Communities in link streams

1 Context

In many contexts, data comes in the form of entities interacting over time. For example, individuals meeting or exchanging phone calls, email exchanges, machines exchanging IP packets, Wikipedia edits, users rating items over time, etc. Real-world examples span a variety of contexts, and typically comprise of thousands to millions of interactions. These interactions can be directed, weighted, labeled, etc. however the principle remains identical.

A link stream is a tuple $L = (T, V, E)$, where $T = [\alpha, \omega]$ is a time interval, V is a set of nodes, and $E \subseteq T \times V \otimes V$.

In graphs, community detection consists in labeling nodes in a way that similar nodes [1] have the same label. Communities are *nodes that are densely connected together, but loosely connected otherwise*, and as such are markers of social groups, thematically similar pages on the Web, among others. In dynamic graphs, methods typically follow the evolution of these groups over time.

However, communities in link streams carry a different meaning: groups of nodes interacting closely over time. They are markers of meetings among people, attacks in IP networks, threads of discussion in email networks, etc. There is, to date, no method suited to community detection in link streams. In this context, it is important to develop new methods.

Toward this goal, different research directions can be considered. The first one is to extend Girvan-Newman's algorithm [], relying on the notion of betweenness centrality to link streams. Betweenness centrality has been extended to link streams [], and evaluates the importance of a node v at time t by assessing the fraction of shortest fastest paths from any node to any other that go through (t, v). It is computable in polynomial time in link streams.

The second one considers the optimization problem of finding the maximal flow in link streams. This problem, given a weighted link stream, a source (t, u) and a target (t', v), finds the maximal amount of flow units that can be transferred from the source to the target.

Finally, a third approach considers the notion of *quasi-cliques*, *i.e.* sub-streams where nearly every node interacts with every other. Multiple algorithms enumerate the maximal cliques of a link stream [], however proper definition and enumeration of quasi-cliques remains challenging.

The goal of this internship is to explore one or more of these models for community detection in real-world link streams.

[1] Where *similar* is context-dependent.

Dr. Tiphaine Viard
tiphaine.viard@riken.jp
Riken Advanced Intelligence Project
Discrete Optimization Unit
1-3-1 Nihonbashi, Chuo-ku, Tokyo

2 Skills

Depending on the applicant’s skills and interests, emphasis can be put on fundamental, algorithmic or experimental aspects to some extent. It is not expected that all directions of research are explored.

Programming knowledge, preferably in Python, is required. Knowledge of some graph theory, community detection methods, algorithms proof methods, will be appreciated, but not required. Time will be dedicated for proper formation up to the state-of-the-art where needed.

3 Internship context and allowance

This internship will last up to 3 months, starting in May or June 2019, and will take place at the Riken Advanced Intelligence Project’s main office, in Nihonbashi.

An allowance of 2000 yens per work day will be provided, as well as up to 3500 yens per day to cover housing expenses. Commuting fees will be fully paid, as well as a round-trip plane ticket to Tokyo.

4 Contact

Please contact `tiphaine.viard@riken.jp` with a resume and a brief email stating your motivations.