WebOCD: A RESTful Web-based Overlapping Community Detection Framework

Mohsen Shahriari
Advanced Community Information Systems (ACIS), RWTH Aachen University, Ahornstr. 55, 52056 Aachen, Germany
shahriari@dbis.rwth-aachen.de

Sebastian Krott
Advanced Community Information Systems (ACIS), RWTH Aachen University, Ahornstr. 55, 52056 Aachen, Germany
krott@dbis.rwth-aachen.de

Ralf Klamma
Advanced Community Information Systems (ACIS), RWTH Aachen University, Ahornstr. 55, 52056 Aachen, Germany
klamma@dbis.rwth-aachen.de

ABSTRACT

We introduce WebOCD, a Web-based framework for developing, evaluating and analysing (overlapping) community detection (OCD) algorithms. WebOCD is not only open source and extensible but also comprises several baseline algorithms. Thus, it provides a test bed for comparison of innovative algorithms. Moreover, all the functionalities are accessible as RESTful Web services and are consequently easily integratable into other software packages. An additional Web client provides a simple interface for end users. Besides the OCD algorithms, the framework allows the generation of benchmark graphs, metric calculation and community visualization.

CCS Concepts

•Information systems → Clustering; RESTful web services; •Theory of computation → Unsupervised learning and clustering; •Computer systems organization → *Client-server architectures;* **•Software and its engineering →** *Peer-to-peer architectures;* **•Mathematics of computing →** Graph algorithms;

Keywords

Community Detection, Benchmark Graphs, Web Services

1. INTRODUCTION

Communities are considered as structures in which intra community density is high and between community density is much lower. Today's research surge is more on overlapping community detection (OCD) [16, 6] . However, when researchers devise a novel algorithm, they face some problems regarding experimentation of the algorithm. They must sometimes develop the baseline algorithms and perform different pre-processing and post-processing steps to compare it with other algorithms. Therefore, the existence of such a standardized and generalized framework is necessary. Due to the popularity of OCD, there are already numerous software packages available for identifying communities. However, implementation of published OCD algorithms are offered e.g. by the authors, often times in the form of simple command line applications. One example is GANXiS[1] which essentially implements a newer version of the Speaker Listener Label Propagation Algorithm. GANXiS is written in Java and compatible with both weighted and directed networks. For Link Communities (LC) [1], there exist even multiple implementations in different programming languages[2], however, they handle only undirected and unweighted graphs.

Although some tools provide different algorithms, they do not offer a generalized framework with a user friendly interface. The application pylouvain-igraph[3][4] implements different versions of the so called Louvain method [2] for both directed and weighted graphs. The Community Detection Modularity Suite[5] contains three algorithms for community detection and OCD which are all based on modularity measures. It is implemented in C++ and permits also a multi-threaded execution of the algorithms. Other algorithms are distributed in combination with more integral tools that provide additional functionalities. For instance CFinder[6] is a Java implementation based on the Clique Percolation Method (CPM) [12] that offers a graphical user interface. The user not only is able to execute the algorithm but also to visualize the outputs. In addition, some statistical properties of graphs and covers as e.g. the degree and membership distribution are displayed via adequate diagrams. Moreover, it is also accessible as a web application. The Map Equation Framework[7] is a similar OCD tool implementing the Infomap algorithm [13]. It can e.g. display dynamic visualizations of the algorithm execution and the community structure and is also available as a Web application. Comparison of WebOCD with existing frameworks is roughly demonstrated in Table 1. Although numerous

[1]https://sites.google.com/site/communitydetectionslpa/
[2]http://jung.sourceforge.net/
[3]https://launchpad.net/pylouvain-igraph
[4]http://www.traag.net/2013/10/25/easy-flexible-framework-for-community-detection/
[5]http://sourceforge.net/projects/cdmsuite/
[6]http://www.cfinder.org/
[7]http://www.mapequation.org/index.html

i-KNOW '15, October 21-23, 2015, Graz, Austria

DOI: http://dx.doi.org/10.1145/2809563.2809593

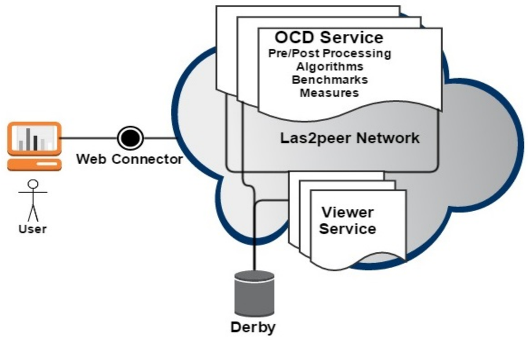

Figure 1: Framework Architecture

(overlapping) community detection tools exist, they are still lacking more convenient functionalities, In other words, the existing tools

- implement only a small number of algorithms
- do not allow input of directed or weighted graphs
- are time consuming to be used because they are based on different programming languages
- do not provide certain pre- or post-processing steps e.g, making graphs undirected and dividing them in to their connected components or normalizing and filtering the memberships of output covers
- do not contain neither OCD benchmarks nor metrics for creating synthetic graphs/covers and determining and comparing the quality of covers

Hence, we propose a novel standardized framework for evaluation, analysis and comparison of overlapping communities named WebOCD [8]. WebOCD supports multiple input and output formats, (un)directed and (un)weighted graphs, various standardized pre- and post-processing approaches, visualization of both graphs and covers, different benchmarks and finally the comparison of different algorithms through metrics. Researchers can either develop their novel algorithms via the framework (or by other programming language) and compare it with existing algorithms in the framework by the RESTful api (uploading the graph and returning the covers) or the Web client provided in the framework.

2. FRAMEWORK SKELETON

The overall architecture is shown in Figure 1. The OCD service is providing core OCD functionalities. The Viewer service is responsible for the generation of graphs and cover visualizations. Both services are based on las2peer[9] which runs on a peer-to-peer network. It provides developers with functionalities on top of which they can build their own applications. The WebOCD framework meets three principal functional requirements. First of all, different OCD algorithms for identifying covers are implemented. Secondly, different metrics for measuring the quality of detected covers are provided. Finally, the service is able to create visualizations of the resulting covers, especially for small-scale networks. Besides, some non-functional requirements were taken into account. The service was implemented in the Java programming language. Moreover, it follows a RESTful design and has a Web client to interact with the system. Additionally, the framework is easily extensible to allow the integration of further OCD algorithms.

[8]https://github.com/rwth-acis/REST-OCD-Services
[9]https://github.com/rwth-acis/LAS2peer

2.1 OCD Service

The core of the implemented framework is formed by the so called OCD service which is in charge of all OCD tasks like running algorithms, metrics and benchmarks. A list of all OCD algorithms is given in Table 2. Implemented metrics include extended modularity [11], extended NMI [8], Omega index [3] and execution time. Last but not least, benchmarks include LFR networks proposed by Lancichinetti [7] and Girvan-Newman Model [4].

2.2 Viewer Service and Web Client

The Viewer service creates visualizations of graphs and covers. The client screenshot from Figure 2 indicates a sample visualization of a cover. Here nodes with a common community share the same color. The color of overlapping nodes is determined by mixing the colors of their corresponding communities. Besides the Web services, we also created a Web client for the OCD framework. The intention was to provide an easy way for end users to access all OCD and visualization functionalities also outside the context of any additional applications. The implemented Web client does so through a simple graphical user interface and has additionally the benefit that it does not require any installations because it can be deployed as a Web page. The client is based entirely on JavaScript, HTML[10] and CSS and makes additional use of the JavaScript library jQuery[11]. It communicates with the services via HTTP requests by using Asynchronous JavaScript and XML (Ajax) which integrates several further technologies such as XML, the Document Object Model (DOM) and the XMLHttpRequest (XHR) API.

Figure 2 gives a basic impression of the client layout and indicates also its overall structure through the menu. With the menu option *graphs*, a user can access any of his graphs and e.g. take a look at its visualization or at corresponding meta information like the node or edge count. He may also run any of the OCD algorithms from Table 2 in order to detect the communities of the graph and to create a corresponding cover. Similarly, the option *covers* allows to manage the user's covers. Actually Figure 2 shows a page representing a cover. Again we have some meta information on the top such as the cover name, the name of the graph that the cover is based on, the number of communities and also the algorithm that the communities were created by. One can as well see the results of any metrics that have been calculated for the cover which allows us to judge the cover quality and to compare it to other communities. We can also observe how long it took the algorithm to calculate the cover. Moreover, we have the option to run a new OCD metric choosing any of the OCD algorithms in Table 2. On the bottom finally there is the cover visualization which also provides some navigation options.

Via *import* new graphs and covers can be uploaded from local files. Under *benchmarks* additionally new synthetic

[10]Hypertext Mark-up Language
[11]http://jquery.com/

Table 1: Comparison of WebOCD with other community detection frameworks from different aspects.

Framework	overlapping	Web Client	P2P	Pre/post processing	Benchmarks	Measures	Visualization
WebOCD	✓	✓	✓	✓	✓	✓	✓
Map Equation	✓	✓	X	✓	X	✓	✓
CD Modularity Suit	✓	X	X	X	X	✓	X
CFinder	✓	✓	X	✓	X	✓	✓
GANXiS	✓	X	X	X	X	X	X
Pylouvain-igraph	✓	X	X	X	X	X	X

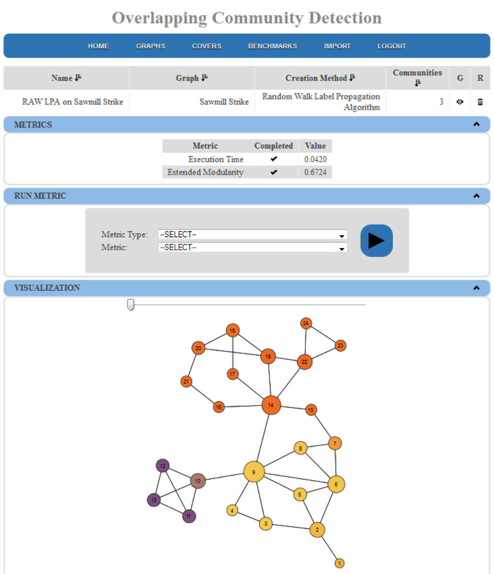

Figure 2: Layout of the Web Client.

Table 2: List of OCD algorithms

Clizz	**Zhang et al. [9]**
DMID	**shahriari et al. [14]**
SLPA	**Xie et al [17]**
LC	**Ahn et al [1]**
MONC	**Havemann et al. [5]**
SSK	**Stanoev et al. [15]**
DOCA	**Nam et al. [10]**

graphs and covers can be created with any of the available OCD benchmarks.

3. EVALUATION AND FUTURE DIRECTIONS

In order to evaluate in particular the usability of the system, we invited eight participants using the different functionalities in an evaluation session. All participating test users were either master or Ph.D students studying in the field of computer science or had already obtained a degree in computer science. Before the start of the actual evaluation each person was asked about their knowledge concerning OCD. The participants were then introduced into any unknown concepts and were given an overview over the basic components of the OCD framework. Afterwards, they received a number of different tasks including importing graphs, running algorithms, computing measures, evaluating results and creating synthetic networks which they had to accomplish with the help of the Web client. Users were able to finish their assigned tasks within 20 minutes. Finally, each participant was sent an invitation to fill out an anonymous Web survey. For means of evaluation all test users were asked to answer a Web survey containing statements which had to be rated on a 5-point scale from "Strongly Disagree" to "Strongly Agree". The statements are listed in Table 3 and the corresponding ratings are displayed through mean, median and standard deviation.

Taking a look at the participants' answers the feedback is very positive throughout. In other words, all of the statements have received mean values above 4.0. The only statement not obtaining a clearly positive rating was statement number five about useful error messages which is 3.0 for mean and median and 0 for standard deviation. Here several respondents stated that they neither agree nor disagree. Since various participants also chose the not applicable option this makes us assume that no real issues were encountered. Other than that all functionalities were confirmed to work well from the user perspective. Also the general design and the handling of the client received satisfactory feedback. The overall impression of the system was positive, too, and so was the rating for its speed of response as can be derived from questions 14 and 15 with mean rating of above 4.5. In addition, the survey contained the final statement "I could imagine to use the system if I had to complete overlapping community detection tasks". All participants did strongly agree on this which again confirms great approval for the framework implementation. As reasons for their agreement the participants mentioned that the framework offers a choice of multiple OCD algorithms and metrics. Moreover, it is flexible and easy to use while it provides a Web frond-end. We as well consider some aspects for future works

- The Web client should offer more explanations about the OCD algorithms in particular for users which are not familiar with them. Moreover, visualization of overlapping nodes should be extended by splitting the colors rather than mixing.
- It would be useful to allow a direct comparison between two different covers.
- At some points the work flow and feedback of the client's user interface can still be improved.
- The evaluation needs to be extended by comparing the framework with other OCD tools and evaluating with users possessing different expertise level.
- The framework should be extended by more algorithms.

Table 3: Responses to the system evaluation survey.

Questions	*Mean*	*Median*	*StandardDeviation*
The client has an intuitive design / is easy to understand.	4.33	4.5	0.67
The client is easy to use.	4.17	4.0	0.17
The client responded to my actions in the way I would expect.	5.0	5.0	0.0
I was able to execute the given tasks without having problems.	4.83	5.0	0.17
When I made a mistake a useful error message was shown.	3.0	3.0	0.0
It is easy to add new graphs to the system.	4.67	5.0	2.67
It is easy to add new covers to the system.	4.83	5.0	0.17
The graph visualization gives a good overview over the graph structure.	4.83	5.0	0.17
The cover visualization gives a good overview over the communities.	4.33	4.0	0.27
The visualizations are easy to handle (i.e. to zoom and to move).	4.5	4.5	0.3
The system is useful for handling OCD algorithms.	4.67	5.0	0.27
The system is useful for handling OCD benchmarks.	4.5	4.5	0.3
The system is useful for handling OCD metrics.	4.67	5.0	0.27
The system responded to my actions with a convenient speed.	4.67	5.0	0.27
The system works well.	4.5	4.5	3.0

4. CONCLUSION

We propose WebOCD as a framework for developing, testing and analysing overlapping community detection algorithms. This framework contains services for data pre/post-processing, visual analysis, benchmarking and evaluation of communities. It is a Web-based framework developed in Java and supports research communities with baseline overlapping community detection algorithms. In future, we plan to extend this framework by adding further algorithms and benchmarks and by improving the client.

Acknowledgment

The work has received funding from the European Commission's FP7 IP Learning Layers under grant agreement no 318209

5. REFERENCES

[1] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. Nature, 466(7307):761–764, 2010.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008.

[3] L. M. Collins and C. W. Dent. Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. *Multivariate Behavioral Research*, 23(2):231–242, 1988.

[4] M. Girvan and Newman, M E J. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[5] F. Havemann, M. Heinz, A. Struck, and J. Gläser. Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels. *Journal of Statistical Mechanics: Theory and Experiment*, 2011.

[6] S. Kelley. *Existence and discovery of overlapping communities in large-scale networks.* Proquest, Umi Dissertation, 2011.

[7] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review*, 80(1):016118, 2009.

[8] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

[9] H. J. Li, J. Zhang, Z. P. Liu, L. Chen, and X. S. Zhang. Identifying overlapping communities in social networks using multi-scale local information expansion. *The European Physical Journal B*, 85(6), 2012.

[10] N. Nguyen, T. Dinh, D. Nguyen, and M. Thai, editors. *Overlapping Community Structures and Their Detection on Social Networks*, 2011.

[11] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(03):P03024, 2009.

[12] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[13] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[14] M. Shahriari, S. Krott, and R. Klamma. Disassortative degree mixing and information diffusion for overlapping community detection in social networks (dmid). *Temporal Web WWW workshop*, 2015.

[15] A. Stanoev, D. Smilkov, and L. Kocarev. Identifying communities by influence dynamics in social networks. *Physical Review*, 84(4), 2011.

[16] J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: the state of the art and comparative study. *CoRR*, abs/1110.5813, 2013.

[17] J. Xie, B. K. Szymanski, and X. Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. *CoRR*, abs/1109.5720, 2011.