A procedure for detection of border communities using convolution techniques.

José Miguel Montañana[♭,1], Antonio Hervás[♮,2],
Samuel Morillas[§], and Pedro Pablo Soriano[◇]

(♭) inpeek GmbH, Germany.
Rheinstraße 40-42, Westtrakt, 64283 Darmstadt.
(♮) IUMM. Universitat Politècnica de València, València, Spain,
(§) IUMPA Universitat Politècnica de València, València, Spain.
(◇) SEPyC. Universitat Politècnica de València, València, Spain,

Abstract

Many complex systems can be modeled by graphs and networks, [4]. In some problems, the study of communities allows quantitative and qualitative approaches and obtaining some knowledge about the structure of the graph and what it represents [1, 3, 5]. There is extensive literature in the study of communities, mostly focused on non-directed graphs [3, 5]. In our case we focus our work on the study of communities in directed graphs, weakly connected, with weights on the edges.
In MME&HB 2016, was presented an algorithm for detection of directional communities in a directed graph [1], with a special interest in the graph representing the process of access to the Spanish Public University System, (SUPE) [1, 2]. The proposed algorithm allowed to obtain communities that provided an approximation to the problem. In MME&HB 2020 we propose a new algorithm based on obtaining the centers of the graph and pruning non-significant edges [6]. Recently a method for obtain communities using convolution techniques was presented in [5].
In this paper, we propose a community detection algorithm that combines a method for calculating potential community centers and convolution techniques to prune non-significant edges.

1 Introduction

Many complex systems in the real world in which there are interaction between the elements can be modeled as graphs or networks.
We can find networks where the elements are highly related among them. This means that there are many more edges than nodes -dense networks-. It is more usual having networks where the number of edges is much smaller than they could have at most: dispersed networks.
Another characteristic element we can observe is the existence of certain groupings of nodes highly connected between them and poor connected with the rest. These groupings constitute what are

[1]jmmontanana@gmail.com
[2]ahervas@imm.upv.es

called communities. Nodes grouped in the same community have common characteristics that make them play a certain role within the network.
Several techniques and algorithms have been developed for its detection and study, and it is a constantly evolving area of work. Depending on the criteria algorithm, we obtain different communities.

2 Objetive

For directed, weakly connected, and bad-conditioned graphs, the algorithms to find communities usually do not work satisfactorily. They are typically designed for well-conditioned and undirected graphs. Examples of this kind of graphs are the problem of reassignment of students in grades, trophic chains, traffic in airports, and public transport networks.
In particular, we consider the directed and bad-conditioned networks. In those networks, there are communities with edges whose weight is small in comparison with the weight of edges in other communities in the same graph. In addition, there are edges connecting different communities which make impossible for many algorithms to identify correctly the communities in the graph.
In this paper is to propose a pruning method that eliminates as many as possible of those later edges connecting different communities. We refer to them as *noise edges*. That will easy the task of the community detection algorithms, or even don't need them if the pruning method is able to make the communities disjoint between them.

1. The algorithm must prune the graph by removing noise edges.
2. The algorithm must to be efficient and fast to be applied in large size graphs.
3. In the ideal case, the modularity keps after apply the algorithm at suitable values. (Close to 1).

3 Methology

The method is fundamentally based on the edge detection method in images based on the convolution product. To explain our method we are going to introduce it first, and then we will show how we adapt it to the case of directed graphs.
The edge detection method on a gray scale image takes as the input image as a matrix of real numbers. The process consists of recursively taking $m \times m$ rectangles from the input matrix and multiplying them by a $m \times m$ constant matrix which is referred as Kernel. See figure 1.

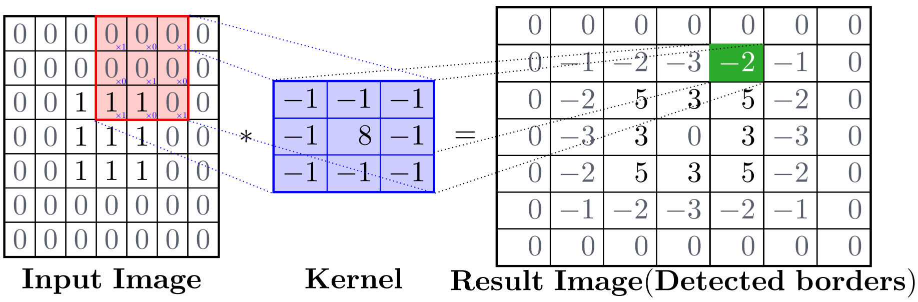


Figure 1: Convolution product for the detection of borders. Positive values in the result correspondes to the borders detected.

The borders detected correspond to the pixels in the result image which value is positive.. The figure 2 shows an example of detection of borders with this method.

(a) (b)

Figure 2: (a) Initial image and (b) the detection of borders with the convolution product.

Convolution product applied to directed graphs

The product of convolution applied to directed graphs corresponds to a sumation of product of weights of edges with a kernel which is defined as a vector of constant real numbers: $\{k_0, k_1, k_2, \ldots\}$. This operation needs to represent the graph in a kind of irregular n-dimensional structure. To do it, each link or edge[3] has to get assigned a set of indexes which are calculated based to the distance or number of hops to reach the other edges. Notice that calculation of the hops considers the direction of the edges.

As an example, the figure 3 shows a graph where edges are labeled with the number of hops to reach the edge in the right side labelled as a. The edges in the figure are labelled as $w_{destination,hops,id}$, where *destination* is the edge for which are calculated the hops, and the *id* is just a index used for counting.

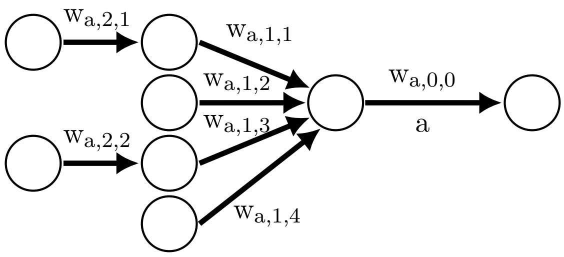


Figure 3: Edges labelled based in hops hops to reach the link a.

Then, the result of convolution product on directed graphs for each edge corresponds to the summation of the product of the weights of the edges $w_{x,h,id}$. The equation 1 shows the result of the convolution product for a particular edge a.

$$result_a = w_{a,0,0} * k_0 + \sum_i w_{a,1,i} * k_1 + \sum_{j1} w_{a,2,j} * k_2 \quad (1)$$

In this paper, we propose to use the detection of borders using the convolution product, but using square the weights. Then, the equation 2 corresponds to the update of the equation for the edge a.

$$result_a = w^2_{a,0,0} * k_0 + \sum_i w^2_{a,1,i} * k_1 + \sum_j w^2_{a,2,j} * k_2 \quad (2)$$

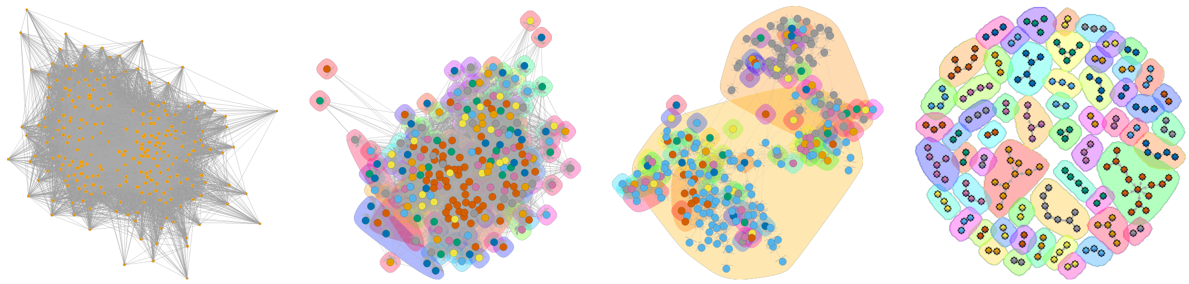
Calculation of the border of the communities in directed graphs, and prune process

The edges that correspond to the border of the communities in the directed graph correspond to those which convolution product result is negative. In this paper, we consider that those edges to be pruned before process to detect the communities in the graph.

[3]Links are referred as edges in the graph literatrure.

4 Results

In this section are shown the results of pruning edges with three different criteria.
The graph 4.a shows the initial graph ....(explicar) [?]
The first two cases corresponds to pruning the edges in the initial graph which convolution product is negative when using the *Laplacian of the Gaussian* and *Log* kernels respectively. The results for these two cases show low values of modularity, which means that the pruning process was not able to separate correctly the communities in the graph. The R software suite is not able to represent the corresponding graphs without overlapping communities due to the large number of interconnections between the communities. Representation of those graphs appear in the figures 4.b and 4.c.
However, the results obtained when pruning edges based on the convolution product using the square of the weights of the edges and the Log kernel shows a high modularity. In addition, we should mention that all the edges pruned on the same graph when applying the In-Out-method [1?] are also pruned in this case. We consider that the proposed method in the case worked correctly because the similar results obtained in the [1?] had been proved to be correct at [2?], where the are used to generate a model that predicted correctly the student's selection of university in following years.
The computation times when applying the method to the graph with 9317 edges and 215 vertices is extremely low, in the order of two seconds. While, the computation with the same of the Girman-Newman algorithm for the detection of communities [3?] in the same graph needs .... t.



(a) (b) (c) (d)

Figure 4: (a) Initial image and (b) the detection of borders with the convolution product.

Equationon	Kernel	Time	Graph	Nodes	Edges	**Modularity**
1.1	Laplacian of the Gaussian	2.250 ms	4.b	215	9283	0.020
1.1	Log	1.920 ms	4.c	215	908	0.307
1.2	Log	2.170 ms	4.d	215	265	**0.970**

5 Conclusions

Considering the results, the use of convolution products seems an appropriate option for obtaining communities. Our intention is to continue with this line of work, firstly apply FUZZY filters, studying how behaves with large and badly conditioned directed graphs, determine the properties of the communities so obtained and analyze their modularity, and finally, performs confirmation tests with synthetic graphs.

References

[1] Hervás A., An algorithm to obtain directional communities in a directed graph. (2016). Proceedings of the Mathematical Modelling in Engineering & Human Behaviour. pp 347-352.

[2] Hervás A, Soriano P.P., Jiménez, A., Peinado J., Capilla R. and Montañana J.M. (2017) Applying a graph model for the Spanish public university system in Modeling Human Behavior: Individuals and Organizations. (Nova Science Publishers, Inc.), pp. 9–24.

[3] Lancichinetti A, Fortunato S (2009) Community detection algorithms: A comparative analysis. Phys. Rev. E 80(5):056117.

[4] Feng S, Hu B, Nie C, Shen X (2016) Empirical study on a directed and weighted bus transport network in China. Physica A: Statistical Mechanics and its Applications 441(C):85–92.

[5] Muñoz, H., Vicente, E., González, I., Mateos, A., and Jiménez-Martín, A. (2021). ConvGraph: Community Detection of Homogeneous Relationships in Weighted Graphs. Mathematics, 9(4), 367.

[6] Montañana, J. M., Hervás A. and Soriano P. P. (2020) Communities' detection in weakly connected directed graphs, Modelling for Engineering & Human Behaviour, pp. 72-77.