Evaluating Clustering Techniques
An Engineering Approach Inspired by Unit-Tests

Universität Karlsruhe (TH)

Summary

The quality of clustering algorithms is often based on their performance according to a specific quality index, in an experimental evaluation. Experiments either use a limited number of real-world instances or synthetic data. While real-world data is crucial for testing such algorithms, it is scarcely available and thus insufficient. Therefore, synthetic pre-clustered data has to be assembled as a test bed by a generator. Evaluating clustering techniques on the basis of synthetic data is highly non trivial. Even worse, we reveal several hidden dependencies between algorithms, indices, and generators that potentially lead to counterintuitive results. In order to cope with these dependencies, we present a framework for testing based on the concept of unit-tests. Moreover, we show the feasibility and the advantages of our approach in an experimental evaluation.

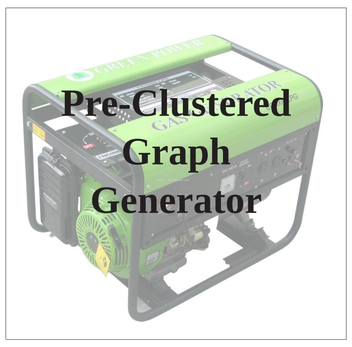
Current Status

The experimental evaluation of clustering techniques has to successfully overcome two major issues: On the one hand, artifacts are inherent in most techniques such as generation of data, the algorithms themselves, or the quality measurements. On the other hand, these technqiues are not independent of each other and, thus, can heavily influence the evaluation.

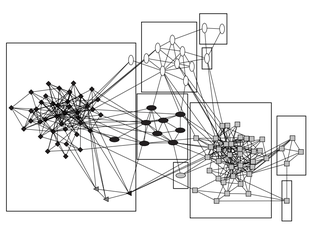
Artifacts – General Rule of Thumbs

‘Every clustering technique suffers some drawbacks’

- each technique uses its own formalization of the ‘perfect clustering’
- even simple paradigms such as intra-cluster density versus inter-cluster sparsity are diversely interpreted
- generators explicitly construct/avoid special (sub-)structures such as cliques, satellites, tree-like appendicies, significant bottlenecks, etc.
- quality measures and clustering algorithms identify/search only for special (sub-)structures
- degrees of freedom, modeled by parameters, often has to be adjusted when the size of the input data changes
- many algorithms have been highly tuned for special tasks and specific applicational requirements



- generates random graphs
- implants pre-clustering
- strentgh of clustering tunable
- parameterized
- efficient, batch-computable
- based on $\mathcal{G}(n,p)$, Voronoi diagrams, etc.



severe dependency

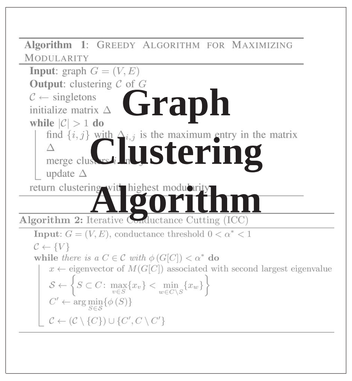
severe dependency

Circular Dependencies

As indicated in the left-hand diagram, the clustering techniques for generators, clustering algorithms, and quality indices are interdependent. More precisely, on a very schematic level, the three techniques can be summarizes as: A clustering algorithm assigns to the each graph G and significance threshold τ a clustering $\mathcal{C}$ which has a significance score larger or equal to τ; a quality index maps a pair consisting of a graph G and a clustering $\mathcal{C}$ to a significance score τ; a pre-clustered graph generators assigns to each clustering $\mathcal{C}$ and significance score τ a graph G such that $\mathcal{C}$ has at least significance τ with respect to G.

Although, this is a simplified model, it reveals the inherent dependencies fairly well. For example, if the quality measure and the clustering algorithm are founded on the same idea, the algorithm will always exhibit good results. While evaluating a algorithm for dense graphs with a generator for comparable sparse graphs will usually produce random results.

- assess quality of a clustering
- based on:
  - counting pairs
  - bottlenecks
  - degree of randomness
- usually normalized to $[0,1]$
- sensitive to specific structures



- community structure in graphs
- large-scale inhomogeneities
- approaches:
  - iterative cutting
  - greedy agglomeration
  - spectral division
  - graph percolation
- sometimes parameterized

severe dependency

Examples of Pitfalls:

- greedy optimization of an index measures only with the index
- evaluation of geometric algorithms based on (too) simple geometric structures
- ignoring density constraints, e. g., feeding a clique partitioner with a planar graph

Remedy: Unit Tests

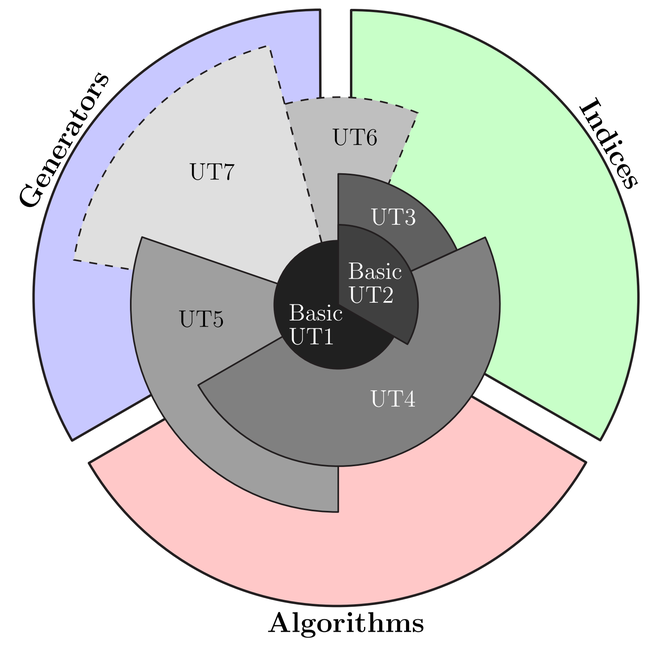
Our goal is to provide an experimental setup for benchmarking and evaluating clustering algorithms, quality indices, comparators, and other clustering-related techniques. More precisely, the evaluation framework consists of (simple) unit test that function as building blocks. The concept of unit tests was originally introduced in the field of software engineering and programming as an independent code module that ensures the correct functionality of a component. For example, such a test ensures that the associated methods of a data structure operate properly. They are frequently used when the implementation of a component is changed due to optimization, yet the functionality should remain. In our case, the provided experiments indicate the usability of a clustering technique. Similar to the tests in software engineering, our tests are only indicators, i. e., a meaningless technique can still successfully pass all test, while a failed test reveals its impracticality. In addition, the results of the tests themselves can be used to compare techniques and deepen the understanding.

In general, our evaluation framework is based on the repeated execution of experiments with fixed parameters. Each experiment consists of the following three stages: (1) generation of preclustered graphs, (2) execution of clustering algorithms, and, finally, (3) evaluation of obtained clusterings using quality indices and comparators. Due to the randomness inherent in the generators for preclustered graphs, each experiment has to be executed until (statistical) significance has been achieved.

Example of a Unit Test

UT1 For fixed generator and number of nodes, an increase (decrease) in the perturbation must not cause an increase (decrease) in coverage of the clustering used by the generator or obtained with an algorithm.

UT5 Given an algorithm passing UT4. The expected behavior of generators should be the following: The initial clustering used by a generator has to be at least as significant as the clustering calculated by the algorithm.



References

[1] Daniel Delling, Marco Gaertler, Robert Görke, Zoran Nikoloski, and Dorothea Wagner. How to Evaluate Clustering Techniques. Technical Report 2006-24, ITI Wagner, Faculty of Informatics, Universität Karlsruhe (TH), 2006.

Contact:
Daniel Delling, Marco Gaertler, Robert Görke, Dorothea Wagner
Universität Karlsruhe (TH)
Institute für Theoretische Informatik
{delling,gaertler,rgoerke,wagner}@informatik.uni-karlsruhe.de
http://i11www.informatik.uni-karlsruhe.de

Additional Authors
Zoran Nikoloski
Max-Planck Institute for Molecular Plant Physiology
Bioinformatics Group
nikoloski@mpimp-golm.mpg.de
http://bioinformatics.mpimp-golm.mpg.de

DELIS: Integrated European Project founded by the ”Complex Systems” Proactive Initiative within the Sixth Framework Programm