A Brief Comparison of Community Detection Algorithms over Semantic Web Data

Jose L. Martinez-Rodriguez[1], Ivan Lopez-Arevalo[1], Ana B. Rios-Alvarado[2], and Xiaoou Li[3]

[1] Cinvestav-Tamaulipas
Victoria, Mexico
lmartinez,ilopez@tamps.cinvestav.mx
[2] Autonomous University of Tamaulipas
Victoria, Mexico
arios@uat.edu.mx
[3] Cinvestav-IPN
Mexico City, Mexico
lixo@cs.cinvestav.mx

Abstract. Community detection is a task responsible for categorizing nodes of a graph into groups that share similar features or properties (e.g. topological structure or node attributes). This is an important task in fields such as social network analysis or pattern recognition that span a large and varied amount of information hiding relations with knowledge. In this sense, an initiative that seeks to extract knowledge from data is the Semantic Web, whose primary goal is to represent Web data into a graph in order to discover facts and relations. In this paper, we developed a strategy to apply community detection algorithms over Semantic Web data graphs. For this purpose, five algorithms were tested to identify groups from a dataset retrieved from the DBpedia knowledge base containing more than 45 thousand nodes and almost 500 thousand edges in the domain of movies. Clustering quality was evaluated by using the modularity measure and the features of the best communities were analyzed.

1 Introduction

Nowadays information about real world things is mostly represented with elements or entities such as people, locations, dates, and so forth. This kind of information commonly shows connections among elements organized into a graph. Thus, a graph or network is composed of a set of objects called vertices (nodes) joined by links (also known as arcs or edges) that allow them to represent binary relations among dataset elements.

Sometimes, nodes tend to be connected with many other nodes that share common features, for example, people that like the same music genre or enjoy a movie classification would take advantage of being organized into a group, probably to share experiences, reviews or for taking decisions. Hence, the community detection task aims to facilitate the labeling and allocating of nodes into groups. However, actually the amount of information grows exponentially and therefore, generating a graph for further assigning nodes to communities that share preferences is a difficult task. In particular, one of the bigger data source hiding relations able to demonstrate knowledge about the world is the Web; but unfortunately, a large portion of the Web lacks a formal structure able to be processed automatically by computers. As a consequence, the Semantic Web was presented as an initiative whose purpose is to create a formal representation of the information expressed on the Web. The Semantic Web follows a basic model used to formally and semantically represent information, it is called RDF triple[4], which is mainly composed of three basic elements: *Subject-Predicate-Object.*

The RDF triple structure and an example of how an RDF statement about a place is located in a country are presented in Figure 1, where nodes represent Villa Nellcôte as Subject, France as Object, joined by locatedIn as Predicate (edge). This linking structure allows to organize data in a directed, labeled graph[5][5].

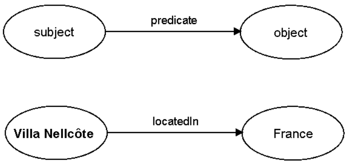

Fig. 1. RDF triple example

One of the most important projects in the Semantic Web is the Linked Open Data (LOD) cloud[6] which provides more than 70 billion triples modelled as a graph with information about government institutions, locations, music descriptions, and so on. In this regard, DBpedia[7] is an important LOD dataset that offers contents extracted from Wikipedia and hence, it implies that information can be separated into subsections adequate to be processed by algorithms.

In a nutshell, in this paper we apply traditional community detection algorithms over information from the Semantic Web, specifically a subsection of the DBpedia knowledge base about movies and actors domain. We provide a strategy to process the information from the Semantic Web through community detection algorithms, databases, and a visualization tool. Because of this proposed strategy, detected groups sharing similar features aim to provide a way to enrich Semantic Web information.

[4] RDF (Resource Description Framework) `http://www.w3.org/RDF/`, [last visit June 10, 2016]

[5] `https://www.w3.org/DesignIssues/LinkedData.html`, [last visit June 10, 2016]

[6] LOD cloud state `http://lod-cloud.net`, [last update August 30, 2014]

[7] `http://wiki.dbpedia.org/`, [last visit June 10, 2016]

The rest of the paper is organized as follows. Section 2 presents related work regarding the community detection algorithms and grouping strategies in the Semantic Web. Our proposed strategy is outlined in Section 3. Section 4 holds an analysis of results. Finally, conclusions are presented in Section 5.

2 Related work

The community detection idea is not new and some already presented algorithms in the state of the art so far are taken into account to guide a conceptual sustenance of this work.

Several works have presented a review of community detection algorithms. Schaeffer [20] presented a study with graph clustering definitions and terminology, moreover, he presented methods and measures to identify communities and an evaluation regarding the overseen techniques. Harenberg *et al.* [10] developed an empirical revision of algorithms for community detection separated in two modalities: overlapping algorithms and disjoint algorithms. At the first place, overlapping techniques (non-exclusive) detect nodes of a dataset belonging to more than one community; for example, a musical singer who also is a movie actor. In the second place, disjoint algorithms differ from the first category in the sense that only an exclusive category is assigned to every node. Focused on machine learning algorithms, Giannini [9] performed one of the first attempts to integrate community detection strategies over Semantic Web data. She mentioned that the purpose of making clustering over RDF graphs is to detect groups of vertices that share common properties or that play a similar role by using only information about the topology of the graph. In this sense, Khosravi-Farsani *et al.* [7] proposed a similarity score model applied to RDF data clustering by considering the number of shortest paths between resources. The authors test their model with information about person resources obtained from DBpedia. The similarity among resources is thoroughly tackled in the Ontology Matching area [6] and some approaches [11],[13] have taken advantage of the measures proposed by this. However, in this paper, we only consider topology characteristics of a subset of the DBpedia graph in order to detect communities and as distinct to proposed approaches, we provide a strategy to apply traditional community detection algorithms.

3 Methodology

In order to carry out the community detection algorithms testing over Semantic Web data, we propose the following strategy composed of four stages:

Semantic Web data acquisition: At this stage, we only consider a Semantic Web subset for testing.

Preprocessing: Information needs to be adapted to apply traditional algorithms.

Community detection labeling: The five selected algorithms are tested in this stage.

Analysis of results: Features of the obtained groups are analyzed.

In following subsections, the process involved in every stage is described.

3.1 Semantic Web data acquisition

Due to the huge amount of information on the Semantic Web (more than 70 billion triples), using the whole graph is not possible for a quick test. Only parts of this graph have been used for applying proposed algorithms [7],[1]. Thus, in order to demonstrate the applicability of community detection algorithms, we only focused on a subset of the Semantic Web graph that has a rich organization and diversity of contents easily understandable by people; we have selected the DBpedia dataset regarding the domain of films: actors and movies. Despite Linked Movie Data Base (LinkedMDB)[8] provides information about movies, we are focus on the implementation of the algorithms on LOD cloud datasets. However, we plan to include diverse and wide datasets in a further implementation.

The straightforward way to obtain information from DBpedia[9] is through their official SPARQL[10] endpoint. We performed queries as the one depicted in Listing 1.1.

Listing 1.1. SPARQL query for movie domain resources retrieval

```
SELECT ?movie ?genre ?actor ?duration
WHERE {     ?movie a dbpedia-owl:Film ;   dbpedia-owl:starring ?actor .
   OPTIONAL { ?movie dbpprop:duration ?duration .   }
   OPTIONAL { ?movie dbpprop:genre ?genre .   } }
```

3.2 Preprocessing

Once the information was retrieved, some preprocessing tasks should be applied in order to clean and prepare the data. The output of the previous stage produced 254,251 rows in CSV format with two classes; actors and movies, obtaining a bipartite graph with actors performing movies. However, as stated by Fortunato [8], multipartite networks are usually projected into unipartite graphs in order to apply standard community detection algorithms, even when there is an information loss by doing this transformation, different configurations may be obtained to produce communities. Thus, according to the output format, we imported the data into a MySQL table for an easy and fast manipulation of information. Then, a second graph was generated where nodes were expressed as movies and edges represented common actors between movies. This was only produced for testing purposes because data may be expressed as authors sharing movies in common. In this way, it was obtained a graph with the properties presented in Table 1.

[8] `http://linkedmdb.org`, [last visit June 10, 2016]

[9] DBpedia SPARQL endpoint `http://dbpedia.org/sparql`, [last visit June 10, 2016]

[10] A SQL like language used to query RDF information

Table 1. Graph properties

Property	Value
Nodes	45960
Edges	460562
Connected Components	299
Diameter (the largest from the components)	20
No. Shortest Paths	2003779804
Avg. clustering coefficient	0.456

In order to obtain the influence of individuals, a measure that defines the importance of a node within a graph is the Centrality [21] and it is based on graph paths. A path or walk is a sequence of edges connecting a sequence of distinct vertices. That is, a route through a graph from vertex to vertex along edges. Therefore, the shortest path between vertices is the sequence of vertices with fewest edges required to connect two vertices. In this sense, two indicators explored in this study were Betweenness Centrality (BC) and Closeness Centrality (CC). The first one is given by the number of shortest paths from all vertices to all others going through a vertex. And the second one measures the amount of steps required to access a node from another. Both BC and CC were computed by using the Gephi tool[11]. Details about these implementations and algorithms are provided by Brandes [3].

The BC distribution of the nodes in the graph is depicted in Figure 2. More than 12 thousand nodes obtained a value close to 0, which means those nodes are away from paths between other nodes and therefore belong only to one the communities. On the contrary, only a few nodes hold values in the range of millions, that is, nodes joining communities. Respect to the CC distribution, the values obtained for the graph are depicted in Figure 3. A total of 478 nodes obtained a high centrality, that is, central nodes with a small average shortest path length to other nodes, which gives an idea of the number of groups in the dataset, as provided in section 3.3.

Information about the movie class was considered as nodes, but it is possible to arrange information in order to allocate actors as nodes connected by movies to see the relation among them, that is, how actors collaborate with each other by means of movies.

3.3 Community detection labeling

There are many community detection algorithms with exclusive and non-exclusive nature. In this paper, only exclusive algorithms were considered, that is, where

[11] Gephi graph visualization `https://gephi.org/`, [last visit June 10, 2016]

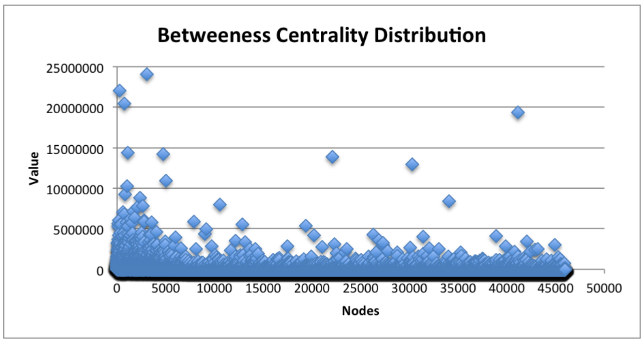

Fig. 2. Betweenness Centrality Histogram

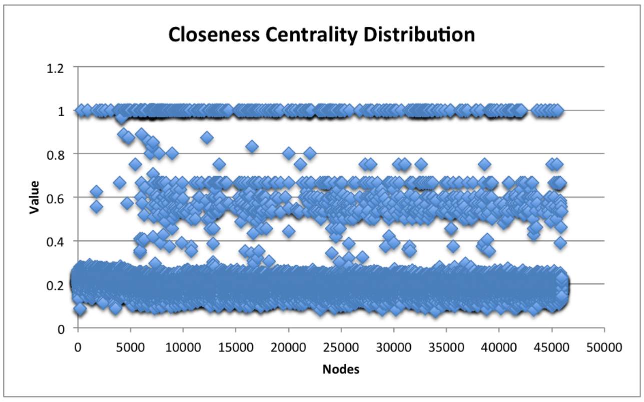

Fig. 3. Closeness Centrality Histogram

individuals only belong to one community. The tested algorithms in this experiment are: Fastgreedy [4], Multilevel [2], Walktrap [18], Infomap [19] and Leading Eigenvector [16]. A common feature among these algorithms is that they apply the modularity measure to evaluate the strength of division of a network into communities. Modularity is defined by Newman and Girman [15] as follows:

$$Q = \frac{1}{2m}\sum(A_{ij} - P_{ij})\delta(C_i, C_j)$$

where m is the number of edges, A is the adjacency matrix, P is the probability of an edge existing between vertices i and j, and $\delta(C_i, C_j) = 1$ iff i and j belong to the same community (cluster). P is calculated as follows:

$$P_{ij} = \frac{k_i k_j}{2m}$$

here k is the degree of a vertex and m is the number of edges. In order to apply the aforementioned algorithms over the selected dataset, the iGraph[12] Phyton package was used. Finally, the result of the grouping was visually plotted with the Gephi tool.

4 Analysis of results

After executing the five community detection algorithms, the obtained results together with a visualization of traces denoting the grouping organization by every algorithm are analyzed in this section. All of the experiments were developed under a computer with Intel Core i5 processor, 8GB RAM, and OS X 10.10. The number of detected communities and the obtained modularity score by the algorithms is given in Table 2, where the first column indicates the evaluated algorithm, the second provides the number of communities, the third is for the Modularity score, the fourth column contains the execution time, and the last column provides the computational complexity of the algorithm.

Algorithm	Communities	Modularity	Time	Order
Walktrap	1338	0.7119	7m17.269s	$O(n^3)$
Multilevel	376	**0.7363**	4.009s	$O(m)$
Infomap	1757	0.5923	10m30.748s	$O(m)$
Fastgreedy	463	0.6587	3m1.411s	$O(nlog^2n)$
L.Eigenvector	324	0.5613	1m47.124s	$O((m+n)n)$

Table 2. Modularity by algorithm

Regarding to the identified communities, the Infomap algorithm obtained the highest number of communities. However, it also got low modularity and the worst execution time of the test; this is due to the principle of random walks that tend to fall into isolated groups of nodes.

With respect to the modularity, it can be seen that Multilevel algorithm obtained the best score, this is due to the way it iteratively moves nodes among communities by considering the fluctuation of the modularity. An interesting fact is the one given by the Walktrap algorithm, at the second position of results; this is because, even this algorithm is based on random walks like as the Infomap algorithm, it takes into account information of the community structure. The distribution of the ten most populated communities obtained by every one of the five tested algorithms can be seen in Figure 4.

The Multilevel algorithm obtained the best modularity; in this sense, some features about movie genres from the nine most populated communities found by this algorithm are analyzed below.

[12] `http://cran.r-project.org/web/packages/igraph/index.html`, [last visit June 10, 2016]

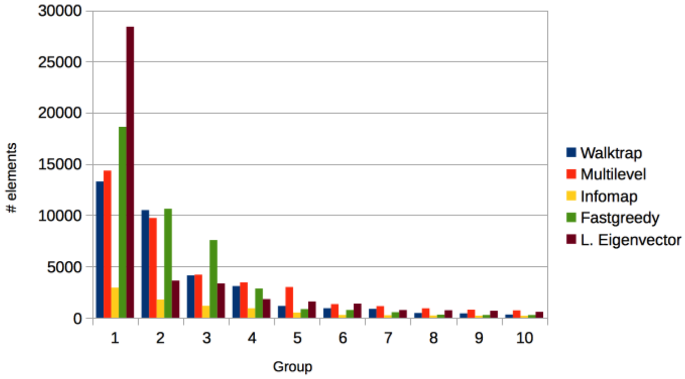

Fig. 4. Most populated communities by algorithm

The organization of nodes given by the Multilevel algorithm (highest modularity) can be visualized in Figure 5, where colors indicate the communities and it gives the idea of the dispersion presented by this algorithm.

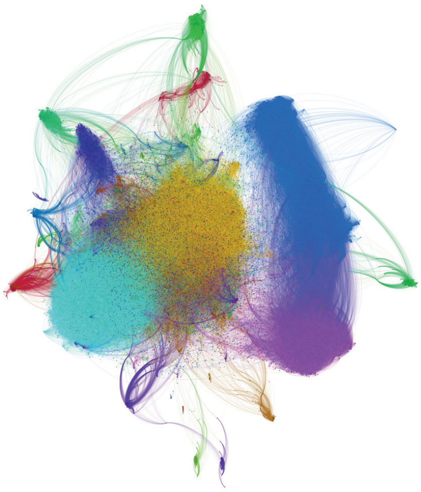

Fig. 5. Multilevel communities visualization

The most usual movie genres in the dataset are *Drama, Suspense, Action, Comedy*, and *Horror*. By taking this into account, the corresponding percentages for every one of the five movie categories in the nine considered communities are presented in Table 3, where letters indicate Category (C), Group (G), Drama

(Dra), Action (Act), Comedy (Com) and Horror (Horr). Despite *Comedy* and *Drama* categories seem to be opposite each other (emotionally) and that exclusive algorithms are only considered in this work, they have more elements than others categories in the first two communities, this overlapping is present in genre categories because of the participation of actors in different movies. Hence, the predominance of categories in every community is considered because it is helpful for different applications such as recommender systems [17], marketing [14] or social networks analysis [12], to seek for associations among users. In this study only a few common features were observed, however, this is encouraging to produce diverse studies from Semantic Web data in such a way that other domains can be processed to discover information and analyze the data distribution and features from groups.

G/C	Dra	Susp	Act	Com	Horr	# elements
368	48.45	14.08	8.59	16.49	12.37	14353
336	43.58	15.38	5.12	23.07	12.82	9725
262	26.66	20	33.3	13.3	6.66	4198
319	21.73	8.69	34.78	34.78	0	3445
330	85.71	0	0	14.28	0	2993
204	0	0	66.66	33.33	0	1326
366	16.66	0	50	0	33.33	1129
202	50	30	10	0	10	912
271	0	0	0	100	0	783

Table 3. Movie genre distribution

5 Conclusions

In this paper, some community detection algorithms over Semantic Web data were tested. Such algorithms are based on topology features and evaluation-measure fluctuation to determine groups or communities with acceptable quality. Information provided by community detection algorithms is helpful to produce observations or predictions from data. In this sense, we aimed to develop a strategy for analyzing and organizing information provided by a structured data source such as the Semantic Web. The Semantic Web offers many benefits such as a graph based-model, facts about real world objects and the integration of heterogeneous data sources, but it is a huge repository which is impractical to process it by a single computer. For this reason, only a relatively small subset of the Semantic Web was selected in order to test the selected algorithms. However, some data adequacies were required, we provided a strategy composed of stages for data acquisition, preprocessing, community detection, and analysis. As a result of testing algorithms and generating communities, we have analyzed information regarding to the actors and movies domain as a proof of concept.

Therefore, we consider that our strategy is able to be used in other domains such as Biology, Medicine or Publishing, to mention a few.

In addition, it is important to note that the community detection algorithms provide new facts that can be used for inference task, which is a very important task in the Semantic Web area. Therefore, the strategy to implement community detection algorithms over Semantic Web data may be used in further tasks such as:

- Testing with diverse Knowledge bases and domains
- Leverage the output of the best community detection algorithm in tasks related to OL&P (e.g. ontology axioms generation)
- Provide an enrichment to the LOD cloud with newly discovered insights (context, group descriptions, inference)

References

1. Bikakis, N., Skourla, M., Papastefanatos, G.: rdf: Synopsviz-a framework for hierarchical linked data visual exploration and analysis. In: European Semantic Web Conference. pp. 292–297. Springer (2014)
2. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment (10) (2008)
3. Brandes, U.: A faster algorithm for betweenness centrality*. Journal of mathematical sociology 25(2), 163–177 (2001)
4. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E 70, 066111 (Dec 2004)
5. Colomo-Palacios, R., Sánchez-Cervantes, J.L., Alor-Hernández, G., Rodríguez-González, A.: Linked data: Perspectives for it professionals. International Journal of Human Capital and Information Technology Professionals 3(3), 1–12 (2012)
6. Euzenat, J., Shvaiko, P.: Ontology Matching (Second Edition), vol. 2. Springer Berlin Heidelberg (2013)
7. Farsani, H.K., Nematbakhsh, M.A., Lausen, G.: Srank: Shortest paths as distance between nodes of a graph with application to RDF clustering. J. Information Science 39(2), 198–210 (2013), http://dx.doi.org/10.1177/0165551512463994
8. Fortunato, S.: Community detection in graphs. Physics Reports 486(3), 103 (2010)
9. Giannini, S.: Rdf data clustering. In: BI Systems Workshops (2013)
10. Harenberg, S., Bello, G., Gjeltema, L., Ranshous, S., Harlalka, J., Seay, R., Padmanabhan, K., Samatova, N.: Community detection in large-scale networks: a survey and empirical evaluation. Wiley Interdisciplinary Reviews (2014)
11. Hotho, A., Maedche, A., Staab, S.: Ontology-based text document clustering. KI 16(4), 48–54 (2002)
12. Jung, J.J., Euzenat, J.: Towards semantic social networks. In: European Semantic Web Conference. pp. 267–280. Springer (2007)
13. Maedche, A., Zacharias, V.: Clustering ontology-based metadata in the semantic web. In: Principles of Data Mining and Knowledge Discovery (2002)
14. Nadori, Y.L.E.M., Erramdani, M., Moussaoui, M.: Semantic web-marketing 3.0: Advertisement transformation by modeling. In: Multimedia Computing and Systems (ICMCS), 2014 International Conference on. pp. 569–574. IEEE (2014)

15. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E 69, 026113 (Feb 2004)
16. Newman, M.E.: Finding community structure in networks using the eigenvectors of matrices. Physical review E 74(3), 036104 (2006)
17. Passant, A.: dbrec - music recommendations using dbpedia. In: The Semantic Web ISWC 2010, LNCS, vol. 6497, pp. 209–224. Springer Berlin Heidelberg (2010)
18. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: Computer and Information Sciences, vol. 3733, pp. 284–293 (2005)
19. Rosvall, M., Axelsson, D., Bergstrom, C.T.: The map equation. The European Physical Journal Special Topics 178(1), 13–23 (2009)
20. Schaeffer, S.E.: Survey: Graph clustering. Comput. Sci. Rev. (2007)
21. Zafarani, R., Abbasi, M.A., Liu, H.: Social media mining: an introduction. Cambridge University Press (2014)