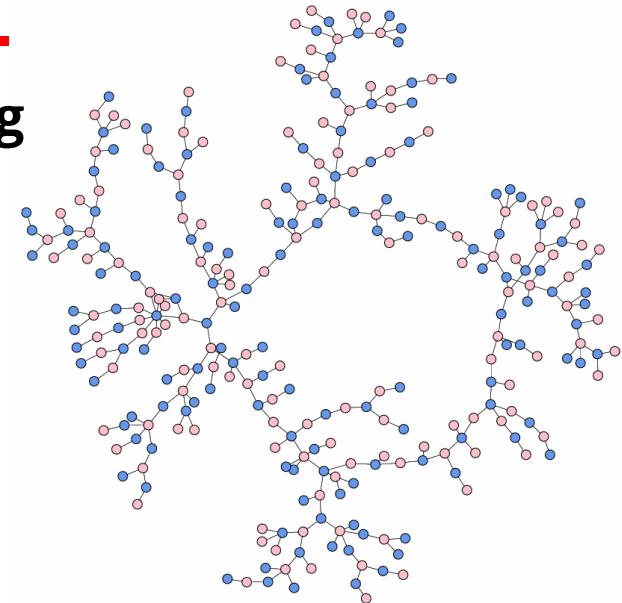
# Social Networks: Examples



High school dating



High school friendship

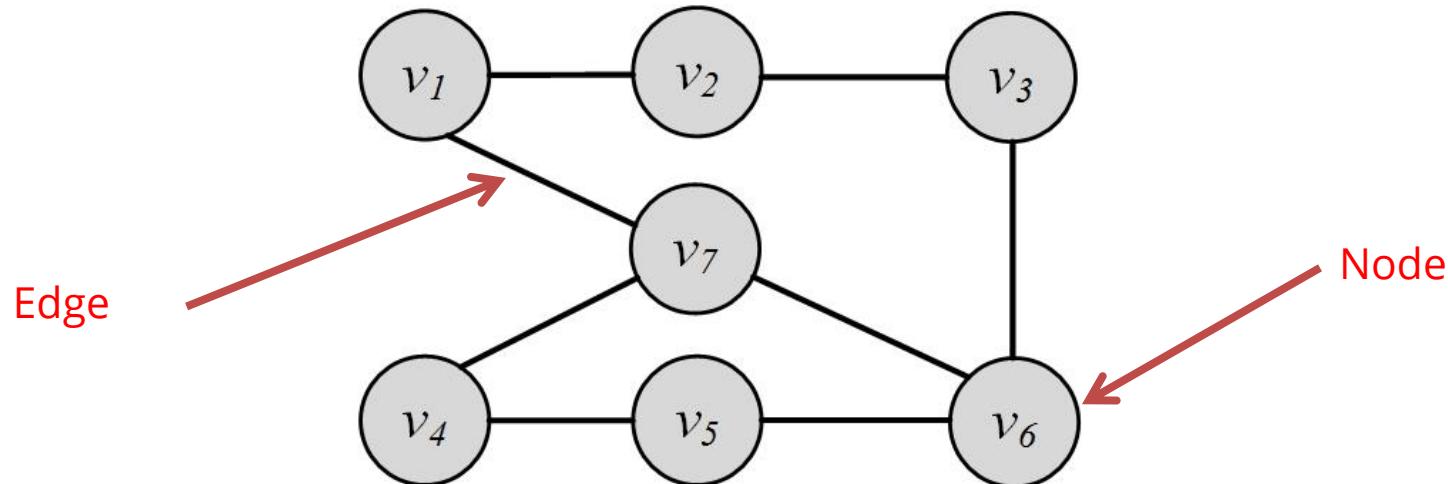




# Graph Basics

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**



- In a friendship social graph, nodes are people and any pair of people connected denotes the friendship between them
- Depending on the context, these nodes are called nodes, or actors
  - In a web graph, “*nodes*” represent sites and the connection between nodes indicates web-links between them
  - In a social setting, these nodes are called actors

$$V = \{v_1, v_2, \dots, v_n\}$$

- The size of the graph is  $|V| = n$

- Edges connect nodes and are also known as **ties** or **relationships**
- In a social setting, where nodes represent social entities such as people, edges indicate internode relationships and are therefore known as relationships or (social) ties

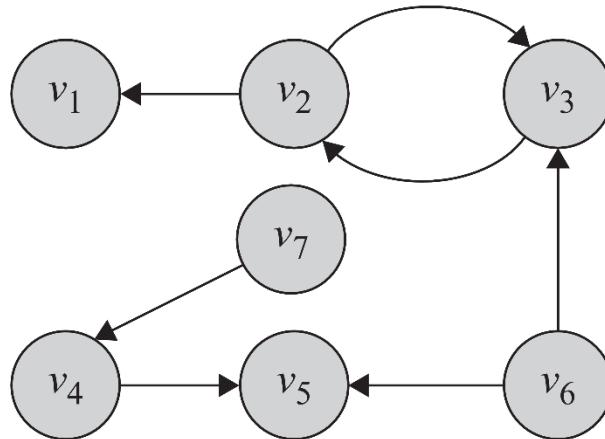
$$E = \{e_1, e_2, \dots, e_m\}$$

- Number of edges (size of the edge-set) is denoted as  $|E| = m$

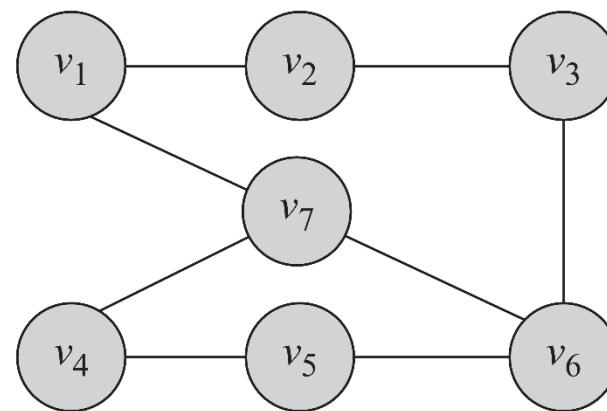
# Directed Edges and Directed Graphs



- Edges can have directions. A directed edge is sometimes called an **arc**



(a) Directed Graph



(b) Undirected Graph

- Edges are represented using their end-points  $e(v_2, v_1)$
- In undirected graphs both representations are the same

# 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

- **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,

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

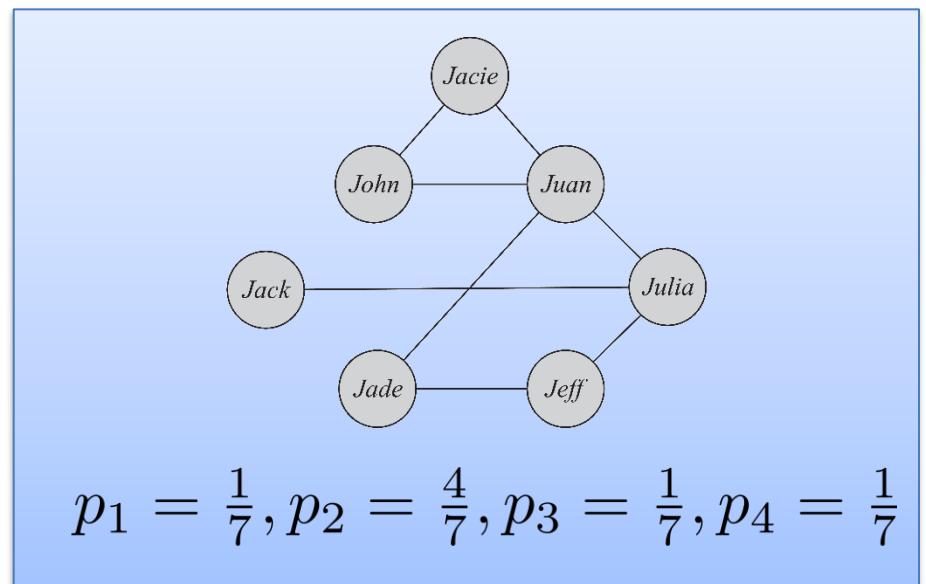
When dealing with very large graphs, how nodes' degrees are distributed is an important concept to analyze and is called ***Degree Distribution***

$$\pi(d) = \{d_1, d_2, \dots, d_n\} \quad (\text{Degree sequence})$$

$$p_d = \frac{n_d}{n}$$

$n_d$  is the number of nodes with degree  $d$

$$\sum_{d=0}^{\infty} p_d = 1$$



# Degree Distribution Plot

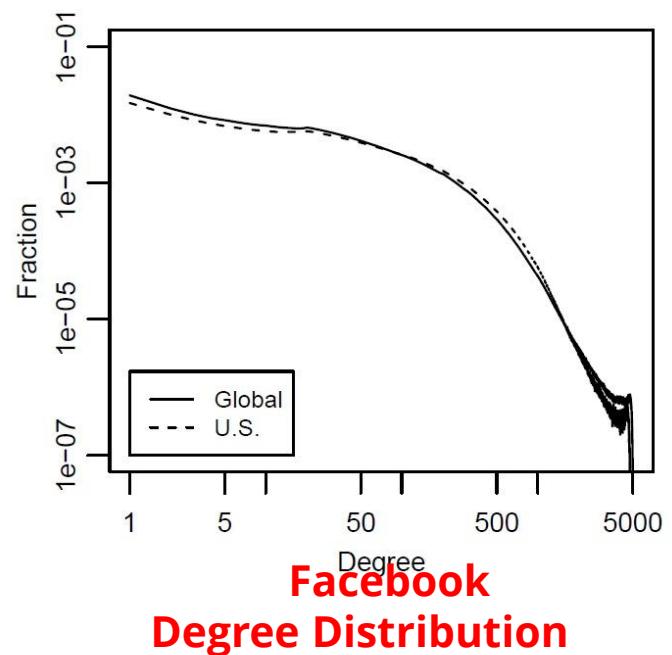


The  $x$ -axis represents the degree and the  $y$ -axis represents the fraction of nodes having that degree

- On social networking sites

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

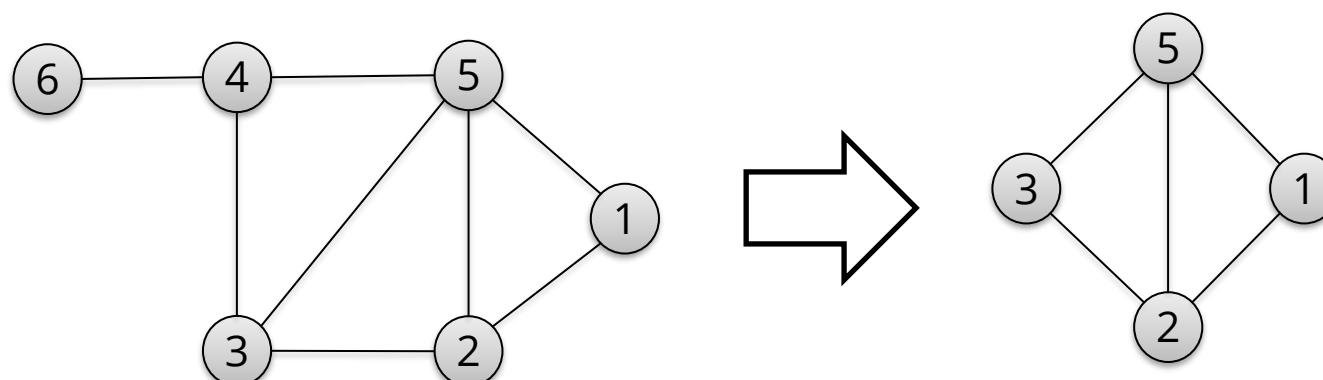
**(Power-law degree distribution)**



- Graph  $G$  can be represented as a pair  $G(V, E)$  where  $V$  is the node set and  $E$  is the edge set
- $G'(V', E')$  is a subgraph of  $G(V, E)$

$$V' \subseteq V$$

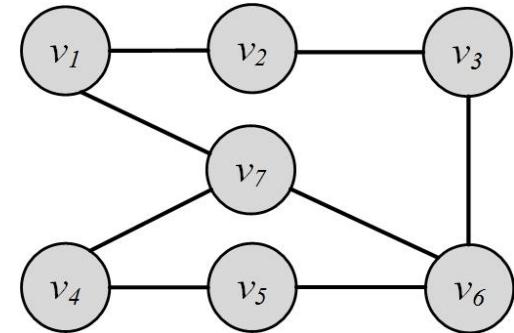
$$E' \subseteq (V' \times V') \cap E$$



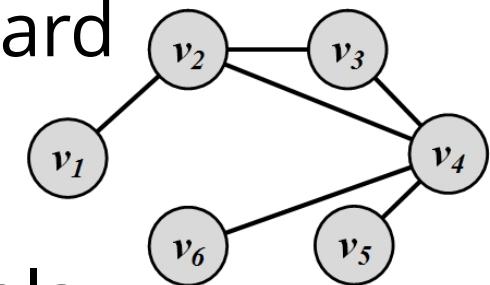


# Graph Representation

- Adjacency Matrix
- Adjacency List
- Edge List



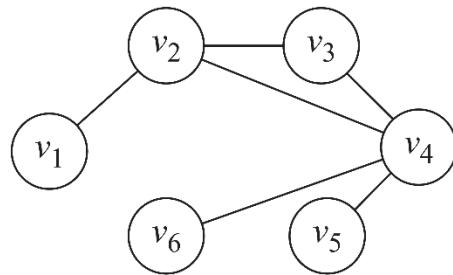
- Graph representation is straightforward and intuitive, but it cannot be effectively manipulated using mathematical and computational tools
- We are seeking representations that can store these two sets in a way such that
  - Does not lose information
  - Can be manipulated easily by computers
  - Can 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

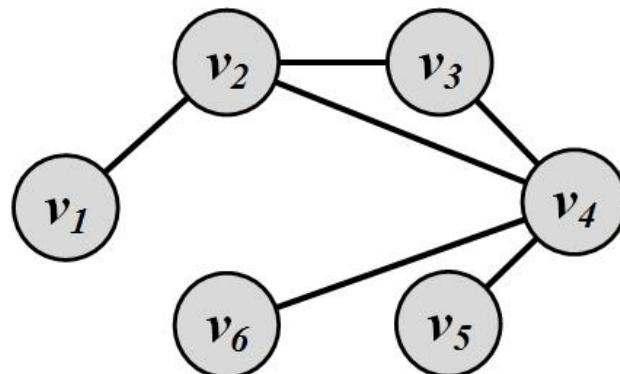
	v <sub>1</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>5</sub>	v <sub>6</sub>
v <sub>1</sub>	0	1	0	0	0	0
v <sub>2</sub>	1	0	1	1	0	0
v <sub>3</sub>	0	1	0	1	0	0
v <sub>4</sub>	0	1	1	0	1	1
v <sub>5</sub>	0	0	0	1	0	0
v <sub>6</sub>	0	0	0	1	0	0

(b) Adjacency Matrix

Diagonal Entries are self-links or loops

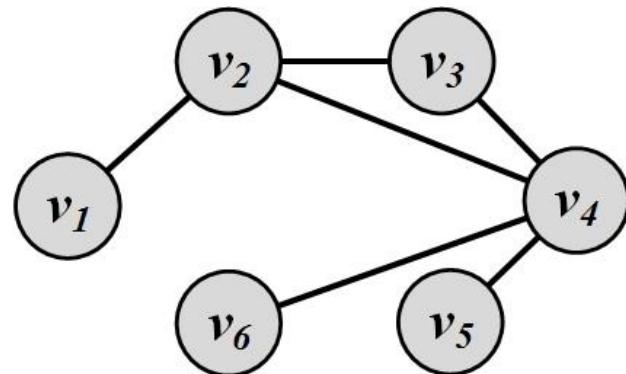
Social media networks have  
very **sparse** Adjacency matrices

- 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
$v_1$	$v_2$
$v_2$	$v_1, v_3, v_4$
$v_3$	$v_2, v_4$
$v_4$	$v_2, v_3, v_5, v_6$
$v_5$	$v_4$
$v_6$	$v_4$

- 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



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



# Types of Graphs

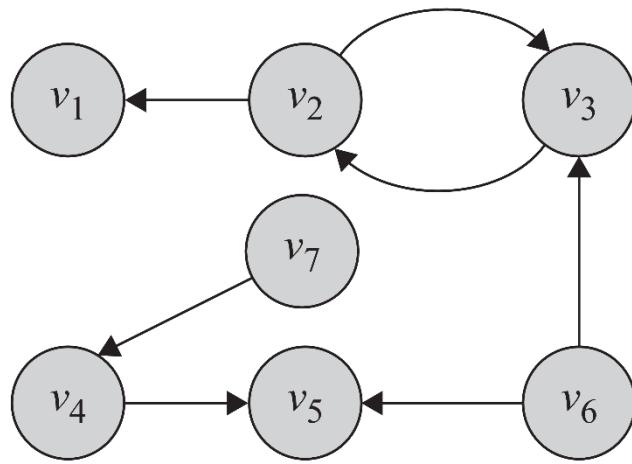
- Null, Empty, Directed/Undirected/Mixed, Simple/Multigraph, Weighted, Signed Graph, Webgraph

- A **null graph** is one where the node set is empty (there are no nodes)
  - Since there are no nodes, there are also no edges

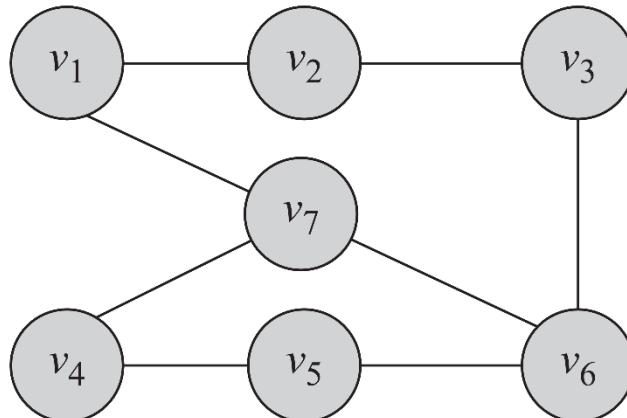
$$G(V, E), V = E = \emptyset$$

- An **empty graph** or **edge-less graph** is one where the edge set is empty,  $E = \emptyset$
- The node set can be non-empty.
  - A null-graph is an empty graph.

# Directed/Undirected/Mixed Graphs



- The adjacency matrix for directed graphs is often not symmetric ( $A \neq A^T$ )
  - $A_{ij} \neq A_{ji}$
  - We can have equality though

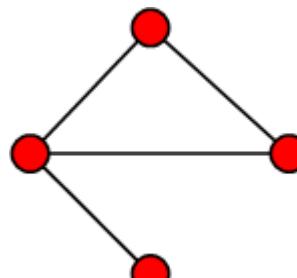


The adjacency matrix for undirected graphs is symmetric ( $A = A^T$ )

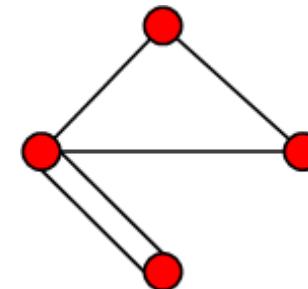
# Simple Graphs and Multigraphs



- Simple graphs are graphs where only a single edge can be between any pair of nodes
- Multigraphs are graphs where you can have multiple edges between two nodes and loops



Simple graph



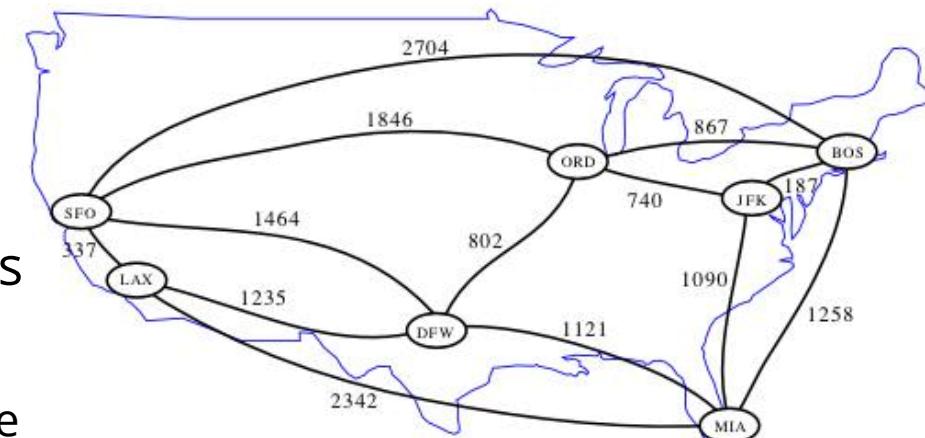
Multigraph

- The adjacency matrix for multigraphs can include numbers larger than one, indicating multiple edges between nodes

# Weighted Graph

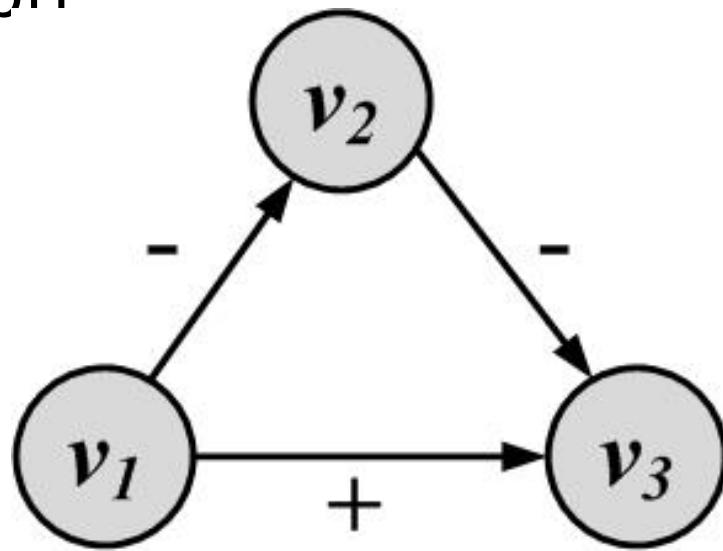


- A weighted graph  $G(V, E, W)$  is one where edges are associated with weights
  - For example, a graph could represent a map where nodes are airports and edges are routes between them
    - The weight associated with each edge could represent the distance between the corresponding cities



$$A_{ij} = \begin{cases} w_{ij} \text{ or } w(i, j), w \in \mathbb{R} \\ 0, \text{ There is no edge between } v_i \text{ and } v_j \end{cases}$$

- When weights are binary (0/1, -1/1, +/-) we have a **signed graph**



- It is used to represent **friends** or **foes**
- It is also used to represent **social status**

- A webgraph is a way of representing how internet sites are connected on the web
- In general, a web graph is a directed multigraph
- Nodes represent sites and edges represent links between sites.
- Two sites can have multiple links pointing to each other and can have loops (links pointing to themselves)



# Connectivity in Graphs

- **Adjacent nodes/Edges,  
Walk/Path/Trail/Tour/Cycle**

# Adjacent nodes and Incident Edges



Two nodes are **adjacent** if they are connected via an edge.

Two edges are **incident**, if they share an endpoint

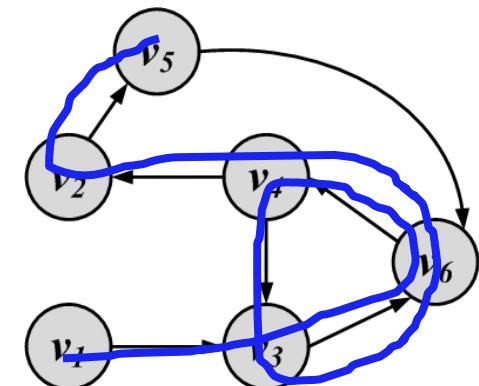
When the graph is directed, edge directions must match for edges to be incident

An edge in a graph can be traversed when one starts at one of its end-nodes, moves along the edge, and stops at its other end-node.

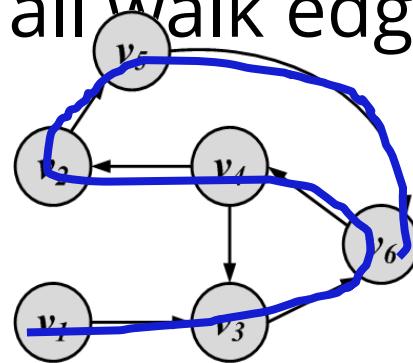
**Walk:** A walk is a sequence of incident edges visited one after another

- **Open walk:** A walk does not end where it starts
  - **Closed walk:** A walk returns to where it starts
- 
- Representing a walk:
    - A sequence of edges:  $e_1, e_2, \dots, e_n$
    - A sequence of nodes:  $v_1, v_2, \dots, v_n$
  - Length of walk:  
the number of visited edges

Length of walk= 8

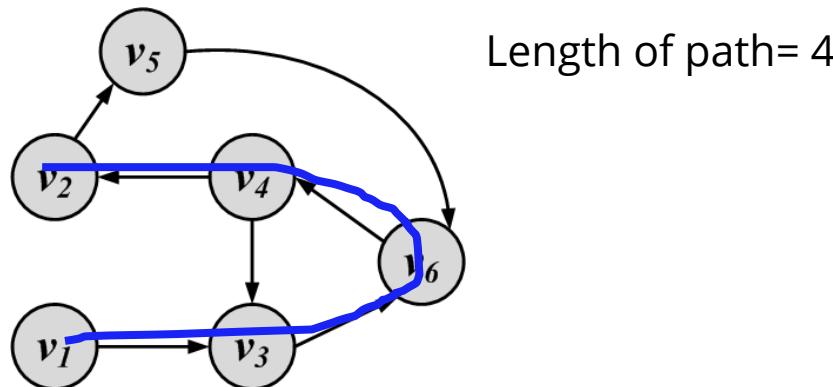


- A trail is a walk where **no edge is visited more than once** and all walk edges are distinct



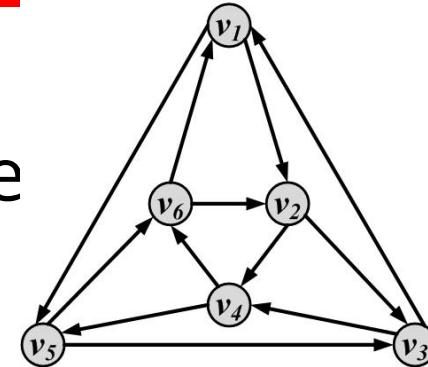
- A closed trail (one that ends where it starts) is called a **tour** or **circuit**

- A walk where **nodes and edges are distinct** is called a **path** and a closed path is called a **cycle**
- The length of a path or cycle is the number of edges visited in the path or cycle



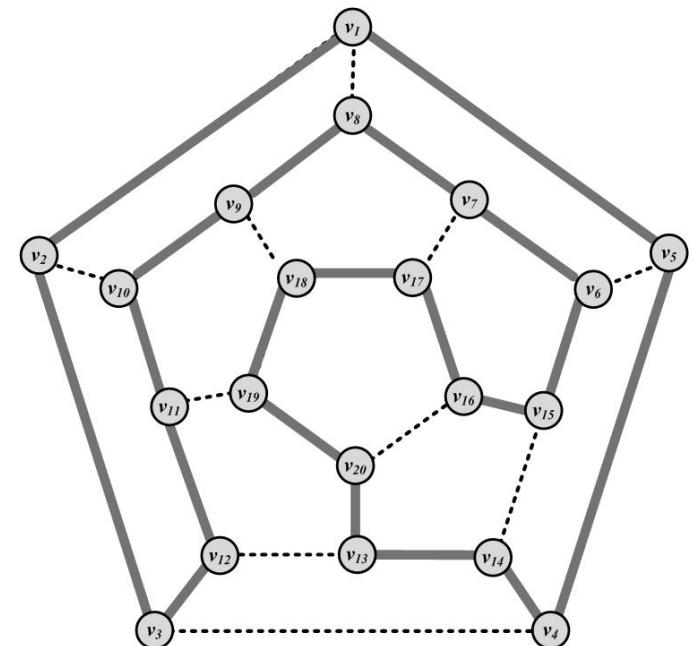
## Eulerian Tour

- All edges are traversed only once
  - Konigsberg bridges



## Hamiltonian Cycle

- A cycle that visits all nodes

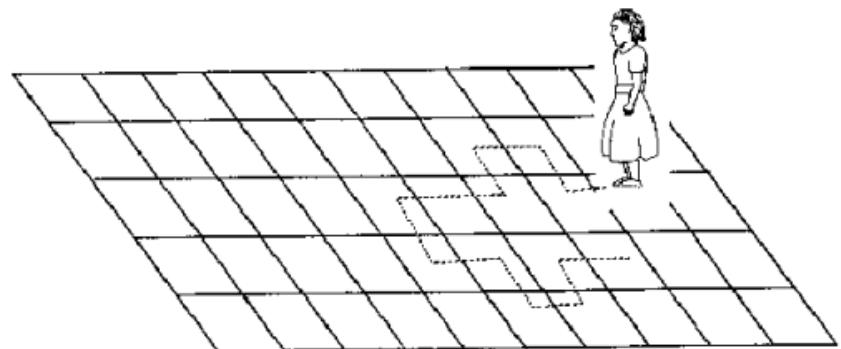


- A walk that in each step the next node is selected randomly among the neighbors
  - The weight of an edge can be used to define the probability of visiting it
  - For all edges that start at  $v_i$  the following equation holds

$$\sum_x w_{i,x} = 1, \forall i, j \quad w_{i,j} \geq 0$$

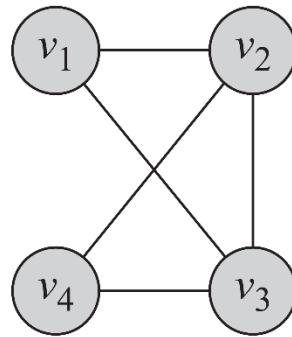
## Mark a spot on the ground

- Stand on the spot and flip the coin (or more than one coin depending on the number of choices such as left, right, forward, and backward)
- If the coin comes up heads, turn to the right and take a step
- If the coin comes up tails, turn to the left and take a step
- Keep doing this many times and see where you end up

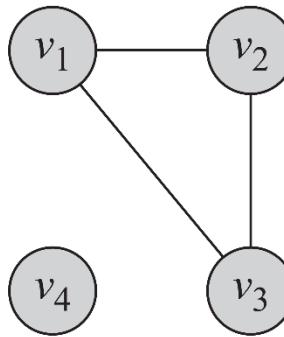


- A node  $v_i$  is connected to node  $v_j$  (or reachable from  $v_j$ ) if it is adjacent to it or there exists a path from  $v_i$  to  $v_j$ .
- A graph is connected, if there exists a path between any pair of nodes in it
  - In a directed graph, a graph is strongly connected if there exists a directed 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.

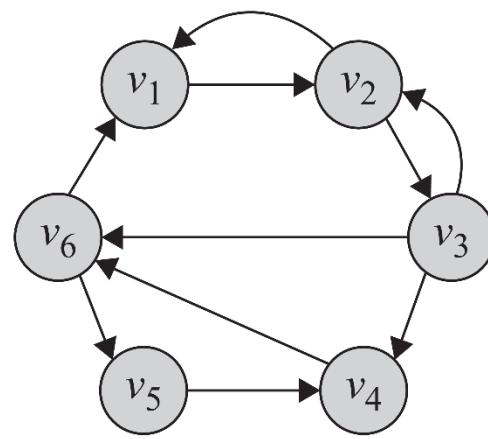
# Connectivity: Example



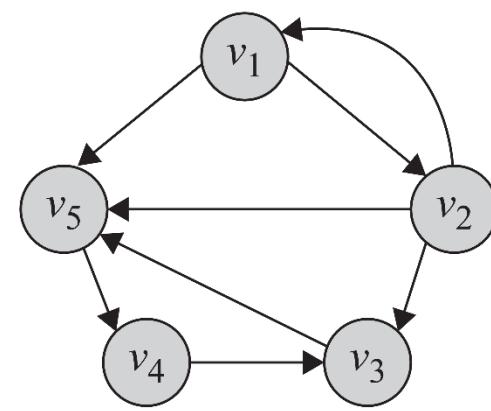
(a) Connected



(b) Disconnected



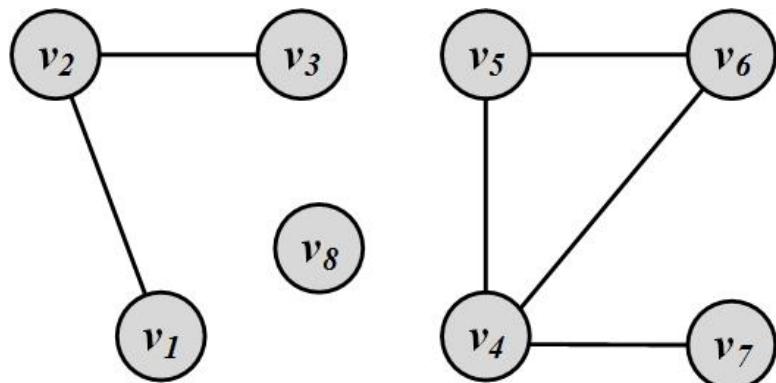
(c) Strongly connected



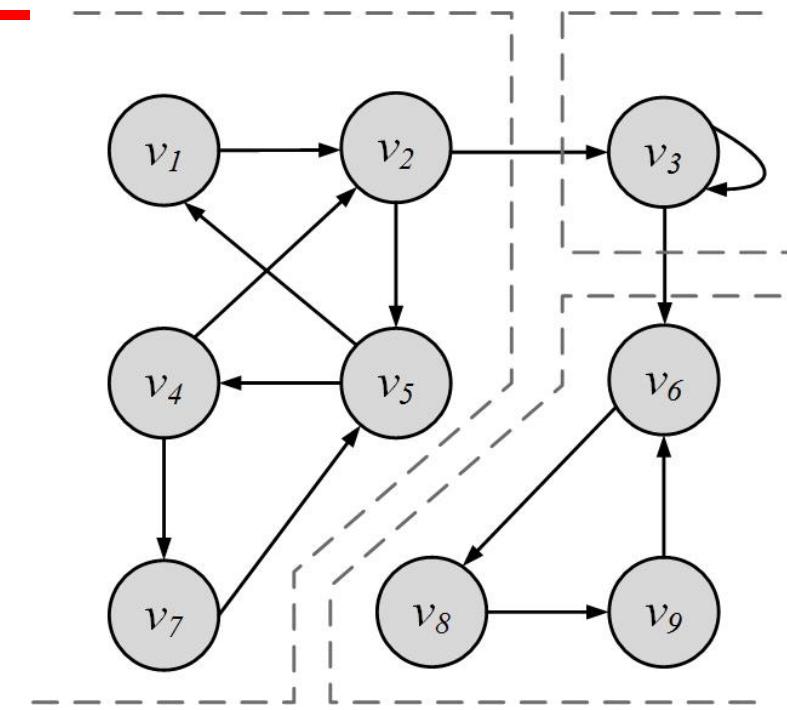
(d) Weakly connected

- 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** 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.

---

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

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

- How big is the diameter of the web?

# Adjacency Matrix and Connectivity

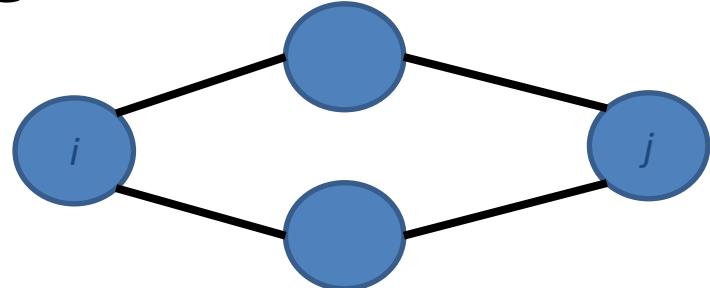


- Consider the following adjacency matrix

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

- Number of Common neighbors between node  $i$  and node  $j$

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

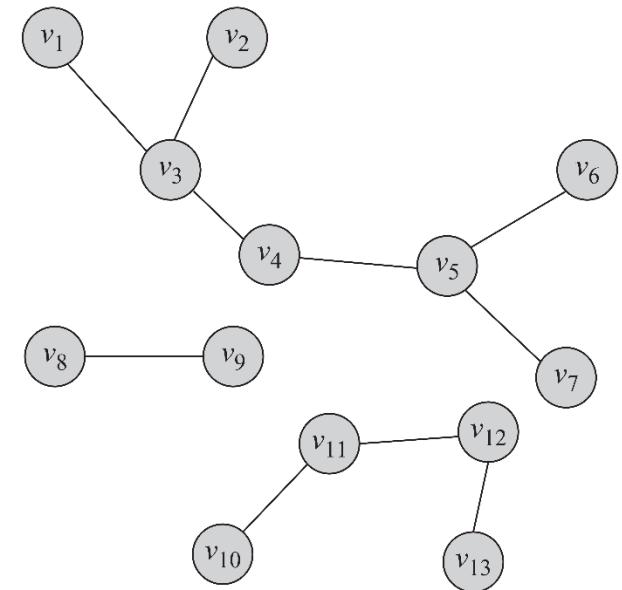


- That's element of  $[ij]$  of matrix  $A \times A^T = A^2$
- Common neighbors are paths of length 2
- Similarly, what is  $A^3$ ?



# Special Graphs

- **Trees** are special cases of undirected graphs
- A tree is a graph structure that has no cycle in it
- In a tree, there is exactly one path between any pair of nodes
- In a tree:  $|V| = |E| + 1$
- A set of disconnected trees is called a **forest**



A forest containing 3 trees



# Graph Algorithms



# Graph/Network Traversal Algorithms

- 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)**

- Depth-First Search (DFS) starts from a node  $v_i$ , selects one of its neighbors  $v_j$  from  $N(v_i)$  and performs Depth-First Search on  $v_j$  before visiting other neighbors in  $N(v_i)$
- The algorithm can be used both for trees and graphs
  - The algorithm can be implemented using a stack structure

---

## Algorithm 2.2 Depth-First Search (DFS)

---

**Require:** Initial node  $v$ , graph/tree  $G(V, E)$ , stack  $S$

```
1: return An ordering on how nodes in  $G$  are visited
2: Push  $v$  into  $S$ ;
3:  $visitOrder = 0$ ;
4: while  $S$  not empty do
5:    $node = \text{pop from } S$ ;
6:   if  $node$  not visited then
7:      $visitOrder = visitOrder + 1$ ;
8:     Mark  $node$  as visited with order  $visitOrder$ ; //or print  $node$ 
9:     Push all neighbors/children of  $node$  into  $S$ ;
10:    end if
11: end while
12: Return all nodes with their visit order.
```

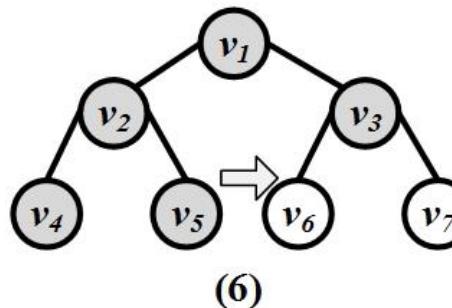
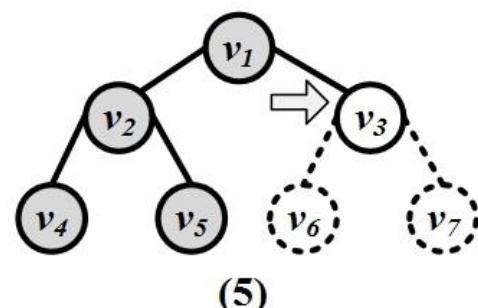
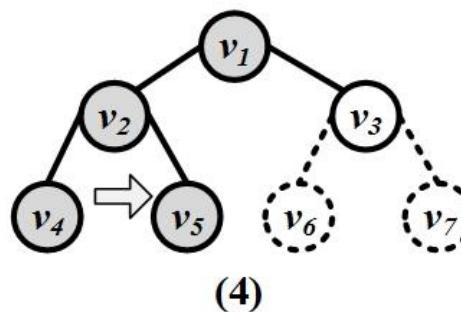
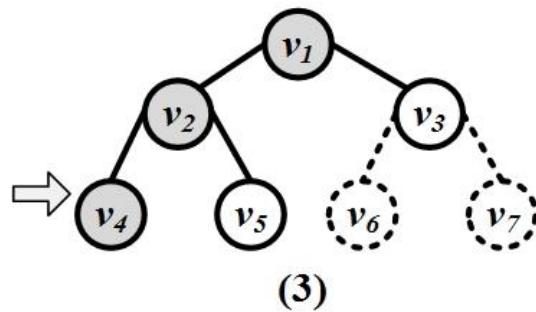
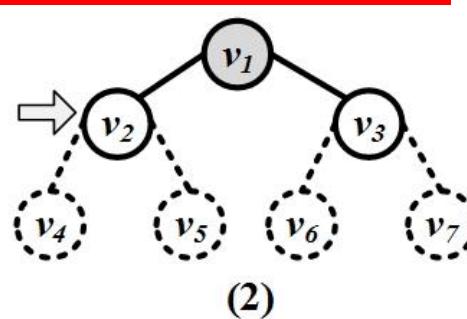
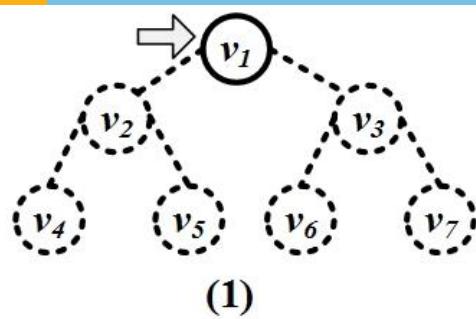
---

# Depth-First Search (DFS): An Example

innovate

achieve

lead



- BFS starts from a node and visits all its immediate neighbors first, and then moves to the second level by traversing their neighbors.
- The algorithm can be used both for trees and graphs
  - The algorithm can be implemented using a queue structure

---

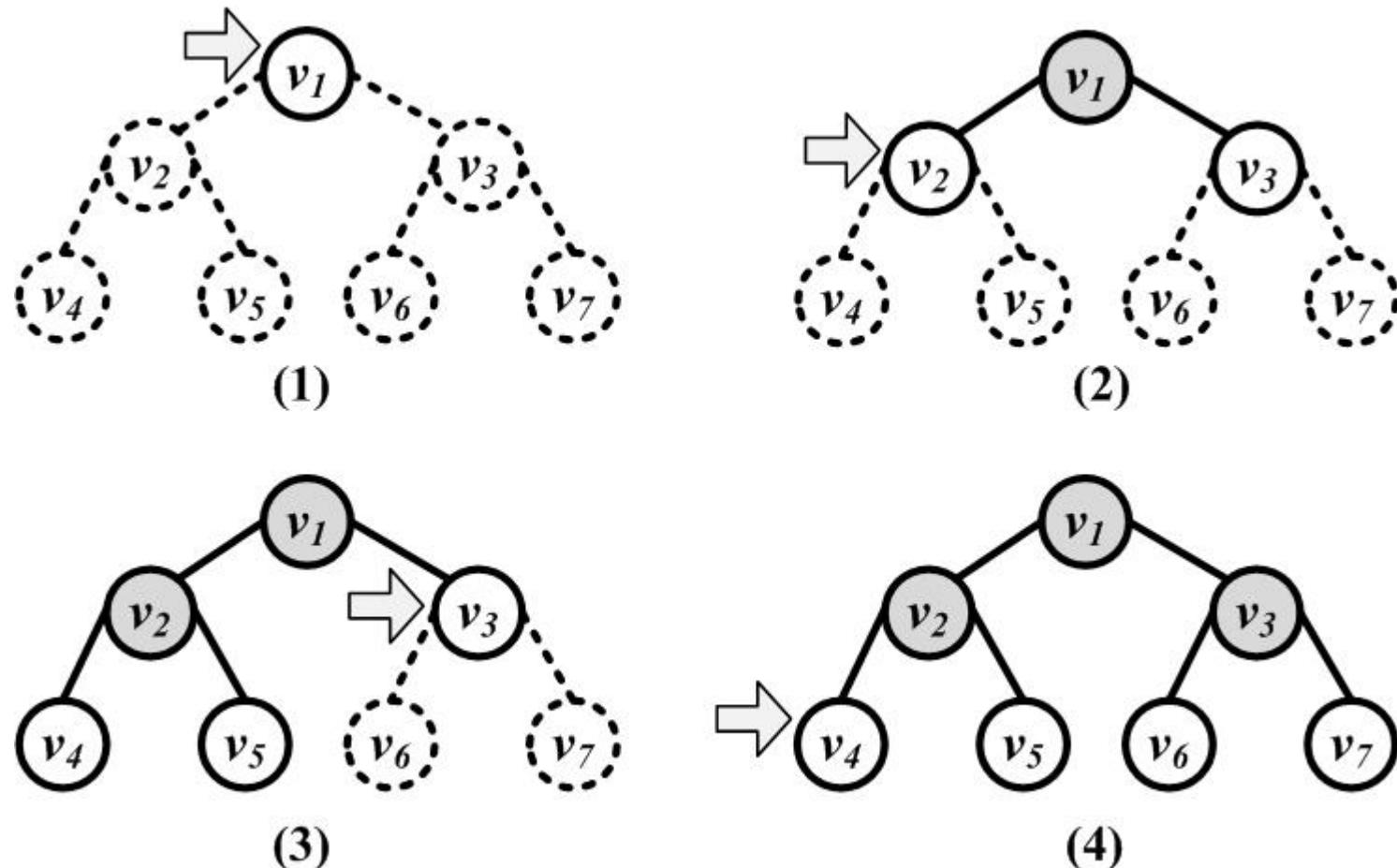
## Algorithm 2.3 Breadth-First Search (BFS)

---

**Require:** Initial node  $v$ , graph/tree  $G(V, E)$ , queue  $Q$

- 1: **return** An ordering on how nodes are visited
  - 2: Enqueue  $v$  into queue  $Q$ ;
  - 3:  $visitOrder = 0$ ;
  - 4: **while**  $Q$  not empty **do**
  - 5:    $node = \text{dequeue from } Q$ ;
  - 6:   **if**  $node$  not visited **then**
  - 7:      $visitOrder = visitOrder + 1$ ;
  - 8:     Mark  $node$  as visited with order  $visitOrder$ ; //or print  $node$
  - 9:     Enqueue all neighbors/children of  $node$  into  $Q$ ;
  - 10:   **end if**
  - 11: **end while**
-

# Breadth-First Search (BFS)





# Finding Shortest Paths

When a graph is connected, there is a chance that multiple paths exist between any pair of nodes

- In many scenarios, we want the shortest path between two nodes in a graph
  - How fast can I disseminate information on social media?

## Dijkstra's Algorithm

- Designed for weighted graphs with non-negative edges
- It finds shortest paths that start from a provided node  $s$  to all other nodes
- It finds both shortest paths and their respective lengths

# Dijkstra's Algorithm: Finding the shortest path

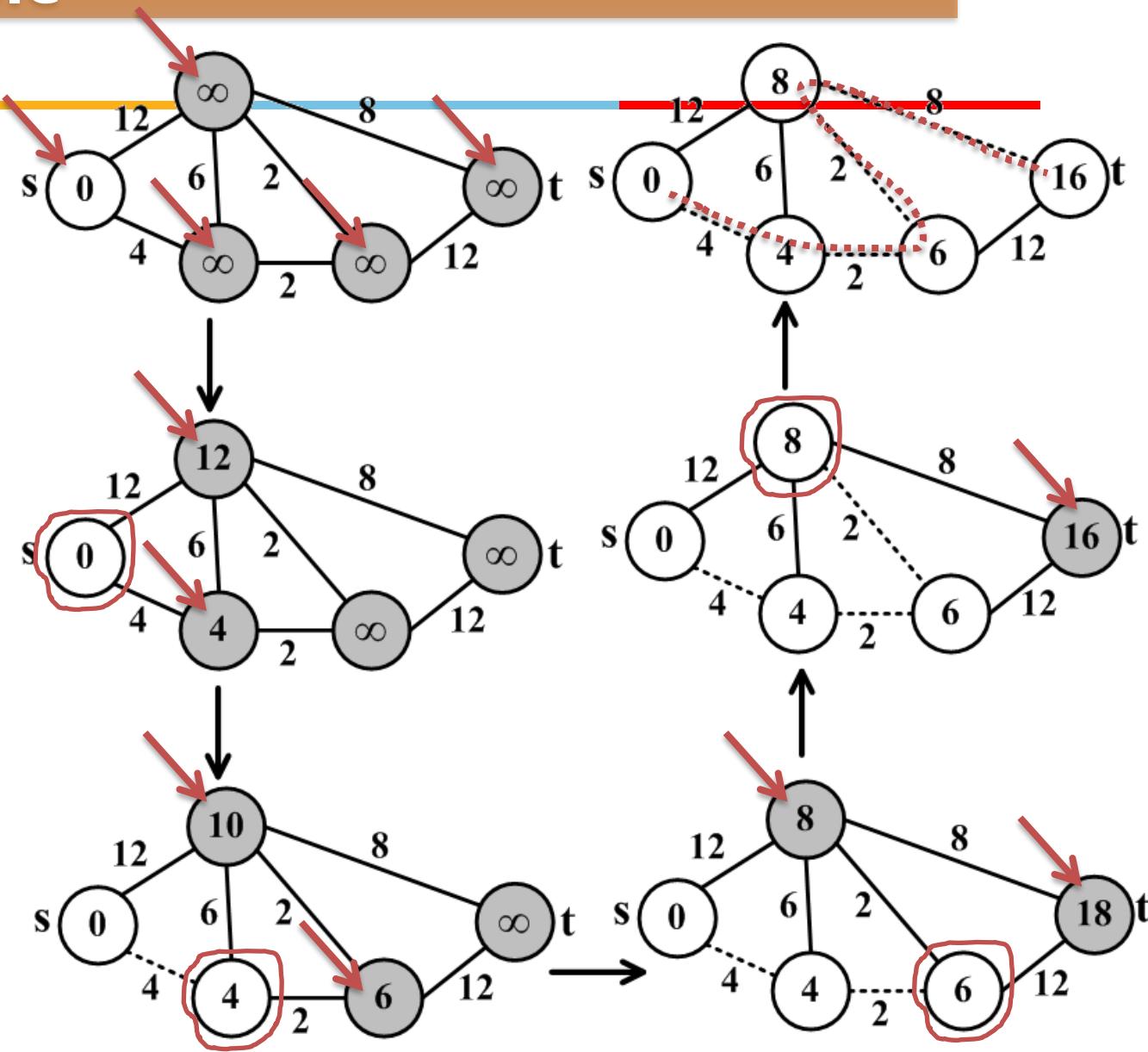


1. Initiation:
  - Assign zero to the source node and infinity to all other nodes
  - Mark all nodes as **unvisited**
  - Set the source node as **current**
2. For the **current** node, consider all of its **unvisited** neighbors and calculate their *tentative* distances
  - If **tentative distance** is smaller than neighbor's distance, then Neighbor's distance = **tentative distance**
3. After considering all of the neighbors of the **current** node, mark the current node as **visited** and remove it from the **unvisited** set
4. If the destination node has been marked **visited** or if the smallest tentative distance among the nodes in the **unvisited** set is infinity, then stop
5. Set the unvisited node marked with the smallest tentative distance as the next "**current** node" and go to step 2

Tentative distance =  
current distance +  
edge weight

A visited node will  
never be checked  
again and its  
distance recorded  
now is final and  
minimal

# Dijkstra's Algorithm: Execution Example



- Dijkstra's algorithm is source-dependent
  - Finds the shortest paths between the source node and all other nodes.
- To generate all-pair shortest paths,
  - We can run Dijkstra's algorithm  $n$  times, or
  - Use other algorithms such as Floyd-Warshall algorithm.
- If we want to compute the shortest path from source  $v$  to destination  $d$ ,
  - we can stop the algorithm once the shortest path to the destination node has been determined



# Finding Minimum 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.

# Prim's Algorithm: Finding Minimum Spanning Tree



Finds MST in a weighted graph

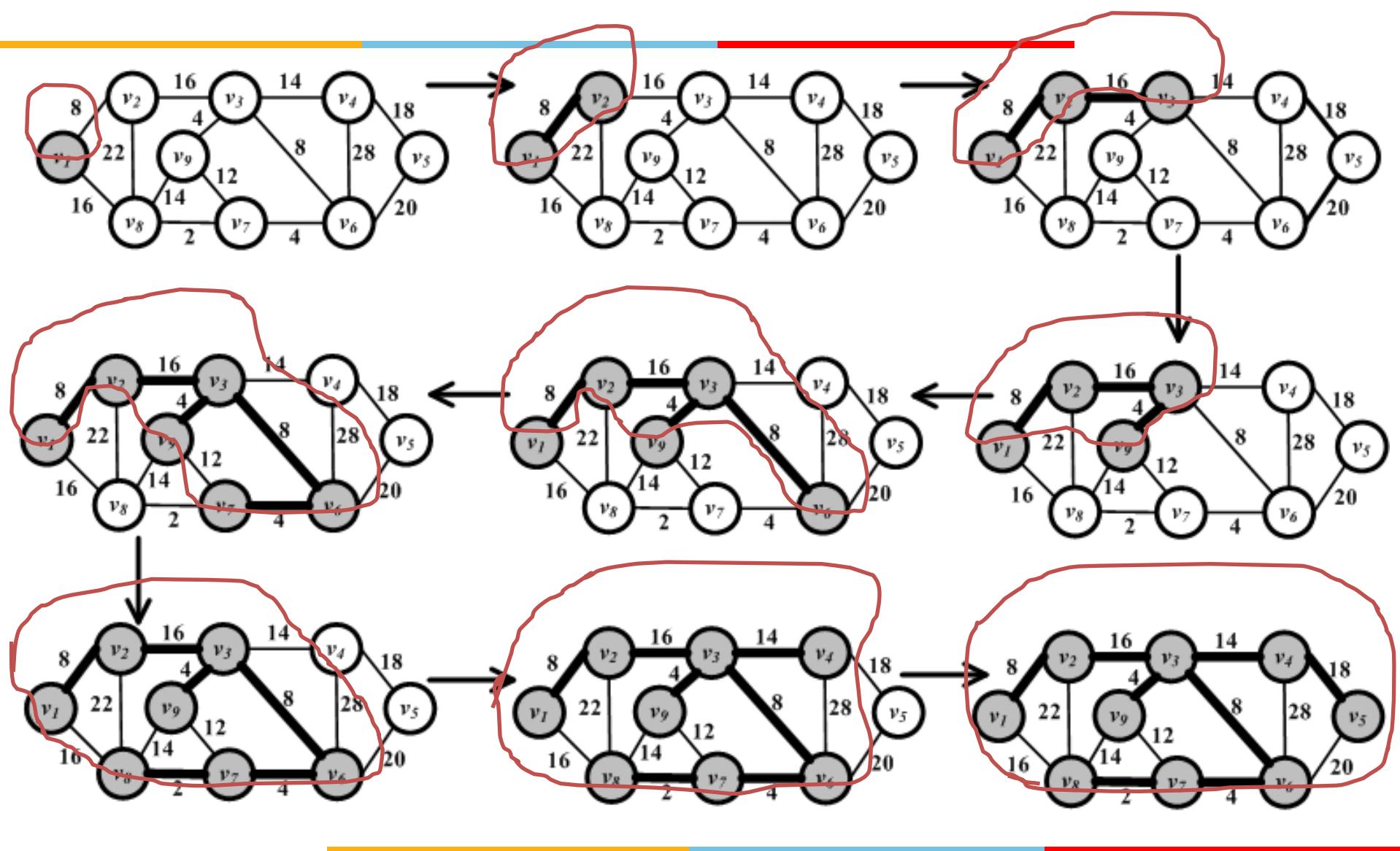
1. Selecting a random node and add it to the MST
2. Grows the spanning tree by selecting edges which have one endpoint in the existing spanning tree and one endpoint among the nodes that are not selected yet. Among the possible edges, the one with the minimum weight is added to the set (along with its end-point).
3. This process is iterated until the graph is fully spanned

# Prim's Algorithm Execution Example

innovate

achieve

lead

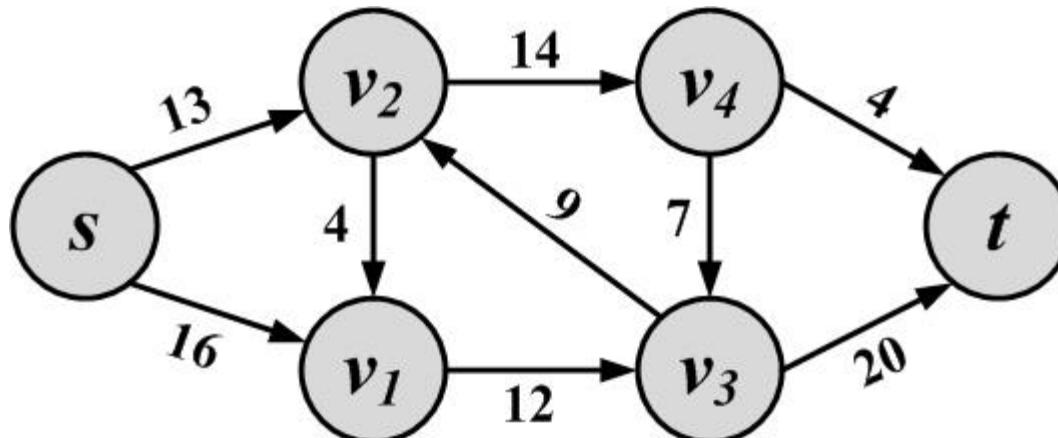




# Network Flow

- 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 Flow network  $G(V,E,C)$  is a directed weighted graph, where we have the following:
  - $\forall (u,v) \in E, c(u,v) \geq 0$  defines the edge capacity.
  - When  $(u,v) \in E, (v,u) \notin E$  (opposite flow is impossible)
  - $s$  defines the source node and  $t$  defines the sink node.  
An infinite supply of flow is connected to the source.

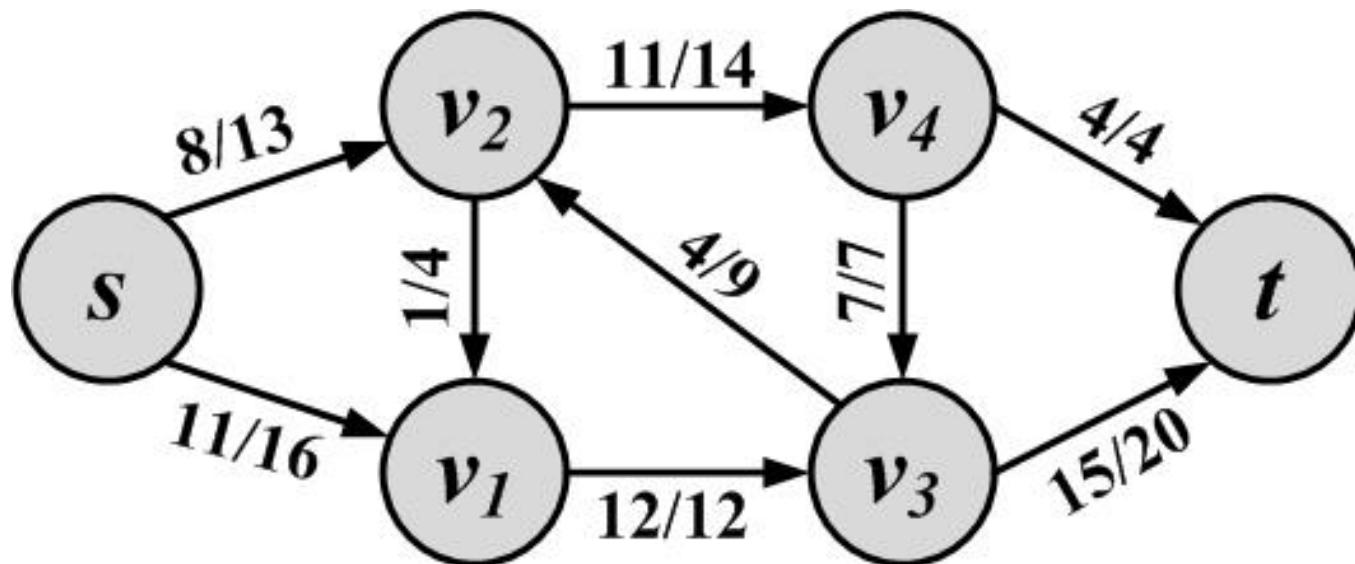


- Given edges with certain capacities, we can fill these edges with the flow up to their capacities (*capacity constraint*)
- The flow that enters any node other than source  $s$  and sink  $t$  is equal to the flow that exits it so that no flow is lost (*flow conservation constraint*)
- $\forall (u, v) \in E, f(u, v) \geq 0$  defines the flow passing through the edge.
- $\forall (u, v) \in E, 0 \leq f(u, v) \leq c(u, v)$  (**capacity constraint**)
- $\forall v \in V - \{s, t\}, \sum_{k:(k,v) \in E} f(k, v) = \sum_{l:(v,l) \in E} f(v, l)$   
**(flow conservation constraint)**

# A Sample Flow Network



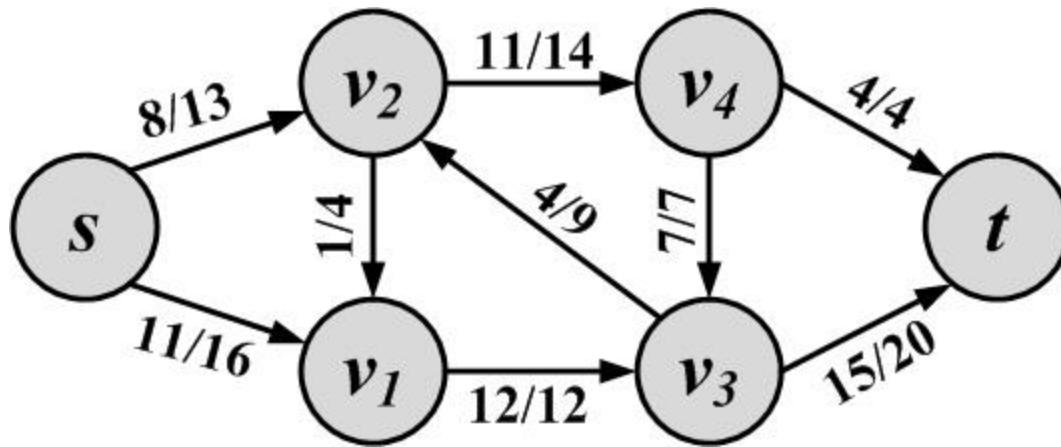
- Commonly, to visualize an edge with capacity  $c$  and flow  $f$ , we use the notation  $f/c$ .



- The flow quantity (or value of the flow) in any network is the amount of
  - Outgoing flow from the source minus the incoming flow to the source.
  - Alternatively, one can compute this value by subtracting the outgoing flow from the sink from its incoming value

$$\text{flow} = \sum_v f(s, v) - \sum_v f(v, s) = \sum_v f(v, t) - \sum_v f(t, v)$$

# What is the flow value?



- **19**
  - **11+8** from **s**, or
  - **4+15** to **t**

- Find a path from source to sink such that there is unused capacity for all edges in the path.
- Use that capacity (the minimum capacity unused among all edges on the path) to increase the flow.
- Iterate until no other path is available.

- Given a flow network  $G(V, E, C)$ , we define another network  $G(V, E_R, C_R)$
- This network defines how much capacity remains in the original network.
- The residual network has an edge between nodes  $u$  and  $v$  if and only if either  $(u, v)$  or  $(v, u)$  exists in the original graph.
  - If one of these two exists in the original network, we would have **two** edges in the residual network: one from  $(u, v)$  and one from  $(v, u)$ .

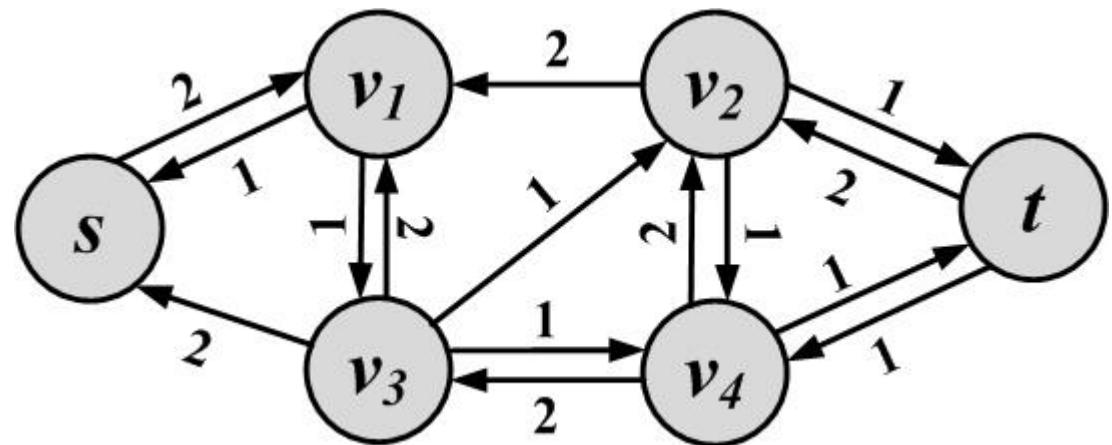
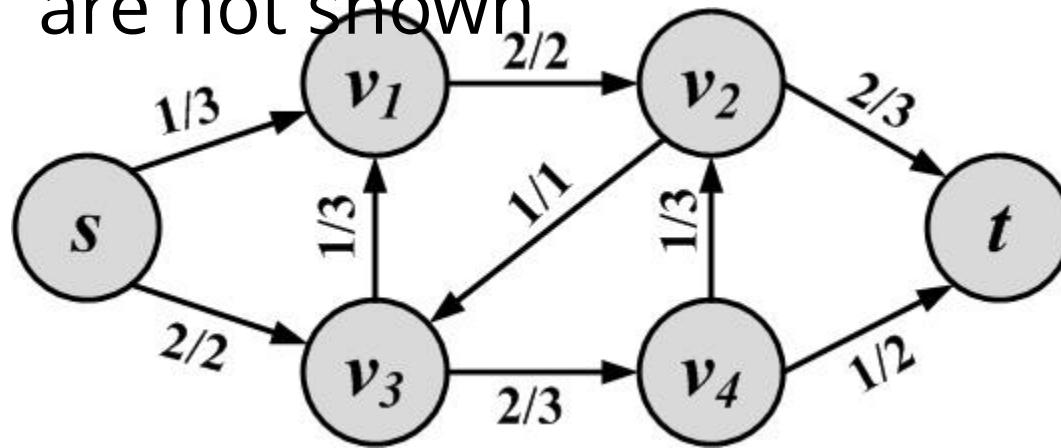
- When there is no flow going through an edge in the original network, a flow of as much as the capacity of the edge remains in the residual.
- In the residual network, one has the ability to send flow in the opposite direction to cancel some amount of flow in the original network.

$$c_R(u, v) = \begin{cases} c(u, v) - f(u, v) & \text{if } (u, v) \in E \\ f(v, u) & \text{if } (u, v) \notin E \end{cases}$$

# Residual Network (Example)



- Edges that have zero capacity in the residual are not shown



1. In the residual graph, when edges are in the same direction as the original graph,
    - Their capacity shows how much **more** flow can be pushed along that edge in the **original** graph.
  2. When edges are in the opposite direction,
    - their capacities show how much flow can be **pushed back** on the **original graph edge**.
- By finding a flow in the residual, we can **augment** the flow in the original graph.

- Any simple path from  $s$  to  $t$  in the residual graph is an *augmenting path*.
  - All capacities in the residual are positive,
    - These paths can augment flows in the original, thus increasing the flow.
  - The amount of flow that can be pushed along this path is equal to the **minimum capacity** along the path
    - The edge with the minimum capacity limits the amount of flow being pushed
    - We call the edge the ***Weak link***

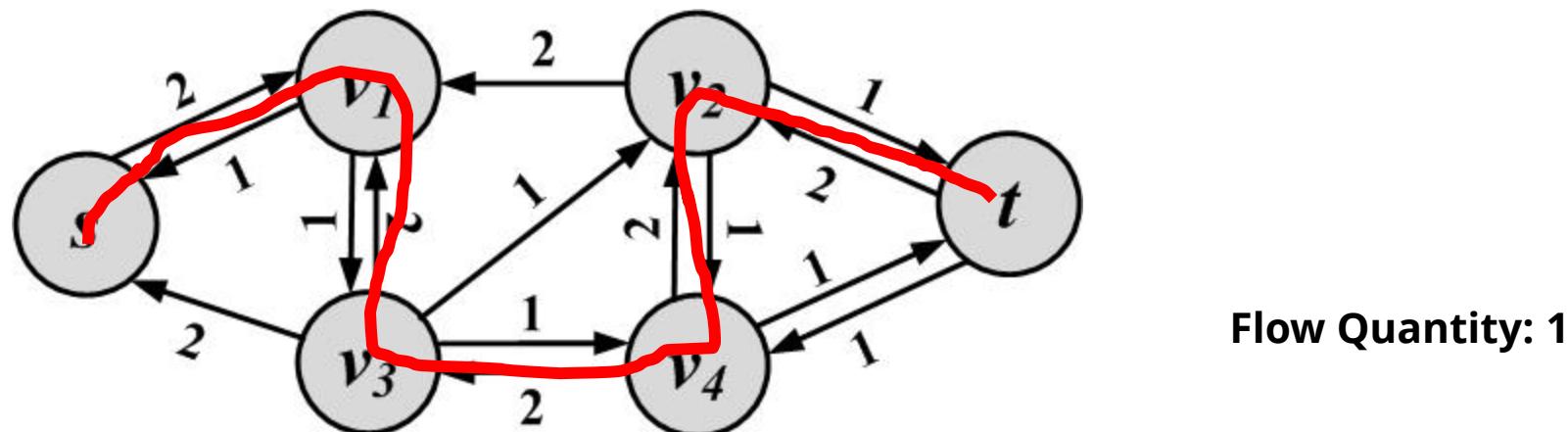
# How do we augment?

innovate

achieve

lead

- Given flow  $f(u, v)$  in the original graph and flow  $f_R(u, v)$  and  $f_R(v, u)$  in the residual graph, we can augment the flow as follows:  
$$f_{\text{augmented}}(u, v) = f(u, v) + f_R(u, v) - f_R(v, u)$$

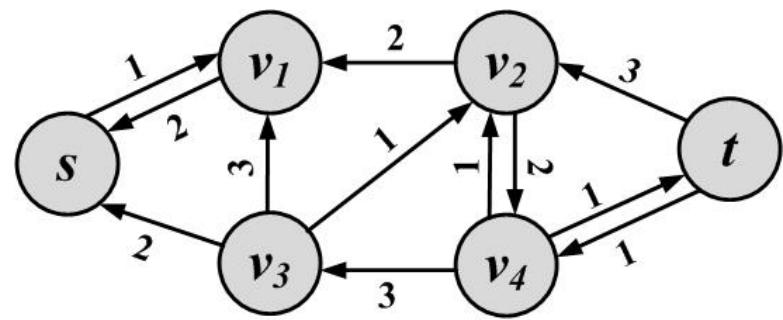
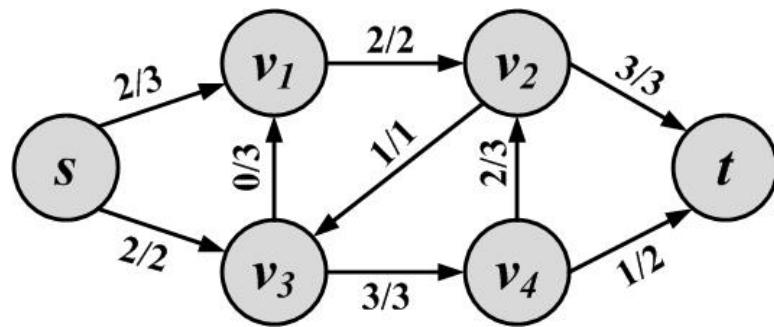
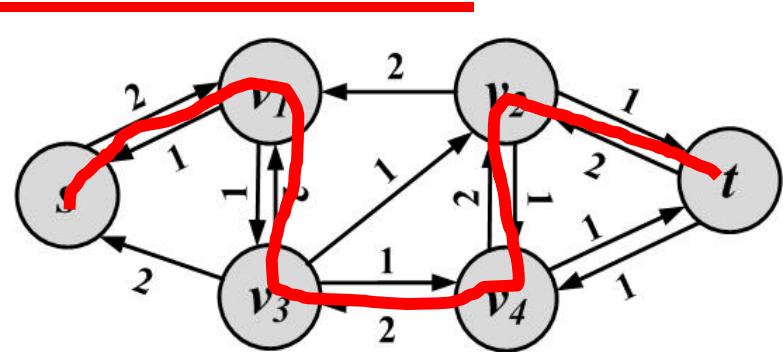
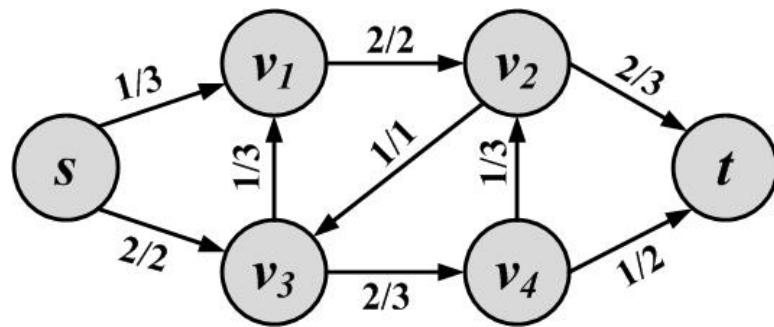


# Augmenting

innovate

achieve

lead



---

## Algorithm 2.6 Ford-Fulkerson Algorithm

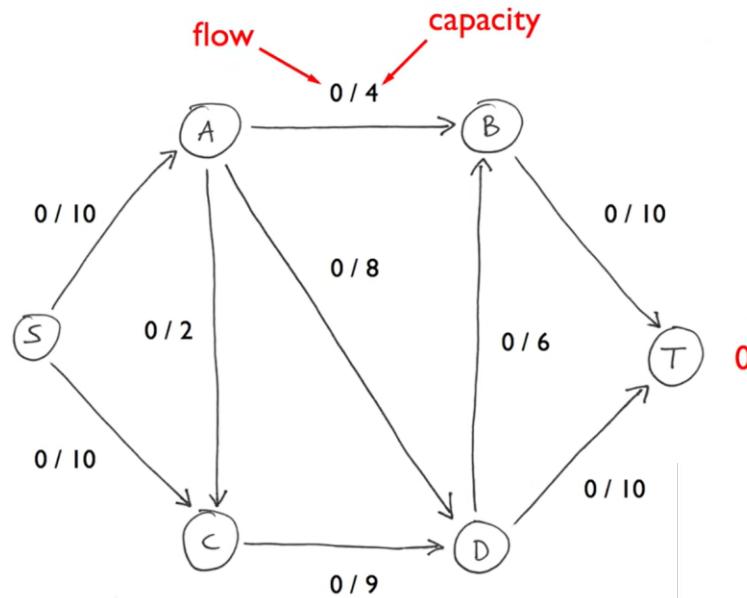
---

**Require:** Connected weighted graph  $G(V, E, W)$ , Source  $s$ , Sink  $t$

- 1: **return** A Maximum flow graph
  - 2:  $\forall(u, v) \in E, f(u, v) = 0$
  - 3: **while** there exists an augmenting path  $p$  in the residual graph  $G_R$  **do**
  - 4:   Augment flows by  $p$
  - 5: **end while**
  - 6: Return flow value and flow graph;
-

# Ford-Fulkerson Algorithm:

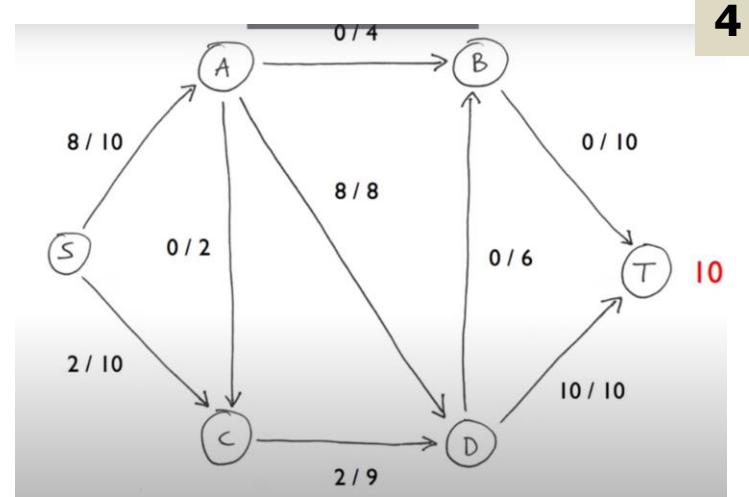
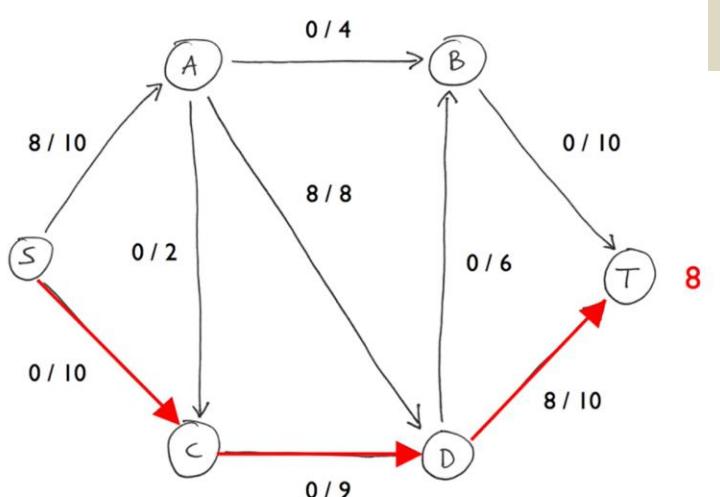
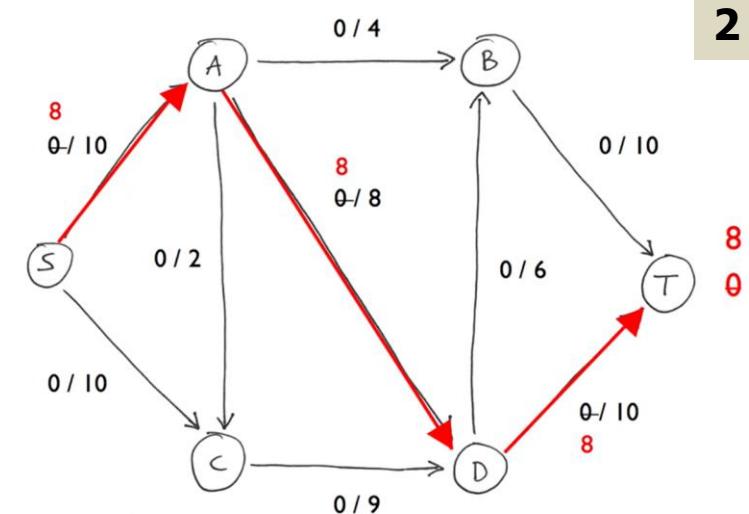
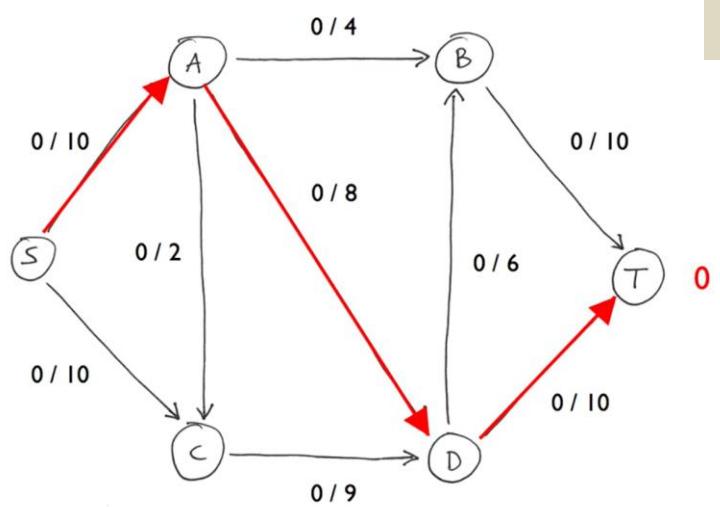
## Example: 1/3



1. find an augmenting path
2. compute the bottleneck capacity
3. augment each edge and the total flow

# Ford-Fulkerson Algorithm:

## Example: 2/3



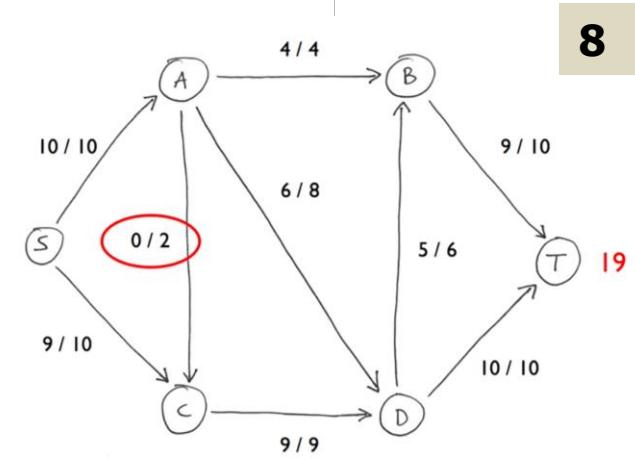
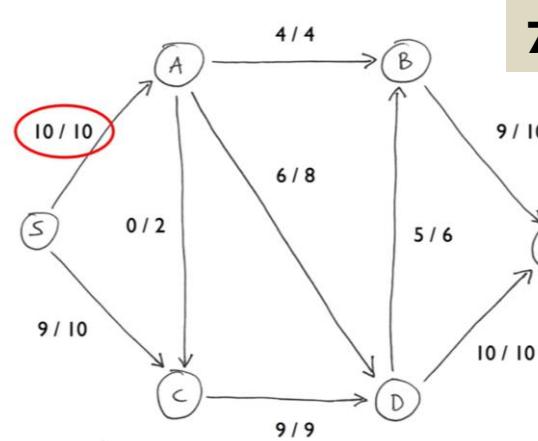
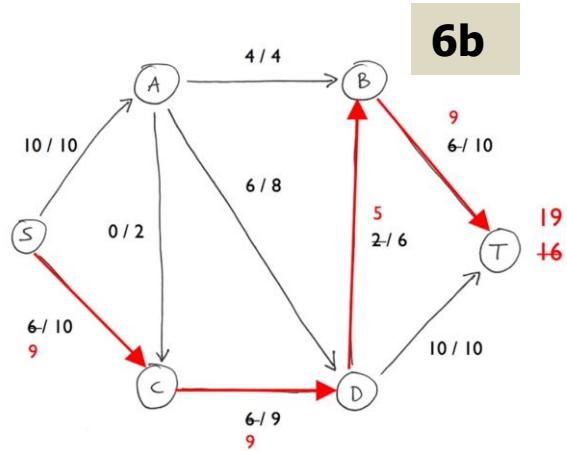
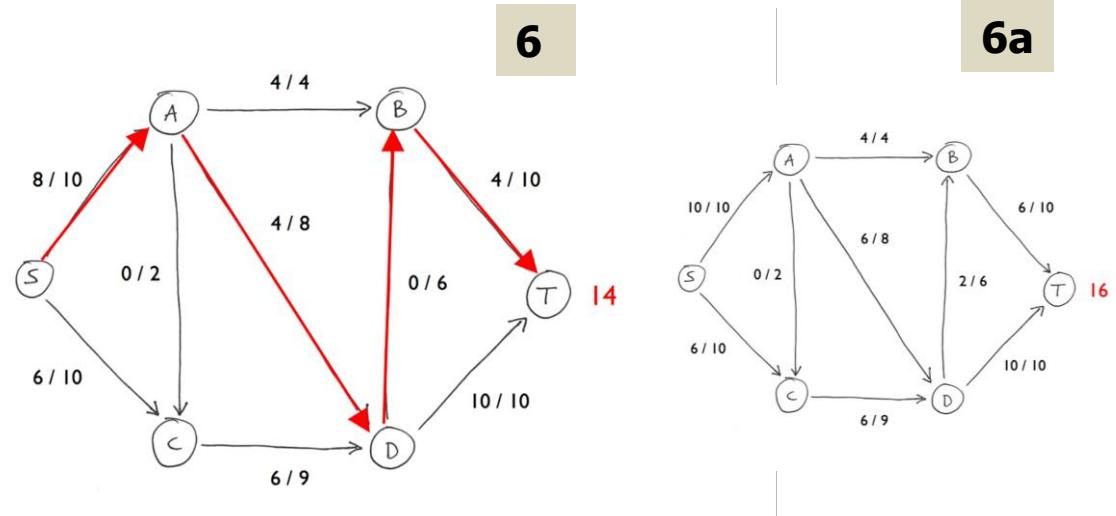
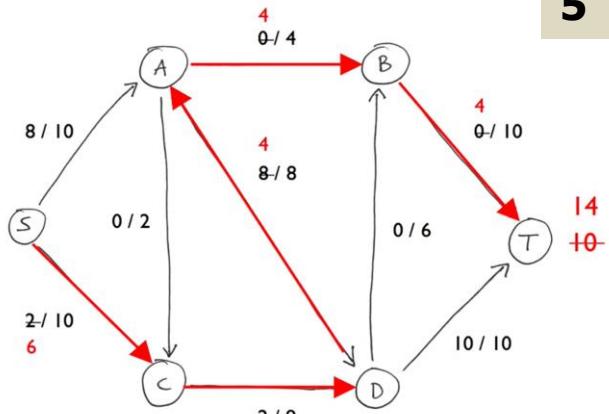
# Ford-Fulkerson Algorithm:

## Example: 3/3

innovate

achieve

lead

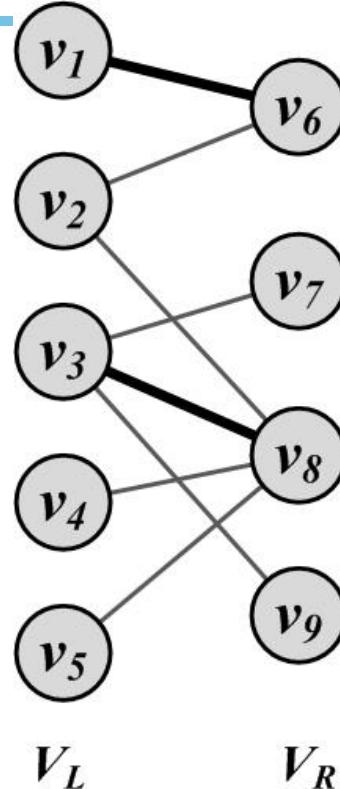




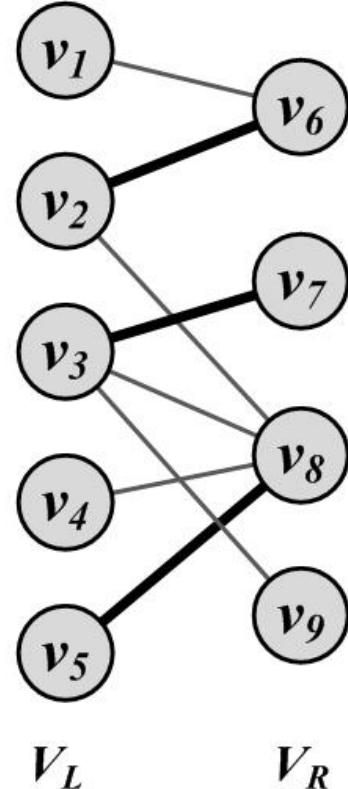
# Maximum Bipartite Matching

# Example

- Given  $n$  products and  $m$  users
  - Some users are only interested in certain products
  - We have only one copy of each product.
  - Can be represented as a bipartite graph
  - Find the maximum number of products that can be bought by users
    - No two edges selected share a node



Matching

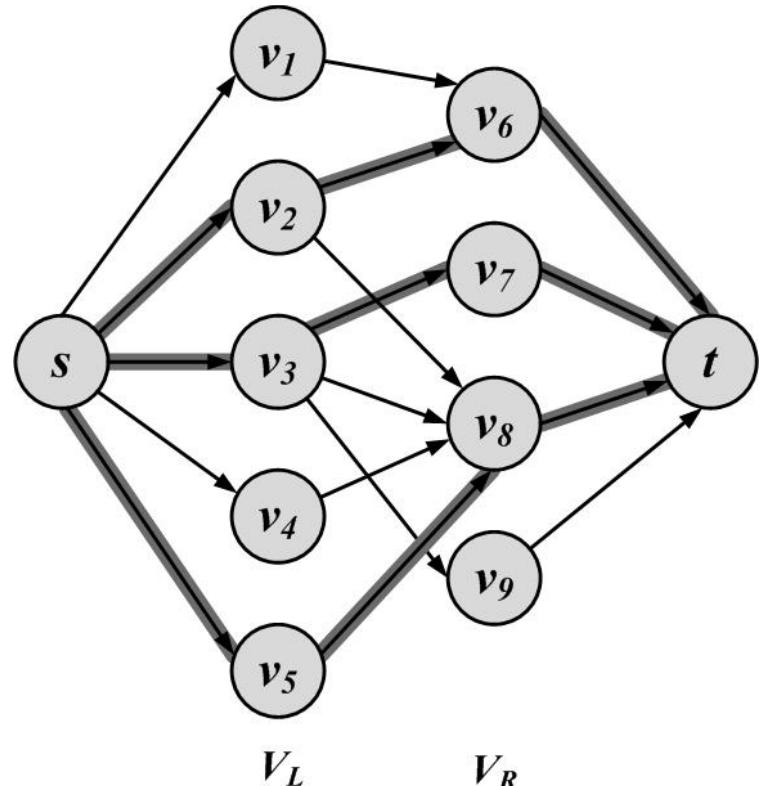


Maximum Matching

# Matching Solved with Max-Flow



- Create a flow graph  $G(V', E', C)$  from our bipartite graph  $G(V, E)$ 
  1. Set  $V' = V \cup \{s\} \cup \{t\}$
  2. Connect all nodes in  $V_L$  to  $s$  and all nodes in  $V_R$  to  $t$
  3. Set  $c(u, v) = 1$ , for all edges in  $E'$

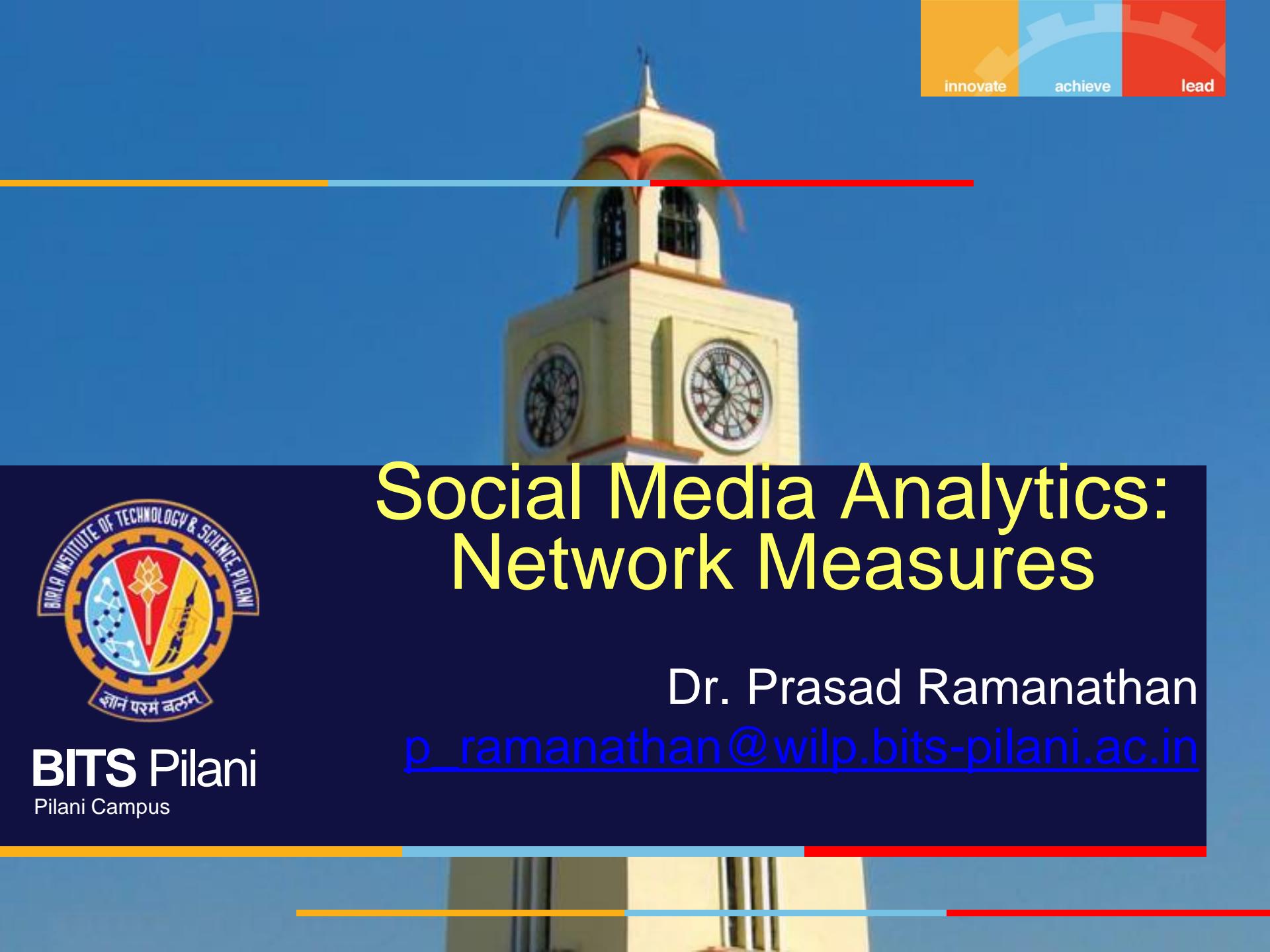




**BITS** Pilani  
Pilani Campus

# Questions?





# Social Media Analytics: Network Measures



**BITS** Pilani  
Pilani Campus

Dr. Prasad Ramanathan

[p\\_ramanathan@wilp.bits-pilani.ac.in](mailto:p_ramanathan@wilp.bits-pilani.ac.in)

# Acknowledgment

Course material from the following source is gratefully acknowledged:

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/>**

**99**



## Barack Obama

ADD +

This account is run by #Obama2012 campaign staff. Tweets from the President are signed -bo.

Influences 2M others



1.7M tweets • 1.7M shares • see more

Influential about 20 topics

- Government
- Politics
- Media

1.7M tweets • 1.7M shares • see all

**It is difficult  
to measure  
influence!**

**92**



## Justin Bieber

ADD +

invite you  
#BELIEVE is on IT!  
MUCH LOVE FOR  
and I will always be

**KLOUT**

*the Standard for Influence*

Influences 10M others



1.7M tweets • 1.7M shares • see more

## Klout Summary for Warren Buffett

Score Analysis



### Warren Buffett

Investor, Philanthropist  
*Omaha, Nebraska*

**36**  
klout score

# 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.



# Centrality

**Centrality defines how important a node is within a network**



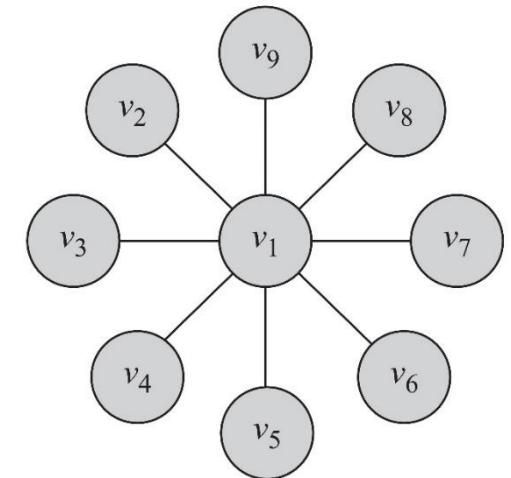
**Centrality in terms of those  
who you are connected to**

**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  $d_j = 1, j \neq 1$



# Degree Centrality in Directed Graphs



In directed graphs, we can either use the in-degree, the out-degree, or the combination as the degree centrality value:

In practice, mostly in-degree is used.

$$C_d(v_i) = d_i^{\text{in}} \quad (\textit{prestige})$$

$$C_d(v_i) = d_i^{\text{out}} \quad (\textit{gregariousness})$$

$$C_d(v_i) = d_i^{\text{in}} + d_i^{\text{out}}$$

$d_i^{\text{out}}$  is the number of outgoing links for node  $v_i$

# Normalized Degree Centrality



Normalized by the maximum  
possible degree

$$C_d^{\text{norm}}(v_i) = \frac{d_i}{n-1}$$

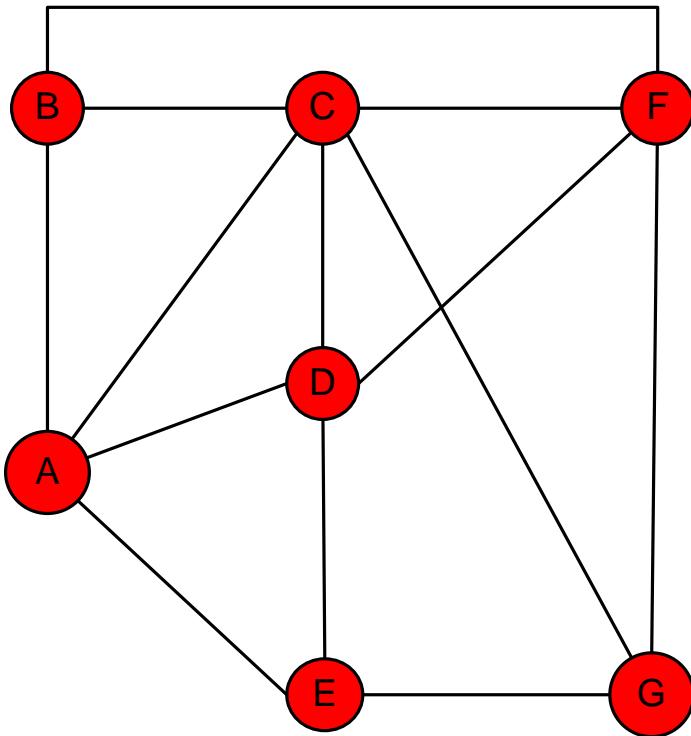
Normalized by the maximum  
degree

$$C_d^{\text{max}}(v_i) = \frac{d_i}{\max_j d_j}$$

Normalized by the degree sum

$$C_d^{\text{sum}}(v_i) = \frac{d_i}{\sum_j d_j} = \frac{d_i}{2|E|} = \frac{d_i}{2m}$$

# Degree Centrality (undirected Graph) Example



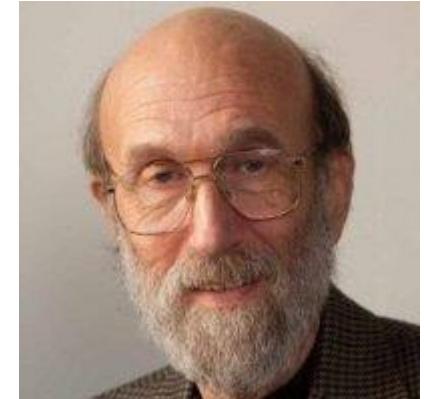
Node	Degree	Centrality	Rank
A	4	2/3	2
B	3	1/2	5
C	5	5/6	1
D	4	2/3	2
E	3	1/2	5
F	4	2/3	2
G	3	1/2	5

Normalized by the maximum possible degree

$$C_d^{\text{norm}}(v_i) = \frac{d_i}{n-1}$$

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

Let's assume the eigenvector centrality of a node is  $c_e(v_i)$  (**unknown**)

We would like  $c_e(v_i)$  to be higher when **important** neighbors (**node  $v_j$  with higher  $c_e(v_j)$** ) point to us

Incoming or outgoing neighbors?

For incoming neighbors  $A_{j,i} = 1$

We can assume that  $v_i$ 's centrality is the summation of its neighbors' centralities

$$c_e(v_i) = \sum_{j=1}^n A_{j,i} c_e(v_j)$$

Is this summation bounded?

We have to normalize!

$\lambda$ : **some fixed constant**

$$c_e(v_i) = \frac{1}{\lambda} \sum_{j=1}^n A_{j,i} c_e(v_j)$$

# Eigenvector Centrality (Matrix Formulation)



Let  $\mathbf{C}_e = (C_e(v_1), C_e(v_2), \dots, C_e(v_n))^T$

→  $\lambda \mathbf{C}_e = A^T \mathbf{C}_e$

This means that  $\mathbf{C}_e$  is an eigenvector of adjacency matrix  $A^T$  (or  $A$  when undirected) and  $\lambda$  is the corresponding eigenvalue

Which eigenvalue-eigenvector pair should we choose?

# Finding the eigenvalue by finding a fixed point...



Start from an initial guess  $C_e(0)$  (e.g., all centralities are 1) and iterate  $t$  times

$$C_e(t) = (A^T)^t C_e(0)$$

We can write  $C_e(0)$  as a linear combination of eigenvectors  $v_i$ 's of the  $A^T$

$$C_e(0) = \sum_i \alpha_i v_i$$

Substituting this, we get

$$C_e(t) = (A^T)^t \sum_i \alpha_i v_i = \sum_i \alpha_i \lambda_i^t v_i = \lambda_1^t \sum_i \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^t v_i$$

$\lambda_1$  is the largest eigenvalue

# Finding the eigenvalue by finding a fixed point...



As  $t$  grows, we will have in the limit

$$C_e(t) \rightarrow \alpha_1 \lambda_1^t v_1$$

Or equivalently

$$A^T C_e(t) = A^T C_e = \lambda_1 C_e$$

If we start with an all positive  $C_e(0)$  all  $C_e(t)$ 's will be positive (why?)

All the centrality values would be positive

We need an eigenvalue-eigenvector pair that guarantees all centralities have the same sign

E.g., for comparison purposes

# Eigenvector Centrality, cont.

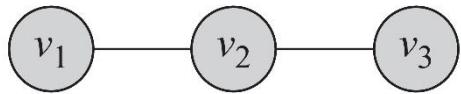


**Theorem 1** (Perron-Frobenius Theorem). *Let  $A \in \mathbb{R}^{n \times n}$  represent the adjacency matrix for a [strongly] connected graph or  $A : A_{i,j} > 0$  (i.e. a positive  $n$  by  $n$  matrix). There exists a positive real number (Perron-Frobenius eigenvalue)  $\lambda_{\max}$ , such that  $\lambda_{\max}$  is an eigenvalue of  $A$  and any other eigenvalue of  $A$  is strictly smaller than  $\lambda_{\max}$ . Furthermore, there exists a corresponding eigenvector  $\mathbf{v} = (v_1, v_2, \dots, v_n)$  of  $A$  with eigenvalue  $\lambda_{\max}$  such that  $\forall v_i > 0$ .*

So, to compute eigenvector centrality of  $A$ ,

1. We compute the eigenvalues of  $A$
2. Select the largest eigenvalue  $\lambda$
3. The corresponding eigenvector of  $\lambda$  is  $\mathbf{C}_e$ .
4. Based on the Perron-Frobenius theorem, all the components of  $\mathbf{C}_e$  will be positive
5. The components of  $\mathbf{C}_e$  are the eigenvector centralities for the graph.

# Eigenvector Centrality: Example 1



$$\lambda \mathbf{C}_e = A \mathbf{C}_e \quad (A - \lambda I) \mathbf{C}_e = 0 \quad \mathbf{C}_e = [u_1 \ u_2 \ u_3]^T$$

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

$$\begin{bmatrix} 0 - \lambda & 1 & 0 \\ 1 & 0 - \lambda & 1 \\ 0 & 1 & 0 - \lambda \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\det(A - \lambda I) = \begin{vmatrix} 0 - \lambda & 1 & 0 \\ 1 & 0 - \lambda & 1 \\ 0 & 1 & 0 - \lambda \end{vmatrix} = 0$$

$$(-\lambda)(\lambda^2 - 1) - 1(-\lambda) = 2\lambda - \lambda^3 = \lambda(2 - \lambda^2) = 0$$

Eigenvalues are

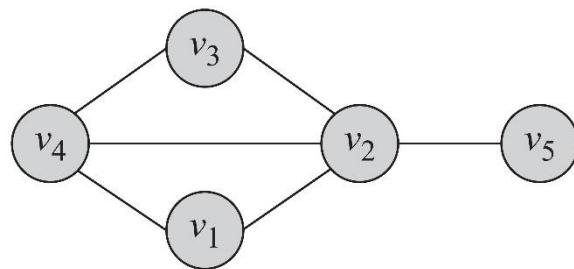
$$(-\sqrt{2}, 0, +\sqrt{2})$$

Largest Eigenvalue

Corresponding eigenvector (assuming  $\mathbf{C}_e$  has norm 1)

$$\begin{bmatrix} 0 - \sqrt{2} & 1 & 0 \\ 1 & 0 - \sqrt{2} & 1 \\ 0 & 1 & 0 - \sqrt{2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{C}_e = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1/2 \\ \sqrt{2}/2 \\ 1/2 \end{bmatrix}$$

# Eigenvector Centrality: Example 2



$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \rightarrow \lambda = (2.68, -1.74, -1.27, 0.33, 0.00)$$

↑  
Eigenvalues Vector

$$\lambda_{\max} = 2.68$$



$$C_e = \begin{bmatrix} 0.4119 \\ 0.5825 \\ 0.4119 \\ 0.5237 \\ 0.2169 \end{bmatrix}$$

# 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  $\beta$  to the centrality values for all nodes

Eigenvector Centrality

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

# Katz Centrality, cont.



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

Controlling term                                  Bias term

Rewriting equation in a vector form

$$\mathbf{C}_{\text{Katz}} = \alpha A^T \mathbf{C}_{\text{Katz}} + \beta \mathbf{1}$$

vector of all 1's

Katz centrality:  $\mathbf{C}_{\text{Katz}} = \beta(\mathbf{I} - \alpha A^T)^{-1} \cdot \mathbf{1}$

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

When  $\alpha=0$ , the eigenvector centrality is removed and all nodes get the same centrality value  $\beta$

- As  $\alpha$  gets larger the effect of  $\beta$  is reduced

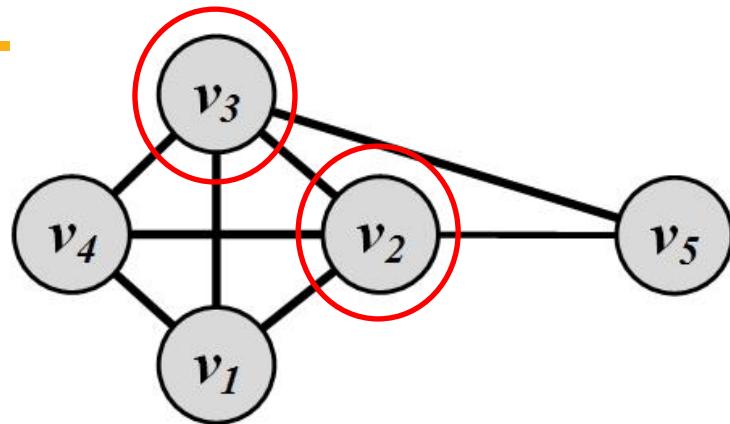
For the matrix  $(I - \alpha A^T)$  to be invertible, we must have

- $\det(I - \alpha A^T) \neq 0$
- By rearranging we get  $\det(A^T - \alpha^{-1} I) = 0$
- This is basically the characteristic equation,
- The characteristic equation **first** becomes zero when the largest eigenvalue equals  $\alpha^{-1}$

The largest eigenvalue is easier to compute (power method)

In practice we select  $\alpha < 1/\lambda$ , where  $\lambda$  is the largest eigenvalue of  $A^T$

# Katz Centrality Example



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

The Eigenvalues are -1.68, -1.0, -1.0, 0.35, 3.32  
We assume  $\alpha=0.25 < \frac{1}{3.32}$  and  $\beta = 0.2$

$$\mathbf{C}_{Katz} = \beta(\mathbf{I} - \alpha A^T)^{-1} \cdot \mathbf{1} = \begin{bmatrix} 1.14 \\ 1.31 \\ 1.31 \\ 1.14 \\ 0.85 \end{bmatrix}$$

**Most important nodes!**

## 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

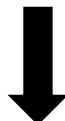
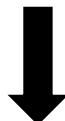
# PageRank, cont.



$$C_p(v_i) = \alpha \sum_{j=1}^n A_{j,i} \frac{C_p(v_j)}{d_j^{\text{out}}} + \beta$$

What if the degree is zero?

$$\begin{cases} d_j^{\text{out}} > 0 \\ D = \text{diag}(d_1^{\text{out}}, d_2^{\text{out}}, \dots, d_n^{\text{out}}) \end{cases} \rightarrow \mathbf{C}_p = \alpha A^T D^{-1} \mathbf{C}_p + \beta \mathbf{1}$$



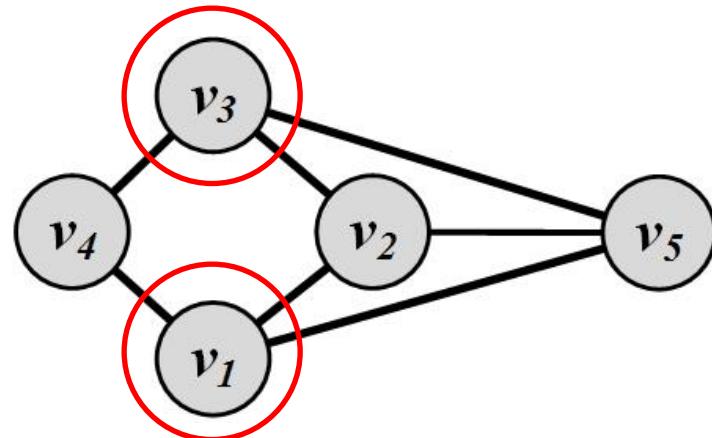
$$\mathbf{C}_p = \beta(\mathbf{I} - \alpha A^T D^{-1})^{-1} \cdot \mathbf{1}$$

Similar to Katz Centrality, in practice,  $\alpha < 1/\lambda$ , where  $\lambda$  is the largest eigenvalue of  $A^T D^{-1}$ . In undirected graphs, the largest eigenvalue of  $A^T D^{-1}$  is  $\lambda = 1$ ; therefore,  $\alpha < 1$ .

# PageRank Example



We assume  $\alpha=0.95 < 1$  and  $\beta = 0.1$



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

$$\mathbf{C}_p = \beta(\mathbf{I} - \alpha A^T D^{-1})^{-1} \cdot \mathbf{1} =$$

$$\begin{bmatrix} 2.14 \\ 2.13 \\ 2.14 \\ 1.45 \\ 2.13 \end{bmatrix}$$



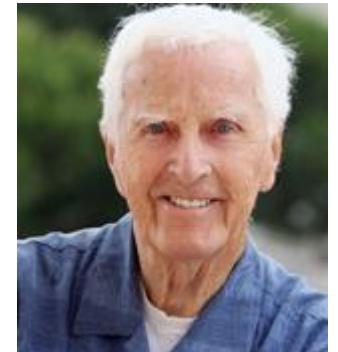
---

**Centrality in terms of how  
you connect others  
(information broker)**

# Betweenness Centrality



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



*Linton Freeman*

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

$\sigma_{st}$

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  $v_i$

# Normalizing Betweenness Centrality



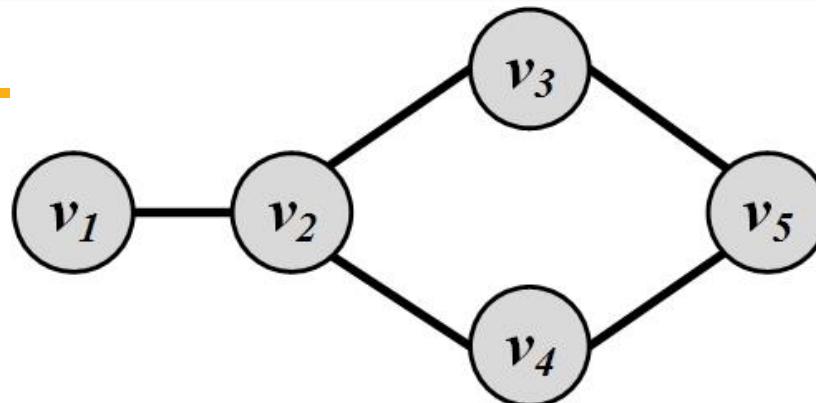
In the best case, node  $v_i$  is on all shortest paths from  $s$  to  $t$ , hence,  $\frac{\sigma_{st}(v_i)}{\sigma_{st}} = 1$

$$\begin{aligned} C_b(v_i) &= \sum_{s \neq t \neq v_i} \frac{\sigma_{st}(v_i)}{\sigma_{st}} \\ &= \sum_{s \neq t \neq v_i} 1 = 2 \binom{n-1}{2} = (n-1)(n-2) \end{aligned}$$

Therefore, the maximum value is  $(n-1)(n-2)$

**Betweenness centrality:**  $C_b^{\text{norm}}(v_i) = \frac{C_b(v_i)}{2 \binom{n-1}{2}}$

# Betweenness Centrality: Example 1



$$C_b(v_2) = 2 \times \left( \underbrace{(1/1)}_{s=v_1, t=v_3} + \underbrace{(1/1)}_{s=v_1, t=v_4} + \underbrace{(2/2)}_{s=v_1, t=v_5} + \underbrace{(1/2)}_{s=v_3, t=v_4} + \underbrace{0}_{s=v_3, t=v_5} + \underbrace{0}_{s=v_4, t=v_5} \right)$$
$$= 2 \times 3.5 = 7,$$

$$C_b(v_3) = 2 \times \left( \underbrace{0}_{s=v_1, t=v_2} + \underbrace{0}_{s=v_1, t=v_4} + \underbrace{(1/2)}_{s=v_1, t=v_5} + \underbrace{0}_{s=v_2, t=v_4} + \underbrace{(1/2)}_{s=v_2, t=v_5} + \underbrace{0}_{s=v_4, t=v_5} \right)$$
$$= 2 \times 1.0 = 2,$$

$$C_b(v_4) = C_b(v_3) = 2 \times 1.0 = 2,$$

$$C_b(v_5) = 2 \times \left( \underbrace{0}_{s=v_1, t=v_2} + \underbrace{0}_{s=v_1, t=v_3} + \underbrace{0}_{s=v_1, t=v_4} + \underbrace{0}_{s=v_2, t=v_3} + \underbrace{0}_{s=v_2, t=v_4} + \underbrace{(1/2)}_{s=v_3, t=v_4} \right)$$
$$= 2 \times 0.5 = 1,$$



---

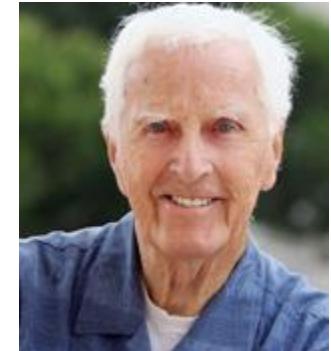
**Centrality in terms of how  
fast you can reach others**

---

# Closeness Centrality



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



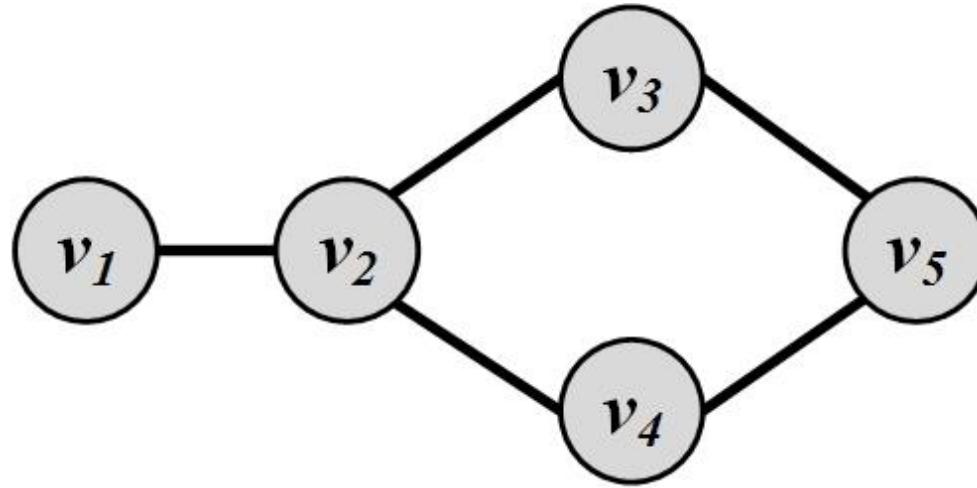
These nodes should have a smaller average shortest path length to others

*Linton Freeman*

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

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

# Closeness Centrality: Example 1



$$C_c(v_1) = 1 / ( (1 + 2 + 2 + 3)/4 ) = 0.5,$$

$$C_c(v_2) = 1 / ( (1 + 1 + 1 + 2)/4 ) = 0.8,$$

$$C_c(v_3) = C_b(v_4) = 1 / ( (1 + 1 + 2 + 2)/4 ) = 0.66,$$

$$C_c(v_5) = 1 / ( (1 + 1 + 2 + 3)/4 ) = 0.57.$$



# Centrality for a group of nodes

All centrality measures defined so far measure centrality for a single node. These measures can be generalized for a group of nodes.

A simple approach is to replace all nodes in a group with a super node  
The group structure is disregarded.

Let  $S$  denote the set of nodes in the group and  $V - S$  the set of outsiders

## I. Group Degree Centrality

$$C_d^{\text{group}}(S) = |\{v_i \in V - S \mid v_i \text{ is connected to } v_j \in S\}|$$

**Normalization:** divide by  $|V - S|$

## II. Group Betweenness Centrality

$$C_b^{\text{group}}(S) = \sum_{s \neq t, s \notin S, t \notin S} \frac{\sigma_{st}(S)}{\sigma_{st}}$$

**Normalization:** divide by  $2 \binom{|V - S|}{2}$

## III. Group Closeness Centrality

$$C_c^{\text{group}}(S) = \frac{1}{\bar{l}_S^{\text{group}}}$$

It is the average distance from non-members to the group

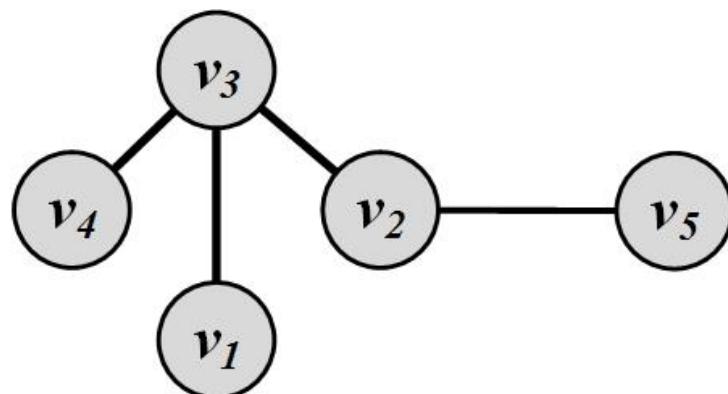
$$\begin{aligned}\bar{l}_S^{\text{group}} &= \frac{1}{|V-S|} \sum_{v_j \notin S} l_{S,v_j} \\ l_{S,v_j} &= \min_{v_i \in S} l_{v_i,v_j}\end{aligned}$$

One can also utilize the *maximum distance* or the *average distance*

# Group Centrality Example



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



Group degree centrality = **3**

Group betweenness centrality = **3**

Group closeness centrality = **1**



# Friendship Patterns

- Transitivity/Reciprocity
- Status/Balance

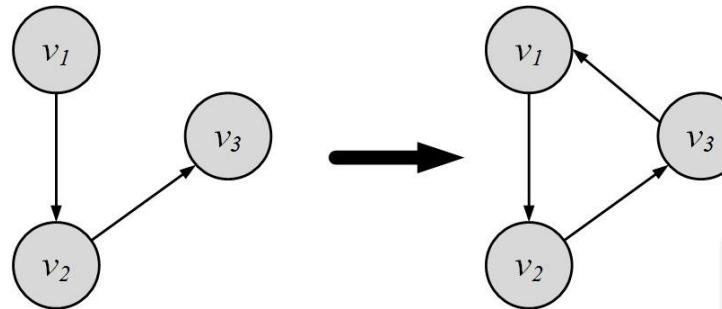


---

# I. Transitivity and Reciprocity

Mathematic representation:

For a transitive relation  $R$ :  $aRb \wedge bRc \rightarrow aRc$



**$cRa$  or  $aRc$  ?**

In a social network:

***Transitivity is when a friend of my friend is my friend***

Transitivity in a social network leads to a denser graph, which in turn is closer to a complete graph

We can determine how close graphs are to the complete graph by measuring transitivity

**Clustering coefficient** measures transitivity in undirected graphs

Count paths of length two and check whether the third edge exists

$$C = \frac{|\text{Closed Paths of Length 2}|}{|\text{Paths of Length 2}|}$$

When counting triangles, since every triangle has 6 closed paths of length 2

$$C = \frac{(\text{Number of Triangles}) \times 6}{|\text{Paths of Length 2}|}$$

# Clustering Coefficient and Triples



Or we can rewrite it as

$$C = \frac{(\text{Number of Triangles}) \times 3}{\text{Number of Connected Triples of Nodes}}$$

**Triple:** an ordered set of three nodes,

connected by two (open triple) edges or  
three edges (closed triple)

A triangle can miss any of its three edges

A triangle has **3 Triples**

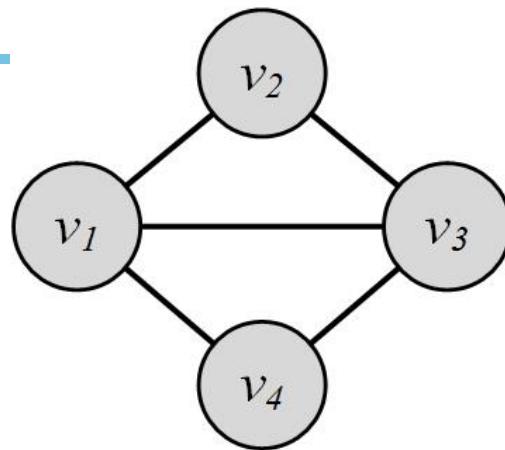
$v_i v_j v_k$  and  $v_j v_k v_i$  are different triples

- The same members
- First missing edge  $e(v_k, v_i)$  and second missing  $e(v_i, v_j)$

$v_i v_j v_k$  and  $v_k v_j v_i$  are the same triple

# [Global] Clustering Coefficient:

## Example



$$\begin{aligned} C &= \frac{\text{(Number of Triangles)} \times 3}{\text{Number of Connected Triples of Nodes}} \\ &= \frac{2 \times 3}{2 \times 3 + \underbrace{2}_{v_2 v_1 v_4, v_2 v_3 v_4}} = 0.75. \end{aligned}$$

Local clustering coefficient measures transitivity at the node level

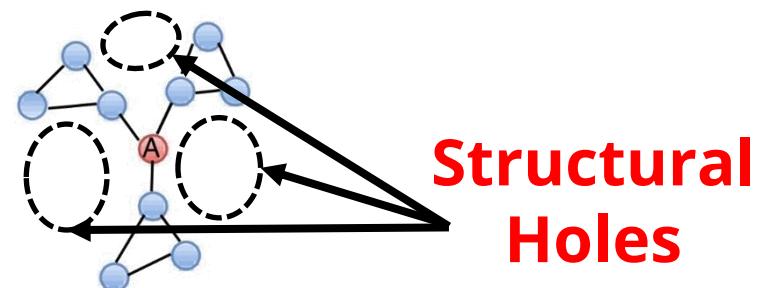
- Commonly employed for undirected graphs
- Computes how strongly neighbors of a node  $v$  (nodes adjacent to  $v$ ) are themselves connected

$$C(v_i) = \frac{\text{Number of Pairs of Neighbors of } v_i \text{ That Are Connected}}{\text{Number of Pairs of Neighbors of } v_i}.$$

In an undirected graph, the denominator can be rewritten as:

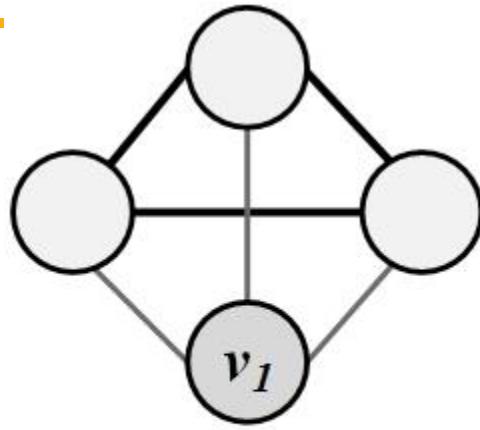
$$\binom{d_i}{2} = d_i(d_i - 1)/2$$

Provides a way to determine **structural holes**

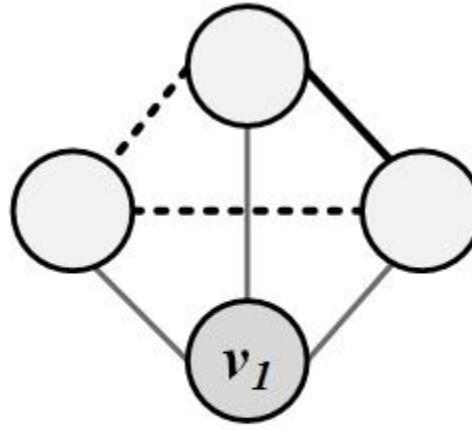


# Local Clustering Coefficient:

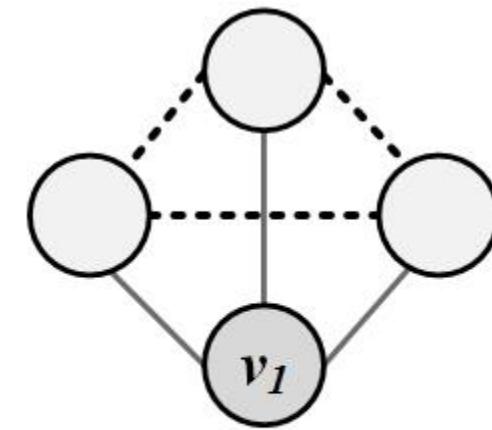
## Example



$$C(v_1) = 1$$



$$C(v_1) = 1/3$$



$$C(v_1) = 0$$

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  $C = 0$

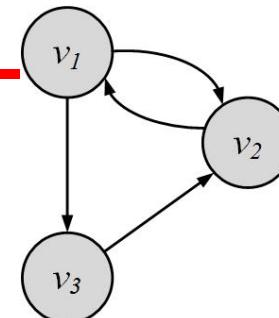
When all neighbors are connected  $C = 1$

***If you become my friend, I'll be yours***

Reciprocity is simplified version of transitivity

It considers closed loops of length 2

If node  $v$  is connected to node  $u$ ,  
 $u$  by connecting to  $v$ , exhibits reciprocity

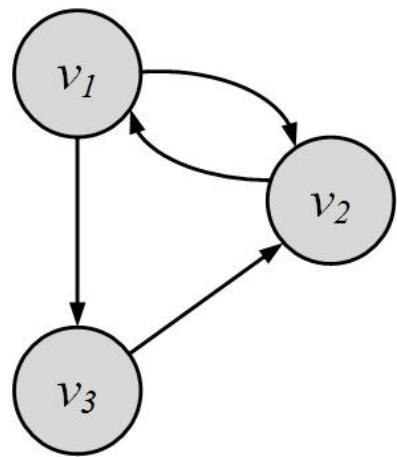


$$\begin{aligned}
 R &= \frac{\sum_{i,j,i < j} A_{i,j} A_{j,i}}{|E|/2}, \\
 &= \frac{2}{|E|} \sum_{i,j,i < j} A_{i,j} A_{j,i}, \\
 &= \frac{2}{|E|} \times \frac{1}{2} \text{Tr}(A^2), \\
 &= \frac{1}{|E|} \text{Tr}(A^2), \\
 &= \frac{1}{m} \text{Tr}(A^2).
 \end{aligned}$$

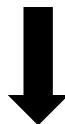
What about  $i = j$ ?

$$\text{Tr}(A) = A_{1,1} + A_{2,2} + \cdots + A_{n,n} = \sum_{i=1}^n A_{i,i}$$

# Reciprocity: Example



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



Reciprocal nodes:  $v_1, v_2$

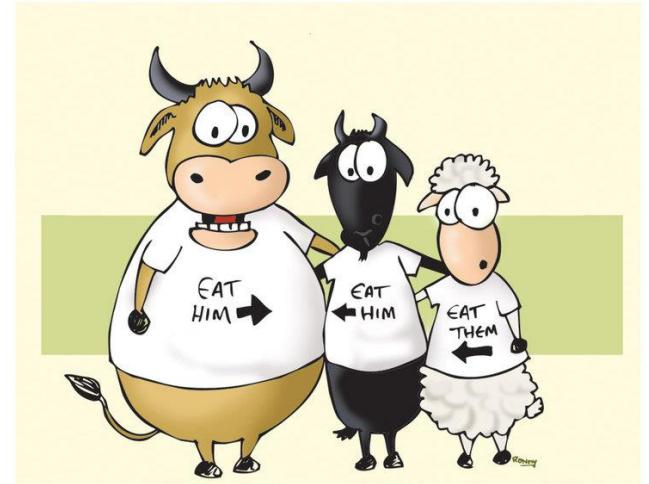
$$R = \frac{1}{m} \text{Tr}(A^2) = \frac{1}{4} \text{Tr} \left( \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \right) = \frac{2}{4} = \frac{1}{2}.$$



## II. Balance and Status



• Measuring consistency in friendships



## Social balance theory

Consistency in friend/foe relationships among individuals  
Informally, friend/foe relationships are consistent when

*The friend of my friend is my friend,  
The friend of my enemy is my enemy,  
The enemy of my enemy is my friend,  
The enemy of my friend is my enemy.*

### In the network

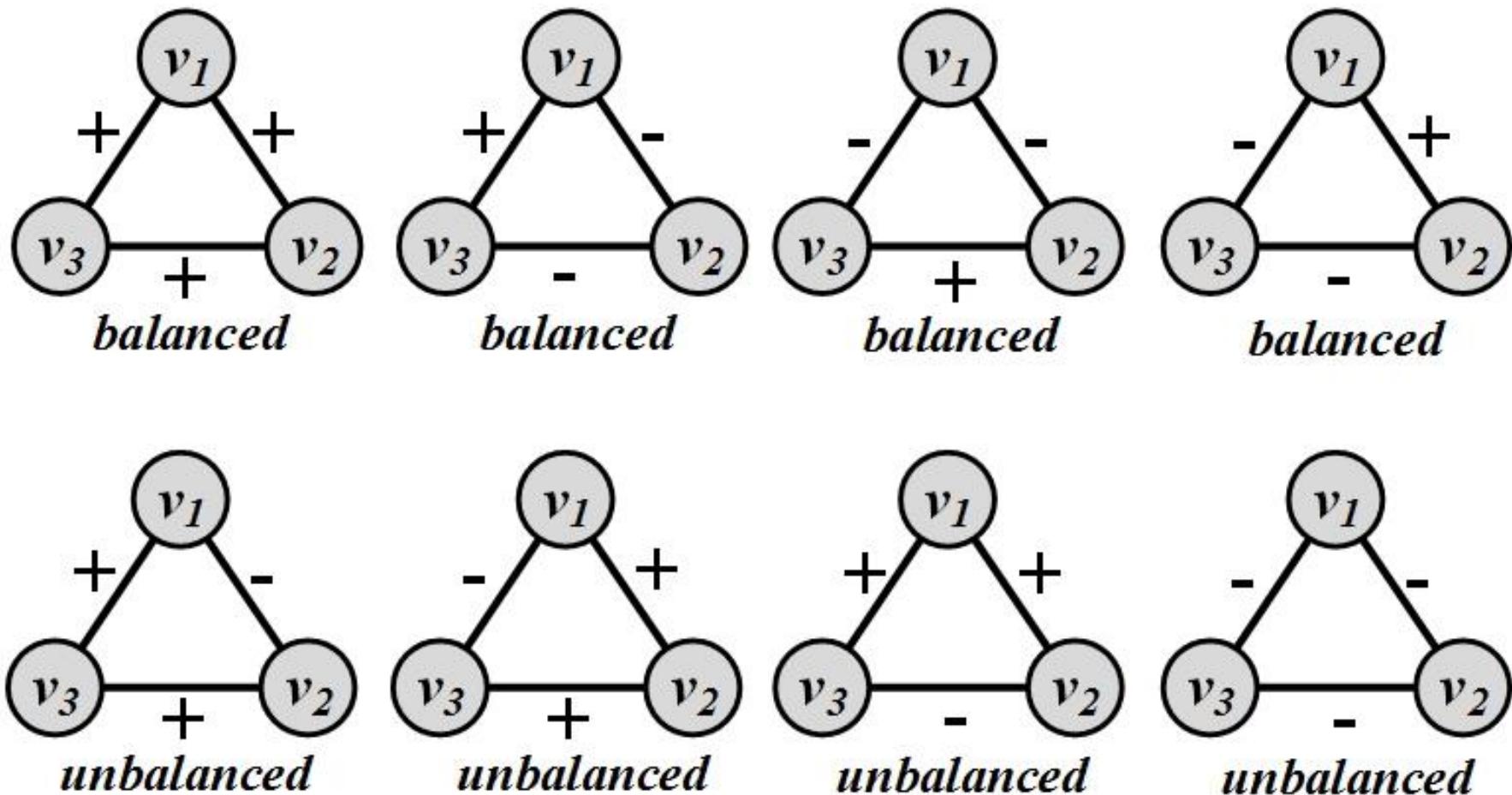
Positive edges demonstrate friendships ( $w_{ij} = 1$ )

Negative edges demonstrate being enemies ( $w_{ij} = -1$ )

Triangle of nodes  $i, j$ , and  $k$ , is balanced, if and only if  
 $w_{ij}$  denotes the value of the edge between nodes  $i$  and  $j$

$$w_{ij}w_{jk}w_{ki} \geq 0.$$

# Social Balance Theory: Possible Combinations



For any cycle, if the multiplication of edge values become positive, then the cycle is socially balanced

---

**Status:** how prestigious an individual is ranked within a society

## **Social status theory:**

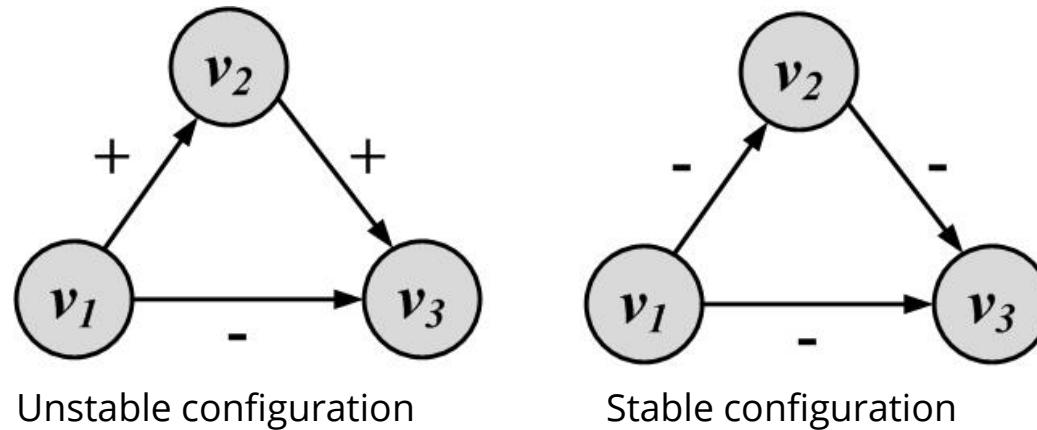
How consistent individuals are in assigning status to their neighbors

Informally,

*If X has a higher status than Y and Y has a higher status than Z, then X should have a higher status than Z.*

---

# Social Status Theory: Example



A directed '+' edge from node  $X$  to node  $Y$  shows that  $Y$  has a higher status than  $X$  and a '-' one shows vice versa



# Similarity

How similar are two nodes in a network?

- Structural Equivalence
- Regular Equivalence

## Structural Equivalence:

We look at the neighborhood shared by two nodes;  
The size of this shared neighborhood defines how similar two nodes are.

### ***Example:***

*Two brothers have in common  
sisters, mother, father, grandparents, etc.*

*This shows that they are similar,*

*Two **random** male or female individuals do not have  
much in common and are dissimilar.*

# 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  $v$ .

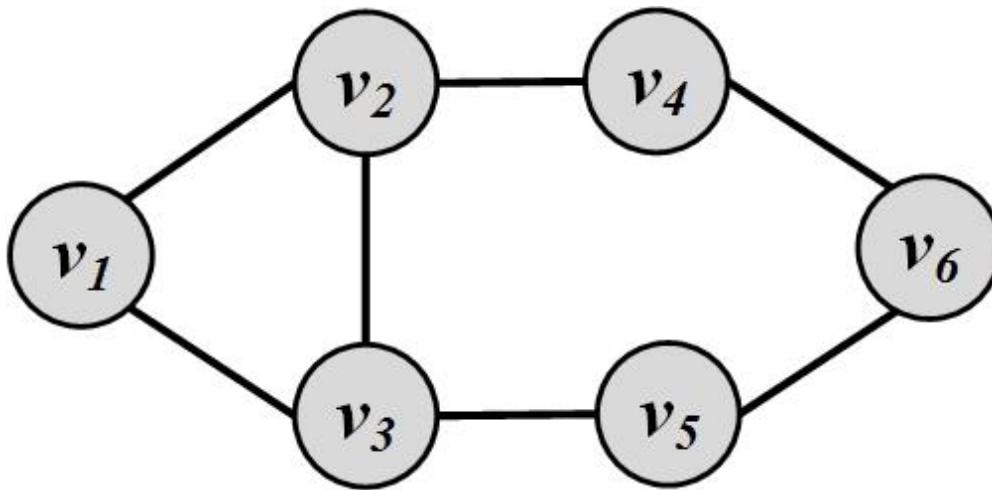
**What can go wrong?**

Connected nodes not sharing a neighbor will be assigned zero similarity

**Solution:**

We can assume nodes are included in their neighborhoods

# Similarity: Example



$$\sigma_{\text{Jaccard}}(v_2, v_5) = \frac{|\{v_1, v_3, v_4\} \cap \{v_3, v_6\}|}{|\{v_1, v_3, v_4, v_6\}|} = 0.25$$

$$\sigma_{\text{Cosine}}(v_2, v_5) = \frac{|\{v_1, v_3, v_4\} \cap \{v_3, v_6\}|}{\sqrt{|\{v_1, v_3, v_4\}| |\{v_3, v_6\}|}} = 0.40.$$

**Measuring Similarity Significance:** compare the calculated similarity value with its expected value where vertices pick their neighbors at random

For vertices  $v_i$  and  $v_j$  with degrees  $d_i$  and  $d_j$  this expectation is  $d_i d_j / n$

There is a  $d_i/n$  chance of becoming  $v_i$ 's neighbor  
 $v_j$  selects  $d_j$  neighbors

We can rewrite neighborhood overlap as

$$\sigma(v_i, v_j) = |N(v_i) \cap N(v_j)| = \sum_k A_{i,k} A_{j,k}$$

# Normalized Similarity, cont.



$$\begin{aligned}\sigma_{\text{significance}}(v_i, v_j) &= \sum_k A_{i,k} A_{j,k} - \frac{d_i d_j}{n} \quad \bar{A}_i = \frac{1}{n} \sum_k A_{i,k} \\ &= \sum_k A_{i,k} A_{j,k} - n \frac{1}{n} \sum_k A_{i,k} \frac{1}{n} \sum_k A_{j,k} \\ &= \sum_k A_{i,k} A_{j,k} - n \bar{A}_i \bar{A}_j \\ &= \sum_k (A_{i,k} A_{j,k} - \bar{A}_i \bar{A}_j) \\ &= \sum_k (A_{i,k} A_{j,k} - \bar{A}_i \bar{A}_j - \bar{A}_i \bar{A}_j + \bar{A}_i \bar{A}_j) \\ &= \sum_k (A_{i,k} A_{j,k} - A_{i,k} \bar{A}_j - \bar{A}_i A_{j,k} + \bar{A}_i \bar{A}_j) \\ &= \boxed{\sum_k (A_{i,k} - \bar{A}_i)(A_{j,k} - \bar{A}_j)} \quad \text{What is this?}\end{aligned}$$

# Normalized Similarity, cont.



**$n$  times the Covariance between  $A_i$  and  $A_j$**

$$\frac{1}{n} \sum_k (A_{i,k} - \bar{A}_i)(A_{j,k} - \bar{A}_j)$$

Normalize covariance by the multiplication of Variances.

$$\sqrt{\frac{1}{n} \sum_k (A_{i,k} - \bar{A}_i)^2} \sqrt{\frac{1}{n} \sum_k (A_{j,k} - \bar{A}_j)^2}$$

We get **Pearson correlation coefficient**

$$\begin{aligned}\sigma_{\text{pearson}}(v_i, v_j) &= \frac{\sigma_{\text{significance}}(v_i, v_j)}{\sqrt{\sum_k (A_{i,k} - \bar{A}_i)^2} \sqrt{\sum_k (A_{j,k} - \bar{A}_j)^2}} \\ &= \frac{\sum_k (A_{i,k} - \bar{A}_i)(A_{j,k} - \bar{A}_j)}{\sqrt{\sum_k (A_{i,k} - \bar{A}_i)^2} \sqrt{\sum_k (A_{j,k} - \bar{A}_j)^2}}\end{aligned}$$

(range of  $\sigma \in [-1,1]$ )

# Regular Equivalence

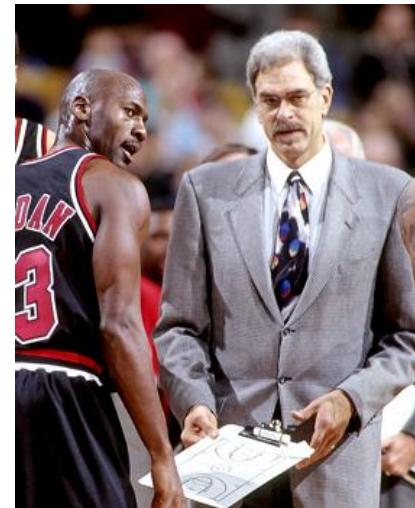


In regular equivalence,  
We **do not** look at  
neighborhoods shared  
between individuals, but  
**How neighborhoods  
themselves are similar**



## Example:

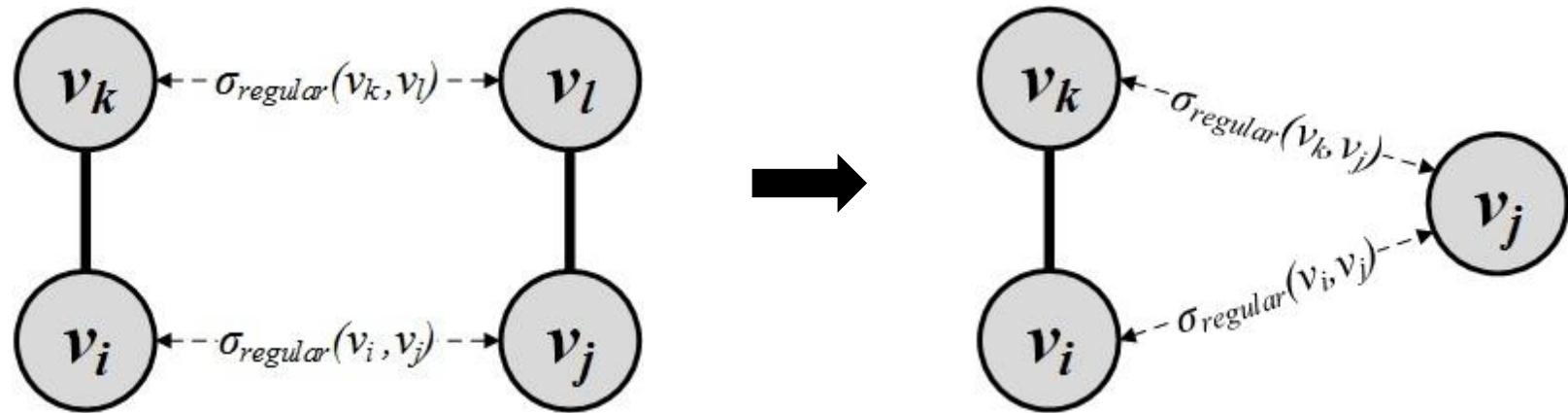
*Athletes are similar not  
because they know each  
other in person, but since  
they know similar  
individuals, such as  
coaches, trainers, other  
players, etc.*



# Regular Equivalence

- $v_i, v_j$  are similar when their neighbors  $v_k$  and  $v_l$  are similar

$$\sigma_{\text{regular}}(v_i, v_j) = \alpha \sum_{k,l} A_{i,k} A_{j,l} \sigma_{\text{regular}}(v_k, v_l)$$



- The equation (left figure) is hard to solve since it is self referential so we relax our definition using the right figure

# 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

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

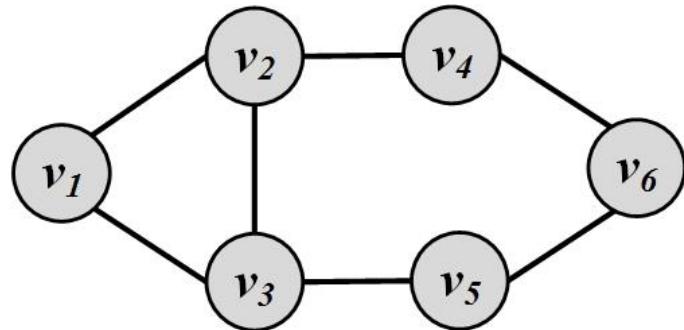
A vertex is highly similar to itself, we guarantee this by adding an identity matrix to the equation

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

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

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

# Regular Equivalence: Example



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

The largest eigenvalue of  $A$  is 2.43

Set  $\alpha = 0.3 < 1/2.43$

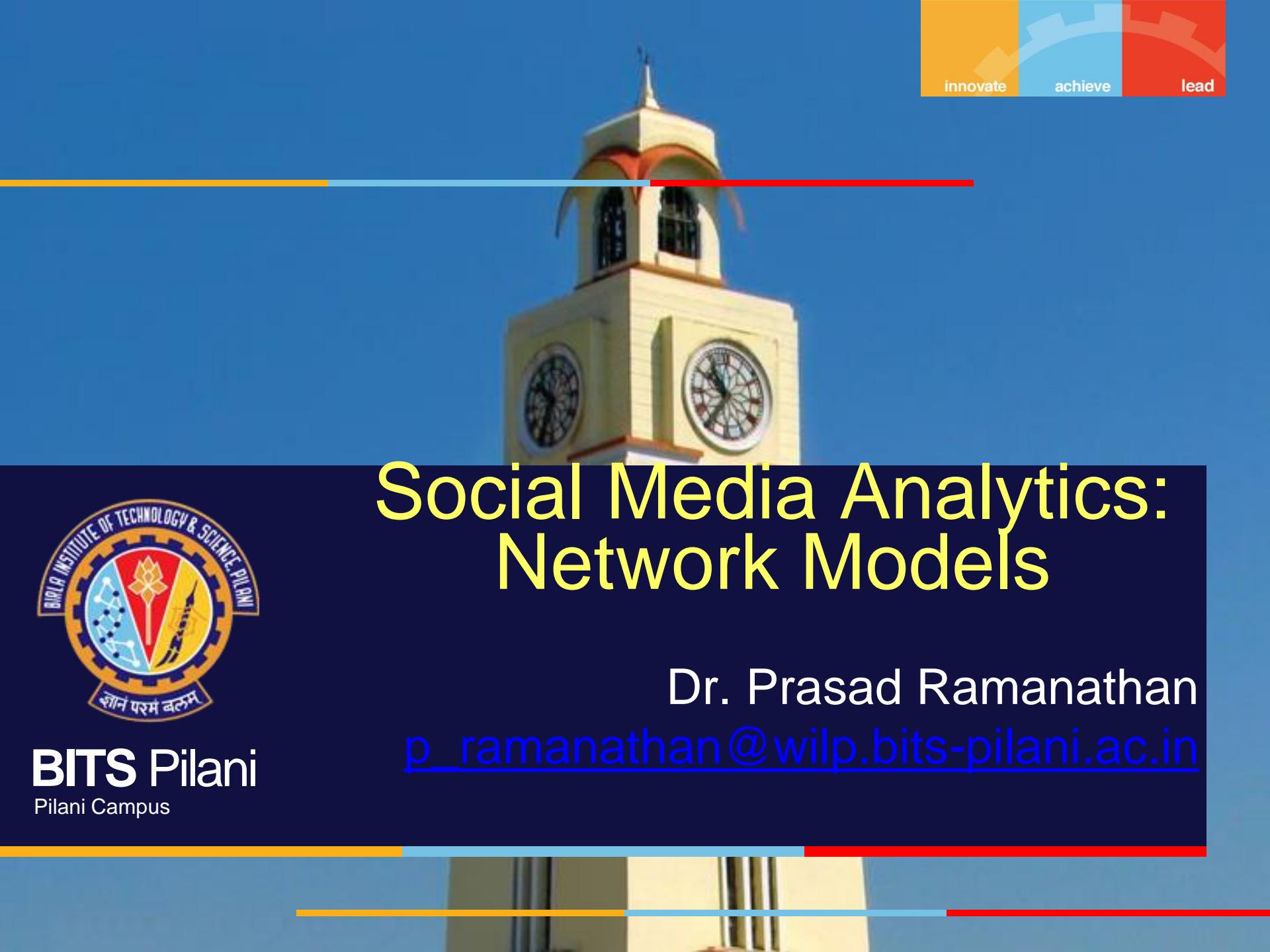
$$\sigma_{\text{regular}} = (I - 0.3A)^{-1} = \begin{bmatrix} 1.43 & 0.73 & 0.73 & 0.26 & 0.26 & 0.16 \\ 0.73 & 1.63 & 0.80 & 0.56 & 0.32 & 0.26 \\ 0.73 & 0.80 & 1.63 & 0.32 & 0.56 & 0.26 \\ 0.26 & 0.56 & 0.32 & 1.31 & 0.23 & 0.46 \\ 0.26 & 0.32 & 0.56 & 0.23 & 1.31 & 0.46 \\ 0.16 & 0.26 & 0.26 & 0.46 & 0.46 & 1.27 \end{bmatrix}$$

Any row/column of this matrix shows the similarity to other vertices  
Vertex 1 is most similar (other than itself) to vertices 2 and 3  
Nodes 2 and 3 have the highest similarity (**regular equivalence**)



---

# Thank you



# Social Media Analytics: Network Models



**BITS** Pilani  
Pilani Campus

Dr. Prasad Ramanathan

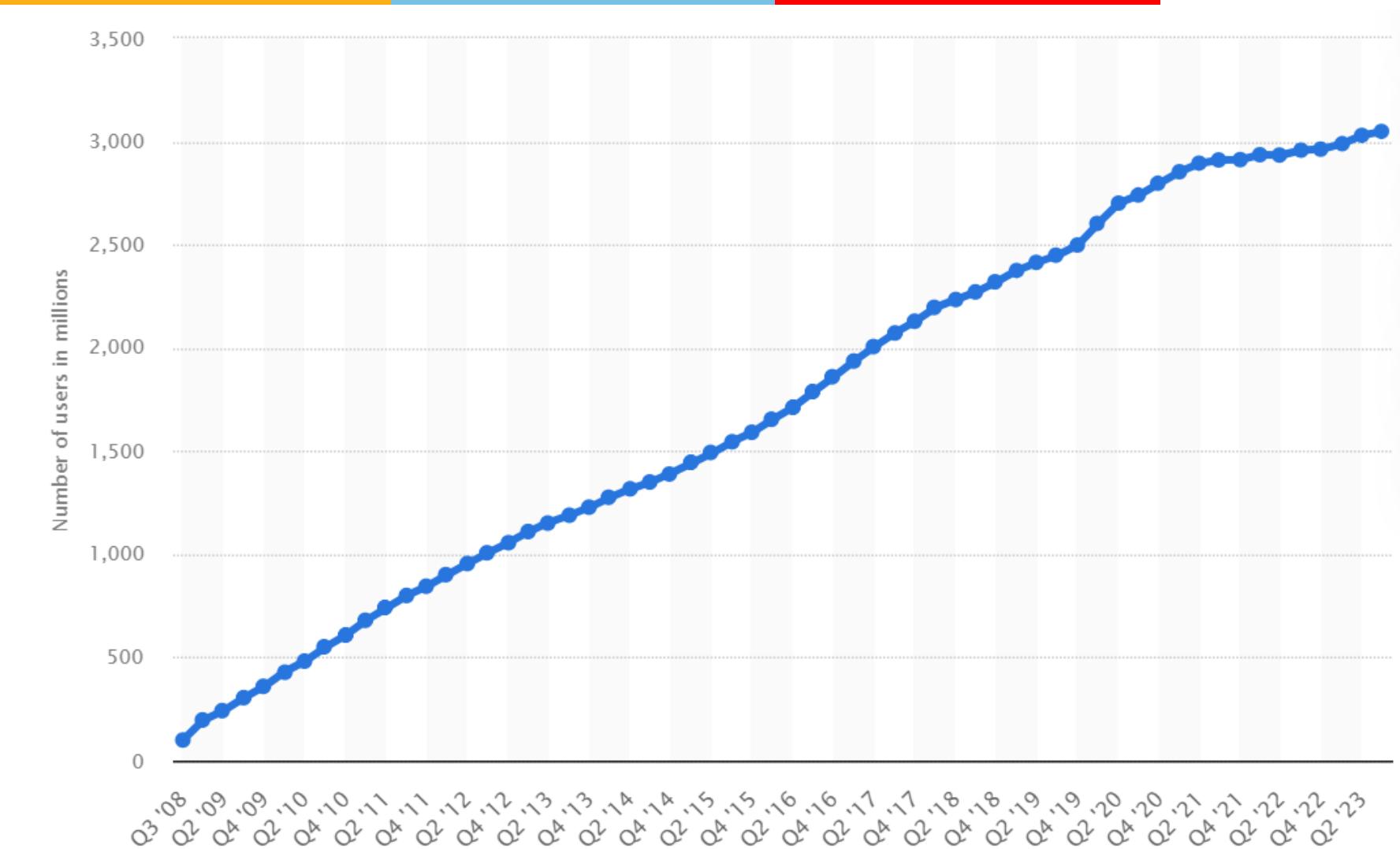
[p\\_ramanathan@wilp.bits-pilani.ac.in](mailto:p_ramanathan@wilp.bits-pilani.ac.in)

# Acknowledgment

Course material from the following source is gratefully acknowledged:

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/>**

# Number of monthly active Facebook users worldwide as of 3rd quarter 2023



# Why should I use network models?



## Facebook

**May 2011:**

- **721 millions** users.
- Average number of friends: **190**
- A total of **68.5 billion** friendships

**September 2015:**

- **1.35 Billion** users

**September 2023:**

- >3 billion users

1. What are the principal underlying processes that help initiate these friendships?
2. How can these seemingly independent friendships form this complex friendship network?
3. In social media there are many networks with millions of nodes and billions of edges.  
**They are complex and it is difficult to analyze them**

# So, what do we do?



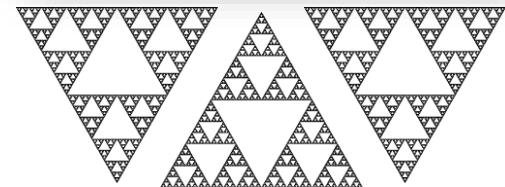
## Design models that generate graphs

The generated graphs should be similar to real-world networks.

If we can guarantee that generated graphs are similar to real-world networks:

1. We can analyze simulated graphs instead of real-networks (**cost-efficient**)
2. We can better understand real-world networks by providing concrete mathematical explanations; and
3. We can perform controlled experiments on synthetic networks when real-world networks are unavailable.

**What are properties of real-world networks that should be accurately modeled?**



**Basic Intuition:**

Hopefully! Our complex output [social network] is generated by a simple process



# Properties of Real-World Networks

**Power-law Distribution  
High Clustering Coefficient  
Small Average Path Length**



# Degree Distribution

## Wealth Distribution:

Most individuals have average capitals,  
Few are considered wealthy.

Exponentially more individuals with  
average capital than the wealthier  
ones.

## City Population:

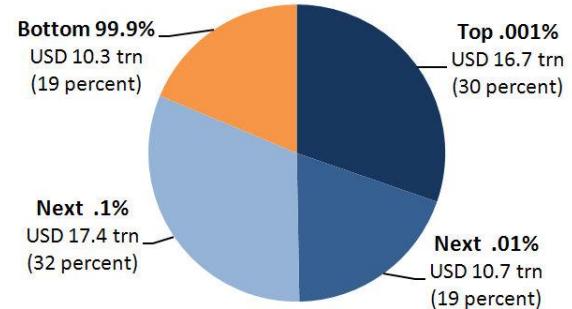
A few metropolitan areas are densely  
populated

Most cities have an average population  
size.

## Social Media:

We observe the same phenomenon  
regularly when measuring popularity  
or interestingness for entities.

Global Distribution of Wealth



James S. Henry, 2012



Herbert A Simon,  
On a Class of Skew Distribution Functions, 1955

The **Pareto principle**  
(80-20 rule): 80% of the effects  
come from 20% of the causes

## **Site Popularity:**

Many sites are visited less than a 1,000 times a month  
A few are visited more than a million times daily

## **User Activity:**

Social media users are often active on a few sites  
Some individuals are active on hundreds of sites

## **Product Price:**

There are exponentially more modestly priced products for sale compared to expensive ones.

## **Friendships:**

Many individuals with a few friends and a handful of users with thousands of friends

## **(Degree Distribution)**

When the frequency of an event changes as a power of an attribute

The frequency follows a **power-law**

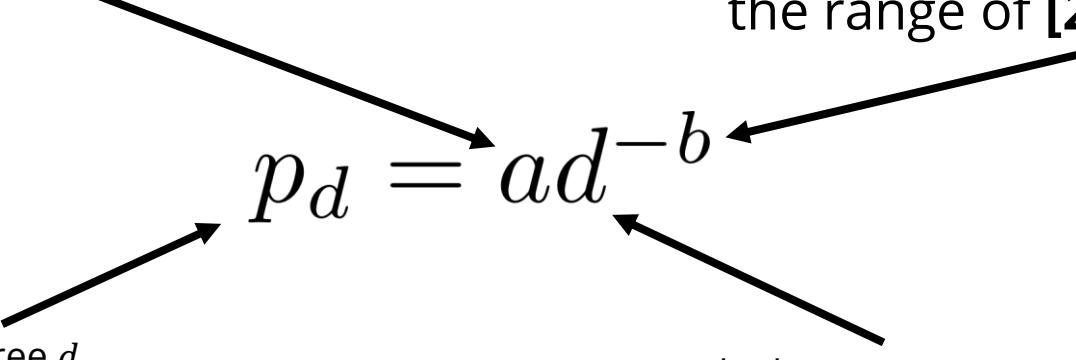
$$p_d = ad^{-b}$$

Power-law intercept

Fraction of users with degree  $d$

Node degree

The power-law exponent and its value is typically in the range of [2, 3]



$$\ln p_d = -b \ln d + \ln a$$

# Power-Law Distribution: Examples



## Call networks:

The fraction of telephone numbers that receive  $k$  calls per day is roughly proportional to  $1/k^2$

## Book Purchasing:

The fraction of books that are bought by  $k$  people is roughly proportional to  $1/k^3$

## Scientific Papers:

The fraction of scientific papers that receive  $k$  citations in total is roughly proportional to  $1/k^3$

## Social Networks:

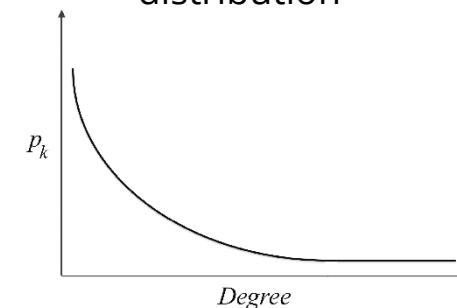
The fraction of users that have in-degrees of  $k$  is roughly proportional to  $1/k^2$

# Power-Law Distribution

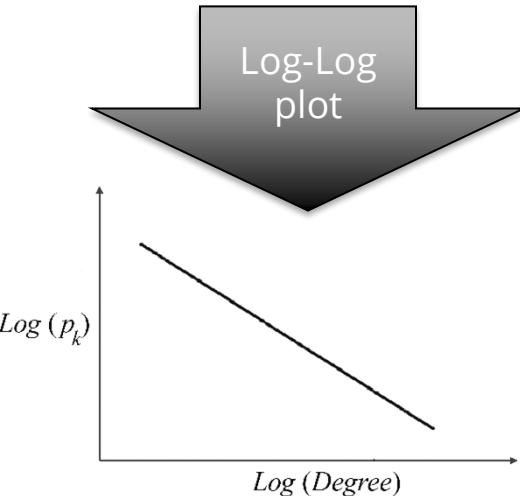


- Many real-world networks exhibit a *power-law* distribution.
- Power-laws seem to dominate
  - When the quantity being measured can be viewed as a type of **popularity**.
- A power-law distribution
  - **Small occurrences:** common
  - **Large instances:** extremely rare

A typical shape of a power-law distribution



(a) Power-Law Degree Distribution



(b) Log-Log Plot of Power-Law Degree Distribution

# Power-law Distribution: An Elementary Test



To test whether a network exhibits a power-law distribution

1. Pick a popularity measure and compute it for the whole network  
Example: number of friends for all nodes
2. Compute  $p_k$ , the fraction of individuals having popularity  $k$ .
3. Plot a log-log graph, where the  $x$ -axis represents  $\ln k$  and the  $y$ -axis represents  $\ln p_k$ .
4. If a power-law distribution exists, we should observe a straight line

**This is not a systematic approach!**

1. Other distributions could also exhibit this pattern
2. The results [estimations for parameters] can be biased and incorrect

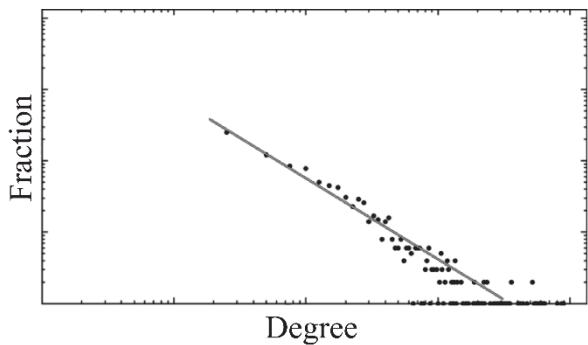
For a systematic approach see:

Clauset, Aaron, Cosma Rohilla Shalizi, and Mark EJ Newman. "Power-law distributions in empirical data." *SIAM review* 51(4) (2009): 661-703.

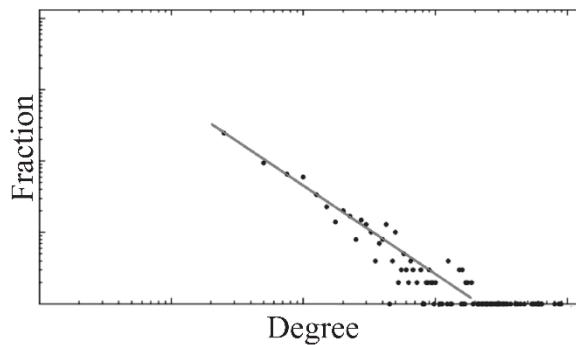
# Power-Law Distribution: Real-World Networks



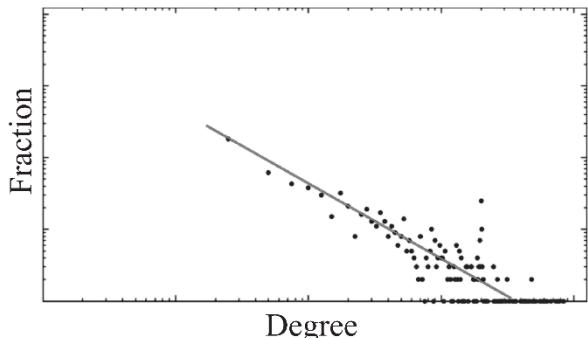
Networks with a power-law degree distribution are called **Scale-Free** networks



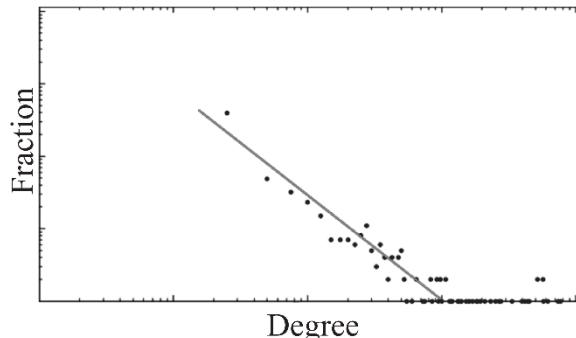
(a) Blog Catalog



(b) My Blog Log



(c) Twitter



(d) My Space

# The tail of the power-law distribution is long!

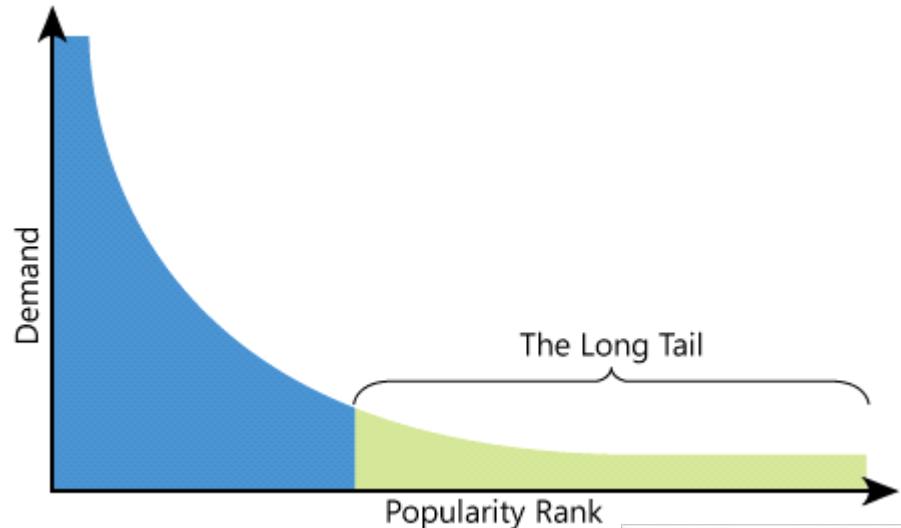


## The Loooooong Tail

Are most sales being generated by a small set of items that are enormously popular?

**OR**

By a much larger population of items that are each individually less popular?



The total sales volume of unpopular items, taken together, is very significant.

- 57% of Amazon's sales is from the long tail





# Clustering Coefficient

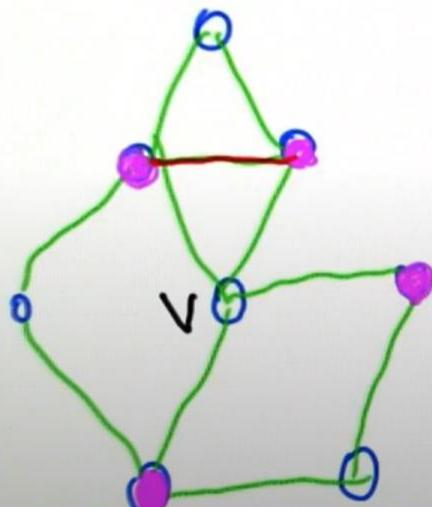
# Clustering Coefficient



Clustering Coefficient - Intro to Algorithms

## Clustering Coefficient

$CC(v)$ :



$v$ : a node

$k_v$ : its degree

$N_v$ : number of links between  
neighbors of  $v$

$$k_v = 4 \quad CC(v) = \frac{2N_v}{k_v(k_v-1)} = \frac{2 \cdot 1}{4 \cdot 3} = \frac{1}{6}$$

$CC(G)$ : average  $CC(v)$

Fraction of possible  
interconnections

$0 \leq CC(v) \leq 1$

STAR

CLIQUE

# Clustering Coefficient



In real-world networks, friendships are highly transitive



## Facebook

May 2011:

- Average clustering coefficient of **0.5** for users with **two** friends
- This indicates that for 50% of all users with two friends, their two friends were also friends with each other

- Friends of a user are often friends with one another
- These friendships form triads
- High average [local] clustering coefficient

Web	Facebook	Flickr	LiveJournal	Orkut	YouTube
0.081	0.14 (with 100 friends)	0.31	0.33	0.17	0.13

# Clustering Coefficient for Real-World Networks



	Network	Type	n	m	C
Social	Film actors	Undirected	449 913	25 516 482	0.20
	Company directors	Undirected	7 673	55 392	0.59
	Math coauthorship	Undirected	253 339	496 489	0.15
	Physics coauthorship	Undirected	52 909	245 300	0.45
	Biology coauthorship	Undirected	1 520 251	11 803 064	0.088
	Telephone call graph	Undirected	47 000 000	80 000 000	
	Email messages	Directed	59 812	86 300	
	Email address books	Directed	16 881	57 029	0.17
	Student dating	Undirected	573	477	0.005
	Sexual contacts	Undirected	2 810		
Information	WWW nd.edu	Directed	269 504	1 497 135	0.11
	WWW AltaVista	Directed	203 549 046	1 466 000 000	
	Citation network	Directed	783 339	6 716 198	
	Roget's Thesaurus	Directed	1 022	5 103	0.13
	Word co-occurrence	Undirected	460 902	16 100 000	
Technological	Internet	Undirected	10 697	31 992	0.035
	Power grid	Undirected	4 941	6 594	0.10
	Train routes	Undirected	587	19 603	
	Software packages	Directed	1 439	1 723	0.070
	Software classes	Directed	1 376	2 213	0.033
	Electronic circuits	Undirected	24 097	53 248	0.010
	Peer-to-peer network	Undirected	880	1 296	0.012
Biological	Metabolic network	Undirected	765	3 686	0.090
	Protein interactions	Undirected	2 115	2 240	0.072
	Marine food web	Directed	134	598	0.16
	Freshwater food web	Directed	92	997	0.20
	Neural network	Directed	307	2 359	0.18

Source: M. E. J Newman



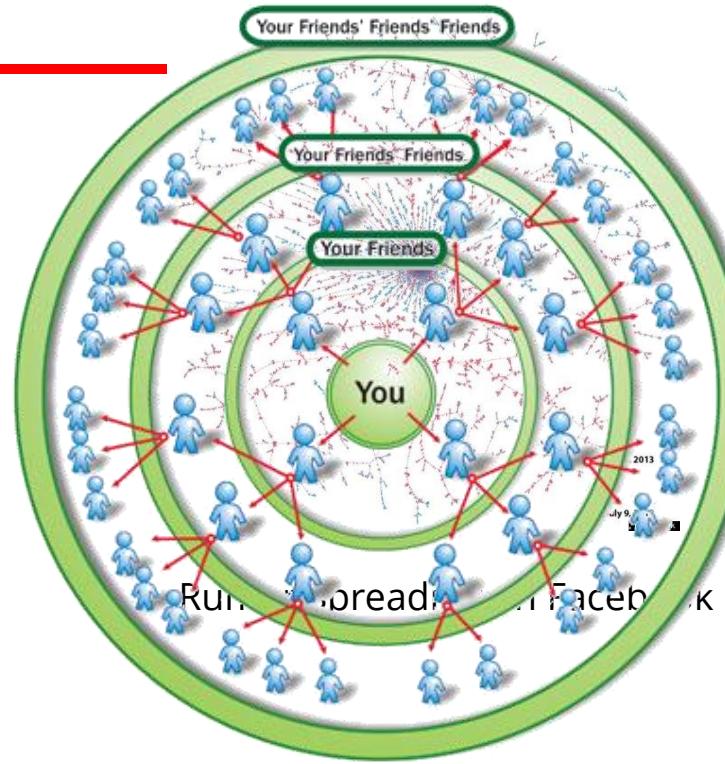
# Average Path Length

# How Small is the World?



A rumor is spreading over a social network.

- Assume all users pass it immediately to all of their friends



1. How long does it take to reach almost all of the nodes in the network?
2. What is the maximum time?
3. What is the average time?

# Milgram's Experiment



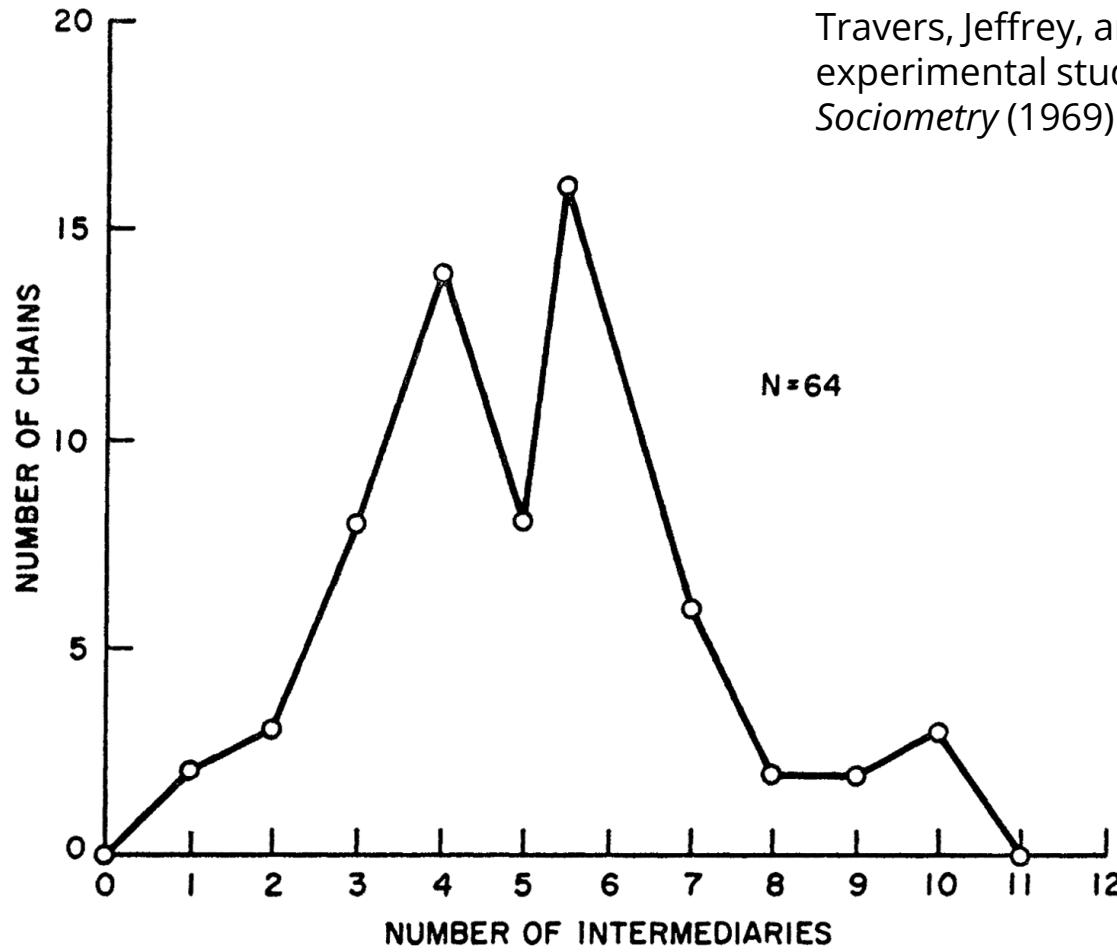
- 296 random people from Nebraska (196 people) and Boston (100 people) were asked to send a letter (via intermediaries) to a stock broker in Boston
- S/he could only send to people they personally knew, i.e., were on a first-name basis



Stanley Milgram (1933-1984)

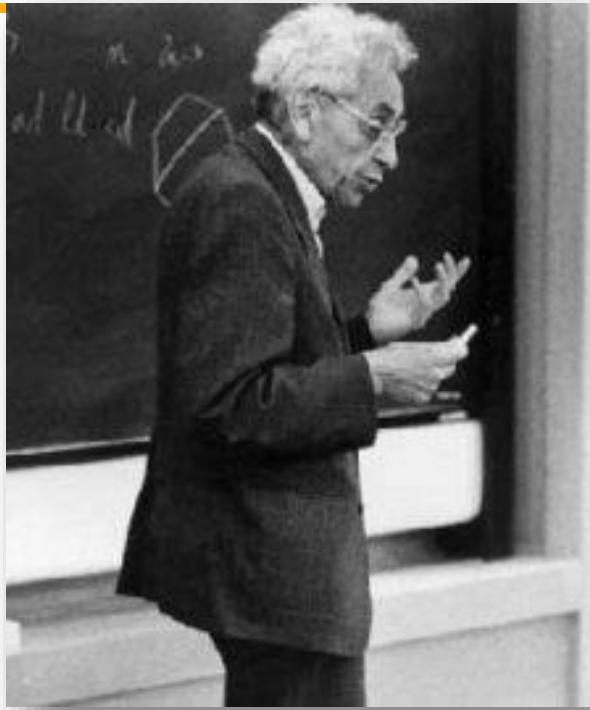
Among the letters that found the target (64), the average number of links was around **six**.

# Milgram's Experiment



Travers, Jeffrey, and Stanley Milgram. "An experimental study of the small world problem." *Sociometry* (1969): 425-443.

Average Number of Intermediate people is 5.2



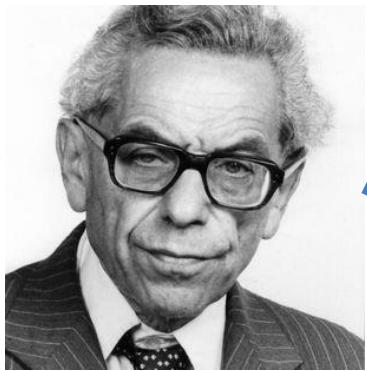
Paul Erdős (1913-1996)

- **Erdős Number:** Number of links required to connect scholars to Erdős, via co-authorship papers
- Erdős wrote 1500+ papers with 507 co-authors.
- The Erdős Number Project allows you to compute your Erdős number:
  - <http://www.oakland.edu/enp/>
- Connecting path lengths, among mathematicians only:
  - Avg. is **4.65** and Maximum is **13**

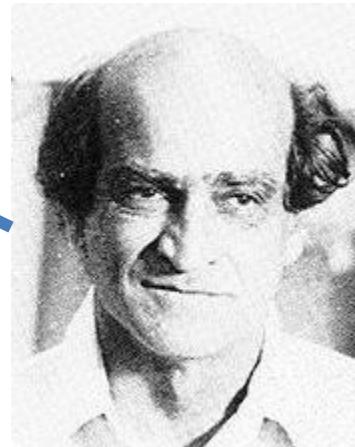
Watch Erdős's documentary "*N is a number*" on YouTube

# An Example of Erdős number 2

## [Einstein]

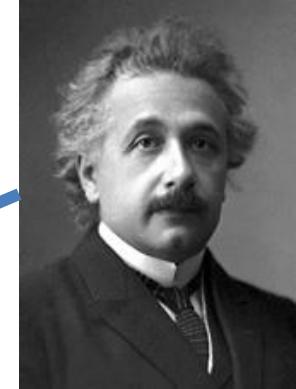


Paul Erdős (1913-1996)



Ernst Gabor Straus  
(1922-1983)

Erdős, Paul, B. Rothschild, and E. G. Straus. "Polychromatic Euclidean Ramsey theorems." *Journal of Geometry* 20.1 (1983): 28-35.



Albert Einstein (1879-1955)

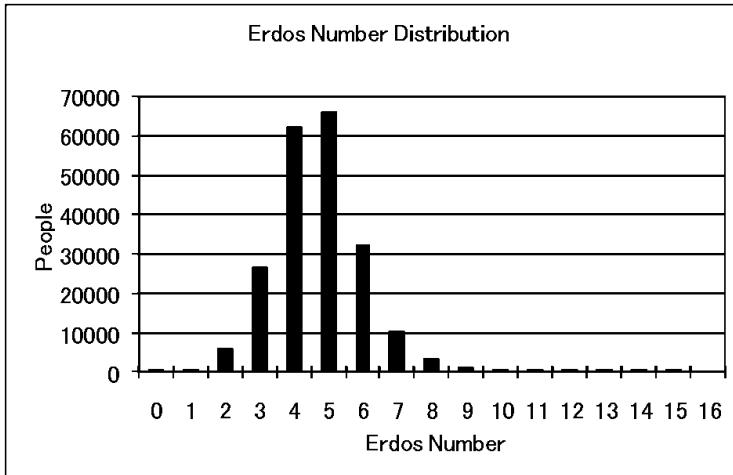
Einstein, Albert, and Ernst Gabor Straus. "A generalization of the relativistic theory of gravitation, II." *Annals of Mathematics* (1946): 731-741.

# Erdös number Distribution

innovate

achieve

lead



- The median Erdös number is **5**
- The mean is **4.65**
- The standard deviation is **1.27**

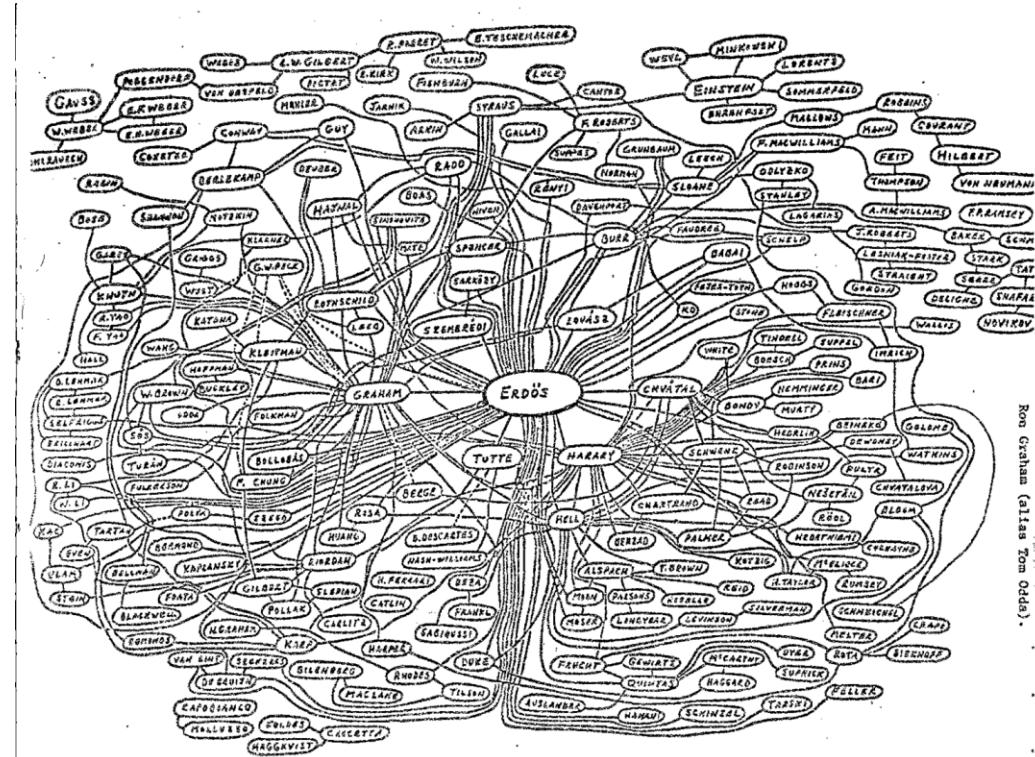


Figure 1  
To appear in Topics in Graph Theory (F. Harary, ed.) New York Academy of Sciences (1979).

Erdös Number Project:

<http://www.oakland.edu/enp/index.html>

# The Average Shortest Path



In real-world networks, any two members of the network are usually connected via a short paths.



## Facebook

- May 2011:
  - Average path length was **4.7**
  - **4.3** for US users

[Four degrees of separation]

## The average path length is small

Web	Facebook	Flickr	LiveJournal	Orkut	YouTube
16.12	4.7	5.67	5.88	4.25	5.10

# The Average Shortest Path in Sample Networks



	Network	Type	<i>n</i>	<i>m</i>	<i>l̄</i>
Social	Film actors	Undirected	449 913	25 516 482	3.48
	Company directors	Undirected	7 673	55 392	4.60
	Math coauthorship	Undirected	253 339	496 489	7.57
	Physics coauthorship	Undirected	52 909	245 300	6.19
	Biology coauthorship	Undirected	1 520 251	11 803 064	4.92
	Telephone call graph	Undirected	47 000 000	80 000 000	—
	Email messages	Directed	59 812	86 300	4.95
	Email address books	Directed	16 881	57 029	5.22
	Student dating	Undirected	573	477	16.01
Information	Sexual contacts	Undirected	2 810	—	—
	WWW nd.edu	Directed	269 504	1 497 135	11.27
	WWW AltaVista	Directed	203 549 046	1 466 000 000	16.18
	Citation network	Directed	783 339	6 716 198	—
	Roget's Thesaurus	Directed	1 022	5 103	4.87
Technological	Word co-occurrence	Undirected	460 902	16 100 000	—
	Internet	Undirected	10 697	31 992	3.31
	Power grid	Undirected	4 941	6 594	18.99
	Train routes	Undirected	587	19 603	2.16
	Software packages	Directed	1 439	1 723	2.42
	Software classes	Directed	1 376	2 213	5.40
	Electronic circuits	Undirected	24 097	53 248	11.05
Biological	Peer-to-peer network	Undirected	880	1 296	4.28
	Metabolic network	Undirected	765	3 686	2.56
	Protein interactions	Undirected	2 115	2 240	6.80
	Marine food web	Directed	134	598	2.05
	Freshwater food web	Directed	92	997	1.90
	Neural network	Directed	307	2 359	3.97

*l̄*: average path length

# More Properties of Real-World Networks



## Friendship Paradox [Feld 1991]

i.e., your friends, on average, have more friends than you

### Why?

High degree nodes appear in many averages when averaging over friends

It holds for 98% of Twitter Users [Hodas et al. 2013]

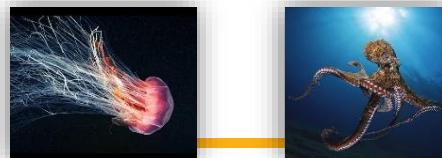
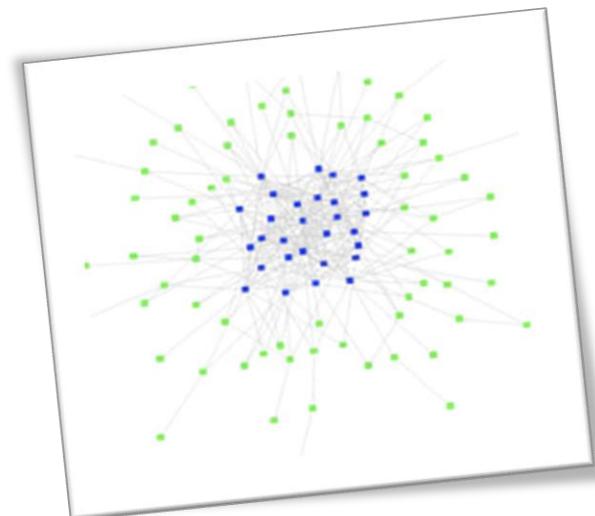
## Core-Periphery Structure

Dense Core

Periphery nodes that connects to the core, but not connected among themselves

Also known as

**Jellyfish** or **Octopus** structures



---

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.

---



# Network Models

- Model-Driven Models!

**Random graphs  
Small-World Model  
Preferential Attachment**



# Random Graphs

We have to assume how friendships are formed  
The most basic form:

**Random Graph assumption:**  
*Edges (i.e., friendships) between nodes (i.e., individuals) are formed randomly.*

We discuss two random graph models  $G(n, p)$  and  $G(n, m)$

Consider a graph with a fixed number of nodes  $n$

Any of the  $\binom{n}{2}$  edges can be formed independently, with probability  $p$

The graph is called a  $G(n, p)$  random graph

Proposed independently by Edgar Gilbert and by Solomonoff and Rapoport.

# Random Graph Model - $G(n, m)$



Assume both number of nodes  $n$  and number of edges  $m$  are fixed.

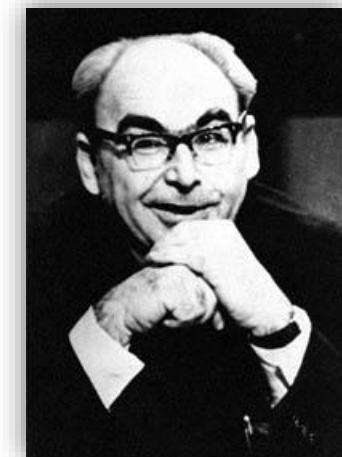
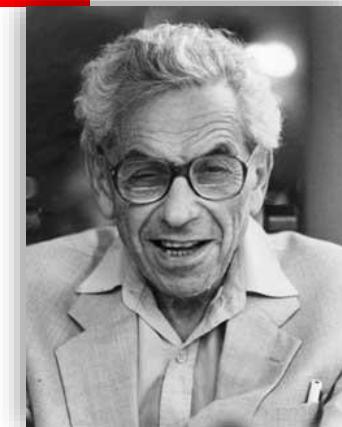
Determine which  $m$  edges are selected from the set of possible edges

Let  $\Omega$  denote the set of graphs with  $n$  nodes and  $m$  edges

There are  $|\Omega|$  different graphs with  $n$  nodes and  $m$  edges

$$|\Omega| = \binom{n}{m}$$

To generate a random graph, we uniformly select one of the  $|\Omega|$  graphs (the selection probability is  $1/|\Omega|$ )



This model was first proposed by  
Paul Erdös and Alfred Rényi

## Similarities:

In the limit (when  $n$  is large), both  $G(n, p)$  and  $G(n, m)$  models act similarly

The expected number of edges in  $G(n, p)$  is  $\binom{n}{2}p$

We can set  $\binom{n}{2}p = m$  and in the limit, we should get similar results

## Differences:

The  $G(n, m)$  model contains a fixed number of edges

The  $G(n, p)$  model is likely to contain none or all possible edges

The expected number of edges connected to a node (expected degree) in  $G(n, p)$  is  $c = (n - 1)p$

## Proof.

A node can be connected to at most  $n - 1$  nodes  
or  $n - 1$  edges

All edges are selected independently with probability  $p$   
Therefore, on average,  $(n - 1)p$  edges are selected

$c = (n - 1)p$  or equivalently,

$$p = \frac{c}{n-1}$$

# Expected Number of Edges



---

The expected number of edges in  $G(n, p)$  is  $\binom{n}{2}p$

## Proof.

Since edges are selected independently, and we have a maximum  $\binom{n}{2}$  edges, the expected number of edges is  $\binom{n}{2}p$

---

# The probability of observing $m$ edges



Given the  $G(n, p)$  model, the probability of observing  $m$  edges is the binomial distribution

$$P(|E| = m) = \binom{n}{m} p^m (1 - p)^{\binom{n}{2} - m}$$

## Proof.

$m$  edges are selected from the  $\binom{n}{2}$  possible edges.

These  $m$  edges are formed with probability  $p^m$  and other edges are not formed (to guarantee the existence of only  $m$  edges) with probability

$$(1 - p)^{\binom{n}{2} - m}$$



# Evolution of Random Graphs

- Create your own Random Graph Evolution demo:  
<https://github.com/dgleich/erdosrenyi-demo>

# The Giant Component

innovate

achieve

lead

In random graphs, as we increase  $p$ ,  
a large fraction of nodes start  
getting connected  
i.e., we have a path between any pair

This large fraction forms a  
connected component:  
**Largest connected component**, also  
known as the **Giant component**

In random graphs:

$$p = 0$$

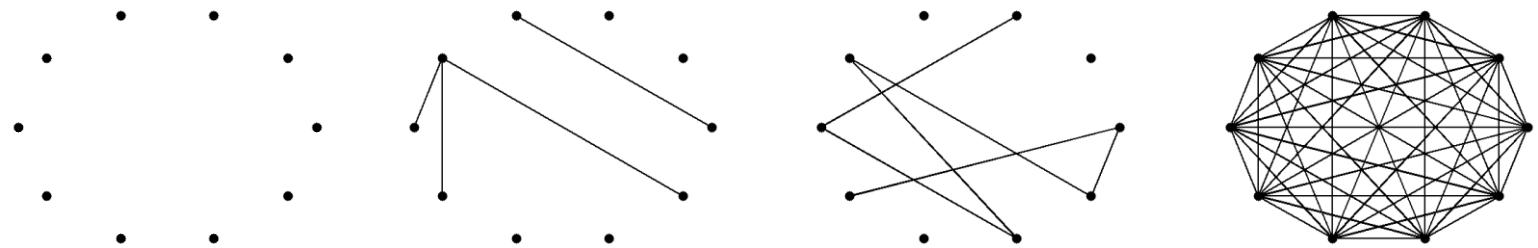
the size of the giant component is 0

$$p = 1$$

the size of the giant component is  $n$

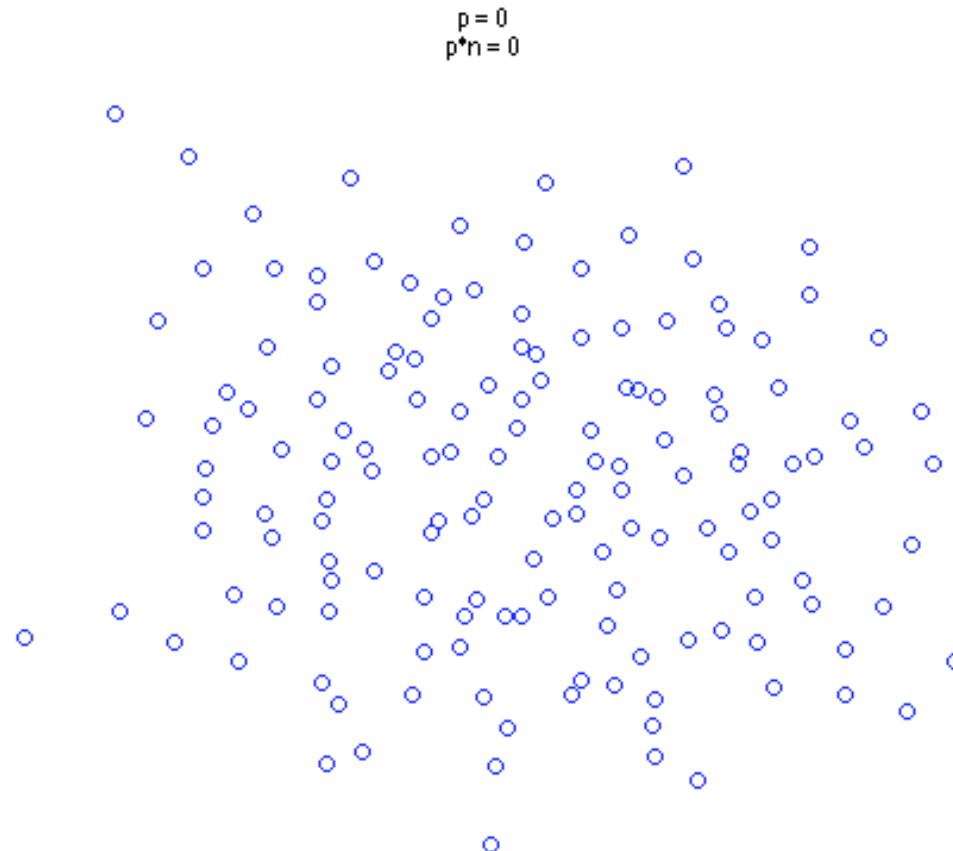


# The Giant Component



Probability ( $p$ )	0.0	0.055	0.11	1.0
Average Node Degree ( $c$ )	0.0	0.8	$\approx 1$	$n-1=9$
Diameter	0	2	6	1
Giant Component Size	0	4	7	10
Average Path Length	0.0	1.5	2.66	1.0

# Demo ( $n = 150$ )



# 1<sup>st</sup> Phase Transition (Rise of the Giant Component)



**Phase Transition:** the point where diameter value starts to shrink in a random graph

We have other phase transitions in random graphs

E.g., when the graph becomes connected

The phase transition we focus on happens when average node degree  $c = 1$  (or when  $p = 1/(n - 1)$ )

At this Phase Transition:

1. The giant component, which just started to appear, starts to grow, and
2. The diameter, which *just* reached its maximum value, starts decreasing.

# Random Graphs



If  $c < 1$ :

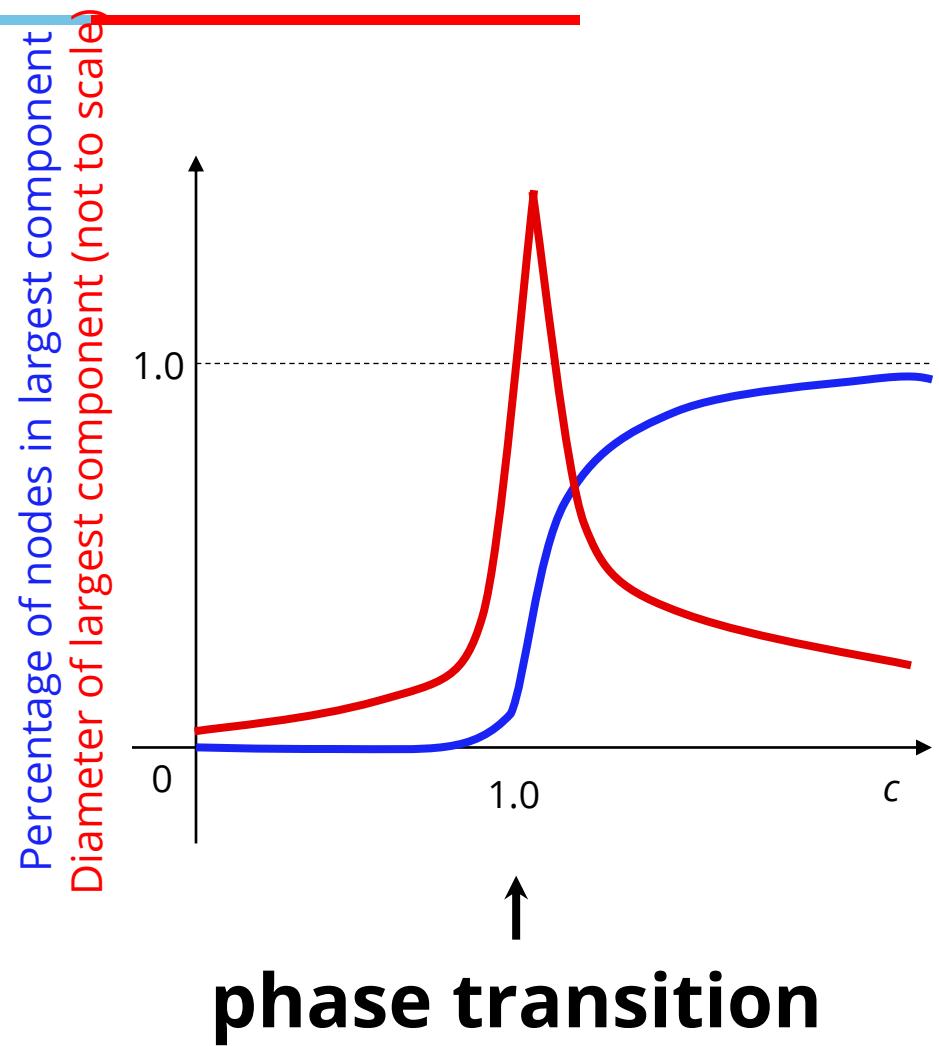
- small, isolated clusters
- small diameters
- short path lengths

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**



# Why $c = 1$ ? [Rough Idea]



Consider a random graph with expected node degree  $c$

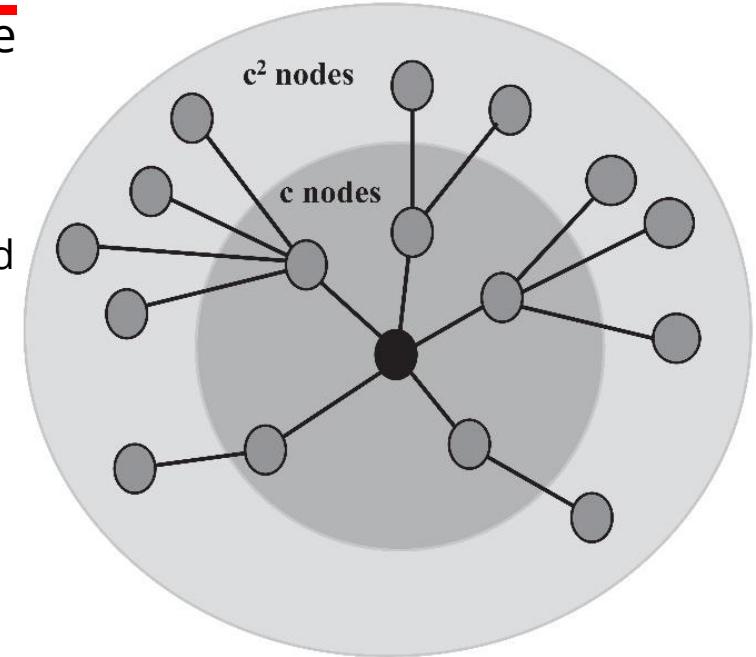
In this graph,

Consider any **connected** set of nodes  $S$ ;  
Let  $S' = V - S$  denote the complement set; and  
Assume  $|S| \ll |S'|$ .

For any node  $v$  in  $S$ ,

If we move one hop away from  $v$ , we visit approximately  $c$  nodes.

If we move one hop away from nodes in  $S$ ,  
we visit approximately  $|S|c$  nodes.



- If  $S$  is small, the nodes in  $S$  only visit nodes in  $S'$  and when moving one hop away from  $S$ , the set of nodes *guaranteed to be connected* gets larger by a factor  $c$ .
- In the limit, if we want this connected component to become the largest component, then after traveling  $n$  hops, its size must grow and we must have

$$c^n \geq 1 \text{ or equivalently } c \geq 1$$



# 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**

When computing degree distribution, we estimate the probability of observing  $P(d_v = d)$  for node  $v$

For a random graph generated by  $G(n, p)$ , this probability is

$$P(d_v = d) = \binom{n-1}{d} p^d (1-p)^{n-1-d}$$

This is a binomial degree distribution. In the limit this will become the Poisson degree distribution

Thus, in the limit, random graphs generate Poisson degree distribution, which differs from the power-law degree distribution observed in real-world networks.

# Expected Local Clustering Coefficient



The expected local clustering coefficient for node  $v$  of a random graph generated by  $G(n, p)$  is  $p$

## Proof.

$$C(v) = \frac{\text{number of connected pairs of } v\text{'s neighbors}}{\text{number of pairs of } v\text{'s neighbors}}$$

$v$  can have different degrees depending on the random procedure so the expected value is

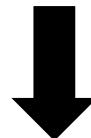
$$\mathbf{E}(C(v)) = \sum_{d=0}^{n-1} \mathbf{E}(C(v)|d_v = d) P(d_v = d)$$



# Expected Local Clustering Coefficient, Cont.

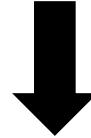


$$\mathbf{E}(C(v)) = \sum_{d=0}^{n-1} \mathbf{E}(C(v)|d_v = d) P(d_v = d)$$



$$\mathbf{E}(C(v)|d_v = d) = \frac{\text{number of connected pairs of } v\text{'s } d \text{ neighbors}}{\text{number of pairs of } v\text{'s neighbors}}$$

$$= \frac{p \binom{d}{2}}{\binom{d}{2}} = p$$



Sums up to 1

$$\mathbf{E}(C(v)) = p \sum_{d=0}^{n-1} P(d_v = d) = p$$

---

The global clustering coefficient of a random graph generated by  $G(n, p)$  is  $p$

## Proof.

The global clustering coefficient defines the probability of two neighbors of the same node being connected.

In a random graph, for any two nodes, this probability is the same

Equal to the generation probability  $p$  that determines the probability of two nodes getting connected

# Random Graphs – Clustering Coefficient



In random graphs, the clustering coefficient is equal to the probability  $p$ ; therefore, by appropriately selecting  $p$ , we can generate networks with a high clustering coefficient.

Note that selecting a large  $p$  is undesirable because doing so will generate a very dense graph, which is unrealistic, as in the real-world, networks are often sparse.

Thus, **random graphs** are considered generally **incapable of generating networks with high clustering coefficients** without compromising other required properties.

# The Average Path Length

## [Rough Idea]



The average path length in a random graph is

$$l \approx \frac{\ln |V|}{\ln c}$$

### Proof.

- Assume  $D$  is the expected diameter of the graph
- Starting with any node and the expected degree  $c$ ,
  - one can visit approximately  $c$  nodes by traveling one edge
  - $c^2$  nodes by traveling 2 edges, and
  - $c^D$  nodes by traveling diameter number of edges
- We should have visited all nodes  $c^D \approx |V|$
- The expected diameter size tends to the average path length  $l$  in the limit

$$c^D \approx c^l \approx |V|$$



$$l \approx \frac{\ln |V|}{\ln c}$$

Compute the average degree  $c$  in the real-world graph

Compute  $p$  using  $c/(n - 1) = p$

Generate the random graph using  $p$

How representative is the generated graph?

**[Degree Distribution]** Random graphs do not have a power-law degree distribution

**[Average Path Length]** Random graphs perform well in modeling the average path lengths

**[Clustering Coefficient]** Random graphs drastically underestimate the clustering coefficient

# Real-World Networks / Simulated Random Graphs



Network	Original Network				Simulated Random Graph	
	Size	Average Degree	Average Path Length	C	Average Path Length	C
Film Actors	225,226	61	3.65	0.79	2.99	0.00027
Medline Coauthorship	1,520,251	18.1	4.6	0.56	4.91	$1.8 \times 10^{-4}$
E.Coli	282	7.35	2.9	0.32	3.04	0.026
C.Elegans	282	14	2.65	0.28	2.25	0.05

Random graphs **perform well in modelling the average path lengths.**

However, when considering the transitivity, the random graph model **drastically underestimates the clustering coefficient.**

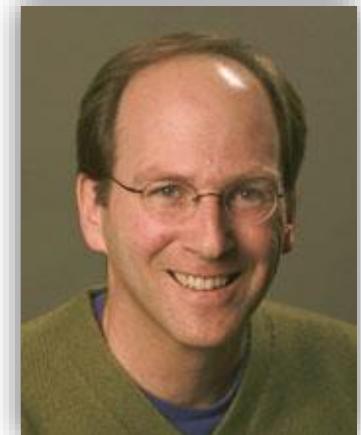


# Small-World Model

Small-world model  
or the **Watts-Strogatz (WS)** model  
A special type of random graph  
Exhibits small-world properties:  
Short average path length  
High clustering coefficient



It was proposed by Duncan J. Watts  
and Steven Strogatz in their joint  
1998 Nature paper



Watts, Duncan J., and Steven H. Strogatz.  
"Collective dynamics of 'small-world' networks."  
*nature* 393.6684 (1998): 440-442.

# Small-world Model



In real-world interactions, many individuals have a limited and often at least, a fixed number of connections

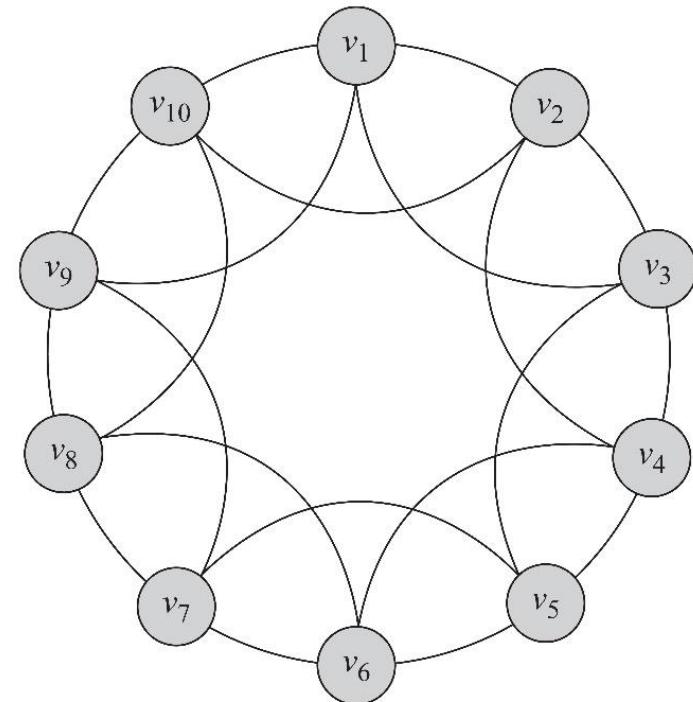
In graph theory terms, this assumption is equivalent to embedding users in a regular network

A regular (ring) lattice is a special case of regular networks where there exists a certain pattern on how ordered nodes are connected to one another

In a regular lattice of degree  $c$ , nodes are connected to their previous  $c/2$  and following  $c/2$  neighbors

Formally, for node set  $V=\{v_1, v_2, v_3, \dots, v_n\}$ , an edge exists between node  $i$  and  $j$  if and only if

$$0 \leq \min(n - |i - j|, |i - j|) \leq c/2$$

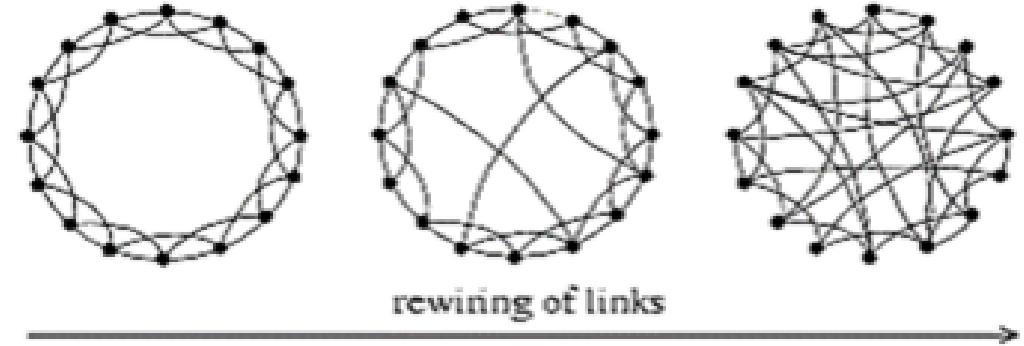


# Generating a Small-World Graph



The lattice has a **high**, but **fixed**, clustering coefficient

The lattice has a **high** average path length



- In the small-world model, a parameter  $0 \leq \beta \leq 1$  controls randomness in the model
  - When  $\beta$  is 0, the model is basically a regular lattice
  - When  $\beta = 1$ , the model becomes a random graph
- The model starts with a regular lattice and starts adding random edges [through **rewiring**]
  - **Rewiring:** take an edge, change one of its end-points randomly

---

## Algorithm 4.1 Small-World Generation Algorithm

---

**Require:** Number of nodes  $|V|$ , mean degree  $c$ , parameter  $\beta$

- 1: **return** A small-world graph  $G(V, E)$
  - 2:  $G =$  A regular ring lattice with  $|V|$  nodes and degree  $c$
  - 3: **for** node  $v_i$  (starting from  $v_1$ ), and all edges  $e(v_i, v_j)$ ,  $i < j$  **do**
  - 4:    $v_k =$  Select a node from  $V$  uniformly at random.
  - 5:   **if** rewiring  $e(v_i, v_j)$  to  $e(v_i, v_k)$  does not create loops in the graph or multiple edges between  $v_i$  and  $v_k$  **then**
  - 6:     rewire  $e(v_i, v_j)$  with probability  $\beta$ :  $E = E - \{e(v_i, v_j)\}$ ,  $E = E \cup \{e(v_i, v_k)\}$ ;
  - 7:   **end if**
  - 8: **end for**
  - 9: Return  $G(V, E)$
- 

As in many network generating algorithms

- Disallow self-edges
  - Disallow multiple edges
-

# Small World Networks – Key observations



The network generated using this procedure has some interesting properties:

Depending on the  $\beta$  value, it can have a **high clustering coefficient**, and also **short average path lengths**.

The **degree distribution**, however, still does not match that of real-world networks.



# Small-World Model Properties

- The degree distribution for the small-world model is

$$P(d_v = d) = \sum_{n=0}^{\min(d-c/2, c/2)} \binom{c/2}{n} (1-\beta)^n \beta^{c/2-n} \frac{(\beta c/2)^{d-c/2-n}}{(d-c/2-n)!} e^{-\beta c/2}$$

- In practice, in the graph generated by the small world model, most nodes have similar degrees due to the underlying lattice.

**Degree distribution is quite similar to the Poisson degree distribution observed in random graphs**

# Regular Lattice vs. Random Graph



- Regular Lattice:
  - Clustering Coefficient (**high**):
$$\frac{3(c-2)}{4(c-1)} \approx \frac{3}{4}$$
  - Average Path Length (**high**):  $n/2c$
- Random Graph:
  - Clustering Coefficient (**low**):  $p$
  - Average Path Length (**ok!**) :  $\ln |V| / \ln c$

# What happens in Between?



- Does smaller average path length mean smaller clustering coefficient?
- Does larger average path length mean larger clustering coefficient?

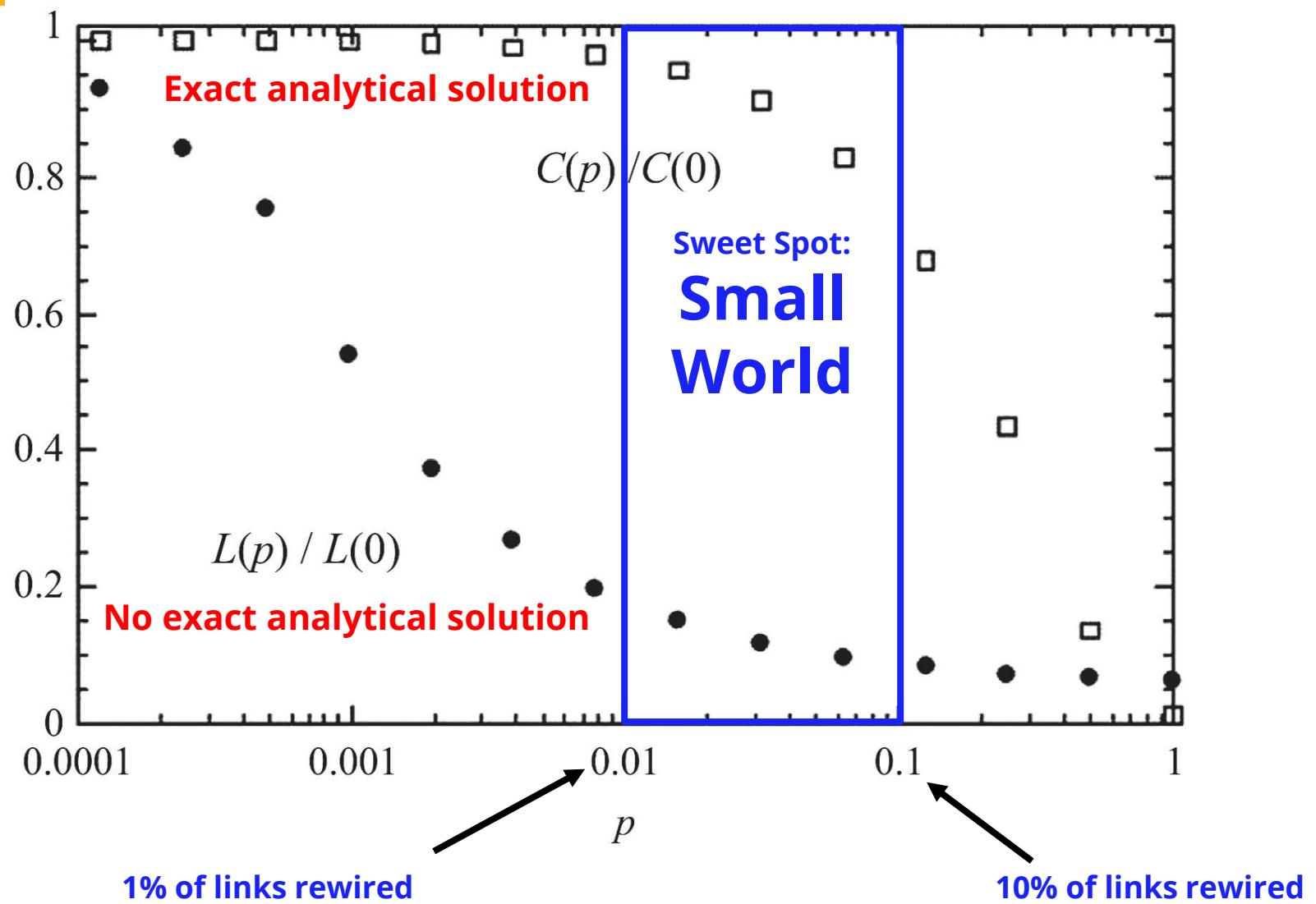
## Numerical simulation:

- We increase  $p$  (i.e.,  $\beta$ ) from 0 to 1
- Assume
  - $L(0)$  is the average path length of the regular lattice
  - $C(0)$  is the clustering coefficient of the regular lattice
  - For any  $p$ ,  $L(p)$  denotes the average path length of the small-world graph and  $C(p)$  denotes its clustering coefficient

## Observations:

- **Fast** decrease of average distance  $L(p)$
- **Slow** decrease in clustering coefficient  $C(p)$

# Change in Clustering Coefficient /Avg. Path Length



# Clustering Coefficient for Small-world model



- The probability that a connected triple stays connected after rewiring consists of
  1. The probability that none of the 3 edges were rewired is  $(1 - p)^3$
  2. The probability that other edges were rewired back to form a connected triple
    - Very small and can be ignored
- Clustering coefficient

$$C(p) \approx (1 - p)^3 C(0)$$

# Modeling with the Small-World Model



- Given a real-world network in which average degree is  $c$  and clustering coefficient  $C$  is given,
  - we set  $C(p) = C$  and determine  $\beta (= p)$  using equation

$$C(p) \approx (1 - p)^3 C(0)$$

- Given  $\beta$ ,  $c$ , and  $n$  (size of the real-world network), we can simulate the small-world model

# Real-World Network and Simulated Graphs



Network	Original Network				Simulated Graph	
	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

## Main assumption:

When a new user joins the network, the probability of connecting to existing nodes is proportional to existing nodes' degrees

For the new node  $v$

Connect  $v$  to a random node  $v_i$  with probability

$$P(v_i) = \frac{d_i}{\sum_j d_j}$$

Proposed by Albert-László Barabási and Réka Albert  
A special case of the Yule process

**Distribution of wealth in the society:**  
**The rich get richer**



Barabási, Albert-László, and Réka Albert. "Emergence of scaling in random networks." *science* 286.5439 (1999): 509-512.

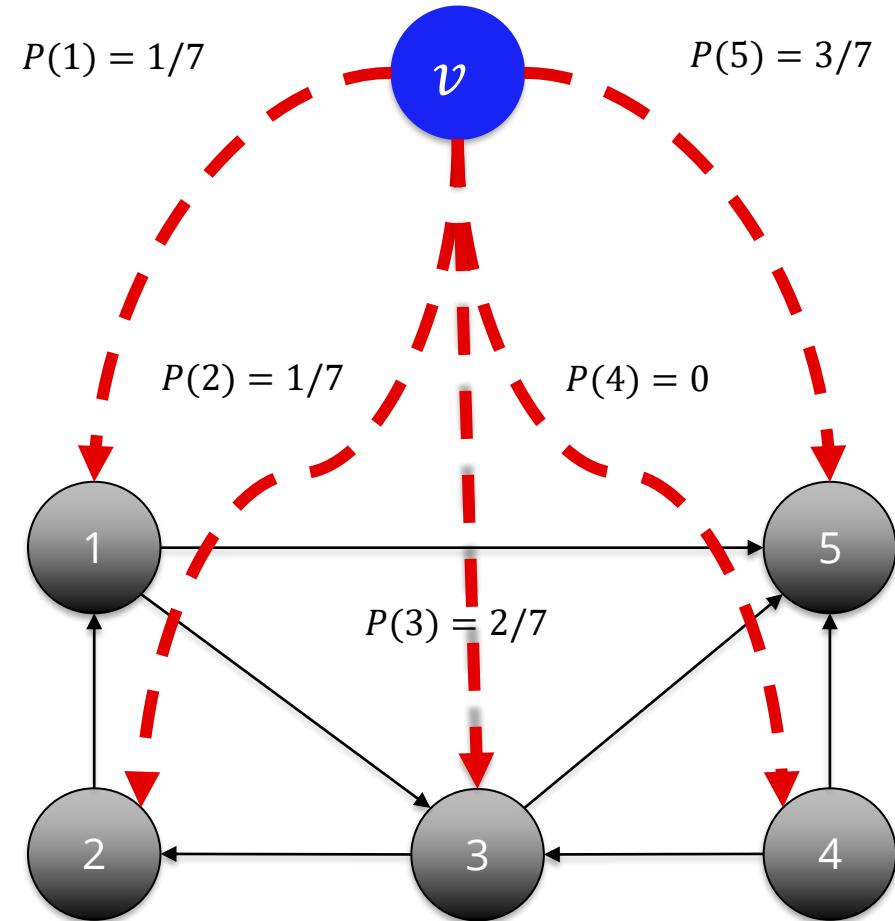
# Preferential Attachment: Example



Node  $v$  arrives

$$P(v_i) = \frac{d_i}{\sum_j d_j}$$

- $P(1) = 1/7$
- $P(2) = 1/7$
- $P(3) = 2/7$
- $P(4) = 0$
- $P(5) = 3/7$



---

## Algorithm 4.2 Preferential Attachment

**Require:** Graph  $G(V_0, E_0)$ , where  $|V_0| = m_0$  and  $d_v \geq 1 \forall v \in V_0$ , number of expected connections  $m \leq m_0$ , time to run the algorithm  $t$

```
1: return A scale-free network
2: //Initial graph with  $m_0$  nodes with degrees at least 1
3:  $G(V, E) = G(V_0, E_0);$ 
4: for 1 to  $t$  do
5:    $V = V \cup \{v_i\}$ ; // add new node  $v_i$ 
6:   while  $d_i \neq m$  do
7:     Connect  $v_i$  to a random node  $v_j \in V, i \neq j$  ( i.e.,  $E = E \cup \{e(v_i, v_j)\}$  )
       with probability  $P(v_j) = \frac{d_j}{\sum_k d_k}$ .
8:   end while
9: end for
10: Return  $G(V, E)$ 
```

---



# Properties of the Preferential Attachment Model

$$P(d) = \frac{2m^2}{d^3}$$

**Degree Distribution:**

**Clustering Coefficient:**

$$C = \frac{m_0 - 1}{8} \frac{(\ln t)^2}{t}$$

**Average Path Length:**

$$l \sim \frac{\ln |V|}{\ln(\ln |V|)}$$

# Modeling with the Preferential Attachment Model



Similar to random graphs, we can simulate real-world networks by generating a preferential attachment model by setting the expected degree  $m$

See Algorithm 4.2 in the book

# Real-World Networks and Simulated Graphs



Network	Original Network				Simulated Graph	
	Size	Average Degree	Average Path Length	C	Average Path Length	C
Film Actors	225,226	61	3.65	0.79	4.90	$\approx 0.005$
Medline Coauthorship	1,520,251	18.1	4.6	0.56	5.36	$\approx 0.0002$
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

# Unpredictability of the Rich-Get-Richer Effects

innovate

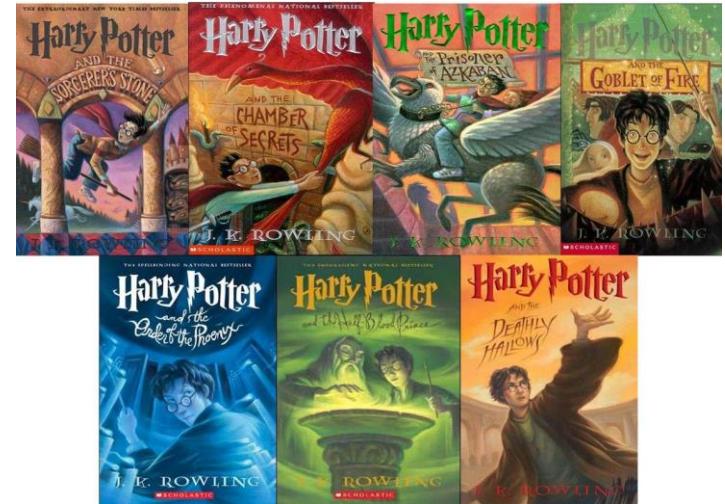
achieve

lead

The initial stages of one's rise to popularity are fragile

Once a user is well established, the rich-get-richer dynamics of popularity is likely to push the user even higher

**But** getting the rich-get-richer process started in the first place is full of potential accidents and near-misses



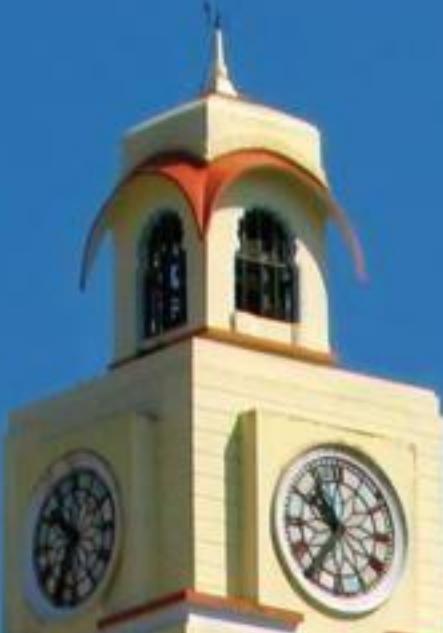
If we could roll time back to 1997, and then run history forward again, would the Harry Potter books again sell hundreds of millions of copies?

See more: Salganik, Matthew J., Peter Sheridan Dodds, and Duncan J. Watts. "Experimental study of inequality and unpredictability in an artificial cultural market." *science* 311.5762 (2006): 854-856.



---

# Thank you



# Social Media Analytics: Data Mining Essentials

Dr. Prasad Ramanathan

[p\\_ramanathan@wilp.bits-pilani.ac.in](mailto:p_ramanathan@wilp.bits-pilani.ac.in)

**BITS** Pilani

Pilani Campus



# Acknowledgment

Course material from the following source is gratefully acknowledged:

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/>**

# Introduction



Data production rate has increased dramatically (**Big Data**) and we are able store much more data

E.g., purchase data, social media data, cellphone data

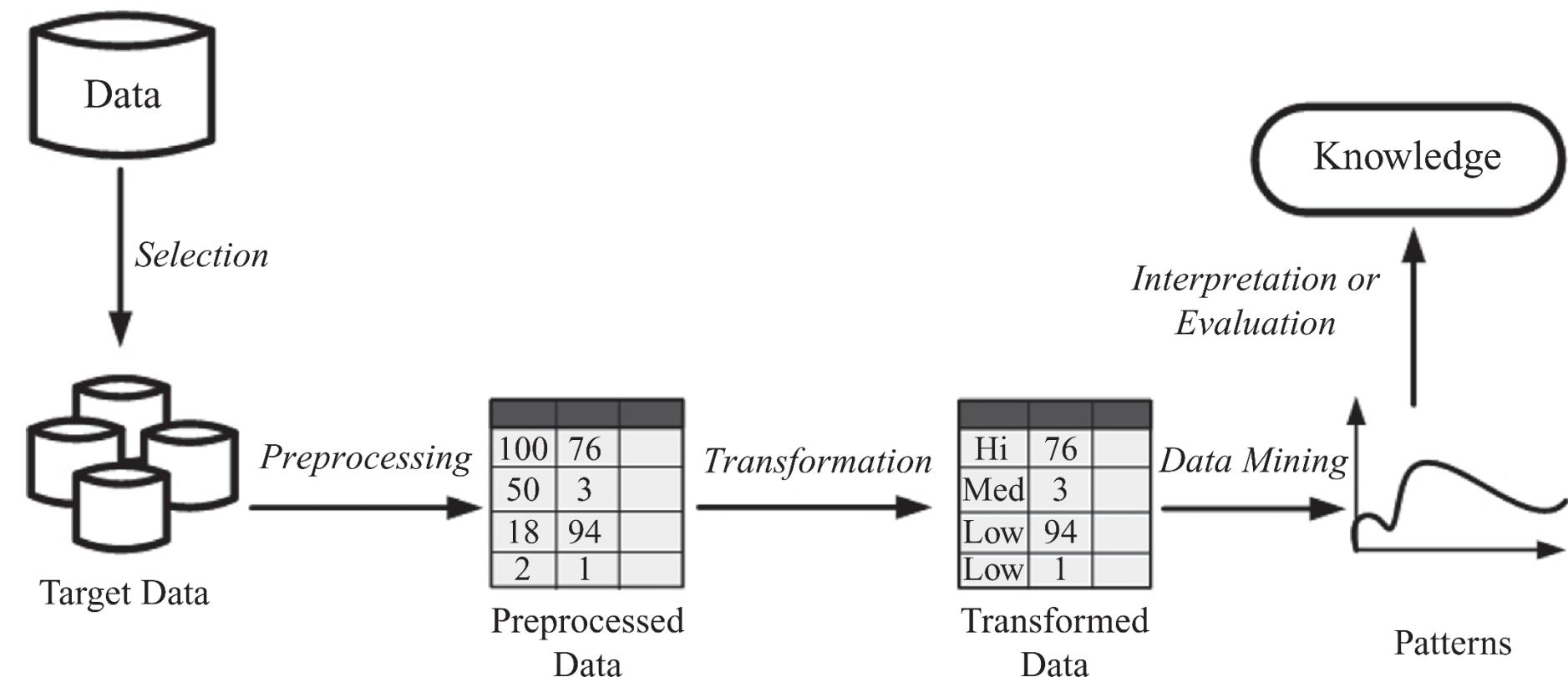
Businesses and customers need useful or actionable knowledge to gain insight from raw data for various purposes

It's not just searching data or databases



The process of extracting useful patterns from raw data is known as **Knowledge Discovery in Databases (KDD)**

# KDD Process



## Collect Raw Data

Use site provided APIs

Flickr's: <https://www.flickr.com/services/api/>

Scrape information directly

## Use Provided Repositories

<http://socialcomputing.asu.edu>

<http://snap.Stanford.edu>

<https://github.com/caesar0301/awesome-public-datasets>

**Data Mining:** the **process** of discovering **hidden** and **actionable** patterns from data

It utilizes methods at the intersection of artificial intelligence, machine learning, statistics, and database systems

Extracting / “mining” knowledge from large-scale data (big data)

Data-driven discovery and modeling of hidden patterns in big data

Extracting information/knowledge from data that is implicit, previously unknown, unexpected, and potentially useful

# Data Mining vs. Databases



**Data mining** is the *process* of extracting hidden and actionable patterns from data

**Database systems** store and manage data

Queries return part of stored data

Queries do not extract hidden patterns

Examples of querying databases

Find all employees with income more than \$250K

Find top spending customers in last month

Find all students from *engineering college* with GPA more than average

# Examples of Data Mining Applications



**Fraud/Spam Detections:** Identifying fraudulent transactions of a credit card or spam emails  
You are given a user's purchase history and a new transaction, identify whether the transaction is fraud or not;  
Determine whether a given email is spam or not

**Frequent Patterns:** Extracting purchase patterns from existing records  
beer  $\Rightarrow$  dippers (80%)

**Forecasting:** Forecasting future sales and needs according to some given samples

**Finding Like-Minded Individuals:** Extracting groups of like-minded people in a given network



# Data

# Data Instances



In the KDD process,  
Data is in a tabular format (a set of **instances**)

Each instance is a collection of properties and features related to an object or person

- A patient's medical record
- A user's profile
- A gene's information

Instances are also called *points*, *data points*, or *observations*

Data Instance:

Attributes					Class
Name	Money Spent	Bought Similar	Visits	Will Buy	Class Attribute
Mary	High	Yes	Rarely	Yes	Class Label
Features ( Attributes or measurements)					Class Label

Predicting whether an individual who visits an online book seller is going to buy a specific book

Attributes				Class
Name	Money Spent	Bought Similar	Visits	Will Buy
John	High	Yes	Frequently	?
Mary	High	Yes	Rarely	Yes

Unlabeled  
Example

Labeled  
Example

Features can be  
**Continuous**: values are numeric values  
Money spent: \$25  
**Discrete**: can take a number of values  
Money spent: {high, normal, low}

# Data Types + Permissible Operations (statistics)



## Nominal

### Operations:

Mode (most common feature value), Equality Comparison

E.g., {male, female}

## Ordinal

Feature values have an intrinsic order to them, but the difference is not defined

### Operations:

same as nominal, feature value rank

E.g., {Low, medium, high}

## Interval

### Operations:

Addition and subtractions are allowed whereas divisions and multiplications are not

E.g., 3:08 PM, calendar dates

## Ratio

### Operations:

divisions and multiplications are allowed

E.g., Height, weight, money quantities

# Sample Dataset - Twitter Users



<b>Activity</b>	<b>Date Joined</b>	<b>Number of Followers</b>	<b>Verified Account?</b>	<b>Has Profile Picture?</b>
High	2015	50	FALSE	no
High	2013	300	TRUE	no
Average	2011	860000	FALSE	yes
Low	2012	96	FALSE	yes
High	2008	8,000	FALSE	yes
Average	2009	5	TRUE	no
Very High	2010	650,000	TRUE	yes
Low	2010	95	FALSE	no
Average	2011	70	FALSE	yes
Very High	2013	80,000	FALSE	yes
Low	2014	70	TRUE	yes
Average	2013	900	TRUE	yes
High	2011	7500	FALSE	yes
Low	2010	910	TRUE	no

Ordinal

Interval

Ratio

Nominal

Nominal

The most common way to represent documents  
is to transform them into vectors  
Process them with linear algebraic operations

This representation is called “***Bag of Words***”  
Vector Space Model

Weights for words can be assigned by **TF-IDF**

Consider a set of documents  $D$   
Each document is a set of words

**Goal:** convert these documents to vectors

$$d_i = (w_{1,i}, w_{2,i}, \dots, w_{N,i})$$

- $d_i$  : document  $i$
- $w_{j,i}$  : the weight for word  $j$  in document  $i$

## How to set $w_{j,i}$

- Set  $w_{j,i}$  to 1 when the word  $j$  exists in document  $i$  and 0 when it does not.
- We can also set  $w_{j,i}$  to the number of times the word  $j$  is observed in document  $i$  (**frequency**)

# Vector Space Model: An Example



## Documents:

$d_1$ : social media mining

$d_2$ : social media data

$d_3$ : financial market data

## Reference vector (**Dictionary**):

(social, media, mining, data, financial, market)

## Vector representation:

	social	media	mining	data	financial	market
$d_1$	1	1	1	0	0	0
$d_2$	1	1	0	1	0	0
$d_3$	0	0	0	1	1	1

# TF-IDF (Term Frequency-Inverse Document Frequency)



TF-IDF of term (word)  $t$ , document  $d$ , and document corpus  $D$  is calculated as follows:

$$w_{j,i} = tf_{j,i} \times idf_j$$

$tf_{j,i}$  is the frequency of word  $j$  in document  $i$

The total number of documents in the corpus

$$idf_j = \log_2 \frac{|D|}{|\{\text{document} \in D \mid j \in \text{document}\}|}$$

↑  
The number of documents where the term  $j$  appears

# TF-IDF: An Example



Document  $d_1$  contains 100 words

Word “apple” appears 10 times in  $d_1$

Word “orange” appears 20 times in  $d_1$

We have  $|D| = 20$  documents

Word “apple” only appears in document  $d_1$

Word “orange” appears in all 20 documents

$$tf - idf(\text{“apple”}, d_1) = 10 \times \log_2 \frac{20}{1} = 43.22$$

$$tf - idf(\text{“orange”}, d_1) = 20 \times \log_2 \frac{20}{20} = 0$$

# TF-IDF: An Example



Documents:

$d_1$ : social media mining

$d_2$ : social media data

$d_3$ : financial market data

TF values:

$$\begin{aligned} idf_{social} &= \log_2(3/2) = 0.584 \\ idf_{media} &= \log_2(3/2) = 0.584 \\ idf_{mining} &= \log_2(3/1) = 1.584 \\ idf_{data} &= \log_2(3/2) = 0.584 \\ idf_{financial} &= \log_2(3/1) = 1.584 \\ idf_{market} &= \log_2(3/1) = 1.584 \end{aligned}$$

	social	media	mining	data	financial	market
$d_1$	1	1	1	0	0	0
$d_2$	1	1	0	1	0	0
$d_3$	0	0	0	1	1	1

TF-IDF

	social	media	mining	data	financial	market
$d_1$	0.584	0.584	1.584	0	0	0
$d_2$	0.584	0.584	0	0.584	0	0
$d_3$	0	0	0	0.584	1.584	1.584

When making data ready for mining, data quality needs to be assured

## Noise

Noise is the distortion of the data

## Outliers

Outliers are data points that are considerably different from other data points in the dataset

## Missing Values

Missing feature values in data instances

### Solution:

Remove instances that have missing values

Estimate missing values, and

Ignore missing values when running data mining algorithm

## Duplicate data

## Aggregation

It is performed when multiple features need to be combined into a single one or when the scale of the features change

Example: image width , image height -> image area (width x height)

## Discretization

From continues values to discrete values

Example: money spent -> {low, normal, high}

## Feature Selection

Choose relevant features

## Feature Extraction

Creating new features from original features

Often, more complicated than aggregation

## Sampling

Random Sampling

Sampling with or without replacement

Stratified Sampling: useful when having class imbalance

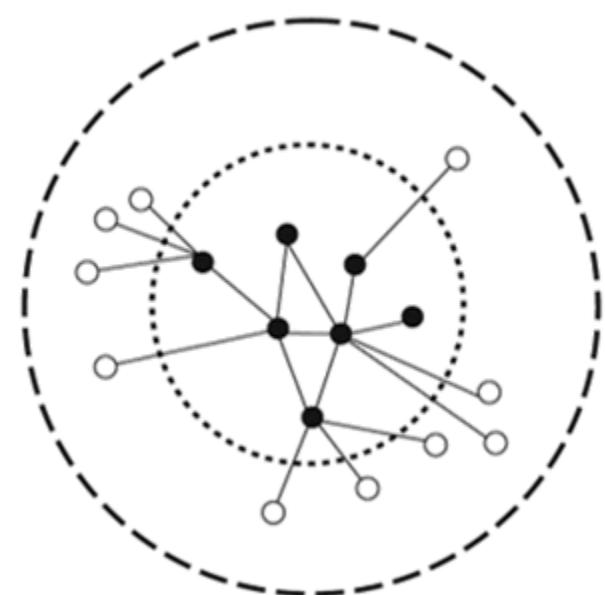
Social Network Sampling

## Sampling social networks:

Start with a small set of nodes  
(seed nodes)

Sample

- (a)** the connected components they belong to;
- (b)** the set of nodes (and edges) connected to them directly; or
- (c)** the set of nodes and edges that are within n-hop distance from them.





# 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

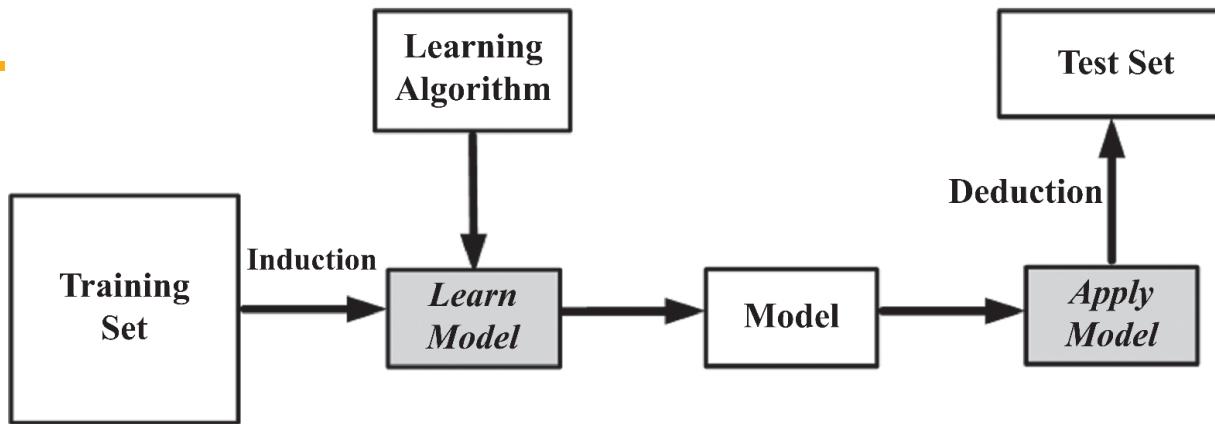
Group similar items together into some clusters

Detect communities in a given social network



# Supervised Learning

# Supervised Learning: The Process



We are given a set of labeled records/instances

In the format  $(X, y)$

$X$  is a vector of features

$y$  is the class attribute (commonly a scalar)

**[Training]** The supervised learning task is to build model that maps  $X$  to  $y$  (find a mapping  $m$  such that  $m(X) = y$ )

**[Testing]** Given an unlabeled instance  $(X', ?)$ , we compute  $m(X')$   
E.g., spam/non-spam prediction

# Classification: An Email Example



A set of emails is given  
Users have manually labeled  
them as **spam / non-spam**

Use a set of features ( $x$ ) to  
identify spam/non-spam  
status of the email ( $y$ )  
We can use words in the email

In this case, classes are  
 $y = \{spam, non-spam\}$

Subject	Mailbox
{Definitely Spam?} Online. The easiest way to get your doctoral	spam_...
{Spam?} Quick loans	Junk
{Definitely Spam?} Train to become a photographer. Find reputab	Junk
{Definitely Spam?} Custom website designing	spam_...
Spam: {Definitely Spam?} Looking to become a nurse?	spam_...
{Spam?} Start your fairy-tale in Orlando.	spam_...
Spam: {Definitely Spam?} Online doctorate programs in your area	spam_...
{Definitely Spam?} Simple, Secure, Mobile. Email Fax	spam_...
Spam: {Definitely Spam?} Online Education Is Easier Than You Th	Junk
{Definitely Spam?} Train to become a photographer. Find reputab	Junk
Spam: {Definitely Spam?} Design Degrees	spam_...
{Spam?} Local maids want to help you	spam_...
{Spam?} Compare beads and save	spam_...
{Spam?} Talking for free has never been so easy	spam_...
{Spam?} Browse our selection of camera phones	spam_...
{Definitely Spam?} Owning a franchise is easy	spam_...
Definitely Spam? Convenient Online Options to Earn Your Degree	spam_...
Spam: {Definitely Spam?} We are offering free cell phones	spam_...
{Definitely Spam?} Need a replacement carburetor?	Junk
{Definitely Spam?} It's good to have a lawyer on your side	Junk
{Spam?} {Disarmed} Interior Design Schools	spam_...
{Definitely Spam?} Borrow the money you need with a personal lo	spam_...
{Definitely Spam?} Join one of the fastest growing professions.	spam_...
{Definitely Spam?} It's not too late to earn your degree	spam_...
{Definitely Spam?} Subsidize your mortgage with a VA loan	spam_...
Spam: {Definitely Spam?} No buster? Housekeeping services insid	spam_...
Definitely Spam? It's a good thing to check your credit report	spam_...
Spam: {Definitely Spam?} An offer for software which manages yo	spam_...
{Definitely Spam?} Affordable lawn care	spam_...
{Definitely Spam?} {Disarmed} Driveways, Patios, and Walks. Fin	spam_...
{Definitely Spam?} Cleaning is easier with hardwood floors	spam_...
{Definitely Spam?} Get awesome deals on hammocks.	spam_...
{Definitely Spam?} Affordable beach vacations	spam_...
Spam: {Definitely Spam?} Let your creativity soar	spam_...
{Definitely Spam?} Join the hospital as a medical biller	spam_...
Definitely Spam? Blow them away with gorgeous diamond jewelry.	spam_...
{Definitely Spam?} Your career in nursing	spam_...
{Definitely Spam?} This is the moment she has been dreaming abo	spam_...
{Definitely Spam?} Medical transcribers work at home	spam_...
{Definitely Spam?} {Disarmed} All types of fence styles availab	spam_...

# A Twitter Example



ID	Celebrity	Verified Account	# Followers	Influential?
1	Yes	No	1.25M	No
2	No	Yes	1M	No
3	No	Yes	600K	No
4	Yes	Unknown	2.2M	No
5	No	No	850K	Yes
6	No	Yes	750K	No
7	No	No	900K	Yes
8	No	No	700K	No
9	Yes	Yes	1.2M	No
10	No	Unknown	950K	Yes

## Classification

Decision tree learning

Naive Bayes Classifier

$k$ -nearest neighbor classifier

Classification with Network information

## Regression

Linear Regression

Logistic Regression



# Decision Tree Learning

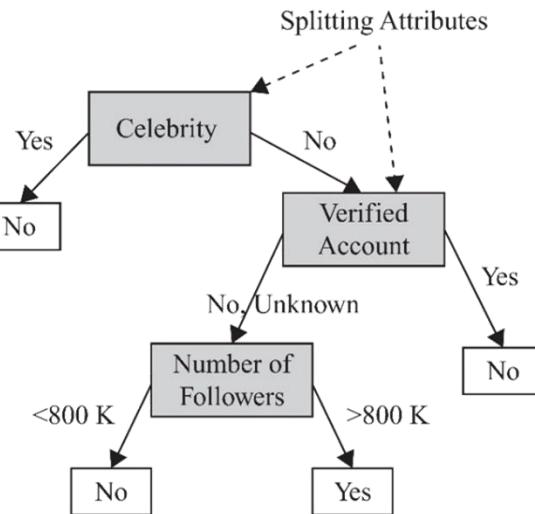
# Decision Tree



A decision tree is learned from the dataset  
(training data with known classes)

ID	Celebrity	Verified Account	# Followers	Influential?
1	Yes	No	1.25M	No
2	No	Yes	1M	No
3	No	Yes	600K	No
4	Yes	Unknown	2.2M	No
5	No	No	850K	Yes
6	No	Yes	750K	No
7	No	No	900K	Yes
8	No	No	700K	No
9	Yes	Yes	1.2M	No
10	No	Unknown	950K	Yes

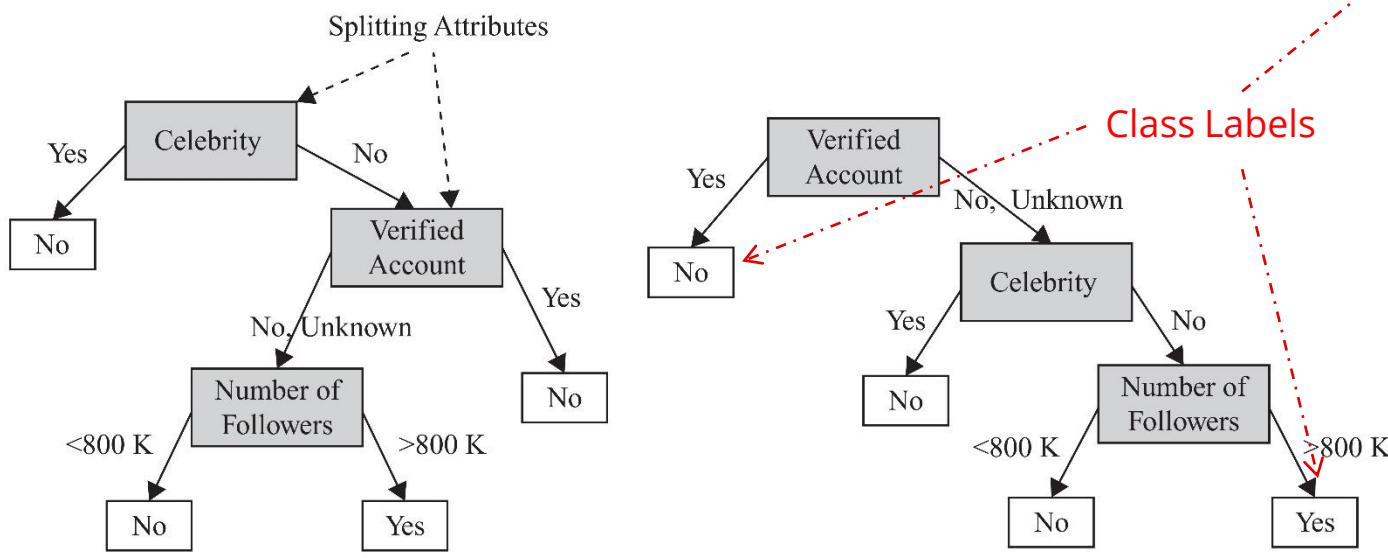
The learned tree is later applied to predict the class attribute value of new data  
(test data with unknown classes)  
Only the feature values are known



# Decision Tree: Example

Multiple decision trees can be learned from the same dataset

ID	Celebrity	Verified Account	# Followers	Influential?
1	Yes	No	1.25M	No
2	No	Yes	1M	No
3	No	Yes	600K	No
4	Yes	Unknown	2.2M	No
5	No	No	850K	Yes
6	No	Yes	750K	No
7	No	No	900K	Yes
8	No	No	700K	No
9	Yes	Yes	1.2M	No
10	No	Unknown	950K	Yes



(a) Learned Decision Tree 1

(b) Learned Decision Tree 2

# Decision Tree Construction



Decision trees are constructed recursively

A top-down greedy approach in which features are sequentially selected.

After selecting a feature for each node,  
based on its attribute values, different branches are created.

The training set is then partitioned into subsets based on the  
feature values,  
each of which fall under the respective feature value branch;  
The process is continued for these subsets and other nodes

When selecting features, we prefer features that partition the  
set of instances into subsets that are more **pure**.

A pure subset has instances that all have the same class  
attribute value.

# Decision Tree Construction



When reaching pure (or highly pure) subsets under a branch,  
the decision tree construction process no longer partitions the subset,  
creates a leaf under the branch, and  
assigns the class attribute value (or the majority class attribute value) for subset instances as the leaf's predicted class attribute value

To measure purity we can use/minimize entropy.

Over a subset of training instances,  $T$ , with a binary class attribute (values in  $\{+, -\}$ ), the entropy of  $T$  is defined as:

$$\text{entropy}(T) = -p_+ \log p_+ - p_- \log p_-$$

$p_+$  is the proportion of positive examples in  $T$   
 $p_-$  is the proportion of negative examples in  $T$

# Entropy Example



Assume there is a subset  $T$  that has 10 instances:

- Seven instances have a **positive** class attribute value
- Three have a **negative** class attribute value
- Denote  $T$  as [7+, 3-]
- The entropy for subset  $T$  is

$$\text{entropy}(T) = -\frac{7}{10} \log \frac{7}{10} - \frac{3}{10} \log \frac{3}{10} = 0.881$$

In a pure subset, all instances have the same class attribute value (**entropy is 0**)

If the subset contains an unequal number of positive and negative instances

- The entropy is between 0 and 1.



# Naïve Bayes Learning

# Naive Bayes Classifier



For two random variables  $X$  and  $Y$ , Bayes theorem states that

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

↑  
class variable      ↙  
the instance features

Then class attribute value for instance  $X$

$$\arg \max_{y_i} P(y_i|X)$$

We assume that features are independent given the class attribute

$$P(y_i|X) = \frac{P(X|y_i)P(y_i)}{P(X)}$$

↓

$$P(X|y_i) = \prod_{j=1}^n P(x_j|y_i) \rightarrow P(y_i|X) = \frac{(\prod_{j=1}^n P(x_j|y_i))P(y_i)}{P(X)}$$

# NBC: An Example



No.	Outlook (O)	Temperature (T)	Humidity (H)	Play Golf (PG)
1	sunny	hot	high	N
2	sunny	mild	high	N
3	overcast	hot	high	Y
4	rain	mild	high	Y
5	sunny	cool	normal	Y
6	rain	cool	normal	N
7	overcast	cool	normal	Y
8	sunny	mild	high	?

$$\begin{aligned}
 P(PG = Y|i_8) &= \frac{P(i_8|PG = Y)P(PG = Y)}{P(i_8)} \\
 &= P(O = \text{Sunny}, T = \text{mild}, H = \text{high}|PG = Y) \\
 &\quad \times \frac{P(PG = Y)}{P(i_8)} \\
 &= P(O = \text{Sunny}|PG = Y) \times P(T = \text{mild}|PG = Y) \\
 &\quad \times P(H = \text{high}|PG = Y) \times \frac{P(PG = Y)}{P(i_8)} \\
 &= \frac{1}{4} \times \frac{1}{4} \times \frac{2}{4} \times \frac{\frac{4}{7}}{P(i_8)} = \frac{1}{56P(i_8)}.
 \end{aligned}$$

$$\begin{aligned}
 P(PG = N|i_8) &= \frac{P(i_8|PG = N)P(PG = N)}{P(i_8)} \\
 &= P(O = \text{Sunny}, T = \text{mild}, H = \text{high}|PG = N) \\
 &\quad \times \frac{P(PG = N)}{P(i_8)} \\
 &= P(O = \text{Sunny}|PG = N) \times P(T = \text{mild}|PG = N) \\
 &\quad \times P(H = \text{high}|PG = N) \times \frac{P(PG = N)}{P(i_8)} \\
 &= \frac{2}{3} \times \frac{1}{3} \times \frac{2}{3} \times \frac{\frac{3}{7}}{P(i_8)} = \frac{4}{63P(i_8)}.
 \end{aligned}$$

$$\frac{1}{56P(i_8)} < \frac{4}{63P(i_8)} \rightarrow \text{Play Golf} = N$$



# Nearest Neighbor Classifier



# Nearest Neighbor Classifier



*k*-nearest neighbor or *k*-NN,

Utilizes the neighbors of an instance to perform classification.

It uses the *k* nearest instances, called **neighbors**, to perform classification.

The instance being classified is assigned the label (class attribute value) that the majority of its *k* neighbors are assigned

When *k* = 1, the closest neighbor's label is used as the predicted label for the instance being classified

To determine the neighbors of an instance, we need to measure its distance to all other instances based on some distance metric.

Often Euclidean distance is employed

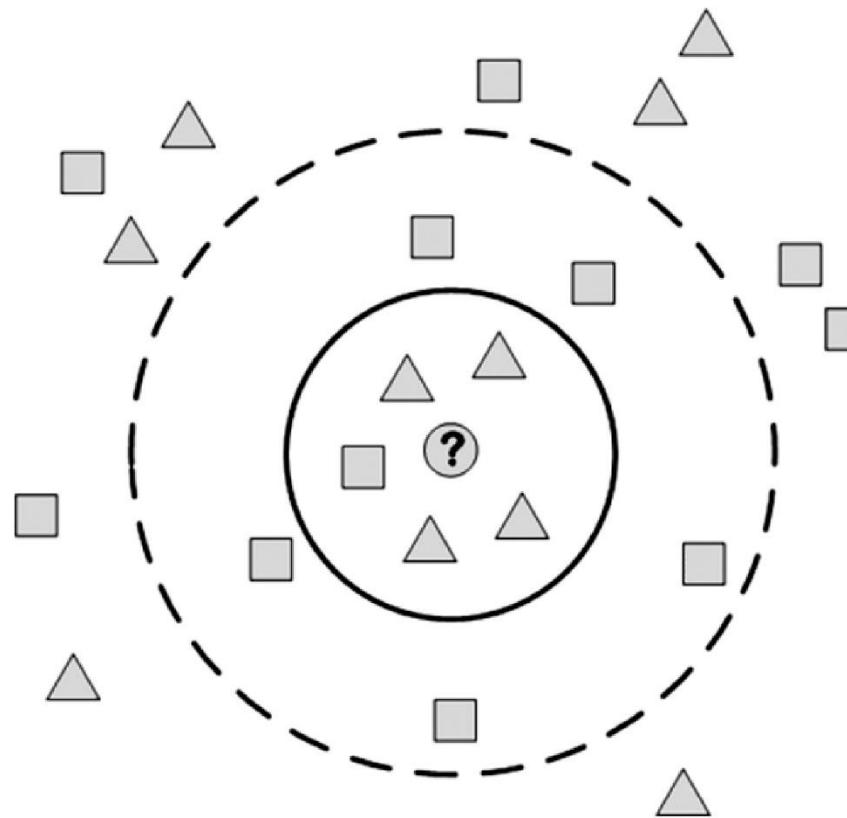
---

## Algorithm 5.1 *k*-Nearest Neighbor Classifier

**Require:** Instance  $i$ , A Dataset of Real-Value Attributes,  $k$  (number of neighbors), distance measure  $d$

- 1: **return** Class label for instance  $i$
  - 2: Compute  $k$  nearest neighbors of instance  $i$  based on distance measure  $d$ .
  - 3:  $l =$  the majority class label among neighbors of instance  $i$ . If more than one majority label, select one randomly.
  - 4: Classify instance  $i$  as class  $l$
-

# *k*-NN example



When  $k = 5$ , the predicted label is:  $\Delta$

When  $k = 9$ , the predicted label is:  $\square$

# *k*-NN: Example 2



No.	Outlook (O)	Temperature (T)	Humidity (H)	Play Golf (PG)
1	sunny	hot	high	N
2	sunny	mild	high	N
3	overcast	hot	high	Y
4	rain	mild	high	Y
5	sunny	cool	normal	Y
6	rain	cool	normal	N
7	overcast	cool	normal	Y
8	sunny	mild	high	?

**Similarity between row 8 and other data instances;**

(Similarity = 1 if attributes have the same value, otherwise similarity = 0)

Data instance	Outlook	Temperature	Humidity	Similarity	Label	K	Prediction
2	1	1	1	3	N	1	N
1	1	0	1	2	N	2	N
4	0	1	1	2	Y	3	N
3	0	0	1	1	Y	4	?
5	1	0	0	1	Y	5	Y
6	0	0	0	0	N	6	?
7	0	0	0	0	Y	7	Y



# Classification with Network Information

# Classification with Network Information



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

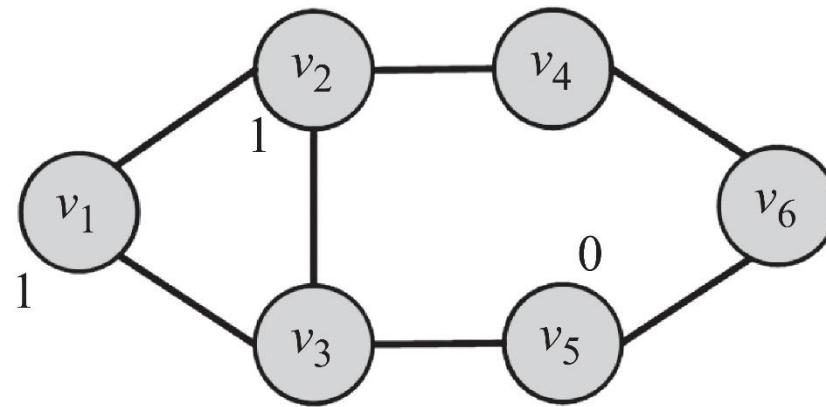
The product seller wants to know who the potential buyers are for this product.

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  $P(y_i = 1 | N(v_i))$ ?

# Weighted-vote Relational-Neighbor (wvRN)



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

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

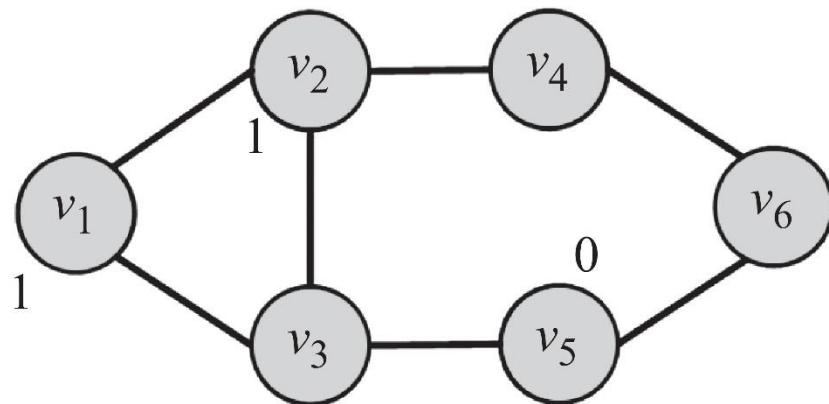
$$P(y_i = 1|N(v_i)) = \frac{1}{|N(v_i)|} \sum_{v_j \in N(v_i)} P(y_j = 1|N(v_j))$$

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



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

$$P(y_3 | N(v_3))$$

$$= \frac{1}{|N(v_3)|} \sum_{v_j \in N(v_3)} P(y_j = 1 | N(v_j))$$

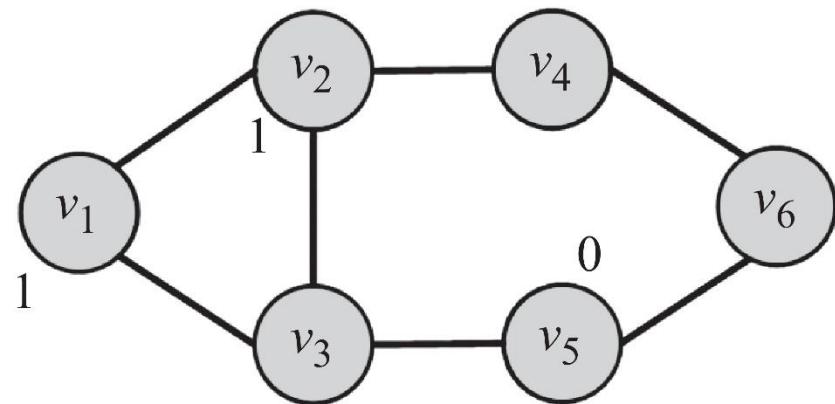
$$= \frac{1}{3}(P(y_1 = 1 | N(v_1)) + P(y_2 = 1 | N(v_2)) + P(y_5 = 1 | N(v_5)))$$

$$= \frac{1}{3}(1 + 1 + 0) = 0.67$$

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

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

# wvRN example



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

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

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

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

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

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

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

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



# Regression

# Regression



In regression,  
Class values are real  
numbers as class values  
In classification, class values  
are categorical

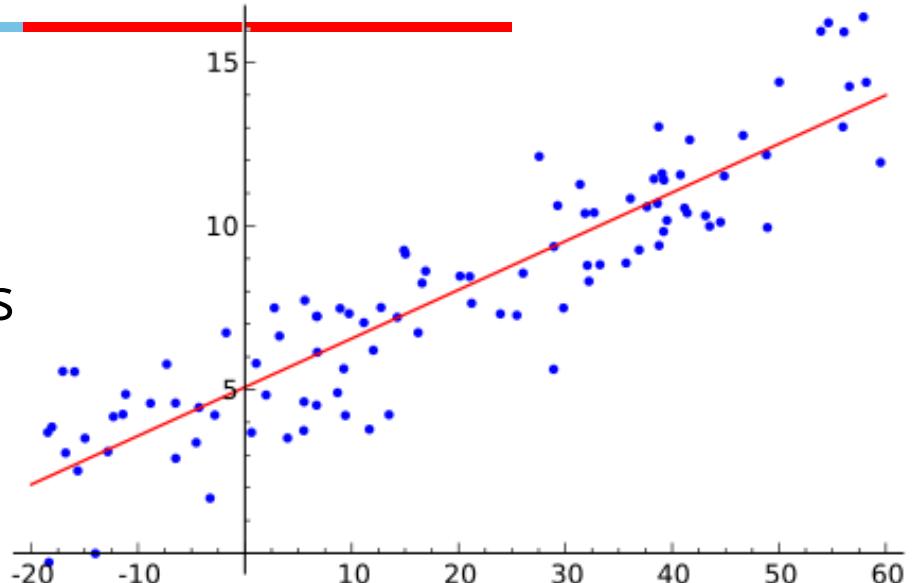
$$y \approx f(X)$$

**Class attribute**  
(dependent variable)

$$y \in R$$

**Features**  
(regressors)

$$X = (x_1, x_2, \dots, x_m)$$



**Goal:** find the relation between  $y$   
and vector  $X = (x_1, x_2, \dots, x_m)$

**Linear regression:** we assume the relation between the class attribute  $Y$  and feature set  $X$  is linear

$$Y = XW + \epsilon$$

$W$  represents the vector of regression coefficients

Regression can be solved by estimating  $W$  and  $\epsilon$  using the provided dataset and the labels  $Y$

**“Least squares”** is a popular method to solve regression

The goal is to minimize  $\epsilon^2 = ||\epsilon^2|| = ||Y - XW||^2$

# Least Squares



Find  $W$  such that minimizing  $\|Y - XW\|^2$  for regressors  $X$  and labels  $Y$

$$\|X\|^2 = X^T X \quad \min \|Y - XW\|^2$$

$$\frac{\partial}{\partial W} \|Y - XW\|^2 = 0$$

$$\|X\|^2 = X^T X \Rightarrow \frac{\partial}{\partial W} (Y - XW)^T (Y - XW) = 0$$

$$\frac{\partial}{\partial W} (Y^T - W^T X^T)(Y - XW) = 0$$

$$\frac{\partial}{\partial W} (Y^T Y - Y^T XW - W^T X^T Y + W^T X^T XW) = 0$$

$$-2X^T Y + 2X^T XW = 0$$

$$2X^T Y = 2X^T XW$$

$$W = (X^T X)^{-1} X^T Y$$

# Simplifying W with SVD



$$X = U\Sigma V^T$$

**SVD of X**

$$\begin{aligned}W &= (X^T X)^{-1} X^T Y \\&= (V \Sigma U^T U \Sigma V^T)^{-1} V \Sigma U^T Y \\&= (V \Sigma^2 V^T)^{-1} V \Sigma U^T Y \\&= V \Sigma^{-2} V^T V \Sigma U^T Y \\&= V \Sigma^{-1} U^T Y\end{aligned}$$

## A probabilistic view of regression

Assuming the class attribute is binary, logistic regression finds the probability  $p$

$$P(Y = 1|X) = p \quad X: \text{Feature vector}$$

We can assume that  $p$  can be obtained from  $X$

$$p = \beta X$$

**Unbounded:**  $X$  can take any values and  $\beta$  is also unconstrained

**Solution:** transform  $p$  using  $g(p)$ , such that  $g(p)$  is unbounded

Fit  $g(p)$  using  $\beta X$

$$g(p) = \ln \frac{p}{1-p}$$

Known as the **logit** function

For any  $p$  between  $[0,1]$

$G(p)$  is in range  $[-\infty, +\infty]$

$$\beta X = \ln \frac{p}{1-p} \rightarrow e^{\beta X} = \frac{p}{1-p} \rightarrow p = \frac{e^{\beta X}}{e^{\beta X} + 1}$$

$$p = \frac{1}{e^{-\beta X} + 1}$$

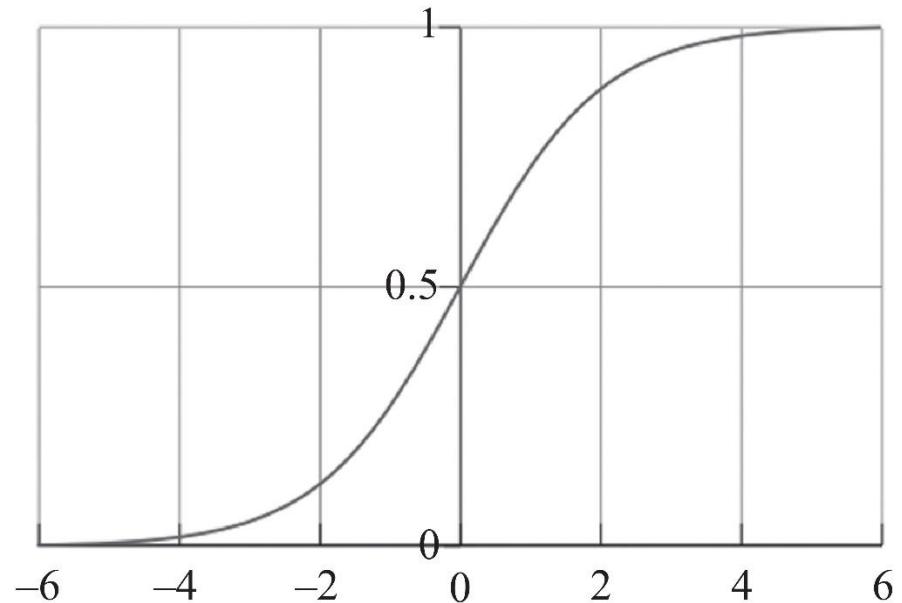
# Logistic Regression



$$p = \frac{1}{e^{-\beta X} + 1}$$

## Logistic Function

Acts as a probability



**Goal:** Find  $\beta$  such that  $P(Y|X)$  is maximized

No closed form solution

Iterative maximum likelihood

Prediction: once  $\beta$  is determined compute  $P(Y|X)$

For a binary class problem if it is more than 0.5, predict 1



# Supervised Learning Evaluation

## Training/**Testing** Framework:

A **training dataset** (i.e., the labels are known) is used to train a model  
the model is evaluated on a **test dataset**.

The correct labels of the test dataset are unknown,

In practice, the training set is divided into two parts,

One used for training and

The other used for testing.

When testing, the labels from this test set are removed.

After these labels are predicted using the model, the predicted labels are compared with the masked labels (**ground truth**).

## Dividing the training set into train/test sets

### **Leave-one-out training**

Divide the training set into  $k$  equally sized partitions

Often called **folds**

Use all folds but one to train and the one left out for testing

### **$k$ -fold cross validation training**

Divide the training set into  $k$  equally sized sets

Run the algorithm  $k$  times

In round  $i$ , we use all folds but fold  $i$  for training and fold  $i$  for testing.

The average performance of the algorithm over  $k$  rounds measures the performance of the algorithm.

# Evaluating Supervised Learning



- As the class labels are discrete, we can measure the accuracy by dividing number of correctly predicted labels ( $C$ ) by the total number of instances ( $N$ )

$$\text{accuracy} = \frac{C}{N}$$

$$\text{error rate} = 1 - \text{accuracy}$$

- More sophisticated approaches of evaluation  
AUC  
F-Measure

---

The labels cannot be predicted precisely

We need to set a margin to accept or reject the predictions

**Example.** When the observed temperature is 71, any prediction in the range of  $71 \pm 0.5$  can be considered as a correct prediction

Or, we can use correlation between predicted labels and the ground truth.

---



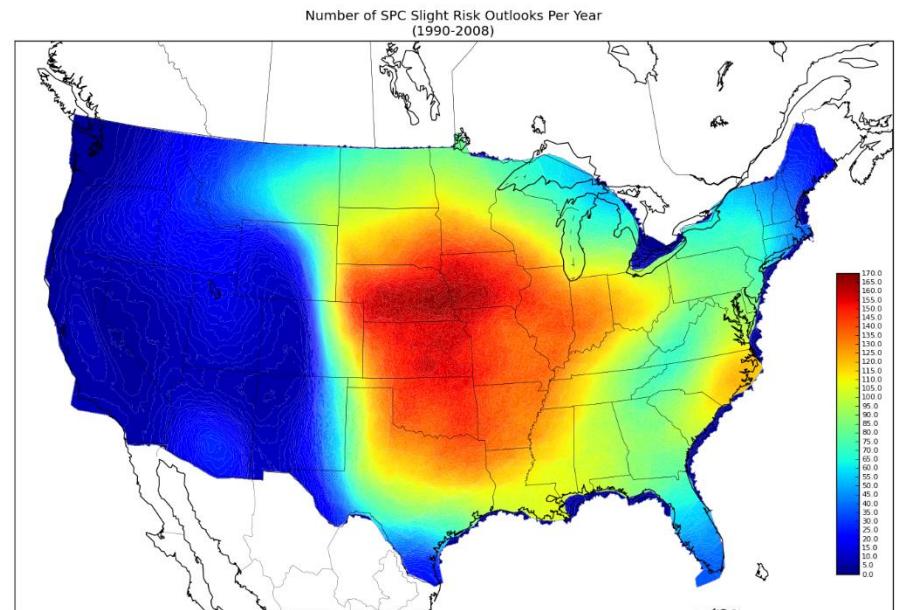
# Unsupervised Learning

## Unsupervised division of instances into groups of similar objects

Clustering is a form of  
**unsupervised learning**

Clustering algorithms  
group together  
**similar items**

The algorithm does not have  
examples showing how the  
samples should be grouped  
together (unlabeled data)



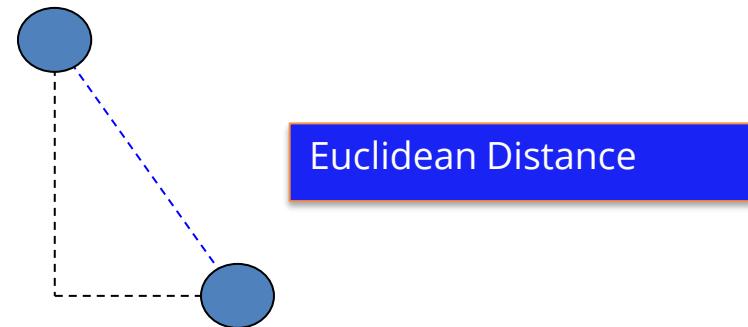
## Kernel Density Estimation

## Clustering Goal: Group together similar items

Instances are put into different clusters based on the distance to other instances

**Any clustering algorithm requires a distance measure**

The most popular (dis)similarity measure for continuous features are ***Euclidean Distance*** and ***Pearson Linear Correlation***



$$d(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

# Similarity Measures



$X$  and  $Y$  are  $n$ -dimensional vectors

$$X = (x_1, x_2, \dots, x_n)$$

$$Y = (y_1, y_2, \dots, y_n)$$

Measure Name	Formula	Description
Mahalanobis	$d(X, Y) = \sqrt{(X - Y)^T \Sigma^{-1} (X - Y)}$	$X, Y$ are features vectors and $\Sigma$ is the covariance matrix of the dataset
Manhattan ( $L_1$ norm)	$d(X, Y) = \sum_i  x_i - y_i $	$X, Y$ are features vectors
$L_p$ -norm	$d(X, Y) = (\sum_i  x_i - y_i ^n)^{\frac{1}{n}}$	$X, Y$ are features vectors

Once a distance measure is selected, instances are grouped using it.

Clusters are usually represented by compact and abstract notations.

“Cluster centroids” are one common example of this abstract notation.

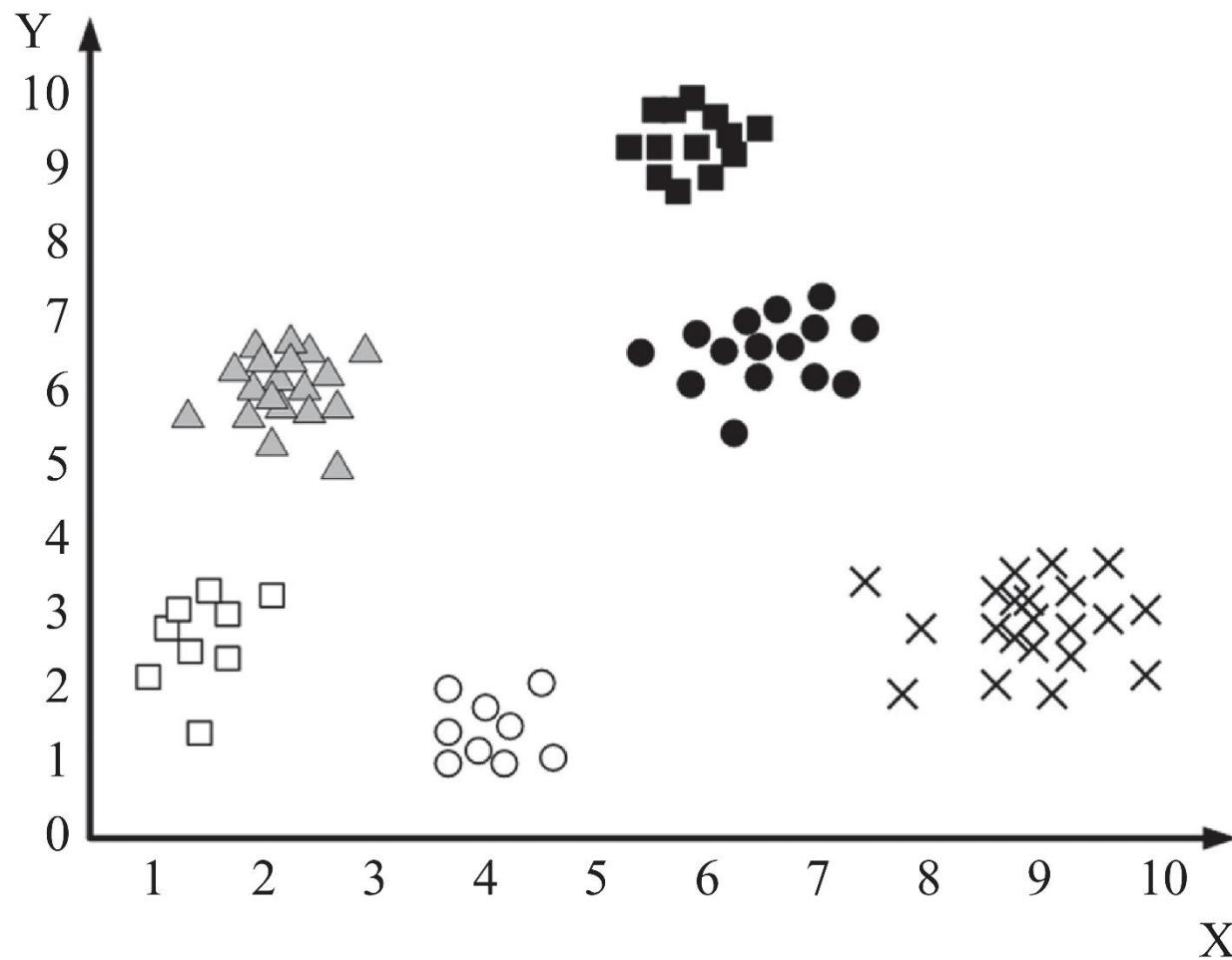
Partitional Algorithms (most common type)

Partition the dataset into a set of clusters

Each instance is assigned to a cluster exactly once  
No instance remains unassigned to clusters.

**Example:**  $k$ -means

# $k$ -means for $k = 6$



---

The most commonly used clustering algorithm  
Related to Expectation Maximization (**EM**) in statistics.

---

### **Algorithm 5.2** *k*-Means Algorithm

---

**Require:** A Dataset of Real-Value Attributes,  $k$  (number of Clusters)

- 1: **return** A Clustering of Data into  $k$  Clusters
  - 2: Consider  $k$  random instances in the data space as the initial cluster centroids.
  - 3: **while** centroids have not converged **do**
  - 4:     Assign each instance to the cluster that has the closest cluster centroid.
  - 5:     If all instances have been assigned then recalculate the cluster centroids by averaging instances inside each cluster
  - 6: **end while**
-

# *k*-means: Algorithm



Given data points  $x_i$  and an initial set of  $k$  centroids  $m_1^1, m_2^1, \dots, m_k^1$  the algorithm proceeds as follows:

**Assignment step:** Assign each data point to the cluster  $S_i^t$  with the closest centroid  
Each data point goes into exactly one cluster

$$S_i^t = \{x_p : \|x_p - m_i^t\| \leq \|x_p - m_j^t\| \forall 1 \leq j \leq k\}$$

**Update step:** Calculate the new means to be the centroid of the data points in the cluster  
After all points are assigned

The procedure is repeated until **convergence**

## Convergence:

Whether centroids are no longer changing

Equivalent to clustering assignments not changing

The algorithm can be stopped when the Euclidean distance between the centroids in two consecutive steps is less than some small positive value

As an alternative, *k*-means implementations try to minimize an **objective function**.

**Example:** the squared distance error:

$$\sum_{i=1}^k \sum_{j=1}^{n(i)} \|x_j^i - c_i\|^2$$

$x_j^i$  is the  $j$ th instance of cluster  $i$

$n(i)$  is the number of instances in cluster  $i$

$c_i$  is the centroid of cluster  $i$ .

## **Stopping Criterion:**

when the difference between the objective function values of two consecutive iterations of the *k*-means algorithm is less than some small value .

---

Finding the global optimum of the  $k$  partitions is computationally expensive (**NP-hard**).

This is equivalent to finding the optimal centroids that minimize the objective function

**Solution:** efficient heuristics

**Outcome:** converge quickly to a local optimum that might not be global

**Example:** running  $k$ -means multiple times

Select the clustering assignment that is observed most often  
or

Select the clustering that is more desirable based on an objective function, such as the squared error.

---



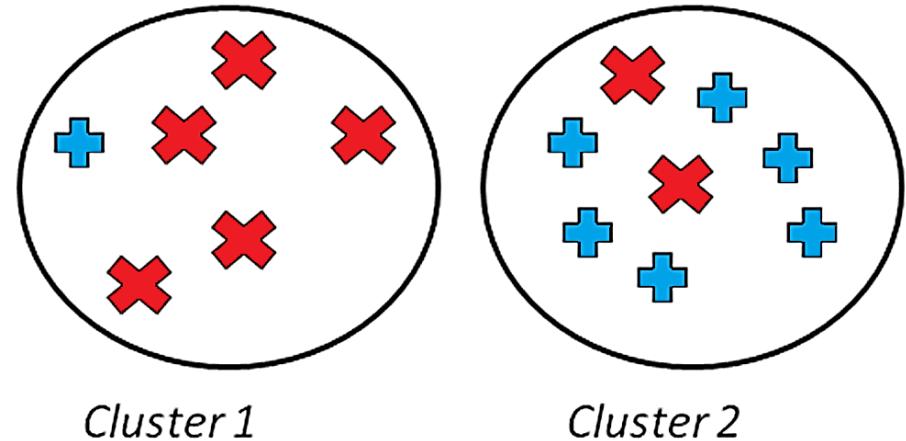
# Unsupervised Learning Evaluation

# Evaluating the Clusterings



We are **given** two types of objects

- In **perfect clustering**, objects of the same type are clustered together.



Evaluation **with ground truth**  
Evaluation **without ground truth**

# Evaluation with Ground Truth



When ground truth is available,

We have prior knowledge on what the clustering  
should be (the correct clustering assignments)

We will discuss these methods in community  
analysis chapter

## Cohesiveness

In clustering, we are interested in clusters that exhibit cohesiveness

In cohesive clusters, instances inside the clusters are close to each other

## Separateness

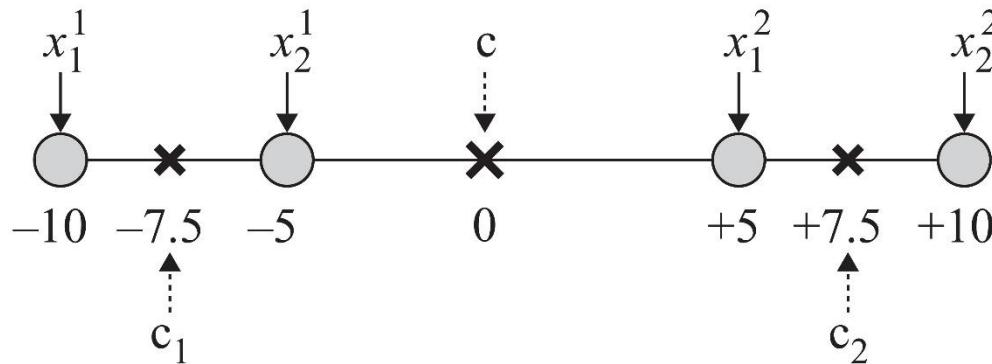
We are also interested in clusterings of the data that generates clusters that are well separated from one another

## Cohesiveness

**In statistics:** having a small standard deviation, i.e., being close to the mean value

**In clustering:** being close to the centroid of the cluster

$$\text{cohesiveness} = \sum_{i=1}^k \sum_{j=1}^{n(i)} \|x_j^i - c_i\|^2$$



$$\text{cohesiveness} = |-10 - (-7.5)|^2 + | -5 - (-7.5)|^2 + | 5 - 7.5 |^2 + | 10 - 7.5 |^2 = 25$$

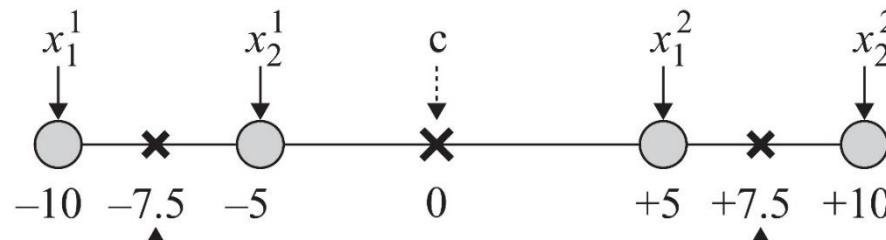
## Separateness

**In statistics:** separateness can be measured by standard deviation

Standard deviation is maximized when instances are far from the mean

**In clustering:** cluster centroids being far from the mean of the entire dataset

$$\text{separateness} = \sum_{i=1}^k \|c - c_i\|^2$$



$$\text{separateness} = | -7.5 - 0 |^2 + | 7.5 - 0 |^2 = 112.5$$

We are interested in clusters that are both **cohesive** and **separate**

*Silhouette index*

It compares

*the average distance value between instances in the **same** cluster*

To

*the average distance value between instances in **different** clusters*

In a well-clustered dataset,

the average distance between instances in the same cluster is **small** (**cohesiveness**) and

the average distance between instances in different clusters is **large** (**separateness**).

For any instance  $x$  that is a member of cluster  $C$

Compute the within-cluster average distance

$$a(x) = \frac{1}{|C|-1} \sum_{y \in C, y \neq x} \|x - y\|^2$$

Compute the average distance between  $x$  and instances in cluster  $G$

$G$  is closest to  $x$  in terms of the average distance between  $x$  and members of  $G$

$$b(x) = \min_{G \neq C} \frac{1}{|G|} \sum_{y \in G} \|x - y\|^2$$

---

Our interest: clusterings where  $a(x) < b(x)$

$$s(x) = \frac{b(x) - a(x)}{\max(b(x), a(x))}$$

$$\text{silhouette} = \frac{1}{n} \sum_x s(x)$$

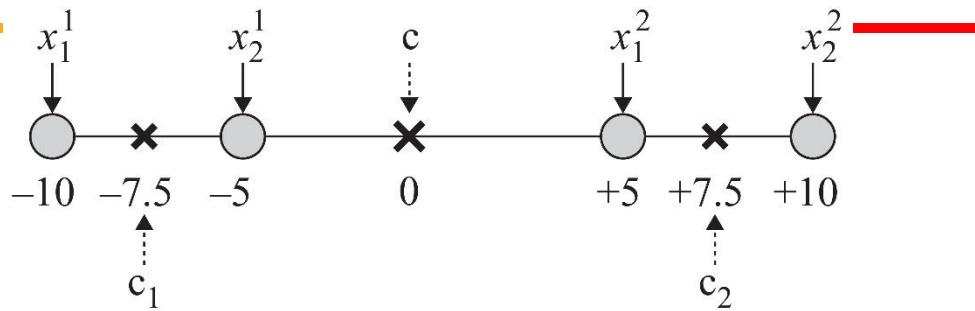
Silhouette can take values between  $[-1,1]$

The best case happens when for all  $x$ ,

$$a(x) = 0, b(x) > a(x)$$

---

# Silhouette Index - Example



$$a(x_1^1) = |-10 - (-5)|^2 = 25$$

$$b(x_1^1) = \frac{1}{2}(|-10 - 5|^2 + |-10 - 10|^2) = 312.5$$

$$s(x_1^1) = \frac{312.5 - 25}{312.5} = 0.92$$

$$a(x_1^2) = |5 - 10|^2 = 25$$

$$b(x_1^2) = \frac{1}{2}(|5 - (-10)|^2 + |5 - (-5)|^2) = 162.5$$

$$s(x_1^2) = \frac{162.5 - 25}{162.5} = 0.84$$

$$a(x_2^1) = |-5 - (-10)|^2 = 25$$

$$b(x_2^1) = \frac{1}{2}(|-5 - 5|^2 + |-5 - 10|^2) = 162.5$$

$$s(x_2^1) = \frac{162.5 - 25}{162.5} = 0.84$$

$$a(x_2^2) = |10 - 5|^2 = 25$$

$$b(x_2^2) = \frac{1}{2}(|10 - (-5)|^2 + |10 - (-10)|^2) = 312.5$$

$$s(x_2^2) = \frac{312.5 - 25}{312.5} = 0.92.$$



---

# Thank you



**BITS** Pilani

Pilani Campus



# Social Media Analytics: Community Analysis – Part 1

Dr. Prasad Ramanathan

[p\\_ramanathan@wilp.bits-pilani.ac.in](mailto:p_ramanathan@wilp.bits-pilani.ac.in)

# Acknowledgment

Course material from the following source is gratefully acknowledged:

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/>**



## [real-world] community

A group of individuals with common *economic*, *social*, or *political* interests or characteristics, often living in *relative proximity*.

# Why analyze communities?

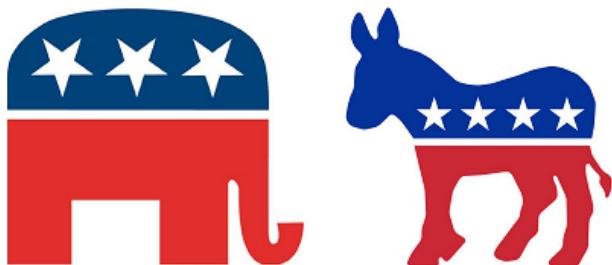
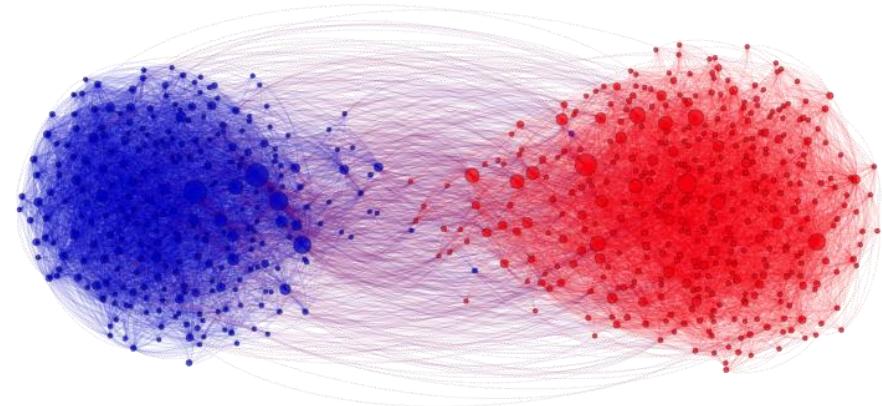


**Analyzing communities helps better understand users**

Users form groups based on their interests

**Groups provide a clear global view of user interactions**

- E.g., find polarization



**Some behaviors are only observable in a group setting and not on an individual level**

- Some republican can **agree** with some democrats, but their parties can **disagree**

## Formation:

When like-minded users on social media form a link and start interacting with each other

## More Formal Formation:

1. A set of at least two nodes sharing some interest, and
2. Interactions with respect to that interest.

## Social Media Communities

**Explicit (emic)**: formed by user subscriptions

**Implicit (etic)**: implicitly formed by social interactions

**Example:** individuals calling Canada from the United States

Phone operator considers them one community for promotional offers

Other community names: *group, cluster, cohesive subgroup, or module*

# Examples of Explicit Social Media Communities



Facebook has groups and communities. Users can post messages and images, can comment on other messages, can like posts, and can view activities of other users



In Google+, Circles represent communities



In Twitter, communities form as lists.

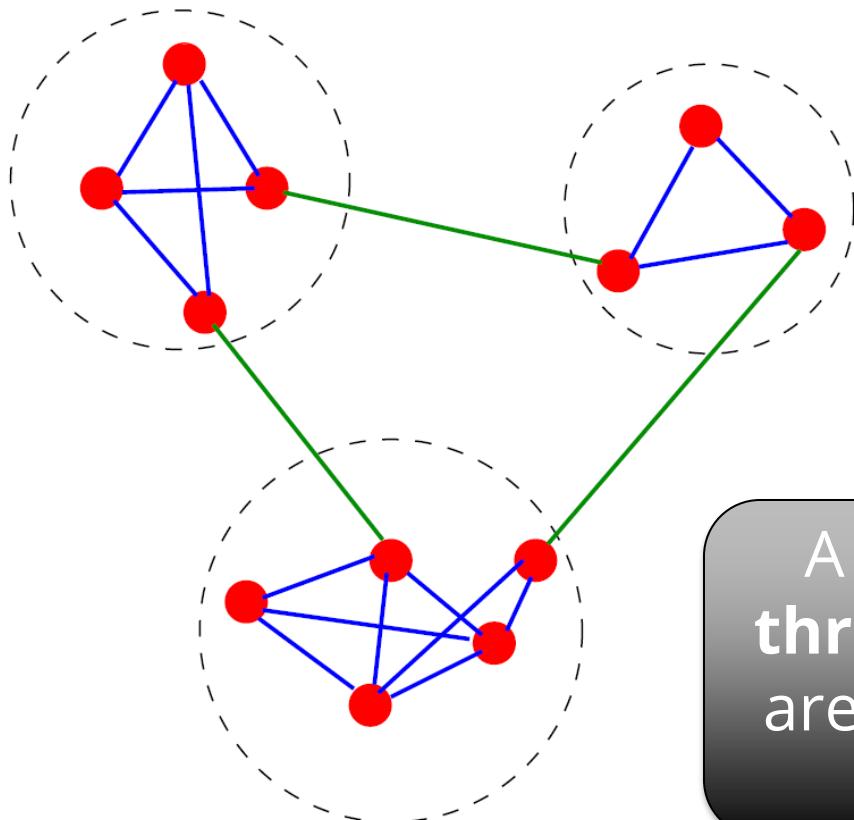
- Users join lists to receive information in the form of tweets



LinkedIn provides *Groups* and *Associations*.

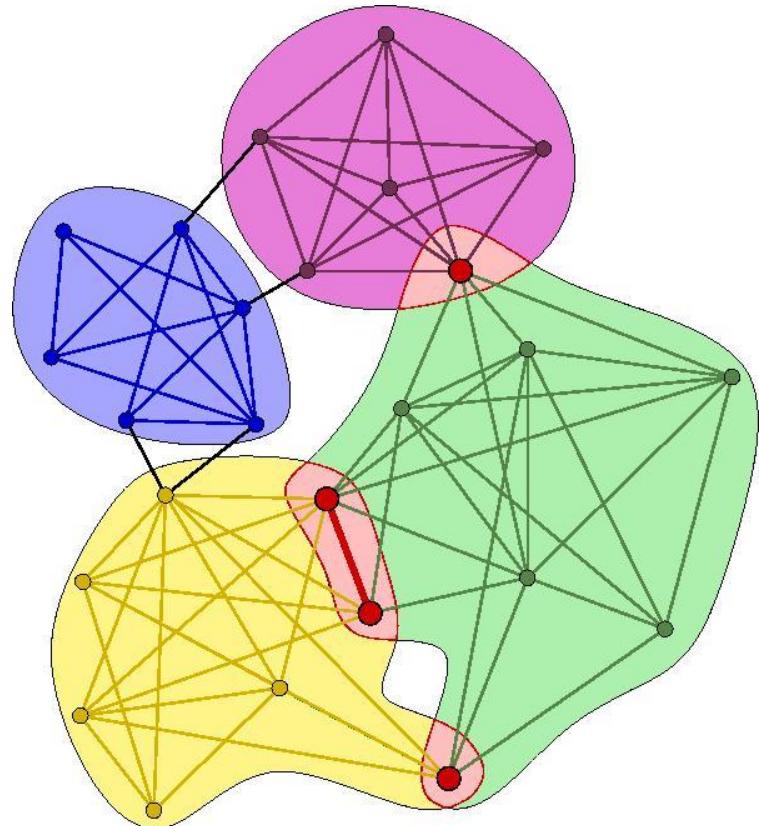
- Users can join professional groups where they can post and share information related to the group

# Finding Implicit Communities: An Example

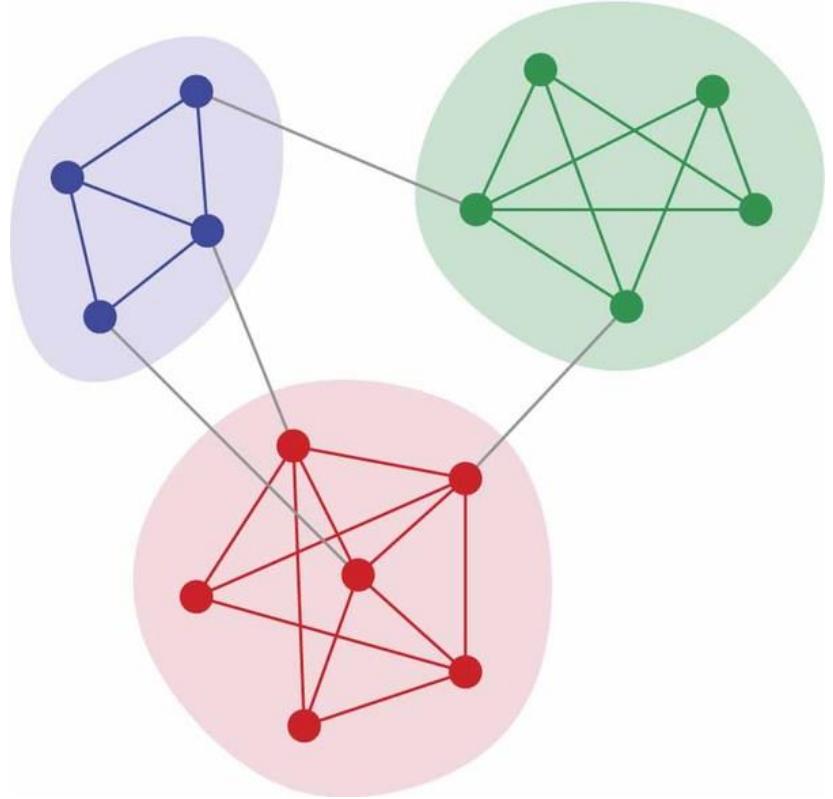


A simple graph in which  
**three** implicit communities  
are found, enclosed by the  
dashed circles

# Overlapping vs. Disjoint Communities



Overlapping Communities



Disjoint Communities

# Implicit communities in other domains



## Protein-protein interaction networks

Communities are likely to group proteins having the same specific function within the cell

## World Wide Web

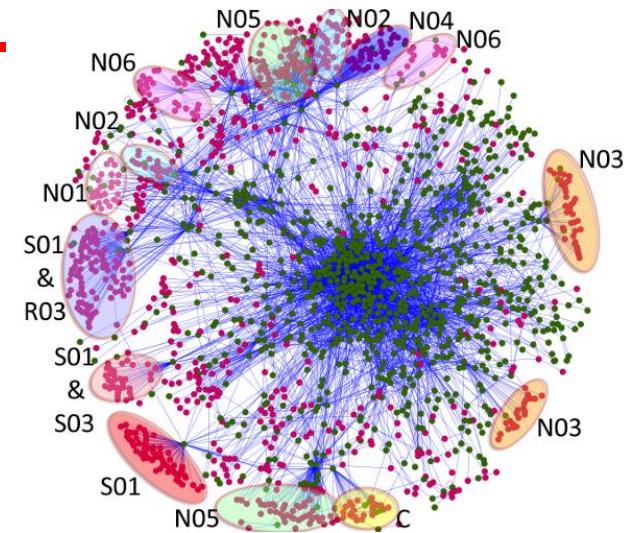
- Communities may correspond to groups of pages dealing with the same or related topics

## Metabolic networks

- Communities may be related to functional modules such as cycles and pathways

## Food webs

- Communities may identify compartments

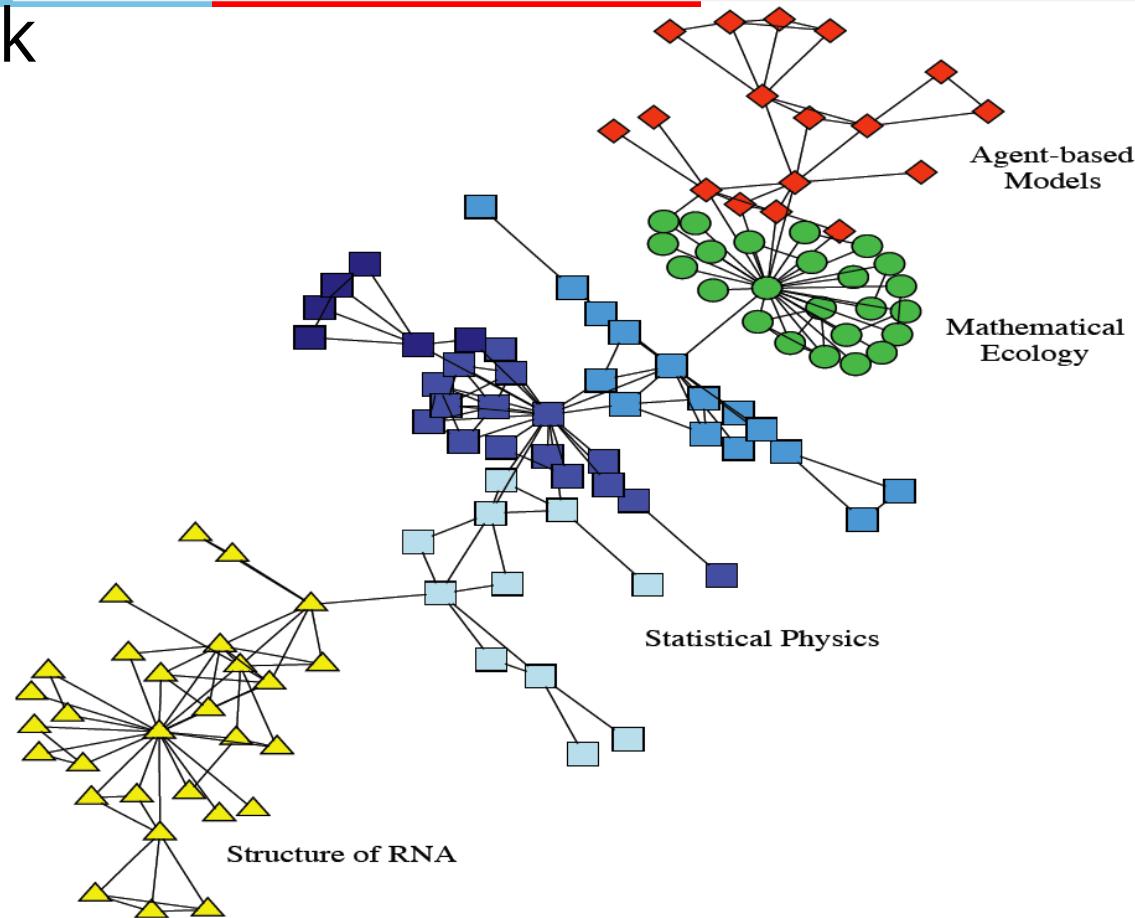


# Real-world Implicit Communities



Collaboration network  
between scientists  
working at the  
Santa Fe Institute.

The colors indicate  
high level  
communities and  
correspond to  
research divisions of  
the institute



# What is Community Analysis?



## Community detection

Discovering implicit communities

## Community evolution

Studying temporal evolution of communities

## Community evaluation

Evaluating Detected Communities



# Community Detection

# What is community detection?



The process of finding clusters of nodes (“*communities*”)

With **Strong** internal connections and  
**Weak** connections between different communities

Ideal decomposition of a large graph

Completely disjoint communities

There are no interactions between different communities.

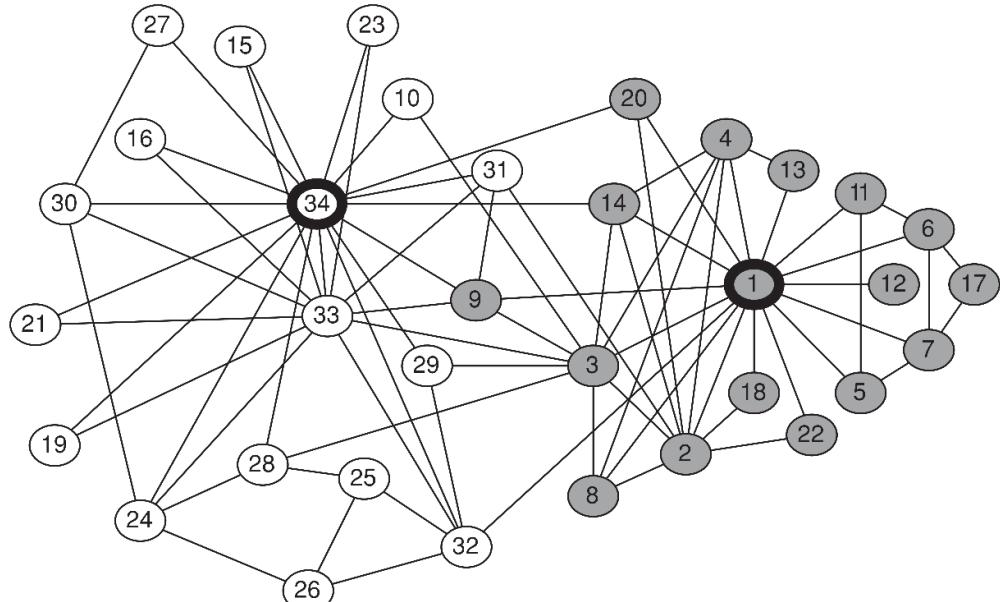
In practice,  
find community partitions that are maximally decoupled.

# Why Detecting Communities is Important?



## Zachary's karate club

Interactions between 34 members of a karate club for over two years



- The club members split into two groups (**gray** and **white**)
- Disagreement between the administrator of the club (node **34**) and the club's instructor (node **1**),
- The members of one group left to start their own club

**The same communities can be found using community detection**

# Why Community Detection?

innovate

achieve

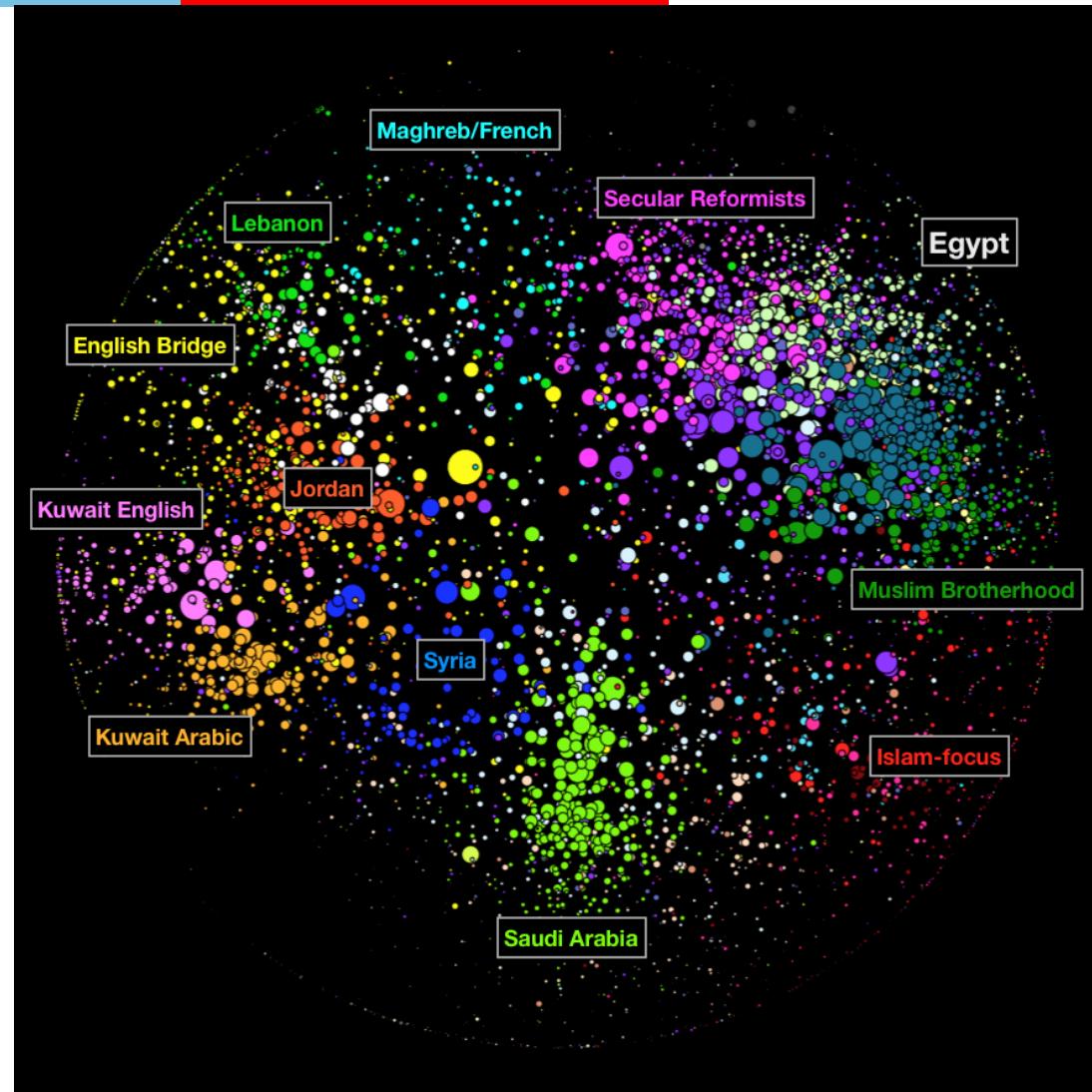
lead

## Network Summarization

A community can be considered as a summary of the whole network  
Easier to visualize and understand

## Preserve Privacy

[Sometimes] a community can reveal some properties without releasing the individuals' privacy information.



# Community Detection vs. Clustering



## Clustering

Data is often non-linked (matrix rows)

Clustering works on the distance or similarity matrix, e.g.,  $k$ -means.

If you use  $k$ -means with adjacency matrix rows, you are only considering the ego-centric network

## Community detection

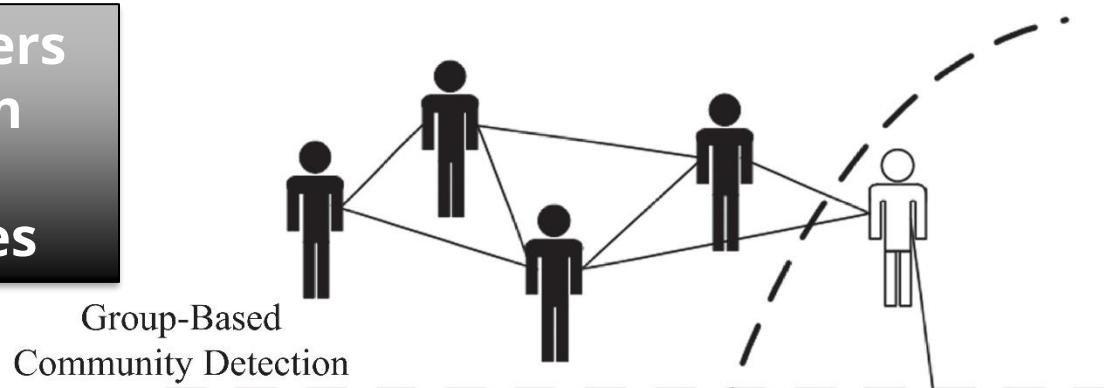
Data is linked (a graph)

Network data tends to be “discrete”, leading to algorithms using the graph property directly  
 $k$ -clique, quasi-clique, or edge-betweenness

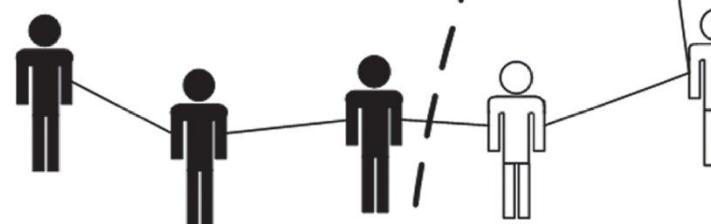
# Community Detection Algorithms



**Group Users  
based on  
Group  
attributes**



**Group Users  
based on  
Member  
attributes**



Member-Based  
Community Detection



# Member-Based Community Detection

# Member-Based Community Detection



Look at node characteristics; and  
Identify nodes with similar characteristics and consider  
them a community

## ***Node Characteristics***

### ***A. Degree***

Nodes with same (or similar) degrees are in one community  
Example: cliques

### ***B. Reachability***

Nodes that are close (small shortest paths) are in one community  
Example:  $k$ -cliques,  $k$ -clubs, and  $k$ -clans

### ***C. Similarity***

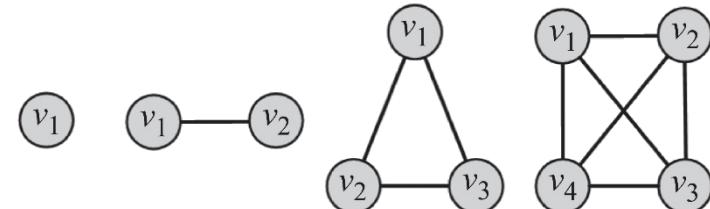
Similar nodes are in the same community

# A. Node Degree



## Most common subgraph searched for:

**Clique:** a maximum complete subgraph in which all nodes inside the subgraph adjacent to each other



Find communities by searching for

1. **The maximum clique:** the one with the largest number of vertices, or
2. **All maximal cliques:** cliques that are not subgraphs of a larger clique; i.e., cannot be further expanded

To overcome this, we can

- I. Brute Force
- II. Relax cliques
- III. Use cliques as the core for larger communities

Both problems are NP-hard

# I. Brute-Force Method



Can find all the maximal cliques in the graph

For each vertex  $v_x$ , we find the maximal clique that contains node  $v_x$

---

#### Algorithm 1 Brute-Force Clique Identification

---

**Require:** Adjacency Matrix  $A$ , Vertex  $v_x$

```
1: return Maximal Clique  $C$  containing  $v_x$ 
2: CliqueStack =  $\{\{v_x\}\}$ , Processed =  $\{\}$ ;
3: while CliqueStack not empty do
4:    $C = \text{pop}(\text{CliqueStack})$ ;  $\text{push}(\text{Processed}, C)$ ;
5:    $v_{last} = \text{Last node added to } C$ ;
6:    $N(v_{last}) = \{v_i | A_{v_{last}, v_i} = 1\}$ .
7:   for all  $v_{temp} \in N(v_{last})$  do
8:     if  $C \cup \{v_{temp}\}$  is a clique then
9:        $\text{push}(\text{CliqueStack}, C \cup \{v_{temp}\})$ ;
10:      end if
11:    end for
12:  end while
13: Return the largest clique from Processed
```

---

### Impractical for large networks:

- For a complete graph of only 100 nodes, the algorithm will generate at least  $2^{99} - 1$  different cliques starting from any node in the graph

# Enhancing the Brute-Force Performance



**[Systematic] Pruning** can help:

When searching for cliques of size  $k$  or larger

If the clique is found, each node should have a degree equal to or more than  $k - 1$

We can first prune all nodes (and edges connected to them) with degrees less than  $k - 1$

More nodes will have degrees less than  $k - 1$

Prune them recursively

For large  $k$ , many nodes are pruned as social media networks follow a power-law degree distribution

# Maximum Clique: Pruning...

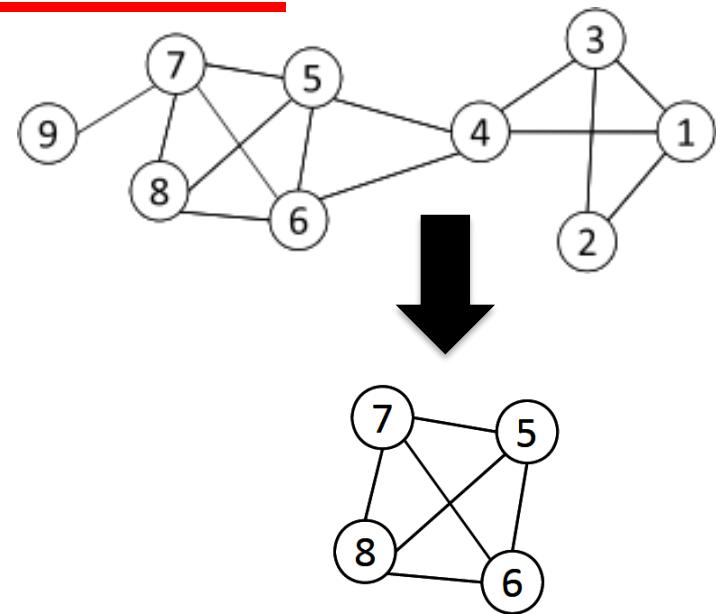


**Example.** to find a clique  $\geq 4$ , remove all nodes with degree  $\leq (4 - 1) - 1 = 2$

Remove nodes 2 and 9

Remove nodes 1 and 3

Remove node 4



Even with pruning, cliques are less desirable

- Cliques are **rare**
- A clique of 1000 nodes, has  $999 \times 1000 / 2$  edges
- **A single edge removal** destroys the clique
- That is less than 0.0002% of the edges!

## II. Relaxing Cliques



**$k$ -plex**: a set of vertices  $V$  in which we have

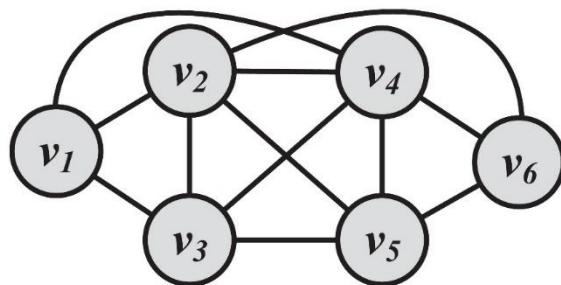
$$d_v \geq |V| - k, \forall v \in V$$

$d_v$  is the degree of  $v$  in the induced subgraph  
Number of nodes from  $V$  that are connected to  $v$

Clique of size  $k$  is a  $1$ -plex

Finding the maximum  $k$ -plex: **NP-hard**

In practice, relatively easier due to smaller search space.



1-plex :  $\{v_2, v_3, v_4, v_5\}$

2-plex :  $\{v_1, v_2, v_3, v_4, v_5\}, \{v_2, v_3, v_4, v_5, v_6\}$

3-plex :  $\{v_1, v_2, v_3, v_4, v_5, v_6\}$

Maximal  $k$ -plexes

# More Cliques Relaxing...

***k*-core:** a maximal connected subgraph in which all vertices have degree at least *k*

Difference with *k*-plex?

***k*-shell:** nodes that are part of the *k*-core, but are not part of the  $(k + 1)$ -core.

## Questions

0-core?

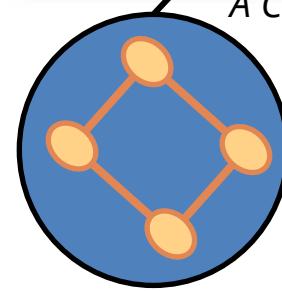
0-shell?

1-core?

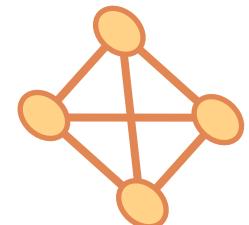
*k*-cores of the complete graph?



A Clique that wants to relax



A 2-plex or a 2-core?



# III. Using Cliques as a seed of a Community



## Clique Percolation Method (CPM)

Uses cliques as seeds to find larger communities  
CPM finds overlapping communities

### Input

A parameter  $k$ , and a network

### Procedure

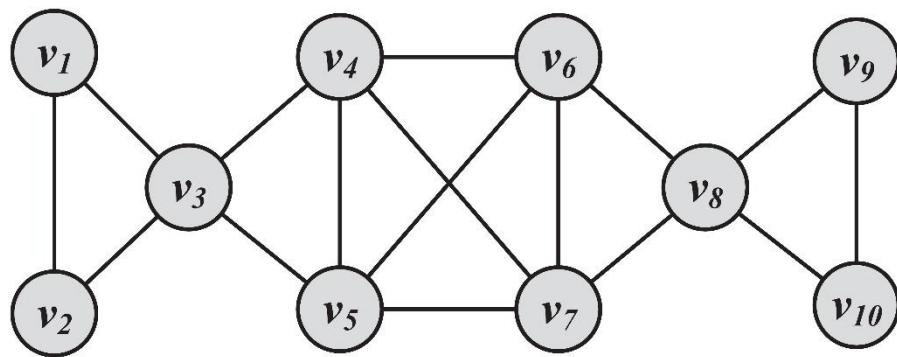
Find out all cliques of size  $k$  in the given network

Construct a clique graph.

Two cliques are adjacent if they share  $k - 1$  nodes

Each connected components in the clique graph  
form a community

# Clique Percolation Method: Example



(a) Graph

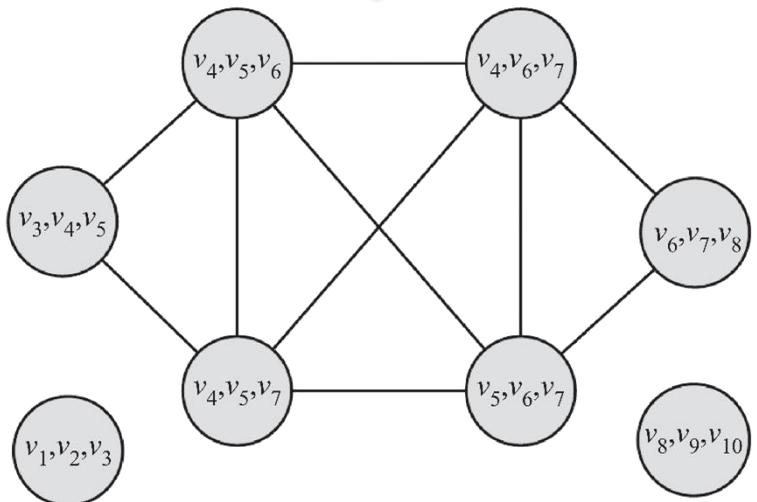
## Cliques of size 3:

$\{v_1, v_2, v_3\}, \{v_3, v_4, v_5\},$   
 $\{v_4, v_5, v_6\}, \{v_4, v_5, v_7\},$   
 $\{v_4, v_6, v_7\}, \{v_5, v_6, v_7\},$   
 $\{v_6, v_7, v_8\}, \{v_8, v_9, v_{10}\}$



## Communities:

$\{v_1, v_2, v_3\},$   
 $\{v_8, v_9, v_{10}\},$   
 $\{v_3, v_4, v_5, v_6, v_7, v_8\}$



(b) CPM Clique Graph

## B. Node Reachability



### The two extremes

Nodes are assumed to be in the same community

1. If there is a path between them (regardless of the distance) or
2. They are so close as to be immediate neighbors.

**How? Find using BFS/DFS**

**Challenge:** most nodes are in one community (giant component)

**How? Finding Cliques**

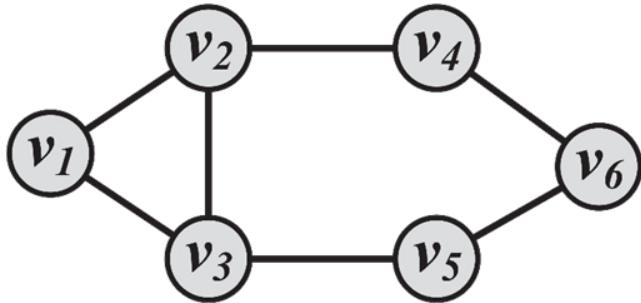
**Challenge:** Cliques are challenging to find and are rarely observed

**Solution:** find communities that are in between **cliques** and **connected components** in terms of connectivity and have small shortest paths between their nodes

# Special Subgraphs



1.  **$k$ -Clique**: a **maximal** subgraph in which the largest shortest path distance between any nodes is less than or equal to  $k$
2.  **$k$ -Club**: follows the same definition as a  $k$ -clique  
**Additional Constraint**: nodes on the shortest paths should be part of the subgraph (i.e., diameter)
3.  **$k$ -Clan**: a  $k$ -clique where for all shortest paths within the subgraph the distance is equal or less than  $k$ .  
All  $k$ -clans are  $k$ -cliques, but not vice versa.



2-cliques :  $\{v_1, v_2, v_3, v_4, v_5\}, \{v_2, v_3, v_4, v_5, v_6\}$

2-clubs :  $\{v_2, v_3, v_4, v_5, v_6\}, \{v_1, v_2, v_3, v_4\}, \{v_1, v_2, v_3, v_5\}$

2-clans :  $\{v_2, v_3, v_4, v_5, v_6\}$

# More Special Subgraphs!



***k*-truss:** the largest subgraph where all edges belong to  $k - 2$  triangles

What is the relationship between a ***k*-core** and ***k*-truss**?

# C. Node Similarity



Similar (or most similar) nodes are assumed to be in the same community.

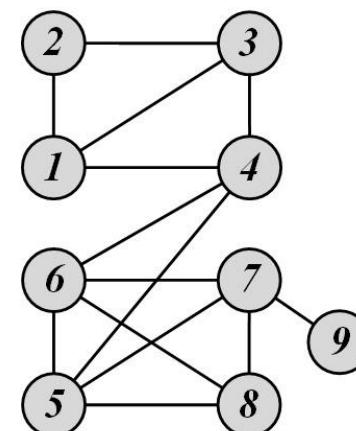
A classical clustering algorithm (e.g.,  $k$ -means) is applied to node similarities to find communities.

Node similarity can be defined

Using the similarity of node neighborhoods (**Structural Equivalence**) – Ch. 3  
Similarity of social circles (**Regular Equivalence**) – Ch. 3

**Structural equivalence:** two nodes are structurally equivalent iff. they are connecting to the same set of actors

*Nodes 1 and 3 are structurally equivalent,  
So are nodes 5 and 7.*



# Node Similarity (Structural Equivalence)

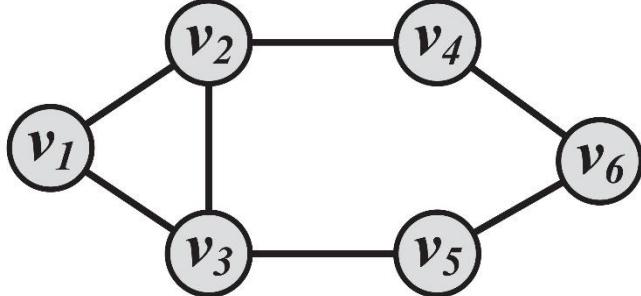


## Jaccard Similarity

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

## Cosine similarity

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



$$\sigma_{\text{Jaccard}}(v_2, v_5) = \frac{|\{v_1, v_3, v_4\} \cap \{v_3, v_6\}|}{|\{v_1, v_3, v_4, v_6\}|} = 0.25$$

$$\sigma_{\text{Cosine}}(v_2, v_5) = \frac{|\{v_1, v_3, v_4\} \cap \{v_3, v_6\}|}{\sqrt{|\{v_1, v_3, v_4\}||\{v_3, v_6\}|}} = 0.40$$



---

# Thank you