Using the Clustering Coefficient to guide a Genetic-based Communities Finding Algorithm

Gema Bello, Héctor Menéndez, and David Camacho

Departamento de Ingeniería Informática. Escuela Politécnica Superior. Universidad Autónoma de Madrid.
C/Francisco Tomás y Valiente 11, 28049 Madrid, Spain
{gema.bello,hector.menendez,david.camacho}@uam.es
http://aida.ii.uam.es

Abstract. Finding communities in networks is a hot topic in several research areas like social network, graph theory or sociology among others. This work considers the community finding problem as a clustering problem where an evolutionary approach can provide a new method to find overlapping and stable communities in a graph. We apply some clustering concepts to search for new solutions that use new simple fitness functions which combine network properties with the clustering coefficient of the graph. Finally, our approach has been applied to the Eurovision contest dataset, a well-known social-based data network, to show how communities can be found using our method.

Keywords: clustering coefficient, social networks, community finding, genetic algorithms

1 Introduction

The clustering problem is based on blind search on a dataset. Some classical solutions such as K-means (for a fixed number of clusters) [7] or Expectation-Maximization [3] (for a variable number of clusters), amongst others, are based on distances or metrics that are used to determine how the cluster should be defined. The clustering problem is harder when is applied to find communities in networks. Some algorithms such as Edge Betweenness [5] or CPM [4] have been designed to solve this problem following a deterministic process.

In our study of the previous problem, we adopt an evolutionary approach based on the K-means algorithm, a popular and well-known algorithm. It is a straightforward clustering guided method (usually by a heuristic or directly by a human) which tries to classify data in a fixed number of clusters (each element is associated to one class). The number of clusters can be predefined or can be estimated using heuristics or other kinds of algorithms, such as genetic algorithms [6].

In the process of community finding problems, K-means cannot be directly applied because it does not allow overlapping. In contrast, it is common for communities to share members. An alternative solution could be fuzzy k-means [8]

which allows every one element to belong to several clusters giving a probability of membership, so same kind of overlapping for an element can be considered.

Communities in networks have been studied using CPM (Clique percolation method) and Edge Betweenness algorithms which have been applied in our previous work [2] for community classification. CPM (Clique percolation method) [4] finds communities using k-cliques (where k is fixed at the beginning and the network is represented as a graph). It defines a community as the highest union of k-cliques. CPM has two variants: directed graphs and weighted graphs. [9] Edge Betweenness [5] is based on finding the edges of the network which connect communities and removing them to determine a good definition of these communities.

Our new approach develops an evolutionary k-means inspired by the concept of fuzzy k-means and with the same objective as CPM and Edge Betweenness algorithms: finding communities or overlapping clusters in the network.

In this work we propose a new way to combine both community finding and clustering algorithms. In our approach, a genetic algorithm is used to find communities in a dataset that represents humans voting on a social network. To guide the genetic algorithm, the fitness takes the clustering coefficient defined in graph theory to improve the results that could be obtained through a simple K-means.

The rest of the paper is structured as follows. Section 2 shows a description about the web dataset used to test our algorithm. Section 3 presents the genetic algorithm used to detect communities in the web dataset. Section 4 presents a discussion about the experimental results obtained. Finally, the conclusions and some future research lines of work are presented.

2 Genetic-based Community Finding Algorithm (GCF)

The Genetic-based community finding algorithm uses a genetic algorithm to find the best k communities in a dataset that could be represented as a graph and where any particular neighbour could belong to different clusters. To describe GCF, we will explain the following: the codification, the genetic algorithm and the fitness function definition.

2.1 GCF Codification

An important problem in any Genetic Algorithm (GA) is related to the codification of the chromosomes. In our case the genotypes are represented as a set of binary values. Each allele represents the membership of a node of the graph and each chromosome is used to represent a community. In this binary representation 1 means the node belongs to the community and 0 the opposite, see Figure 1 which exemplifies nodes as countries because of the data set to be used for experimentation.

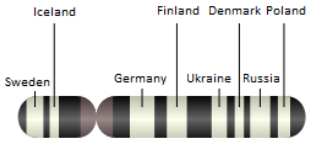

Fig. 1. A Chromosome representing graph nodes. In this case, each node represents a country and its belonging or not to the current community.

This simple codification allows us to represent nodes belonging to several communities (as we have in fuzzy k-means and CPM algorithm), and also provides a simple method to define reproduction, crossover and mutation using a standard Genetic Algorithm strategy[10].

2.2 GCF Evolutionary Approach

The GCF strategy works as follows:

1. A random population of communities is generated.
2. The population evolves using a standard GA.
3. The chromosomes that are the k-best solution of the algorithms are selected. The selection process subsumes the communities which have better fitness and belong to a bigger community. The process has the following steps:
   (a) A list of k communities is created.
   (b) The chromosomes are sorted by their fitness value.
   (c) If there is an empty position in the list or one of the members is contained in the chromosome that we are going to check, we add the checked chromosome in the mentioned position (or in an empty position) subsuming the other.
   (d) If the list is full and the chromosome that we are going to check does not satisfy the last condition, the algorithm stops. It also stops if the fitness of the new chromosome is bigger than a fixed value (in this case, the value is fixed as half of the maximum fitness).

Defining and selecting an appropriated fitness function, which we will now discuss, is the most critical issue in the GCF algorithm as it will be used to optimize the quality of communities.

2.3 GCF Fitness functions

For this problem we have implemented three kind of fitness functions, each of which has a different goal. The first one tries to find nodes with a similar rating behaviour (minimal distance fitness), the second one tries to find clusters using the clustering coefficient (maximum clustering coefficient fitness) and, finally, the

last fitness function combines both strategies trying to find communities with similar rating behaviours whose members are connected between them (hybrid fitness).

Minimal distance fitness (MDF) The objective of this fitness function is to find communities of nodes that are similar. The evaluation of this fitness function are done using the following criteria:

1. Each node belonging to a community is represented as a vector of attributes. The definition of these attributes depends on the problem being solved.
2. The average euclidean distance between vectors of attributes within a community is calculated. The fitness calculates distances to be taken into account from peer to peer, between all vectors.
3. The fitness value for the community is the average distance of the values calculated in previous step (we are trying to minimize the fitness). It is a measure of similarity for those rows, hence it checks if they follow the same ballot pattern. We call this average distance d_{in} (see Figure3).
4. Fitness penalizes those cases where the community has a single node, giving it a value of zero.

Maximum clustering coefficient fitness (MC^2F) The goal of this fitness is to discover communities whose members are connected between them. It is measured through the clustering coefficient, defined as follows:

Definition 1. *Let $G = (V, E)$ be a graph where E is the set of edges and V the set of vertices. Let $v_i \in V$ be a vertex and $e_{ij} \in E$ an edge from v_i to v_j. Let Σ_{v_i} be the neighbourhood of the vertex v_i defined as $\Sigma_{v_i} = \{v_j \mid e_{ij}, e_{ji} \in E\}$. If k is considered as the number of neighbours of a vertex, we can define the clustering coefficient of a vertex as follows:*

$$C_i = \frac{|\{e_{jk}\}|}{k(k-1)}$$

Where $|\{e_{jk}\}|$ satisfies that $v_j, v_k \in \Sigma_{v_i}$.

Definition 2. *The clustering coefficient of a graph is defined as:*

$$C = \frac{1}{|V|} \sum_{i=0}^{|V|} C_i$$

Where $|V|$ is the number of vertices.

The fitness takes the sub-graph defined by the community and calculates its clustering coefficient. It returns the inverse value, because the genetic algorithm tries to minimize the fitness function.

Hybrid fitness (HF) This last fitness function combines both Clustering Coefficient and Distance fitness ideas: it tries to find a set of communities satisfying both conditions already defined. With this method we try to find strong and similar communities (members are highly connected between them and they have similar behaviour). The function defined is a simple weighted function: suppose that $F(x, y)$ is the fitness function, CC the clustering coefficient and d_{in} the value of HF fitness is:

$$F_i(CC, d_{in}) = w_1 * \frac{CC_i}{Max(\{CC_i\}_{i=1}^{K})} + w_2 * \frac{d_{in_i}}{Max(\{d_{in_i}\}_{i=1}^{K})}$$

Where w_i are the weights given to each fitness: $w_i \in (0, 1)$. The values were set experimentally to $w_1 = 0.1$ and $w_2 = 0.9$.

3 The Dataset Description

The Eurovision Song Contest has been studied using different clustering methods since the nineties. The main interest was to study and analyse alliances between countries, which has already been reflected by clustering and communities. The data used in this work has been extracted from Eurovision's official website.

3.1 The Dataset representation: The Eurovision voting system

Since 1975, the scoring system in the Eurovision Contest consists of the following rules:

- Each country distributes among others participants the following set of points: 1, 2, 3, 4, 5, 6, 7, 8, 10, 12.
- These countries give the highest punctuation to the best song and the lower to the less popular on preferred.
- When all countries cast their votes, the final ranking is obtained and the country with the highest punctuation wins the contest.

This data can be easily represented using a graph for each year of the contest. In this graph, the vertices will be countries and the points emitted can be used to weight the edges. The graph could be *directed* (the edges represent votes), or *undirected* (the edges only connect countries which have exchanged points in any direction). If we consider the latter, it is similar to setting edge weights uniformly to 1. According to this problem, the dataset will be represented as the latter case, we named this representation Eurovision graph, or Eurovision network.

3.2 Study and comparison of the Eurovision network in a random context

The first approximation that shows patterns can be obtained using a simple comparison between the Eurovision graph and a randomly generated graph with the

same rules applied in the contest. Namely, each participant country assigns its ten set of points (generating an edge for every point cast) randomly among the remaining participant countries. We call this representation Random network.

The random network model assumes that a given country does not favours or penalize other countries and all songs have equal musical quality. So a country X will give points randomly to another ten countries. If, for example, there are N countries then the probability that country X votes for country Y is given by P = 10/(N-1). Usually, in social networks, two vertices with corresponding edges to a third vertex have a higher probability of being connected to each other. Hence, it may be possible to observe the same effect in the Eurovision network. Therefore, to study this effect it is reasonable to analyse the clustering coefficient defined in section 2.

When we compare two different graphs, Eurovision and Random graphs, a greater CC in the Eurovision graphs means there is an "intention of vote" between countries. So the graph distribution of edges is not random and we could conclude that communities, or alliances between countries, exist.

Figure 2 shows the clustering coefficients calculated for years ranging between 1992 to 2010. It can be seen how Eurovision clustering coefficients are always greater than random network values. Hence, the results provide an evidence that the voting system is not random and there are some partnerships between countries.

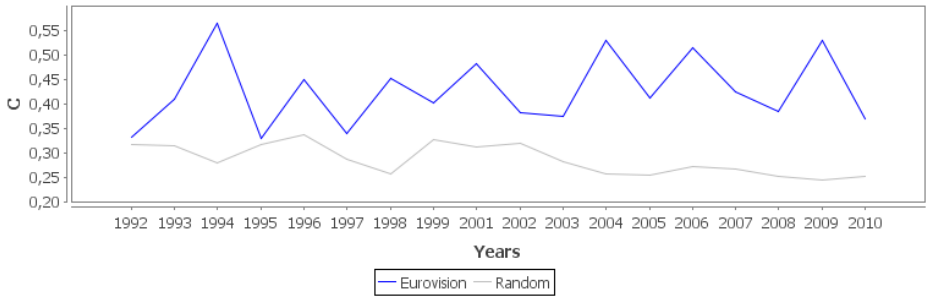

Fig. 2. Clustering Coefficient comparison between the Eurovision network and a random graphs.

4 Experimental Results

The preliminary data analysis, showed in Figure 2, confirms the existence of alliances between participant countries. Specifically, 2009 has the greatest difference in clustering coefficient. This means it contains a large set of different

communities. Hence, we have selected this year to perform the experimental analysis of our algorithm.

We have calculated the distance between the community centres to compare the results obtained; we call this measure d_{out} as shown in Figure 3. A large distance between countries is preferable as it means a bigger gap between classes or communities, and thus better results.

The genetic parameters of GCF have been set as:

- crossover probability: 0.1
- mutation probability: 0.2
- generations: 2500
- population size: 3000
- selection criteria: $\mu + \lambda$ where μ is the original population (we choose 200 best chromosomes for reproduction process and they also survive), λ is the population generated in the reproduction process
- number of communities (K): 6

K is a parameter of the genetic algorithm that sets the number of communities. Table 1 presents the communities obtained using K equal to 6 for every fitness. This value was experimentally obtained simulating different executions of our algorithm for values of K ranging between 2 and 10. The optimal number of communities with minimal overlapping was found to be 6. In the following subsections we explain the results obtained attending to each fitness.

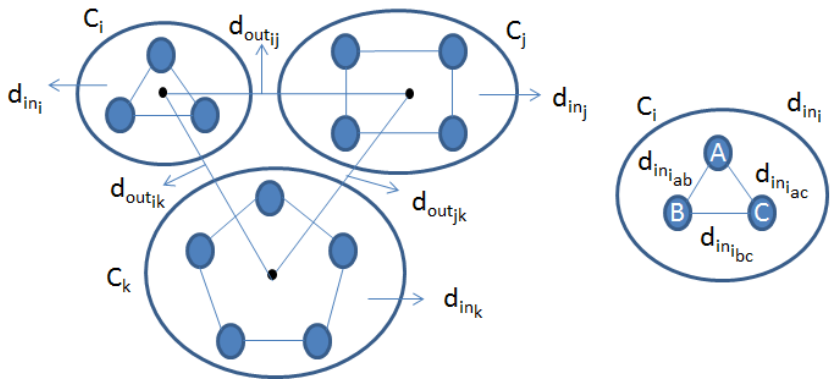

Fig. 3. Sample network graph illustrating three communities and the distances that are calculated in the experimental phase. The distance d_{in} represents the average distance calculated between the countries which belong to a community. And the distance d_{out} represents the distance between community centres.

4.1 Distance model

The first fitness, MDF, takes the minimum distances (d_{in}) between the points that represent the countries trying to find communities that vote in a similar way. This algorithm was described in section 3.2. From this experiment it can be noticed that the number of countries contained in these communities is dramatically small, as can be seen in Table 1. The d_{in} distance values obtained are lower, meaning that the communities found cast their points very similar, but all of these groups only have two countries.

4.2 Clustering Coefficient model

This model is based in the clustering coefficients of a network, and it tries to find groups of countries that they are giving votes between them. The resulting communities are shown in Table 1 identified by the fitness called MC^2F.

Analyzing the found communities, we see that many of them present high overlapping among countries. This effect was also noticed in the distance between centres (d_{out}), it has decreased dramatically from *14.65* (obtained by the previous model) to *5.40*. Therefore, the communities found are very close to each other, and present a higher overlapping.

Considering the intra-community distance, d_{in}, increases of up to twice the previous values are observed. We can conclude that we have achieved the goal of finding larger groups, but now these groups present too much overlapping to be considered as stable communities. So the final goal of the algorithm has not been really achieved.

4.3 Hybrid Model

Finally, these fitness functions have been combined in a new hybrid fitness (see previous section). The first fitness finds communities which are too small, formed by only 2 countries. The second has a good clustering coefficient and the communities are larger, but the distance between communities is not as good as in the first case, therefore overlapping is too high.

In this last model, combining the two GCF cost functions enables discovering groups of countries which cast votes in a similar way, and also exchange points between them. The communities found are shown in Table 1.

It is interesting to compare these results to the equivalent values for the previous models. The distance between centres, d_{out}, has been greatly improved and now is closer to the value obtained by the first fitness function (*11.26*). The intra-cluster distance, d_{in}, and the clustering coefficient take values lying between the first and second models' values. In addition, we found that the given communities have an appropriate size with a reduced overlapping.

This model allows us to answer two different questions about what standing closer or belonging to the same community means for a group of countries.

On the one hand, we can use the similarities in the voting process to establish relationships and, on the other hand, we can consider the points that any country assigns to the other members in its community. Therefore, we can consider that the partnerships found with this model will be stronger and more useful to measure the quality of the community found. They have similar votes and also many of these votes are exchanged between them, globally, these communities have a high number of points.

Table 1. Communities found with K = 6 using Clustering Coefficient. The distances between centres (d_{out}) obtained by fitness are: (a) MDF = 14.65 (b) MC^2F = 5.40 (c) HF = 11.26.

Fitness	**Communities**	d_{in}	**CC**
MDF	**Lithuania Latvia**	**10,91**	**0**
MDF	**Sweden Denmark**	**11,04**	**0**
MDF	**Sweden Hungary**	**11,31**	**0**
MDF	**Cyprus Moldova**	**11,40**	**0**
MDF	**Israel Netherlands**	**11,66**	**0**
MDF	**Albania Germany**	**11,83**	**0**
MC^2F	**Sweden Bosnia and Herzegovina Moldova Russia Finland Ukraine Iceland Turkey Germany**	**20,57**	**1**
MC^2F	**France Sweden Moldova Russia Finland Iceland Germany Azerbaijan UnitedKingdom**	**21,20**	**1**
MC^2F	**France Sweden Moldova Finland Romania Iceland Germany Azerbaijan UnitedKingdom**	**21,78**	**1**
MC^2F	**France Estonia Sweden Finland Iceland Germany UnitedKingdom**	**20,93**	**1**
MC^2F	**Sweden Moldova Russia Finland Ukraine Iceland Azerbaijan**	**20,55**	**1**
MC^2F	**Estonia Sweden Bosnia and Herzegovina Finland Iceland Turkey Germany**	**21,89**	**1**
HF	**Estonia Sweden Finland Iceland**	**18,03**	**1.0**
HF	**Sweden Moldova Russia Finland Ukraine Iceland**	**19,52**	**1.0**
HF	**Norway Sweden Denmark Iceland**	**18,77**	**0.92**
HF	**Moldova Russia Ukraine Poland**	**16,40**	**0.75**
HF	**Armenia Russia Lithuania Ukraine**	**16,56**	**0.75**
HF	**France Germany United-Kingdom**	**19,93**	**1.0**

5 Conclusions and Future Work

To find communities in a web dataset that can be represented by a social network, we have designed and implemented a genetic algorithm based on the graph clustering coefficient. We have centred our research around how to guide the fitness to improve the results that could be obtained through a classical K-means.

We have implemented three different fitness functions: the first one based on a euclidean distance, the second one based on the clustering coefficient, and the last one as a combination of the previous two (hybrid model).

Our experimental findings show that, using the clustering coefficient defined in graph theory to guide the hybrid fitness,is able to reach the best result. This model find communities that have an appropriate size, reduced overlapping and closer distances between centres.

Finally some improvements could be made in the fitness function. In our hybrid model, the fitness could be adapted to accept a weighted clustering coefficient[1] to obtain a better distance. This new fitness could be used in the future to measure the strength of a community. Also, for the Eurovision dataset, other features such as geographical distances or historical behaviours could be included in future fitness functions to study the analysis of the GCF algorithm.

References

1. A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, March 2004.
2. Gema Bello, Raul Cajias, and David Camacho. Study on the impact of crowd-based voting schemes in the eurovision european contest. In ACM press, editor, *1st International Conference on Web Intelligence, Mining and Semantics (WIMS11)*, May 2011.
3. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
4. Imre Derényi, Gergely Palla, and Tamás Vicsek. Clique Percolation in Random Networks. *Physical Review Letters*, 94(16):160202–1 – 160202–4, Apr 2005.
5. M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, June 2002.
6. Antonio Gonzalez-Pardo, Ana Granados, David Camacho, and Francisco D. Rodrguez. Influence of music representation on compression-based clustering. In *IEEE World Congress on Computational Intelligence*, pages 2988 – 2995. IEEE, 2010.
7. J. B. Macqueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
8. M. Oussalah and Samia Nefti. On the use of divergence distance in fuzzy clustering. *Fuzzy Optimization and Decision Making*, 7:147–167, June 2008.
9. Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June 2005.
10. Michael D. Vose. *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, Cambridge, MA, USA, 1998.