

Social Media Network Simplification Using Morphological Operators

Erick López-Ornelas, Rocío Abascal Mena

Universidad Autónoma Metropolitana - Departamento de Tecnologías de la Información,
Cuajimalpa, México

{elopez, mabascal}@correo.cua.uam.mx
<http://www.cua.uam.mx/>

Abstract. Complex networks arise in many diverse contexts, ranging from web pages and their links, computer networks, and social networks interactions. The modelling and mining of these large-scale, self-organizing systems is a broad effort spanning many disciplines. This article proposes the use of morphological operators, based on Mathematical Morphology, to simplify a set of interactions in a complex social media network. By applying these morphological operators, it is possible to simplify the social network and thus identify important communities and actors in the network. An analysis based on the visualization of the communities was carried out to verify the pertinence of the detection and simplification.

Keywords. Morphological operators, mathematical morphology, social media network, network simplification.

1 Introduction

Social media networks such as microblogs and social networks e.g. Twitter, LinkedIn and Facebook, are provide interactive and cheaper way for user to share ideas, exchange information and stay connected with people. Ease in using social media applications on mobile devices achieves rapid growth in social media network users and leads to generate vast amount of user generated content.

This large user base and their discussions produces huge amount of user generated data. Such social media data comprises rich source of information which is able to provide tremendous opportunities for companies to effectively reach out to a large number of audience.

With the current popularity of these Social media networks (SMN), there is an increasing interest in their measurement and modelling. In addition to other complex networks properties, SMN exhibit shrinking distances over time, increasing average degree, and bad spectral expansion.

Unlike other complex networks such as the web graph, models for SMN are relatively new and lesser known. In this kind of networks, models may help detect, simplify and classify communities, and better clarify how news and gossip is spread in social networks.

Network simplification can provide benefits to applications of various domains and for suggesting like-minded people to user which are still unknown to him/her.

An important practical problem in social networks is to simplify the network of users based on their shared content and relationship with other users. Community in network is a pattern with dense links internally and sparse links externally. These links can be characterized by the content similarity between users, friendship between them and also other similarities in their personal data such as their location, gender, age etc.

In other hand, Mathematical morphology is generally studied as an aspect of image processing [1]. As digital images are usually two-dimensional arrangements of pixels, where spatial relationships between elements of the image are essential features.

Mathematical Morphology is a theory that studies the decomposition of lattice operators in terms of some families of elementary lattice operators [2]. When the lattices are considered as a multidimensional graph (e.g. Social Media Network), the elementary operators can be characterized by structuring functions. The representation of structuring functions by neighborhood graphs is a powerful model for the construction of morphological operators.

This article proposes the use of morphological operators, based on Mathematical Morphology, to simplify a set of interactions in a complex social network. By applying these morphological operators, it is possible to simplify the social network and thus execute important queries in the network.

The structure of the article is as follows. In Section 2 the materials and methods are shown giving the essentials of mathematical morphology and how this technique is applied on SMN. The essential operators applied are shown and explained (Opening and Closing). An example of this simplification is shown in section 3 and the query modeling is explained. The conclusions and directions for further work are given in the final section.

1.1 Complex Networks Properties

Researchers are now in the position of observing how SMN evolve over time, and how the network is constructed. Unlike in traditional social network analysis, social interaction of millions of people from across the world can be elicited and some properties can be described. For definitions of the terms used below, see [3].

(i) Large-scale. SMN are examples of complex networks with number of nodes often in the millions; further, some users have disproportionately high degrees. About half a billion users are registered on Facebook [4]. Some of the nodes of Twitter corresponding to well-known celebrities including Katy Perry, Justin Bieber or Barak Obama have degree over ninety million [5].

(ii) Small world property and shrinking distances. The small world property, is a central notion in the study of complex networks [6]. The small world property demands a low diameter and a higher clustering coefficient than can be found in this kind of network [7]. Low diameter and high clustering coefficient were reported in the Twitter by Java et al. [8].

(iii) Power law degree distributions. Power laws were observed over a decade ago in subgraphs sampled from the web graph and are ubiquitous properties of complex networks. Kumar [9] studied the evolution of Flickr and found that this network exhibit power-law degree distributions.

(iv) Bad spectral expansion. Social networks often organize into separate clusters in which the intra-cluster links are significantly higher than the number of intercluster links. In particular, social networks contain communities, where some groups correspond to the clusters [10]. As a result, this kind of network possess bad spectral expansion properties realized by small gaps between the first and second eigenvalues of their adjacency matrices.

1.2 Some Techniques of Network Simplification

Similar work has been conducted on simplifying networks. In [11], the authors developed 3 different algorithms. The first decomposes a large network into some smaller sub-networks, generally overlapped. The remaining two carry out simplification based on commute times within the network. The algorithms produce a multilayered representation. All three algorithms use their simplified representations to perform matching between the input network.

In [12], the author uses a simplification algorithm to generate simplified network for input into their network layout algorithms. The network is not visualized and presented to the user as a way to help them better understand the network. Instead, a series of progressively simplified networks are used to guide the positioning of the nodes in the network.

Additional simplification algorithms have been proposed to assist in robot path planning [13], classifying the topology of surfaces [14], and improving the computational complexity and memory requirements of dense graph processing algorithms [15].

Simplification may also be accomplished through the clustering technique. In [16], authors define one such clustering algorithm for better visualizing the community structure of network graphs. The kind of graphs that they are targeting with this technique are those that would have a naturally occurring community structure.

Some authors present another approach to visually simplifying large scale graphs [17]. They have developed different methods for randomly sampling a network and using that sampling to construct the visual representation of the complex network.

In [18], the authors focus on providing metrics for simplifying graphs that represent specific network topologies. The goal of this work is to simplify and visualize complex network graphs while maintaining their semantic structures. Although network topologies are certainly reasonable candidates for these visualization techniques, there are other sets of graph data that could greatly benefit from simplification techniques for

visualization. Their general approach is similar to ours in that they are using characteristics of the graphs that frequently occur, using Morphological operators some physical characteristics are kept and some nodes “grow” and some “shrink” in order to obtain a better simplification. This process will be explained in next section.

2 Materials and Methods

2.1 Mathematical Morphology and Networks

The study on Mathematical morphology, started at the end of the Sixties and was proposed by Matheron and Serra [19]. Mathematical morphology rests in the study of the geometry and forms; the principal characteristic of the morphological operations is image segmentation and conservation of the principal features forms of the nodes [1].

Despite its origin, it was recognized that the roots of this theory were in algebraic theory, notably the framework of complete lattices [20]. This allows the theory to be completely adaptable to non-continuous spaces, such as graphs and networks. For a survey of the state of the art in mathematical morphology, we recommend [21].

2.2 Social Media Network Representation

SMN is a graph representation having a set of data where some pairs of data are connected by links. Once a SMN representation is adopted, the interconnected data are called vertices or nodes of the Network and the links that connect vertices are called edges or arcs. An edge of the graph is then simply a pair of connected vertices. Thus, a network is made of a set of vertices and of a set of edges. We can also associate to each vertex and/or to each edge a weight that represents some kind of measure on the network (in our case friends, likes, follows, mentions, etc.), leading to weighted graphs.

In terms of notation, Social Media networks employ the typical graph notation $G = (V, E)$, where G stands for the whole network, V stands for the set of all vertices and E for the set of all edges. Due to the different types of vertices and edges in such networks, it is common to consider sets of vertices and edges within V and E that contain vertices and edges of the same type. For instance, in the case of a photo sharing and tagging network, one can consider the set of vertices V to comprise the users, photos and tags of the system, i.e. $V = \{U, P, T\}$. Similarly, the set of edges in such an application would comprise the set of user-photo, photo-tag and user-tag associations, $E = \{UP, PT, UT\}$.

This graph (V, E) will have the following characteristics: (i) it is “non-oriented”, that is: $\forall (v, w) \in E, (w, v) \in E$; (ii) it is 1-graph: the same pair cannot be represented more than one time in the family of arcs E ; (iii) it has no loops: $\forall v \in V, (v, v) \notin E$.

2.3 Morphological Operators

The algebraic basis of Mathematical morphology is the lattice structure and the morphological operators act on lattices [2]. In other words, the morphological operators map the elements of a first lattice to the elements of second one (which is not always

the same as the first one). A lattice is a partially ordered set such that for any family of elements, we can always find a least upper bound and a greatest lower bound (called a supremum and an infimum). The supremum (resp., infimum) of a family of elements is then the smallest (greatest) element among all elements greater (smaller) than every element in the considered family.

The supremum is given by the union and the infimum by the intersection. A morphological operator is then a mapping that associates to any subset of nodes another subset of nodes. Similarly, given a graph, one can consider the lattice of all sub sets of vertices [22] and the lattice of all subsets of edges. The supremum and infimum in these lattices are also the union and intersection. In some cases, it is also interesting to consider a lattice whose elements are graphs, so that the inputs and outputs of the operators are graphs.

The algebraic framework of morphology relies mostly on a relation between operators called adjunction [2]. This relation is particularly interesting, because it extends single operators to a whole family of other interesting operators: having a dilation (resp., an erosion), an (adjunct) erosion (resp., a dilation) can always be derived, then by applying successively these two adjunct operators a closing and an opening are obtained in turn (depending which of the two operators is first applied), and finally composing this opening and closing leads to alternating filters.

Firstly, they are all increasing, meaning that if we have two ordered elements, then the results of the operator applied to these elements are also ordered, so the morphological operators preserve order. Additionally, the following important properties hold true:

- the dilation (resp., erosion) commutes under supremum (resp., infimum);
- the opening, closing and alternating filters are indeed morphological filters, which means that they are both increasing and idempotent (after applying a filter to an element of the lattice, applying it again does not change the result);
- the closing (resp., opening) is extensive (resp., antiextensive), which means that the result of the operator is always larger (resp., smaller) than the initial object;

In the graph $G(V, E)$, if the vertex (V_i) of the graph constitutes the digital grid and its neighbors their interactions, then the process compares and affects the interaction value of v_i on the graph constructed using the morphological transformations. These transformations are the core of the simplification.

The principle of the growing/shrinking the graph consists in transform the $G(v)$ value by affecting the nearest interaction value $val(v_i)$ present among the v neighbor's nodes. The new graph v_n is then the result of the fusion of nodes. To carry out this transformation, the morphological operations on the graph are applied and a loop is generated until the reach of one threshold parameter.

Let us assume that we have a flat structuring element that corresponds to the neighbor's nodes Structuring Element ($SE \equiv NE(v)$). Then the eroded graph $\varepsilon(G(v))$ is defined by the infimum of the values of the function in the neighborhood [23]:

$$\varepsilon(G(v)) = \{ \wedge G(v_i), v_i \in NE(v) \cup \{v\} \}.$$

Dilation $\delta(G(v))$ is similarly defined by the supremum of the neighboring values and the value of $G(v)$ as:

$$\delta(G(v)) = \{v \vee G(v_i), v_i \in N_E(v) \cup \{v\}\}.$$

Classically, opening γ is defined as the result of erosion followed by dilation using the same SE:

$$\gamma(G(v)) = \delta(\varepsilon(G(v))).$$

Similarly, closing φ is defined as the result of dilation followed by erosion with the same SE:

$$\varphi(G(v)) = \varepsilon(\delta(G(v))).$$

The geometrical action of the openings and closings transformations, $\gamma(G(v))$ and $\varphi(G(v))$ respectively, produce a growing/shrinking of the graph. Of course, this fusion process can be regulated using parameters for the opening and closing, but also we can regulate the growing depending on the information that we need to compare. The graph has to be updated to keep aggregating the different nodes always applying the morphological transformations of $\gamma(G(v))$ and $\varphi(G(v))$ until their parameter value is reached. In figure 1 and figure 2, some morphological operations are shown. We can see the difference applying different morphological operators on the same graph.

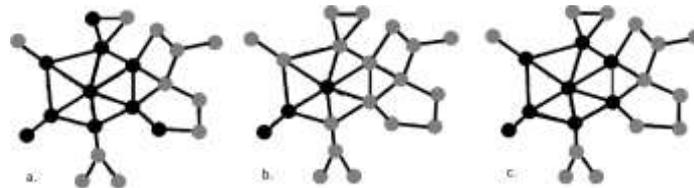


Fig. 1. a. Random graph selected. b. Eroded $\varepsilon(G(v))$. c. Dilated graph $\delta(G(v))$.

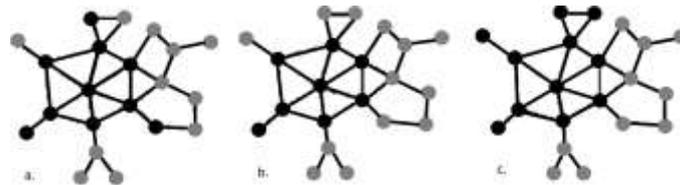


Fig. 2. a. Random graph selected. b. Opening transformation $\gamma(G(v))$. c. Closing transformation $\varphi(G(v))$.

3 Experiment

3.1 Social Network Simplification

In this article, as an experiment, we show a set of interactions on Twitter. This information was extracted from Twitter and explores a trend topic appeared in México. The hashtag is *#estafamaestra*, and it was arising from the corruption scandals generated in Mexico in September 2017. Among the multiple elements of analysis, we decided to

use morphological operators in order to simplify the original network, the study was made taking into account the characteristics of each node for their simplification.

The information concentrated in figure 3 corresponds to the extraction carried out on September 5, 2017. In the network are represented 3399 nodes (tweeters) and 5502 arcs (interactions) that were made between them. This is a complex interaction network, so it is important to simplify it in order to perform a better analysis of the interactions that were generated in the social network.

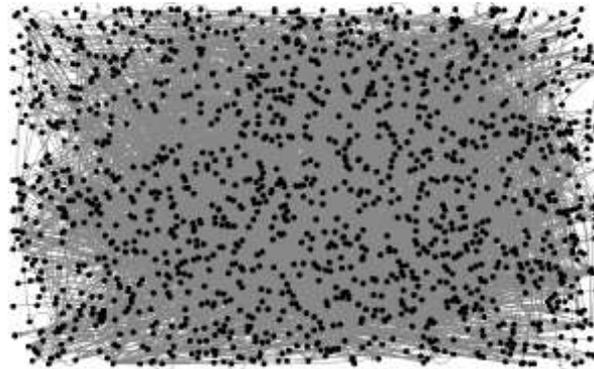


Fig. 3. Complete random graph selected with 3399 nodes and 5502 interactions.

3.2 Social Media Network Representation

At the most abstract level, given a Social Media network $G = (V, E)$, where G stands for the whole network, V stands for the set of all vertices and E for the set of all edges, each Social Media interaction can be defined as a subgraph of the network comprising a set $V_C \subseteq V$ of Social Media entities that are associated with a common element of interest.

This element can be as varied as a topic, a real-world person, a place, an event, an activity or a cause.

For instance, in the case of Twitter network, one can consider the set of vertices V to comprise the users, mentions, tweet content, tweet favorites and retweets, i.e. $V = \{U, M, Tc, Tf, Rt\}$. The edges in such an application would comprise the set of followed, followers, tweet number, image profile and location, $E = \{Fd, Fs, Tn, Ip, L\}$.

Even if we can use all these characteristics to apply morphological operators, we have decided to only use 3 node elements to carry out the simplification. These elements are the mentions, number of favorites and retweets represented by $V = (M, Tf, Rt)$.

3.3 Decisional Node Aggregation

The principle of the union of nodes consists in transform the $G(v)$ value by affecting the nearest Tf value $val(v_i)$ present among the v neighbors, and the grouping process is the union of nodes $(v_i \cup v_j = v_n)$. The new node v_n is then the result of the fusion of

nodes. To carry out this transformation, the morphological operations on the graph are applied.

Let us assume that we have a flat structuring element that corresponds to the neighborhood Structuring Element ($SE \equiv N_E(v)$). Then the eroded graph $\varepsilon(G(v))$ is defined by the infimum of the values of the function in the neighborhood and represents the minimum value found on the neighbors [24]:

$$\varepsilon(G(v)) = \min \{G(v_i), v_i \in N_E(v) \cup \{v\}\}.$$

Dilation $\delta(G(v))$ is similarly defined by the supremum of the neighboring values and the value of $G(v)$ and it is represented by the maximum value found on the neighbors as:

$$\delta(G(v)) = \max \{G(v_i), v_i \in N_E(v) \cup \{v\}\}.$$

Classically, opening γ is defined as the result of erosion followed by dilation using the same SE:

$$\gamma(G(v)) = \delta(\varepsilon(G(v))).$$

Similarly, closing φ is defined as the result of dilation followed by erosion with the same SE:

$$\varphi(G(v)) = \varepsilon(\delta(G(v))).$$

The geometrical action of the openings and closings transformations, $\gamma(G(v))$ and $\varphi(G(v))$ respectively, produce a growing or shrinking of the selected graph. Of course, this fusion process can be regulated using parameters for the opening and closing, but also we can regulate the fusion depending on the mentions, tweet favorites or retweets. The graph has to be updated to keep aggregating the different nodes always applying the morphological transformations of $\gamma(G(v))$ and $\varphi(G(v))$ until their parameter value is reached.

For merging two adjacent nodes in a graph, certain V conditions should be verified. We can define some mention parameters that condition the difference between these values of two adjacent nodes that can be aggregate at the opening and closing operations. These parameters are called the minimal mention parameter d_1 and the maximal mention threshold d_2 . To use them, we should calculate, in a first time, the mention differences in the graph. So, we calculate $d_1(G(V_i), \max(G(V)))$, the difference between the maximum value of mentions in the neighboring nodes, and $d_2(G(V_i), \min(G(V)))$ the minimal difference. If the maximal mention parameter is higher than d_1 , the opening operation $\gamma(G(v_i))$ does not merge nodes. Also, if the minimal mention parameter is higher than d_2 , the closing operation $\varphi(G(V_i))$ does not merge nodes. A loop is he required to perform all the necessary aggregations for the simplification of the graph. In figures 4-9 we show the simplification process in different steps.

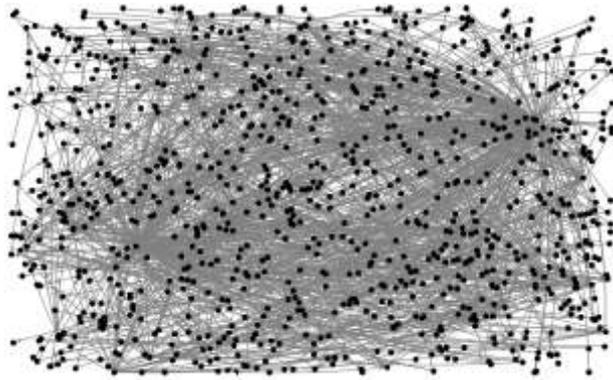


Fig. 4. Graph iterations = 5 Morphological operations applied = γ and φ , $d_1 = 50$, $d_2 = 30$, Nodes = 1535 , Interactions = 2045.

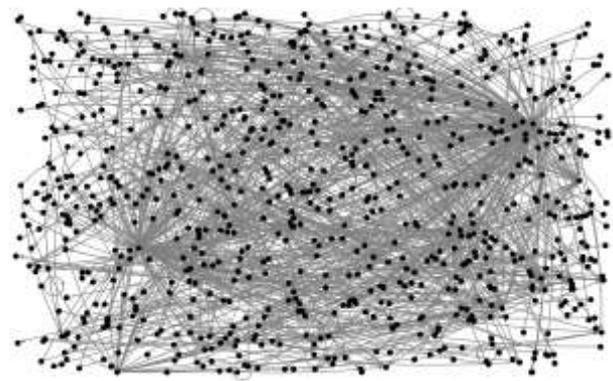


Fig. 5. Graph iterations = 20 Morphological operations applied = γ and φ , $d_1 = 50$, $d_2 = 30$, Nodes = 1381, Interactions = 1715.

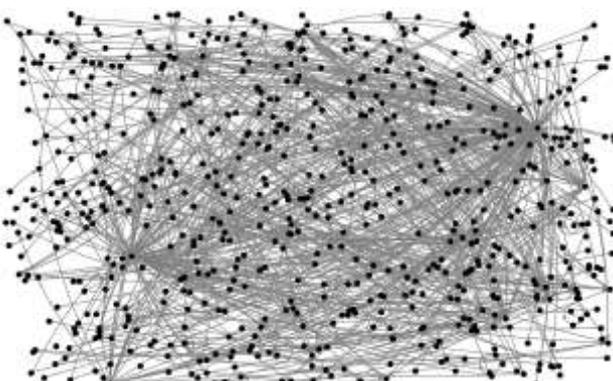


Fig. 6. Graph iterations = 35 Morphological operations applied = γ and φ , $d_1 = 50$, $d_2 = 30$, Nodes = 925, Interactions = 1251.

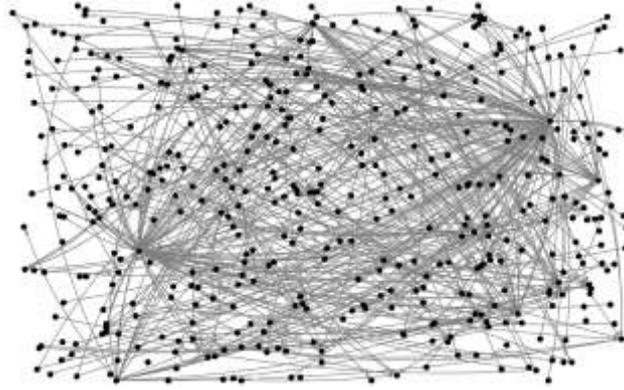


Fig. 7. Graph iterations = 50 Morphological operations applied = γ and φ , $d_1 = 50$, $d_2 = 30$, Nodes = 509, Interactions = 563.



Fig. 8. Graph iterations = 65 Morphological operations applied = γ and φ , $d_1 = 50$, $d_2 = 30$, Nodes = 221, Interactions = 330.

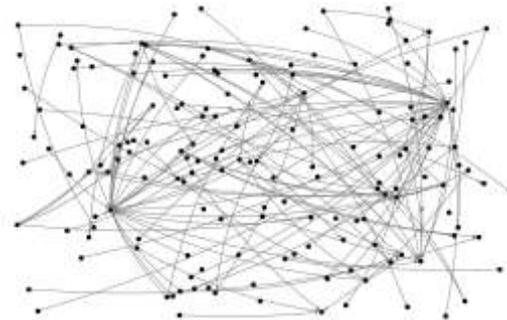


Fig. 9. Final graph. Graph iterations = 80 Morphological operations applied = γ and φ , $d_1 = 50$, $d_2 = 30$, Nodes = 118 , Interactions = 105.

In figure 10, we show the simplification rate when the morphological operators are applied. It is interesting to note that simplification is more significant in the first iterations, usually in the first 5 iterations, which is normal regarding the parameters d_1 and d_2 used. Then, the parameters do not cause so much effect and the simplification rate remains stable. It is at that time that the iterations and the use of morphological operators must stop.



Fig. 10. Simplification rate using morphological operators.

The computation time needed to perform simplification is not significant. Each graph simplification having 5000 nodes requires about 5 seconds to apply morphological operations and compute the entire network. Even if the parameters chosen d_1 and d_2 were widely dispersed, the time is up to 15 seconds to complete the computation process on an Intel Core i9.

3.4 Network Information Retrieval

The final node characteristics is calculated using the final graph after the use of the morphological transformations of $\gamma(G(v))$ and $\varphi(G(v))$. These characteristics $\{C\}$ are then stored separately in a database, which is useful to make meanly queries.

These features $\{C\}$ called “metadata” [25, 26] characterizing each node are then stored and handled separately.

There are two different features extracted from the graph: (i) “node properties”, that are specific to each node (user name, friends, followed, followers, tweet number, image profile and location, etc.) and (ii) “interaction characteristics”, that describe the tweet (mentions, tweet content, tweet favorites and retweets).

The graph G is then considered as a set of layers information $\{L\}=(l_1, \dots, l_n)$. Each layer is associated with a set of characteristics $\{C\}$, with their values $\{v\}$, which defines the structure of each instance stored according to the layer. Each instance is identified by a unique node identifier O_{id} and a descriptor called the value $v(v=(v_1, \dots, v_n))$.

To retrieve information from the final graph we decided to use *Cypher* [27] that is a declarative graph query language that allows for expressive and efficient querying and updating of the graph store. *Cypher* is a simple but powerful query language. This language allows you to focus on the domain instead of getting lost in graph database access.

Being a declarative language, *Cypher* focuses on the clarity of expressing “what” to retrieve from a graph, not on “how” to retrieve it.

Retrieving information is made through a rule expressed by a query [28]. Defining a query relies on the formal notions of:

$$Q = (Oid, L, C),$$

with *Oid* the node, *C* the conditions of the instances in *L*. The equivalent query via the *Cypher* query language would be:

```
MATCH (L:Node{name:'Final-node'})  
WHERE tweet.likes > 300 and tweet.mentions >5  
RETURN (Oid)
```

So, to retrieve meanly information, we have to select the node or the interaction that are interested to us. We have tried different queries using *Cypher* with very interesting results. As an example, we show network and the node retrieved using this Query (Figure 11).

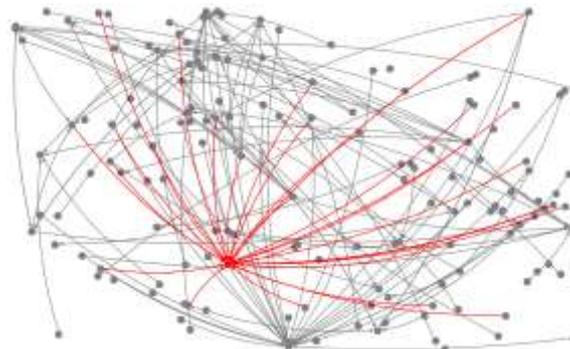


Fig. 11. Automatic selection using *Cypher* language.

6 Conclusions

Complex graphs, contains thousands of nodes of high degree, that are difficult to visualize. Displaying all of the nodes and edges of these graphs can create an incomprehensible cluttered output. We have presented a simplification algorithm that may be applied to a complex graph issue of a Social Media Network, in our case a Twitter network, in order to produce a simplify graph. This simplification was proposed by the use of morphological operators, that are based on Mathematical morphology.

We have represented the Social Media Network as a complete Lattice. In doing this, mathematical morphology has been developed in the context of a relation on a set. It has been shown that this structure is sufficient to define all the basic operations: dilation, erosion, opening and closing, and also to establish their most basic properties.

The simplification of the graph provides an approach to visualizing the fundamental structure of the graph by displaying the most important nodes, where the importance may be based on the topology of the graph and their interaction. The simplification algorithm consists in the iterative use of Opening and Closing operations that cause a growing or shrinking effect in the graph. This process generates the simplification of the network.

As can be seen from this paper, SMA have been and currently are a prominent topic in Network s analysis and simplification. With the advent of the so-called Big Data, we expect this trend to be extremely persistent [29] and promising for opening novel research directions. Indeed, there is no reason to restrict the application of this simplification process the very same ideas we have described here to networks. Any kind of data can be processed with these techniques, notably, image processing.

A special note concerns the relation of community detection to the emergence of core periphery structure in Social Media networks. The process explained in this article, focuses on simplification Network and not in community detection, though we believe that using the correct parameters and exploiting on a correct way the network query language proposed (*Cypher*) some interesting contribution and direction could be envisaged.

In the proposed method based on morphological simplification, we have realized that the parameterization is a fundamental step and we must dedicate special attention to get a homogeneous simplification of nodes and interactions.

This parameterization leads the process of simplification by physical characteristics of the graph and permits to interpret in a simple way the relationship among the nodes, interactions and all characteristics associated.

The notions of dilation, erosion, opening and closing for graph simplification described here appear to be novel, and there is further work needed in order to determine the algebraic properties of these operations. In terms of practical applications, a theory of graphs, perhaps in connection with other forms of granularity for graphs, might be used to model networks at multiple levels of detail.

Future work may be to design query-based simplification techniques that would take user's interests into account when simplifying a network. It would also be interesting to combine different network abstraction techniques with network simplification, such as a graph compression method to aggregate nodes and interactions. Also, it would be interesting to develop additional importance metrics, as well as testing and evaluating our approach with other simplification methods and on other types of graphs.

References

1. Serra, J., Soille, P.: *Mathematical Morphology and its applications to image processing*. Kluwer Academic Publishers (1994)
2. Serra, J.: A lattice approach to image segmentation. *Journal of Mathematical Imaging and Vision*. (24), 83–130 (2006)
3. Bonato, A.: *A Course on the Web Graph*, American Mathematical Society Graduate Studies Series in Mathematics, Providence, Rhode Island (2008)

4. Facebook statistics, <http://www.facebook.com/press/info.php?statistics>, last accessed 2017/07/01
5. Twitaholic, <http://twitaholic.com/>, last accessed 2017/07/01
6. Kleinberg, J.: The small-world phenomenon: An algorithmic perspective. In: Proceedings of the 32nd ACM Symposium on Theory of Computing (2000)
7. Adamic, L. A., Buyukkokten, O., Adar, E.: A social network caught in the web. *First Monday*, 8 (2003)
8. Java, A., Song, X., Finin, T., Tseng, B.: Why we twitter: understanding microblogging usage and communities. In: Proceedings of the Joint 9th WEBKDD and 1st SNA-KDD Workshop (2007)
9. Kumar, R., Novak, J., Tomkins, A.: Structure and evolution of on-line social networks. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2006)
10. Newman, M. E. J. Park, J.: Why social networks are different from other types of networks. *Phys. Rev. E* 68 036122 (2003)
11. Qiu, H., Hancock, E.: Graph simplification and matching using commute times. *Pattern Recognition* 40, 2874–2889 (2007)
12. Frishman, Y., Tal, A.: Multi-level graph layout on the GPU. *Transactions on Visualization and Computer Graphics* 13, 6 Nov, 1310–1319 (2007)
13. Rizzi, S.: A genetic approach to hierarchical clustering of Euclidean graphs. *Pattern Recognition* 2, 1543–1545 (1998)
14. Ban, T., Sen, D.: Graph based topological analysis of tessellated surfaces. In Proc. Eighth ACM Symposium on Solid Modeling and Applications, 274–279 (2003)
15. Kao, M., Occhiogrosso, N., Teng, S.: Simple and efficient graph compression schemes for dense and complement graphs. *Journal of Combinatorial Optimization*, 2(4), 351–359 (1998)
16. Girvan, M., Newman, M. E. J.: Community Structure in Social and Biological Networks. *PNAS* 99(12), 7821–7826 (2002)
17. Davood R., Stephen, C.: Effectively Visualizing Large Networks Through Sampling. 16th IEEE Visualization 2005, (VIS 2005), 48 (2005)
18. Gilbert, A., Levchenko, K.: Compressing network graphs. In Proc LinkKDD 04 (2004)
19. Serra, J.: *Image Analysis and Mathematical Morphology*, Academic Press, London (1982)
20. Heijmans, H.: *Morphological Image Operators*. Advances in Electronics and Electron Physics Series, Academic Press, Boston (1994)
21. Najman, L., Talbot, H.: *Mathematical Morphology: from Theory to Applications*. ISTE-Wiley (2010)
22. Vincent, L.: Graphs and mathematical morphology. *Signal Processing* 16, 365–88 (1989)
23. Flouzat, G., Amram, O. : Segmentation d’images satellitaires par analyse morphologique spatiale et spectrale. *Acta Stereologica*, 16, 267–274 (1997)

24. Zanoguera, F.: Segmentation interactive d'images fixes et séquences séquences vidéo basée sur des hiérarchies de partitions. Thèse de Doctorat en Morphologie Mathématique, ENSMP (2001)
25. Amous, I., Jedidi, J., Sèdes, F.: A contribution to multimedia document modeling and organizing. 8Th International conference on Object Oriented Information Systems, OOIS'02, Springer LNCS. 2425, 434–444 (2002)
26. Chrisment, C., Sedes, F.: Multimedia Mining, A Highway to Intelligent Multimedia Documents. Multimedia Systems and Applications Series, Kluwer Academic Publisher, V. 22, 245 (2002)
27. Holzschuher, F., Peinl, R.: Performance of graph query languages: comparison of cypher, gremlin and native access in Neo4j. In Proceedings of the Joint EDBT/ICDT Workshops. 195–204, ACM (2013)
28. Lafferty, J., Zhai, C.: Document language models, query models, and risk minimization for information retrieval. In ACM SIGIR Forum. 51(2). 251–259 (2017)
29. Zwaenepoel, W.: Really Big Data: Analytics on Graphs with Trillions of Edges. In LIPIcs-Leibniz International Proceedings in Informatics, 70 (2017)