Lecture 1 - What is graph mining?

Sunday, January 9, 2022 11:30 PM

Plan for today:

- About me and the course
- Go over syllabus
- Go over website: https://www.cs.rpi.edu/~slotag/classes/SP22m/index.html
- Talk about project
- Lecture
  - Define graphs ✓
  - Real-world graph properties ✓
  - Graph mining applications ✓
  - Graph processing approaches
  - NetworkX and connected components

Project Details

- Groups of 1-4
- Take a look at data repos
- Prior projects
  - * Predicting chess match outcomes from a competition
  - * Using vertex centrality to identify population centers in a road network
  - * Recommender systems on Amazon co-purchasing network
- Grad students: you can use

- Grad students: you can use your research topic

Graph definitions

Graph → vertex set $V(G)$
edge set $E(G)$
weights $W(G)$

$$G = \{V(G), E(G), W_V(G), W_E(G)\}$$

$$V(G) = \{v_1, v_2, \ldots, v_n\}$$
$$E(G) = \{e_1 = (v_i, v_j), e_2 \ldots e_m\}$$

Vertices => represent discrete objects

Edges => represent connections between objects

Weights => strength of connections
importance of objects

Vertices/Edges can also have associated meta-data/labels

Temporal graph: vertices and/or

Temporal graph: vertices and/or edges can have associated timestamps

---

Real-world graph properties

Sparsity: $|E(G)| << |V(G)|^2$

$m << n^2$

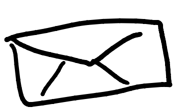
Degree Skew: # low degree verts >> # of high degree verts

Hubs: high degree or otherwise exceptionally "important" vertices

Irregularity: information, social, etc. nets not physically constrained

Small-world: average shortest paths lengths are small relative to the size of the network

"6 degrees of Kevin Bacon"

Note : Properties are typical, BUT not every real-world network not every real-world network will exhibit ALL properties

---

Graph Mining Applications

Take a look at syllabus and context for real-world SNAP data

Similarity to "data mining"

- Classification => can we "classify" a vertex with same property
- Clustering => approach for classification, e.g. clustering similar vertices together
- Prediction => how will our network evolve over time
- Diffusion => how does "information" dynamically move through a network

- Measurement => how do we measure properties of a network

Graph Mining == Data Mining, but on graphs

Note: any dataset can be made graphical

Graph Processing Approaches

"Vertex-centric" processing

- Each vertex has some "state"
- We update that state
- We perform some # of iterations
- Many/most graph algorithms can be implemented in a vertex-centric way

Algorithmically:

Input: $G = \{V, E\}$

For all $v \in V(G)$:

$S(v) = \text{initialize}()$

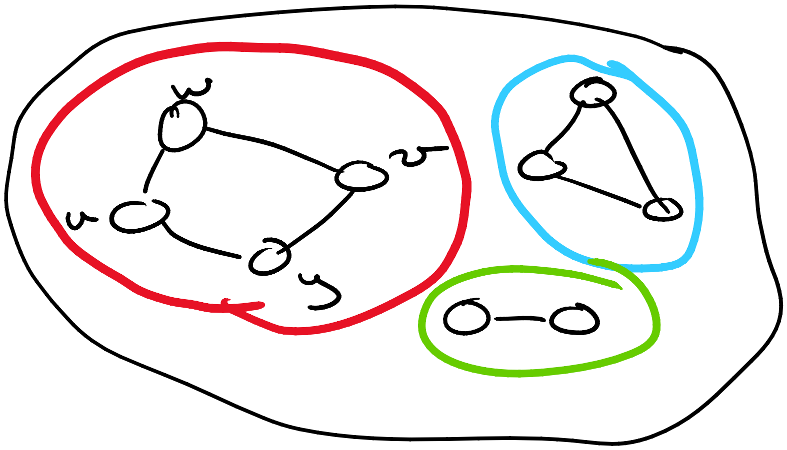
For some # iterations:
    updateAlgorithmicData()
    For all $v \in V(G)$:
        For all $u \in N(v)$ ← neighbors of v
            $S(v)$ = updateState($S(v)$, $S(u)$)

For all $v \in V(G)$:
    $S(v)$ = Finalize()

Implementing connectivity as vert-centric

u, v are connected if there exists a path from u to v

→ connected components: maximal subgraphs where all vertices are connected



G has 3 connected components