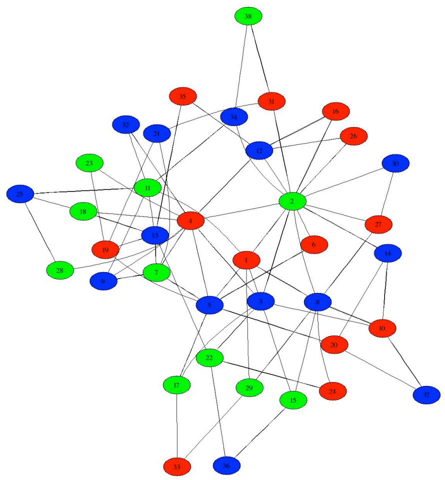
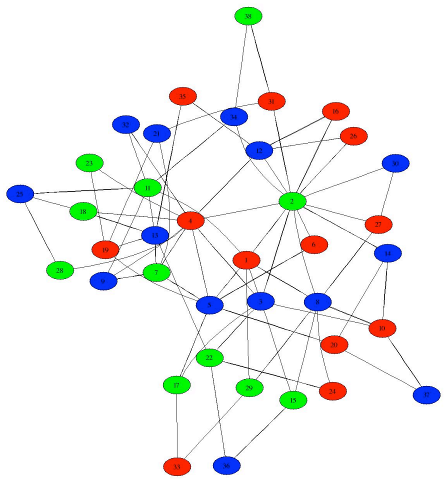
“Important” Vertices and the PageRank Algorithm

Networked Life
NETS 112
Fall 2014
Prof. Michael Kearns



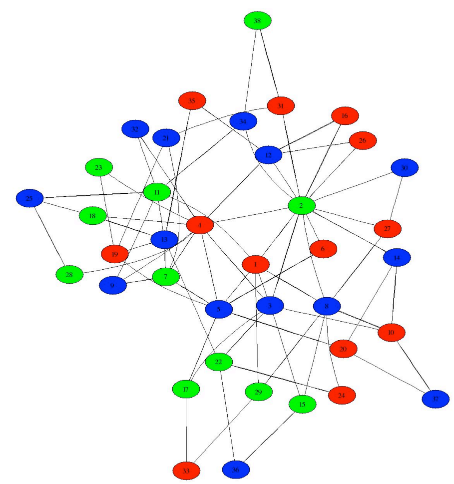
Lecture Roadmap

- Measures of vertex importance in a network
- Google's PageRank algorithm



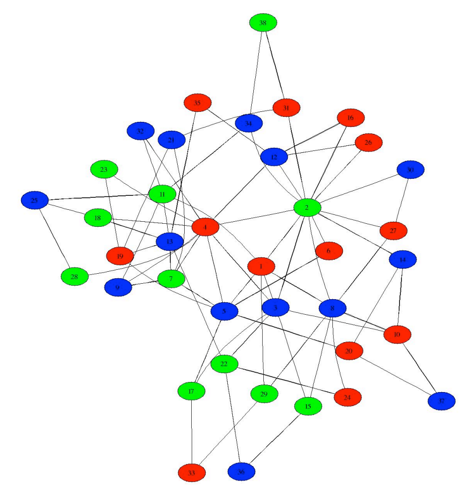
Important Vertices

- So far: have emphasized macroscopic aspects of network structure
  - degree distribution: all degrees
  - diameter: average over all vertex pairs
  - connectivity: giant component with most vertices
  - clustering: average over all vertices, compare to overall edge density
- Also interesting to identify "important" individuals within the network
- Some purely structural definitions of importance:
  - high degree
  - link between communities
  - betweeness centrality
  - Google's PageRank algorithm



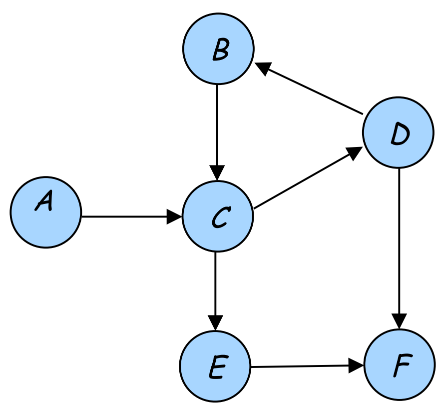
Background on Web Search

- Most important sources of information: words in query and on web pages
- For instance, on query "mountain biking":
  - realize "bike" and "bicycle" are synonymous under context "mountain"
  - but "bike" and "motorcylce" are not
  - find documents with these words and their correlates ("Trek", "trails")
- Subject of the field of information retreival
- But many other "features" or "signals" may be useful for identifying good or useful sites:
  - font sizes
  - frequency of exclamation marks
- PageRank idea: use the link structure of the web



Directed Networks

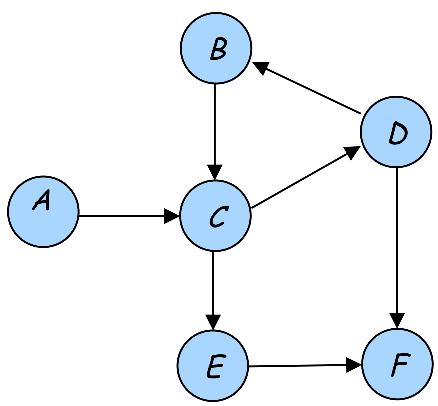
- The web graph is a directed network:
  - page A can point/link to page B, without a reciprocal link
  - represent directed edges with arrows
  - each web page/site thus has "in-links" and "out-links"
  - in-degree = # of in-links; out-degree = # of out-links
- Q: What constitutes an "important" vertex in a directed network?
  - could just use in-degree; view directed links as referrals
- PageRank answer: an important page is pointed to by lots of other important pages
- This, of course, is circular…



The PageRank Algorithm

- Suppose p and q are pages where q → p
- Let R(q) be rank of q; let out(q) be out-degree of q
- Idea: q "distributes" its rank over its out-links
- Each out-link of q receives R(q)/out(q)
- So then p's rank should be:

$$R(p) = \sum_{q \in POINTS(p)} R(q) / out(q)$$

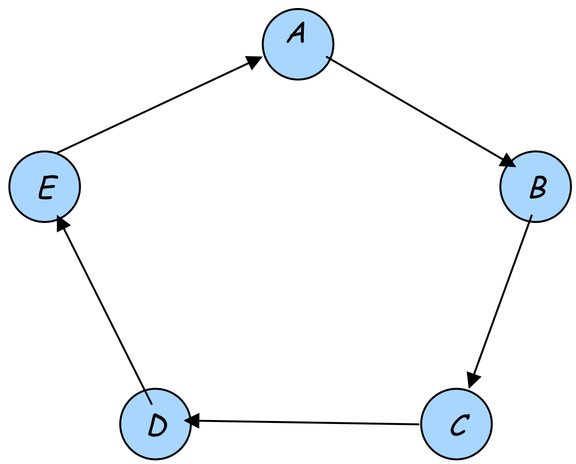


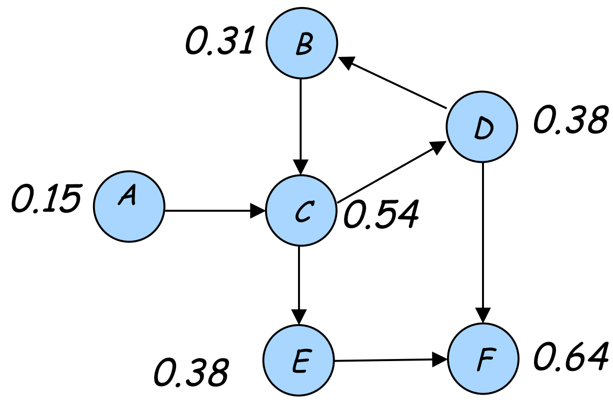
The PageRank Algorithm

- What guarantee do we have that all these equations are consistent?
- Idea: turn the equations into an update rule (algorithm)
- At each time step, pick some vertex p to update, and perform:

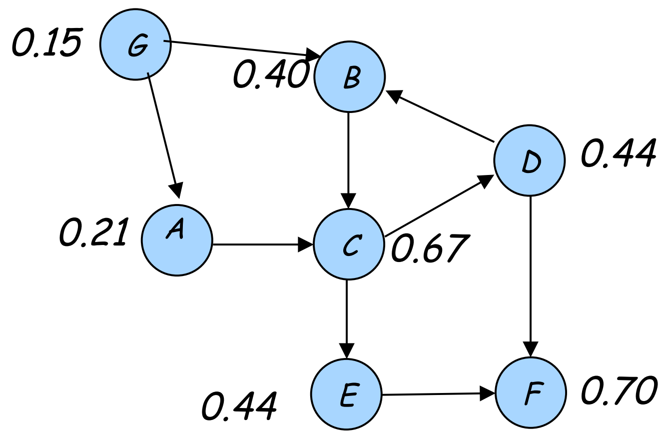
$$R(p) \leftarrow \sum_{q \in POINTS(p)} R(q) / out(q)$$

- Right-hand side R(q) are current/frozen values; R(p) is new rank of p
- Q: What if POINTS(p) is empty? A: "Random Surfer"
- Claim: Under broad conditions, this algorithm will converge:
  - at some point, updates no longer change any values
  - have found solution to all the R(p) equations
- Let's experiment with this PageRank Calculator


A
B
C
D
E



0.31
B
D
0.38
0.15
A
C
0.54
0.38
E
F
0.64


0.15
G
0.40
B
D
0.44
0.21
A
C
0.67
0.44
E
F
0.70

Summary

- PageRank defines importance in a circular or self-referential fashion
- Circularity is broken with a simple algorithm that provably converges
- PageRank defines R(p) globally; more subtle than local properties of p