Subgroup Detection in Bipartite Graphs

The Clustering Coefficient and Community Structure in Bipartite Graphs

Bárbara Valera Muros

barbara.valera-muros@stud.uni-duisburg-essen.de

Abstract. In real-world, many networks display a natural bipartite structure of the data, which lead us to consider the bipartite networks when studying real world scenarios. The clustering coefficient is one of the most important properties in classical networks but its standard definition can only be applied in monopartite networks due to its own definition. On this paper we define a new clustering coefficient for the bipartite case and, following the same procedure, we extend the clustering coefficient as a measure based on the edges that join different nodes in a network, using that "edge clustering coefficient" as a divisive algorithm for the subgroup detection.

Keywords: squares · edge · bipartite networks · cycles · community · clustering coefficient

1 Introduction

A network is composed of a set of nodes and edges which represent the relationship between two nodes. Considering this relationship, we difference between monopartite networks, where only one type of node is involved, and bipartite ones, that describe interactions between two different types of nodes. Defining a bipartite network as a network with two non-overlapping sets of nodes where all links must have one end node belonging to each set, we could notice that in real world many networks display this structure of the data. The most remarkable examples of a natural bipartite structure are the affiliation networks (such as the actors-films network and the papers- scientists network), some biological networks (such as the metabolic network or the human disease network of genes and diseases) and the information networks (such as the word-document network). Thus, is essential to consider bipartite networks in the study of real-world situations.

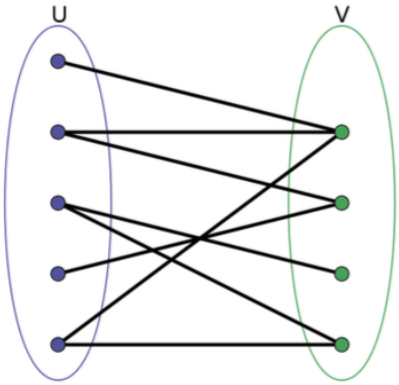

Fig. 1. Example of bipartite network with sets U and V

Regarding the clustering coefficient, it is used to measure "the cliquishness of a typical neighborhood" in the network and given by the average fraction of neighbors which are interconnected with each other. In other words, it defines the fraction of the number of observed triangles to all possible triangles in networks, where a triangle corresponds to one pair of linked neighbors. It has been used to characterize small-world networks, to understand synchronization in scale-free networks and to analyze networks of social relationships.

$$C_i = \frac{2n_i}{k_i\,(k_i - 1)} \quad . \tag{1}$$

It can be easily computed with (1) where n_i is the number of edges between the neighbors of node i and k_i is the number of neighbors that node i has.

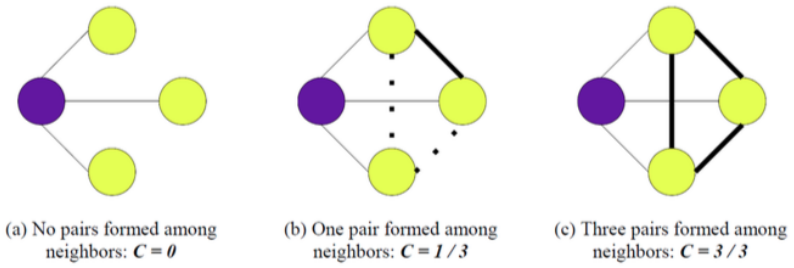

Fig. 2. Example of the clustering coefficient in a monopartite network

However, while triangles may be abundant in networks of identical nodes, they cannot be formed in bipartite networks, since two types of nodes exist and

connections link only nodes of different types. Thus, the standard clustering coefficient will be always zero and we will have to change the number of nodes of the basic clique, using squares for this kind of networks.

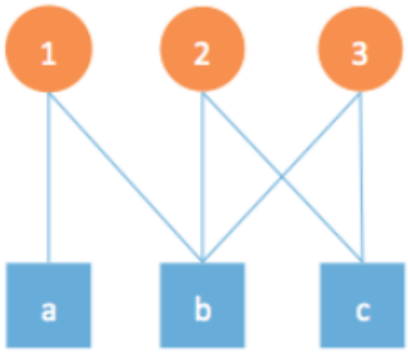

Fig. 3. Example of a bipartite network with the square as the basic clique

2 4-Size Clustering Coefficient

The basic clique in bipartite networks is a square. Thus, the clustering coefficient should in this case quantify the density of squares similarly as it does with triangles in the monopartite case. In social language, it calculates the probability of that my friends have common friends except me. Similarly to the 3-size coefficient, a cluster coefficient $C_4(i)$ with squares is the fraction between the number of observed squares and the total number of possible squares. For a given node i, the number of observed squares is given by the number of common neighbors among its neighbors, while the total number of possible squares is given by the sum over each pair of neighbors of the product between their degrees, after subtracting the common node i and an additional one if they are connected.

$$C_{4,mn}(i) = \frac{q_{imn}}{(k_m - \eta_{imn})(k_n - \eta_{imn}) + q_{imn}} \quad . \tag{2}$$

$$\eta_{imn} = 1 + q_{imn} + \theta_{mn} \quad . \tag{3}$$

In (2), we consider m and n a pair of neighbors of node i, and q_{imn} the number of squares which include these three nodes. We consider η_{imn} as expressed in (3), with $\theta_{mn} = 1$ if neighbors m and n are connected with each other and 0 otherwise.

However, there is a drawback [1] of considering the total number of possible squares, so we change the denominator of the equation, being the final equation as shown in (4).

$$C_{4,mn}(i) = \frac{q_{imn}}{(k_m - \eta_{imn}) + (k_n - \eta_{imn}) + q_{imn}} \quad . \tag{4}$$

In order to demonstrate it, we consider the network in Fig. 4, where m and n are the neighbors of node i.

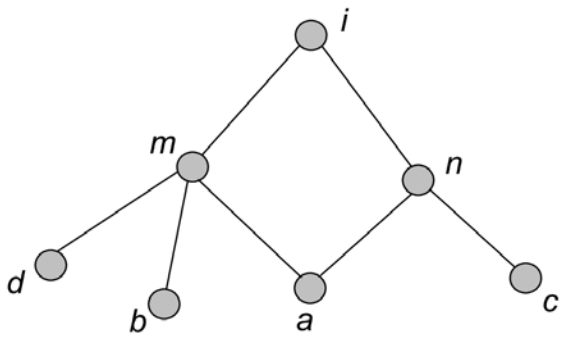

Fig. 4. Example of a network to demonstrate (4)

We could clearly see that the number of possible squares is 4 (*iman*, *imbn*, *imdn* and *imcn*), but with (3) we get $q_{imm} = 1$ square and $k_m = 4$, $k_n = 3$, $\theta_{mn} = 0$, resulting a denominator of 3, which is wrong. From now on we will consider (4) as the definition of the clustering coefficient in bipartite networks, $C_4(i)$. While $C_3(i)$ gives the probability that two neighbors of node i are connected with each other, $C_4(i)$ is the probability that two neighbors of node i share a common neighbor (different from i).

2.1 Sexual Contact Study

As an example, we applied both coefficients C_3 and C_4 to analyze two real networks of sexual contacts [2] [3], in which the nodes represent the people and the links are the sexual contacts between them.

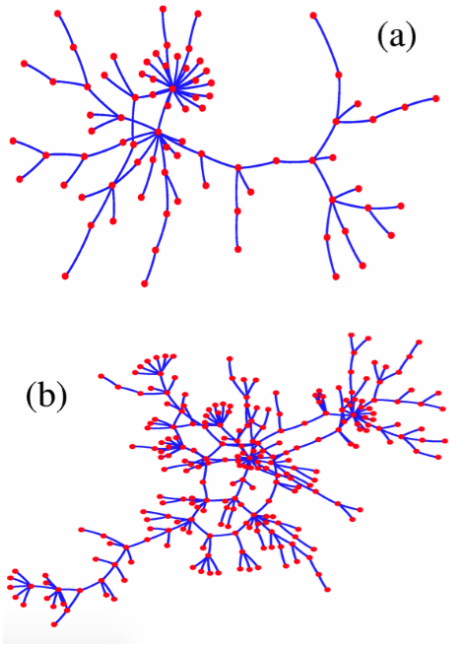

Fig. 5. Sexual Contact Study Networks

In the example, the monopartite network will be the homosexual one shown in Fig.5 (b) (N = 250 nodes and L = 266 connections) and the one with heterosexual contacts shown in Fig.5 (a) (N = 82 nodes and L = 84 connections) will be bipartite. We can observe that the bipartite has some squares but no triangles and the mono-partite has both of them.

Table 1. Sexual Contact Study Results

	N	L	T	Q	C_3	C_4
Heterosexual (Fig.5 a)	82	84	0	2	0	0.00486
Homosexual (Fig.5 b)	250	266	11	6	0.02980	0.00192

The results for both networks are shown in table 1, where we find that although the heterosexual network has less squares than the homosexual one due to its smaller size, C_4 is much larger in that case. The reason of this result is the definition of the coefficient itself, since C_4 is defined as the number of observed squares divided by the number of possible ones and in the bipartite case only nodes of different type can be connected, so the number of possible squares

is much lower. With this study, it's also concluded that this second clustering coefficient enables one to quantify the cliquishness in bipartite networks where triangles are absent.

3 Communities Detection

Qualitatively, a community is defined as a subset of nodes within the graph such that connections between the nodes are denser than connections with the rest of the network. Thus, in a bipartite network the basic observation on which the community definition relies is that a typical community consists of several complete sub-bigraphs that tend to share many of their nodes, and that will be the main principle when detecting communities within a given network [4].

Similar to classical networks, community structure of bipartite networks is the groups of nodes as we can observe in Fig. 6, where there are three communities of dense internal links (solid lines), with sparse connections (dot lines) between them.

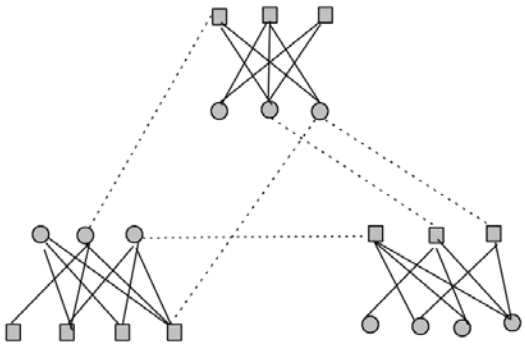

Fig. 6. Schematic representation of a bipartite network with community structure

3.1 Edge Clustering Coefficient

The edge clustering coefficient is defined in analogy with the usual node-clustering coefficient as the number of triangles to which a given edge belongs, divided by the number of triangles that might potentially include it, given the degrees of the adjacent nodes. Based on the edge clustering coefficient, a divisive algorithm is proposed [5]. The idea behind the use of this quantity in a divisive algorithm is that edges connecting nodes in different communities are included in few or no triangles, and tend to have small values of edge clustering coefficient. On the other hand many triangles exist within clusters. Here we also can define the edge-clustering coefficient of bipartite networks, as the number of squares to which

a given edge belongs,divided by the number of squares that might potentially include it.

For the edge-connecting top node i to bottom node j, the edge-clustering coefficient C_{ij} is calculated according to (5), where m is a neighbor of node i, and n is one of j's neighbors; $q_{ijmn} = 1$ if neighbors m and n are connected with each other and 0 otherwise.; θ_{ijmn} is opposite to q_{ijmn}; k_m is the degree of node m and k_n is the degree of node n.

$$C_{ij} = \frac{\sum_{m=1}^{k_i} \sum_{n=1}^{k_j} q_{ijmn}}{\sum_{m=1}^{k_i} \sum_{n=1}^{k_j} \theta_{ijmn} \left(\sum_{m=1}^{k_i} k_m - 1\right) + \left(\sum_{n=1}^{k_j} k_n - 1\right) - \sum_{m=1}^{k_i} \sum_{n=1}^{k_j} q_{ijmn}} . \tag{5}$$

This algorithm works as the GN algorithm [1], however, the edge with the smallest value of C_{ij} should be cut at each step.

3.2 Algorithm Implementation

The code for the divisive algorithm based on the edge clustering coefficient, attached in the appendix, was implemented for R language. The main purpose is to detect subgroups in bipartite networks by clustering the networks. It is composed basically by two methods: one to get the edge clustering coefficient for all the edges and another to compare those coefficients, eliminate the edge with the minimum value and repeat this procedure until the number of maximum clusters is reached.

Therefore, in order to use it properly, the number of clusters in which the network is going to be divided should be decided previously, taking into account that the fewer the clusters the faster the algorithm will end. As a recommendation, the maximum number of clusters should be 4 or 5, considering also the dataset used since the algorithm calculates the coefficient for all the edges, eliminate the lowest one and then repeats all the computation from the beginning, which takes a long execution time (specially for datasets with several number of nodes or in case we want to divide the network in several clusters).

In consequence, we consider the execution time of the algorithm its main drawback. During the research, different datasets were proposed, such as the author-page from Wikipedia and the Marvel comic characters and their appearance in comic books, but it was due to this drawback that we focused on the largest connected component to reduce them, since they were too dense and the networks were not connected. On the other hand, in order to present some results and compare this method with the divisive algorithm based on biclique communities, the example of Davis's Southern Club Women was use as reduced dataset.

[1] The Girvan-Newman algorithm is a hierarchical method used to detect communities in complex systems based on the edge betweenness

```
  [1] 1 2 2 1 2 2 2 2 1 2 1 2 2 2 1 2 2 1 2 1 2 2 1 2 2 2 1 1 2 1 2 2 1 2 1 1 2 1 1 2 2 2 1 2 2 2 1 1 2 1 2 2 1 2 2
 [56] 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
[111] 2 2 2 2 2 2 2 2 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1
  [1] "ANGEL/WARREN KENNETH" "ANT-MAN/DR. HENRY J." "BEAST/HENRY &HANK& P" "BLACK KNIGHT V/DANE "
  [5] "BLACK PANTHER/T'CHAL" "BLACK WIDOW/NATASHA " "CAPTAIN AMERICA"      "CAPTAIN MARVEL II/MO"
  [9] "COLOSSUS II/PETER RA" "CRYSTAL [INHUMAN]"    "CYCLOPS/SCOTT SUMMER" "DAREDEVIL/MATT MURDO"
 [13] "DR. DOOM/VICTOR VON " "DR. STRANGE/STEPHEN " "DRAX/ARTHUR DOUGLAS"  "FALCON/SAM WILSON"
 [17] "GALACTUS/GALAN"       "HAVOK/ALEX SUMMERS "  "HAWK"                 "HERCULES [GREEK GOD]"
 [21] "HULK/DR. ROBERT BRUC" "HUMAN TORCH/JOHNNY S" "ICEMAN/ROBERT BOBBY " "INVISIBLE WOMAN/SUE "
 [25] "IRON MAN/TONY STARK " "JARVIS, EDWIN "       "LIVING LIGHTNING/MIG" "MARVEL GIRL/JEAN GRE"
 [29] "MOCKINGBIRD/DR. BARB" "MOONDRAGON/HEATHER D" "MR. FANTASTIC/REED R" "NIGHTCRAWLER/KURT WA"
 [33] "NOVA/RICHARD RIDER"   "PROFESSOR X/CHARLES " "PSYLOCKE/ELISABETH B" "QUASAR III/WENDELL V"
 [37] "QUICKSILVER/PIETRO M" "ROGUE /"              "SASQUATCH/WALTER LAN" "SCARLET WITCH/WANDA "
 [41] "SERSI/SYLVIA"         "SHE-HULK/JENNIFER WA" "SILVER SURFER/NORRIN" "SPIDER-MAN/PETER PAR"
 [45] "SPIDER-WOMAN II/JULI" "STARFOX/EROS"         "STORM/ORORO MUNROE S" "STRONG GUY/GUIDO CAR"
 [49] "SUB-MARINER/NAMOR MA" "THANOS"               "THING/BENJAMIN J. GR" "THOR/DR. DONALD BLAK"
 [53] "THUNDERSTRIKE/ERIC K" "TIGRA/GREER NELSON"   "USAGENT/CAPTAIN JOHN" "VISION "
 [57] "WARLOCK II/ADAM WARL" "WASP/JANET VAN DYNE " "WOLVERINE/LOGAN "     "WONDER MAN/SIMON WIL"
 [61] "COC 1"                "H2 Z78"               "M/SHSW 1"             "M/SHSW 2"
 [65] "M/SHSW 3"             "M/SHSW 4"             "M/SHSW 5"             "M/SHSW 6"
 [69] "M/SHSW 7"             "M/SHSW 8"             "M/SHSW 9"             "M/SHSW 10"
 [73] "M/SHSW 11"            "M/SHSW 12"            "ROM 65"               "COC 3"
 [77] "H2 279"               "FF@ 22"               "DAMCON3 4"            "A 3"
 [81] "MARVELS 2"            "ASM@ 1"               "UTSM '97"             "FF 36"
 [85] "FF@ 3"                "ST 156"               "A 60"                 "A 151"
 [89] "M/TE 13"              "COH 3"                "A Z11"                "M/GN 1"
 [93] "DAZZ 21"              "IW 1"                 "IW 2"                 "MK3 41"
 [97] "NW 27"                "Q 38"                 "FF 368"               "IW 3"
[101] "WIW 8"                "IW 4"                 "FF 369"               "AF 111"
[105] "Q 39"                 "IW 5"                 "WM2 15"               "IW 6"
[109] "FF 370"               "X 55"                 "H2 323"               "WCA@ 1"
[113] "A 305"                "A 308"                "A@ 18"                "A@ 19"
[117] "A 329"                "IG 2"                 "A 332"                "A 345"
[121] "CA 401"               "FF 367"               "WM2 13"               "A 400"
[125] "O:X"                  "O:MU"                 "HR:R 4"               "A3 1"
[129] "A3 2"                 "A3 3"                 "A3 4"                 "A3 10"
[133] "A3 11"                "FF3 27"               "MAXSEC 2"             "MAXSEC 3"
[137] "M/SS 1"               "A 181"                "M/TIO 96"             "H2 277"
[141] "SWII 9"               "H2 321"               "FW 10/2"              "SWII 5"
[145] "A 118"                "A 333"                "IG 3"                 "M/TIO@ 7"
[149] "IG 4"                 "IG 6"                 "WM2 14"
```

Fig. 7. Results of Marvel dataset after applying the ECC algorithm

Regarding the reduced dataset of Marvel comic characters and their appearance in comic books, it will have 151 nodes in one set and 60 in the other, and between them 2159 edges in total. For that reason, it was discarded as a possibility in the analysis and comparison between algorithms. Although, it was computed for the division in two clusters and the execution time was 8 hours. The results are shown in Fig. 7, as the output of the algorithm where we can find two clusters (1 and 2) and the nodes ordered following the sequence of clusters previously displayed: in our example "ANGEL/WARREN KENNETH" is part of the first cluster while "ANT-MAN/DR. HENRY J." and "BEAST/HENRY &HANK& P" are on the second one and again "BLACK KNIGHT V/DANE". Continuing with the order of the results our network will be completely divided in two different clusters.

Considering this, we can clearly conclude that the algorithm is only efficient in cases with small datasets, as the Davis's Southern Club Women (Fig. 8), that took only 8 seconds for its execution time. In this second dataset, the nodes represent observed attendance at 14 social events by 18 Southern women. As a test of the efficiency regarding the number of clusters, we can study the execution

time for different number of clusters, as shown in table 2, with the division in Fig. 9-13.

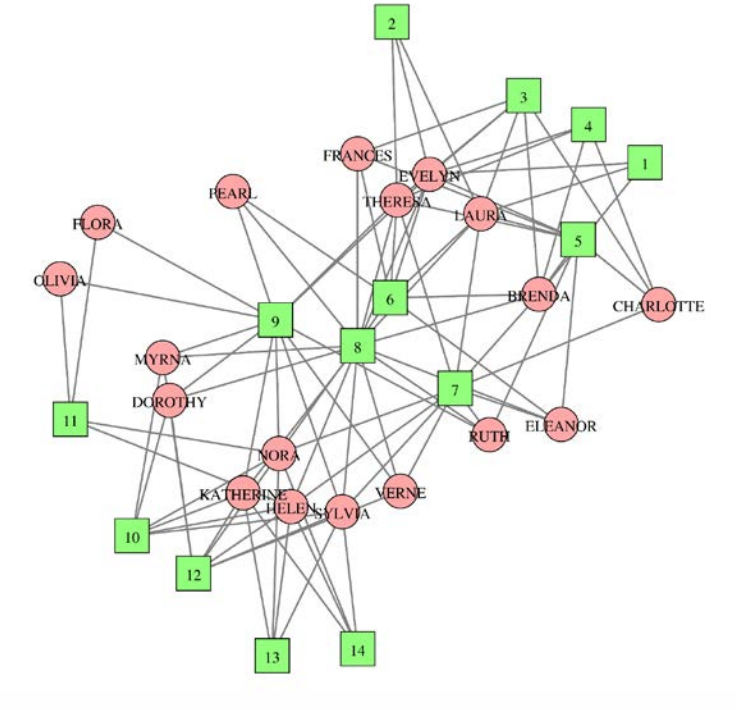

Fig. 8. Davis's Southern Club Women Network

Table 2. Davis's Southern Club Women execution time for different clusters

Clusters	2	3	4	5	6	7
Time (s)	8	11	11.68	12.70	13.16	13.16

In conclusion, the execution time of the algorithm increases briefly with the number of clusters in which we want to divide our network, but notably with the size of the dataset. Another finding is that when we try to divide this network in six clusters the algorithm is unable to do it due to the structure of the network itself, using seven clusters and taking in consequence the same execution time for 6 and 7 clusters.

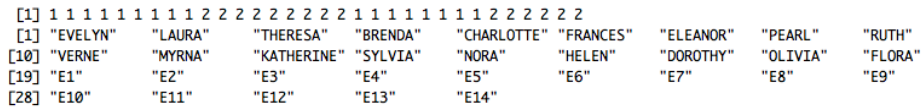

Fig. 9. Results of Davis dataset for 2 clusters

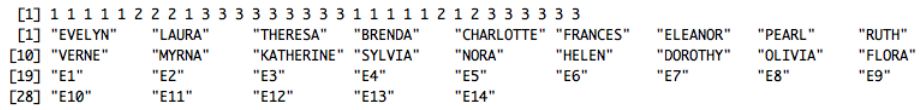

Fig. 10. Results of Davis dataset for 3 clusters

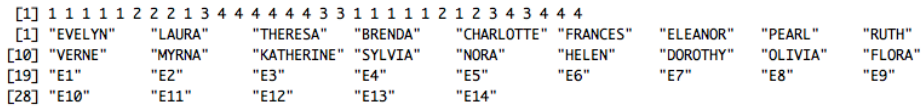

Fig. 11. Results of Davis dataset for 4 clusters

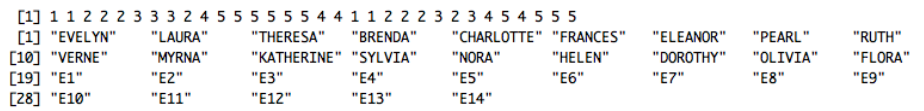

Fig. 12. Results of Davis dataset for 5 clusters

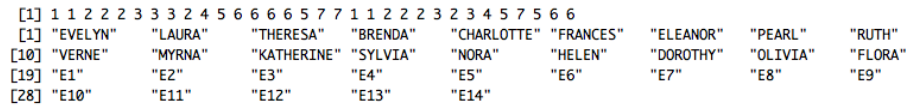

Fig. 13. Results of Davis dataset for 6 and 7 clusters

3.3 Biclique Communities

Here, a method for detecting communities in bipartite networks based on an extension of the k-clique community detection algorithm is presented. Mainly,

the clique percolation method builds up the communities from k-cliques, which correspond to complete (fully connected) sub-graphs of k nodes, considering two k-cliques adjacent if they share $k-1$ nodes. A community is defined as the maximal union of k-cliques that can be reached from each other through a series of adjacent k-cliques. As shown in red in Fig. 14, overlaps between the communities (color coded) are allowed [6].

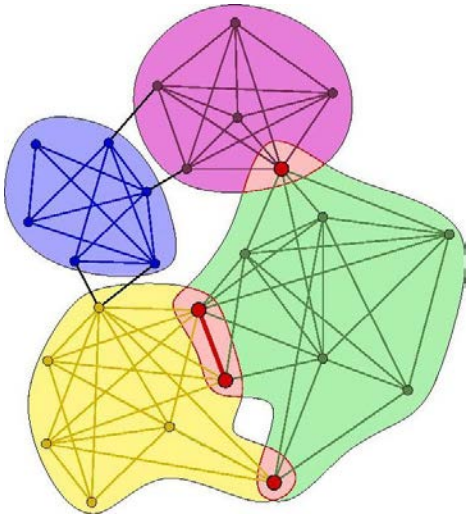

Fig. 14. Overlapping communities obtained with the Clique Percolation Method

The difference with the monopartite networks is that in this case the basic clique will have four nodes instead of three, but as it happened in the monopartite networks, when we use an algorithm based on cliques, it will overlap. This is the main difference with the clustering coefficient method. The other difference is that in a biclique community, when a node is out of a clique it won't be part of the community even if it's part of the network or it's connected to some of the nodes (but not enough to complete a clique). In Fig. 15 we find the nodes out of the communities in green, and the two different communities in red, overlapping in nodes 2 and b. An important feature of the biclique community approach is that the biclique method provides an immediate context to the communities that are detected. In this sense, the bipartite community information is more valuable that the one obtained by finding structure in the two monopartite projections because it provides specific links between the communities that are present in the two node sets [7].

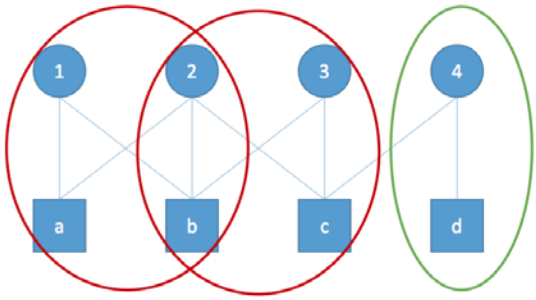

Fig. 15. Example of biclique communities

Comparison with the Edge Clustering Coefficient Algorithm If we apply the biclique division algorithm in Davis's Southern Club Women network, the communities are organized as in Fig. 16, where the orange and blue nodes represent the two different clusters in which the network has been divided, the black ones represents the overlaps and the white ones are the nodes out of clusters. As mentioned before, these two are the main differences with the edge clustering coefficient, shown in Fig. 17 where the results from Fig. 9 are applied to the previous graph in order to have two communities completely separated without black or white nodes. In this second case, considering the order of Fig. 9, the first 9 nodes (from Evelyn to Ruth) will be part of the first cluster with the first 8 events, while the second cluster will be composed by the other 9 nodes (from Verne to Flora) and the last 6 events, being clear that now all the nodes are included in the clusters and that they are part only of one of them.

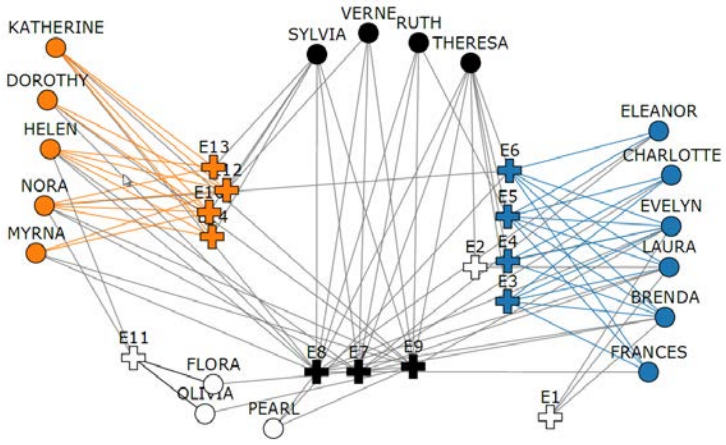

Fig. 16. Results of Davis dataset after applying the biclique communities algorithm

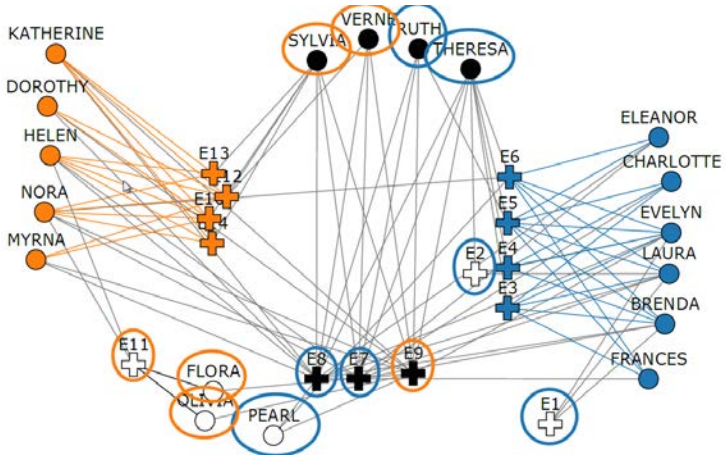

Fig. 17. Results of Davis dataset after applying the ECC algorithm

4 Conclusions

We have implemented a new algorithm based on the edge clustering coefficient for bipartite networks in which the basic cycle is a square instead of a triangle. In our case, we have adapted the standard clustering coefficient for monopartite networks in order to use it in bipartite ones, and this new 4-size clustering coefficient enables us to quantify the cliquishness in networks where triangles are absent. Thus, one should take triangles and squares simultaneously as the two basic cycle units in any network. Similarly, this clustering coefficient was adapted to the edges instead of the nodes and finally to the edges of bipartite networks.

As in the monopartite case, we can find different methods for the detection of subgroups, and the behavior of all of them will be similar as in the monopartite case. Here, we find the biclique method, that will present overlapping clusters and nodes that are out of all the clusters; and specially the edge clustering coefficient, that will divide the network in completely separated communities, with the inconvenience of the time of execution due to the amount of calculations to be performed in cases where the datasets are too dense.

Finally, we would like to mention that in the analysis of networks we have to consider our goals, being the edge clustering coefficient accurate when we just want to divide completely a network but the biclique method when we are studying the special cases, since the ECC eliminates all the variations organizing them in the different available clusters while the biclique communities could be useful in the study of the cases out of the norm.

A Algorithm Code

```
require(igraph)

# Calculates the separatedness of clusters (See http://arxiv.org/pdf/0707.1616v3.pdf).
# This can be used as a criterion instead of a fixed number of communities.
bipartiteModularity <- function(g, s) {

  nodesType0 <- which(V(g)$type)
  nodesType1 <- which(!V(g)$type)
  adj <- get.adjacency(g)

  deg <- rowSums(adj)

  sum(sapply(nodesType0, function(i) {
    sapply(nodesType1, function(j) {
      if (s[i] == s[j]) {
        adj[i,j] - ((deg[i] * deg[j]) / length(E(g)))
      } else {
        0
      }
    })
  })) / length(E(g))
}

getEdgeClusteringCoefficients <- function(graph) {

  # For all edges
  sapply(E(graph), function(edgeId) {

    # Compute the edge clustering coefficient
    edge <- get.edge(graph, edgeId)
    n1 <- edge[1] # First incident node of the edge
    n2 <- edge[2] # second incident node of the edge
    #####

    # The number of common neighbors of the neighbors of n1 and n2
    # correspond to the number of squares the edge is contained in.
    q=0 # Qijmn
    v1 <- neighbors(graph, n1) #m
    v2 <- neighbors(graph, n2) #n

    nv1 <- unlist(neighborhood(graph, 1, v1))

    # Number of observed squares
```

```
    q <- length(which(nv1 %in% v2))
    # Number of possible squares
    possibleQ <- length(v1) * length(v2)
    # ECC
    q / possibleQ
  })
}

# Compute bipartite clusters using the edge clustering coefficient approach.
# graph: Bipartite graph
# maximumNumberOfClusters: Maximum number of clusters when the algorithm
# should stop decomposing the graph.
# (Bipartite modularity could be used instead)
getBipartiteClusters <- function(graph, maximumNumberOfClusters) {

  # Calculate connected components
  components <- clusters(graph)

  while (components$no < maximumNumberOfClusters) {
    print(components$no)
    # Calculate edge clustering coefficients
    ecc <- getEdgeClusteringCoefficients(graph)

    # Remove edge with lowest clustering coefficient
    edgeToRemove <- which(ecc == min(ecc))
    graph <- delete.edges(graph, edgeToRemove)

    components <- clusters(graph)
  }

  # Extract and return cluster membership vector.
  # Each connected component of the graph correspond to one cluster.
  # Each element corresponds to a node in the graph and the value
  # corresponds to the cluster of the node.
  components$membership
}


# Start the script call: start()
start <- function() {

  g <- read.graph("<Path to the graph file>", "pajek")
  print(getBipartiteClusters(g, <number of communities>))
  print(V(g)$id)
}
```

References

[1] Zhang et al.: The clustering coefficient and community structure of bipartite networks. (2013)
[2] Lind, P. G., González, M. C., Herrmann, H. J.: Cycles and clustering in bipartite networks. (2004)
[3] Barber, M. J.: Modularity and community detection in bipartite networks. (2008)
[4] Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. (2002)
[5] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. (2004)
[6] Palla, G., Derényi, I., Vicsek, T.: The critical point of k-clique percolation in the Erdös-Rényi graph. (2006)
[7] Lehman, S., Schwarz, M., Hansen, L. K.: Biclique Communities. (2008)