

## Combinations of Affinity Functions for Different Community Detection Algorithms in Social Networks

Javier Fumanal-Idocin<sup>1</sup>, Oscar Cordón<sup>2</sup>, María Minárová<sup>3</sup>, Amparo Alonso-Betanzos<sup>4</sup>, Humberto Bustince<sup>1</sup>  
<sup>1</sup>Public University of Navarra, Pamplona, Spain, <sup>2</sup>Slovak University of Technology in Bratislava, Bratislava, Slovakia  
<sup>3</sup>University of A Coruña, A Coruña, Spain <sup>4</sup>University of Granada, Granada, Spain  
[javier.fumanal@unavarra.es](mailto:javier.fumanal@unavarra.es)

### Abstract

*Social network analysis is a popular discipline among the social and behavioural sciences, in which the relationships between different social entities are modelled as a network. One of the most popular problems in social network analysis is finding communities in its network structure. Usually, a community in a social network is a functional sub-partition of the graph. However, as the definition of community is somewhat imprecise, many algorithms have been proposed to solve this task, each of them focusing on different social characteristics of the actors and the communities. In this work we propose to use novel combinations of affinity functions, which are designed to capture different social mechanics in the network interactions. We use them to extend already existing community detection algorithms in order to combine the capacity of the affinity functions to model different social interactions than those exploited by the original algorithms.*

### 1. Introduction

Social network analysis has become an important part of many disciplines such as physics [1], public health and administration [2, 3], and biology [4]. The idea behind studying complex systems using this technique is that the elements composing such systems are part of multi-lateral interactions and processes, which affect deeply the system and each of its composing individuals. Many problems in computer science have been tackled using social network analysis [5]. This is the case of information propagation [6], leader detection [7], recommendation systems [8, 9] and decision making [10, 11]. One of the most popular problems of this kind is community detection.

A community in a social network is a group of nodes that present a significant interaction among them, and much less with the rest of the network. Community detection has been used to identify groups of friends

[12], a protein complex [13], and to discover routing strategies [14]. Many algorithms have been proposed to solve community detection, a great deal of them using the concept of modularity to measure how good are the sub-partitions detected [15]. The most popular community detection algorithm is the Lovaine algorithm [16], which uses changes in modularity to obtain the optimal partition. Other popular algorithms that use the idea of modularity are the proposal in [15], which is a greedy algorithm designed to quickly reach a solution, and the proposal in [17], in which the authors propose to iteratively eliminate edges in a network using the idea of edge betweenness. Further approaches include using deep learning to generate an embedding of the graph and a reconstructed adjacency matrix in order to obtain a representation where the community structure can be seen clearly [18]; spreading the labels in the network as if it was an epidemic [19]; or using an equation, called the map equation, to model the information flows of the network [20].

Another proposal that does not include a modularity optimization process is [21], in which the authors propose to use a new kind of functions, the affinity functions, that model the local interactions between actors according to different social behaviours. They do so in order to propose a community detection algorithm, the Borgia Clustering, that simulates the dynamics of the conquests of Cesare Borgia in the Italian Renaissance. However, the possibilities of using affinity functions with other community detection algorithms were not studied. This possibility is of spatial relevance because affinity functions could improve the network features that some community detection algorithms exploit. In fact, some algorithms compute different representations of the network using computationally expensive processes, like a convolution [18]. Affinity functions could work in a similar way in those algorithm that do not compute other representations for the original network. Besides, although some combination of affinity functions were proposed in [21], some mathematical properties of this combination could be

relaxed in order to explore further results.

Taking these considerations into account, in this work we aim to:

1. Extend the convex combination of affinity functions to a more general expression.
2. Explore the possibilities of using affinity functions with community detection algorithms other than the Borgia Clustering.

In order to achieve our aims, we use different combinations of affinity functions without restricting them to be convex combinations, and then we study these combinations with three of the most popular modularity-based community detection algorithms: the Lovaine algorithm [16], the Greedy modularity algorithm [15] and Girvan-Newman algorithm the [17], and compare the results with the Borgia Clustering algorithm.

The rest of the work follows this structure: in Section 2 we describe the affinity functions used and how we combine them, and we also discuss the community detection algorithms used in our experimentation. In Section 3 we discuss the datasets used in our experimentation and the results obtained. Finally, in Section 4 we draw our conclusions for this work and the future lines for our research.

## 2. Methods

In this section we recall the notion of affinity function, the functions studied and the algorithms tested to perform community detection.

### 2.1. Affinity functions

Affinity functions were proposed in [21] to measure the strength of the relationship between a pair of actors in a social network by capturing the nature of their local interactions. They are defined as functions that take as input two different actors,  $x$  and  $y$ , and return a number in the  $[0, 1]$  interval:

$$F_C : (x, y) \rightarrow [0, 1]$$

where  $C$  is the adjacency matrix whose each entry  $C(x, y)$  quantifies the relationship for the pair of actors  $x, y$  in a weighted network. The affinity value will be higher or lower depending on which aspect of the relationship we are taking into account, like the number of friends in common or the relative strength of the interaction between  $x$  and  $y$ . Some affinity functions are:

- *Best friend affinity*: it measures the importance of a relationship with an actor  $y$  for the actor  $x$ , in relation to all the other relationships of  $x$ :

$$F_C^{BF}(x, y) = \frac{C(x, y)}{\sum_{a \in V} C(x, a)} \quad (1)$$

- *Best common friend affinity*: it measures the importance of the relationship taking into account how important are the common connections between the connected nodes to  $x$  and  $y$ , in relation to all other relationships of  $x$  in the network:

$$F_C^{BCF}(x, y) = \frac{\max_{a \in V} \{\min\{C(x, a), C(y, a)\}\}}{\sum_{a \in V} C(x, a)} \quad (2)$$

- *Machiavelli affinity*: it computes how affine two actors  $x$  and  $y$  are based on how similar is the social structure that surrounds them:

$$F_C^{Mach}(x, y) = 1 - \frac{|I_x - I_y|}{\max\{I_x, I_y\}}, \quad (3)$$

where  $I_a = \sum_{z \in Z} D(z)$ , where  $Z$  is the set of actors where  $C(a, z) > 0, \forall z \in Z$ , and  $D(z)$  is the centrality degree of  $z$ .

Regarding the interpretation of the affinity value obtained, a 0 value means that no affinity has been found at all while an 1 value means that there is a perfect match according to the analyzed factors. Since affinities are not necessarily symmetrical, the strength of this interaction depends on who the sender and receiver are, as it happens in human interactions. We denote a network resulting from calculating all its edges with an affinity function an “affinity network”.

There are two different kinds of affinity functions: personal and structural affinities. Personal affinities are computed using the edges of the  $x, y$  actors, while the structural affinities use different centrality measures of these actors. Two examples of personal affinity functions are the best friend and best common friend affinities. The Machiavelli affinity is an example of a structural affinity.

Depending on the affinity function computed, the resulting affinity network will have different structural properties. For example, for the case of the ratio of connections in the network with the respect to all the potential ones (commonly called density): the Machiavelli affinity will result in a network with density equal to 1; the best friend affinity will preserve the

same density as in the adjacency network; and the best common friend affinity tends to increase the density with respect to the adjacency network.

More affinity functions and further details about them can be found in [21].

## 2.2. Combinations of affinity functions

The convex combination of two affinity functions is also an affinity function. This proof is straightforward: the result of an affinity function is a number in the  $[0, 1]$  interval. The convex combination of two numbers in the  $[0, 1]$  interval is another number in the same interval. So the operation that computes the convex combination of two affinity functions is also a valid affinity function.

However, some combinations of affinity functions that are not convex are also affinity functions. As long as we guarantee that the output is inside the  $[0, 1]$  interval, the resulting value is a valid affinity function. We are interested in these expressions because some affinity functions tend to have a higher average value than others. Combining affinity functions can be problematic because the values of one of the functions can be irrelevant compared to the other in most cases. One possible solution to this problem is to extremely skew the value of the  $\alpha$  to favour the irrelevant affinity. However, this solution presents two problems:

- This solution results not only in decreasing one affinity, it also increases the other, which is not an indented behaviour. Also, finding the proper equilibrium between the mixing parameters can require an expensive fine-tuning.
- The interpretability of this solution is counter-intuitive. If one of the mixing parameters is much higher than the other, we can interpret that one affinity function is much important than the other in our analysis. However, this would not be the case, as we did not increase on purpose that parameter, we only tried to decrease the average value of the other affinity function.

Using a non-convex combination, might solve both of these problems. By simply reducing one mixing parameter without affecting the other, we are able to scale both affinity functions to an appropriate average value without performing an expensive fine-tuning or affecting the interpretability of the combination.

To do so, we start by substituting the expression of a convex combination of two affinity functions:

$$F_C^{CC} = \alpha F_C^{BF}(x, y) + (1 - \alpha) F_C^{BCF}(x, y) \quad (4)$$

with  $\alpha \in [0, 1]$ , for the linear combination:

$$F_C^{LC} = \alpha F_C^{BF}(x, y) + \beta F_C^{BCF}(x, y) \quad (5)$$

with  $\alpha, \beta \in [0, 1]$ .

However Eq. (4) might not be always an affinity function, so we establish the constraint that:

$$\alpha + \beta \leq 1 \quad (6)$$

Of course, if  $\alpha + \beta = 0$ , then we obtain a matrix of zeros as result. Although that result it is technically a valid affinity function, we also add another constraint:

$$\alpha + \beta > 0 \quad (7)$$

In this way, if  $\alpha + \beta = 1$  we recover a convex combination, and if not, we are obtaining a value lesser than the one obtained using a convex combination, so it will never be above 1. We denote the expression Eq. (5), as long as it holds Eq. (6) and Eq. (7), a less than-convex combination.

It is straightforward that the less than-convex combination of two affinity functions is another valid affinity function, because it will always be lower than the convex combination of affinity functions (which is a valid affinity function), and always bigger than 0.

## 2.3. Community detection algorithms

For our experimentation we have studied three very popular community detection algorithms based on different modularity optimization processes:

- *Lovaine community detection algorithm* [16]: this method optimizes the modularity of the community structure found as the algorithm progresses. The algorithm is divided in two phases: (1) First, small communities are found by optimizing modularity locally on each node. We do so by computing the changes in modularity when each node  $i$  is joined in the same community as each of their neighbours, and choosing the biggest positive change in modularity for each node. We do so for each node until no positive change in modularity is possible. (2) Then, each of these communities is grouped into one node and the process repeats iteratively.
- *Girvan-Newman community detection algorithm* [17]: this algorithm progressively removes edges from the original network based on the idea of how likely an edge is to be a bridge between communities, which is called “edge betweenness”. The algorithm repeats iteratively this process until there are no more edges: (1) We

compute all the edge betweenness. (2) We remove the one with the highest value. Finally, we obtain a dendrogram where the leaves are the individual nodes.

- *Greedy modularity community detection algorithm* [15]: is a greedy algorithm designed to reach a solution in few steps. It works by optimizing the modularity value of each partition in the graph. Each node starts as the sole member of its community. Then, we select the edge from the original graph that has the highest positive modularity change. We do so iteratively, until only all the nodes are connected. Then, we choose the partition with better modularity as the final result.

We chose these algorithms as they are very popular, and the concept of modularity is one of the most popular ones in community detection in social network analysis. The procedure to use them with affinity functions is the same for the three algorithms: we compute the affinity function or the combination of affinity functions on the original adjacency matrix, and then, we compute the algorithm on the resulting affinity network.

For the sake of comparison, we have also compared the results obtained with these algorithms with those obtained by the Borgia Clustering [21]. The Borgia Clustering is a community detection algorithm that generalizes the gravitational clustering algorithm [22] using affinity functions, among other changes. This algorithm was explicitly designed to work with affinity functions, so we think that it is interesting to compare the results obtained using this algorithm with the rest of the community detection algorithms using affinity functions.

### 3. Experimental results

In this section we have compared the results performing community detection in three datasets:

- Zachary Karate Club [23]: is a social network of a university karate club that consists of 34 people and how they split into two communities.
- Dolphin [24]: a social network of bottlenose dolphins, where each link represents the frequency in which they played.
- Polbooks [25]: nodes represent books about US politics sold by Amazon. Edges represent frequent co-purchasing of books by the same buyers, as indicated by the "customers who bought this book also bought these other books" feature on Amazon.

### 3.1. Performance metrics

Normalized Mutual Information (NMI) is a normalization of the Mutual Information score to scale the results between 0 (no mutual information) and 1 (perfect correlation). We have used the NMI to evaluate the results obtained in community detection against ground truth labels.

**3.1.1. Normalized Mutual Information** Given the conditional entropy of two populations:

$$H(Y|X) = - \sum_x \sum_y p(x, y) \log p(y|x) \quad (8)$$

then, the mutual information of two random discrete variables can be defined as:

$$I(X; Y) = H(X) - H(X|Y), \quad (9)$$

which measures the mutual dependence between  $X$  and  $Y$ . So, the final expression for the NMI is:

$$NMI(X, Y) = \frac{2I(X; Y)}{H(X) + H(Y)} \quad (10)$$

**3.1.2. Modularity** Modularity is a metric of the structure of networks that measures the strength of partition into different communities [26]. Networks with high modularity have dense connections between the nodes within communities but scarce connections between nodes in different ones.

The expression of modularity,  $Q$ , is the following:

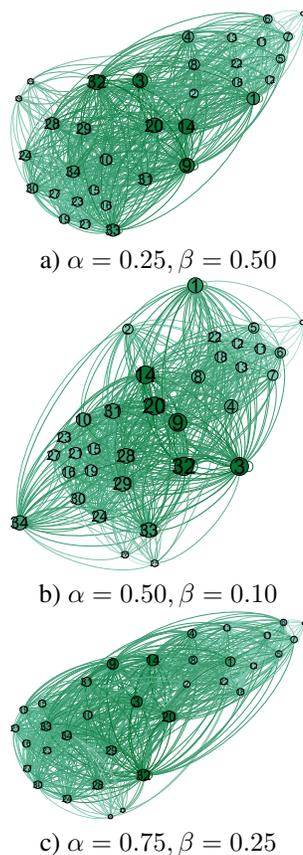
$$Q = \sum_{i=1}^c (e_{ii} - a_i^2) \quad (11)$$

where  $c$  is the number of communities, where  $e_{u,v}$  is the fraction of edge with one end in a community,  $u$ , and another end in community  $v$ , and  $a$  is the fraction of edges with one end in the  $i$ -th community.

### 3.2. The effects of different affinity combinations in the networks

In this subsection we have tested the effects in each dataset using different affinity functions compared to the original adjacency matrix. We expect the best friend affinity not to alter the degree in each node, but other centrality measures may be affected. And for the best common friend affinity, we expect it to increase the degree in each node, which might reduce other centrality measures, specially the betweenness, because more edges will be present in the network.

An example of three different affinity functions applied to the Zachary karate network is shown in Figure 1. We can see that adding the best common friend affinity did increase the number of edges and the network, and the role that some actors present in the network varies significantly depending on the affinity function used. We expect the results in community detection to change considerably since the centrality measures of some actors change significantly. Specially relevant are the changes of actors “34” or “2”. It is also noteworthy how some actors do not change their role in their network independently of the affinity used, like “17” or “8”. In this particular case, we can see that even though there are clearly two communities in three affinity networks, the border between them varies significantly between them since some actors in between the communities seem to differ their position remarkably in the different affinity networks.



**Figure 1. Visual representation of the Zachary Karate club Network using three different combinations of affinities, using the parameters indicated for each one. We can observe that two communities are clearly present in the three networks, although the frontier between both is more clearly present in (b) compared to (a) and (c).**

### 3.3. Community detection experiments

In this subsection we have evaluated the results obtained using the combinations of affinity functions with community detection algorithms other than the Borgia Clustering, and then we have compared them with those obtained using this algorithm. We use the Modularity as internal index and the NMI as an external one to evaluate our results.

In Table 1 we have displayed the results obtained for each dataset and algorithm using the original adjacency matrix. We can see that the Greedy Modularity algorithm obtained the best average results in the NMI index and the Lovaine algorithm did so for the Modularity index.

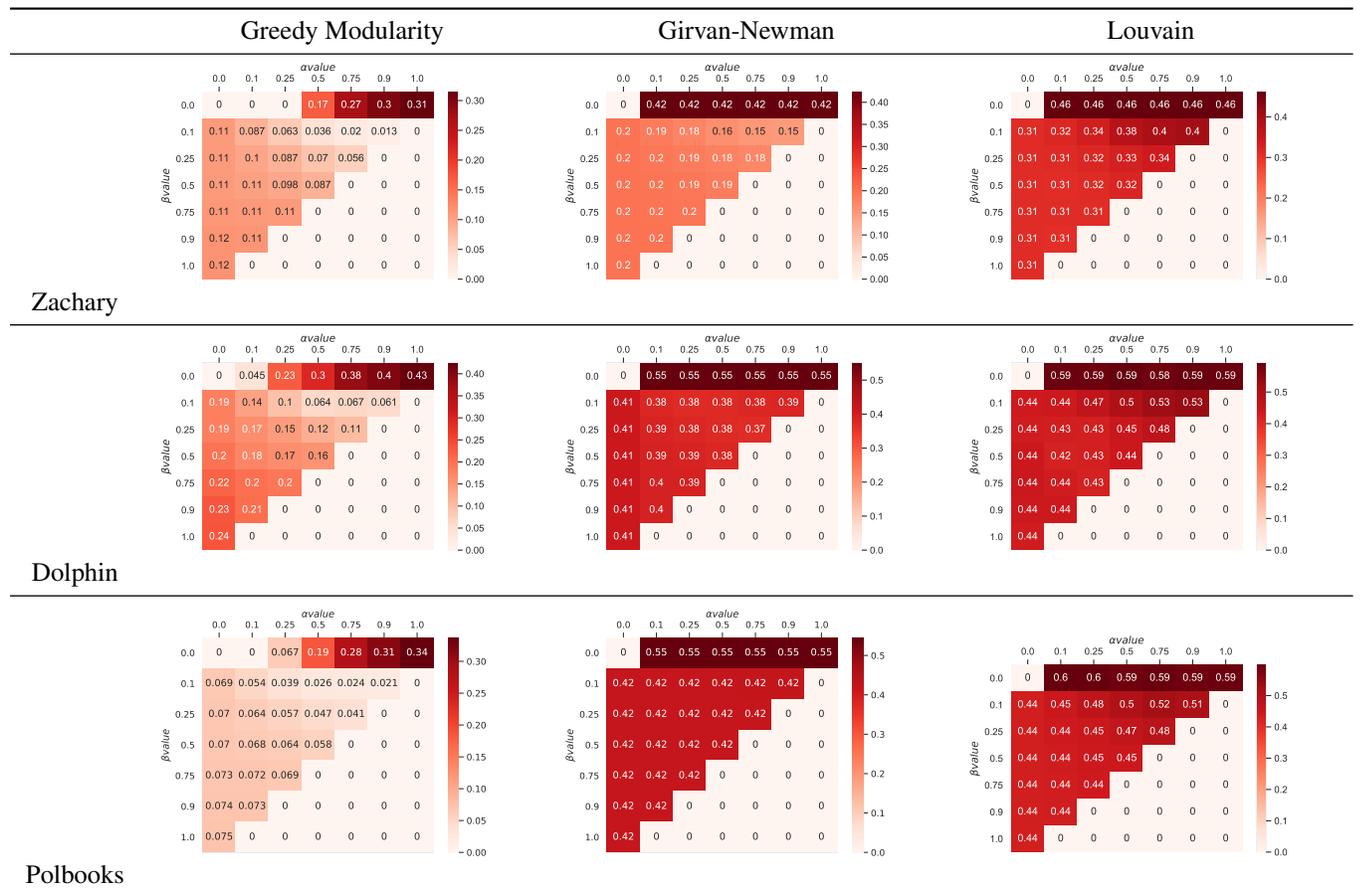
We have studied how these indexes change when we apply the three algorithms using the affinity matrix resulting when using the combinations proposed in Eq. (5) instead of the original adjacency matrix. We have reported the modularity results in Figure 2 for the modularity index, and in Figure 3 for the NMI index.

A summary of the best results can be found in Table 2, alongside the results of the Borgia Clustering for the same trials. We found that in most cases the best friend affinity obtains the best results, specially in the case of the Greedy Modularity and the Girvan-Newman algorithms. However, in the case of the Louvain algorithm, non-convex combinations of affinity functions resulted best for the NMI index. Compared to the results obtained using the adjacency matrix, NMI index decreased dramatically for the Dolphin and Polbooks datasets, but modularity seemed to improve in most cases, specially for the Louvain algorithm.

All the studied algorithms seemed to obtain good modularity results, both in the case of the adjacency and the affinity matrices. In fact, these algorithms seemed to over perform the Borgia Clustering in terms of the modularity index, although the Borgia Clustering presented much higher NMI values than the rest of the algorithms, probably because the Borgia Clustering computed correctly the number of real communities for each dataset.

## 4. Results and Discussion

In this experimentation we studied the performance of different community detection algorithms using less than-convex combinations of affinity functions, in three different datasets. We found satisfactory results for the modularity metric in all datasets for all the algorithms tested, and for the two metrics studied in the Zachary karate network. Also, we found the best friend affinity to be the most effective affinity function to perform



**Figure 2. Modularity results for the community detection algorithms using affinity networks. Best friend affinity overperforms the combination with the best friend affinity in all cases. Results are generally better than those obtained with the adjacency matrix according to this metric.**

community detection.

The Louvain algorithm seems to be the most successful classical algorithm when using affinity functions, according to the modularity index. The Girvan-Newman algorithm also obtained a moderate performance increase. On the contrary, the Greedy modularity algorithm performed better with the original adjacency networks compared to the affinity functions tested. None of the studied algorithms performed better than the Borgia Clustering in terms of real world labels, but they did in terms of modularity.

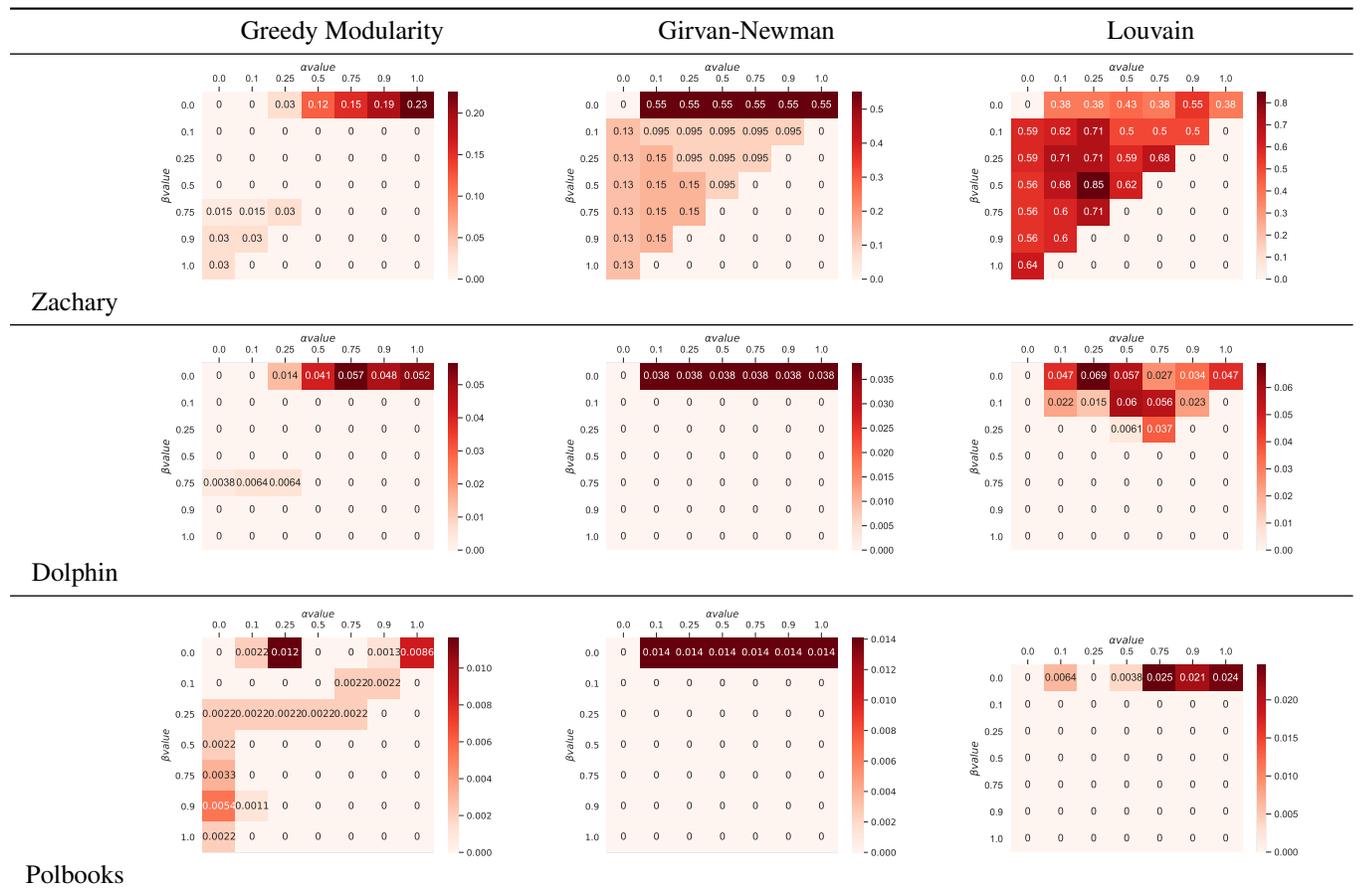
A possible explanation for the disparity between the modularity and NMI indexes performance could be due to the social interactions exploited by the affinity functions, which could be different from the dynamics registered by the ground truth labels. Besides, the behaviour of some of these algorithms can be affected dramatically by the affinity function chosen. For example, the best common friend affinity leads to an increase in the number of edges in the network. This will

for sure affect the performance and execution time of the Girvan-Newman algorithm, that needs to recompute the edge betweenness in each iteration.

## 5. Conclusions

In this work we have proposed a new method to construct affinity functions using non-convex combinations of affinity functions. We have explained how these combinations work and how some centrality measures change based on the mixing factors of the combination. Besides, we have used the affinity functions for the first time in classical community detection algorithms designed to work with the adjacency matrix. In order to test our proposal, we have computed a series of experiments using different combinations of affinity functions, and we have compared the best results obtained with the classical algorithms with the Borgia Clustering.

We found that the use of affinity functions can have



**Figure 3. NMI results for the community detection algorithms usgin affinity networks. We found good results for the Zachary karate network dataset, but underwhelming results for the Dolphin and Polbooks in all cases.**

a positive impact on the modularity metric performance, although evaluation with ground truth labels decreased in most cases. However, in the case of the Borgia Clustering we found the opposite results. The ground truth labels evaluation was in general terms better than the rest of the algorithms but the modularity values were lower.

Future research shall study the relationship between the Borgia Clustering and the rest of the community detection algorithms. As the Borgia Clustering is very expensive to compute in large networks, we are particularly interested in a possible hybrid algorithm that combines the Louvain algorithm and the Borgia Clustering, in order to surpass some of the limitations of the latter in terms of memory and execution time. Also, we shall research the interaction of affinity functions with specific community detection algorithms in order to develop specific functions to enhance the performance for each individual algorithm.

## 6. Acknowledgments

Javier Fumanal Idocin and Humberto Bustince's re-search has been supported by the project PID2019-108392GB-I00 (AEI/10.13039/501100011033).

Maria Minárová research has been funded by the project work was supported by the projects APVV-17-0066 and APVV-18-0052.

Oscar Cordon's research was supported by the Spanish Ministry of Science, Innovation and Universities under grant EXASOCO (PGC2018-101216-B-I00), including, European Regional Development Funds (ERDF).

## References

- [1] M. Newman, "The physics of networks," *Physics today*, vol. 61, no. 11, pp. 33–38, 2008.
- [2] T. W. Valente, *Social networks and health: Models, methods, and applications*, vol. 1. Oxford University Press New York, 2010.
- [3] A. P. Chaves, I. Steinmacher, and V. Vieira, "Social networks and collective intelligence applied to public

Dataset	Algorithm	NMI	Mod.
Zachary	Greedy Mod.	0.45	0.38
	Girvan-Newman	0.35	0.41
	Lovaine	0.35	0.41
Dolphin	Greedy Mod.	0.41	0.49
	Girvan-Newman	0.44	0.51
	Lovaine	0.33	0.51
Polbooks	Greedy Mod.	0.49	0.50
	Girvan-Newman	0.48	0.51
	Lovain	0.56	0.49

**Table 1. Results for community detection for the three datasets considered using three different algorithms. In this case we used the adjacency matrix to compute the network partition.**

Dataset	Algorithm	NMI	Mod.
Zachary	Greedy Mod.	0.31	0.23
	Girvan Newman	0.42	0.55
	Louvain	0.31	0.85
	Borgia Clustering	0.83	0.36
Dolphin	Greedy Mod.	0.052	0.43
	Girvan Newman	0.038	0.55
	Louvain	0.089	0.59
	Borgia Clustering	1.00	0.37
Polbooks	Greedy Mod.	0.086	0.55
	Girvan Newman	0.014	0.55
	Louvain	0.024	0.59
	Borgia Clustering	0.56	0.49

**Table 2. A summary for the best results for community detection computed over affinity networks for the datasets and algorithms studied.**

- transportation systems: A survey," *Viii Sbsc*, pp. 16–23, 2011.
- [4] S. Horvath, *Weighted network analysis: applications in genomics and systems biology*. Springer Science & Business Media, 2011.
- [5] M. Newman, *Networks*. Oxford university press, 2018.
- [6] I. Lobel and E. Sadler, "Information diffusion in networks through social learning," *Theoretical Economics*, vol. 10, no. 3, pp. 807–851, 2015.
- [7] S. M. H. Bamakan, I. Nurgaliev, and Q. Qu, "Opinion leader detection: A methodological review," *Expert Systems with Applications*, vol. 115, pp. 200–222, 2019.
- [8] W. Yu and S. Li, "Recommender systems based on multiple social networks correlation," *Future Generation Computer Systems*, vol. 87, pp. 312–327, 2018.
- [9] J. Wu, S. Wang, F. Chiclana, and E. Herrera-Viedma, "Two-fold personalized feedback mechanism for social network consensus by uninorm interval trust propagation," *IEEE Transactions on Cybernetics*, 2021.
- [10] H. Zhang, F. Wang, Y. Dong, F. Chiclana, and E. Herrera-Viedma, "Social trust-driven consensus reaching model with a minimum adjustment feedback mechanism considering assessments-modifications willingness," *IEEE Transactions on Fuzzy Systems*, 2021.
- [11] E. Herrera-Viedma, F. J. Cabrerizo, F. Chiclana, J. Wu, M. J. Cobo, and S. Konstantin, "Consensus in group decision making and social networks," *Studies in Informatics and Control*, vol. 26(3), pp. 259–268, 2017.
- [12] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.
- [13] O. Mason and M. Verwoerd, "Graph theory and networks in biology," *IET systems biology*, vol. 1, no. 2, pp. 89–119, 2007.
- [14] N. P. Nguyen, T. N. Dinh, Y. Shen, and M. T. Thai, "Dynamic social community detection and its applications," *PloS one*, vol. 9, no. 4, p. e91431, 2014.
- [15] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, p. 066133, 2004.
- [16] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, oct 2008.
- [17] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [18] L. Wu, Q. Zhang, C.-H. Chen, K. Guo, and D. Wang, "Deep learning techniques for community detection in social networks," *IEEE Access*, vol. 8, pp. 96016–96026, 2020.
- [19] S. E. Garza and S. E. Schaeffer, "Community detection with the label propagation algorithm: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 534, p. 122058, 2019.
- [20] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," *The European Physical Journal Special Topics*, vol. 178, no. 1, pp. 13–23, 2009.
- [21] J. Fumanal-Idocin, A. Alonso-Betanzos, O. Cordón, H. Bustince, and M. Minárová, "Community detection and social network analysis based on the italian wars of the 15th century," *Future Generation Computer Systems*, vol. 113, pp. 25–40, 2020.
- [22] W. Wright, "Gravitational clustering," *Pattern Recognition*, vol. 9, no. 3, pp. 151–166, 1977.
- [23] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.
- [24] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.
- [25] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 us election: divided they blog," in *Proceedings of the 3rd international workshop on Link discovery*, pp. 36–43, ACM, 2005.
- [26] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.