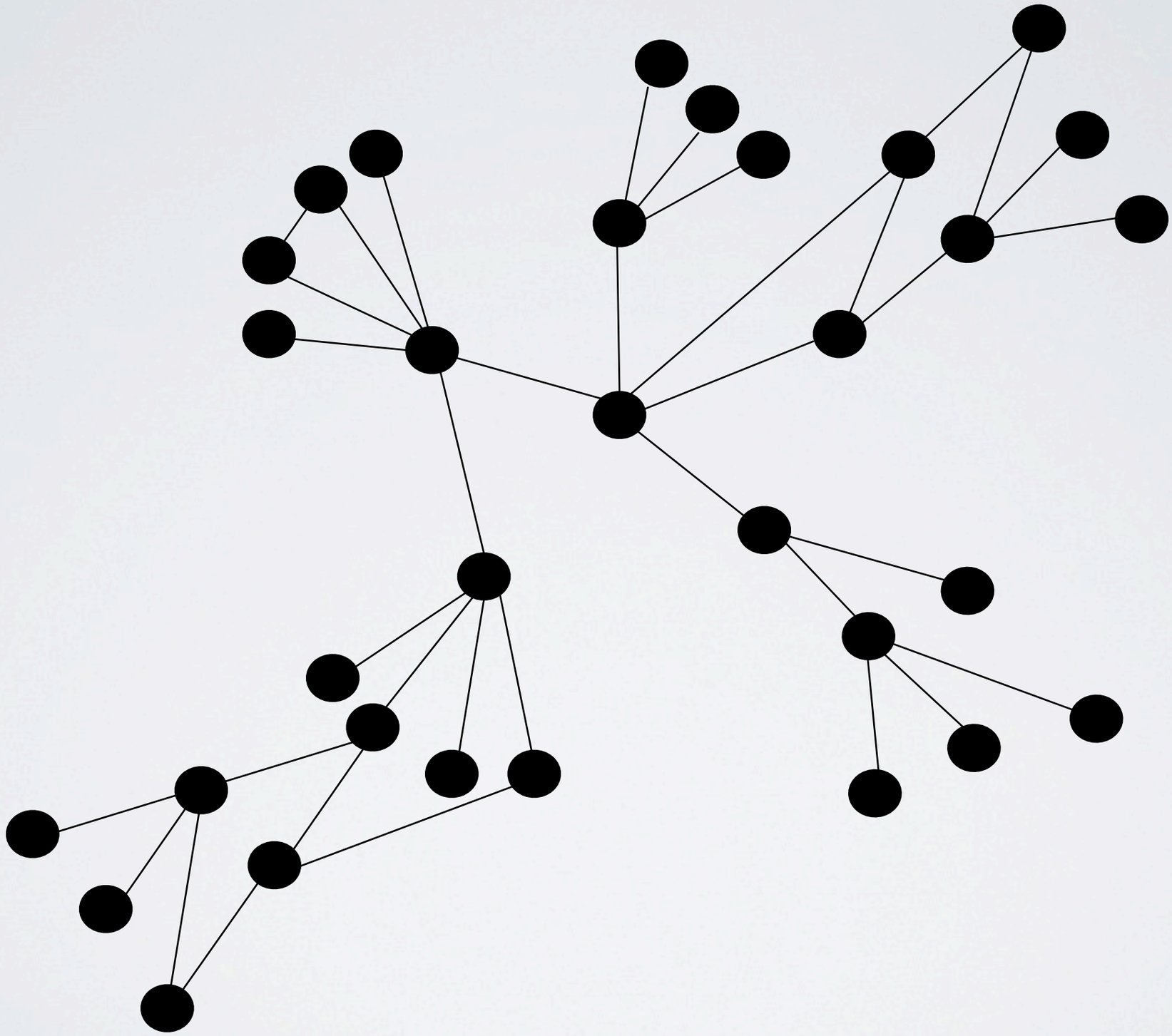
MODULARITY CLUSTERING FOR COMMUNITY DETECTION

Yael Elmatad, ph.d. Data Scientist @ Tapad, Inc.
@y_s_e, yael@tapad.com
github.com/yaelelmatad

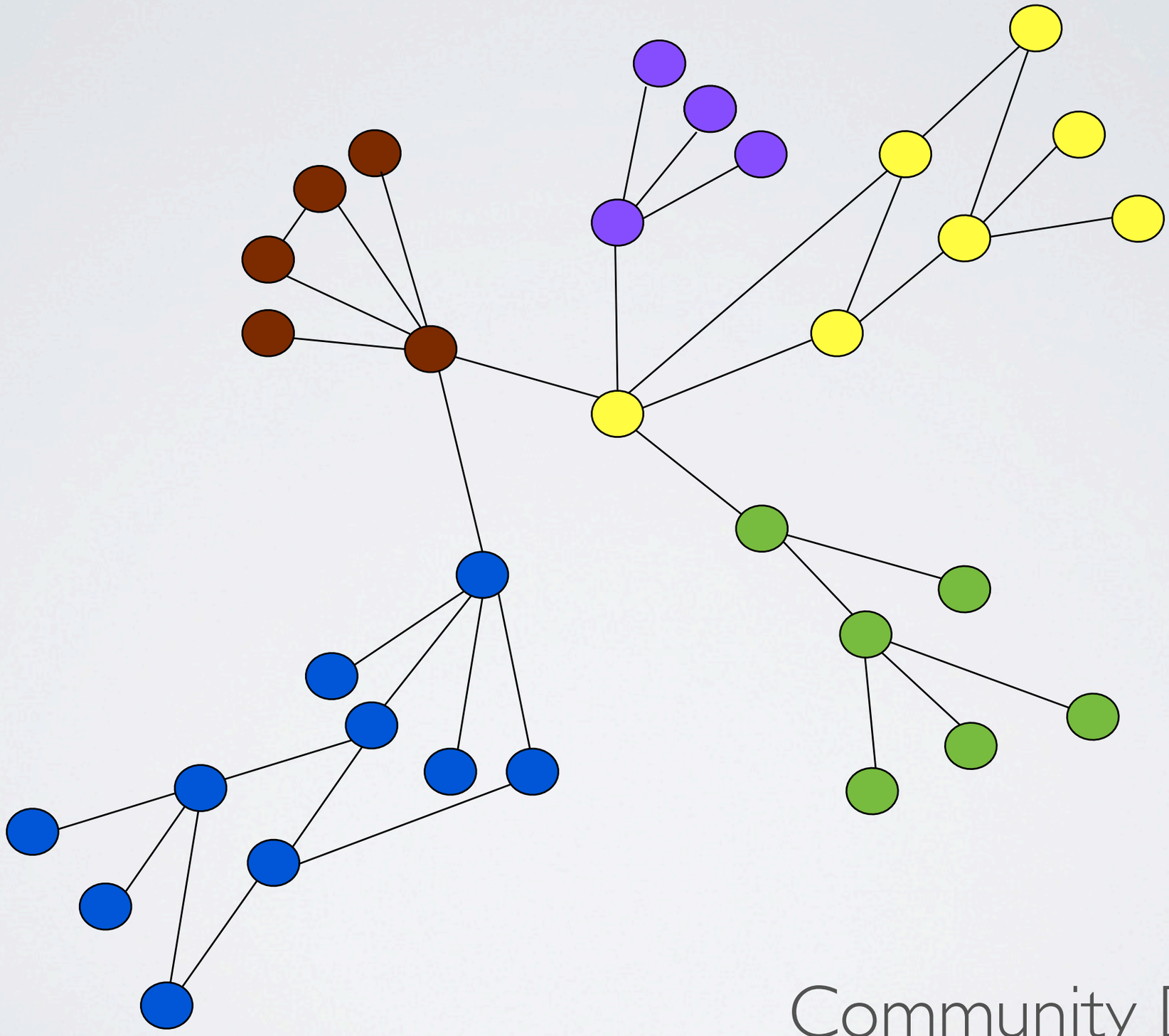
OUTLINE

- Graph Clustering Problem
- Description of Modularity Clustering
- Walkthrough of Code

A TYPICAL, SMALL GRAPH



DESIRE: TO CLUSTER THE GRAPH



Community Detection

GRAPH MODULARITY

Based on Newman, *Phys. Rev. E,* **69**, 066133 (2004)
"Fast algorithm for detecting community structure in networks"

Goal: Maximize the number of meaningful links *within* a community while minimizing the number of meaningful links *between* communities.

WHAT IS MODULARITY (Q)

e_{ij} = the fraction of the edges that connect community i to community j

Define quantity:

$$a_i = \sum_j e_{ij}$$

Which is now the fraction of edges that connect community i to all other communities (including itself)

We define a quantity Q - the modularity - as:

$$Q = \sum_i (e_{ii} - a_i^2)$$

This is the fraction of edges within community i minus the number of edges we would expect for a totally random graph.

MODULARITY

The more random-like the partitioning of the graph is the smaller the value of Q is.

If a cluster has real, meaningful community structure, we expect that Q should be large. If the graph is just random edges we expect no meaningful correlations and a value of Q that is very close to 0.

The goal of modularity clustering it find the partitioning of the graph which produces the largest value of Q.

Ideal values are usually where $Q > 0.3$

MODULARITY CLUSTERING (ALGORITHM)

--Greedy Algorithm [Finds a locally optimum solution]

Initialization: initialize each node in the graph to its own community (cluster) and calculate e's, a's, an Qs.

Iteration: Each iteration seeks to reduce the number of communities by one by combining two communities. Choose to merge the two communities which give the largest change in Q (change can be negative!). Recompute e's and a's between the new cluster and all other clusters.

Completion: When all the nodes are in a single cluster emit the value of Q which was largest along the path.

CODE DEMO TIME!

(show code now + karate club example)

Some speed ups/improvements:

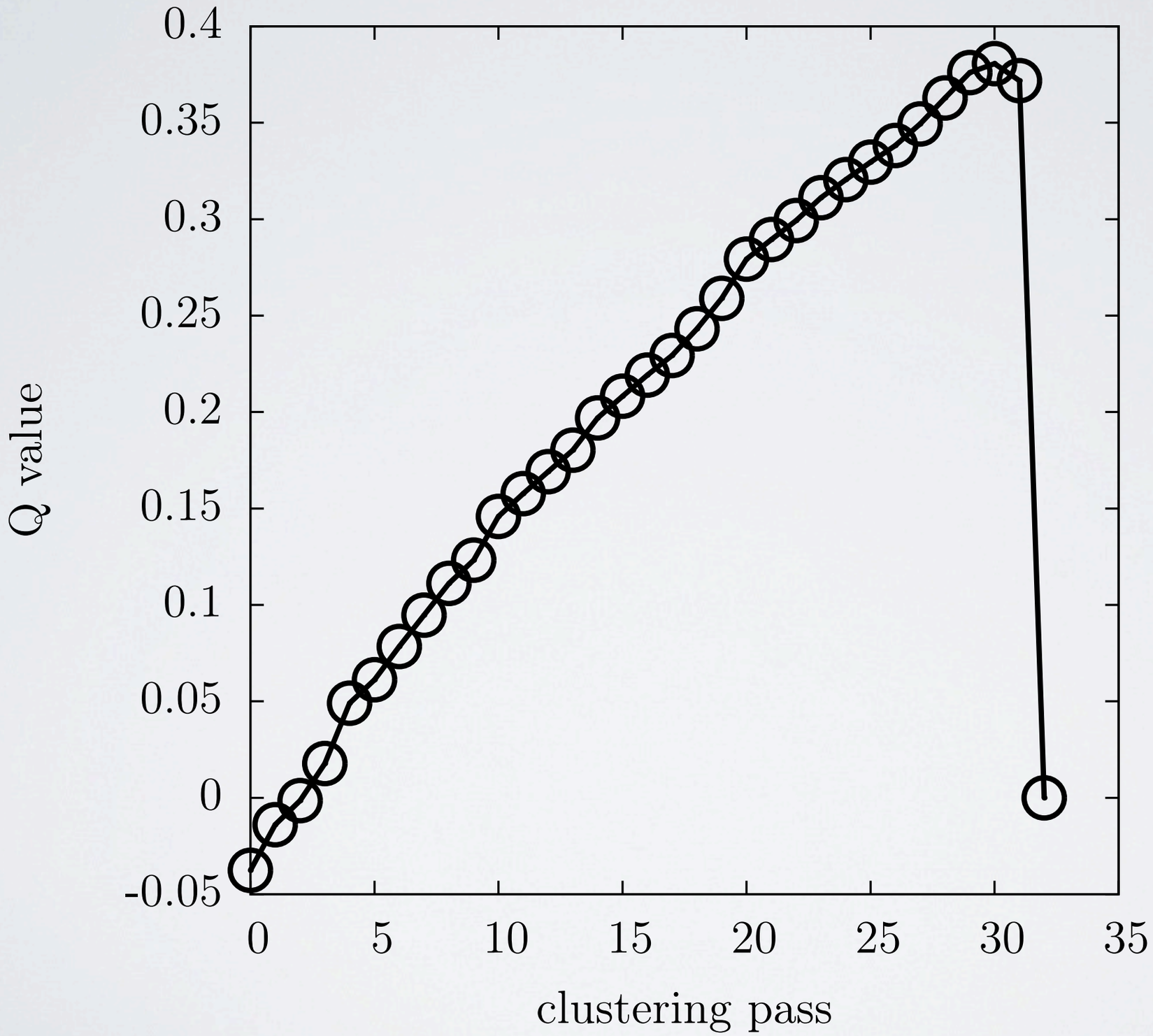
+ Only update connections (e's/a's) from/to last merged cluster

+ Only update ΔQ values from/to last merged cluster

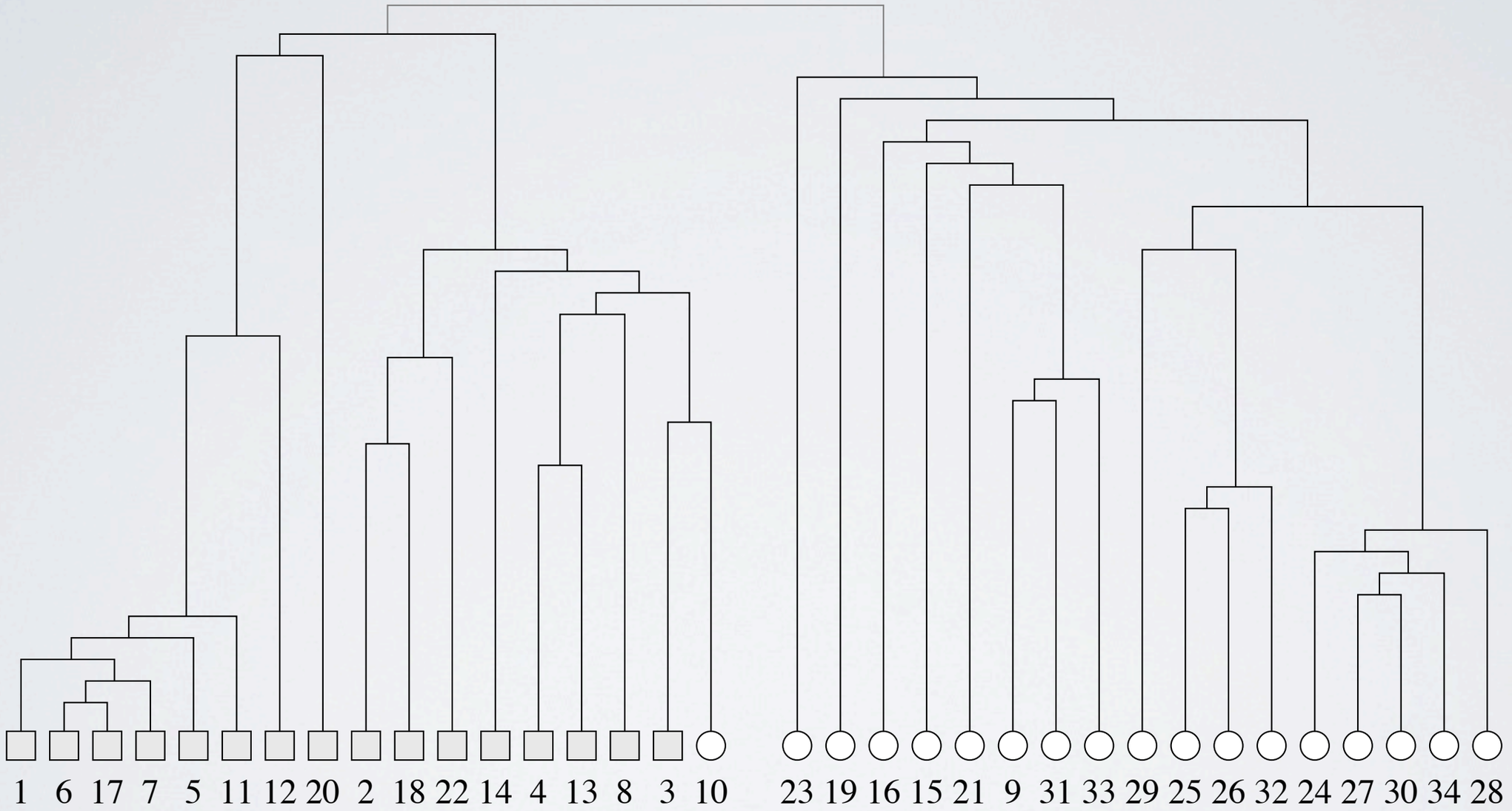
+ Remove from consideration clusters with no outbound links (disconnected components/subgraphs)

+ Allow for weighted edges

OUTCOME: Q PATH



OUTCOME: HIERARCHICAL CLUSTERING



"Karate Club" Data -- Shapes = Split after internal dispute of club.

From: Newman, *Phys. Rev. E,* **69**, 066133 (2004)

VISUALIZING CLUSTERS

Community detected clusters represented by the three colors. Graph generated by d3 force directed graph (knows nothing about communities).

LIMITATIONS

Because modularity seeks to group things together that have links with higher rate than whole graph, if the graph has very few links (islands/sparse) then the algorithm will tend towards locating all the disconnected components - and not the communities WITHIN those disconnected components.

Workaround: Find all connected components (self-contained subgraphs) and run modularity clustering on those subgraphs.

6-degrees-of-CELEBRITY-NAME like graphs = Good
Graphs with lots of islands = Bad