The Journal of Open Source Software

Local clustering

Peter Volf[1]

1 None

DOI: 10.21105/joss.00954

Software

- Review
- Repository
- Archive

Submitted: 16 September 2018
Published: 17 September 2018

Introduction

Graph clustering algorithms aim to divide the nodes of a graph into one or more groups or clusters based on some measure of similarity. There are many different ways to define a cluster depending on the goal of the analysis. In graph cluster analysis the usual definition of a cluster is a group of nodes that are relatively densely interconnected but have few connections towards the nodes outside the cluster.

For an overview of graph clustering see (Schaeffer, 2007). The more developed field within graph clustering is global clustering. Global algorithms are allowed to look at the whole graph while calculating its clusters. Well known global clustering algorithms include (Blondel, 2008), (Dongen, 2000), (G. Newman M., 2004), (M. Newman, 2006), (Pons, 2006) and (Raghavan, 2007).

On the other hand, local clustering algorithms get a small set of source nodes (typically a single node) as input and calculate the cluster they belong to in the graph. While doing so, local algorithms are only allowed to look at the already visited nodes of the graph and their neighbours.

Algorithm

This Python package implements the Hermina-Janos local clustering algorithm and its hierarchical variation. The algorithms are independent of the used cluster definition, instead they define a set of requirements cluster definitions must fulfill in order to work with the algorithms. The package provides one such cluster definition.

Local clustering algorithm

The Hermina-Janos algorithm is a simple iterative process that repeats the following two steps until it reaches a stable state:

1. Expansion step: For each node in the neighbourhood of the cluster, decide whether adding it to the cluster would increase the cluster's quality, collect all the neighbours whose addition would improve the cluster and add them to the cluster in one step.
2. Reduction step: For each node on the border of the cluster, decide whether removing it from the cluster would increase the cluster's quality, collect all the nodes whose removal would improve the cluster and remove them from the cluster in one step.

The process can be thought of as a sort of label propagation (Raghavan, 2007), where the cluster competes against its surroundings until an equilibrium is reached.

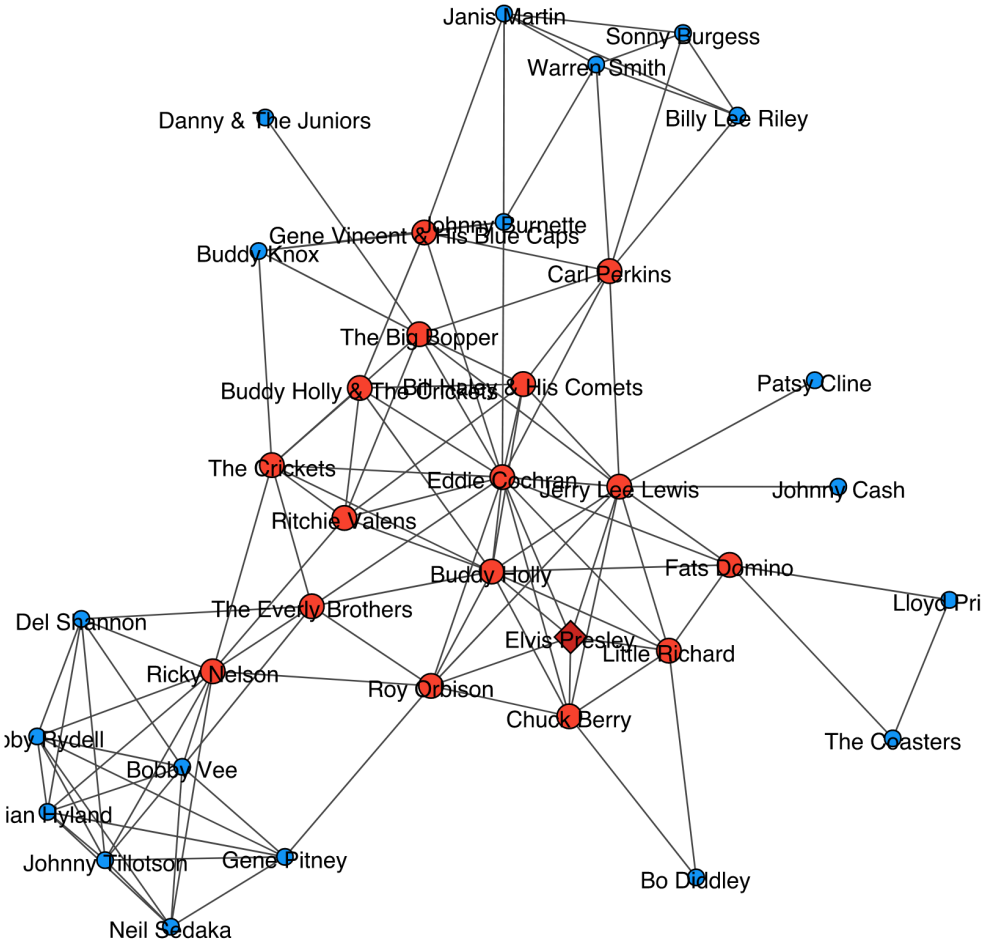

Figure 1: The cluster of Elvis Presley in Spotify's Related Artists graph.

Hierarchical local clustering algorithm

The hierarchical version of the Hermina-Janos local clustering algorithm extends the base version with an extra layer that allows it uncover clusters at different hierarchy levels or "distances" from the source nodes of the cluster analysis.

Similarly to the base algorithm, the hierarchical Hermina-Janos algorithm is also an iterative process with the following two steps:

1. Local clustering step: use the Hermina-Janos local clustering algorithm with the current configuration of the used cluster definition to calculate the cluster.
2. Cluster definition relaxation step: this is a highly cluster definition-dependent step where the algorithm adjusts or relaxes the cluster definition's parameters so in the next iteration the local clustering algorithm will be able to further extend the cluster.

Tools and utilities

The package provides a ranking component that can be used to rank the nodes in the cluster and their neighbours by their importance or contribution to the cluster.

A component for recording the steps the algorithms have taken is also provided. It makes it possible to trace back each decision and step the algorithms have taken to see exactly how the result was calculated.

Ideas for future work

Here are some ideas for future work:

- Reimplementation for parallel computing: Most of the calculations the algorithms make (the only exception being the actual cluster update) can be executed in parallel, that could significantly improve performace.
- New cluster definitions: Only one cluster definition is provided in the package. More cluster definitions can be implemented for example by building on cluster quality metrics such as modularity (G. Newman M., 2004).
- Analysis of how cluster definitions should be configured for graphs with different characteristics.
- Result comparison with global clustering algorithms on well-known and -analyzed graphs such as the Zachary karate club (Zachary, 1977).

Resources

These are the main resources besides the source code:

- This document provides a thorough description of the algorithms and the included cluster definition.
- This notebook provides a demo and in-depth evaluation of the algorithms and the ranking component using Spotify's Related Artists graph.

References

Blondel, G., V. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*, P10008.

Dongen, S. van. (2000). A cluster algorithm for graphs. *Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam*, *Technical Report INS-R0010*.

Newman, G., M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, *69*, 026113.

Newman, M. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, *74*, 036104.

Pons, L., P. (2006). Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, *10*, 191–218.

Raghavan, A., U. N. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, *76*, 036106.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, *1*, 27–64.

Zachary, W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, *33*, 452–473.