

An Algorithm for Temporal Analysis of Social Positions

Scott Christley, Greg Madey
Dept. of Computer Science and
Engineering
University of Notre Dame

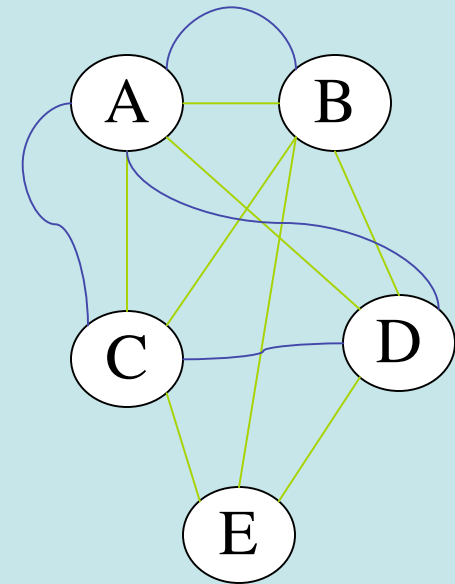
*Supported in part by National Science Foundation, CISE/IIS-Digital Society &
Technology, under Grant No. 0222829*

Motivation

- Successful software development requires various positions to be filled; developers, testers, administrators, management, end-users, etc.
- People in Open Source Software community self-select into a social position on a software project.
- We don't know what these positions are; emerged from the self-organization of the community.
- Do people stay in same social position, or does there position change over time?
- Positional analysis seeks to group actors into disjoint subsets according to their social position in the network.

Structural Equivalence

- Actors who are similarly embedded occupy similar social position.
- $C \sim D$ have same relationships with same other actors.
- Exact equivalence is too strict so use an approximate measure, like Euclidean distance.
- Weighted relationships



Clustering

- Standard data mining algorithms
 - K-means, Expectation-Minimization (EM)
- What's wrong with Euclidean distance?
 - Data mapped to points in an N-dimensional space.
 - Points “close” in space are in same cluster.
 - Normalization techniques very important.
 - Not comparing the underlying distributions.
- Assume Gaussian (normal) distribution
- What can we use instead of a distance metric?
 - Statistical test

Clustering with a Statistical Test

- Fisher's contingency-table test (non-parametric)
 - Chi-square family of goodness-of-fit tests
- Given two independent samples
 - First sample, S_1 , with n_1 random variables
 - Second sample, S_2 , with n_2 random variables
 - Where n_1 not necessarily equal to n_2 , each r.v. in each samples placed in one of C categories.
- H_0 : The distributions of S_1 and S_2 do not differ.
- H_A : The distributions S_1 and S_2 differ.
- Structural In-equivalence

Algorithm (Intersection)

While (still unclustered samples)

Put all unclustered samples into one cluster.

While (some samples not yet pairwise compared)

A = Pick sample from cluster

For each other sample, B, in cluster

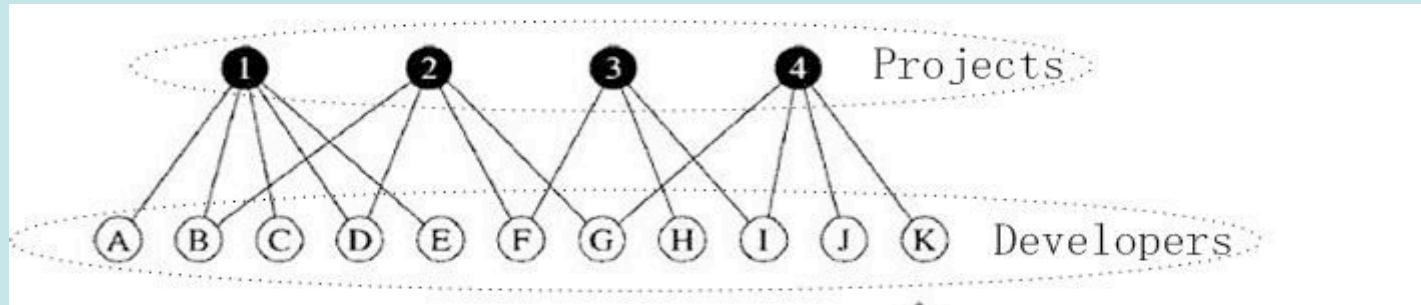
Run statistical test on A and B.

If significant result

Remove B from cluster.

- Rejection of null hypothesis means A and B **must** be in different clusters.
- Confidence level tightens/broadens cluster inclusion.
- Any statistical test for a two-sided test problem.

OSS Activity



- User performs an activity for a project.
- 21 activities; submit bug, submit feature request, assign bug, post forum message, create file release, create project task, etc.
- Multi-relational, weighted, bipartite network.
 - Activity = relation, weight = activity count
- Activity distribution for user/project pair defines a sample for our statistical test.
- That is, the activity a user performs on a project defines their social position for that project.

Social Positions of OSS

Social Position	Size	# of clusters
Brief Flame	122654	1
Message Posting	50067	4
Task Management	2762	5
Release Management	6509	5
Documentation	1266	4
Job Posting	899	2
Artifact Management	1674	6
Administrators	10377	4
Not Categorized	13786	1546
Total User/Project Pairs	209994	

Temporal Analysis

- Previous analysis, activity over 10 years, lose knowledge of evolution of positions.
- How to deal with time (data)?
 - Global time; snapshot of the whole network at points in time: node/edge add/remove, attribute change, tends to get aggregate measures.
 - Local time; user/project's first activity is time 0, aligns actors in a time-relative way to the network, egocentric viewpoint.
- Chunk data into monthly activity, run clustering algorithm for data for each time period.

Temporal Social Positions of OSS

Social Position	Period 1	Period 2	Period 3	Period 4
Brief Flame	127302	0	0	0
Message Posting	49754	1418	828	151
Administrators	10356	5415	905	496
Release Management	6304	1001	796	869
Task Management	3466	625	254	401
Artifact Management	1967	0	0	0
Documentation	1130	0	0	0
Job Posting	1125	0	0	0
Not Categorized	4904	2002	1313	1105
Handyperson		7282	8280	6664
Total User/Project Pairs	206308	17743	12376	9686
Total Clusters	397	183	143	139

Summary

- Clustering algorithm using a statistical test.
 - Don't have to specify # of clusters a priori.
 - No assumption of underlying distribution.
 - Must be appropriate statistical test.
- Temporal Analysis
 - How you organize/view your data is important.
 - Global metrics --> global time
 - Egocentric measures --> local time

Iterative Classification

- Order of comparison matters.
- Clustering is NP-complete so intractable to check all combinations to find the optimal.
- Iterative approach
 - Perform initial clustering
 - Calculate cluster center