Artificial Intelligence: Search & Mining

2015 人工知能: 探索とマイニング

Graph Mining

Kevin Duh

2015-06-02

Today's Agenda

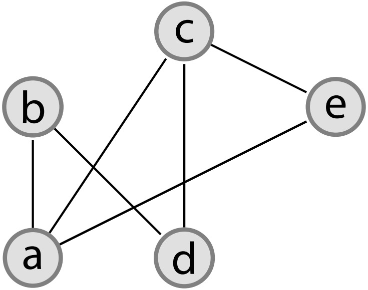
Graph Data

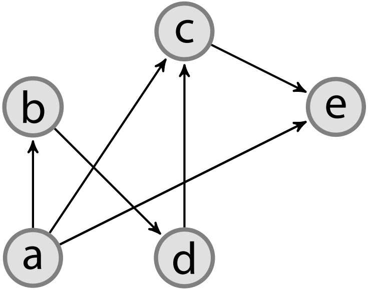
Properties of Graphs

Community Detection

Graph data

Graph $G = (\text{Vertices } V, \text{Edges } E)$
Edges may be weighted, **undirected** or **directed**.





Graph data appears everywhere

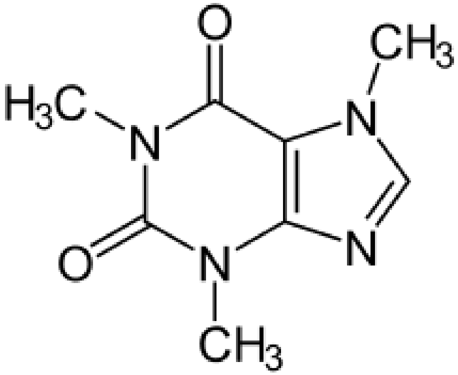


Figure : Chemical structure of caffeine

http://en.wikipedia.org/wiki/Caffeine#mediaviewer/File:Koffein_-_Caffeine.svg

Graph data appears everywhere

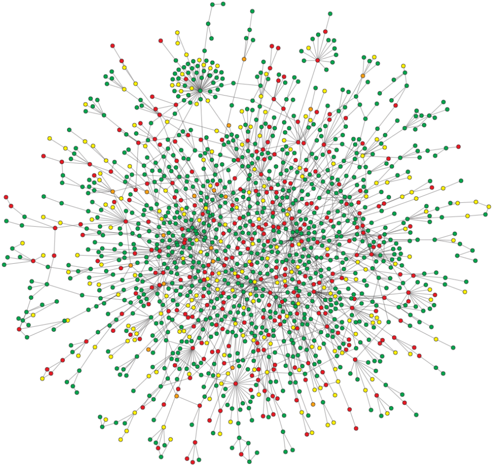


Figure : Yeast protein interaction network

http://www.nature.com/nature/journal/v411/n6833/full/411041a0.html

Graph data appears everywhere

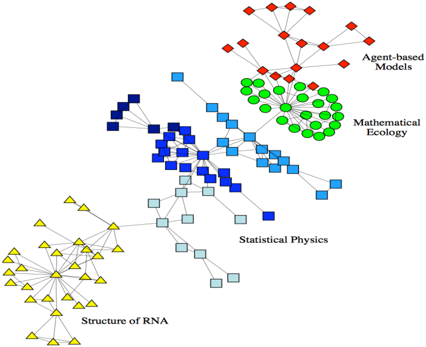


Figure : Collaboration graph among researchers

http://www.pnas.org/content/99/12/7821.full

Graph data appears everywhere

Figure : Facebook Friendship Graph

https://www.facebook.com/notes/facebook-engineering/visualizing-friendships/469716398919

Many ways to make graphs

Facebook example:

- Friendship graph: vertices = users; edges = is-a-friend

Many ways to make graphs

Facebook example:

- Friendship graph: vertices = users; edges = is-a-friend
- Activity graph: vertices = users; edges = recently-talked

Many ways to make graphs

Facebook example:

- Friendship graph: vertices = users; edges = is-a-friend
- Activity graph: vertices = users; edges = recently-talked
- Like! graph: vertices = posts and users (bipartite); edges = user-likes-post

Many ways to make graphs

Facebook example:

- Friendship graph: vertices = users; edges = is-a-friend
- Activity graph: vertices = users; edges = recently-talked
- Like! graph: vertices = posts and users (bipartite); edges = user-likes-post

Graph mining questions we might ask

- Drug design
  - What are frequent sub-structures in a chemical database?
  - Can we search for similar chemicals?

Graph mining questions we might ask

- Drug design
  - What are frequent sub-structures in a chemical database?
  - Can we search for similar chemicals?
- Biology research
  - What are the central proteins in a metabolic pathway, if any?

Graph mining questions we might ask

- Drug design
  - What are frequent sub-structures in a chemical database?
  - Can we search for similar chemicals?
- Biology research
  - What are the central proteins in a metabolic pathway, if any?
- Social network analysis
  - Does there exist distinct communities?
  - How do links form?
  - How do messages get disseminated?
- etc.

Tools/Concepts for answering graph mining questions

- Community Detection
- Graph Clustering
- Centrality Analysis, e.g. PageRank
- Link Prediction
- Frequent sub-graph mining
- Information diffusion on graphs
- Graph evolution, etc.

Today’s Agenda

Graph Data

Properties of Graphs

Community Detection

Characterizing Graphs: Diameter

- Diameter of graph G = *maximum* distance between all pairs of vertices

Characterizing Graphs: Diameter

- Diameter of graph G = *maximum* distance between all pairs of vertices
    - Distance between a pair of vertices (v_1, v_2) is measured by length of *shortest path* from v_1 to v_2.

Characterizing Graphs: Diameter

- Diameter of graph G = *maximum* distance between all pairs of vertices
  - Distance between a pair of vertices (v_1, v_2) is measured by length of *shortest path* from v_1 to v_2.
- Related concept: average distance

Characterizing Graphs: Diameter

- Diameter of graph G = *maximum* distance between all pairs of vertices
  - Distance between a pair of vertices (v_1, v_2) is measured by length of *shortest path* from v_1 to v_2.
- Related concept: average distance
- Small-World Phenomenon: 6 degrees of separation between any two people (Milgram experiment)

Characterizing Graphs: Degree

- Degree of a vertex v_i:
  d_i = Number of edges for vertex v_i
  - For directed graphs: separately define in-degree & out-degree

Characterizing Graphs: Degree

- Degree of a vertex v_i:
  d_i = Number of edges for vertex v_i
  - For directed graphs: separately define in-degree & out-degree
- Average degree = average number of edges per vertex

Characterizing Graphs: Degree

- Degree of a vertex v_i:
  d_i = Number of edges for vertex v_i
  - For directed graphs: separately define in-degree & out-degree
- Average degree = average number of edges per vertex
- Degree distribution:
  - uniform or power-law?
  - are there popular hub vertices?

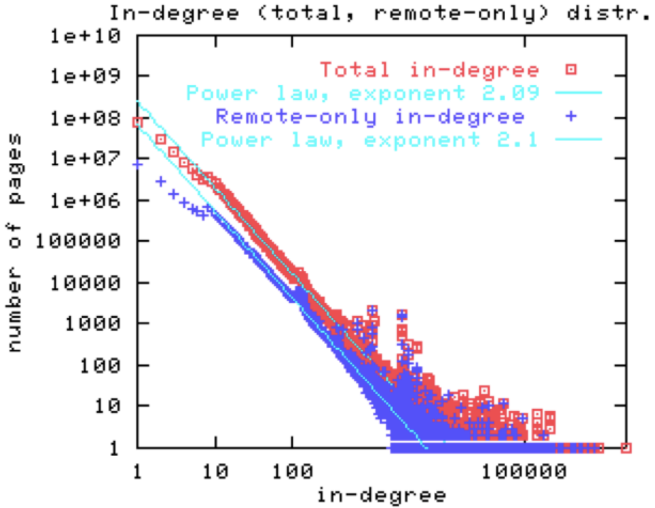
Power-law degree distribution is prevelant in real graphs

- Consider Gaussian distribution: $p(d) \propto exp(-(d-\mu)^2)$: exponentially fast decay as d moves away from μ

Power-law degree distribution is prevelant in real graphs

- Consider Gaussian distribution: $p(d) \propto exp(-(d-\mu)^2)$: exponentially fast decay as d moves away from μ
- Power law: $p(d) \propto 1/d^\beta$ gives heavy-tail, i.e. vertices with very high degree can exist
  - straight-line on log-log plot: $\log(p(d)) = \beta \log(d)$

Power-law in WWW graphs



[Broder et. al., Graph Structure in the Web]

Characterizing Graphs: Clustering coefficient

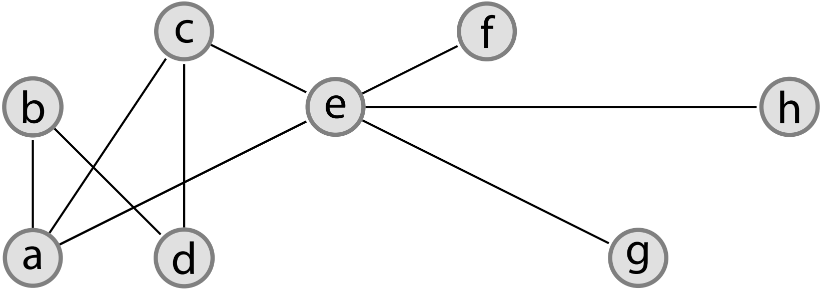
- Neighborhood of vertex v_i:
  $N_i = \{v_j : e_{ij} \in E \wedge e_{ji} \in E\}$
- Cluster coefficient of v_i:

$$C_i = \frac{|e_{jk} : v_j \in N_i, v_k \in N_i, e_{jk} \in E|}{|N_i|\left(|N_i| - 1\right)}$$

  i.e. percentage of triangles (i,j,k)
- Cluster coefficient C of graph = avg C_i

Quiz

What is the diameter? degree distribution? cluster coefficient of vertex a?



Erdős-Rényi model of random graph

1. Start with N vertices
2. Connect every pair of vertices with probability p

Graph will have about $pN(N-1)/2$ edges distributed randomly

Erdős-Rényi model of random graph

1. Start with N vertices
2. Connect every pair of vertices with probability p

Graph will have about $pN(N-1)/2$ edges distributed randomly

- Diameter = log(N) $\rightarrow$ "small world"
- Degree distribution = Poisson(pN), not power-law
- Clustering coefficient = p, no hierarchical clusters

Properties of Real-world Graphs

From: Albert & Barabási, Statistical mechanics of complex networks, 2002

Data	WWW [Broder]	Co-Author [Newman]	Movie [Watts]
size |V|	2×10^8	56,627	225,226
avg degree	7.5	173	61
power-law β	2.71, 2.1	1.2	n/a
avg distance ℓ	16	4	3.65
$\ell_{randomgraph}$	8.85	2.12	2.99
cluster coeff C	n/a	0.726	0.79
$C_{randomgraph}$	n/a	0.003	0.00027

Today's Agenda

Graph Data

Properties of Graphs

Community Detection

Community Detection

Given a graph G=(V,E), find subsets of V that form communities

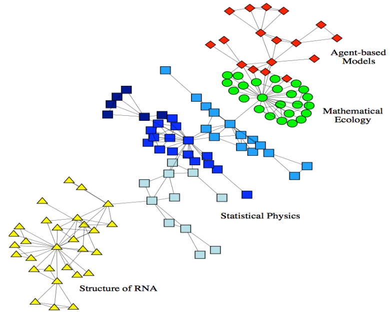


Figure : Do you see distinct communities of researchers in this collaboration graph?

http://www.pnas.org/content/99/12/7821.full

A Method for Community Detection

Betweenness of edge (A,B) = # pairs of endpoints X & Y such that (A,B) lies on the shortest path between X and Y

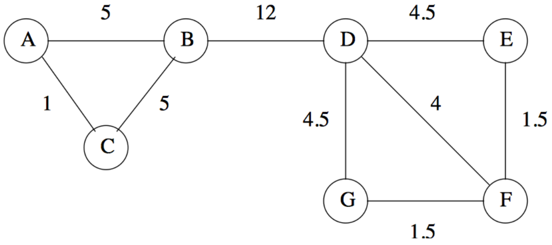


Figure : Betweenness example

All figures in this section come from http://infolab.stanford.edu/~ullman/mmds/ch10.pdf

A Method for Community Detection

To detect communities, delete edges with high betweeness

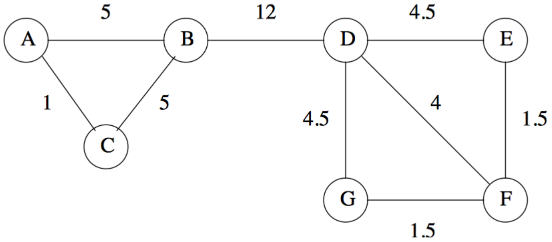


Figure : (B,D) has highest betweeness. So communities are {A,B,C} and {D,E,G,F}

All figures in this section come from http://infolab.stanford.edu/~ullman/mmds/ch10.pdf

Betweenness Calculation: Girvan-Newman Algorithm

1. Run breadth-first search from a vertex
2. Label each vertex and edge with the # of shortest paths that passes through it.

Repeat for each vertex, sum edge scores / 2.

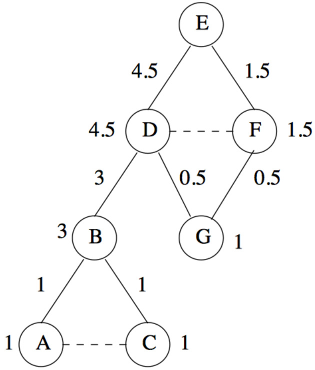


Figure : BFS from E

Betweenness Calculation: preparation

label from top-down:

- root: 1
- other vertex: sum of parent labels

result: for each X, # of shortest paths from E to X is known

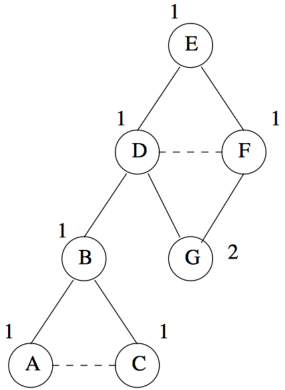


Figure : top-down labeling (preparation)

Betweenness Calculation: vertex/edge labeling in detail

label from bottom-up:

- leaf vertex: 1
- internal vertex: 1 + children edge scores
- edge: a fraction of the child vertex score

fraction computed by # of shortest paths to child through edge (preparation)

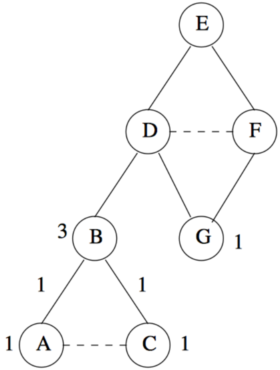


Figure : bottom-up labeling

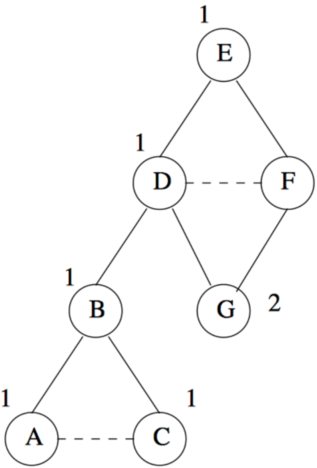


Figure : top-down labeling (preparation)

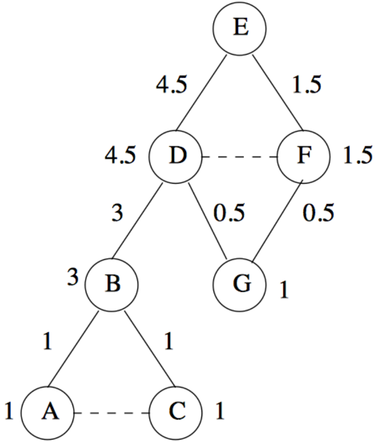


Figure : bottom-up labeling: score indicates # of shortest paths from E that passes through.

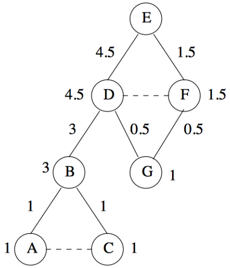
Wrap-up: Community Detection by Betweenness

Betweenness calculation by BFS

To find community, delete edges with high betweenness

Cost: O(|E|) per BFS & labeling, so O(|V||E|) total

Many other methods available!



Summary

- Graph data are everywhere

Summary

- Graph data are everywhere
- Many graph mining tools & problems
  - frequent sub-graph mining, centrality analysis, link prediction, community detection, etc.

Summary

- Graph data are everywhere
- Many graph mining tools & problems
  - frequent sub-graph mining, centrality analysis, link prediction, community detection, etc.
- Properties of graphs:
  - diameter, small-world
  - degree distribution, power-law
  - cluster coefficient

Summary

- Graph data are everywhere
- Many graph mining tools & problems
  - frequent sub-graph mining, centrality analysis, link prediction, community detection, etc.
- Properties of graphs:
  - diameter, small-world
  - degree distribution, power-law
  - cluster coefficient
- Community Detection
  - a method based on betweenness