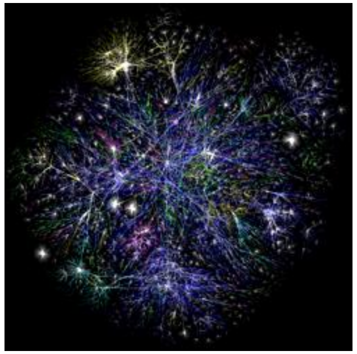
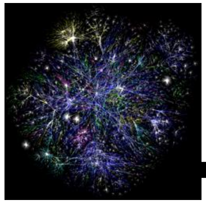
Information Networks

Graph Clustering

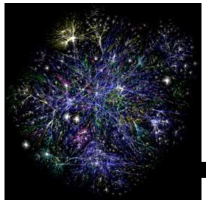
Lecture 14





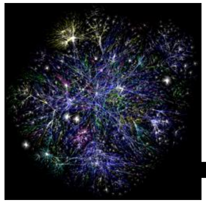
Clustering

- Given a set of objects V, and a notion of similarity (or distance) between them, partition the objects into disjoint sets $S_1, S_2, \ldots, S_k$, such that objects within the each set are similar, while objects across different sets are dissimilar



Graph Clustering

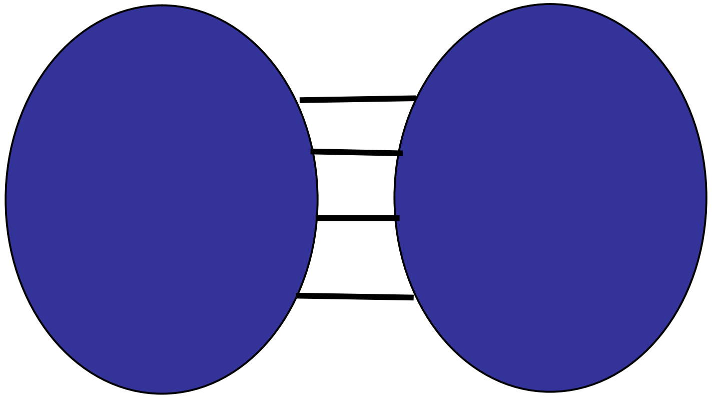
- Input: a graph G=(V,E)
  - edge (u,v) denotes similarity between u and v
  - weighted graphs: weight of edge captures the degree of similarity
- Clustering: Partition the nodes in the graph such that nodes within clusters are well interconnected (high edge weights), and nodes across clusters are sparsely interconnected (low edge weights)
  - most graph partitioning problems are NP hard



Measuring connectivity

- What does it mean that a set of nodes are well interconnected?

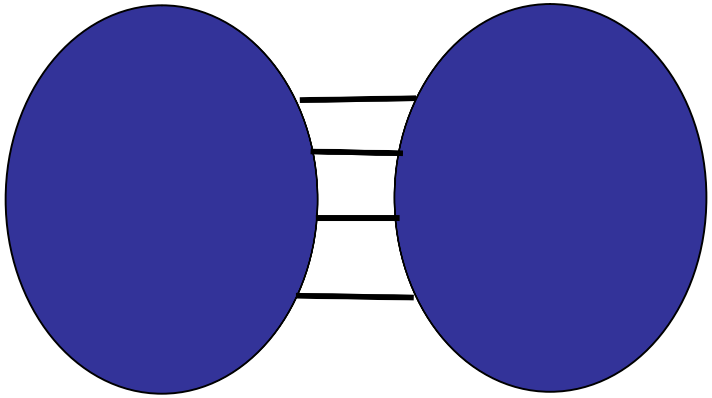
- min-cut: the min number of edges such that when removed cause the graph to become disconnected
  - large min-cut implies strong connectivity

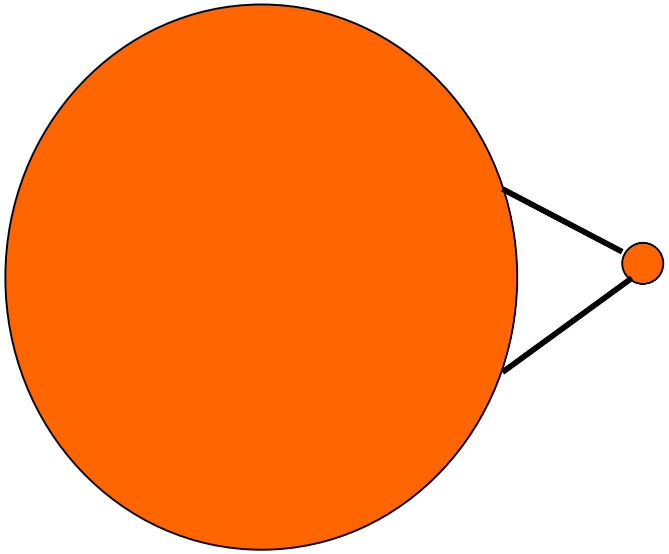


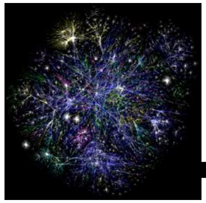
Measuring connectivity

- What does it mean that a set of nodes are well interconnected?

- min-cut: the min number of edges such that when removed cause the graph to become disconnected
  - not always true!





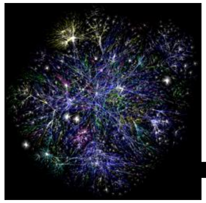


Graph expansion

- Normalize the cut by the size of the smallest component
- Graph expansion:

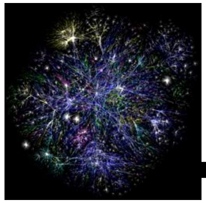
$$\alpha(G) = \min_{U} \frac{E(U, V - U)}{\min\{|U|, |V - U|\}}$$

- We will now see how the graph expansion relates to the eigenvalue of the adjanceny matrix A



Spectral analysis

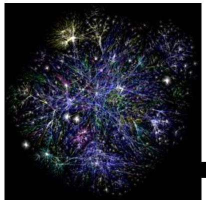
- The Laplacian matrix $L = D – A$ where
  - A = the adjacency matrix
  - $D = \mathrm{diag}(d_1,d_2,\ldots,d_n)$
    - d_i = degree of node i
- Therefore
  - $L(i,i) = d_i$
  - $L(i,j) = -1$, if there is an edge (i,j)



Laplacian Matrix properties

- The matrix L is symmetric and positive semi-definite
    - all eigenvalues of L are positive

- The matrix L has 0 as an eigenvalue, and corresponding eigenvector $w_1 = (1,1,\ldots,1)$
    - $\lambda_1 = 0$ is the smallest eigenvalue



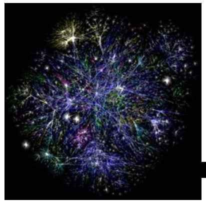
The second smallest eigenvalue

- The second smallest eigenvalue (also known as Fielder value) λ_2 satisfies

$$\lambda_2 = \min_{x \perp w_1, \|x\|=1} x^T L x$$

- The vector that minimizes λ_2 is called the Fielder vector. It minimizes

$$\lambda_2 = \min_{x \neq 0} \frac{\sum_{(i,j)\in E} \left(x_i - x_j\right)^2}{\sum_i x_i^2} \quad \text{where} \quad \sum_i x_i = 0$$

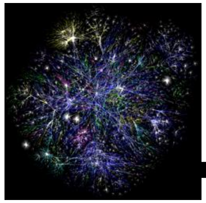


Fielder Value

- The value λ_2 is a good approximation of the graph expansion

$$\frac{\alpha^2}{2d} \le \lambda_2 \le 2\alpha$$

$$\frac{\lambda_2}{2} \le \alpha \le \sqrt{\lambda_2 (2d - \lambda_2)}$$



Spectral ordering

- The values of x minimize

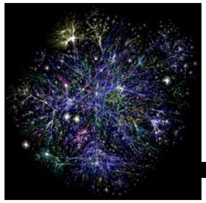
$$\min_{x \neq 0} \frac{\sum_{(i,j)\in E} \left(x_i - x_j\right)^2}{\sum_i x_i^2}$$

- For weighted matrices

$$\min_{x \neq 0} \frac{\sum_{(i,j)} A[i,j]\left(x_i - x_j\right)^2}{\sum_i x_i^2}$$

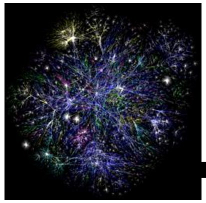
- The ordering according to the x_i values will group similar (connected) nodes together

- Physical interpretation: The stable state of springs placed on the edges of the graph



Spectral partition

- Partition the nodes according to the ordering induced by the Fielder vector
- If $u = (u_1, u_2, \ldots, u_n)$ is the Fielder vector, then split nodes according to a value s
  - bisection: s is the median value in u
  - ratio cut: s is the value that maximizes $\alpha(G)$
  - sign: separate positive and negative values ($s=0$)
  - gap: separate according to the largest gap in the values of u
- This works provably well for special cases



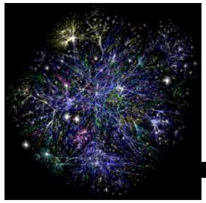
Conductance

- The nodes with high degree are more important
- Graph Conductance

$$\varphi(G) = \min_{U} \frac{E(U, V - U)}{\min\{d(U), d(V - U)\}}$$

- Conductance is related to the eigenvalue of the matrix $M = D^{-1}A$

$$\frac{\varphi^2}{8} \leq 1 - \mu_2 \leq \varphi$$

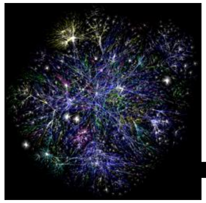


Clustering Conductance

- The conductance of a clustering is defined as the minimum conductance over all clusters in the clustering.

- Maximizing conductance seems like a natural choice

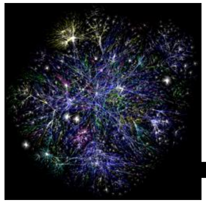
- …but it does not handle well outliers



A clustering bi-criterion

- Maximize the conductance, but at the same time minimize the inter-cluster edges

- A clustering $C = \{C_1,C_2,\ldots,C_n\}$ is a (c,e)-clustering if
  - The conductance of each C_i is at least c
  - The total number of inter-cluster edges is at most a fraction e of the total edges

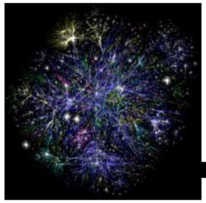


The clustering problem

- Problem 1: Given c, find a (c,e)-clustering that minimizes e

- Problem 2: Given e, find a (c,e)-clustering that maximizes c

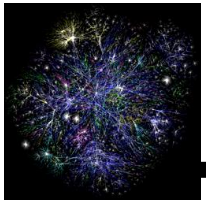
- The problems are NP-hard



A spectral algorithm

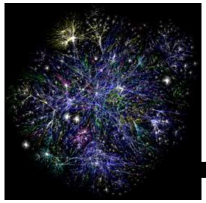
- Create matrix $M = D^{-1/2}A$
- Find the second largest eigenvector v
- Find the best ratio-cut (minimum conductance cut) with respect to v
- Recurse on the pieces induced by the cut.

- The algorithm has provable guarantees



Discovering communities

- Community: a set of nodes S, where the number of edges within the community is larger than the number of edges outside of the community.

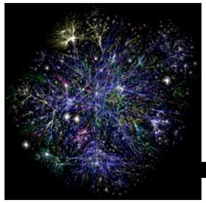


Min-cut Max-flow

- Given a graph G=(V,E), where each edge has some capacity c(u,v), a source node s, and a destination node t, find the maximum amount of flow that can be sent from s to t, without violating the capacity constraints

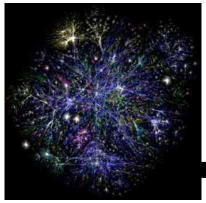
- The max-flow is equal to the min-cut in the graph (weighted min-cut)

- Solvable in polynomial time



A seeded community

- The community of node s with respect to node t, is the set of nodes reachable from s in the min-cut that contains s
  - this set defines a community



Discovering Web communities

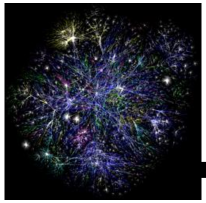
- Start with a set of seed nodes S
- Add a virtual source s
- Find neighbors a few links away
- Create a virtual sink t
- Find the community of s with respect to t

A more structured approach

- Add a virtual source t in the graph, and connect all nodes to t, with edges of capacity α
- Let S be the community of node s with respect to t. For every subset U of S we have

$$\frac{c(S,V-S)}{|V-S|} \le \alpha \le \frac{c(U,S-U)}{\min\{|U|,|S-U|\}}$$

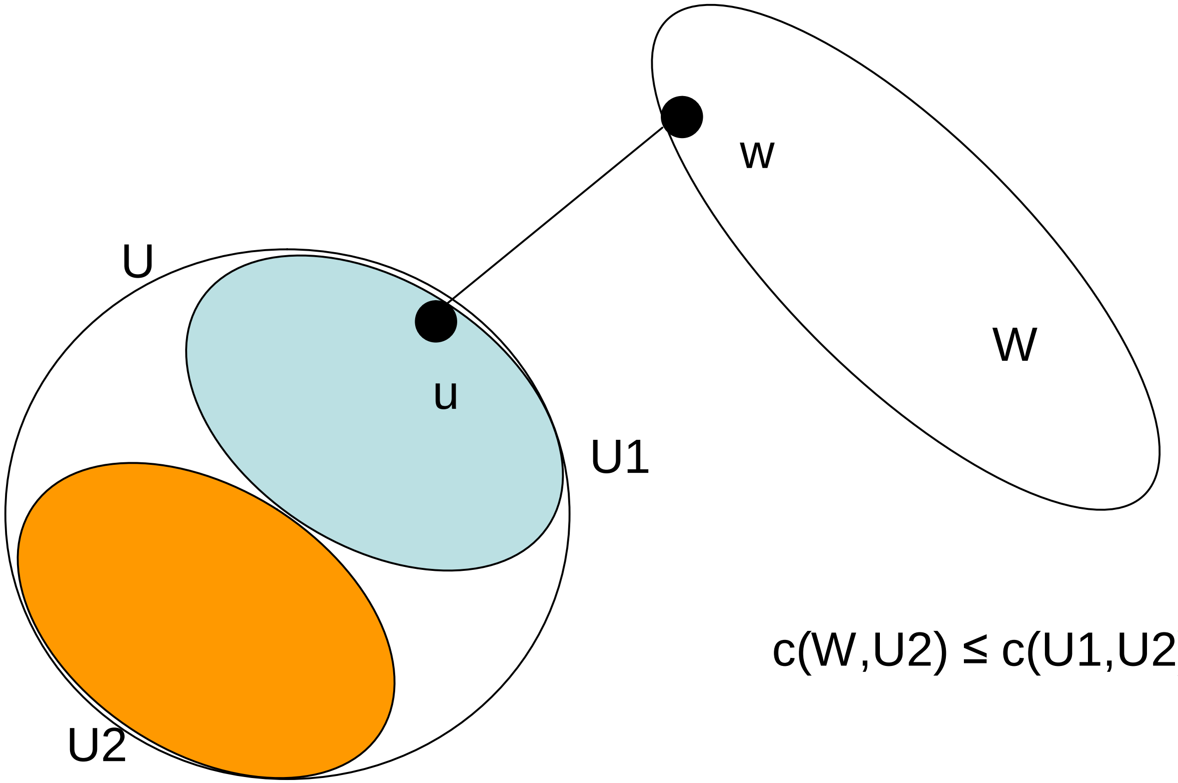
- Surprisingly, this simple algorithm gives guarantees for the expansion and the inter-community density



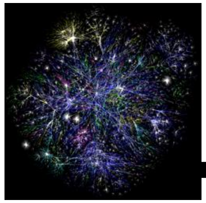
Min-Cut Trees

- Given a graph G=(V,E), the min-cut tree T for graph G is defined as a tree over the set of vertices V, where
  - the edges are weighted
  - the min-cut between nodes u and v is the smallest weight among the edges in the path from u to v.
  - removing this edge from T gives the same partition as removing the min-cut in G

Lemma 1



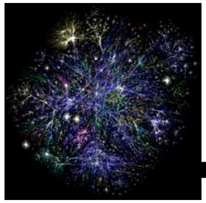
c(W,U2) ≤ c(U1,U2)



Lemma 2

- Let S be the community of the node s with respect to the artificial sink t. For any subset U of S we have

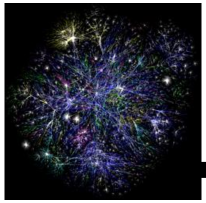
$$\alpha \leq \frac{c(U, S-U)}{\min\{|U|, |S-U|\}}$$



Lemma 3

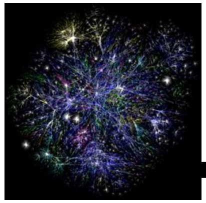
- Let S be the community of node s with respect to t. Then we have

$$\frac{c(S, V-S)}{|V-S|} \leq \alpha$$



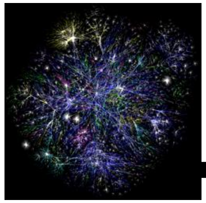
Algorithm for finding communities

- Add a virtual sink t to the graph G and connect all nodes with capacity α à graph G’
- Create the min-cut tree T’ of graph G’
- Remove t from T’
- Return the disconnected components as clusters



Effect of α

- When α is too small, the algorithm returns a single cluster (the easy thing to do is to remove the sink t)
- When α is too large, the algorithm returns singletons
- In between is the interesting area.
- We can explore for the right value of α
- We can run the algorithm hierarchically
  - start with small α and increase it gradually
  - the clusters returned are nested



References

- J. Kleinberg. Lecture notes on spectral clustering
- Daniel A. Spielman and Shang-Hua Teng. Spectral Partitioning Works: Planar graphs and finite element meshes. Proceedings of the 37th Annual IEEE Conference on Foundations of Computer Science, 1996. and UC Berkeley Technical Report number UCB CSD-96-898.
- Ravi Kannan, Santos Vempala, Adrian Vetta, On clusterings: good, bad and spectral. Journal of the ACM (JACM) 51(3), 497--515, 2004.
- Gary Flake, Steve Lawrence, C. Lee Giles, Efficient identification of Web Communities, SIGKDD 2000
- G.W. Flake, K. Tsioutsiouliklis, R.E. Tarjan, Graph Clustering Techniques based on Minimum Cut Trees, Technical Report 2002-06, NEC, Princeton, NJ, 2002. (click here for the version that appeared in Internet Mathematics)