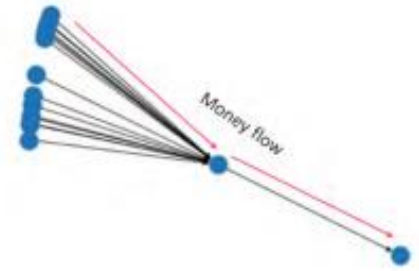
COMP261
Algorithms and Data Structures
2024 Tri 1

Jyoti Sahni
jyoti.sahni@ecs.vuw.ac.nz
Office Hours (COMP261): AM414, Thursday 10:00 – 12:00

Centrality

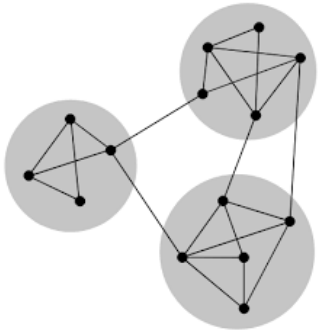
Relates to importance / ranking of nodes



Use cases:

- Financial - Find accounts through which most of the money flows

- Business: Engaging with social media influencers to promote brand awareness



- Network resilience: How important a node is to maintain connectivity within the network

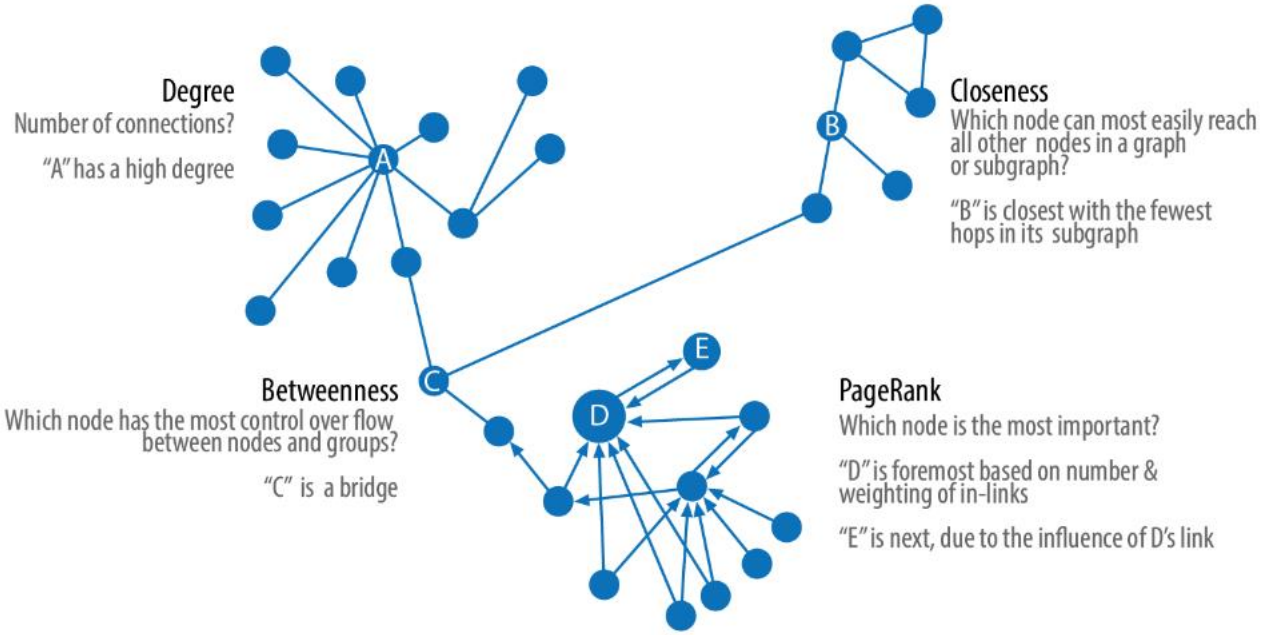
Centrality

Centrality algorithms are used to understand the roles of particular nodes in a graph and their impact on that network.

- Degree centrality – Baseline metric
- Closeness centrality – How central a node is to the group
- Between centrality – finding control points
- Ranking – Overall influence

Different centrality algorithms can produce significantly different results based on what they were created to measure.

Centrality



Source: Graph Algorithms by Mark Needham and Amy E. Hodler

Centrality:

Degree centrality, Closeness centrality Between centrality, Page Rank

Example	Centrality Measure	What it does
Estimating a person's popularity by looking at their in-degree and using their out-degree to estimate gregariousness	**Degree Centrality**	Measures the number of relationships a node has
Finding the optimal location of new public services for maximum accessibility	**Closeness Centrality**	Calculates which nodes have the shortest paths to all other nodes
Finding nodes that have more control over the network as more information pass through this node	**Betweenness Centrality**	Measures the number of shortest paths that pass through a node
Providing an importance score to nodes in a network to find the most influential nodes within the network	**PageRank**	Estimates a current node's importance from its linked neighbours and their neighbours (used by Google search engine for ranking web pages)

Note: Many centrality algorithms calculates shortest paths between every pair of vertices. This works well for small and medium sized graphs but for large graphs this can be computationally prohibitive. To avoid long runtimes on large graphs, some algorithms (e.g. Betweenness centrality)have approximating algorithms.

Degree Centrality

Simplest of all the centrality measures

Counts the number of incoming and outgoing relationships from a node.

Captures the immediate reach(direct relationships) of a node.

Average degree of network: total number of relationships divided by the total number of nodes.

It can be heavily skewed by high degree nodes.

Works well if we are only concerned with immediate connectedness

Also useful for global analysis – minimum degree, maximum degree and standard deviation across the entire graph

Closeness Centrality

Closeness centrality indicates how close a node is to all other nodes in the network

Closeness is calculated as the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph

Closeness is defined as the reciprocal of the farness that is:

$$C(u) = \frac{1}{\sum_{v=1}^{n-1} d(u,v)}$$

where

$d(u,v)$is the distance (length of the shortest path) between vertices u and v.

Closeness Centrality

Thus, the more central a node is, the closer it is to all other nodes

It is common to normalize this score so that it represents the average length of the shortest paths instead of their sum.

$$C_{norm}(u) = \frac{n-1}{\sum_{v=1}^{n-1} d(u,v)}$$

The normalization of closeness simplifies the comparison of nodes in graphs of different sizes.

Betweeness Centrality

Betweenness centrality represents the degree to which nodes stand between each other.

A node with higher Betweenness centrality would have more control over the network, because more information will pass through that node.

Betweenness centrality for each vertex is the number of the shortest paths that pass through the vertex.

$$B(u) = \sum_{s \neq u \neq t} \frac{\sigma_{st}(u)}{\sigma_{st}}$$

Where, σ_{st} is the total number of shortest paths between node s to node t and $\sigma_{st}(u)$ is the number of those paths that pass through u.

Betweeness Centrality

Betweenness centrality is used to find bottlenecks, control points and vulnerabilities

The Betweenness Centrality algorithm can be very resource intensive to compute especially for large graphs.

Many a times approximation algorithms are used.

E.g. Randomized approximate Brandes (RA Brandes)

Betweenness centrality makes an assumption that all communication between the nodes happen along the shortest path and with the same frequency, which isn't always the case in real life. It doesn't necessarily give a perfect view of the most influential nodes

Page Rank

All the centrality measures (covered till now) measure the direct influence of a node. Page rank measure the transitive influence of nodes (influence of neighbours and neighbours of neighbours)

Page Rank, named after both “web page” and Google co-founder Larry Page, was the first algorithm that was used by Google to rank websites in their search engine results.

How PageRank works for the Google search engine: It counts the number and quality of links to a page to determine a rough estimate of how important the website is.

The underlying assumption is that a page with more incoming and more influential incoming links is more likely a credible source.

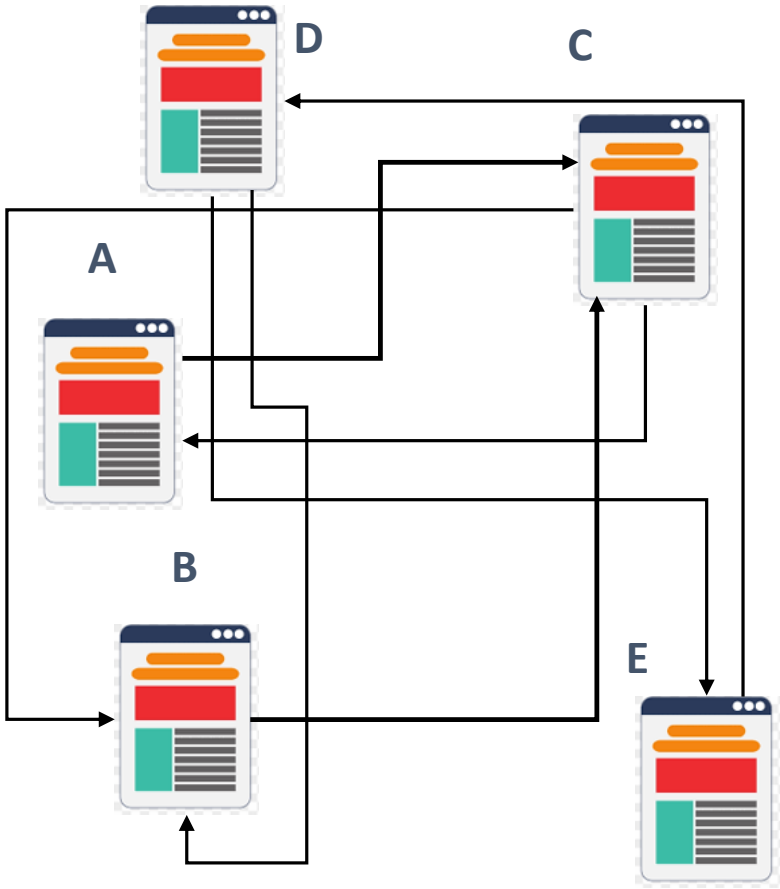
Some Preliminaries: Page Rank

- Backedge: If $page\ A$ links out to $page\ B$, then $page\ B$ is said to have a "backlink" from $page\ A$.

- $PR(P)$ — Each page P has a notion of its own page rank.

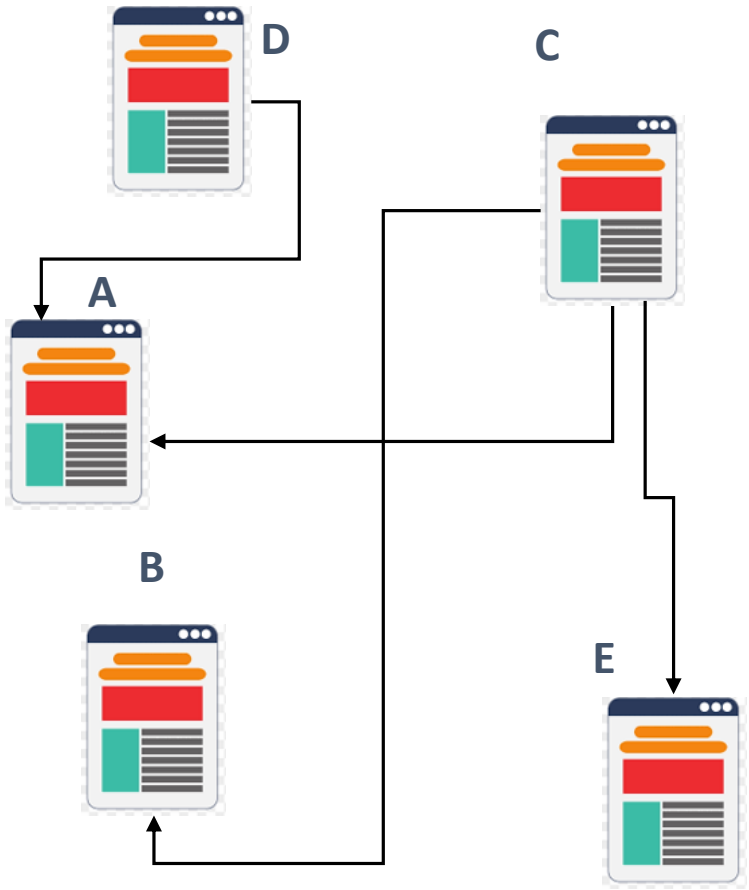
- $C(P)$ — Count of outgoing links for page P. Each page spreads its vote out evenly amongst all of it's outgoing links.

- We'll study a simplified version



Some Preliminaries: Page Rank

- Page rank of a page P is dependent on the page rank of the pages that have an outgoing link to P (nodes pointing to P)
- E.g. Page rank of A depends upon page ranks of C and D.



- The PageRank transferred from a given page to the targets of its outgoing links is divided equally among all outbound links.
- E.g. If the page rank of C is .3, both the page rank transferred to A, B and E will be $.3/3 = .1$

Some Preliminaries: Page Rank

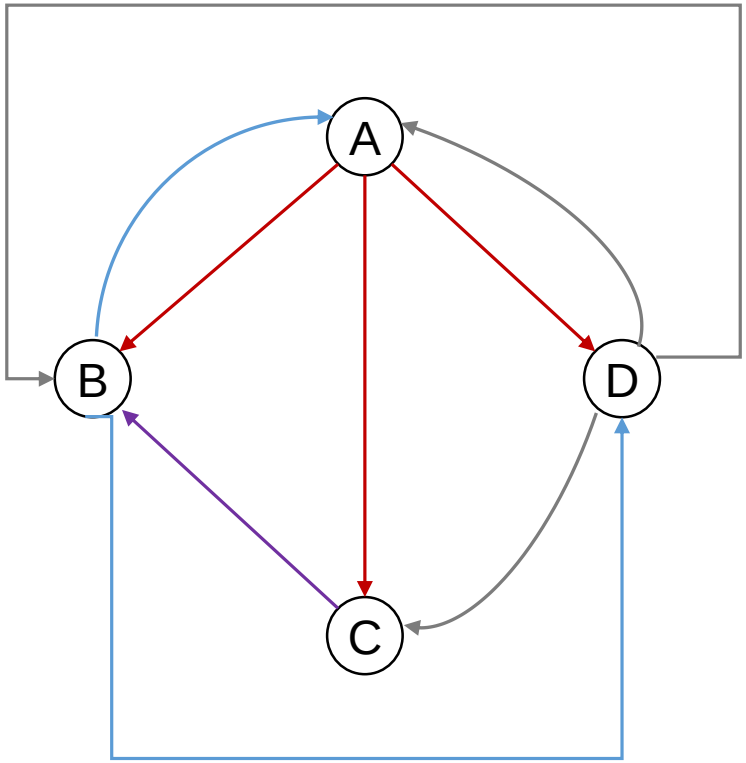
- **Random Surfer model:** Models how someone might browse the web without any particular goal in mind. The PageRank theory holds that an imaginary surfer who is randomly clicking on links will eventually stop clicking.
- **Damping factor:** The probability, at any step, that the person will continue following links is a damping factor d. The probability that they instead jump to any random page is $1 - d$.
- When calculating PageRank, pages with no outbound links are assumed to link out to all other pages in the collection.

$$PR(A) = \left(\frac{1-d}{N}\right) + d \sum_{B \text{ in } inLinkNeighbours \text{ of } A} \frac{PR(B)}{count(outbound\ links\ of B)} + \sum_{K \text{ with } noOutLinks} \frac{PR(K)}{count(Number\ of\ Nodes\ in\ G)}$$

```
computePageRank(Graph graph, int iter, double dampingfactor){
   nNodes = get count of nodes in the graph
   //initialize
   for each node n in graph
      set pageRank(n) = 1.0/nNodes
   endFor
   count = 1
   Repeat
    noOutLinkShare = 0
    for each node n in the graph that has no outlinks
       noOutLinkShare = noOutLinkShare + dampingfactor x (pageRank(n)/nNodes)
    endFor
    for each node n in the graph
       nRank = noOutLinkShare + (1 − dampingFactor )/nNodes // Page rank from
                                                            // random jumps
       neighboursShare = 0
       for each backneighbour b of n
          neighboursShare = neighboursShare + pageRank(b)/count of outedges of b
       EndFor
       NewpageRank(n) = nRank + dampingFactor x neighboursShare
   endFor
   Update pageRank with NewpageRank       //update page-rank for each page
                                          //with the newly computed values
   count++;
   Until count > iter
}
```

Page Rank Example

- Page rank score transfer between nodes:
- A: shared between 3 (B, C, D)
- B: Shared between 2 (A, D)
- C: Shared with 1 (B)
- D: Shared with 3 (A, B, C)



$$\mathrm{PR(A)} = \mathrm{PR\,(B)} = \mathrm{PR(C)} = \mathrm{PR(D)} = \frac{1}{4}$$

Iter 1:

$$\mathrm{PR(A)} = \frac{.15}{4} + .85 * (\frac{\mathrm{PR(B)}}{2} + \frac{\mathrm{PR(D)}}{3})$$

$$\mathrm{PR(B)} = \frac{.15}{4} + .85 * (\frac{\mathrm{PR(A)}}{3} + \frac{\mathrm{PR}(C)}{1} + \frac{\mathrm{PR(D)}}{3})$$

$$\mathrm{PR(C)} = \frac{.15}{4} + .85 * (\frac{\mathrm{PR(A)}}{3} + \frac{\mathrm{PR(D)}}{3})$$

$$\mathrm{PR(D)} = \frac{.15}{4} + .85 * (\frac{\mathrm{PR(A)}}{3} + \frac{\mathrm{PR(B)}}{2})$$

Next Lecture

- Cycles and Spanning trees