

The Open Source Software Community Structure

The background of the slide features a large, faded seal of the University of Notre Dame. The seal is circular and contains the Latin text "UNIVERSITAS DOMINICAE" at the top and "NOTRE DAME" at the bottom. In the center, there is a shield with a cross and other symbols.

Jin Xu, Scott Christley & Gregory Madey
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556

Supported by NSF, CISE/IIS-Digital Science and
Technology

NAACSOS, Notre Dame
June 2005



COMMUNITY STRUCTURE

- In most networks, some nodes are grouped together by a high density of edges, while there are few edges between those groups
- Communities: tightly-connected groups (Newman & Girvan 2004)
- Applications:
 - Scientists grouped together by similar research topics or methodology (Givan & Newman 2002)
 - Functionally related genes in gene networks (Wilkinson & Huberman 2004)
 - Actual social relationships in email networks (Tyler 2003)



INFLUENCE IN OSS NETWORK

- Identify projects which might have related subjects, similar programming environment, or common developers.
- Study projects interaction during their growth.
- Get information about the communication path and knowledge flow within or between communities. Such information can help us adjust and improve the robustness of communications in OSS



PREVIOUS RESEARCH

- Edge betweenness (Girvan & Newman 2002)
 - Top-down, remove edges with the highest betweenness
 - $O(m^2n)$ time on a network with m edges and n vertices
 - $O(n^3)$ on a sparse graph
- Greedy Algorithm (Newman 2004)
 - Bottom-up, two communities are picked to join
 - $O((m+n)n)$ time on a random network
 - $O(n^2)$ on a sparse graph
- Implementation of more sophisticated data structure (Clauset & Newman 2004)
 - $O(n \log(n^2))$



CLAUSET ALGORITHM

- Modularity Q : the fraction of edges within communities subtracts the expected value of the same quantity if edges fall in a random network
- The best community structure is where Q is the largest

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

$$a_i = \frac{k_i}{\sum e_{ij}}$$

e_{ij} is the fraction of edges that connect nodes i and j

a_i is the fraction of edges that connect to node i in a random network
the total edges in the network

k_i is the number of edges connecting to group i

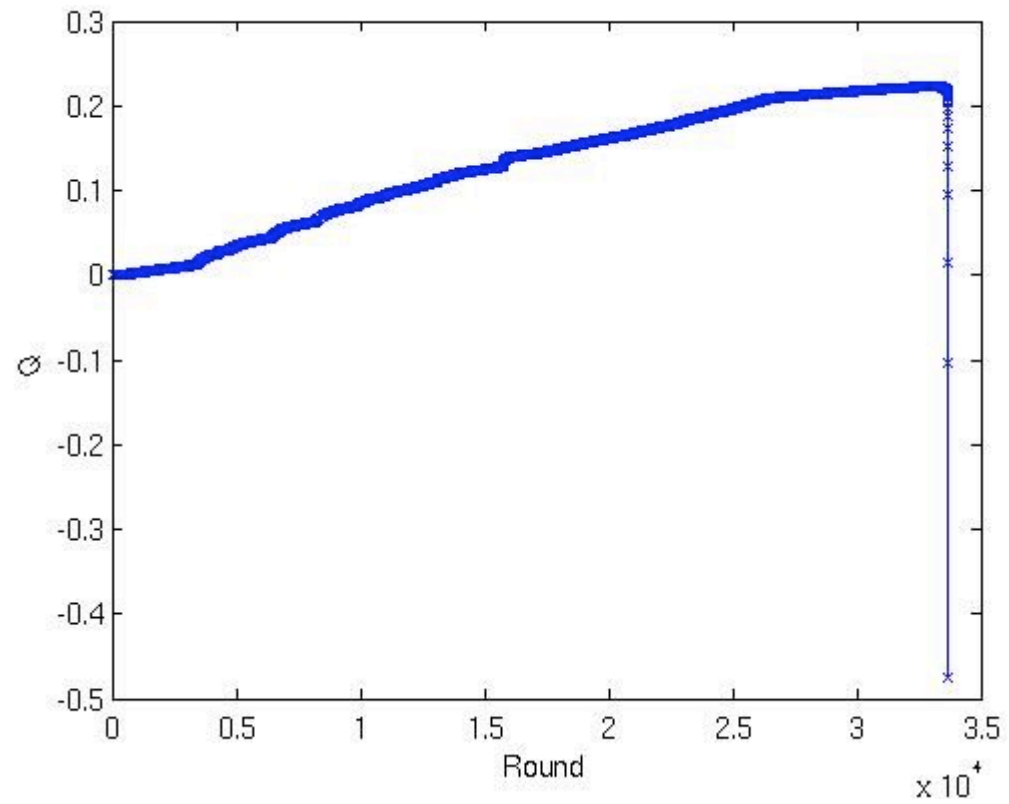


RESULTS

- Project network: 2 projects are connected if they have 1 or more common developers
- The largest component of the project network in Jan. 2003.
- Exclude project 1 which is the Sourceforge because it links to 10,000 projects
- The project network consists of 27,834 nodes and 173,644 edges



THE VALUE OF Q





ANALYSIS

- Max $Q = 0.2227$
- 611 groups
- The largest group consists of 3467 projects
- Many small communities with size less than 10
- The 10 largest groups include 64.8% of the whole projects
- The important communication paths are those connections between communities
- Common developers are keys to transfer information between two project groups



OSS ASSORTATIVE MIXING

- Several explanations exist for the community structure
- Mutual acquaintance: two nodes with a common neighbor are more likely to link to each other
- Homophily: two nodes with the same attributes are more likely to link to each other
 - Measure: assortative mixing



ASSORTATIVE MIXING

- Assortative coefficient

$$r = \frac{\sum_i e_{ii} - \sum_i a_i b_i}{1 - \sum_i a_i b_i}$$

e_{ij} is the fraction of edges in a network w of type i to nodes of type j

b_j is the fraction of edges of each type w with nodes of type i



ASSORTATIVE MIXING FOR OSS

- Topic – 0.1009
- Operating System – 0.1078
- User interface – 0.0893
- Development status – 0.0553
- Intended audience – 0.0449
- Programming language – 0.1541



CONCLUSION

- Community structure exists among Sourceforge project network
- Key communication paths are identified among groups
- Projects with the same programming languages, operating systems and topics are more likely to group together.